CYCLUS : An Open, Modular, Next Generation Fuel Cycle Simulator Platform

Kathryn Huff | P. Wilson | M. Gidden | R. Elmore, Nuclear Engineering and Engineering Physics U. Wisconsin - Madison

CYCLUS *is a top-level, next generation, nuclear fuel cycle simulation framework designed with an open development process and modular platform. It employs lessons learned from previous efforts to pursue quantitative assessment of worldwide nuclear energy production, material flows, energy costs, environmental impact, proliferation resistance, and robustness against supply disruption.*

Lessons Learned

Cyclus is the result of a refactoring of the GENIUSv2 fuel cycle simulator. In addition to replicating GENIUSv2 functionality, *Cyclus* seeks to emphasize a design basis in computational science and software development that can deliver:

- **Usability** directed toward a wide range of user sophistication
- **Performance** solving simple problems in interactive time scales
- **Fidelity** accommodating a range of levels of detail commensurate with user sophistication

A number of other software development practices have been identified as highly valuable for this effort.

- **Openness** ensures transparency and lowers institutional & technical obstacles to collaboration.
- **Modularity** allows the core infrastructure to be independent of proprietary and/or sensitive data and/or models.
- **Extensibility** with a focus on both robustness and flexibility allows for myriad potential developer extensions.

Dynamically Loadable Modules

Dynamically-loadable modules are the primary mechanism for extending *Cyclus* with new models. The primary benefit of this approach is **encapsulation**: the trunk of the code is made completely independent of the individual models and all customization and extension is implemented only in the loadable module.

