Introduction to Cyclus: History, Strategy, Development and Future

Paul Wilson
for the Cyclus Development Team

Cyclus Development Team

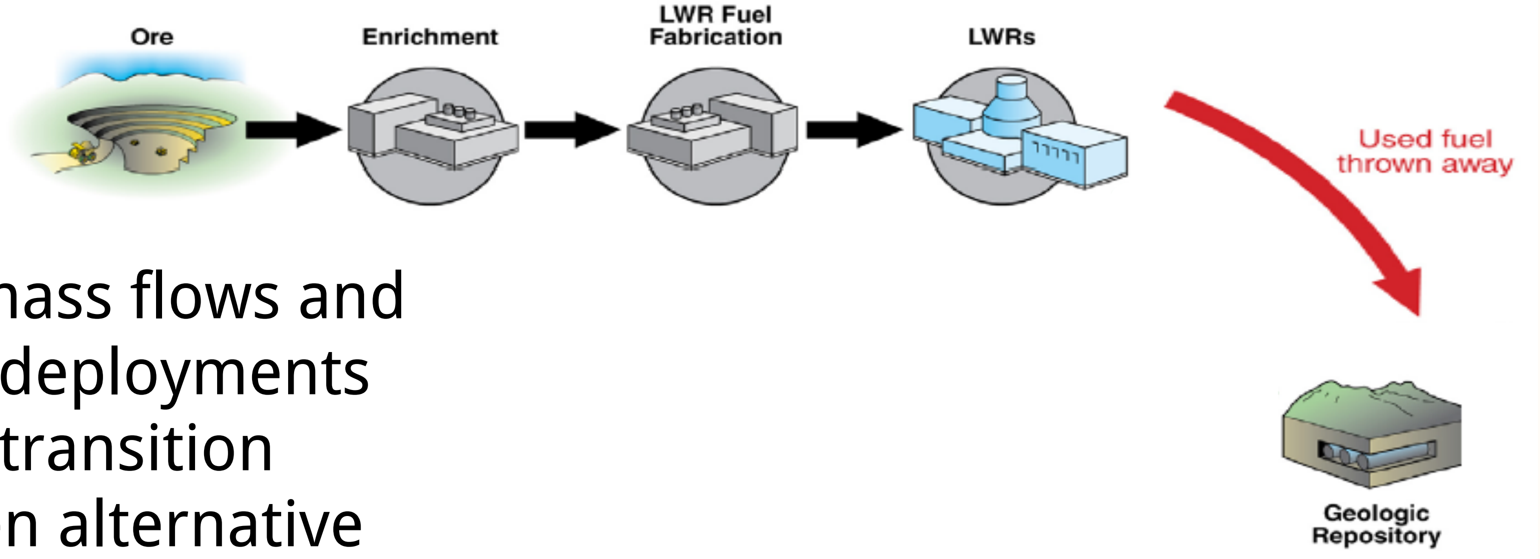
- University of Wisconsin
  - Nuclear Engineering: Robert Carlsen, Matthew Gidden, Michael Gionet, Kathryn Huff(*), Meghan McGarry, Arrielle Opotowsky, Olzhas Rakhimov, Anthony Scopatz, Zach Welch, **Paul Wilson**
  - Life Science Communication: Ashley Anderson, **Dominique Brossard**, Nan Li, Dietram Scheufele
- University of Texas
  - Nuclear Engineering: Robert Flanagan, **Erich Schneider**
- University of Utah
  - Computer Science: Haya Agur, **Yarden Livnat**
- University of Idaho
  - Computer Science: **Robert Hiromoto**, Teva Velupillai

(*) Currently University of California-Berkeley

Overview

- Fuel Cycle Simulators Background
- Next Generation Fuel Cycle Simulator
- Cyclus History
- Cyclus Strategy
- Moving Forward

