A Cross-Platform Multi-physics Reactor Transient Simulation Framework through a SALOME-Backbone Joint Platform

Yu Liu[1], Tao Xu[2], Marat Seydaliev[1], Boone Tensuda[3]

[1] Fuel and Fuel Channel Safety Branch, Canadian Nuclear Laboratories, Chalk River, Ontario, Canada K0J 1J0

[2] Thermalhydraulics Branch, Canadian Nuclear Laboratories, Chalk River, Ontario, Canada K0J 1J0

[3] Institute for Aerospace Studies, University of Toronto, Toronto, Ontario, Canada M3H 5T6

yu.liu@cnl.ca; tao.xu@cnl.ca; marat.seydaliev@cnl.ca; boone.tensuda@mail.utoronto.ca

ABSTRACT – Predicting nuclear reactor behaviour during transient conditions can be enhanced through numerical multi-physics simulation framework involving neutron transport, thermalhydraulics, and fuel behaviour codes to improve nuclear reactor safety analysis. Two tools are used in this work to accomplish the coupling of these codes. One is SALOME, an open source software that provides a generic platform for Pre- and Post-Processing of numerical simulations. Another one is Backbone, developed at the Canadian Nuclear Laboratories, which has the capability of coupling multi-physics codes under different operating systems. Also, Backbone is capable of synchronizing various scientific codes through a Time Step Controller (TSC), which permits appropriate and efficient data exchange at the time-step level. This paper studies a SALOME-Backbone joint platform for code coupling with comparisons made to a Large Break Loss of Coolant Accident (LB-LOCA) experiment performed in a five channel per pass configuration at the CNL Whiteshell RD-14M facility.

Keywords: Code Coupling, Thermal-hydraulics, Neutron Transport, Backbone, SALOME

1. Introduction

Multi-scale multi-physics simulations play an increasingly important role in predicting the behaviour of nuclear reactors. Recently, the availability of affordable high performance computing resources has increased. Thus, reactor transient predictions can be enhanced via a multi-physics simulation framework that couples neutron transport, thermalhydraulics, and fuel behaviour with the overall goal of improving performance and safety. This requires capturing many physical interactions between complex phenomena through rigorous coupling at the time step level [1]. Therefore, multiple computer simulations of different scientific disciplines must exchange data during run-time to address these inter-dependencies in a distributed system, which may also use heterogeneous environments.

Although many tools or platforms have been developed for code coupling (e.g., MOOSE [2], LIME [3], SALOME [4], Backbone [5]), the capabilities and intended applications of each one differ. For example, Backbone, developed at the Canadian Nuclear Laboratories (CNL), has the capability of synchronized coupling of various codes through a Time Step Controller (TSC), which permits appropriate and efficient data exchange at the time step level. However, Backbone has a limited Pre- and Post-Processing suite of capabilities. SALOME [4], on the other hand, is an open-source code that provides a generic platform for Pre- and Post-Processing of numerical simulations, but it does not support cross-platform code coupling (e.g., one code may be running on Linux, while another one will be operating on Windows) and the TSC by default. At CNL, the thermalhydraulics code - Canadian Algorithm for Thermalhydraulic Network Analysis (CATHENA) [6] and Plant Controller were developed under the Windows operating system (OS), which is intended to be coupled with BISON [7], which was developed for Linux. For many existing or legacy codes developed under different OS platforms, coupling becomes a rather difficult task. One approach is to port all the codes into the same platform. This approach will involve additional cost in terms of code licensing and effort that can be prohibitive. Conversely, Backbone completely supports code interactions developed on different operating systems, which is not supported by other platforms like MOOSE, LIME and SALOME. Thus, a joint platform of Backbone and SALOME (called the SALOME-Backbone joint platform) provides:

- a cross-platform code coupling capability through a mixed MPI-CORBA approach,
- Backbone’s TSC,
- a Pre- and Post-Processing module, and
- a job management module for numerical simulation.

Co-ordination and synchronization of data exchange time intervals between coupled codes in the SALOME-Backbone joint platform is an essential and complex task. On the contrary, SALOME does not have a TSC module to synchronize the coupling of various codes. Thus, it is quite advantageous to integrate Backbone into the SALOME platform, which provides an efficient TSC available for code coupling. Centralized TSC is integrated into Backbone, which offers a few different synchronization time interval selection strategies. A number of classes of TSC are outlined by Soderlind [8], which provide the basis for configurable Proportional Integral Derivative (PID) TSC and General Third Order TSC.

SALOME requires integration of individual codes by developing them into a SALOME compatible module. In contrast, Backbone only requires an interface wrapper for stand-alone codes to be integrated into coupling schema. The comparison of SALOME, Backbone and SALOME-Backbone is given in Table 1 below.

