5-2016

Development and Application of Nuclear Fuel Cycle Simulators for Evaluating Potential Fuel Cycle Options

Jennifer Lynn Littell
University of Tennessee - Knoxville, jlittell3@gmail.com

To the Graduate Council:

I am submitting herewith a thesis written by Jennifer Lynn Littell entitled "Development and Application of Nuclear Fuel Cycle Simulators for Evaluating Potential Fuel Cycle Options." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

Steven E. Skutnik, Major Professor

We have read this thesis and recommend its acceptance:

Ivan G. Maldanado, Ondrej Chvala

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Development and Application of Nuclear Fuel Cycle Simulators for Evaluating Potential Fuel Cycle Options

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Jennifer Lynn Littell

May 2016

For my mom, who motivates me to look on the bright side and never give up.

Acknowledgements

I would like to express my thanks my advisor, Dr. Steve Skutnik, for giving me the opportunity to work on this project, allowing me to adjust my research path in order to more fully pursue my interests, and providing guidance throughout my graduate experience. I also want to thank Dr Ondrej Chvala and Dr Ivan Maldonado for serving on my committee and providing constructive input and advice in the final stages of my thesis.

In addition to the professors at UT, I want to extend my appreciation to the members of the Fuel Cycle Options group at ORNL who I worked closely with: Andy Worall, Eva Sunny, Joshua Peterson, and Jeffrey Powers. The work I performed at ORNL gave me an important learning opportunity about the much broader aspects of fuel cycle studies.

This work would not have been possible without the monetary support from the US Department of Energy, Office of Nuclear Energy's NEUP and the Nuclear Engineering Science Laboratory Synthesis internship program through ORNL.

Abstract

The Nuclear Fuel Cycle Evaluation and Screening Study was chartered by the DOE in order to weigh the relative benefits and challenges of potential future fuel cycle options. In order to efficiently implement these alternative fuel cycles, the transition from the current once-through cycle to the most promising of these potential fuel cycles must also be analyzed. This analysis requires the use of fuel cycle simulators which have the capability to quickly calculate the mass flows between numerous facilities over hundreds of years. In this work, Cyclus and ORION have both been utilized to simulate transitions from the current once-through fuel cycle to one which involves fast reactors with continuous reprocessing of spent fuel. This transition was found to take approximately 140 years while staying within the constraints of maintaining the mass of excess plutonium in storage below 100 tonnes, introducing fast reactors gradually in the first years, and waiting until 2050 to begin reprocessing. Before completing this transition analysis, Cyclus was also used to create a handful of less sophisticated simulations in order to demonstrate its range of capabilities.

In addition to using Cyclus to contribute to the Evaluation and Screening Study, this work contains the beginning of an ORIGEN-based repository of modules for use with Cyclus. This repository, called CyBORG, incorporates ORIGEN's isotopic depletion and decay calculations directly into Cyclus. The first module added to CyBORG is a reactor facility which uses ORIGEN to calculate its spent fuel isotopics based on reactor specifications from the user such as assembly type, fresh fuel recipe, and power capacity. By creating problem-specific cross section libraries for the depletion calculations, combined with ORIGEN's capability to track more than

2000 isotopes, accurate spent fuel isotopics can be created which will reflect how any changes to the system affect the availability of fissile material.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The creation of the Atomic Energy Commission in 1946 and President Eisenhower's Atoms for Peace speech in 1953 began a large push to utilize nuclear technology for commercial power production [1]. The concept of producing huge amounts of clean energy with nuclear reactors was an undeniably attractive alternative for a technology that had previously been known primarily for its destructive powers. At this time the Experimental Breeder Reactor I and the reactors which were used to produce plutonium for the war effort were already operational [1]. The first large-scale nuclear reactor to produce power for commercial use was started up in Shippingport, Pennsylvania in 1957 and from here, the nuclear power industry saw incredible growth over the next 30 years. In Figure 1.1 the rapid construction of nuclear reactors in the 70's and 80's is illustrated by the amount of nuclear power capacity in each year. This figure also illustrates the effects of accidents such as Three Mile Island (1979) and Chernobyl (1986) coupled with growing concerns over how spent nuclear fuel should be handled. Increased apprehensions about the materials produced in the nuclear fuel cycle not only hurt the growth of the existing nuclear power industry, but has had lasting effects on our ability to introduce new technologies and stages to the fuel cycle. However, while apprehension about nuclear power is still high in some areas, society is also becoming more and more aware of the need for a clean energy source to cut down on carbon emissions, as evidenced by the Environmental Protection Agency's Clean Power Plan [2]. Although the original version of this

Plan inadvertently discounted existing nuclear plants [3], incentives to move towards clean energy production were clearly there. There are many alternative energy sources that can contribute to the amount of clean energy generated such as wind and solar; however, nuclear is the largest scale option with the highest capacity factor [4, 5]. With this growing need for nuclear energy, along with the safety enhancements created to avoid large-scale accidents [6], as well as significant efforts by the nuclear community to educate the public on its benefits and safety features [7, 8], it is the hope that the nuclear industry will soon resume the rapid growth it experienced in earlier years. This changing tide in the future of nuclear power provides an opportunity for not only growth, but also reorganization of the nuclear fuel cycle in order to create a more sustainable system.

Improvements to the nuclear fuel cycle can include determining more sustainable approaches for the current reactor fleet as well as planning the future direction of the fuel cycle so that current research can be focused in those areas. In order to efficiently study many different possibilities for each of these areas, fuel cycle simulators are needed. These codes can quickly simulate the mass flows between each facility in a nuclear fuel cycle. This can provide valuable information about the amount of natural resources needed to sustain a reactor fleet in the coming years, reprocessing capacities needed for recycling spent fuel, or how much spent fuel will be generated. Some simulators also incorporate physics calculations such as radioactive

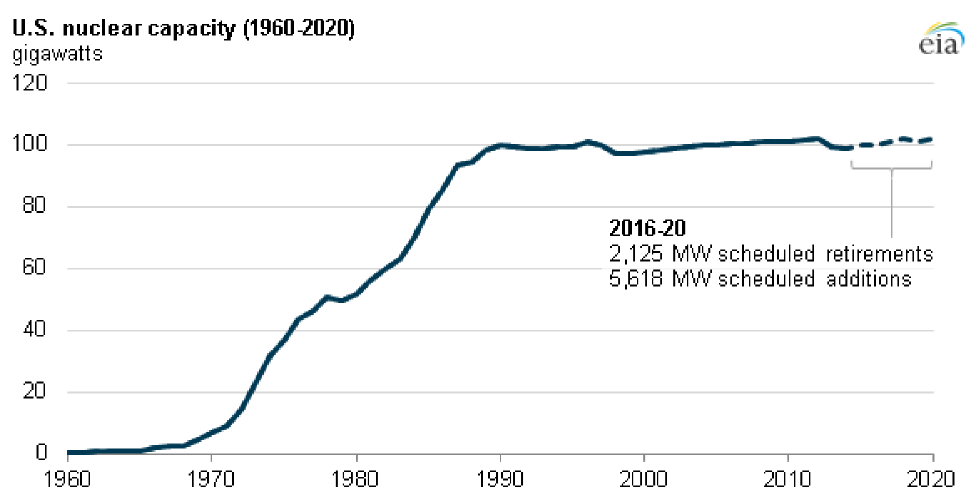

Figure 1.1: Timeline of US nuclear power capacity which illustrates the effects of Three Mile Island and Chernobyl [9].

decay and reactor depletion which allow the code to inform on things like effects of reactor burnup on the fuel cycle and radioactivity of spent fuel at various times. The work presented here seeks to utilize fuel cycle simulators to investigate possible approaches for introducing new technologies into the current fuel cycle as well as to contribute to the development of a simulator which has accurate physics calculations built in. The simulators used for the optimization work are Cyclus, an open-source code pioneered by the University of Wisconsin and continuously being developed at various other universities in the US [10], and ORION, an established code developed for more than 15 years at the National Nuclear Laboratory in the UK [11]. The simulator development portion of this work is focused primarily on incorporating ORIGEN, an isotopic depletion and decay code, into a repository of modules for use with Cyclus [12].

1.1 Evaluation and Screening Study

Exploring alternatives to the current once-through nuclear fuel cycle is imperative to the future of nuclear power. While there are few new reactors being constructed, there is not a clear plan for the future of the nuclear industry in the US. The Department of Energy (DOE), recognizing the need for improvement in our current fuel cycle plan, called for the identification of promising fuel cycle options in order to guide the future of research and development in this area. In response, multiple national laboratory and industry experts collaborated to produce the Nuclear Fuel Cycle Evaluation and Screening Final Report [13]. This report looked at all of the possible nuclear fuel cycle combinations with regard to reactor type, fuel type, which elements (if any) were recovered for reuse within the cycle, and whether enrichment was needed or not. The comprehensive set totaled to 4398 possible fuel cycle options without even considering specific technologies such as reprocessing procedures. This expansive list was then condensed down to 40 Evaluation Groups (EGs) by combining similar fuel cycle options into a single group. The groupings were determined by the following characteristics: once-through or recycle (limited or continuous), reactivity of the irradiation device, neutron spectrum (thermal, intermediate, or fast), feed material (U or Th), if uranium enrichment is needed, and recycled

elements (U, Pu, minor actinides, transuranic elements, Th, and/or fission products). Once grouped, these EGs were to be evaluated based on the criteria set forth by the DOE in order to inform on the relative benefits and challenges of each. These criteria consisted of waste generation, proliferation risk, nuclear material security risk, safety, environmental impact, resource utilization, development and deployment risk, institutional issues, and financial risk and economics. By evaluating all of these criteria for each of the EGs, they were able to be assigned relative benefit and challenge values and plotted in Figure 1.2 [13]. This figure utilized equal weighting of each benefit and challenge criteria.