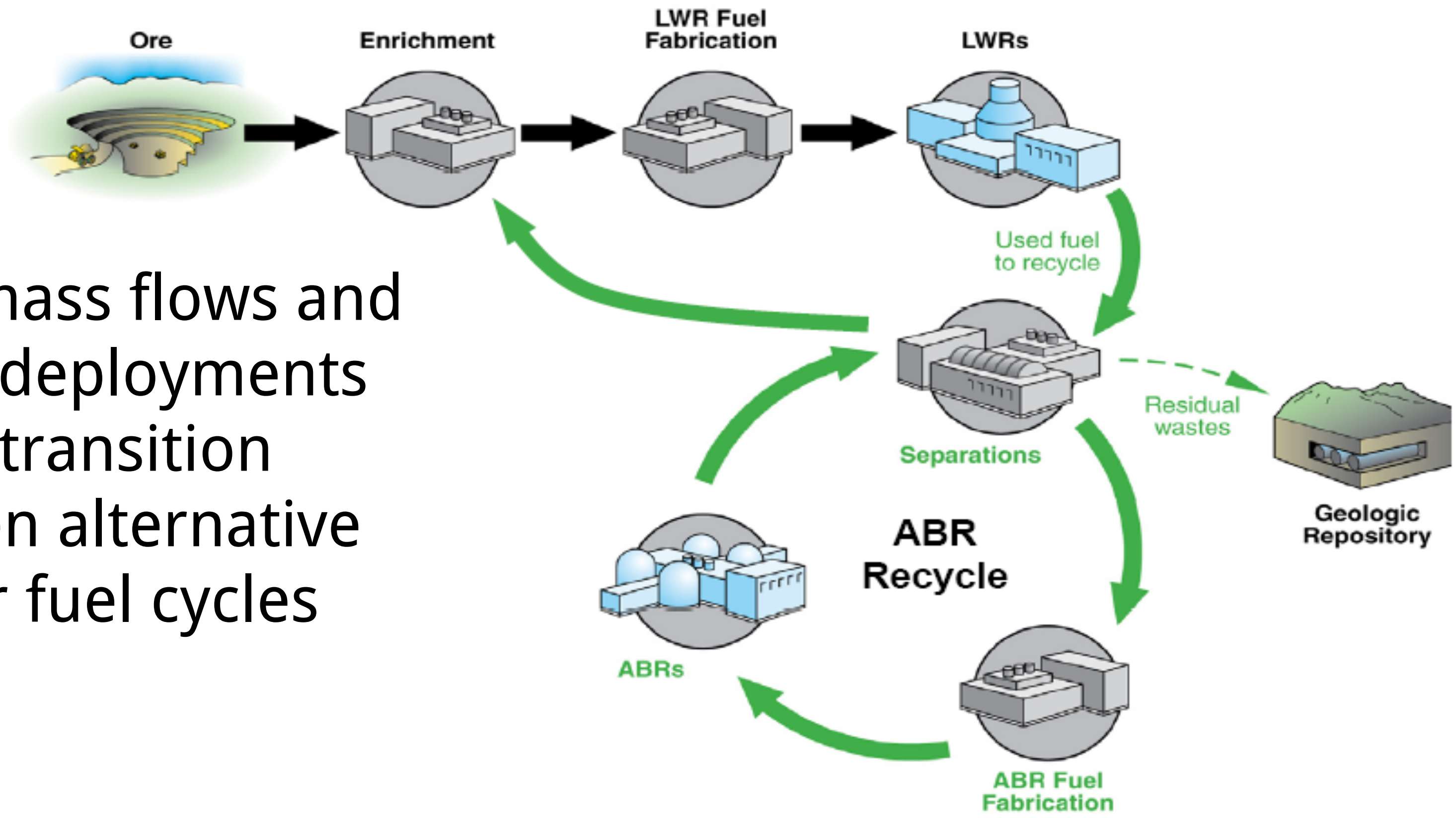
Fuel Cycle Simulator - Purpose



- Track mass flows and facility deployments during transition between alternative nuclear fuel cycles

Fuel Cycle Simulator - Purpose

- Track mass flows and facility deployments during transition between alternative nuclear fuel cycles



Fuel Cycle Simulator - Purpose

- Track mass flows and facility deployments during transition between alternative nuclear fuel cycles

- Translate fundamental output to environmental and socio-economic impacts

- Inform technical and non-technical decision makers when making strategic nuclear fuel cycle choices

FCS Origin Stories

Two primary motivations

- Immediate commercial interest in managing nuclear fuel
  - Built around a high-fidelity in-reactor simulation
  - Limited flexibility for novel systems/technologies
- Strategic decision making
  - Begin with simple low-fidelity flow sheet approach
  - Complexity increases with desire for detail/nuance
  - Encounter limitations of software infrastructure
  - Accessibility to non-technical audiences diminishes

