

# An Integrated Used Fuel Disposition and Generic Repository Model

A Nuclear Engineering and Engineering Physics PhD Preliminary Report

Kathryn D. Huff

University of Wisconsin-Madison

September 1, 2011





# Outline

## ① Introduction

Motivation

Methodology

## ② Literature Review

Repository Capabilities within Systems Analysis Tools

Conceptual Discussion of Disposal Environments

Models of Radionuclide Transport

Models of Heat Transport

## ③ Modeling Paradigm

CYCLUS Simulator Paradigm

Repository Modeling Paradigm

## ④ Proposed Work

Demonstration Case

Base Case

Extensions

Summary



# Future Fuel Cycle Options

## Domestic Fuel Cycle Options

Title	Description	Challenges
Open	Once Through	High Temperatures, Volumes
Modified Open	Partial Recycling	Both high volumes and myriad fuel streams
Closed	Full Recycling	Myriad fuel streams

Table: Domestic Fuel Cycle Options

# Abstraction





# Outline

## ① Introduction

Motivation  
Methodology

## ② Literature Review

Repository Capabilities within Systems Analysis Tools  
Conceptual Discussion of Disposal Environments  
Models of Radionuclide Transport  
Models of Heat Transport

## ③ Modeling Paradigm

CYCLUS Simulator Paradigm  
Repository Modeling Paradigm

## ④ Proposed Work

Demonstration Case  
Base Case  
Extensions  
Summary



# Top Level Fuel Cycle Simulators



## Need For an Integrated Repository Model



# Clay Disposal Environments

Clay Repository Features			
Hydrology	Geochemistry	Design Concepts	Thermal Behavior
Very low conductivity High porosity (up to 0.5) Low effective porosity Slow water velocity diffusion dominated	Reducing Saline Saturated	no/bentonite/concrete backfill ~ 100m deep closed horizontal or vertical emplacement	alteration limited 100° C limit

**Table:** Clay geological repository concept demonstrates certain dominant physical phenomena.



# Granite Disposal Environments

Granite Repository Features			
Hydrology	Geochemistry	Design Concepts	Thermal Beha
Low porosity (~ 0.01) High Fracturation Low permeability High Water Velocity	Reducing in Near Field Slightly Oxidizing in Far Field Increasing saline with depth [?] Cement causes alkalinity [?] Saturated or Unsaturated	Single WP tunnels Carbon-Steel [?] or Copper overpack Bentonite buffer Crushed granite backfill [?] ~ 100m deep	Closed Bentonite Lim

**Table:** Granite repository concepts demonstrate certain dominant physical phenomena.



# Salt Disposal Environments

Salt Repository Features			
Hydrology	Geochemistry	Design Concepts	Thermal Beh
Dry Waste Package	Reducing in Near Field	Alcove Emplacement	180°C limit
Dry Backfill	Far Field Slightly Oxidizing	Crushed Salt Backfill	Heat induced
Saturated Far Field	Very saline brines	~ 100m deep	Closed
Very low permeability		Multiple Packages	limited data
Brine pockets in far field		Breached only from intrusion	

**Table:** Salt geological repository concept demonstrates certain dominant physical phenomena.



# Deep Borehole Disposal Environment

Borehole Repository Features			
Hydrology	Geochemistry	Design Concepts	Thermal Behav
Crystalline rock Low porosity (~ 0.01) Limited fracturation at depth Rock Permeability (~ $10^{-19}$ ) EBS Permeability (~ $10^{-16}$ ) Very Limited Upward Flow	Reducing at depth Less Reducing at surface limited solubility enhanced sorption high salinity saturated	~ 5km deep disposal in lower 2km 1km bentonite seal bentonite grout bentonite plugs 400 packages per borehole closed	cracking unim may affect flo high conducti high density