For reference, EG01 represents the current once-through fuel cycle in use today. This fuel cycle option clearly provides the least challenge as it is already implemented and requires no introduction of additional technologies. However, EG01 does not necessarily provide the greatest benefits. One reason for this is that it does not utilize reprocessing which would

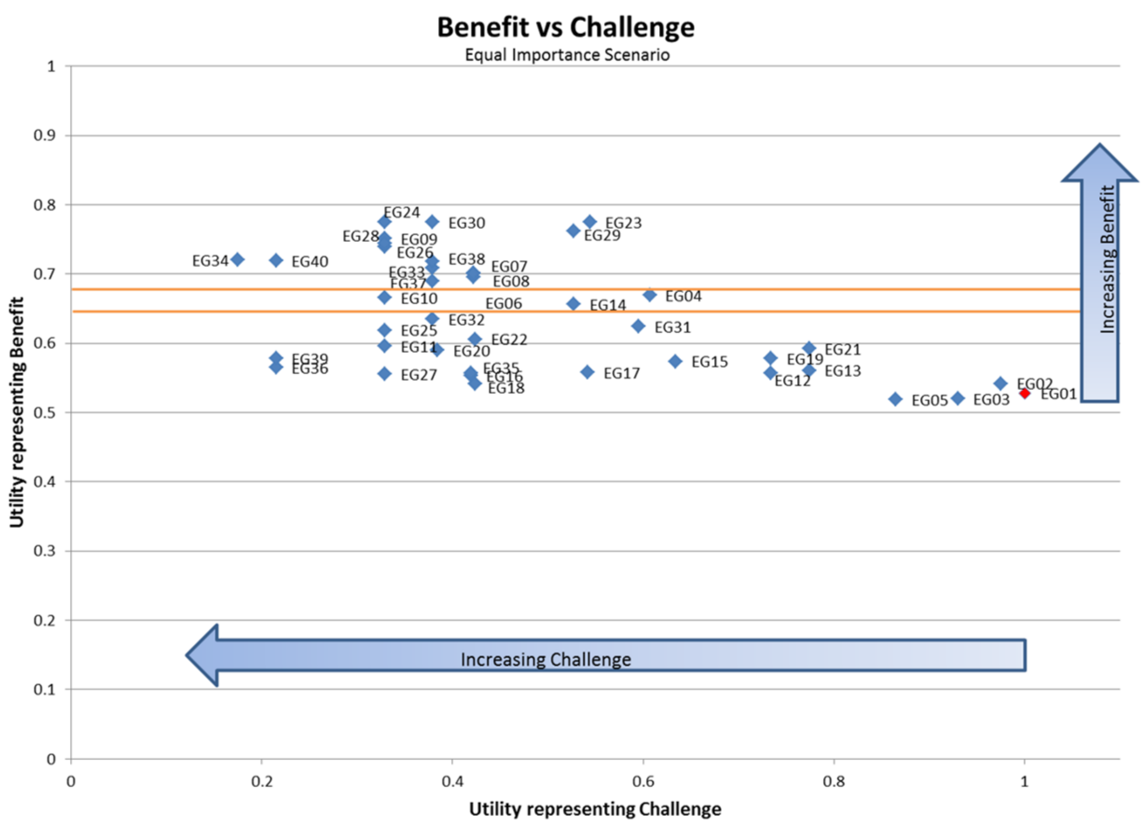

Figure 1.2: Evaluation Groups plotted with their relative benefit and challenge values [13].

decrease the amount of natural resources needed to sustain the cycle. Conversely, the fuel cycle options deemed to be the "most promising" can be found at the top of the figure with the highest benefit and only median amounts of challenges. These most promising EGs are EG23, EG24, EG29, and EG30 and the defining characteristics of each can be found in Table 1.1. Each of these options include the introduction of reprocessing and fast reactors. The only differences between these groups are the types of fuel utilized (U/Pu or U/TRU) and the use of exclusively fast reactors, as opposed to a combination of fast and thermal reactors. The study also concluded that none of these fuel cycle options introduced a significant amount of proliferation risk or nuclear material security risk as long as they were implemented appropriately [13].

Table 1.1: Summary of select fuel cycle options as defined in the Evaluation and Screening Study [13]

	Recycle Strategy	Reactivity	Neutron Spectrum	Feed Material	Enrich. Needed	Recycled Elements
EG01	Once-Through	Critical	Thermal	U	Yes	-
EG23	Continuous Recycle	Critical	Fast	U	No	U/Pu
EG24	Continuous Recycle	Critical	Fast	U	No	U/TRU
EG29	Continuous Recycle	Critical	Thermal & Fast	U	No	U/Pu
EG30	Continuous Recycle	Critical	Thermal & Fast	U	No	U/TRU

1.2 Transition Analysis

The results of the Evaluation and Screening report provided valuable information about which fuel cycle options present the most improvement upon the current fuel cycle based primarily on resource utilization and the amount of nuclear waste generated in a steady-state calculation. Following up on this study is an investigation into the most efficient process for transitioning between the current, once-through fuel cycle to each of these most promising options. This transition analysis looks to minimize the amount of time needed to completely implement

the new fuel cycle and verify that there is sufficient fissile material available to sustain the conversion. In order to evaluate these transition scenarios fully and effectively, fuel cycle simulation codes must be used, as the required mass flow calculations required are much too complex to be efficiently performed by hand. Nuclear fuel cycle simulators simplify this problem by quickly evaluating hundreds of years of transition scenario in a matter of seconds or minutes, depending on the complexity of the problem. There are many simulators currently being used for these transition scenarios: ORION, used by Oak Ridge National Laboratory for this purpose; Cyclus; DYMOND, developed and used by Argonne National Laboratory; and the Verifiable Fuel Cycle Simulation Model (VISION), developed and utilized by Idaho National Laboratory [10, 11, 14, 15]. Multiple simulators have been used to this point in the transition analyses in order to verify that each one produces similar results. These codes are complex and the scenarios modeled are quite involved which makes having multiple independent solutions important to maintain accuracy. Having different groups working on these transition scenarios has also provided important insights into some of the assumptions necessary to create more realistic and valuable simulations.

1.3 Cyclus Based ORIGEN Development

Thus far in the Evaluation and Screening transition analyses, simulations have utilized a recipe-based approach. This method takes the isotopic concentrations from the Fuel Cycle Data Packages calculations or other external calculations and uses them as input into the simulation [16]. In these cases, the fuel cycle simulator itself does not perform any depletion or decay calculations and is primarily used to compute mass flows and material availability. Holding the material isotopics constant in this way allows the simulation to be completed quite fast without sacrificing very much accuracy for these specific cases. However, there are scenarios where the simulator must be able to perform the physics calculations in real time in order to accurately represent the amount of material available at each stage. For example, EG29 involves a combination of fast and thermal reactors and continuous reprocessing of the spent fuel. While the recipe-based method is effective for approximating mass flows with exclusively fast

reactors, it is possible that large discrepancies could be introduced here due to the increased impacts of changing plutonium fissile content within thermal reactors over the course of the simulation. There are many additional scenarios that benefit from the use of depletion calculations within the simulations, such as determining the effects of changing reactor burnup on the rest of the fuel cycle. For these reasons, development of an ORIGEN-based module repository for use with Cyclus, termed CyBORG (**Cy**clus-**B**ased **OR**I**G**EN), has begun [17].

Chapter 2

Fuel Cycle Simulation Methodologies

2.1 Overview of Modern Fuel Cycle Simulators

There are many fuel cycle simulators in existence that have been created by various national laboratories, universities, and utilities, such as the ones being used in the E&S transition studies [10, 11, 14, 15] as well as others like the French Atomic Energy Commission's Comellini Sicard (COSI) [18] and Los Alamos National Laboratory's Nuclear Fuel Cycle Simulator (NFCSim) [19]. Each simulator has its strengths and weaknesses; some simulators sacrifice accuracy in fuel composition in order to increase flexibility and decrease processing power needed to run a simulation, while others sacrifice processing time in order to track more nuclides for greater physical fidelity. As well as varying in how accurately the materials within the fuel cycle are tracked, simulators also vary in the types of facilities they model and how the mass flow calculations between these facilities are performed. The source code for most fuel cycle simulators is proprietary. In the past, each time a different organization would like to simulate their facilities, they must either get permission to use an existing code which has been optimized for a different set of facilities, or they must create their own code. This results in multiple efforts to create very similar simulators rather than a combined effort to improve those which already exist.

While existing fuel cycle simulators are enhancing their capabilities, there is still a large overlap in the efforts of the developers and no single code addresses all of the issues in terms of both availability and functionality. For example, ORION utilizes an intuitive graphical user interface to simulate discrete fuel cycle facilities and the mass flows between them. ORION also provides the capability to utilize material recipes or perform on-the-fly depletion calculations while tracking over 2000 isotopes or lumped material isotopics of 5 categories: uranium-235, uranium-238, lumped uranium, lumped plutonium, and fission products/minor actinides [20]. However, due to the proprietary nature of the code, Oak Ridge National Laboratory is the only domestic entity able to use the tool at this time. COSI is a flexible simulator that tracks individual facilities and can include calculations such as cycle-by-cycle burnup and depletion along with economic analysis [18]. Similar to ORION, COSI is developed abroad which makes gaining access difficult.

Domestically-developed simulators are slightly more accessible for use but similarly restricted for development purposes. VISION is one such code which focuses on computational speed by utilizing a fleet-based approach for simulating fuel cycle facilities and relying solely on recipes for neutronics evaluations [15]. VISION is a system dynamics-based code based in PowerSim which focuses on the interactions between facilities and how it effects the system [21]. DYMOND was the precursor code for VISION and tracks facilities and materials with a similar system dynamics approach. It is primarily concerned with determining reactor deployment profiles and the associated material needs for a given energy demand growth rate [14]. NFCSim tracks discrete facilities and material objects with the ability to perform decay calculations [22]. Further analysis of fuel cycle simulators can be found in [22] where multiple tools are compared with regards to specific functionalities. These functionalities include the ability to discretely track materials and facilities, perform in-depth uncertainty analysis, optimize multiple objectives within a simulation, and be created with open and accessible software. The findings here further reinforce the idea that each of these simulators have their own strengths and intended applications, but this functionality is spread across many different tools which are each difficult to obtain. In addition to being unavailable for use, each of these codes are developed by specific entities and provide no easy means for additional developers to

contribute. The Cyclus fuel cycle simulator was created in order to address these situations [10].

2.2 The Cyclus Fuel Cycle Simulator

Cyclus is an open source simulator which encourages outside developers to contribute, which is one reason that it was chosen as the framework for the CyBORG repository. Additionally, Cyclus is an agent-based simulator which allows the user to choose which facilities to incorporate into a simulation based on their specific needs. The agents included in the simulation are then discretely tracked and managed, as opposed to being treated as a system. These agents can be representations of facilities, institutions, or regions in order to fit within the Cyclus structure shown in Figure 2.1. Facilities are the heart of the simulation which trade and manipulate the resources, such as an enrichment facility or a fuel fabrication plant. Institutions are then the managing entities for the facilities, specifying when to start up and shut down, as well as implementing material preferences on the facilities which ensure that the highest priority

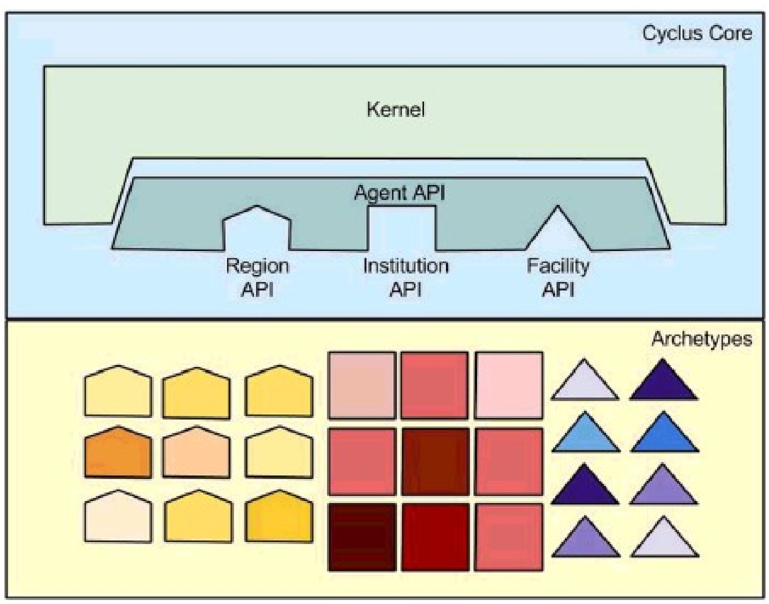

Figure 2.1: Representation of the Cyclus classes which encompass the agent-based architecture. Here, any combination of the optional regions, institutions, or facilities can be utilized within a simulation [10].

facilities receive their material first. The final agent type is the region, which acts as a supervisor for the institutions. Much like an institution, regions are able to specify material bidding preferences between the institutions it oversees.