FCS Proliferation

1. Researcher has new nuclear technology idea
    - Desires understanding of system impact
2. Examines available FCS options
    - Inflexible underlying modeling assumptions
    - Specialized software infrastructure
    - High barrier to adoption
3. Start with new simple FCS based on single technology
    - Low barrier to entry
4. New FCS follows 1 of 2 paths
    - Remains simple/limited for single technology
    - Becomes more complex (see #2, above)

Partial Listing of FCS

- USA
  - Vision (INL)
  - DANESS (ANL)
  - DYMOND (ANL)
  - CAFCA (MIT)
  - NFCSim (LANL)
  - NUWASTE (NWTRB)
  - Genius v1 (ISU)
  - Genius V2 (UW)
  - Cyclus (UW)
- France
  - COSI
  - CLASS
- Russia
  - DESAE
- International
  - NFCSS/Vista (IAEA)
  - SMAFS (OECD/NEA)

FCS Code Comparisons

- A number of code comparisons have been conducted
- Each FCS implements
  - A different set of features
  - A different set of modeling choices/assumptions for common features
- Comparison is reduced to common denominator

Next Generation FCS Goals

- Modeling
  - Discrete facilities with discrete material tracking
  - Optimization and sensitivity analysis
- Flexibility
  - Permit modeling of innovative/unconventional technologies
  - Minimal inherent technology assumptions
- Software
  - Low barrier to adoption with rapid payback
  - Commonly available software infrastructure

Next Generation FCS UI Goals

- Different UI layers for different audiences
  - Common physics infrastructure
  - Varying levels of input control
  - Varying levels of output exploration

- Interactive operation for many audiences

System Dynamics Weaknesses

- Entire network must be setup *a priori*
  - Clever design can create limited flexibility
- Not suited to discrete facility modeling
  - Fleets of average facilities
- Not suited to discrete material modeling
  - Continuous flows of homogenous materials
- Apparent benefit of drag-and-drop interface diminishes with model complexity
- Fundamentally a system of first-order ODE's
- Difficult/expensive to automate

Discrete Facilities/Materials

- More faithful model of reality

- Necessary for tracking individual facility performance
  - Startup/shutdown
  - Disruptions

- Allow individual material objects to be tracked
  - Effects of individual facility performance
  - Forensic tracking of material object ownership

NGFCS Funding History

- Contemplated by DOE as IRP project in 2010
  - Typically $1M-$3M/year for 3 years
  - Designed for team of many institutions

- Implemented as NEUP R&D project
  - Maximum $400k/year for 3 years
  - Difficult to engage multiple institutions

NEUP FY11 Workscope FC-7

Systems Analysis

"The *key university research need for this activity is the development of modules for the Fuel Cycle Simulator, including fuel cycle modules, interface modules and data modules*. These modules will focus on specific aspects of the nuclear fuel cycle simulator and should be created in such a way that they can plug into an **overall framework, which will be developed in coordination with the Systems Analysis Campaign.**"

"The first step in development of the Fuel Cycle Simulator will be design and development of the over-arching framework, or information backbone, for the FCS and **will be lead [sic] by the Fuel Cycle Technologies Systems Analysis Campaign**."

"Proposals should focus on the development of specific modules as described above, but proposals related to the areas listed below will also be considered:

- basic modules for each function of the fuel cycle
- front end GUI development to support a wide range of users
- flexible back end GUI development to support range of module output information
- assistance in building libraries of historic facility/infrastructure information (national/global)
- innovative concepts for interaction with and communication of simulator results to decision makers and other non-expert users, including determination of the key factors on public decision making as related to the deployment of complex technologies."

NGFCS User Experience Proposal

- Assume DOE base program to develop NGFCS
- Five thrust areas:
  1. Stakeholder, parameter & metric identification
  2. User interface and model generation
  3. Metric translation
  4. Visualization environment
  5. Efficient design of a client-server model

NGFCS User Experience Scope Change

- **No** DOE base program to develop NGFCS
- Introduced additional thrust
  0. Cyclus kernel/infrastructure development
  1. Stakeholder, parameter & metric identification
  2. User interface and model generation
  3. Metric translation
  4. Visualization environment
  5. Efficient design of a client-server model

Cyclus History

- UW group began with DANESS/VISION
  - Integrating repository benefit into fuel cycle analysis
- Participated in SINEMA (Joint ANL/INL LDRD project)
  - Ambitious plan to build connected modeling & simulation framework from first principle models through systems analysis
  - GENIUS developed as systems analysis tool
    - Version 1 at Idaho State University
    - Version 2 at U. Wisconsin