Table 1 Comparison of the Coupling Platforms

Coupling Platform / Capabilities	SALOME	CNL Backbone	SALOME-Backbone
Cross-platform Capability	No	Yes	Yes

Pre- and Post- Processing	Yes	No	Yes
Implementation	Integration	Stand-alone	Stand-alone

The following two sections describe the infrastructure and design strategies of the SALOME-Backbone joint platform and the corresponding TSC. The rest of the paper demonstrates this code coupling capability through an example involving CATHENA and the neutron transport code 3D-Solver. In this study, the authors used a General Third Order TSC to synchronize CATHENA with 3D-Solver. The simulation results are then compared to the experimental data. The experimental test case data used in this evaluation is taken from a Large Break Loss of Coolant Accident (LB-LOCA) experiment performed in a five channel per pass configuration at the CNL Whiteshell RD-14M facility.

2. General Infrastructure

2.1 SALOME

SALOME was developed as a distributed objects architecture and it is based on the distributed Common Object Request Broker Architecture (CORBA) components [9]. It has two layers: the lower layer and a modules layer. The lower layer contains the kernel and GUI (Graphic User Interface). The kernel is the core part of SALOME, and it provides the communication between the distributed modules and the core functionalities of this platform. The modules layer includes many distributed modules that are necessary for the general functionalities of SALOME. Additionally, users can develop their own user defined modules to extend the functionality of SALOME. The general architecture representation of SALOME is shown in Figure 1.

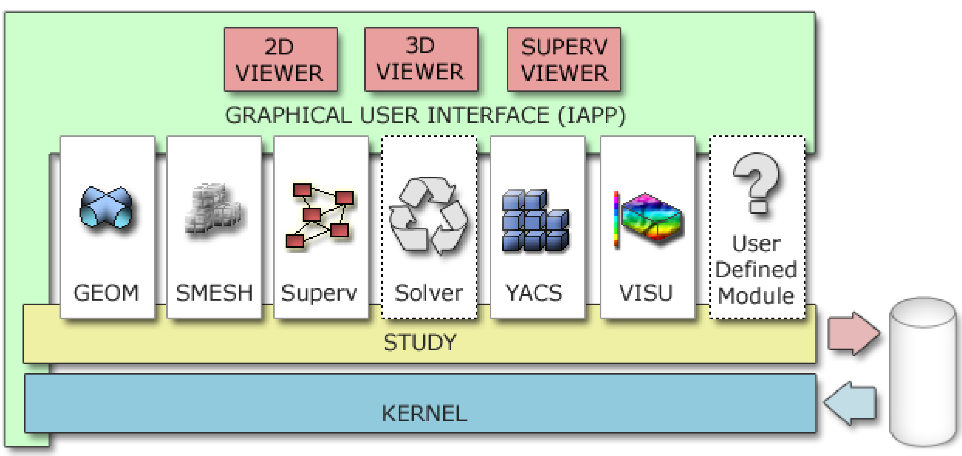

Figure 1 General Architecture of the SALOME Platform [9].

SALOME modules, which are represented by the row of boxes in the middle of Figure 1, can be extended by developing modules specific to each scientific discipline of coupling codes. These disciplines may include neutron transport (e.g., NESTLE [10], SERPENT [11]), thermalhydraulics (e.g., CATHENA, CATHARE [12]), and fuel performance (e.g., BISON).

2.2 Backbone

Backbone design is based on a combination of the CORBA [13] standard for distributed applications and the Message Passing Interface (MPI) [14] standard for parallel applications. Such a combination gives a number of advantages. First of all, Backbone treats a code almost as black box, while in most other multi-physics code coupling, knowledge of the internal structure of the code and access to the source code is required in order to perform the decomposition of the program. This provides greater flexibility on the programmer's part. Second, codes that are coupled in such an environment do not have to be CORBA objects -- they are either pure MPI-based codes or they may use an MPI-based wrapper in order to communicate with Backbone using MPI. Since MPI's main strength is to enable parallelism and CORBA is designed for distributed programming, the goal was to create a server that can run parallel programs in a distributed heterogeneous environment using CORBA's client-server approach.

2.3 SALOME-Backbone

Figure 2 illustrates the detailed infrastructure of the SALOME-Backbone joint platform. SALOME acts as the front-end of this joint platform, while Backbone is the back-end. Two SALOME modules are involved this joint platform. The Dynamic Parallel Coupling Supervisor (YACS) module allows building, editing and executing calculation schemes. The Backbone module is a SALOME user defined module, which was developed for integrating Backbone within SALOME. A calculation scheme set by the YACS module defines a chain or a coupling scenario of computer codes. Then, SALOME calls Backbone module to launch the codes being coupled according to the calculation scheme though Backbone.