**Table:** Borehole geological repository concept demonstrates certain dominant physical phenomena.

## Waste Form Release Models



# Waste Package Failure Models





# Transport Through Buffer Material



# Transport Through Geology



# Impact of Repository Designs



# Heat Limits In Geology



# Analytical Models



## Detailed Techniques

## Lumped Parameter Technique





# Outline

## ① Introduction

- Motivation
- Methodology

## ② Literature Review

- Repository Capabilities within Systems Analysis Tools
- Conceptual Discussion of Disposal Environments
- Models of Radionuclide Transport
- Models of Heat Transport

## ③ Modeling Paradigm

- CYCLUS Simulator Paradigm
- Repository Modeling Paradigm

## ④ Proposed Work

- Demonstration Case
- Base Case
- Extensions
- Summary

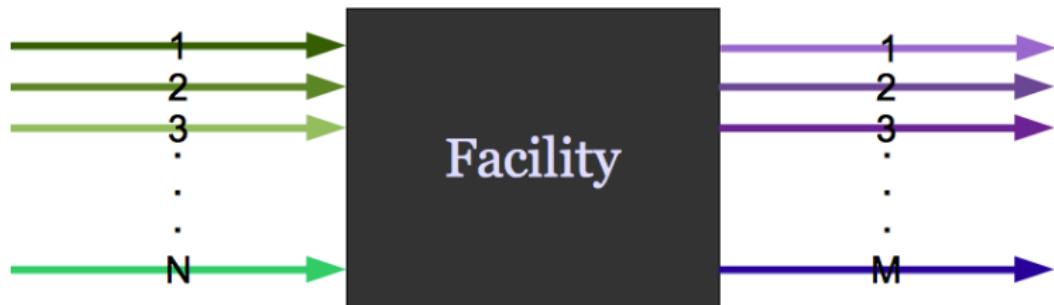


## Cyclus Modular Architecture and Open Development

The combination of modular encapsulation within the software architecture and an open development paradigm allows for collaboration at multiple levels of simulation detail and data security.



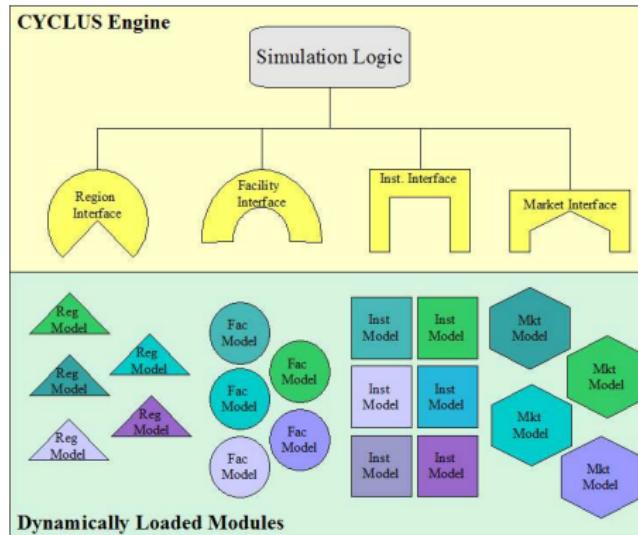
## Encapsulation



**Figure:** Regions, Institutions, Facilities, and Markets are all black boxes.



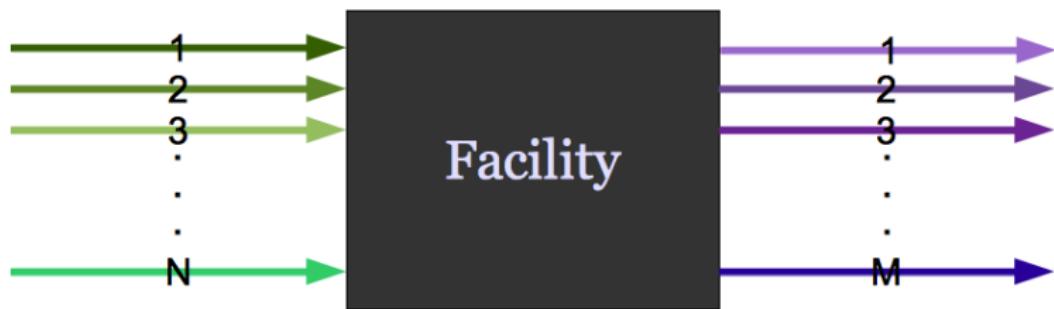
## Module Interfaces



**Figure:** Well defined model interfaces facilitate model interchange. The user may choose the model at their desired level of detail.