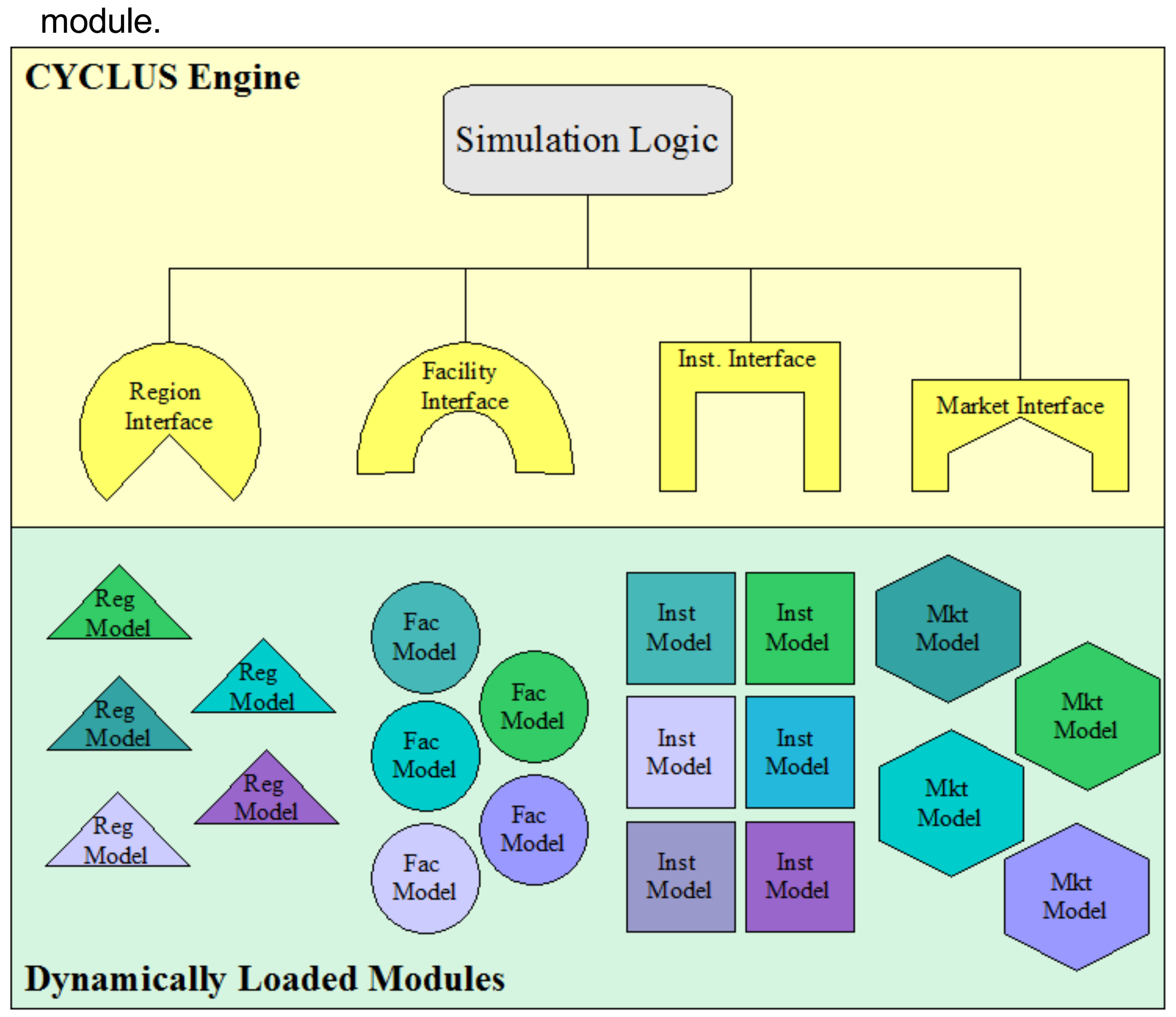

Figure 1. *Cyclus* software design emphasizes encapsulation in which universal model interfaces allow the simulation logic to be completely independent of the interchangeable model libraries.

Open Source Development Process

Transparency

- **Open source code** on a publicly available repository encourages peer review of algorithm implementation of fundamental simulation framework and basic fuel process models volunteered by developers.
- **Facilitates collaboration** among a large group of geographically dispersed developers.

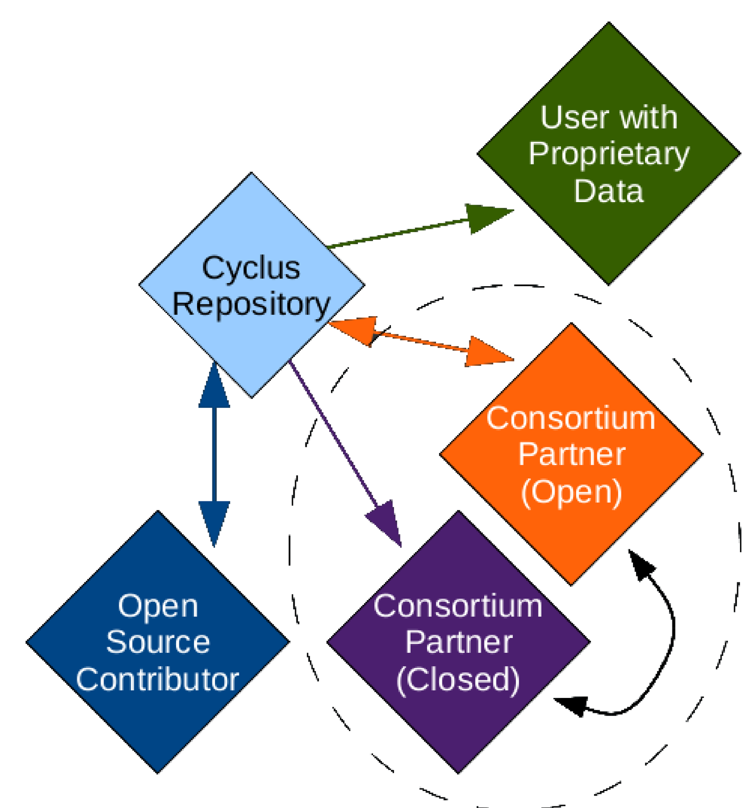

Figure 2. Only the skeletal *Cyclus* engine, basic fuel cycle models, and volunteered developer content are made publicly available. Proprietary and secure data and algorithms can be kept private or shared selectively.