Cyclus treats each of these agents as a black box, taking the appropriate input and output information but requiring no knowledge of the inner workings. This functionality allows the user to initialize any possible combination of agents in order to create their fuel cycle simulation. This differs from other fuel cycle codes which require that some of the agents be initialized in a specific order, such as requiring that a storage buffer follow a reactor before a reprocessing plant can be added. While these constraints might help to ensure the realism of a simulation in some cases, it requires the simulator to have previous knowledge of the archetypes available. By removing this previous understanding of what each agent does, Cyclus makes the integration of new agents by external developers easier. The external developers are not required to make any changes to the Cyclus core functionality in order to introduce their agent to a simulation and the Cyclus core developers do not have to make any changes in order for Cyclus to use the new agent.

Each agent within a Cyclus simulation passes their material specifications to the Dynamic Resource Exchange (DRE) which handles the mass flow calculations [23]. By accepting specifications from any facility that has been developed for use with Cyclus, the DRE provides a large amount of flexibility not seen in other simulators. The DRE relies on market-driven calculations to determine the mass flows between facilities. This means that the DRE first gathers information about which facilities need material, the isotopic recipe of that material, and how much is needed in the form of a "buy" bid. This bid can contain multiple material specifications with associated preference rankings. Following this information, the DRE determines which facilities have material to offer, the isotopic recipe of that material, and how much is available conveyed with a "sell" bid. With these sets of information, as well as potential priority rankings of facilities implemented by the institutions or regions managing the simulation, the DRE is able to optimize the material flow between facilities. By contributing to Cyclus, this work was able to take advantage of the existing simulator features like the DRE and the existing fundamental archetypes found in the **Cyc**lus **A**dditional **Mo**dule **Re**pository

(Cycamore) and focus on the integration of ORIGEN depletion and decay calculations for accurate depletion calculations.

2.3 ORIGEN Integration

ORIGEN was chosen as the driving force for the physics calculations included in CyBORG because it is an established and considerably validated depletion and decay analysis tool which tracks more than 2000 nuclides and has recently been restructured to be more accessible for incorporation into other codes [24, 25]. ORIGEN's depletion calculations utilize pre-generated cross section libraries which have been created using transport codes such as the TRITON radiation transport module. With ORIGEN's Automatic Rapid Processing (ARP) interpolation module, problem-specific libraries can be created. This process takes the TRITON-generated reactor data libraries and interpolates between them for fixed parameters such as enrichment and moderator density. From the new, problem-specific cross section library, the depletion and decay calculations are performed by solving a series of Bateman Equations (Equation 2.1) which describe the rate of change for each nuclide in terms of the production rate less the removal rate. By solving this system of first-order differential equations at a user-specified time step, the new material isotopics are calculated.

$$\frac{dN_i}{dt} = \sum_{j=1}^{m} l_{ij}\lambda_j N_j + \overline{\Phi}\sum_{k=1}^{m} f_{ik}\sigma_k N_k - (\lambda_i + \overline{\Phi}\sigma_i + r_i)N_i + F_i\,,\ (i = 1, ..., m) \tag{2.1}$$

where

- N_i = atom density of nuclide i
- λ_i = radioactive disintegration constant of nuclide i
- σ_i = spectrum-averaged neutron absorption cross section of nuclide i
- $\overline{\Phi}$ = space- and energy-averaged neutron flux
- l_{ij} = branching fractions of radioactive disintegrations from other nuclides j

- f_{ik} = branching fractions for neutron absorption by other nuclides k that lead to the formation of species i
- r_i = continuous removal rate of nuclide i from the system
- F_i = continuous feed rate of nuclide i.

The ARP interpolation methods are incredibly useful for performing depletion calculations quickly and accurately without needing to call the time-consuming radiation transport code each time; however, the architecture of ARP makes introducing new reactor libraries a difficult process. Historically, the parameters available within the libraries used in ARP have been described in a file named `arpdata.txt`. In order to implement new reactor libraries, the user would be required to add the necessary information to this file before the library could be used. In addition to being difficult to introduce new reactor types, ARP only interpolates over fixed parameters. While the parameters chosen are valuable for the existing data libraries (LWR and MOX), this method limits the flexibility of the interpolation system as new parameters may be needed with the introduction of a new reactor type.

Before integration into Cyclus was possible, changes to ORIGEN architecture and methodologies needed to be completed. This work, completed by another student on this project [26], focused on creating an alternative to the ORIGEN ARP interpolation methods in order to increase their flexibility. In order to address the limitations in the current ARP methodology, a new approach which utilizes generalized interpolation parameters has been introduced. This approach utilizes the capability to interpolate over any parameter in any reactor data library which is properly initialized. By generalizing the techniques, any parameter included in the reactor data library in question can be used as a basis for interpolation. In addition to increasing the flexibility of the interpolation parameters, a new approach to initializing the reactor data libraries has been implemented. In order to avoid the need for constant updating of an `arpdata.txt`-type file to keep up with new reactor types, a method of creating self-describing libraries has been applied. By removing the library's dependencies on descriptor files, the interpolation and depletion engine becomes portable as well. This makes it available

for use outside of the SCALE package and usable by external codes. All of these developments primed ORIGEN for a seamless integration into Cyclus.

Chapter 3

Evaluation and Screening Transition Analyses

Following the E&S Final Report which quantitatively ranked potential fuel cycle options based on criteria such as resource utilization and nuclear waste generation, a study began which sought to evaluate potential pathways for transitioning to the most promising of these options. Here, Cyclus and ORION are both used to look at one approach for transitioning from the current once-through fuel cycle to a cycle which includes continuous reprocessing of spent fuel as well as fast critical reactors (EG23). Before performing these evaluations, however, the Cyclus fuel cycle simulator required preliminary development of a `Storage` facility. In addition, integration tests were performed which verified Cyclus' ability to perform accurate mass flow calculations for simple cases before launching into the full scale transition evaluations.

3.1 Cyclus Development

At the beginning of this project there were a number of standard fuel cycle facilities available within Cycamore [27]. These included a material source, an enrichment facility, a recipe-based reactor, and a material sink. In order to simulate more complex fuel cycles, Cycamore required additional agents such as fuel fabrication, reprocessing, and storage facilities. As part of this

project, the University of Tennessee was tasked with developing and testing a storage facility model. This mutually beneficial assignment allowed the experienced Cyclus developers to focus on the other facilities and upgrades to the Cyclus core while providing an opportunity for the new developers to become familiar with the Cyclus ecosystem and best practice while taking on the relatively less complicated storage facility. Development of a versatile storage facility for Cyclus began with the framework from a partially developed existing module known as `CommodConverter` [28]. This facility had the ability to take in material, hold it for a user specified amount of time, then convert the material's isotopic makeup and release it. Each of these steps, with the exception of the conversion process, fulfilled the requirements for the Storage facility. The `CommodConverter` archetype had very few tests in place, so in order to validate that this facility operated as expected and to become more familiar with the code, unit tests were developed for `CommodConverter`. These tests, written with Google Test, included checks for how the facility functioned under various feasible circumstances [29]. This involved attempting to add too much material to the facility to be sure it would only accept the correct amount, as well as adding multiple small batches of material to verify that the facility held the material for an appropriate amount of time. Descriptions of each of the tests added in this work are shown in Table 3.1.

Once the `CommodConverter` archetype was fully tested and shown to operate as expected, the conversion to a `Storage` facility began. This included renaming the class, removing the conversion capabilities, and adding in updates to the original `CommodConverter` code in order to conform to more recent Cyclus style guides. To do this, the storage facility's interface with the Dynamic Resource Exchange needed to be updated. There are two major interfacing strategies within Cyclus for use with the DRE. In the first, complex "bidding" and "trading" functions can be added into an archetype to provide precise material preference and availability information. Alternatively, new functionality has been added to Cyclus which allows the archetype to simply define a "buy" and "sell" policy for the facility. The buy policy specifies the type of material that the facility wishes to take in and, conversely, the sell policy specifies the type of material that the facility has to offer. Since the `Storage` facility is a relatively simple archetype, these buy and sell policies were ideal for handling its material trades. In addition to updating the DRE

Table 3.1: Unit tests developed within this project for the `Storage` facility for use within Cyclus

Test Name	Purpose
NoProcessTime	Ensure the material is passed through in a single time step when the `residence_time` of the facility is set to 0 time steps.
NoConvert	Verify that the output material recipe is the same as the input material recipe when no radioactive decay is included.
MultipleSmallBatches	Test the facility's ability to handle processing numerous material objects coming in at different times.
ChangeCapacity	Verify that changes to the `throughput` of the facility are reflected in its operation.
TwoBatchSameTime	Show that facility can handle multiple batches of material entering at the same time.
ChangeProcessTime	Verify that changes to the `residence_time` are reflected in the facility's operation.
DifferentRecipe	Ensure that the facility will accept any commodity recipe that is offered to it and not change its composition.

interface, the methods for tracking material within the facility were updated. In order to both hold the material in the facility for a user specified `residence_time` and restrict the mass of material processed in a time step with the `throughput` variable, the internal material handling was split into two separate material buffers. The first was used to hold the material until the `residence_time` had passed. From here, the material was passed to a second buffer where it was to be made available for selling in batches less than or equal to the `throughput` amount. Finally, a user-defined state variable was added which determined whether the material inside the facility is handled in discrete or continuous batches. This means that if the storage facility is operating in discrete mode and accepting spent fuel in assembly-sized batches, then the output from the facility will also be in assembly-sized batches, whereas if the facility is in continuous mode the assembly-sized batches would be lumped into a single `throughput`-sized batch. Currently, the Cyclus resource exchange defaults to continuous mode, however this functionality was added in for possible future uses.