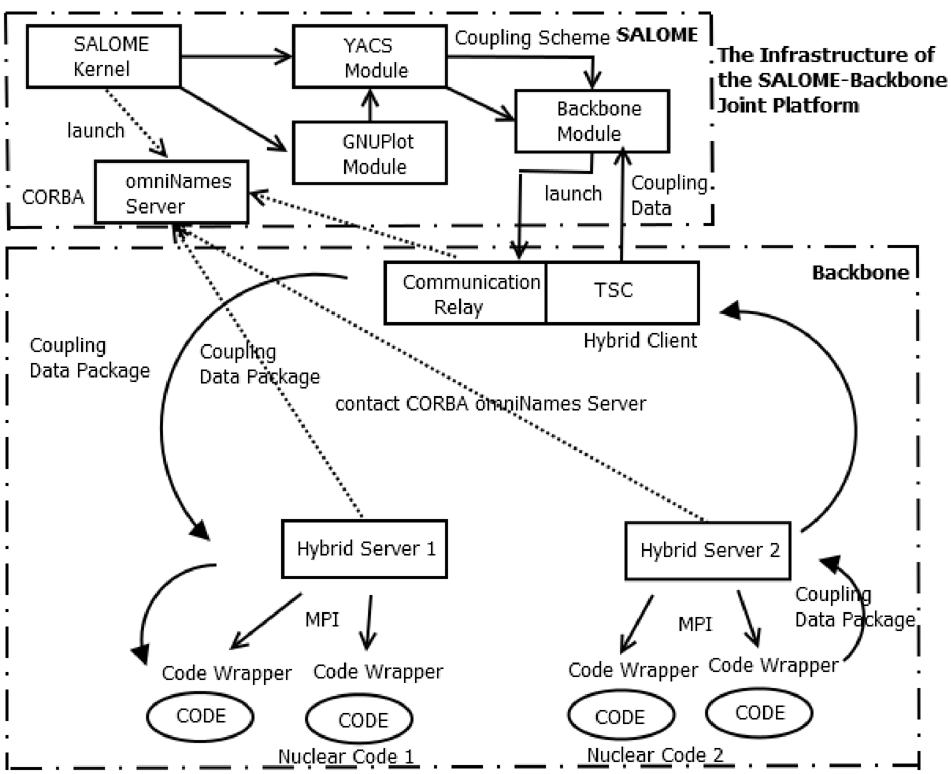

Figure 2 The Programming Infrastructure of the SALOME-Backbone Joint Platform.

In this framework, scientific codes are spawned by Backbone as separate processes and then the data exchange between them is coordinated by the TSC. During the progress along the data exchange time interval, Backbone checks for acceptability of data. If the data does not meet acceptability criteria (i.e., due to unacceptable error, failure to converge or other similar problems), Backbone resets data to the start of the time step, otherwise it proceeds to the next time interval. In case of a failure of the acceptability test, a code checks if it needs to issue a reset to all other codes, and does so if necessary and discards its data for the current time-step. Otherwise, it transfers its data and waits for the next message from Backbone, which will either be a reset of the time-step for a shorter one, or a request to execute another time-step. In addition, the coupling data among codes is relayed through Backbone, however, the coupling data is also sent to SALOME in order to be displayed by the GNUPlot module in the runtime.

3. Time Step Controller of SALOME-Backbone Joint Platform

Backbone is a framework designed to allow the coupling of disparate simulation codes through well-defined interfaces. The data structure being exchanged contains arrays of double-precision floating point numbers. A code accepts an array of numbers, executes, and then returns other arrays of numbers representing the parts of its state intended for another code, which then executes in turn using that information. With the addition of a TSC, Backbone offers superior control of errors, improved robustness (i.e., dynamic adjustment to chaotic or stable patches of simulation) and efficiency (i.e., decelerating over periods of rapid change or acceleration over stable periods). The TSC is an essential component of Backbone that ensures proceeding of coupled codes only with the data for their next time-step, and limiting estimated error of those codes for their various parameters. It also makes configuring time steps for a coupled simulation a simple run-time decision.

TSC supports a number of strategies to govern time step selection based on user specified inputs. It is able to employ three different time step selection strategy:

- simple controller,
- PID controller, or
- General Third Order controller.