GENIUS v1.0 Design Goals

Requirements Organized According to Four Technical Functions

Function or Requirement	Description
Function 1	Characterize and deploy individual fuel cycle facilities and reactors
R1.1	Simulations must be able to reflect all significant design data for elements of a fuel cycle.
R1.2	Track quantities of natural resources as a function of time, location, and accessibility.
R1.3	Track process materials as a function of time, location, and accessibility (element and isotope).
R1.4	Track the operations status of each production facility as a function of time, location, and capacity.
R1.5	Track the operations status of each storage facility as a function of time, location, and capacity.
R1.6	Track the operations status of each disposal facility as a function of time, location, and capacity.
R1.7	Track status of nuclear materials transportation as a function of time, location, and type.
R1.8	Track products and by-products as a function of time, location, and type (electricity, heat, hydrogen, etc.).
R1.9	Track costs/expenditures as a function of time, location, and type.
Function 2	Perform component and aggregate uncertainty analyses
R2.1	Capable of performing uncertainty analysis for each element of the fuel cycle.
R2.2	Generate sensitivity coefficients for each element of the fuel cycle.
R2.3	Capable of performing aggregate uncertainty analysis by propagating the uncertainty in each element of the fuel cycle.
Function 3	Optimize simultaneously across multiple objective functions
R3.1	Capable of running in a semiautomated mode using inputs to produce desired model outputs. (e.g., nonproliferation, economic, and waste management targets).
R3.2	Be able to dynamically perturb local optimum solutions to test robustness and adaptability.
R3.3	Support a graphical user interface.
Function 4	Open and accessible code software and documentation
R4.1	Maintain abstractions between data and process algorithms.
R4.2	Open and accessible code architecture, source code, and documentation.
R4.3	Be able to communicate with other codes through weak links/databases.

C. Juchau, M.L. Dunzik-Gouagar, “Modeling the Nuclear Fuel Cycle.” Nuclear Technology, 171, pp 136-141 (2010)

GENIUS V2.x Learned from V1.x

- Performance challenges attributed to Python language and simple text-based record keeping
  - Switch to C++ object-oriented model
- Eliminate rigid heuristics for facility deployment decisions
  - Provide user-driven deployment option
- Eliminate rigid definition of allowable material exchange pathways
  - Introduce markets as network flow optimization for material exchange

Cyclus Learned from GENIUS v2.x

- Remove dependence of simulation kernel on available facilities
  - Introduce concept of plug-in facilities
- Replace rigid network flow problem with concept of plug-in markets
  - Since replaced by *dynamic resource exchange (DRE)*
- Migration towards agent-based paradigm
  - Extension of object-oriented model
  - Introduction of DRE gives control of decision making to agents

Cyclus Development History

- *March 2010:* First line of code by Katy Huff
- *February 2011:* Cycamore module repository establishes ecosystem model
- *September 2011:* Third active developer joins team
- *October 2011:* First external funding from NEUP
- *October 2012:* First external funding for non-UW developers: Schneider @ U. Texas
- *November 2012:* First Cyclus results published
  - Once through fuel cycle benchmarks, by Matt Gidden, *et al (ANS Transactions)*
- *June 2013:* Project lead hired: Anthony Scopatz

Cyclus Development History

- *August 2013:* First Cyclus-based thesis: Katy Huff
  - Demonstrates complex facility module for geologic repository
- *October 2013:* First external funding for new developers: Skutnik @ U. Tennessee
- *October 2013:* Additional NEUP funding for Cyclus optimization
- *May 2014:* Cyclus kernel release v1.0
  - 1st stable interface for module/visualization developers
- *September 2014:* Cyclus kernel release v1.1
- *October 2014:* First non-NE funding for Cyclus: NNSA

Cyclus Funding History

NEUP Nuclear Energy University Programs

U.S. Department of Energy

Heavily leveraging student support from:

NEUP Nuclear Energy University Programs U.S. Department of Energy

Argonne NATIONAL LABORATORY

NSF

U.S.NRC

U.S. DEPARTMENT OF HOMELAND SECURITY

Cyclus Development Strategy

1. Open source simulation kernel
2. Ecosystem of plug-in modules
3. Open source analysis and visualization tools

Cyclus Open Source Kernel

- Modeled after many successful open source software packages
- Collaborative development facilitated via Github
  - Leading free web service
  - Thorough code review before contributions added
  - Continuous testing on various software platforms
- Mailing lists for wide communication network
  - Developer list to discuss enhancements
  - User list to share knowledge among new/experienced users