3.2 Nuclear Fuel Cycle Integration Tests

With many different fuel cycle simulators contributing to the Evaluation and Screening Study, it became important to show that each of them was able to produce similar results for simple test cases before beginning the complex simulations required by the Study. Two straightforward test cases were created at Brookhaven National Laboratory as a means for verifying this as well as testing specific functionality that would be needed for future, more complicated simulations. These included the Reactor Deployment and Growth tests, specifications for which can be found in Appendix A. Each of these tests was created with Cyclus.

3.2.1 Reactor Deployment Test

The first test aimed to test the simulator's ability to deploy a reactor in a specified year, maintain steady state operations with an infinite source of fuel, and accurately accumulate spent fuel in cooling storage until it is ready to be moved to dry storage. This was to be a simple 100-year simulation in which a reactor started up in year 15 and operated through to the end. The spent fuel was to spend 4 years in cooling storage before being transferred to dry storage for the remainder of the simulation. This reactor deployment test case was built in Cyclus by utilizing the Cycamore archetype library elements of `Source`, `Reactor`, `Storage`, `DeployInst`, and `NullRegion`. The material flow began at the `Source` facility with an endless supply of fresh fuel as its output commodity. The fuel was then requested by the `Reactor` facility which produced the power specified. After being transmuted in the reactor, the material was sent to the first `Storage` facility for cooling for 4 years and subsequently moved into another `Storage` facility in order to separately track cooling and cooled spent fuel. The `DeployInst` was utilized in order to deploy the reactor at the appropriate time step, along with the `NullRegion` which required no inputs and managed the mass flows scenario. This simulation was set to begin in January 2000 and continue for 100 years. The Cyclus default time step of one month was used here as well. More detail on the initialization of this simulation can be found in Table A.1.

In the solution to this unit test, 0.9 GWe-yr is generated from year 15 to year 99, as shown in Figure 3.1. No electricity is generated from year 0 to year 14 before the reactor is started up.

Initial core loading is 90 tU at start of year 15 because the entire core is being loaded at the beginning of the reactor's life. Subsequent annual loadings are 20 tU/y through to the end of the simulation, shown in Figure 3.2. The fluctuations seen in fuel loading are due to Cyclus' monthly time step, with fuel only being loaded at the beginning of each year and remaining in the reactor until the end of the year. These annual fuel discharges are illustrated in Figure 3.3 by the amount of spent fuel accumulating in cooling storage. Since the spent fuel remains here for 4 years, the mass builds up to 80 tonnes and fluctuates between 80 and 100 for the remaining years. This fluctuation is once again due to the Cyclus time step, where spent fuel is sent to cooling storage one time step before the oldest fuel is sent to cooled storage. This accumulation of spent fuel in cooled storage is shown in Figure 3.4.

The electricity profile presented here matches the expected steady state production of 0.9 GWe-yr. In addition, the annualized mass flow values produced in the simulation are consistent with the hand-calculated values for this simple case.

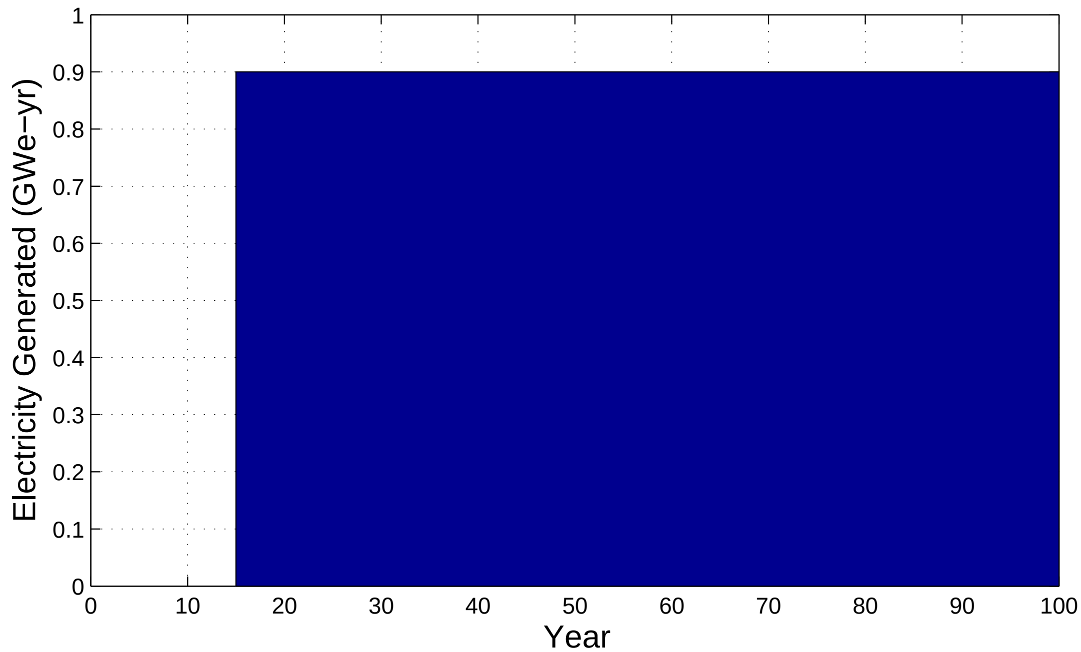

Figure 3.1: Electricity generated (GWe-y) by a single Cyclus `Reactor` in the Reactor Deployment Test

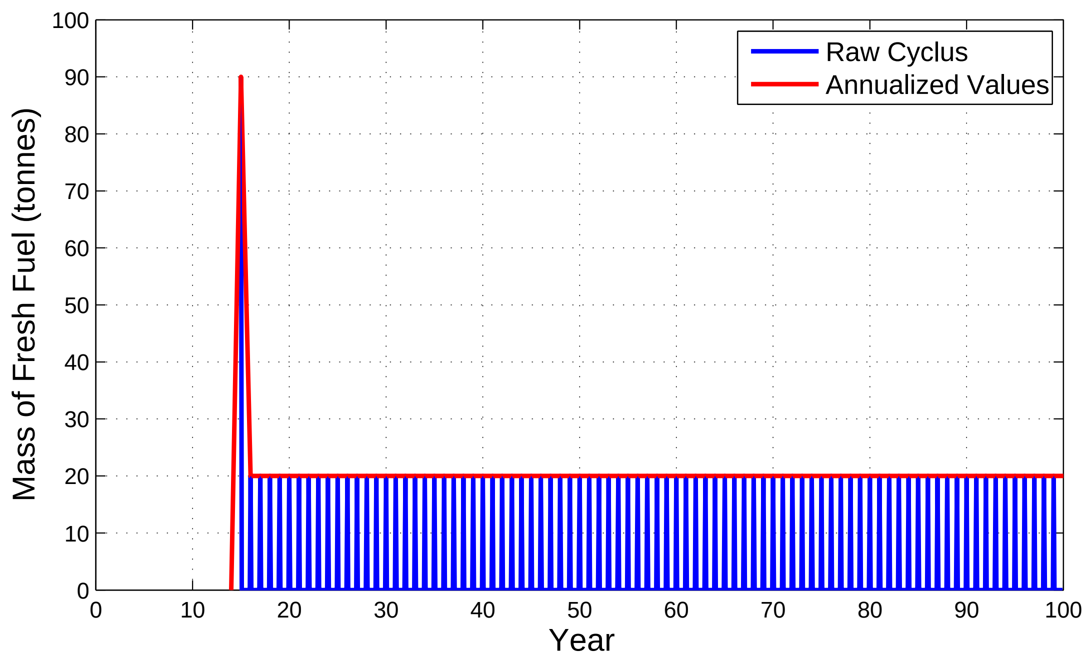

Figure 3.2: Mass of fresh fuel (tonnes) loaded into the Cyclus `Reactor` in the Reactor Deployment Test

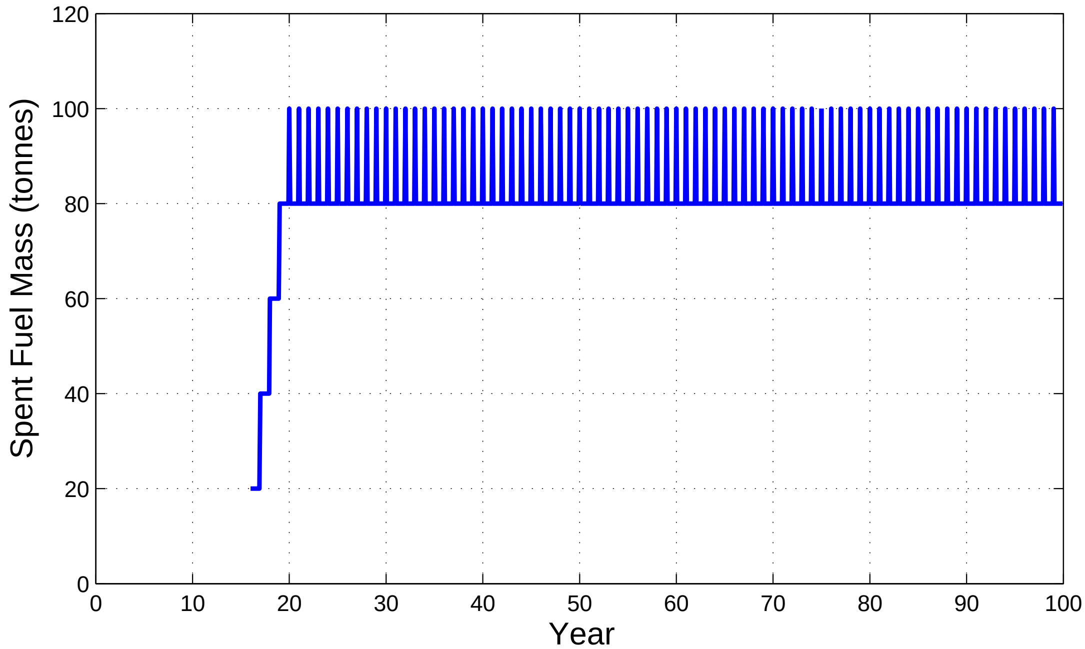

Figure 3.3: Amount of spent fuel (tonnes) in cooling storage in the Reactor Deployment Test