These three controllers all have distinctly different origins and behaviours. The simple controller is a heuristic one; it increases the step size after each successful time-step (i.e., a time step accepted by the code based on convergence criteria), and decreases it in case of an unsuccessful time-step (i.e., a time step rejected by the code also based on the same convergence criteria). The amount by which the time step increases or decreases is pre-defined by the user and is assigned a fixed value. This yields a TSC that increases the data exchange time step size very slowly and smoothly, and decreases at a moderate rate. In contrast, the PID and General Third Order controllers are more complex configurations and are able to adapt fairly aggressively to both small and large changes in the numerical system with an optimal time step size. The PID controller is defined by Equation 1,

$$ts_{i+1} = \left(\frac{err_{i-1}}{err_i}\right)^{b_1} \left(\frac{TOL}{err_i}\right)^{b_2} \left(\frac{(err_{i-1})^2}{err_i(err_{i-2})}\right)^{b_3} ts_i \quad (1)$$

and the General Third Order controller is defined by Equation 2 [5] below.

$$ts_{i+1} = \left(\frac{(TOL)(\gamma)}{err_i}\right)^{b_1} \left(\frac{TOL(\gamma)}{err_{i-1}}\right)^{b_2} \left(\frac{TOL(\gamma)}{err_{i-2}}\right)^{b_3} \left(\frac{ts_{i-1}}{ts_i}\right)^{a_1} \left(\frac{ts_{i-2}}{ts_{i-1}}\right)^{a_2} ts_i \quad (2)$$

where *TOL* is error tolerance (user defined), γ is the safety factor, usually between 0.8 and 1, ts_i is the size of the i^{th} time step, err_i is the magnitude of error in the i^{th} step (specifically, the greatest magnitude of error for all coupled variables of all codes, and error is usually the convergence error in the code), a_n and b_n are configurable parameters (described in this paper as "step weight" and "error weight", respectively). Note that the PID TSC is a special case of the Third Order TSC.

4. Integration of Backbone into SALOME

The following steps were made to accomplish the integration of Backbone into SALOME:

- Develop a communication layer using communication functions of SALOME to create generic "send" and "receive" functions that are coded into Backbone. A dynamic library based on Backbone was created;
- Construct a SALOME module based on Backbone using YACS Generator (YACSGEN), a Python based package available in the SALOME distribution. YACSGEN constructs the SALOME module automatically from a short description of its three components: Backbone and two post-processing modules. The data ports and data stream ports were defined in a Python script to provide the required information. These ports will communicate between Backbone and the post-processing modules to monitor and exchange data and simulation results between them during runtime;
- Develop a post-processing module to use the Python interface of GNUPlot to receive and visualize the data sent from Backbone. Backbone will collect all the transmitted data from coupled codes (in this work, they are CATHENA and 3D-Solver prototype);
- A computational schema using the coupling capabilities of SALOMES's YACS supervisor module has been created, which is made of a set of calculation nodes corresponding to single actions. In this work, GNUPlot module (one of the calculation nodes) made with a Python script was used (took action) for a post-processing service, and Backbone (the other calculation node) took action to be used as a client, and etc. These nodes communicate with each other using input and output data stream ports corresponding to variables of different types, and are implemented using the CORBA protocol. Data stream ports were transmitting information via Calcium ports to make it possible to synchronize data during the execution of the node.

- Developed a wrapper for each code coupled to communicate between Backbone and coupling codes. The coupling code will not communicate via SALOME directly, which makes Backbone a data hub for the exchanged data involved.

5. Comparison of SALOME-Backbone Coupling to Experiment

5.1 LB-LOCA Experiment at RD14M Facility

RD-14M, established in 1984 at CNL Whiteshell site, is a thermalhydraulic test-facility representing major components of the primary heat transport system of a CANDU reactor, including emergency cooling systems. Each channel contains a 6.0 m long heated section comprised of seven indirectly electrically heated fuel element simulators (FES) with a total power of 11 MW [15]. The thermal characteristics of the FES are similar to CANDU fuel in terms of heat flux and heat capacity. The pressurized-water loop is designed with the versatility to conduct many different types of experiments for various CANDU reactor designs. The primary impetus of the experimentation is to simulate Loss-of-Coolant Accident (LOCA) scenarios (e.g., inlet header breaks, outlet header breaks, and pump suction breaks), as well as natural circulation tests. A schematic of the RD-14M facility is shown in Figure 3.

The LB-LOCA B0106 experiment conducted at the RD-14M facility was carried out as part of a series of tests to investigate the rate of coolant voiding in fuel channels during a LOCA. In addition, the experiment was conducted in a five channel per pass configuration at this facility. For these tests, orifice plates with various sized openings were installed between inlet header 8 and the break valve to simulate different break sizes. Also, an orifice plate was used to simulate a critical break where flow in the downstream channels reaches near-stagnation conditions following break opening. After warming the loop and achieving the desired steady state conditions, data collection began. Shortly after isolation of the surge tank (within 5 seconds), the break valve was opened. Within 2 seconds of the break valve opening, the power was run down to decay heat levels and a rotation-controlled pump rundown began. Data collection lasted for approximately 3 minutes after the break valve opened. Detailed information of this experiment (e.g., header pressure, differential pressure, fuel sheath temperature, etc.) can be found in Oussoren and Delja [15].