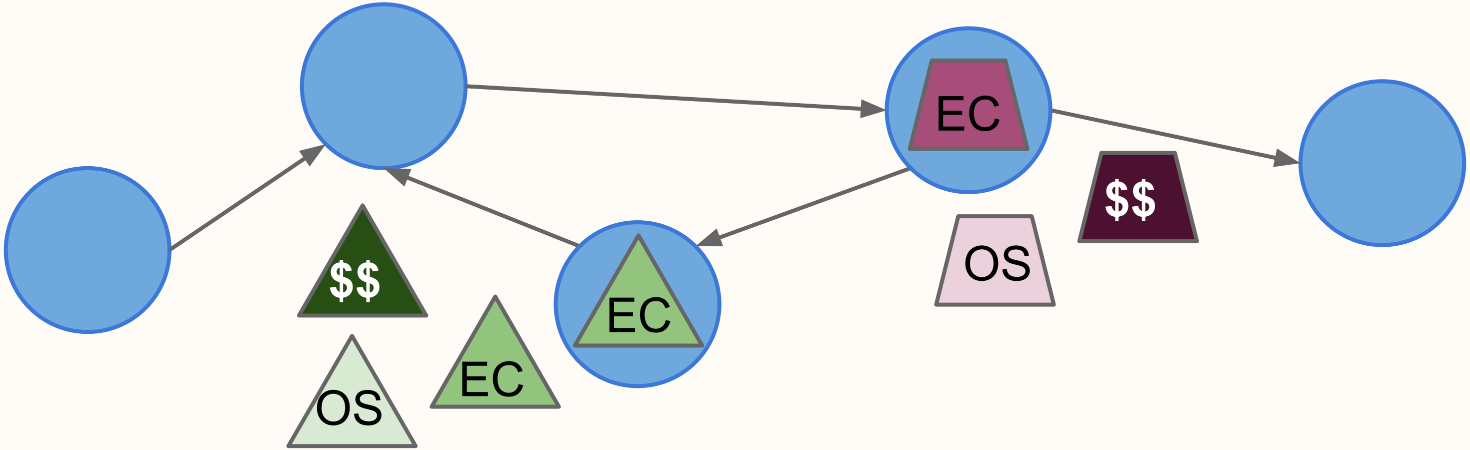
Fundamental Concepts

- Agent-based approach
  - Facility agents include physics modeling choices and social interaction models
  - Region and Institution agents influence their Facility agents choices
- Dynamic Resource Exchange
- Discrete Material Tracking

Cyclus Module Ecosystem

- Archetype modules developed by independent teams
- Quality assessed by community
  - Tests and documentation provided by developers
  - Potential module users perform independent testing
- Diversity driven by use cases of developers

Cyclus Module Ecosystem



- Facility archetypes can be exchanged without changes to the kernel
- Example: increase reactor modeling fidelity
    - Low fidelity: fixed input/output recipes
    - Medium fidelity: lookup tables for output given input
    - High fidelity: burnup calculation based on given input
- Various distribution models are possible

Cyclus Analysis & Visualization

- Separate from simulation kernel
- Different tools for different purposes
  - Interactive data exploration
  - Automated generation of standardized images
  - Parameter sweeps
  - Wrappers for
    - Sensitivity study
    - Optimization
- Each tool uses state-of-the-art technology
- Open source development options