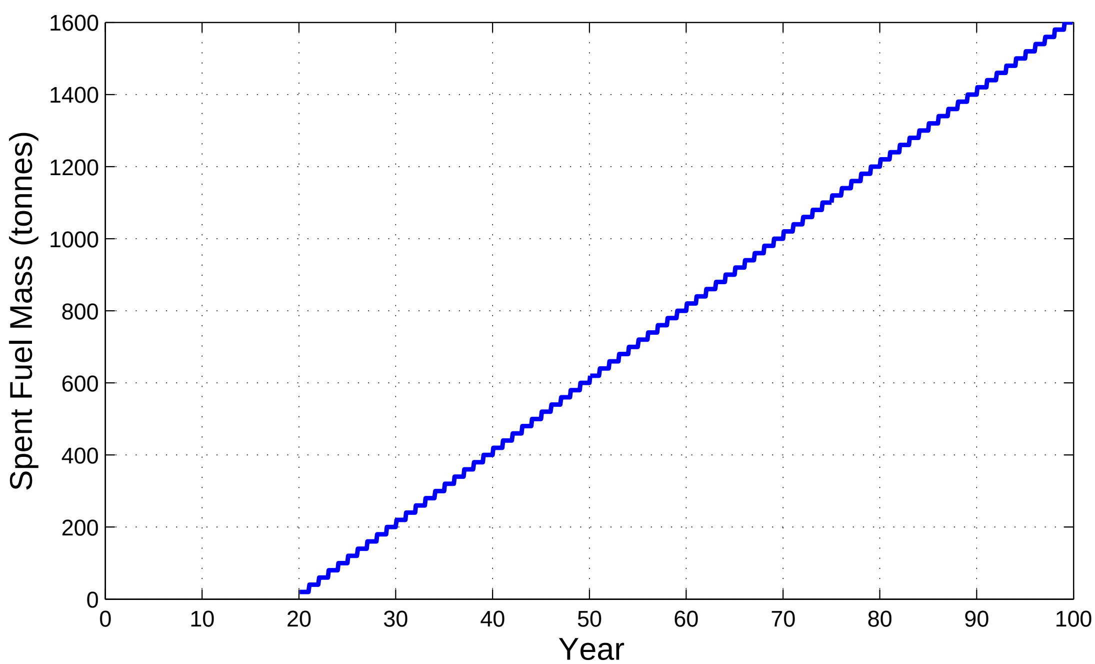

Figure 3.4: Mass of spent fuel (tonnes) accumulating in cooled storage in the Reactor Deployment Test

3.2.2 Growth Test

The second unit test was created to evaluate the ability to deploy reactors according to an increasing energy demand. This case was to last for 200 years with the electricity generated increasing by 1% annually. The material flow in this Cyclus simulation began at a `Source` facility with an endless supply of fresh fuel as its output commodity. The fuel was then requested by a Cycamore `Reactor` facility which produced the power specified. After being transmuted in the reactor, the material was sent to a Storage facility for cooling for 4 years and subsequently moved into another `Storage` facility in order to separately track cooling and cooled spent fuel. A Cycamore `DeployInst` was utilized in order to deploy reactors to meet the growing energy demand, along with a `NullRegion` which managed the mass flows in the scenario. The simulation was also tested using a `GrowthRegion`, however, at this time this agent was still in development so the results shown below utilize a `DeployInst` along with a hand-calculated deployment schedule. This simulation was set to begin in January 2000 and continue for 200 years. More detail on the initialization of this simulation can be found in Table A.2.

In the solution to this unit test, 90 GWe-yr is generated in the first year of the simulation. The energy demand grows by 1% per year until year 199. Figure 3.5 shows this growth in power generation based upon the demand. The deployment schedule used as input to the `DeployInst` in order to achieve this growth in electricity generation is illustrated in 3.6. The deployment schedule given to Cyclus only implemented new reactors at each year time step in order to simplify the simulation (rather than monthly as is the default for Cyclus), causing the supply of power to, at times, be a portion of a year behind or ahead of the demand as shown in Figure 3.7. These fluctuations are acceptably low (<0.5% difference) and allow the constraints of the test to be satisfied. The amount of spent fuel accumulating in cooling storage is illustrated in Figure 3.8 and, like the Reactor Deployment spent fuel results, the effects of the month time step in Cyclus can be seen. Similarly, the annualized values presented here are consistent with those previous hand-calculated values.

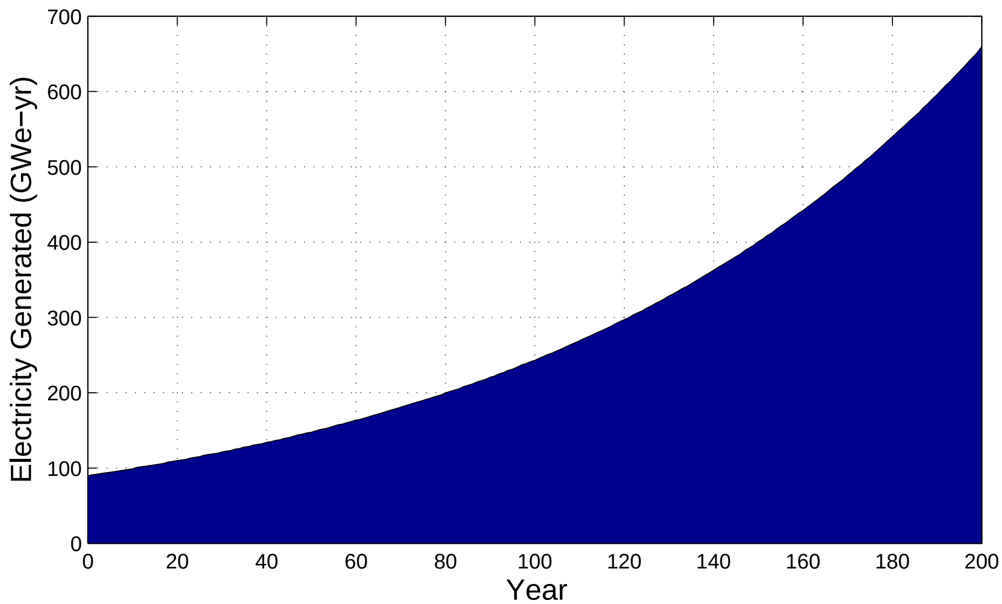

Figure 3.5: Electricity Generated (GWe-y) in Cyclus for the Growth Unit Test

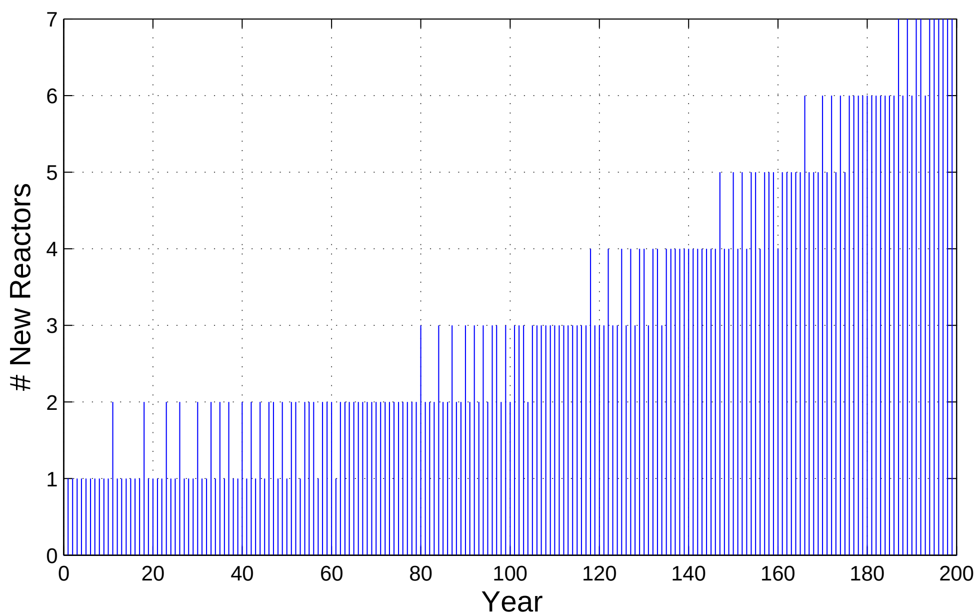

Figure 3.6: Deployment Schedule utilized in Cyclus for the Growth Unit Test

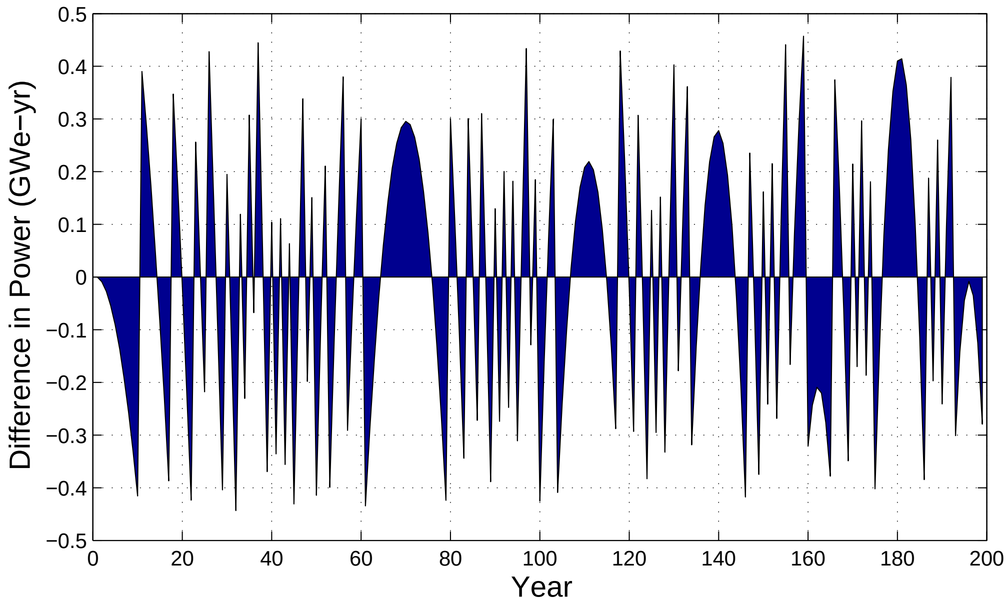

Figure 3.7: Difference between electricity demand and actual electricity generated in Cyclus for the Growth Unit Test