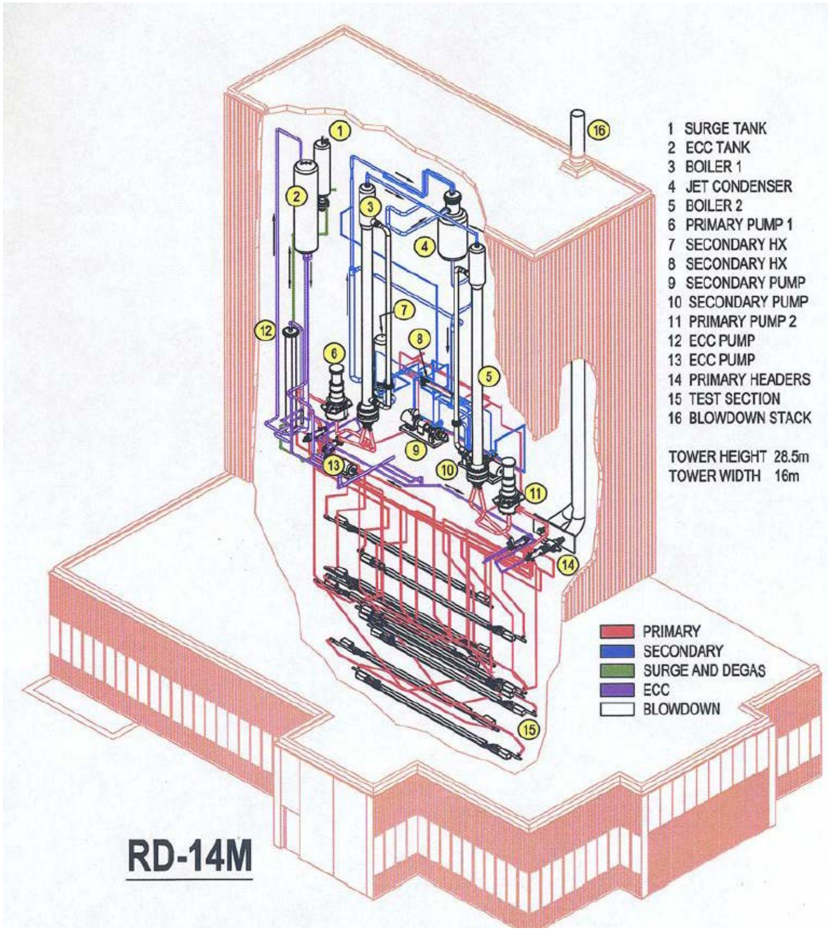

Figure 3 RD-14M Facility Schematic [15].

5.2 Multi-physics Reactor Transient Coupling

CATHENA is a non-equilibrium, two phase network analysis code in active development since the 1970s. The 3D-Solver is a prototype neutron transport code, which solves the time-independent multi-group neutron diffusion equation. CATHENA was developed in Fortran 95 and is being executed on the Windows OS, while 3D-Solver was developed in Python and is being executed on Linux. One advantage of the SALOME-Backbone joint platform is the ability to couple codes across multiple platforms. During execution, CATHENA provided 3D-Solver with an array of values corresponding to coolant density, coolant temperature and average fuel temperatures at each time step for each fuel site. This data was used as input for the 3D-Solver, which returned to CATHENA the appropriate neutronic power coefficient corresponding to the thermalhydraulics conditions received. CATHENA, upon receiving the new power coefficient, updated the internal heat generation rate in its fuel model and continued to the next time step with an updated fuel power. This transient neutronic-thermalhydraulic coupling was managed by the SALOME-Backbone joint platform and the synchronization interval is coordinated through the General Third Order TSC of Backbone, which is demonstrated in Figure 4.

Note that the 3D-Solver is a prototype code under development at CNL. It solves time independent equations for both steady-state and transient status. However, these calculations are based on the coupling data from CATHENA, such as temperature and density, which are time dependent. Since the coupling data is time dependent for transient problems, 3D-Solver's computed results effectively become time dependent as well. The main purpose of this work is to test the performance of Backbone with TSC under "blowdown". In the future, a more realistic coupling (e.g., CATHENA and NESTLE-C) would be implemented to do the reactor transient simulation.