## Facilities Are Black Boxes



**Figure:** Each facility in the simulation makes requests and offers to fill its stocks and empty its inventory respectively.



## Facilities Are Black Boxes



Figure: A facility might only make offers.



## Facilities Are Black Boxes

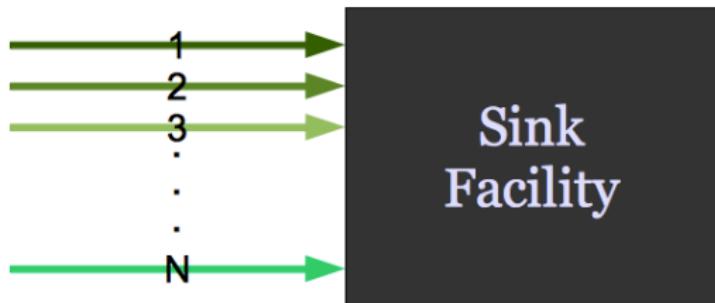


Figure: A facility might only make requests.



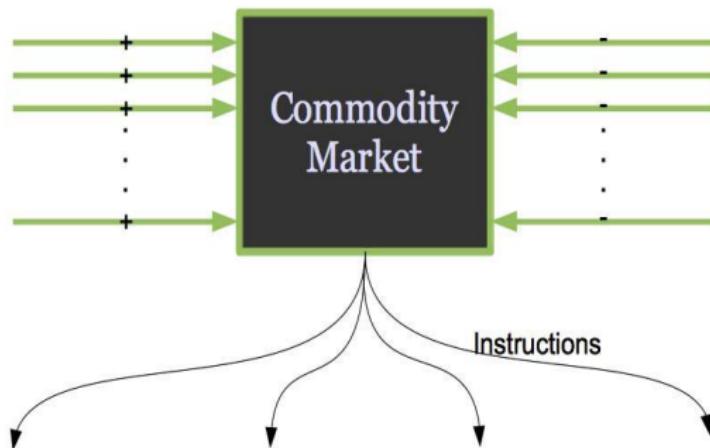
## Each Commodity is Associated with a Market



Figure: A market receives offers and requests concerning its commodity.



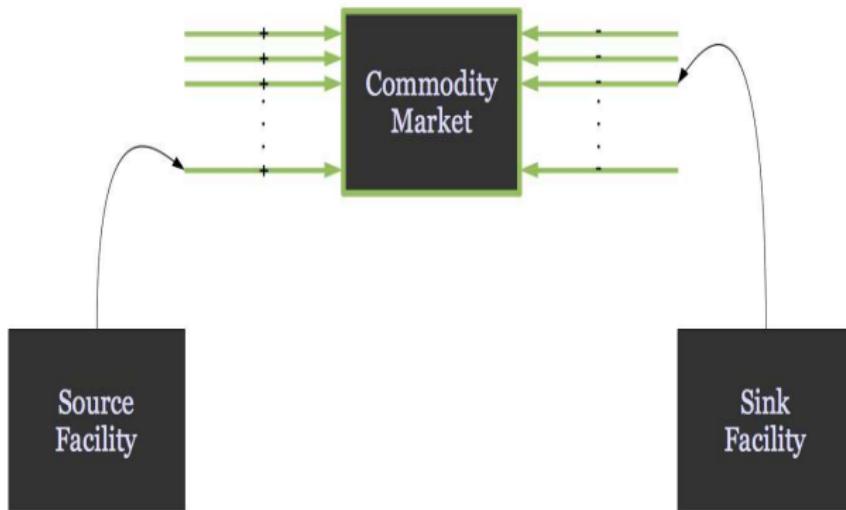
## The Market Solves the Matching Problem



**Figure:** When the Market's arbitrary algorithm solves the matching problem, the Market sends instructions to the offering facilities.



