Introduction to Cyclus

Paul Wilson
and the Cyclus Development Team

Cyclus Development Team

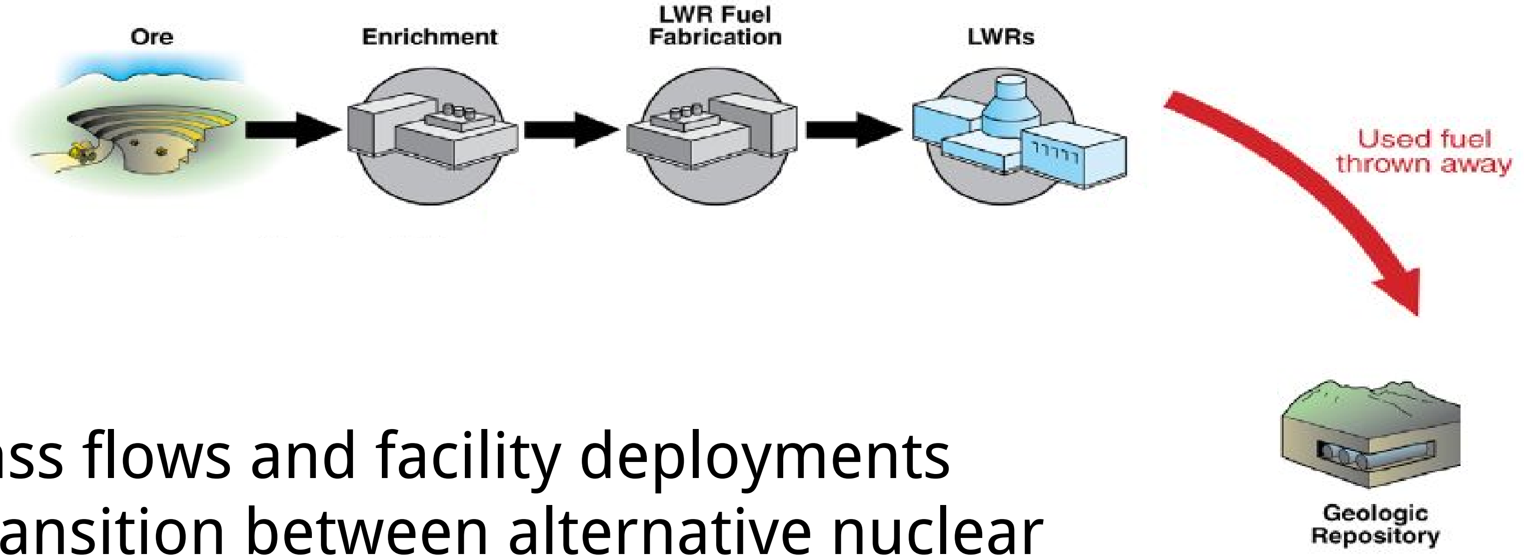
- University of Wisconsin
  - Nuclear Engineering: Robert Carlsen, Vincent Cloitre, Matthew Gidden, Michael Gionet, Kathryn Huff[1], Meghan McGarry, Baptiste Mouginot, Arrielle Opotowsky, Olzhas Rakhimov, Zach Welch, **Paul Wilson**
  - Life Science Communication: Ashley Anderson, **Dominique Brossard**, Nan Li, Dietram Scheufele
- University of Texas
  - Nuclear Engineering: Cem Bagdatlioglu, **Erich Schneider**
- University of South Carolina
  - Nuclear Engineering: Anthony Scopatz, Robert Flanagan
- University of Utah
  - Computer Science: Haya Agur, **Yarden Livnat**
- University of Idaho
  - Computer Science: **Robert Hiromoto**, Teva Velupillai

[1] Currently University of Illinois at Urbana-Champaign

Overview

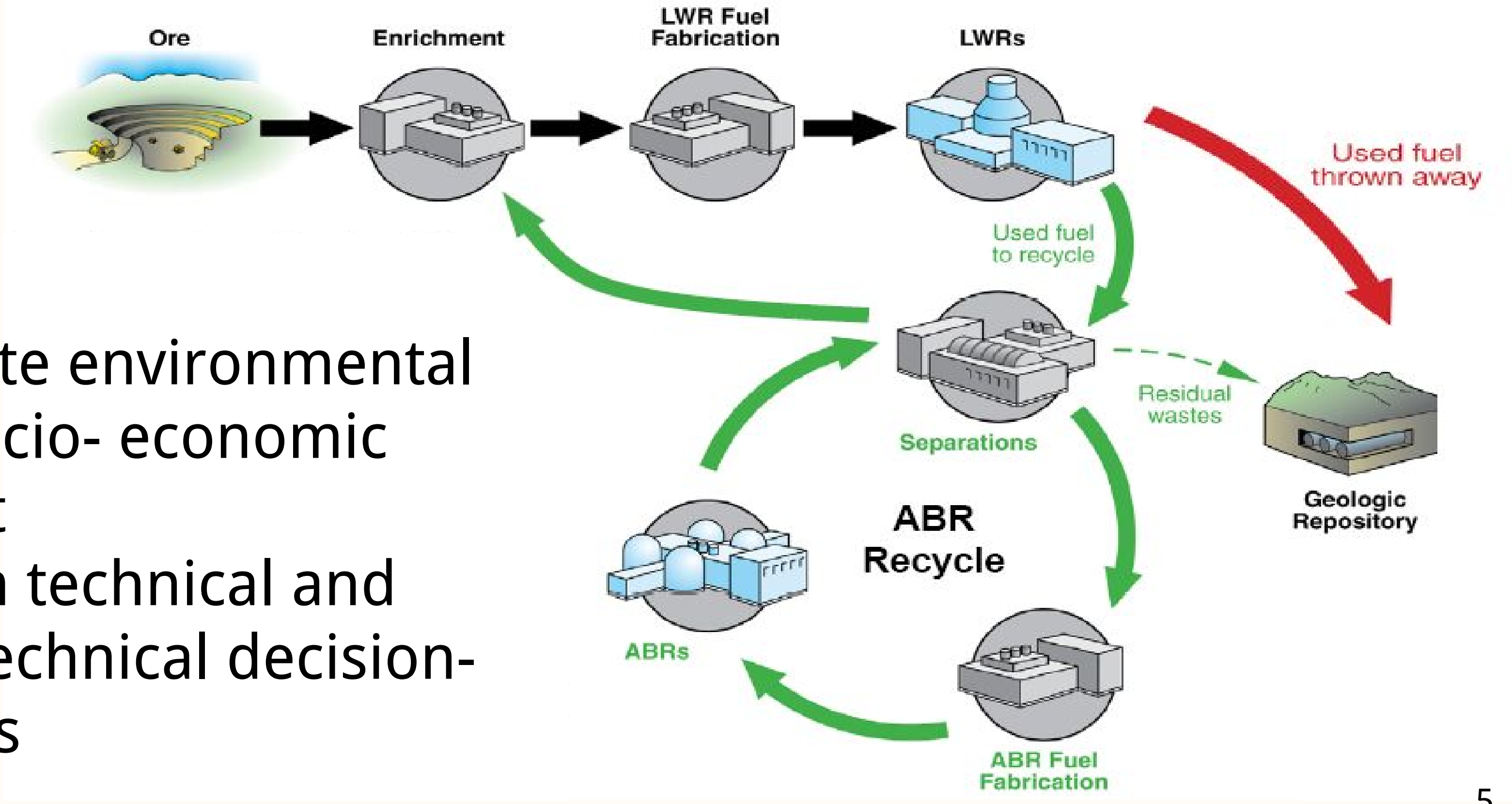
- Fuel Cycle Simulators Background
- Next Generation Fuel Cycle Simulator
- Cyclus History
- Cyclus Strategy
- Moving Forward