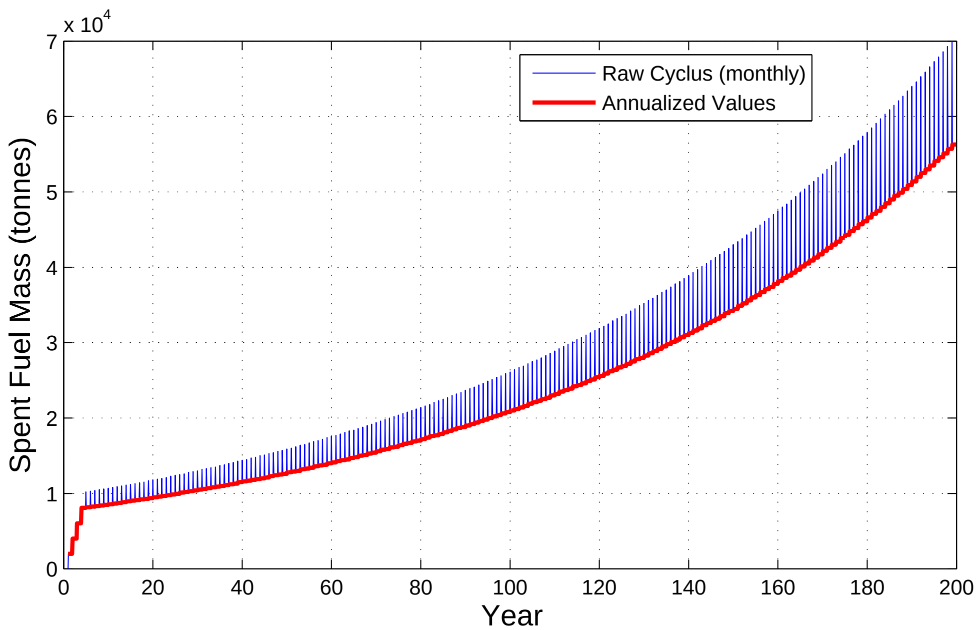

Figure 3.8: Mass of spent fuel (tonnes) accumulating in cooling storage in the Growth Unit Test

3.3 Steady-State Evaluation Group Simulations

Before beginning transition simulations for the E&S Study, steady-state calculations were performed for both of the EGs to be considered. The first of these was EG01 which represents the current once through fuel cycle implemented in the US. This fuel cycle utilizes enriched UO_2 fuel within thermal critical reactors. The second EG to be simulated was EG23 which includes continuous recycle of U/Pu fuel in fast critical reactors. These steady state simulations were created in order to check the simulated mass flow calculations against the hand calculations included in the Fuel Cycle Data Packages (FCDP) that were used to quantify the mass flow-depended criteria in the E&S Study [16]. The FCDP mass flows througout these EGs are shown in Figure 3.9. For this reason, the input parameters used here are as close as possible to the assumptions made within the FCDP calculations. Complete information on the input specifications for each of these simulations can be found in Appendix A. In some cases, such as reactor fuel capacity, the input parameters restrict the ability to input the exact value used in the FCDP, therefore creating small discrepancies in the results.

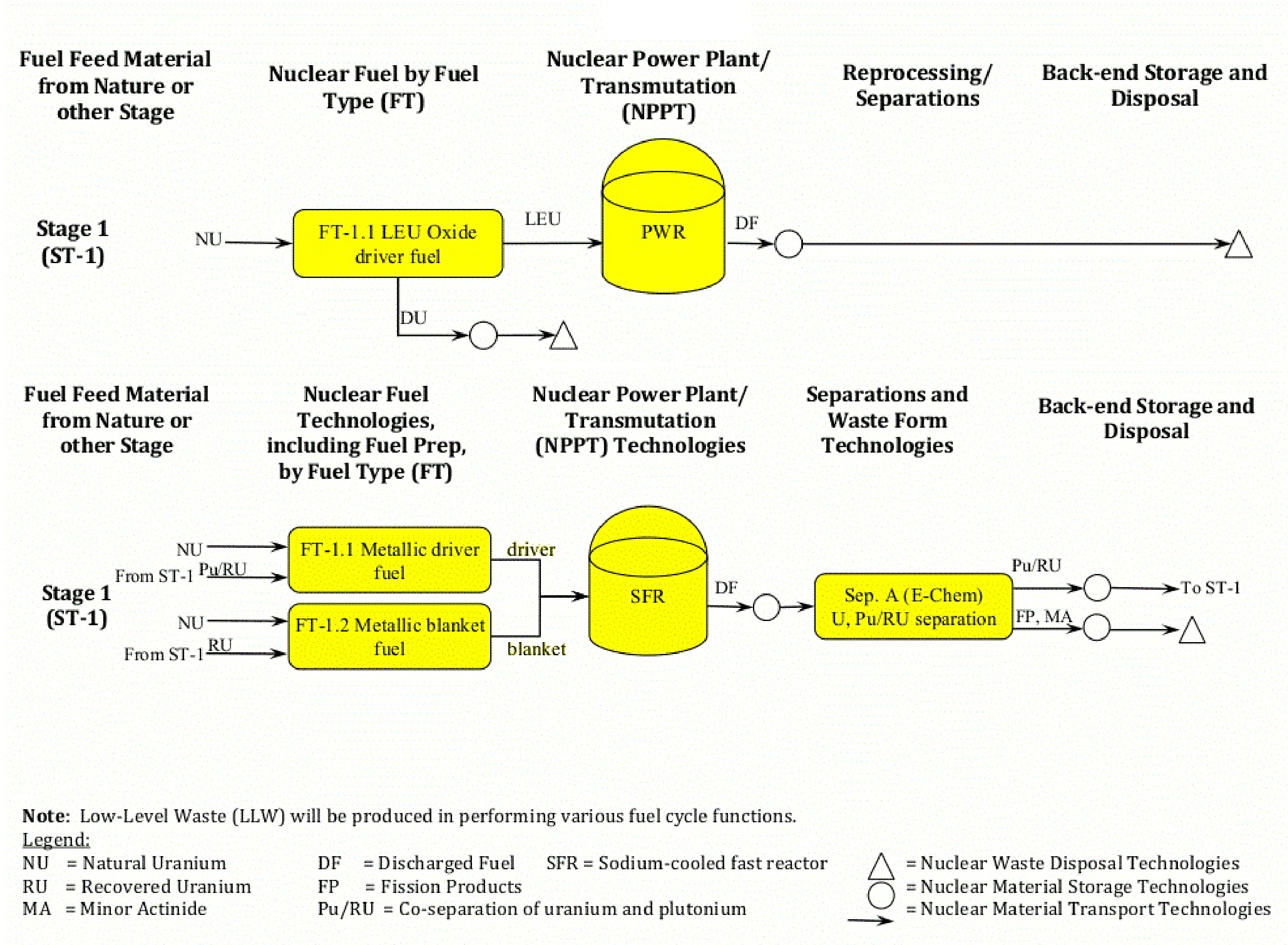

Figure 3.9: Mass flow diagrams for EG01 (top) and EG23 (bottom) as used in the Fuel Cycle Data Package calculations [16]

3.3.1 EG01 - Once-Through Fuel Cycle

The steady state EG01 model was created in Cyclus and the results were compared with FCDP data as well as simulation results from ORION. This model utilized the following Cycamore agents: `Source`, `Enrichment`, `Reactor`, `Storage`, `DeployInst`, and `NullRegion`. The specific input parameters for each of these agents can be found in Table A.3. In summary, these agents were used to simulate a once-through fuel cycle which operates for 100 years beginning in 2015. The mass flow through this simulation is illustrated in Figure 3.10. The average annual feed or product of nuclear materials (in tonnes) at various points in the fuel cycle are shown in Table 3.2. Clearly, the results found with Cyclus for the EG01 steady-state case are in agreement with the FCDP values, as well as those found with ORION.

Table 3.2: Steady state amounts of nuclear material, in tonnes, at various stages in the EG01 cycle as calculated by ORION and Cyclus.

		ORION		**Cyclus**	
	FCDP Analytical Solution	**Solution**	**Difference from FCDP (%)**	**Solution**	**Difference from FCDP (%)**
Natural Uranium	18,862.80	19,043.00	0.96	18,749.94	0.60
Enrichment DU	16,666.90	16,822.20	0.93	16,558.10	0.65
Depleted Fuel	2,191.50	2,216.37	1.13	2,191.84	0.02
Processing Losses	4.40	4.44	0.91	*	*

* Cyclus does not currently track fabrication losses

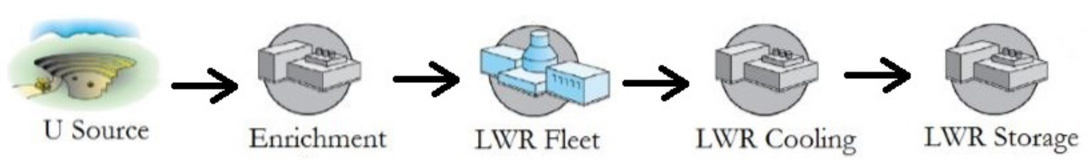

Figure 3.10: Illustration of mass flows through the EG01 steady state simulation in Cyclus

3.3.2 EG23 - Fast Reactors with Full Recycle

The EG23 steady state model was also created in Cyclus and compared to FCDP data and ORION output. An illustration of the mass flows used in this scenario can be found in Figure 3.11. In this simulation of continuous recycle with fast reactors, an external fissile source was added in order to jump start the beginning of simulation. This external source of plutonium had just enough inventory for the first few years until the cycle could become self-sustaining. In addition, the fast reactor in this case was divided into 2 parts: the plutonium breeding blanket portion, and the driver which produces the bulk of the energy. This separation was made in order to accurately account for the mass flows while maintaining the different residence times for each section of the reactor. Specifics on the input parameters used for the Cyclus EG23 simulation can be found in Table A.4. The power distribution between the SFR driver and blanket came directly from the input specifications for the EG23 steady state simulation used in ORION. The annual average amounts of material, in tonnes, found at various stages in this fuel cycle are shown in Table 3.3. Included here are the FCDP analytical values for the EG23 case compared with those found with ORION and Cyclus. Slight discrepancies, such as the Cyclus natural U and MA values, result from differing reactor batch handling techniques and time step lengths, or rounding errors in the material recipe specifications.

3.4 Transition Analysis

Once the steady state simulations for EG01 and EG23 were completed, the analysis of the transition between the two began. For this work, these transition scenarios were simulated in both Cyclus and in ORION. After some consideration, the constraints listed below were placed upon the transition simulations.

- Restrict the mass of separated plutonium to stay below 100 tonnes.
- Begin reprocessing in the year 2050 and no earlier.
- Introduce fast reactors gradually into the scenario to prevent large start-up of second generation LWRs.