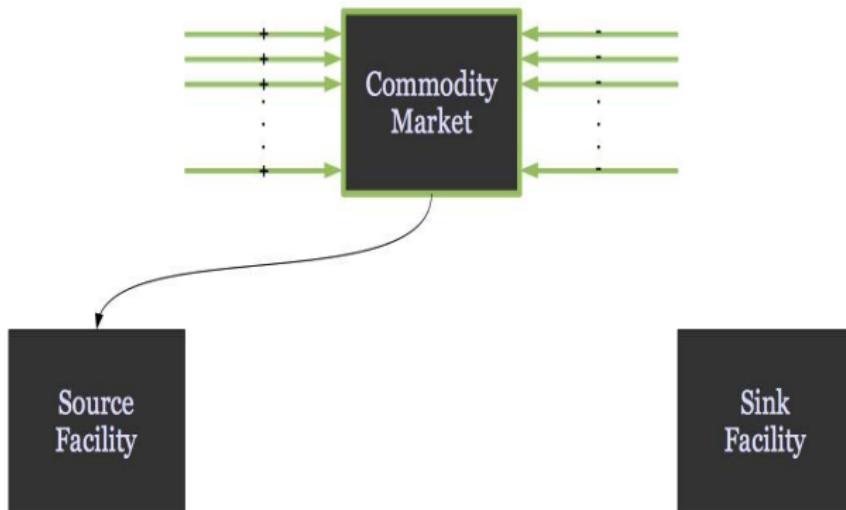
## A Simple Example



**Figure:** The source sends an offer and the sink sends a request.



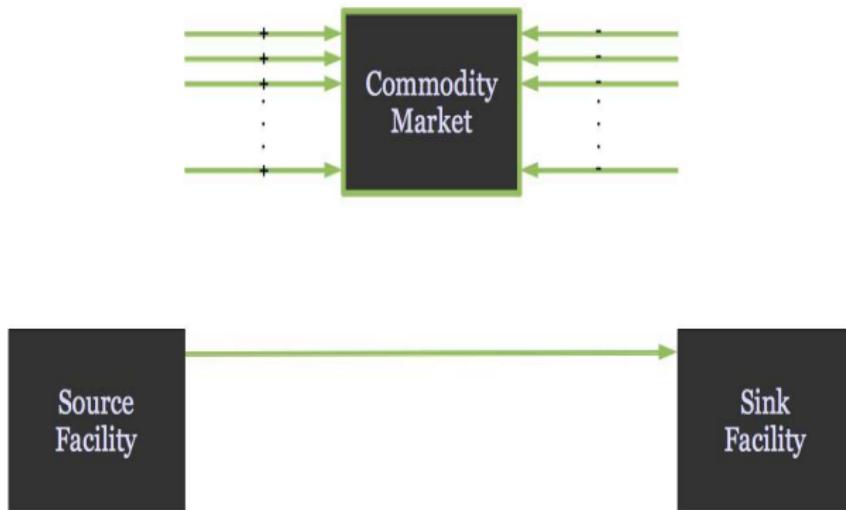
## A Simple Example



**Figure:** The Market solves the problem and instructs the source facility to send a certain amount to the sink facility.



