Cyclus: Once-Through Fuel Cycle Simulation Comparisons with VISION

Matthew J. Gidden, Paul P. H. Wilson, Kathryn D. Huff, Robert W. Carlsen

Department of Nuclear Engineering & Engineering Physics, University of Wisconsin, Madison, WI 53706
gidden@wisc.edu

INTRODUCTION

The *Cyclus* project at the University of Wisconsin – Madison is the result of previous lessons learned and is designed to be a modular application, allowing for a variety of collaborators and a potentially diverse user-base. Built in C++, *Cyclus* uses XML-based pre-processing and manages a SQL-based output database. The *Cyclus* team has recently grown and now incorporates visualization expertise through collaboration with the University of Utah, server-client expertise via Idaho National Lab, and social communication expertise through collaborators at UW-Madison to assist the mission-critical goal of relevancy vis-a-vis policy makers. Accordingly, the *Cyclus* project is expanding efforts from structural capability to benchmarking calculations.

CYCLUS DEVELOPMENT

The paradigm under which *Cyclus* has been developed and is distributed is unique and offers a number of advantages regarding its potential wide-spread adoption by a diverse user base. The *Cyclus* team has worked hard to provide an overview of its code development to the community.[2]

Open Development Paradigm

As one of the leading principles guiding the development of *Cyclus*, an open-source repository provides a high degree of flexibility and a large amount of exposure to potential collaborators and developers. The *Cyclus* repository is publicly available via GitHub.[3] A number of tools are available to *Cyclus* developers in order to maintain software development best-practices, including distributed version control, automatic documentation, and in-depth issue management. Additionally, *Cyclus* developers have taken advantage of existing, external open-source libraries in order to conform to current standards, including an expansion of the C++ standard library and fully benchmarked linear and integer programming solvers. Perhaps the most important attribute of the open development paradigm is that it allows for unfettered access to the base *Cyclus* source code. Combined with modular software development, *Cyclus* is a easily transferable framework on which to build a strong fuel cycle simulation community.

Dynamic Module Capability

Incorporation of dynamically loadable, independently construced modules is another key design concept in *Cyclus*. The core simulation engine comprising *Cyclus* is physically separate from the various modules determining the specific behavior that occurs in each simulation. Interaction between the core and modules occurs via dynamic linking, allowing for encapsulated development and providing developers the ability to focus on specific modules without involving the simulation engine. This increases efficiency and decreases the overall programming experience required. In addition, this encapsulation allows for a spectrum of collaboration levels, because developers can choose the level of privacy with which they wish to handle their simulation data and module processes; this decision is independent for each *Cyclus* user entity. An example of this type of behavior is shown in Figure 1.

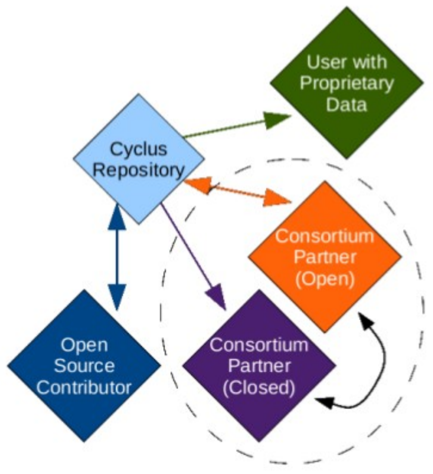
User with Proprietary Data
Cyclus Repository
Consortium Partner (Open)
Open Source Contributor
Consortium Partner (Closed)

Fig. 1. The *Cyclus* open-repository paradigm.

COMPARING CYCLUS WITH VISION

Simulation Infrastructure

Cyclus employs a hierarchical Region – Institution – Facility infrastructure. That is, a simulation in *Cyclus* is comprised of Regions, which own one or more Institutions, which in turn own one or more Facilities.

Interactions, e.g. the trading of material or the negotiation of contracts, can occur between all three levels of the hierarchy. Decision making within *Cyclus* can be thought of as occurring at two levels, the microscopic level, e.g. month-to-month exchanges of material between facilities, and the macroscopic level, e.g. strategic decision making of facility construction. The *Cyclus* simulation engine is nominally discrete, i.e. global events occur according to a simulation clock; however, actions within modules need not occur in a discrete manner.

VISION uses a combination of Microsoft Excel files to accept input and provide output and uses Powersim Studio, commercially available software, as its simulation engine. Simulations in VISION model the fuel cycle from a systems-level perspective instead of modeling each discrete facility, i.e. it models "types" of facilities and processes rather than each individually. The VISION simulation engine is also discrete in time.[1]

Once-Through Scenarios Compared

Cyclus and VISION were both run on a number of scenarios varying from relatively simple to relatively complex. In general, the number of reactors, number of storage facilities, power demand, and timing of storage strategy are all variables in a simple once-through simulation. The *Cyclus* team began benchmarking with one reactor, one repository, exponential power demand, and immediate storage of material. Each variable was increased in order to systematically increase the complexity of the simulations. Eventually, a simulation of all 103 U. S. nuclear reactors with various demand predictions were modeled.

RESULTS

Cyclus has performed well in benchmarks of once-through fuel cycles with VISION, an industry standard. It has the capability to model a variety of growth scenarios with varying fuel storage strategies. Further work is currently underway to completely match VISION's capability in this regard, after which work will begin to include scenarios under which recycling of fuel occurs and strategies utilizing fast reactors.

REFERENCES

1. J. J. JACOBSON, et al, VISION User Guide - VISION (Verifiable Fuel Cycle Simulation) Model, INL/EXT-09-16645, (2009).
2. K. HUFF, P. P. H. WILSON, and M. GIDDEN, "Open Architecture and Modular Paradigm of Cyclus, a Fuel Cycle Simulation Code," ANS Summer Conference (2011).
3. K. HUFF, P. P. H. WILSON, M. GIDDEN, and R. CARLSEN, "Cyclus: A Nuclear Fuel Cycle Code from the University of Wisconsin Madison," http://cyclus.github.com/ (2011).

ACKNOWLEGEMENTS

This research is being performed using funding received from the DOE Office of Nuclear Energy's Nuclear Energy University Programs. The author thanks the NEUP for its generous support.