Fuel Cycle Simulator - Purpose



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

Fuel Cycle Simulator - Purpose



- Evaluate environmental and socio- economic impact
- Inform technical and non- technical decision- makers

Motivations for New Simulators

Flexibility

- Accommodate innovative systems and cycles
- Carefully study impacts of modeling choices
- Implement as part of optimization and sensitivity analysis
- Minimize inherent technology assumptions
- Allow for maximum fidelity
  - Discrete facilities with discrete material quanta

Accessibility

- Open source development using commonly available tools

Cyclus Development Strategy

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

Features of Cyclus Kernel

Agent-Based Approach

- Encapsulate physics and interaction behavior in each ***Facility***
- Each facility operated by an ***Institution*** in a geopolitical ***Region***

Dynamic Resource Exchange

- Constant deployment gives changing material flow paths
- Material substitution complicates matching of supply/demand

Discrete Material Tracking

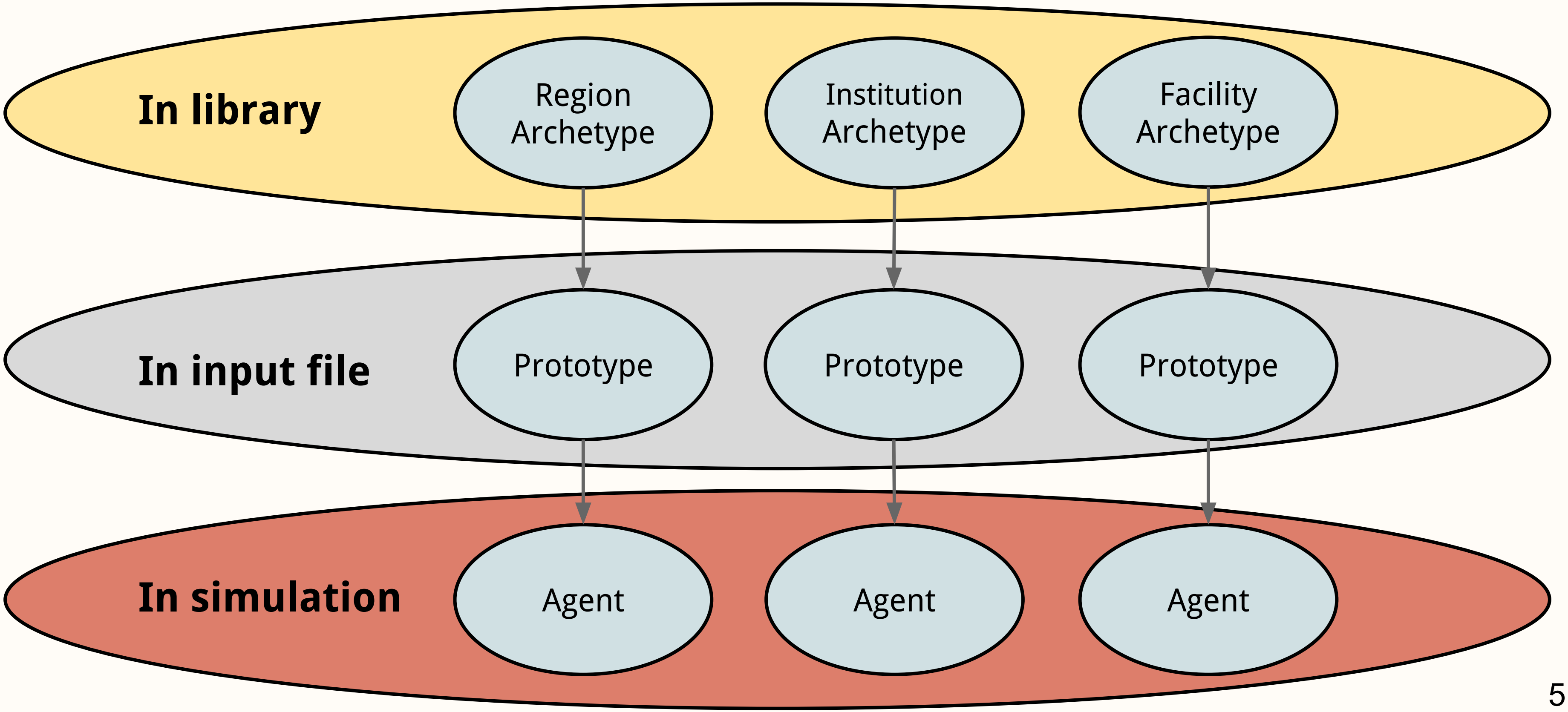
- Enable analysis based on tracking history of individual material objects
- Investigate: transportation, forensics, etc.

Cyclus Kernel Basics: Simulation

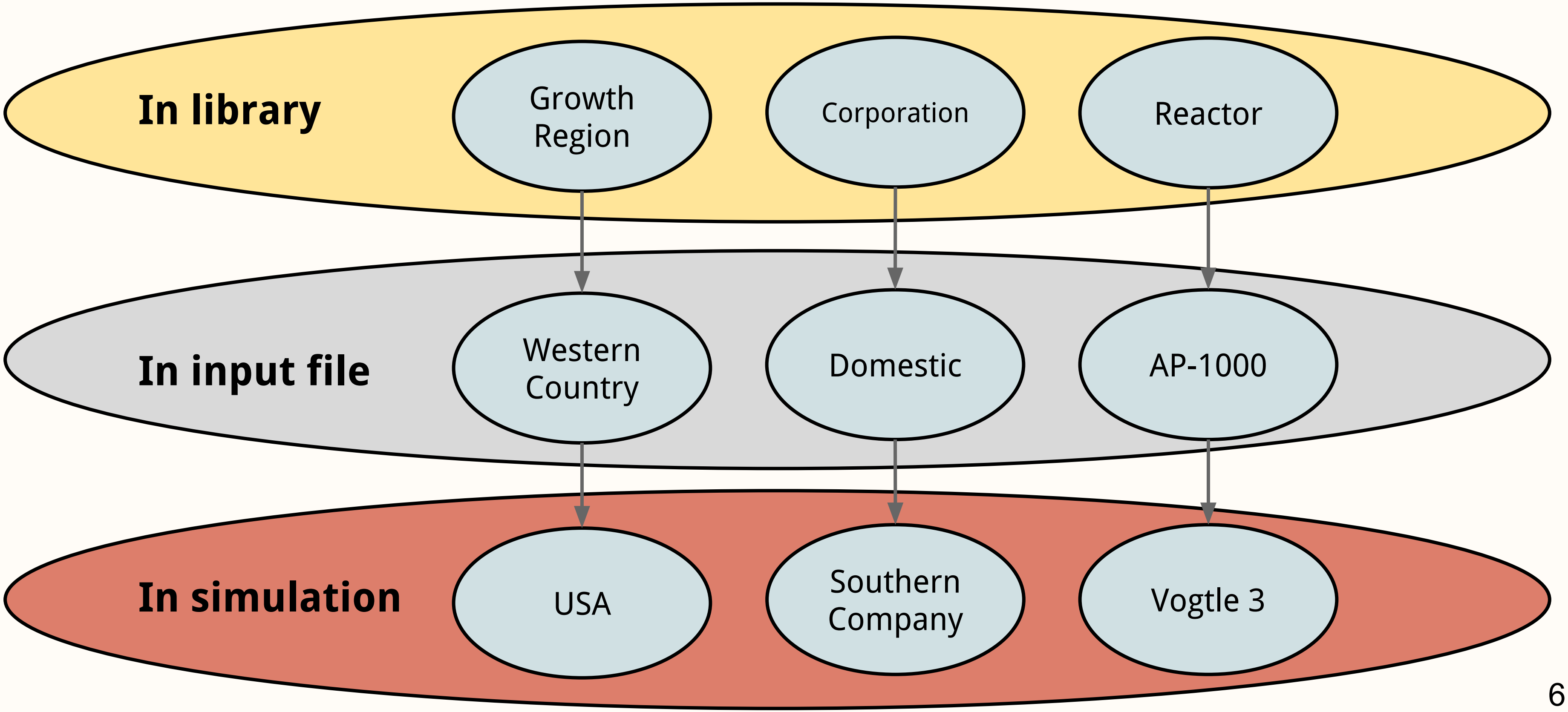
- Provides ecosystem for agent interaction and resource transactions