Modularity

- Secure and proprietary model developers can be similarly encouraged to utilize the *CYCLUS* framework
- Another aspect of this development process is a preference for open source third party libraries. This includes basic infrastructure such as file input/output, as well as model-specific capabilities like integer programming for network flow optimization.
- A secondary benefit of this encapsulation is the ability for contributors to choose different distribution and licensing strategies for their contributions.
- Modular framework allows individual developers to explore different levels of complexity within their modules, or wrap other simulation tools as loadable modules for *Cyclus* .

Quality Control

- This open source repository employs **a version control** system for provenance and reproducibility of results.
- A built-in **issue tracking system** allows ticketing of development milestones and public bug reporting.
- 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.

Materials Tracking at the Isotope Level

- Quantitative assessments are driven by information about the state of the world's nuclear material at all stages of the nuclear fuel cycle.
- Cyclus therefore tracks the history of each isotope involved in the entire simulation.
- Each facility type deals with the fuel at a different stage of its life cycle. These stages are treated as different commodities in the Cyclus simulation.

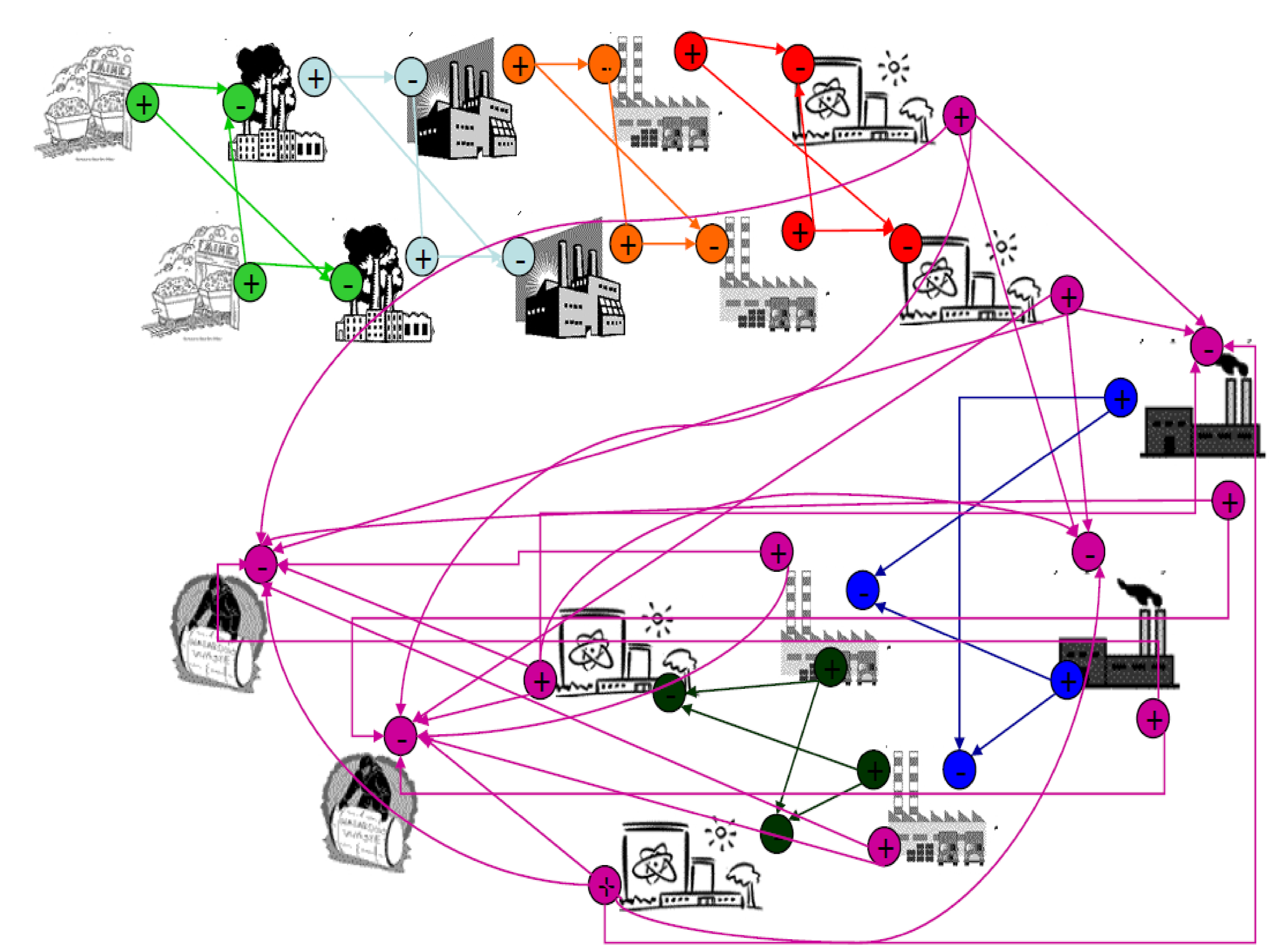

Figure 3. Each color represents a separately traded commodity. While each facility receives a different input feed, many facilities produce spent fuel destined for the repository.

Facility Hierarchy in the Nuclear Fuel Cycle

- Facilities in the nuclear industry owned by distinct governments and institutions buy, sell, and trade nuclear materials.

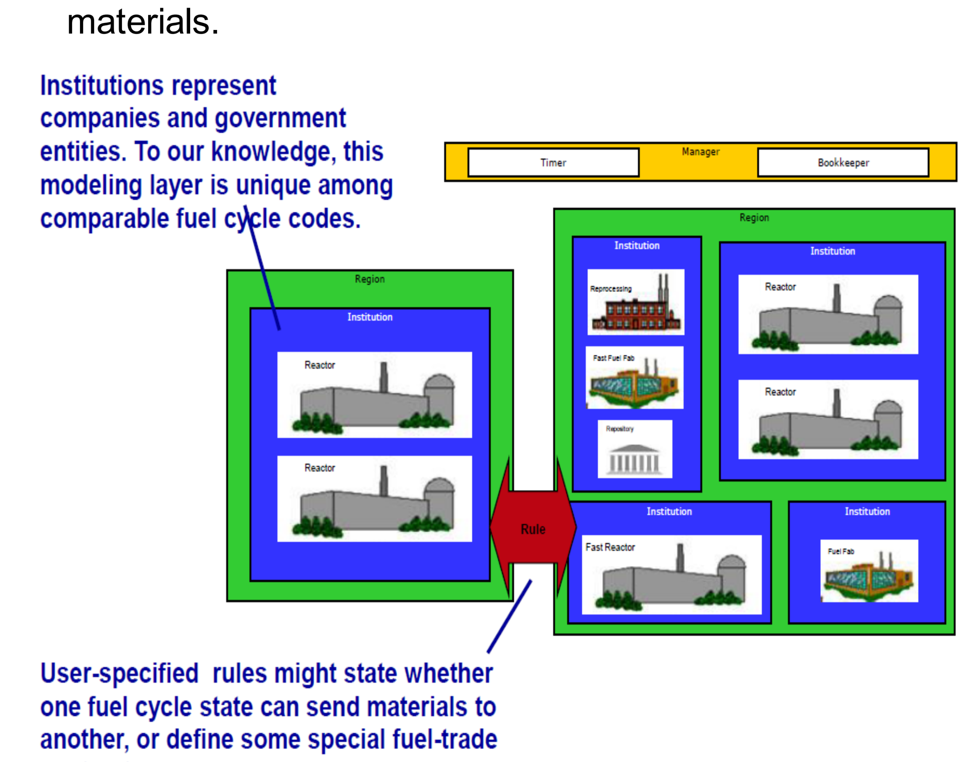

Figure 4. The facility hierarchy obeys dynamic rules regulating trade between regions and facilities.