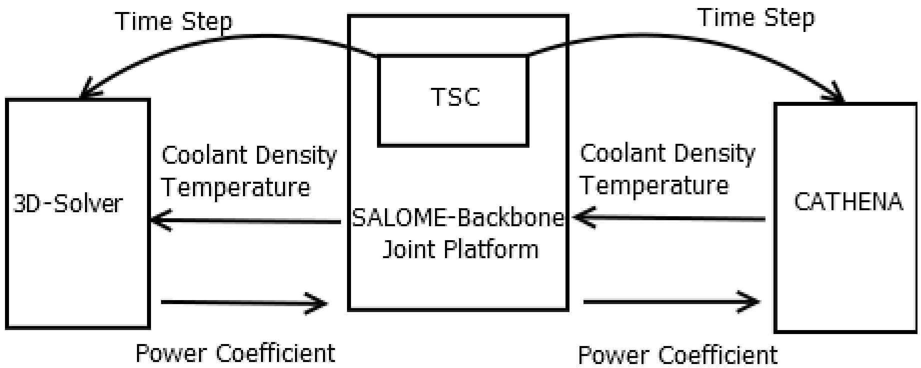

Figure 4 Multi-physics Reactor Transient Coupling through the SALOME-Backbone Joint Platform.

6. Evaluation Methodology

The performance of the multi-physics transient coupling through the SALOME-Backbone joint platform with TSC under the General Third Order TSC has been evaluated. As mentioned in Section 5.1, the LB-LOCA B0106 test conducted at the RD-14M facility was used as the test case in this work. For the B0106 case, the following conditions were considered:

- Break size: 44 mm
- Break time: 60 s
- Break valve location: Header HD8
- No emergency coolant injection
- Typical CANDU steam generator

The simulation results are compared to the experimental results to validate the correctness of the simulations. For a discussion of the CATHENA idealization for the RD-14M facility refer to Kim et al. [16]. The Heating Section HS14 was regarded as a fuel channel and a solid wall model was imposed in the idealization. The steam generator was simplified by some thermal boundaries in this work. System control models were used to control the opening of the break valve, surge tank isolation, heated section powers, primary pumps rundown, and maximum simulation time step. Pump rundown tables were developed from the experimental data. The experimentally measured heated section powers were corrected before they were applied in the simulations.

To be consistent to this experiment, the following conditions were set for the simulation using CATHENA.

- Reactor core length: 6 m

- Primary operating system pressure: 10 MPa
- System temperature: up to 350°C

In addition, the experiment performed at the RD-14M facility was based on a fixed temperature. In other words, no neutronic effects were involved in the B0106 experiment. The code coupling between thermalhydraulic and neutron diffusion code in this work is only used for the initial evaluation of the SALOME-Backbone joint platform. In order to match this experiment, the authors have to use the same fixed temperature in the CATHENA simulation, although 3D-Solver uses the coupling data from CATHENA and returns the power coefficient to CATHENA through Backbone.

The principle TSC settings at Backbone are shown below, including the setting for the simple TSC, the PID TSC, and the General Third Order TSC. These settings are valid for both CATHENA and 3D-Solver.

- Initial Time Step: 1×10^{-5} s
- Maximum Time Step: 0.1 s
- Minimum Time Step: 1×10^{-6} s
- Error Tolerance (for the PID and third order controller): 5×10^{-5}
- Step-up Ratio (for the simple controller): 1.001
- Step-down Radio (for the simple controller): 0.1

As discussed in Section 4, a simulation schema is requested to start the coupling work through the SALOME-Backbone joint platform, which is shown in Figure 5. In this schema, the “runBackbone” module will contact Backbone at the back-end of this joint platform to launch the coupling between CATHENA and 3D-Solver. The simulation data will be post-processed by the “gnuplotk” module to generate the figure.