- Steps through time in uniform increments
  - Regions request that Institutions deploy Facilities
  - Dynamic Resource Exchange determines resource transactions
  - Regions request that Institutions decommission Facilities

Vocabulary


In library
Region Archetype
Institution Archetype
Facility Archetype
In input file
Prototype
Prototype
Prototype
In simulation
Agent
Agent
Agent

Vocabulary: Example



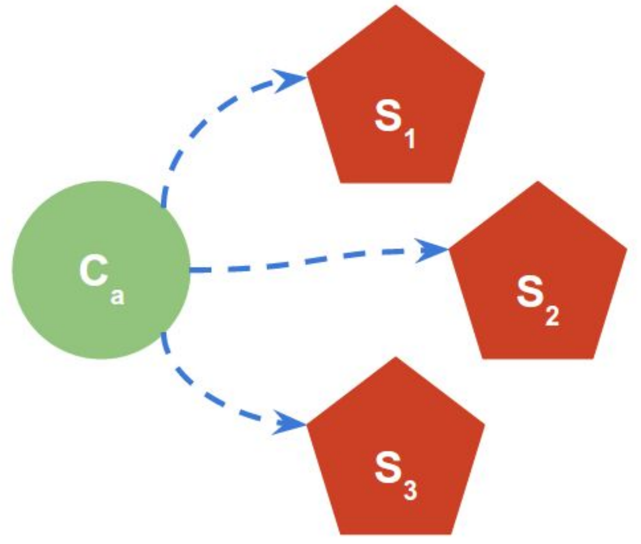
Dynamic Resource Exchange

- DRE: Core algorithm for fuel cycle simulation
- Recomputed at each time step
- Solves economic problem dynamically
  - no hard-coded supply-demand behavior
- Enables complicated/creative fuel cycles

Dynamic Resource Exchange

Request for Bids

Queries each requesting Agent in the simulation that ***demands*** a resource



Phase 1: Request for bids

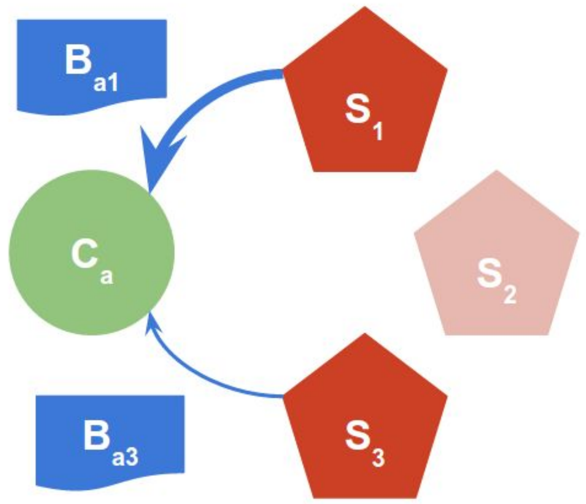
Dynamic Resource Exchange

Request for Bids

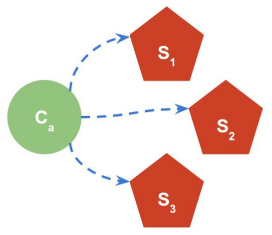
Queries each requesting Agent in the simulation that ***demands*** a resource