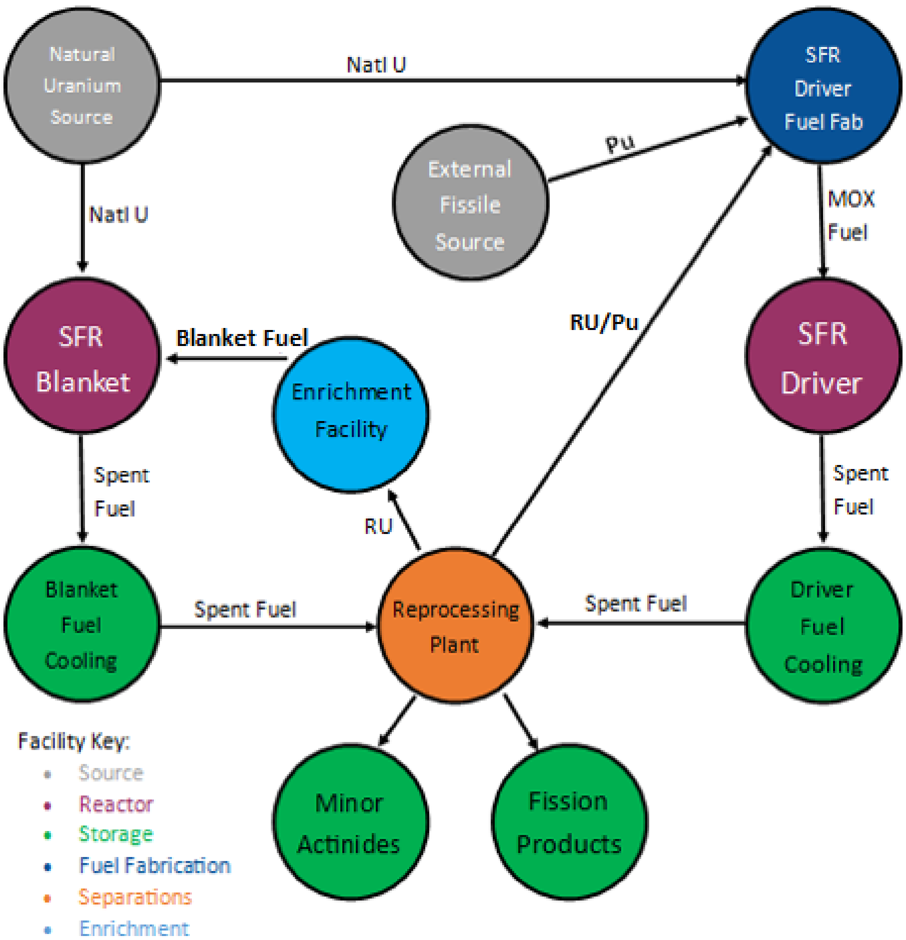

Figure 3.11: Illustration of mass flows through the EG23 steady state simulation in Cyclus

- A 5% surplus/deficit in energy is allowable.
- Once the transition is complete, the breeding ratio should be reduced in new fast reactors.

The first code used to create a transition scenario which followed all of these guidelines was DYMOND. This simulation utilized a just-in-time approach to reprocessing, meaning that the reprocessing capacities at each time step were specified to only produce enough plutonium to supply the reactors in the following year. This maintained the amount of separated plutonium at zero at the end of each year. This solution not only meets the criteria placed on plutonium storage but improves upon it within the scope of the simulation. This simulation also verified that there is sufficient plutonium to sustain the quickly growing fast reactor fleet throughout the transition. Since this simulation fulfilled all of the requirements for a successful transition scenario, initial Cyclus and ORION calculations were based upon the input specifications used in DYMOND shown in Tables A.5 & A.6. This provided a means of verifying that each tool was able to produce very similar results, thus also verifying that the DYMOND simulation utilized reasonable reactor operation assumptions and mass flow calculations. DYMOND's initial reactor deployment schedule, which was used as input to Cyclus and ORION, is illustrated by the amount of power generated by each reactor type in Figure 3.12. The reprocessing capacities

Table 3.3: Steady state amounts of nuclear material, in tonnes, at various stages in the EG23 cycle as calculated by ORION and Cyclus.

	FCDP Analytical Solution	ORION Solution	ORION Difference from FCDP (%)	Cyclus Solution	Cyclus Difference from FCDP (%)
Mass of U charged	1095.20	1096.50	0.12	1096.30	0.10
Mass of Pu charged	162.20	162.40	0.12	162.40	0.12
Recovered U (99%)	987.10	987.70	0.06	987.50	0.04
Recovered Pu (99%)	163.10	164.10	0.61	162.30	0.49
Recovered MA (99%)	1.50	1.50	0.00	1.60	6.67
Recovered FP (99%)	93.10	92.90	0.21	92.80	0.32
Processing Losses	15.10	15.10	0.00	*	*
Natural U feed	110.60	111.30	0.63	116.70	5.52

* Cyclus does not currently track fabrication losses

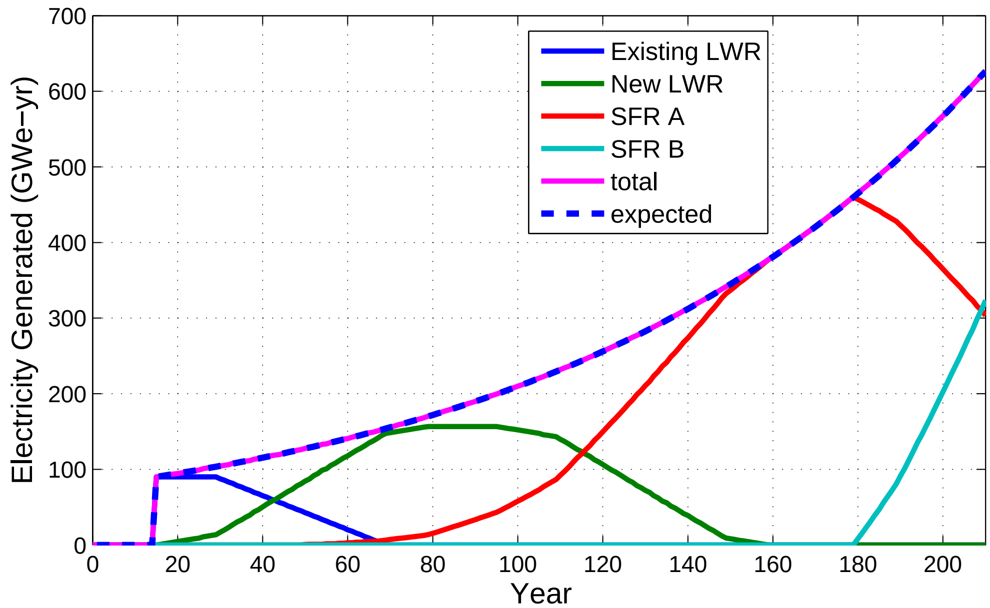

Figure 3.12: Reactor deployment schedule first created in DYMOND and reproduced in ORION and Cyclus for the transition from EG01 to EG23, which includes second generation fast reactors for a lower breeding ratio once the transition is complete.

from DYMOND were also used in one way or another as input to these simulations, however due to a difference in time step length they were adjusted accordingly and produced slightly different results for the amount of plutonium in storage at the end of each year.

3.4.1 Cyclus Transition Analysis

An initial Cyclus transition scenario was created using the mass flow information shown in Figure 3.15 which sought to reproduce the DYMOND electricity generation profile but neglected the plutonium constraint. In this case, the reprocessing capacities were essentially unlimited, causing the reprocessed plutonium to build up in the MOX fuel fabrication plant's incoming material buffer. This scenario produced the expected electricity generation profile with mass flow values being approximately the same as the DYMOND solution with the exception of the reprocessing output. Since the reprocessing capacity was unconstrained, the excess amount of separated Pu being produced was quite large in the first year of operation of the Light Water Reactor Reprocessing plant, but remained relatively low in the following years, as shown in

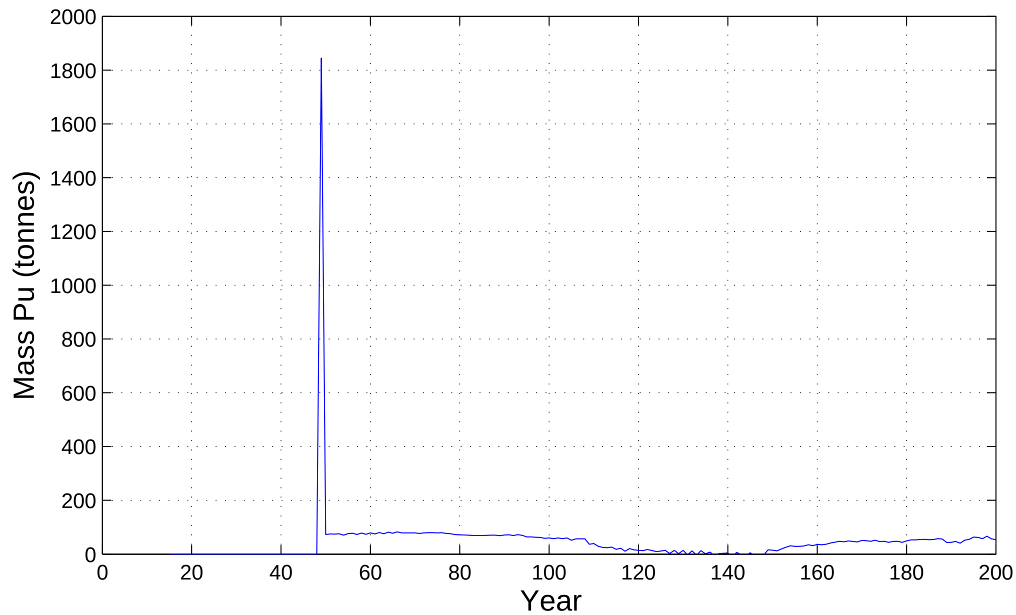

Figure 3.13: Excess amount of separated Pu produced each year in the unconstrained Cyclus transition scenario

Figure 3.13. This jump in excess plutonium in year 49 is a result of the LWR reprocessing facility processing all of the LWR spent fuel that has built up until this time. From this time step forward, the reprocessing facilities are reprocessing the spent fuel as it is created, allowing the values to remain below approximately 75 tonnes/year in the remaining years. Clearly this approach does not address the separated plutonium case and was only used as a first step to ensure that a working transition could be achieved.

In order to recreate the DYMOND reprocessing capacities in Cyclus, a new approach was needed as Cyclus input does not accept time step-specific capacity values. To account for this, incrementally sized LWR and SFR reprocessing facilities were created so that they could be added or removed as needed to represent the fluctuations in reprocessing capacity throughout the simulation. This method maintained the amount of plutonium in storage each year below the 100 tonne limit, as shown in Figure 3.14. The fluctuations in plutonium storage amounts can be attributed to the slight differences in the small reprocessing facility's capacity and the change in plutonium needs. As the goal here was to maintain the amount of Pu in storage

below 100 tonnes in any given year and there are thousands of tonnes of Pu in circulation, these fluctuations are acceptably low in comparison. However, the amount of separated Pu that builds up as a result of these fluctuations could be problematic if they were to become much larger. Clearly, the differences in fuel cycle specification procedures between Cyclus and DYMOND create some difficulties in producing the exact same results, however, Cyclus is still capable of producing a transition scenario which meets the constraints.