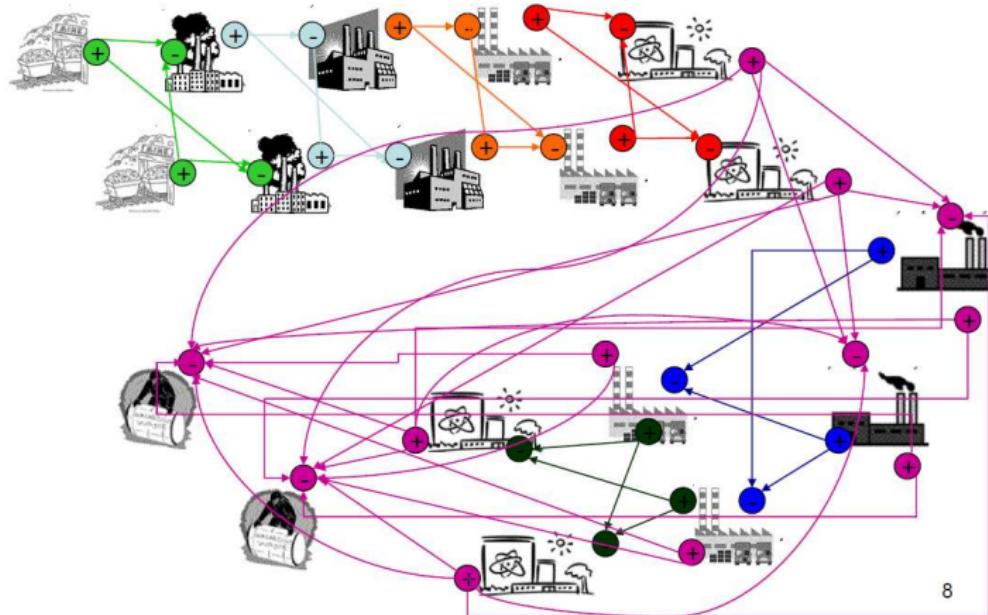
## A Simple Example



**Figure:** The source facility sends the material directly to the sink facility.



## This Market Model Scales for Complex Systems



**Figure:** Well designed interfaces and strict encapsulation support scalability of the Market-based simulation paradigm [?]



## Dynamic Module Loading : Developer

With a dynamic, plug-in implementation, the simulation logic is independent of the available models and models are loaded as shared libraries at runtime.

	Model.cpp
	<pre>#include Model.hpp #include dlfcn.h .  mdl_ctr* loadModel(name){     void* model = dlopen(name.c_str(), RTD_LAZY)     mdl_ctr* new_model = (mdl_ctr*)dlsym(model, "construct");      .     .      return new_model; }</pre>
	RecipeReactor.cpp
	<pre>#include RecipeReactor.hpp .  extern "C" Model* construct() {     return new RecipeReactor(); }  .</pre>

**Figure:** Dynamic c library loading separates simulation logic from knowledge of available models, supporting extensions by developers with minimal lines of code.



## Dynamic Module Loading : User

With a dynamic, plug-in implementation, the simulation logic is independent of the available models and models are loaded as shared libraries at runtime.

	input.xml
	<pre>&lt;simulation&gt;   &lt;startYear&gt;1962&lt;/startYear&gt;   &lt;duration&gt;1200&lt;/duration&gt;   &lt;region&gt;     &lt;name&gt;UChicago&lt;/name&gt;     &lt;DeployRegionModel&gt;       &lt;deployment&gt;         &lt;facility&gt;           &lt;name&gt;ChiPile1&lt;/name&gt;           &lt;model&gt;RecipeReactor&lt;/model&gt;         &lt;/facility&gt;         &lt;year&gt;1942&lt;/year&gt;       &lt;/deployment&gt;     &lt;/DeployRegionModel&gt;   .   .   &lt;/region&gt;   .   . &lt;/simulation&gt;</pre>

Figure: XML input parsing and a relaxNG schema provide a simplified XML interface is available for the end user to define available module implementations.



# Open Source Repository

This open source repository provides a centralized location for documentation, developer history, and unhindered developer access.

code.google.com/p/cyclus/source/browse/#svn%2Ftrunk%2Fsrc

Apple | Yahoo! | Google Maps | News | Wikipedia | Popular | proxy | Note in Reader | http://frit.iss.wisc.edu | NEUPFC6 | Office - Windows Live | Other Bookmarks katyhuff@gmail.com | My favorites | Profile | Sign out