Response to Request for Bids

Queries each responding Agent in the simulation that ***supplies*** a resource



Phase 2: Response to request for bids



Phase 1: Request for bids

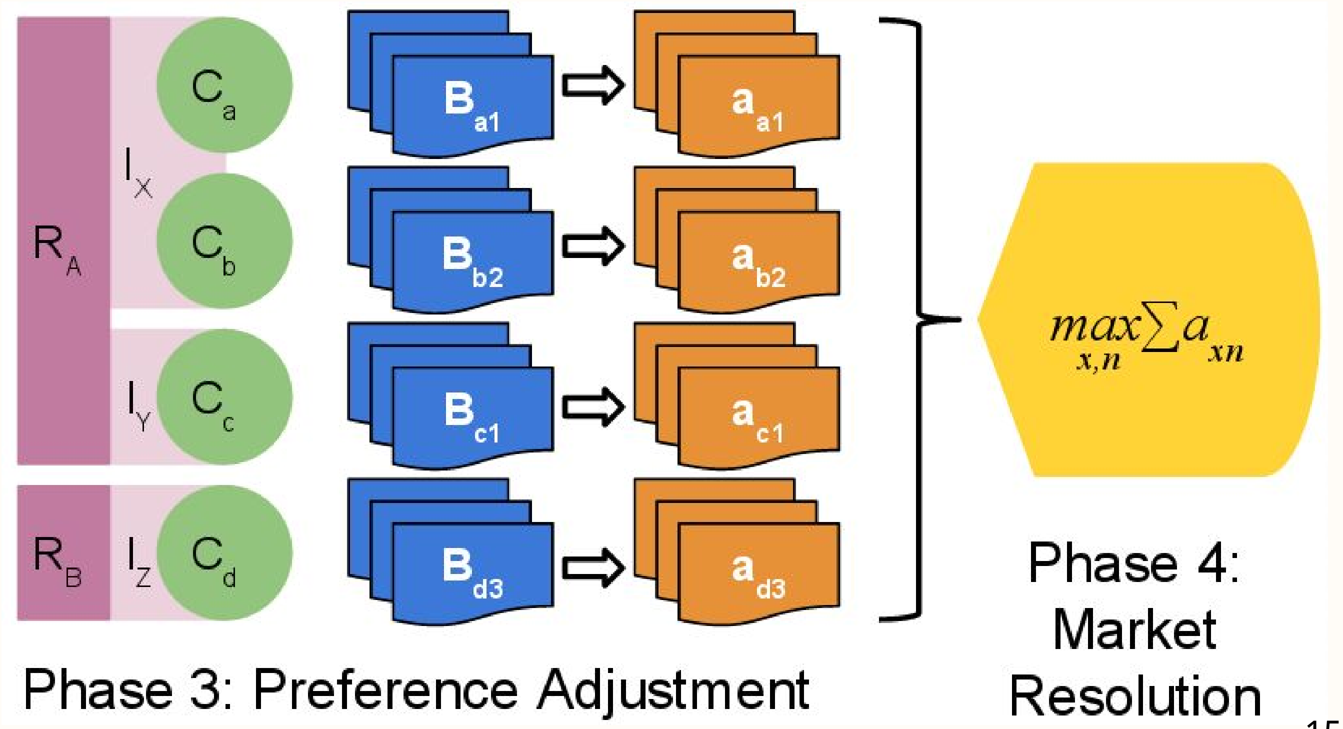
Dynamic Resource Exchange

Preference Adjustment

Agent ***reviews*** all matches; opportunity for preference adjustment

Solution

Matches ***selected*** for satisfiable requests



Dynamic Resource Exchange

Request for Bids

Queries each requesting Agent in the simulation that ***demands*** a resource

Response to Request for Bids

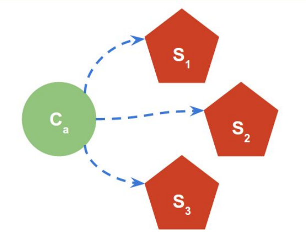
Queries each responding Agent in the simulation that ***supplies*** a resource

Preference Adjustment

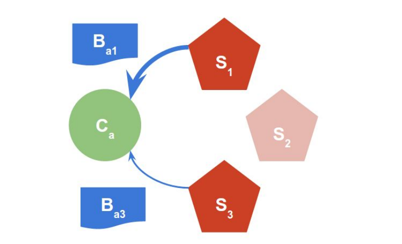
Agent ***reviews*** all matches; opportunity for preference adjustment

Solution

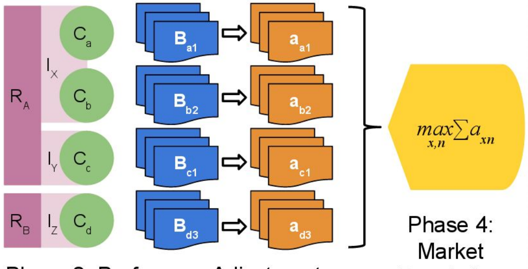
Matches ***selected*** for satisfiable requests



Phase 1: Request for bids



Phase 2: Response to request for bids



Phase 3: Preference Adjustment

Phase 4: Market Resolution

Dynamic Resource Exchange

Request for Bids

Queries each requesting Agent in the simulation that ***demands*** a resource

Response to Request for Bids

Queries each responding Agent in the simulation that ***supplies*** a resource

Preference Adjustment

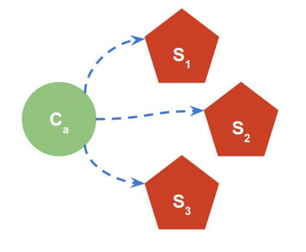
Agent ***reviews*** all matches; opportunity for preference adjustment

Solution

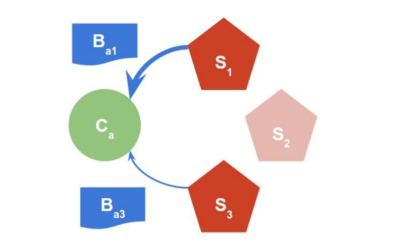
Matches ***selected*** for satisfiable requests

Trade Execution

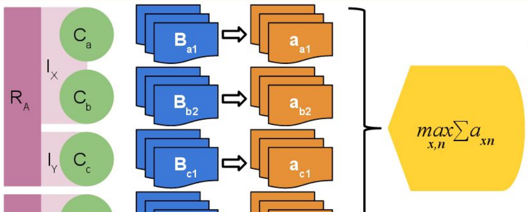
Material ***transaction*** takes place



Phase 1: Request for bids



Phase 2: Response to request for bids



Phase 3: Preference Adjustment

Phase 4: Market Resolution

LP Formulation

Variable	Description
H, h	Commodities
I, i	Bids
J, j	Requests
K, k	Capacities
c	Cost of commodity
x	Decision variable
β	Capacity coefficient
s	Supply capacity
d	Demand capacity

$$\min_{x} \quad z = \sum_{i \in I} \sum_{j \in J} \sum_{h \in H} c_{i,j}^{h} x_{i,j}^{h}$$

$$\text{s.t.} \quad \sum_{j \in J} \sum_{h \in H} \beta_{i,k} x_{i,j}^{h} \leq s_{i,k} \qquad \forall\, k \in K_I, \forall\, i \in I$$

$$\sum_{i \in I} \sum_{h \in H} \beta_{j,k} x_{i,j}^{h} \geq d_{j,k} \qquad \forall\, k \in K_J, \forall\, j \in J$$

$$x_{i,j}^{h} \geq 0 \qquad \forall\, x \in X$$

LP Supply Constraint: Example

Variable	Description
x	Decision variable
ε	Requested enrichment level
s	Supply capacity
f	Conversion function

$$\sum_{j \in J} f_{SWU}(\varepsilon_j) x_{i,j}^{EU} \leq s_{i,SWU}$$

$$\sum_{j \in J} f_{NU}(\varepsilon_j) x_{i,j}^{EU} \leq s_{i,NU}$$

LP Supply Constraint: General

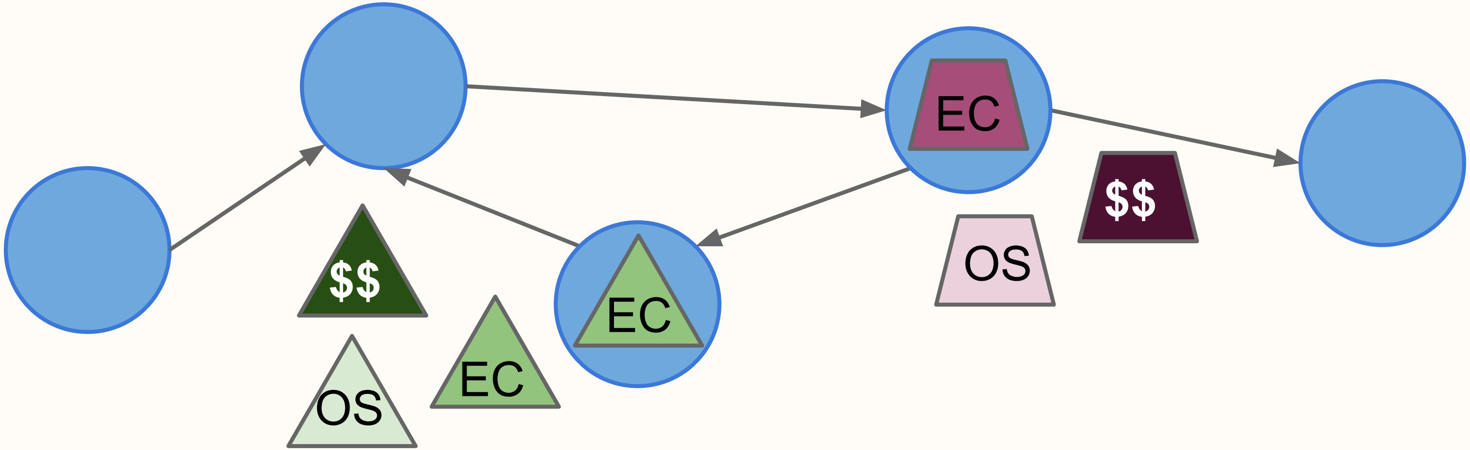
Variable	Description
H, h	Commodities
I, i	Bids
J, j	Requests
K, k	Capacities
$\beta_{i,k}(q_i^h)$	Conversion function
x	Decision variable
s	Supply capacity

