Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 12-13, 2016

An Open Source-based Approach to the Development of Research Reactor Simulator

Sungmoon Joo* YongSuk Suh, Cheol Park
Korea Atomic Energy Research Institute, Daedeok-daero 989-111, Yuseong-gu, Daejeon, Korea
**Corresponding author: smjoo@kaeri.re.kr*

1. Introduction

In reactor design, operator training, safety analysis, or research using a reactor, it is essential to simulate time-dependent reactor behaviors such as neutron population, fluid flow, and heat transfer. Furthermore, in order to use the simulator to train and educate operators, a mockup of the reactor user interface is required. There are commercial software tools available for reactor simulator development. However, it is costly to use those commercial software tools. Especially for research reactors, it is difficult to justify the high cost as regulations on research reactor simulators are not as strict as those for commercial Nuclear Power Plants(NPPs).

Currently, no economical development software framework exists for research reactor simulators. To fill this gap, we present an approach for developing research reactor simulators using open source software tools.

2. Research Reactor Simulator

2.1 Research Reactor

The primary function of NPP is to generate electricity. Unlike NPPs, the primary purpose of research reactors (RRs) is to provide neutrons for research or produce radioisotope. RRs are smaller and simpler relative to NPPs; most research reactors range up to 100 MW, operate at lower temperatures. As a means to maximize neutron production, RRs usually have a moderator or reflector to slow down the neutrons and enhance fission, and to reduce neutron loss from the core, respectively.

While NPPs are designed, constructed, and operated under strict regulations, there are no clear regulatory guidelines for research reactors except for guidelines for safety, and it is often recommended to follow regulations for NPPs, if possible [10,11].

2.2 Reactor Simulator

Nuclear Power Plant (NPP) simulators are playing an important role in the entire NPP lifecycle including licensing, traditional operator training and education; a simulator can be applied in research, design or operation of NPP. Along with NQA1, which is the nuclear quality assurance requirement endorsed by ASME, ANSI/ANS-3.5 (2009) is the only formal standard that can be referenced for guidance of simulator development [4]. In this standard, verification testing, validation testing, and performance testing are defined, and acceptance criteria are provided. ANSI/ANS-3.5 aims at meeting the requirements of code 10 CFR 55 (Code of Federal Regulations, 1998) of the Nuclear Regulatory Commission. To assure the conformance to the regulations, generally, an NPP simulator is composed of high-performance computers running high-fidelity codes, and an exact replica of the man-machine interface system (MMIS) developed by using high-cost commercial software tools.

Compared with the case of NPP, there are no strict regulations that govern the development and usage of RR simulators. Since it is not necessary to have such a high fidelity RR simulator for licensing and operator training, use of costly commercial software tools are not desirable. In addition, simulator development for RRs is more agile because the design of RR is not as standardized (i.e. routinely) and complex as that of NPP, and design ~ construction period is relatively short.

3. Open Source Software Tools for Reactor Simulator

To develop a reactor simulator for research reactors, we propose adopting several open source software tools such as Experimental Physics and Industrial Control System (EPICS), Control System Studio (CSS), Python, and PyRK library.

3.1 Open Source Software

Open source software (OSS) is computer software with its source code readily available. In brief, OSS can be freely used, modified, and shared. Strictly speaking, however, open source licenses are licenses that comply with the Open Source Definition (OSD), and must go through the Open Source Initiative's (OSI's) license review process [6].

The following is a partial list of popular OSI-approved open source licenses:

- Apache License 2.0
- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- GNU General Public License (GPL)
- GNU Library or "Lesser" General Public License (LGPL)
- MIT license
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License

3.2 Python and PyRK

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems.

Python is developed under an OSI-approved open source license, making it freely usable and distributable, even for commercial use. Most Python releases have been GPL-compatible. This, however, does not mean that Python is distributed under GPS; all Python licenses, unlike the GPL, allow the distribution of a modified version without making the modifications open source. The GPL-compatibility makes it possible to combine Python with other software under the GPL [7].

