Open Architecture and Modular Paradigm of CYCLUS , a Fuel Cycle Simulation Code

Kathryn D. Huff, Paul P.H. Wilson, Matthew J. Gidden
Department of Nuclear Engineering & Engineering Physics, University of Wisconsin, Madison, WI, 53706
khuff@cae.wisc.edu

INTRODUCTION

The CYCLUS project at the University of Wisconsin - Madison is the result of lessons learned from experience with previous nuclear fuel cycle simulation platforms. The modeling paradigm follows the transacation of discrete quanta of material among discrete facilities, arranged in a geographic and institutional framework, and trading in flexible markets. Key concepts in the design of CYCLUS include open access to the simulation engine, modularity with regard to functionality, and relevance to both scientific and policy analyses. **The combination of modular encapsulation within the software architecture and an open development paradigm allows for a balance between collaboration at multiple levels of simulation detail and security of proprietary or sensitive data.**

When comparing different nuclear fuel cycle concepts, it can be a challenge to find any two systems analyses that compare across a common set of metrics with a similar set of underlying assumptions. Each analysis is likely to focus on a set of metrics that are of interest to the team performing the analysis and involve both implicit and explicit assumptions and constraints about the behavior of the fuel cycle system. While a strict prescription of these metrics, assumptions and constraints could be proposed for comparison purposes, another solution is to provide a systems analysis simulation tool that provides sufficient modularity, extensibility and open access, that it can be a basis for harmonzing to a common set. If it becomes easier to modify an existing simulation tool to support the needs of a new analysis than it is to develop a new tool, then it is possible that such a tool will be adopted more universally for such analysis.

MODELING PARADIGM

The modeling paradigm adopted by CYCLUS includes a number of fundamental concepts which comprise the bedrock on which other, more flexible, design choices have been made.

Discrete Materials and Facilities

The CYCLUS modeling infrastructure is designed such that every facility in a global nuclear fuel cycle is treated and acts individually. While modeling options will exist to allow collective action, this will be as a special case of the individual facility basis. Each facility has two fundamental tasks: the transaction of goods or products with other facilities and the transformation of those goods or products from an input form to an output form. For example, a reactor will receive fresh fuel assemblies from a fuel fabrication facility, transform them to used used fuel assemblies using some approximation of the reactor physics, and supply those used fuel assemblies to a storage facility.

Market-based Material Transactions

The transaction of nuclear materials takes place in markets that act as brokers matching a set of requests for material with a set of offers for that material. A variety of market models will be available to perform this brokerage role. It is important to note that each market is defined for a single commodity and acts independently of other markets. Once the requests and offers have been matched by each market in a simulation, the facilities exchange material objects.

Region-Institution-Facility Hierarchy

Many times in a fuel-cycle simulation, parameters describing relationships between facilities are required. For instance, one may wish to account for the presence of a contract between two facilities, or one may wish to model two facilities operated by the same entity. Accordingly, CYCLUS implements this ability via a Region-Instituion-Facility hierarchy. Every discrete facility in CYCLUS is considered to be owned by an institution that operates in a geographic region. An institution can be used to represent any entity that may own and operate a facility such as a private corporation, a government agency, or a non-governmental organization, among others. A region can be used to represent any geographic area, typically a politically relevant area such a sub-national region (e.g. a U.S. State), a nation-state, or a supranational region (e.g. the E.U.). While some performance parameters of the facility may depend on its institutional ownership or geographical location, this capability is more useful in modeling the way in which a facility engages in a market for trade of nuclear material based on its ownership entity and/or region.

SOFTWARE ARCHITECTURE AND DEVELOPMENT

Dynamic Module Loading

The ability to dynamically load independently constructed modules is a heavy focus of CYCLUS development. Dynamically-loadable modules are the primary mechanism for extending CYCLUS ' capability. The primary benefit of this approach is encapsulation: the trunk of the code is completely independent of the individual models. Additionally, any customization or extension is implemented only in the loadable module. A secondary benefit of this encapsulation is the ability for contributors to choose different distribution and licensing strategies for their contributions. By allowing models to have varied availability, the security concerns of developers can be assuaged. Finally, this strategy allows individual developers to explore different levels of complexity within their modules, including wrapping other simulation tools as loadable modules for CYCLUS . This last benefit of dynamically-loadable modules addresses another goal of CYCLUS : ubiquity amongst its potential user base. By engineering CYCLUS to easily handle varying levels of complexity, a single simulation engine can be used by both users keen on big-picture policy questions as well as users interested in more detailed, technical analyses.

Open Development Paradigm

CYCLUS obeys an open development paradigm that utilizes current best practices in distributed code development and facilitates inter-institutional collaborations. The CYCLUS development framework employs a modern, open source philosophy that ensures transparency, attracts contribution from a varied community of collaborators, and guarantees institution-independent access to all potential users.

A public source-code repository provides unhindered access to the fundamental simulation framework and basic fuel-process models volunteered by developers. Granting unfettered access only to the CYCLUS engine allows for separation of the source code from any input data, thereby allowing secure and proprietary model developers to be similarly encouraged to utilize the CYCLUS framework. This modern development model passively distributes specialized content to interested research groups, and facilitates parallel model development efforts by institutions with complimentary goals. The transparency inherent in this type of open source development path also facilitates code review by exposing available content to verification and validation by collaborators with diverse areas of specialization and levels of expertise.

This process also benefits from the ability to identify and utilize third-party, open-source libraries when such libraries provide necessary additional functionality. Thus far, such libraries have been used to provide basic infrastructure functionality such as file input/output, as well as model-specific capabilities for advanced network flow optimization. An indepently developed integer programming module is an example of one such addition.

CONCLUSIONS

The above sections outline a proposed fuel cycle simulation platform currently under development at the University of Wisconsin at Madison. CYCLUS presents an effective framework for transparent, accessible, yet compartmentalized code development. Its software architecture and modeling paradigm promises to maximize ease of use, ease of contribution, technical utility, and relevance to both technical and policy-based inquiries, which may represent an optimal way forward for the next generation of nuclear fuel cycle modeling.

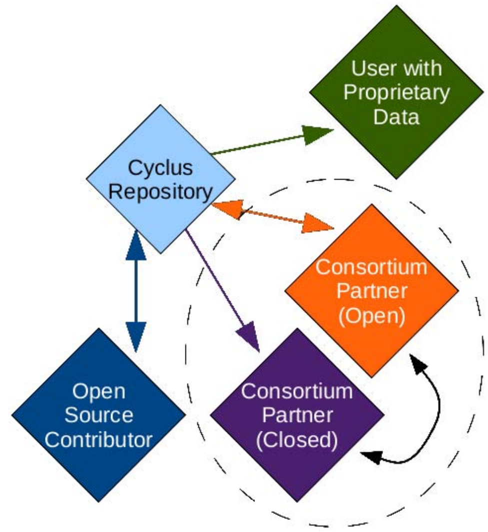

Fig. 1: CYCLUS Open Repository Paradigm

REFERENCES

1. K. HUFF, P. P. H. WILSON, and M. GIDDEN, "Cyclus: A Nuclear Fuel Cycle Code from the University of Wisconsin Madison," http://code.google.com/p/cyclus/ (2010).