These example constraints were a function of the isotopic profile of the request (quality, q_i). The conversion function of a supplier would be a function of the quality ($\beta_{i,k}(q_i^h)$):

$$\sum_{j \in J} \beta_{i,k}(q_j^h) x_{i,j}^h \leq s_{i,k} \; \forall \, k \in K_i^h, \forall \, i \in I, \forall \, h \in H$$

Mixed integer linear program (MILP) can guarantee exclusive trades

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

Features of 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

Cycamore: Standard Module Repository

Facilities

- Source
- Enrichment
- Fuel Fabrication
- Recipe Reactor
- Separations
- Storage
- Stream mixing
- Sink

Institutions

- Fixed Deployment
- Demand Response

Region

- Demand Growth

Ongoing Module Development

THE UNIVERSITY OF TEXAS AT AUSTIN

THE UNIVERSITY of TENNESSEE

Bright-lite

- Given an initial composition, calculate a burnup
- Given a target burnup and two or more streams, determine blend

Nuclear Fuel Inventory Module

- Provide ORIGEN capability to Cyclus
- Multiple possible applications including reactor, separations, fuel fabrication

Ongoing Module Development

cyCLASS

- Wrap CLASS neural network methods for
  - Fuel fabrication
  - Depletion

Consortium for Verification Technology

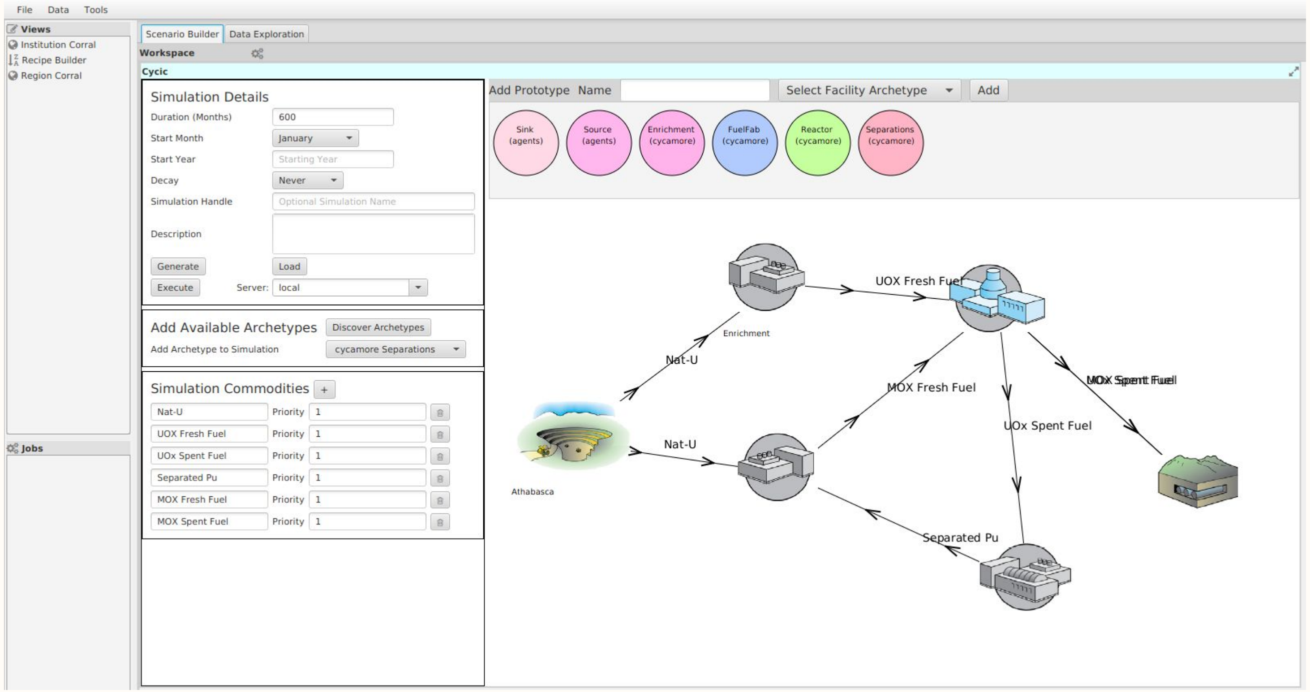
- Facility archetypes with clandestine behavior
- Region/Institution archetypes that track multi-lateral relationships

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

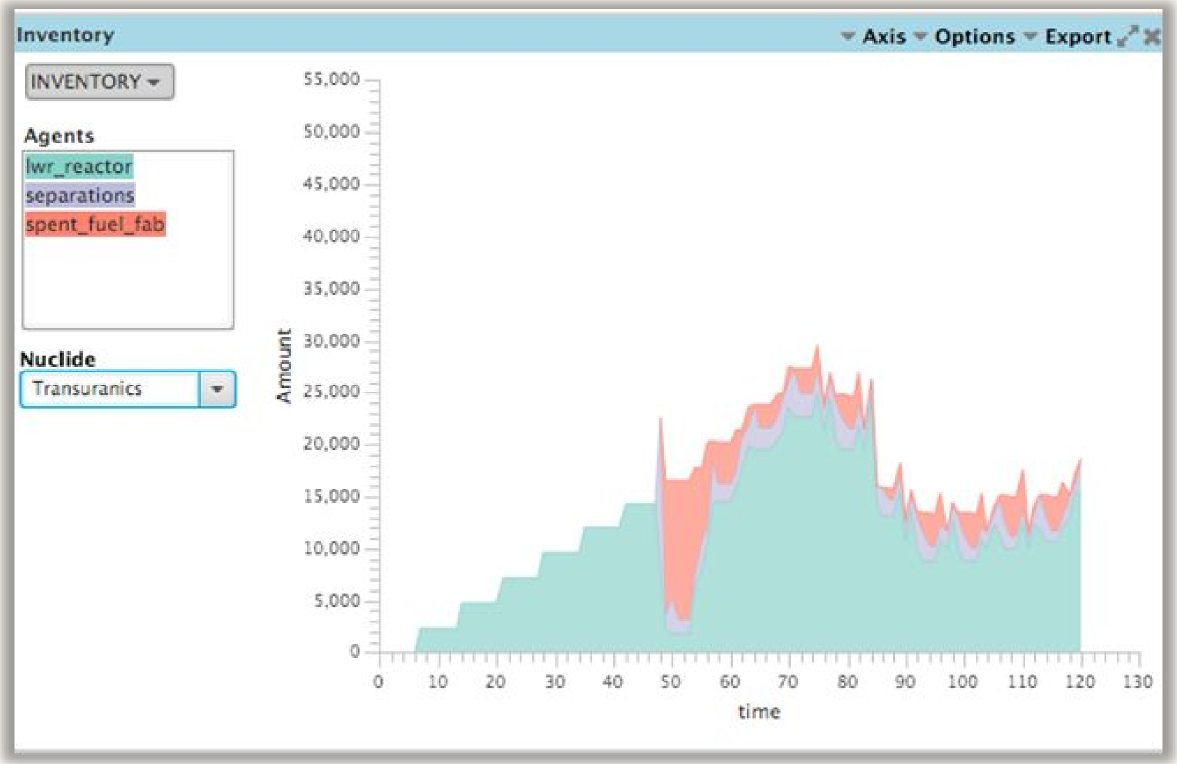
Cyclist Simulation Building

- Drag-and-drop interface
- Enables creative fuel cycle design
- Different modes for various user types