## cyclus

A Nuclear Fuel Cycle Simulation Code from the University of Wisconsin - Madison

Project Home Downloads Wiki Issues Source Administrator

Checkout Browse Changes Search Trunk Request code review

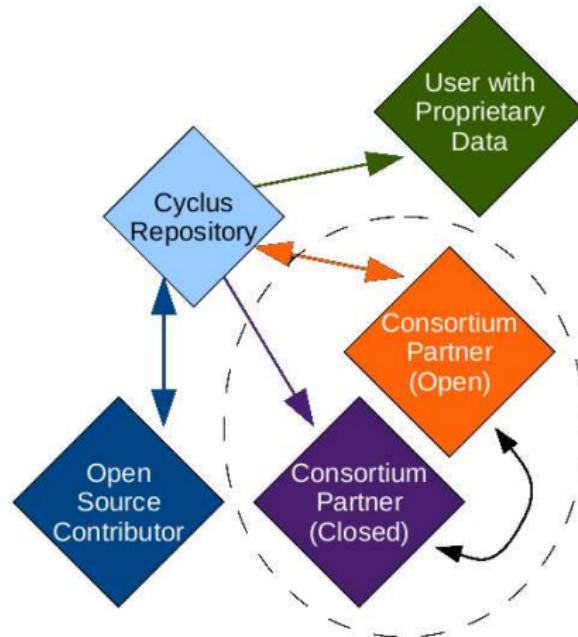
Source path: svn/

Directories	Filename	Size	Rev	Date	Author
svn	<a href="#">App.cpp</a>	2.0 KB	r314	May 19, 2011	katyhuff
branches	<a href="#">CMakeLists.txt</a>	3.5 KB	r289	Apr 30, 2011	katyhuff
doc	<a href="#">Commodity.cpp</a>	903 bytes	r111	Jul 24, 2010	katyhuff
trunk	<a href="#">Commodity.h</a>	2.3 KB	r111	Jul 24, 2010	katyhuff
input	<a href="#">Communicator.cpp</a>	1.8 KB	r117	Jul 29, 2010	katyhuff
src	<a href="#">Communicator.h</a>	1.6 KB	r117	Jul 29, 2010	katyhuff
	<a href="#">InputXML.cpp</a>	7.0 KB	r301	May 5, 2011	katyhuff
	<a href="#">InputXML.h</a>	9.2 KB	r115	Jul 28, 2010	katyhuff
	<a href="#">Logician.cpp</a>	8.4 KB	r240	Feb 23, 2011	Matthew.Gidden
wiki	<a href="#">Logician.h</a>	7.2 KB	r166	Oct 19, 2010	katyhuff
	<a href="#">Material.cpp</a>	22.8 KB	r334	Jun 4, 2011	Matthew.Gidden
	<a href="#">Material.h</a>	14.5 KB	r334	Jun 4, 2011	Matthew.Gidden

Your project is using approximately 7.0 MB out of 4096 MB total quota.  
You can [reset this repository](#) so that svn:sync can be used to upload existing code history.



## 'Modified Open' Source



**Figure:** License, architecture, and development paradigm allow varying levels of code sharing and data security.



## Version Control

This open source repository employs a version control system for provenance, developer access, and reproducibility of results.

code.google.com/p/cyclus/source/browse/#svn%2Ftrunk%2Fsrc

Apple | Yahoo! | Google Maps | News | Wikipedia | Popular | proxy | Note in Reader | http://frt.iss.wisc.edu | NEUPFC6 | Office - Windows Live | Other Bookmarks katyhuff@gmail.com | My favorites | Profile | Sign out