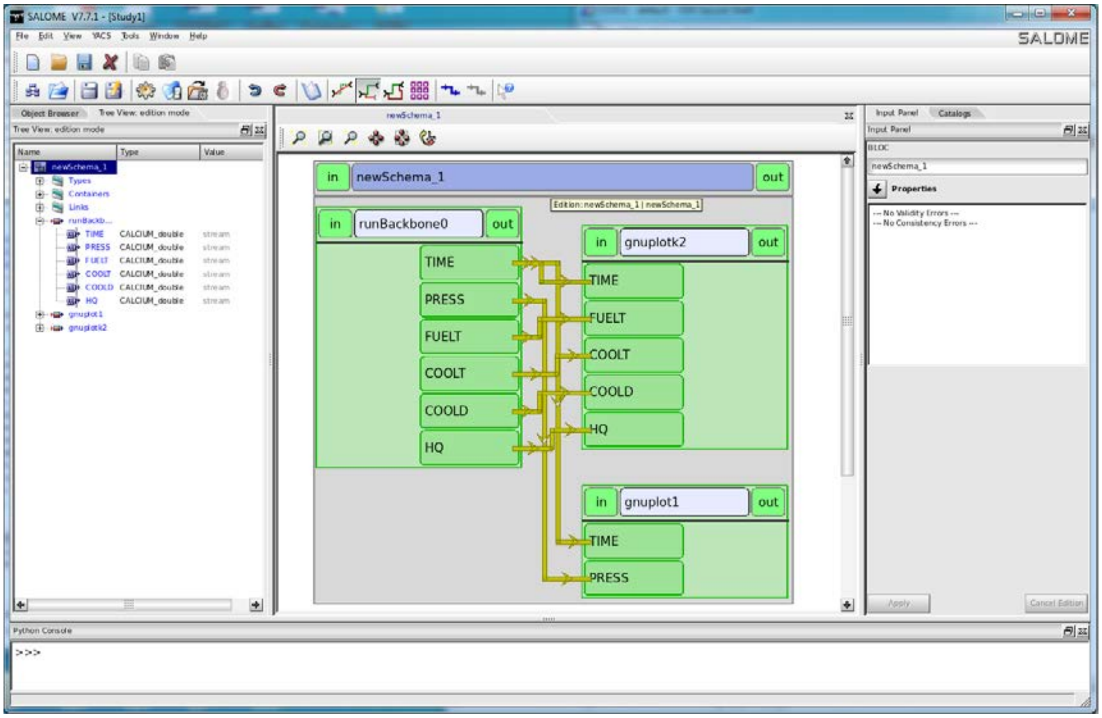

Figure 5 The Coupling Schema Generated through the SALOME YACS Module.

7. Test Results

Figure 6 shows the comparison of pressure difference between outlet Header HD5 and inlet Header HD8 between the simulation data and the experimental data under the LB-LOCA test case. Also, this simulation is synchronized by the Third Order TSC through the SALOME-Backbone joint platform. In Figure 6, it is shown that the break happened at 60 seconds and the trend of break discharge characteristics and pressure transient behaviour of the LB-LOCA Problem were captured. When the break happens, the pressure tends to drop sharply within a very short period of time, and then subsequently stabilizes. The agreement between the experimental data and calculated data is overall very good.

In addition to the previous RD-14M idealization, a fuel channel model was added to test functionality of the code coupling for the SALOME-Backbone joint platform. The average channel temperature of HS14, the coolant density and temperature data were sent from CATHENA to 3D-Solver. On the other hand, heat generation data was provided by the 3D-Solver. All those data were monitored and visualized by the post-processing modules of SALOME. Note that in this preliminary test, the heat generation rate calculated by 3D Solver was not used because 3D-Solver is a prototype. In our future plan, a real reactor transient simulation will be made using a Point Kinetic model to replace the 3D-Solver.

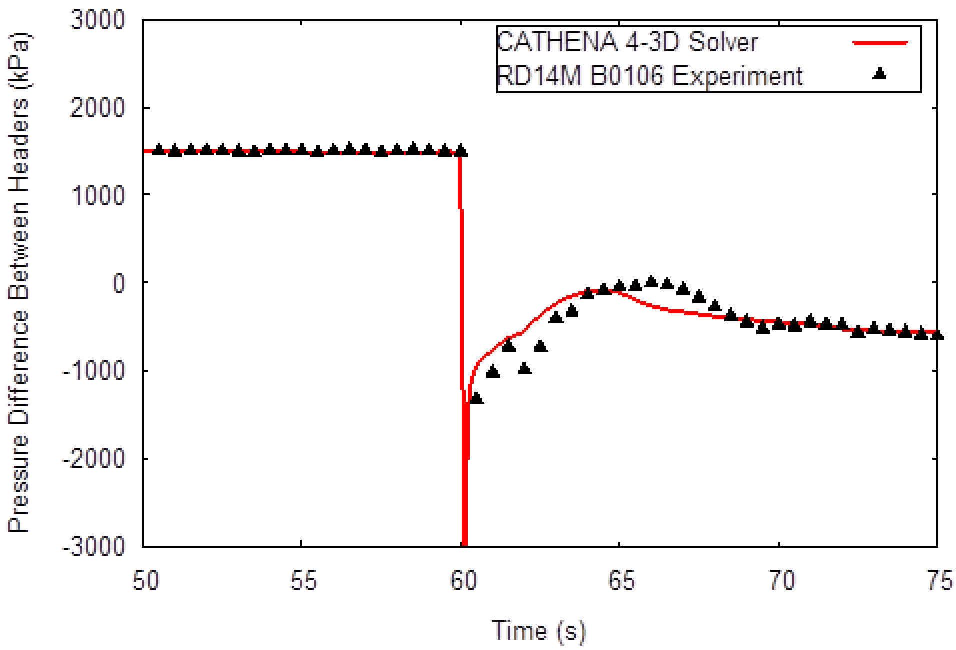

Figure 6 The Comparison of Simulated and Experimental Pressure Different between Headers (kPa) under the LB-LOCA Test Case.

8. Conclusion

Unlike other code coupling suites, the SALOME-Backbone joint platform was designed to have the capability of coupling existing industry codes with minimal or no modification to the original source code, which may be written in various programming languages and executed on different operating systems. Moreover, this joint platform combines the advantages of both SALOME and Backbone, which supports inter-platform coupling and provides a complete solution for pre- and post-processing of numerical simulations.