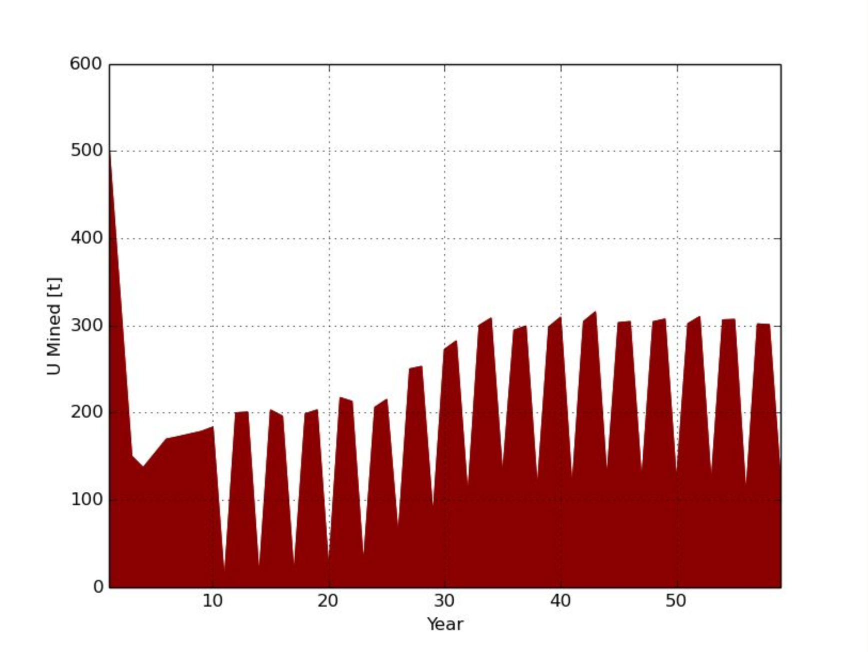
Cyclist Data Analysis Environment

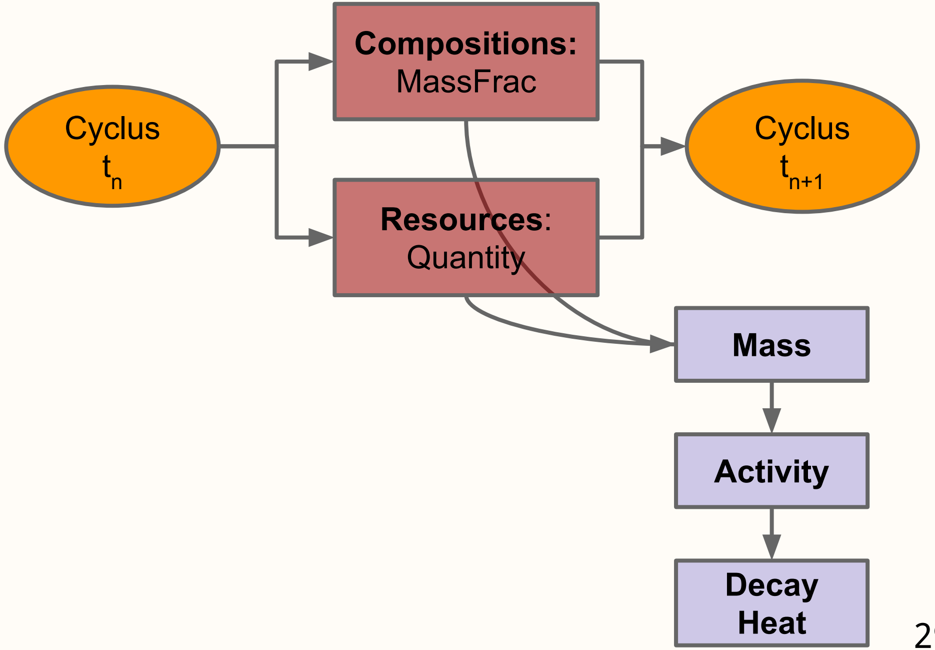
- Explore data dynamically
- Visualization mode matched to combination of data type and user needs.



Cymetric Extensible Tool

- Extensible metrics design: Users can add new metrics derived from existing metrics





Cyclus Application:

Fuel Cycle Options Transition Analysis

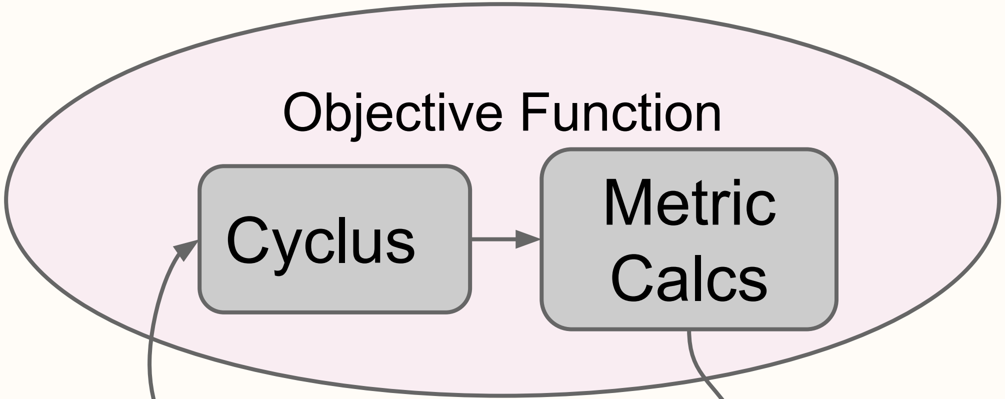
- Began as participant in code comparison/benchmarking
- Now performing transition analysis in parallel with
  - VISION (INL)
  - DYMOND (ANL)
  - ORION (ORNL)
- Confirm that tools can perform transition analysis with necessary metrics

Flanagan - Wed - 14:00; Mouginot - Fri - 9:40

Cyclus Application:

Deployment Optimization

- Transition deployment optimization study
- Large-scale parallelization
- Disruption analysis research