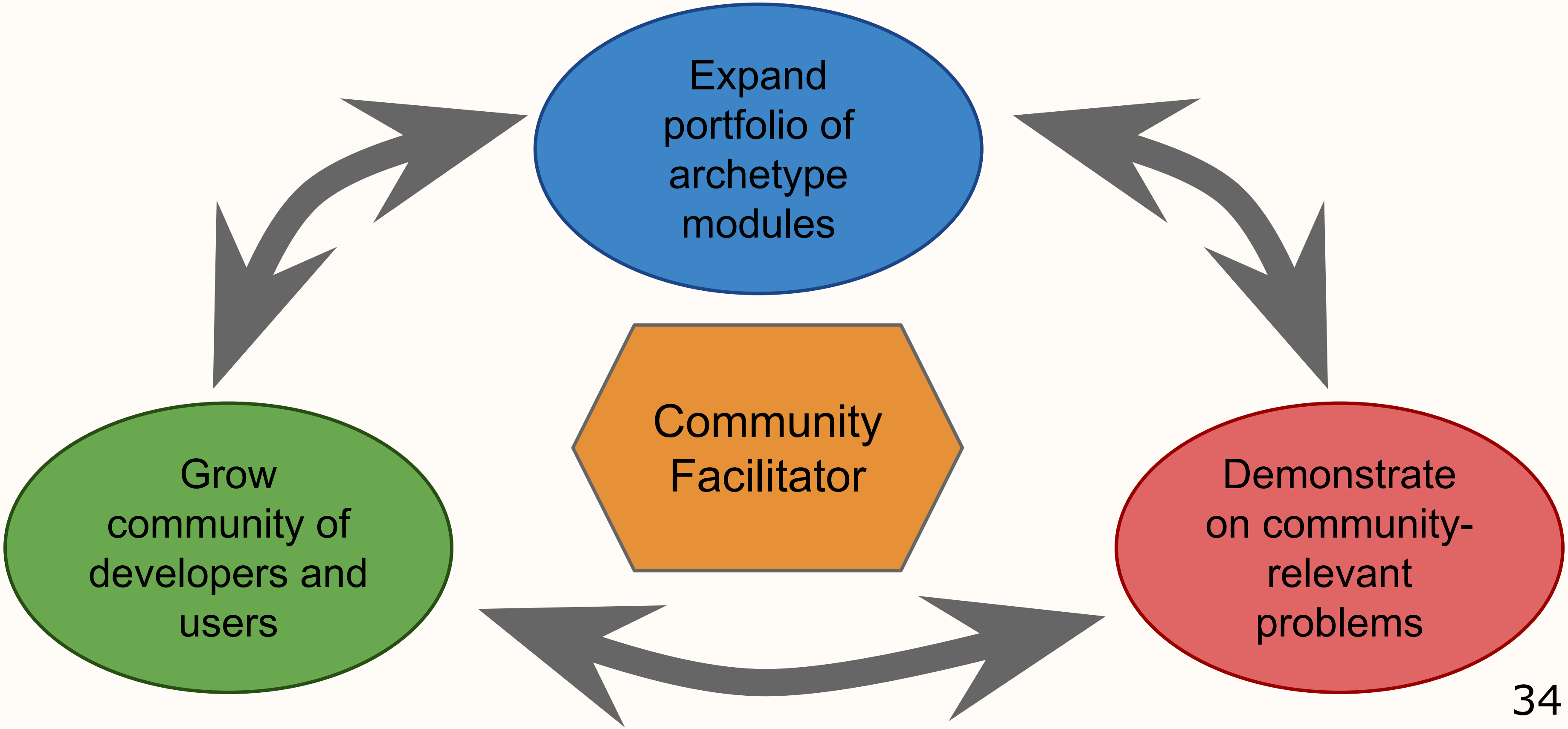
List of Cyclus Tools/Projects

Cyclus	Simulation kernel
Cycamore	Basic module library
Cycic	Input control - embedded in Cyclist
Cyclist2	Interactive data exploration environment
Ciclus	Continuous Integration scripts for Cyclus
Cycstub	Skeleton for clean slate module development
Cyan	Cyclus analysis tool
Cloudlus	Tools for running Cyclus in a cloud environment

Cyclus Development Challenges

- Lack of DOE-based NGFCS development program
  - Required expansion of scope on fixed budget
  - Delayed all other thrusts
- Survey obstacles
  - DOE-NE delayed survey on protocol concerns
  - Delayed communications research
- Level of engagement by active fuel cycle analysis efforts
  - Primary development team excluded from transition analysis exercise
    - Assisted by Katy Huff via LLNL
  - Delayed development of key archetypes

Moving Forward with Cyclus



Development Priorities

- Distribute reference set of archetypes
  - Primarily from existing archetypes
  - Enable analysis of standard fuel cycle options
  - Reference deployment strategy unclear

- Provide broad set of fuel cycle metrics
  - Enable richer analysis of interesting outcomes
  - Integrate with data exploration environment
  - Perhaps defined by FCO metrics

Potential Users

- DOE and DOE-funded group
  - NEUP funded universities
- Industry users, e.g. AREVA, EPRI
- Foreign DOE-equivalents, e.g. AECL
- Foreign universities, e.g. Cambridge University

Questions?

http://www.fuelcycle.org