The PyRK provides a modular simulation environment for calculation in nuclear engineering [5]. It is a freely distributable Python library under open source BSD 3-clause license [8]. PyRK can be used to simulate reactor behavior with precursor data, a modular material definition framework, and coupled lumped parameter thermal hydraulics with zero-dimensional neutron kinetics (also known as point kinematics equations, PKEs)

The PKEs are the set of equations that capture neutronics and thermal hydraulics when the time-dependent, geometric distribution of the neutron flux is negligible. The RKEs are numerically quite stiff as two mechanisms which have very different time characteristic scales are coupled by reactivity. PyRK provides efficient and accurate solutions for PKEs.

3.3 EPICS

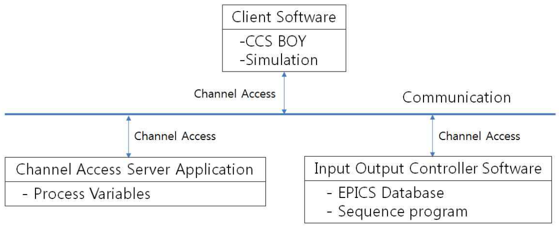
EPICS is a comprehensive software development platform for creating real-time, distributed control systems. It is a toolkit comprises hundreds of pieces of software. As an open source toolkit, EPICS has been developed by worldwide collaborations and used for large scientific experimental systems such as particle accelerators, telescopes and neutron source facilities [1, 3]. EPICS is available under the EPICS Open license. It is more or less like the modified BSD license, and intended to comply with the OSD. However, it includes certain wording required by the US Department of Energy, and has not been submitted for OSI's review [9].

A typical EPICS control system can have the following building blocks (Fig. 1(a)).

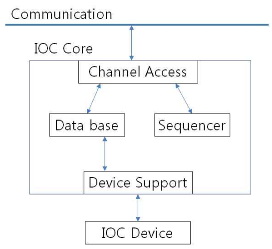
- Channel Access (CA) protocol
- Input Output Controller (IOC) Software
- CA server application
- Client software

An IOC is a computer which runs software called “IOC Core.” The computer can be almost any machine, i.e. PC running Windows, Linux, RTEMS, Mac running OSX, or Embedded processor running RTEMS, Linux. An EPICS control system must have at least one CA server (usually an IOC). CA allows other programs (CA clients) to see and change values of process variables in an IOC (CA servers). CA servers give clients access to named process variables. EPICS also provides users with tools for operating those variables (e.g. logical operations). Fig.1(b) depicts the structure of a typical IOC Core. An IOC has one or more database loaded, which tells the IOC what to do.

Features needed to develop user interface (e.g. synoptic, archive, trend, alarm) are provided by Control System Studio (CCS). CCS is a user interface framework based on Eclipse Rich Client Platform (RCP). For the development of operator interface (e.g. operator panels), an operator interface editor and runtime called BOY (Best OPI Yet) is used [1].



(a) EPICS Control System Configuration



(b) Major Components of an Input Output Controller Core

Fig. 1. EPICS Control System

4. Preliminary Results and Conclusion

An open source-based simulator for a research reactor is configured as a distributed control system based on EPICS framework. Fig. 2 illustrates the schematics of a simulator which consists of multiple computers running GUI applications, and a reactor simulation module. PyRK provides simulation interface to solve the time-dependent behavior of the reactor simulation module.

To demonstrate the use of the simulation framework proposed in this work, we consider a toy example [5]. This example approximates a 1-second impulse-reactivity insertion in a reactor, which represents the instantaneous removal and reinsertion of a control rod (Fig. 3). The change in reactivity results in a slightly delayed change in power and corresponding increases in temperatures throughout the system (Fig. 4 and 5).

We proposed an approach for developing research reactor simulator using open source software tools, and showed preliminary results. The results demostrate that the approach presented in this work can provide econimical and viable way of developing research reactor simulators.