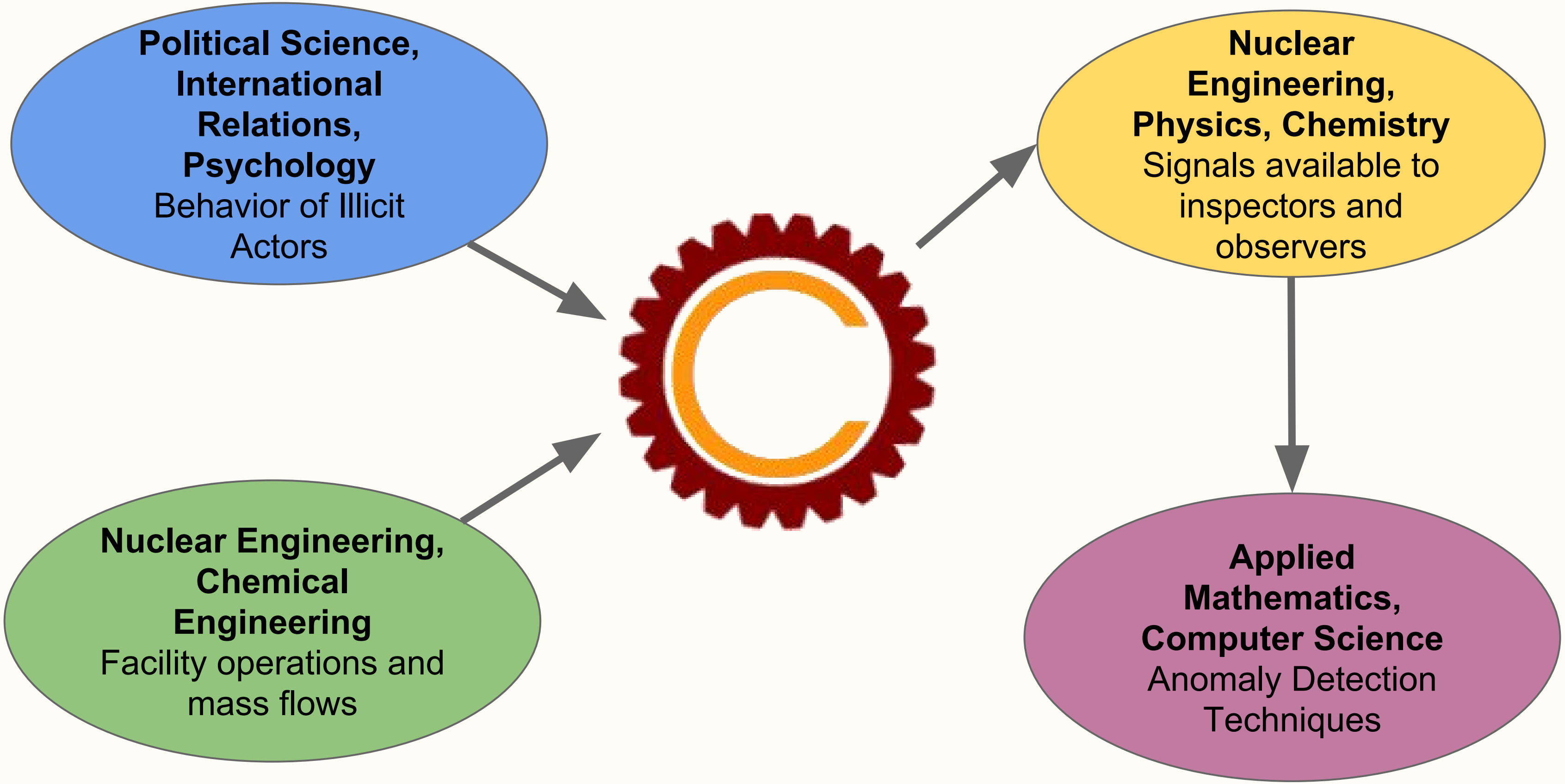
Facility	LWR				Repository			Fuel Fab	Objective
Year	1	2	3	…	1	2	…	…	
Trial 1	5	1	3	…	0	1	…	…	233.6
Trial 2	3	1	2	…	0	0	…	…	

Optimizer

Carlsen - Thu - 11:20

Cyclus Application:

Treaty Verification



Cyclus Documentation

www.fuelcycle.org

- Introduction to Cyclus Fundamentals
- User guide
- Archetype developer guide
- Kernel developer guide
- Cyclus enhancement proposals

Cyclus Funding History

Nuclear Energy University Programs

U.S. Department of Energy

Heavily leveraging support from:

Nuclear Energy University Programs U.S. Department of Energy

Argonne NATIONAL LABORATORY

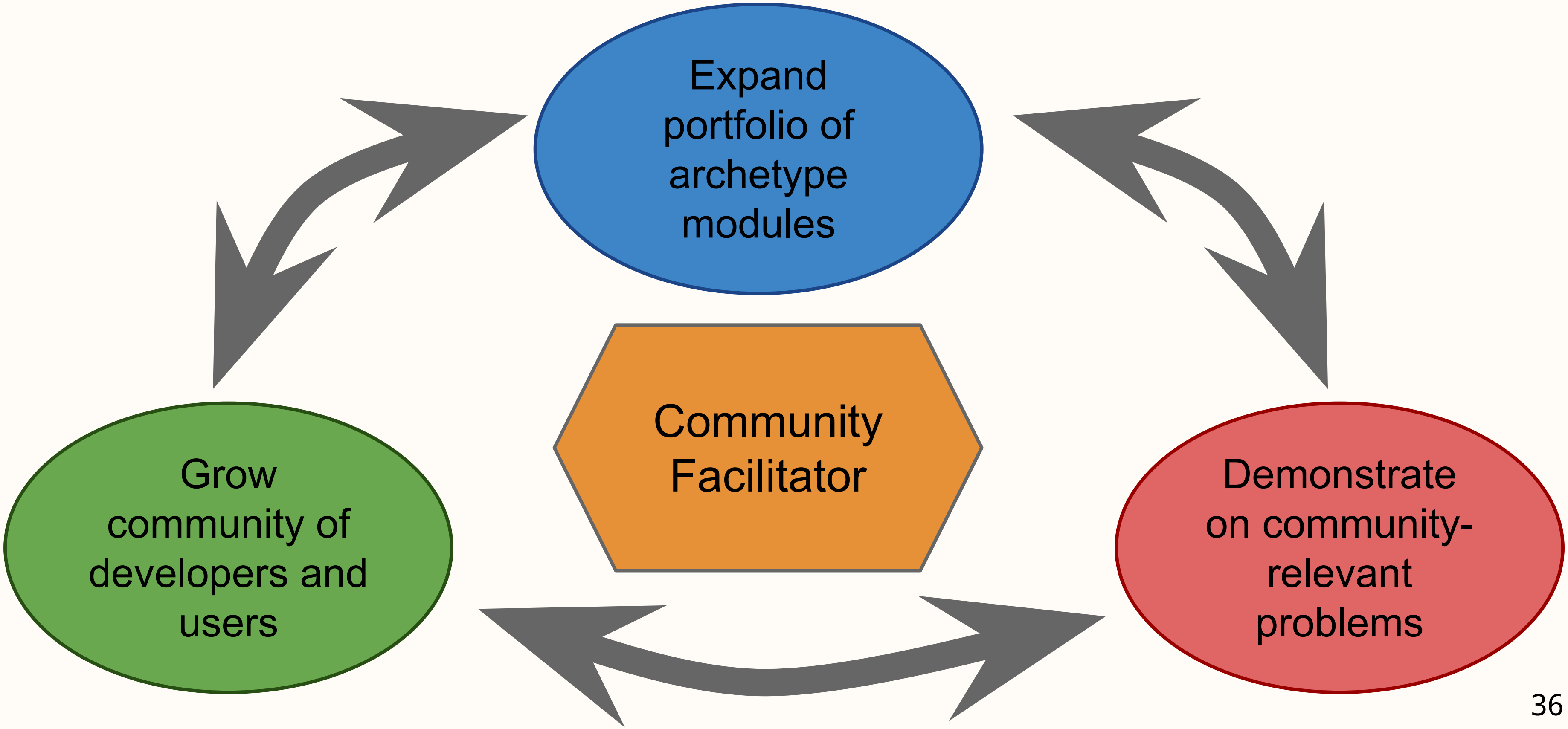
NSF

U.S.NRC

Potential Users

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

Moving Forward with Cyclus



Motivations for New Simulators

Current suite of simulators are difficult to benchmark:

- Commercial fuel cycle management
  - High-fidelity in-reactor simulation
  - Limited flexibility for novel systems/technologies
- Strategic decision making
  - Low-fidelity flow sheet approach
  - Complexity increases with need for detail
  - Limitations of software infrastructure
  - Low accessibility to non-technical audiences

Next Generation FCS Goals

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

Cyclus is Flexible

- Individual facility modeling
  - Startup/shutdown
  - Disruptions
- Discrete material tracking at nuclide level
  - Effects of individual facility performance
  - Forensic tracking of material object ownership
- No inherent physics assumptions
  - Low fidelity, systems level models
  - High fidelity, facility level models
- Agent-based approach incorporates social/behavior models

Cyclus v1.0: Released May 30, 2014

Carlsen, Robert W.; Gidden, Matthew; Huff, Kathryn; Opotowsky, Arrielle C.; Rakhimov, Olzhas; Scopatz, Anthony M.; Welch, Zach; Wilson, Paul (2014): **Cyclus v1.0.0.** figshare. http://dx.doi.org/10.6084/m9.figshare.1041745

Linear Programming (LP) Background

Minimization or Maximization Objectives

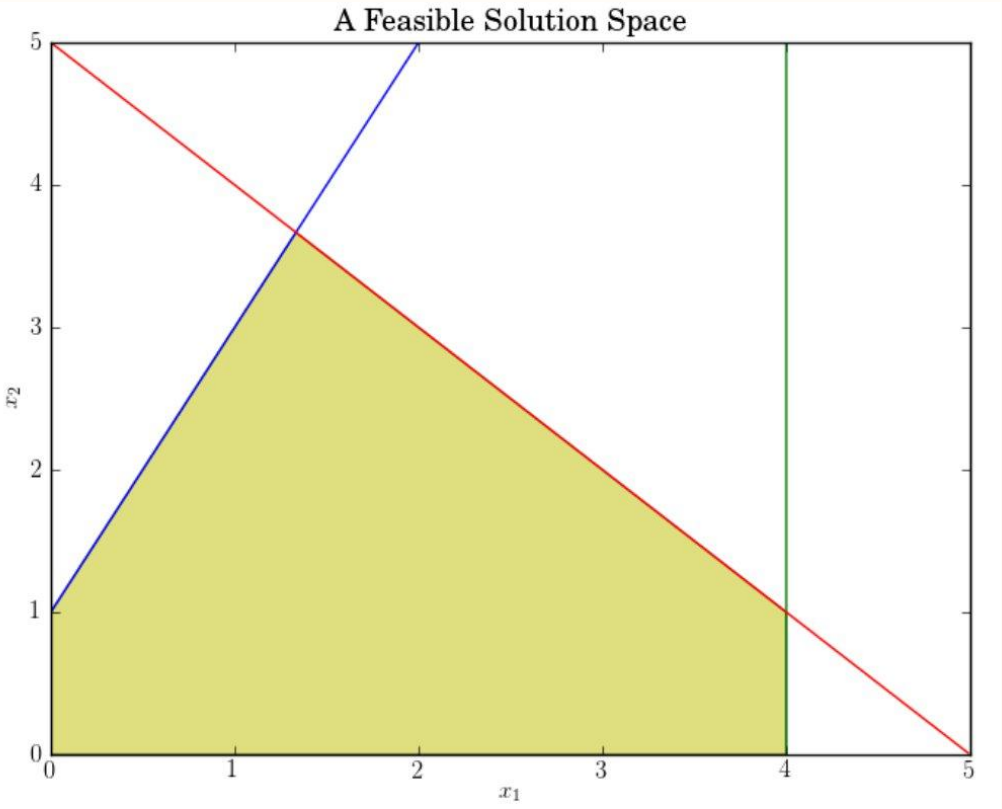
$$\min_{x} \; z = c^{\top} x$$

Equality Constraints

$$\text{s.t.} \quad Ax \geq b$$

$$x \geq 0$$

Variable	Description
c	Cost vector
x	Decision variable
A	Constraint matrix
b	Threshold vector



Mixed Integer-Linear Programming (MILP)

Required to allow two groups of consumers:

1. those that require *exclusive* orders
2. those that allow *partial* orders

$$J = J_p \cup J_e$$

Introduce a **binary variable**, $y_{i,j}^{h}$:

- 1 if consumer *j* is sent commodity *h* by supplier *i*
- restrict number of resource flows to consumer *j* to 1

$$\sum_{h \in H_j} \sum_{i \in I} y_{i,j}^{h} = 1 \; \forall j \in J_e$$

Variable	Description
H, h	Commodities
I, i	Bids
J, j	Requests
$y_{i,j}^{h}$	Binary variable

MILP Supply Constraint: General

$$\sum_{j \in J_p} \beta_{i,k}(q_j^h) x_{i,j}^h + \sum_{j \in J_e} \beta_{i,k}(q_j^h) y_{i,j}^h \tilde{x}_j^h \leq s_{i,k}^h \forall i \in I, \forall k \in K_i^h, \forall h \in H$$

Variable	Description
H, h	Commodities
I, i	Bids
J, j	Requests
K, k	Capacities
$\beta_{i,k}(q_i^h)$	Conversion function
x, y	Decision variable
s	Supply capacity

MILP Formulation

Variable	Description
H, h	Commodities
I, i	Bids
J, j	Requests
K, k	Capacities
c	Cost of commodity
x, y	Decision variable
β	Capacity coefficient
s	Supply capacity
d	Demand capacity

$$\min_{x,y} \; z = \sum_{h \in H} \sum_{i \in I} \sum_{j \in J_p} c_{i,j}^h x_{i,j}^h + \sum_{h \in H} \sum_{i \in I} \sum_{j \in J_e} c_{i,j}^h y_{i,j}^h \tilde{x}_j^h$$

$$\text{s.t.} \quad \sum_{j \in J_p} \beta_{i,k}(q_j^h) x_{i,j}^h + \sum_{j \in J_e} \beta_{i,k}(q_j^h) y_{i,j}^h \tilde{x}_j^h \leq s_{i,k}^h$$

$$\forall\, i \in I,\ \forall\, k \in K_i^h, \forall\, h \in H$$

$$\sum_{i \in I} \sum_{h \in H_j} \beta_{i,k}(q_j^h) x_{i,j}^h \geq d_j(H_j) \qquad \forall\, k \in K_j,\ \forall\, j \in J_p$$

$$\sum_{i \in I} \sum_{h \in H_j} \beta_{i,k}(q_j^h) y_{i,j}^h \tilde{x}_j^h \geq d_j(H_j) \qquad \forall\, k \in K_j,\ \forall\, j \in J_e$$

$$\sum_{h \in H} \sum_{i \in I} y_{i,j}^h = 1 \qquad \forall\, j \in J_e$$

$$x_{i,j}^h \geq 0 \qquad \forall\, x \in X$$

$$y_{i,j}^h \in \{0, 1\} \qquad \forall\, y \in Y$$

Greedy Solver Algorithm

```
order request portfolios by average preference;
forall the request portfolios do
    order requests by average preference;
    matched ← 0;
    while matched ≤ q_J and ∃ a request do
        get next request;
        order arcs by preference;
        while matched ≤ q_J and ∃ an arc do
            get next arc;
            remaining ← q_J - matched;
            to_match ← min{remaining, Capacity(arc)};
            matched ← matched + to_match;
        end
    end
end
```