### cyclus

A Nuclear Fuel Cycle Simulation Code from the University of Wisconsin - Madison

Project Home Downloads Wiki Issues Source Administrator

Checkout Browse Changes Search Trunk Request code review

Source path: svn/

Directories	Filename	Size	Rev	Date	Author
svn	<a href="#">App.cpp</a>	2.0 KB	r314	May 19, 2011	katyhuff
branches	<a href="#">CMakeLists.txt</a>	3.5 KB	r289	Apr 30, 2011	katyhuff
doc		903 bytes	r111	Jul 24, 2010	katyhuff
trunk	<a href="#">Commodity.cpp</a>	2.3 KB	r111	Jul 24, 2010	katyhuff
input	<a href="#">Commodity.h</a>	1.8 KB	r117	Jul 29, 2010	katyhuff
src	<a href="#">Communicator.cpp</a>	1.6 KB	r117	Jul 29, 2010	katyhuff
	<a href="#">Communicator.h</a>	7.0 KB	r301	May 5, 2011	katyhuff
	<a href="#">InputXML.cpp</a>	9.2 KB	r115	Jul 28, 2010	katyhuff
	<a href="#">InputXML.h</a>	8.4 KB	r240	Feb 23, 2011	Matthew.Gidden
wiki	<a href="#">Logician.cpp</a>	7.2 KB	r166	Oct 19, 2010	katyhuff
	<a href="#">Logician.h</a>	22.8 KB	r334	Jun 4, 2011	Matthew.Gidden
	<a href="#">Material.cpp</a>	14.5 KB	r334	Jun 4, 2011	Matthew.Gidden
	<a href="#">Material.h</a>				

Your project is using approximately 7.0 MB out of 4096 MB total quota.  
You can [reset this repository](#) so that svn:sync can be used to upload existing code history.



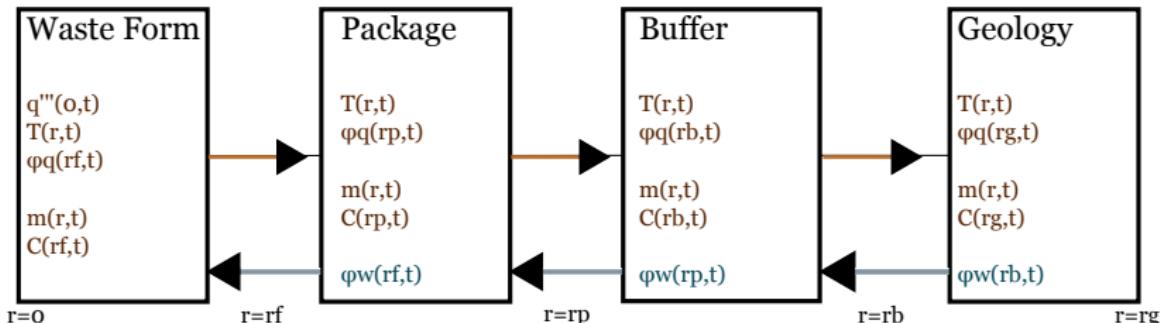
## Testing Framework

A testing framework built on a cross-platform, multi-language build system (CMake) allows developers to incorporate unit and integration tests into their code before it is committed.



## Nested Components

### Quantities Calculated Each Timestep



**Figure:** The nested components supply thermal flux and concentration information to each other at the boundaries.

# Waste Stream



# Waste Form



# Waste Package



# Buffer



# Geological Environment





# Outline

## ① Introduction

Motivation  
Methodology

## ② Literature Review

Repository Capabilities within Systems Analysis Tools  
Conceptual Discussion of Disposal Environments  
Models of Radionuclide Transport  
Models of Heat Transport

## ③ Modeling Paradigm

CYCLUS Simulator Paradigm  
Repository Modeling Paradigm

## ④ Proposed Work

Demonstration Case  
Base Case  
Extensions  
Summary



i++i

i++i

i++*i*



i++i

i++i

i++*i*



i++i

i++i

i++*i*



i++i

i++i

i++*i*



i++i

i++i

i++*i*



i++i

i++i

i++*i*



## Demonstration Case : Concept



## Demonstration Case : Testing



## Base Case : Component Abstraction



## Base Case : System Level Abstraction



## Base Case : Concept



## Base Case : Testing



## Extensions : Abstraction



## Extensions : Testing



## References I

[1] ANDRA.

Granite: Evaluation de la faisabilite du stockage geologique en formation granite.  
Technical report, Paris, 2005.

[2] W. von Lensa, R. Nabbi, and M. Rossbach.

*RED-IMPACT Impact of Partitioning, Transmutation and Waste Reduction Technologies on the Final Nuclear Waste Disposal.*  
Forschungszentrum Julich, 2008.