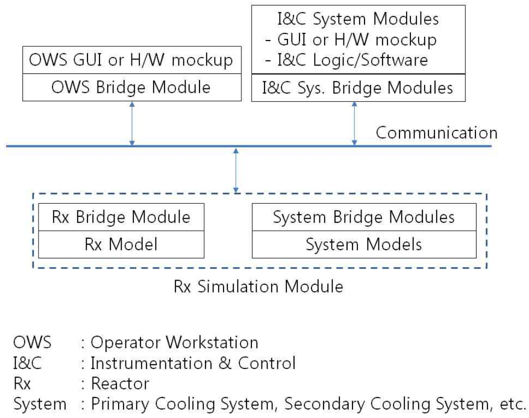


Fig. 2. Simulator Architecture using EPICS

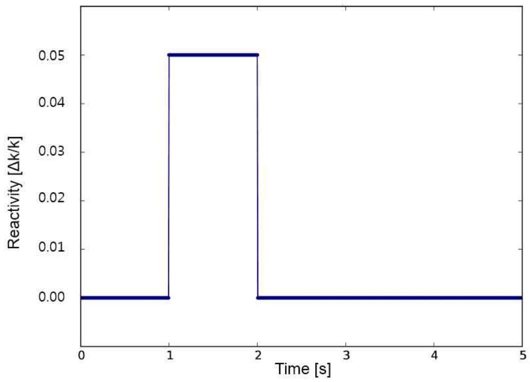


Fig. 3. Reactivity Insertion

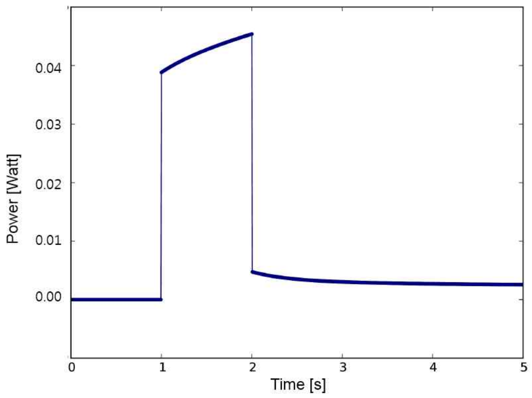


Fig. 4. Reactor Power Response

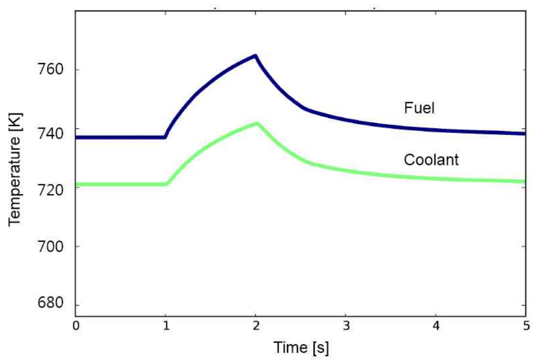


Fig. 5. Temperature Response

Acknowledgements

This work was supported by the National Research Foundation of Korea grant funded by the Ministry of Science, ICT and Future Planning (MSIP) (Grant Code: 2012M2C1A1026912).

REFERENCES

[1]Experimental Physics and Industrial Control System [Web log] Retrieved March 9, 2016, from http://www.aps.anl.gov/epics/

[2]Janos S. Janosy, Simulation and Simulators for Nuclear Power Generation, Nuclear Power – System Simulations and Operation, Pavel Tsvetkov(Ed.), ISBN:978-953-307-506-8, 2011.

[3] A. Grelle, Y. Cao, Y. Gohar, Y. Park, T. Wei, Neutron Source Facility Simulator(NSFS), ANL Technical Report, ANL-14/07, Aug. 2014.

[4] Ye Cheng, Ni Chao, Zheng Tian, Zhang Zhicheng, Zhang Ronghua, Quality Assurance for a Nuclear Power Plant Simulator by Applying Standards for Safety-Critical Software, Progress in Nuclear Energy, Vol.70, p. 128-133, 2014.

[5] K. Huff, PyRK: A Python Package For Nuclear Reactor Kinetics, Proceedings of the 14th Python In Science Conference (SCIPY 2015), Jul.13-15, 2015, Austin, TX.

[6] https://opensource.org (accessed on March 11, 2016)

[7] https://docs.python.org/2/license.html (accessed on March 11, 2016)

[8] https://pyrk.github.io (accessed on March 11, 2016)

[9] https://www.aps.anl.gov/epics/license (accessed on March 11, 2016)

[10] IAEA Safety Standard Series No. SSG-20, "Safety Assessment for Research Reactors and Preparation of the Safety Analysis Report."

[11] NUREG-1537, "Guidelines for Preparing and Reviewing Applications for Licensing of Non-Power Reactors."