In this paper, a pilot study is presented that evaluated this joint platform to synchronize the thermal-hydraulics code CATHENA with the neutron diffusion code 3D-Solver. This evaluation was done by comparing the simulation results to experimental data. The simulation results clearly indicate that it captures the correct behaviour of the LB-LOCA Problem reflected by the experimental data. In the future, CNL will continue working on the SALOME-Backbone joint platform to support more realistic test cases. In addition, the effort will be dedicated to neutronic effects involved in the LB-LOCA experiment and the simulation in a longer transient time for more realistic applications.

9. Acknowledgement

The authors would like to thank Dr. Markus Piro at CNL for helpful discussions.

10. References

[1] K. Ivanov and M. Avramova, Progress and challenges in the development and qualification of multi-level multi-physics coupled methodologies for reactor analysis, in

the Proceedings of the International Congress on Advances in Nuclear Power Plants, Anaheim, CA, 2008 June.

[2] D. Gaston, C. Newman, G. Hansen and D. Lebrun-Grandie, MOOSE: A parallel computational framework for coupled systems of nonlinear equations, in the Journal of Nuclear Engineering and Design, 239(10) (2009) 1768-1778.

[3] R. Schmidt, N. Belcourt, R. Hooper and R. Pawlowski, An Introduction to LIME 1.0 and its Use in Coupling Codes for Multiphysics Simulations, Sandia Report, SAND2011-8524, 2011.

[4] A. Baudron, N. Crouzet, C. Doderlein, A. Geay, J. Lautard, E. Richebois and P. Sireta, Unstructured 3D MINOS/FLICA4 coupling in SALOME application to JHR transient analysis, in the Proceedings of the International Conference on the Physics of Reactors, Interlaken, Switzerland, 2008 September.

[5] Y. Liu, M. Nishimura, M. Seydaliev and M. Piro, Backbone: A multi-physics framework for coupling nuclear codes based on CORBA and MPI, in the Journal of Nuclear Engineering and Radiation Science, accepted.

[6] N.U. Aydemir, Recent Developments in CATHENA 4: A Fully-Implicit Network Thermalhydraulics Code, in the Proceedings of NURETH-15, Pisa, Italy, 2013 May.

[7] R.L. Williamson, J.D. Hales, S.R. Novascone, M.R. Tonks, D.R. Gaston, C.J. Permann, D. Andrs and R.C. Martineau, Multidimensional Multiphysics Simulation of Nuclear Fuel Behaviour, in the Journal of Nuclear Materials, 423 (2012), pp. 149-163.

[8] G. Soderlind, Digital Filters in Adaptive Time-Stepping, in the Journal of ACM Transactions on Mathematical Software, Vol. 29, Issue 1, 2003 March, pp. 1-26.

[9] The SALOME official website: http://www.salome-platform.org/.

[10] P.J. Turinsky, R.M.K. Al-Chalabi, P. Engrand, H.N. Sarsour, F.X. Faure and W. Guo, NESTLE: A Few-Group Neutron Diffusion Equation Solver Utilizing the Nodal Expansion Method (NEM) for Eigenvalue, Adjoint, and Fixed-Source Steady-State and Transient Problems, Technical Report of EG&E Idaho, EGG-NRE-11406, 1994 June.

[11] M. Aufiero, A. Cammi, C. Fiorina, J. Leppänen, L. Luzzi and M. Ricotti, An Extended Version of the Serpent-2 Code to Investigate Fuel Burn-up and Core Material Evolution of the Molten Salt Fast Reactor, in the Journal of Nuclear Materials, Volume 441, Issues 1–3, October 2013, pp. 473–486.

[12] O. Antoni, G. Geffraye, D. Kadri, G. Lavialle, B. Rameau and A. Ruby, CATHARE 2 V2.5_2: A Single Version for Multi-applications, in the Journal of Nuclear Engineering and Design, Volume 241, Issue 11, 2011 November, pp. 4456-4463.

[13] Object Management Group: http://www.omg.org.

[14] MPI: A Message-Passing Interface Standard Version 2.2, Message Passing Interface Forum, High Performance Computing Center, 2009.

[15] A. Oussoren and A. Delja, International Agreement Report – Assessment of Channel Coolant Voiding in RD-14M Test Facility using TRAC, Published by U.S. Nuclear Regulatory Commission, 2014 August.

[16] H.T. Kim, J.H. Park and S.B. Kim, “Simulation of RD-14M Small Break LOCA Experiments by CATHENA Code”, in the Proceedings of the Korean Nuclear Society Autumn Meeting, Gyeongju, Korea, 2009 October 29-30.