Discrete Region, Institution, Facility Data Model

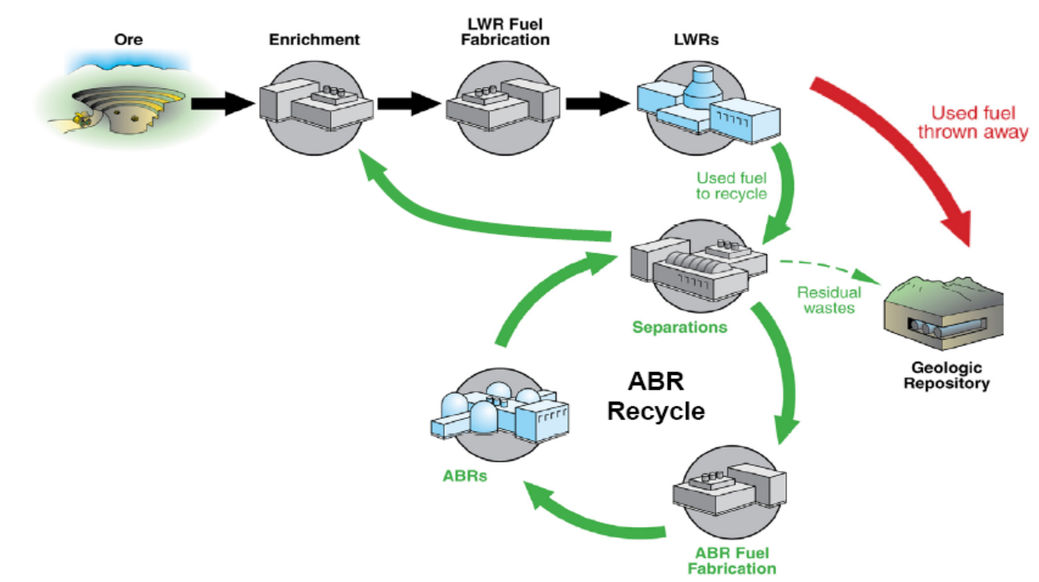

Figure 5. Distinct facilities in the nuclear fuel cycle perform distinct tasks, but are significantly interdependent, each relying on the production capacity of others upstream.

Market-Based Materials Trading

- Each facility in the simulation makes requests in order to fulfill its needs for feed commodities every month.
  - For example, a reactor with no fresh fuel on site might request some in order to be ready for refuel requirements of its next cycle.
- Simultaneously, each facility makes an offer of its current stocks of product commodity.
  - For example, a fuel fabrication facility might offer to sell the amount of fuel it has in stock from fabrication in the previous months.
- Finally, Market Models apply a matching algorithm of their choice to find a best fit for all of the offers and requests for that month. The match is then sent in the form of instructions to the various facilities which then buy, sell, and trade accordingly.

Current and Future Applications

- Benchmarking is a key step in software development, and Cyclus is currently being benchmarked against similar codes.
- Economic Analysis to evaluate the cash flows in the fuel cycle will be pursued in the near term.
- Analysis of fuel cycle robustness against supply disruption is in the beginning stages.
- Development of generic geological repository models and interchangeable waste package models are a top priority in Cyclus development.

References

1. Jain, R. and Wilson, P. P. H. (2006). "Transitioning to global optimization in fuel cycle system study tools." *Transactions of the American Nuclear Society*, **95**, pages 162–3.
2. Juchau, C. (2008). "Development of the global evaluation of nuclear infrastructure and utilization scenarios (GENIUS) nuclear fuel cycle systems analysis code." Master's thesis, Idaho State University.
3. Juchau, C. & Dunzik-Gougar, M.L. (2006). "A Review of Nuclear Fuel Cycle Systems Codes." Accessed 10 October 2009 from SINEMA Web site: http://thesinema.org/.
4. Dunzik-Gougar, M.L. et al. (2007). "Global Evaluation of Nuclear Infrastructure Utilization Scenarios.
5. Lisowski, P. (2007). Global Nuclear Energy Partnership. In *Global Nuclear Energy Partnership Annual Meeting*, Litchfield Park, AZ. Global Nuclear Energy Partnership.
6. Wilson, P., Huff, K. and Gidden, M. (2010). "Cyclus Introduction" whitepaper.
7. K. Huff and P. Wilson "Cyclus: A Nuclear Fuel Cycle Code from the University of Wisconsin Madison," http://code.google.com/p/cyclus/ (2010).