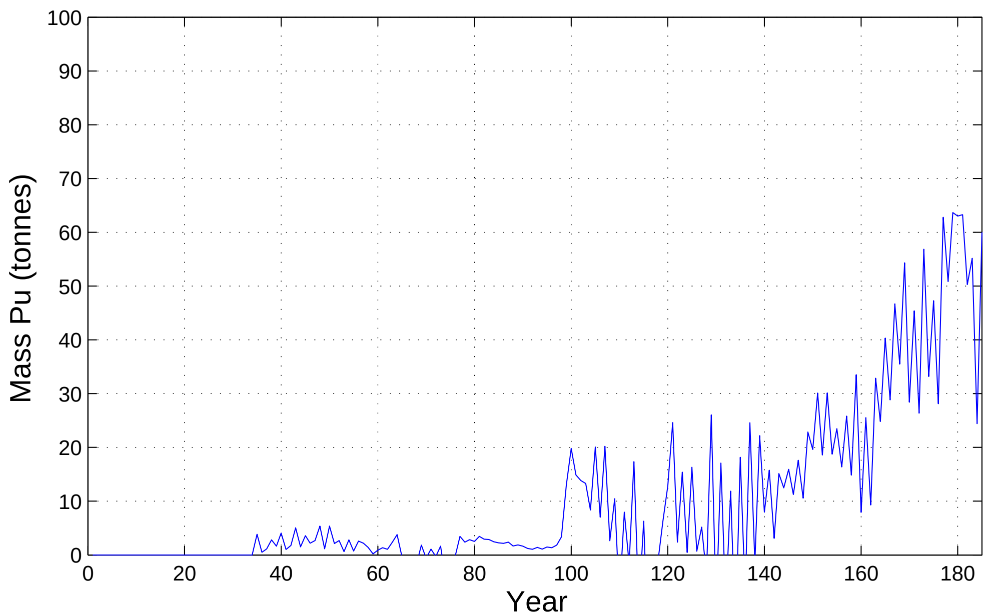

Figure 3.14: Net amount of Pu in storage at the end of each year in the Cyclus transition scenario

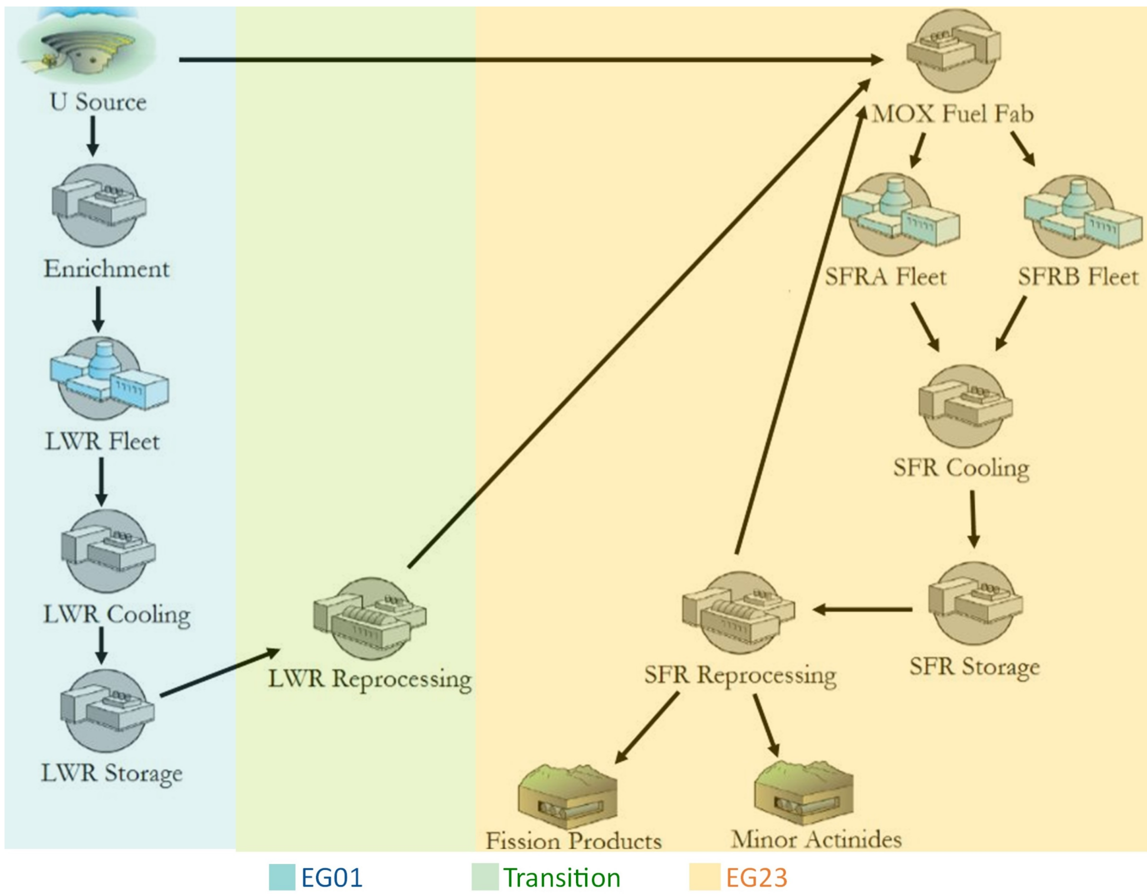

Figure 3.15: Illustration of mass flows through the EG01-EG23 transition scenario in Cyclus

3.4.2 ORION Transition Analysis

The mass flows through the ORION transition simulation can be found in Figure 3.16. Much like the Cyclus transition analysis, ORION was initially provided with the same input values, wherever possible, that were used in the DYMOND transition scenario in order to replicate the simulation. This included directly inputting the DYMOND reactor deployment schedules and reprocessing capacities. However, for ORION, this approach was only the first (Case 1). Once the DYMOND simulation had been replicated, additional simulations were created which more fully utilize ORION's suite of capabilities. ORION contains a Dynamic Reactor Control (DRC) feature which allows the user to specify an energy demand curve as well as guidelines for when to initialize new reactors of each type. In these cases, the DRC was initialized to start up new reactors to meet the 1% growth in energy demand with the new reactor being a fast reactor if sufficient plutonium was available and a LWR if there was not. ORION's DRC was used to create three additional simulations in order to explore the effects of changing the startup rate of fast reactors on the transition time. In the first of these simulations (Case 2), the original constraints on fast reactor startup rate were kept in place in order to recreate the DYMOND transition scenario with the DRC. For Cases 3 and 4, the fast reactor startup rate was restricted less and less while still preventing a second wave of new LWRs. These cases are summarized in Table 3.4.

Case 1 - DYMOND Input

In the first iteration of simulations within ORION, the DYMOND reactor deployment schedules and reprocessing capacities were directly input into ORION. This approach resulted in the

Table 3.4: Transition cases presented with ORION.

Case #	Initial Reactor Deployment Constraint	Uses DRC?
1	1 Reactor per year for first 20 years	No
2	1 Reactor per year for first 20 years	Yes
3	1 Reactor per year for first 10 years	Yes
4	1 Reactor per year for first 5 years	Yes

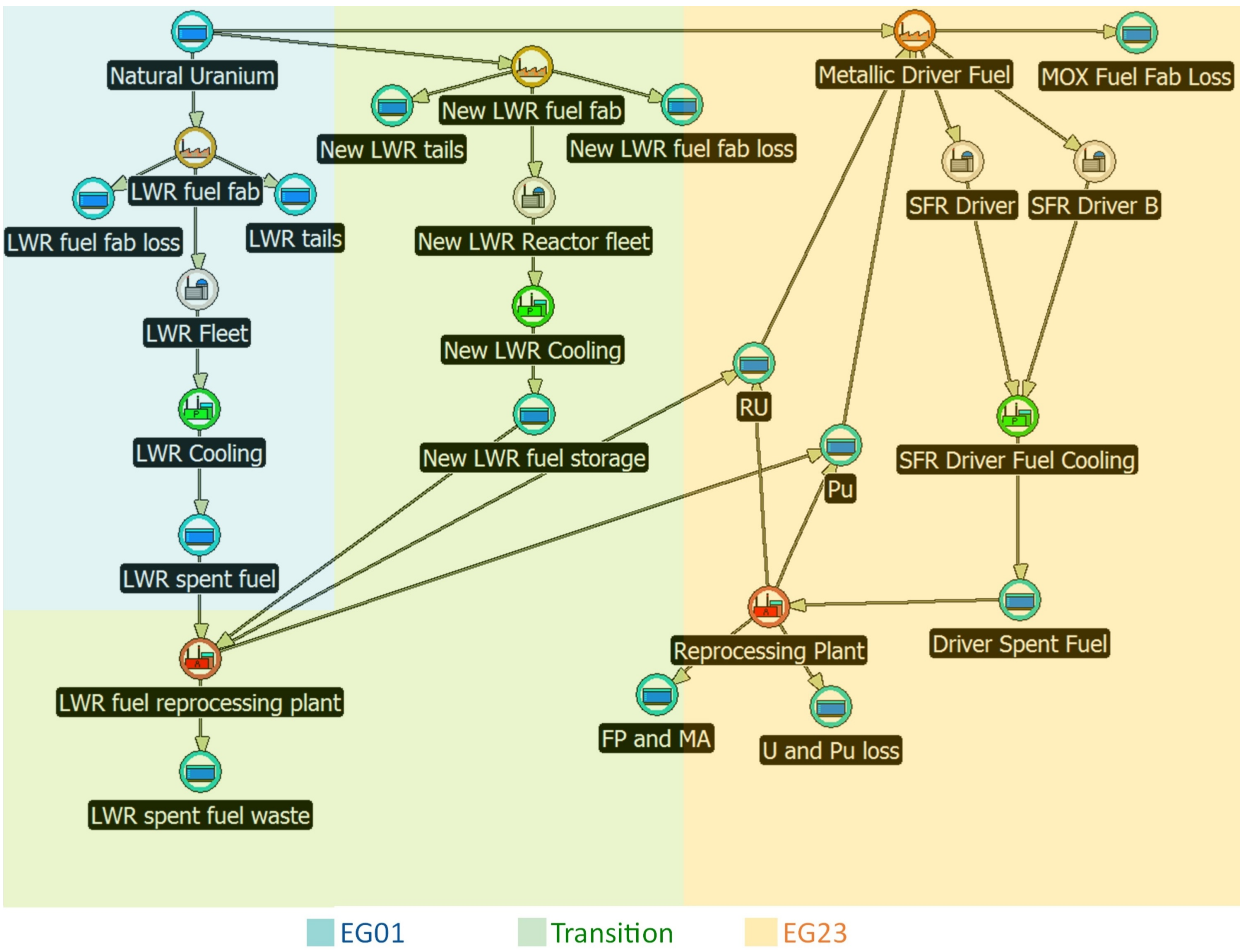

Figure 3.16: Illustration of mass flows through the EG01-EG23 transition scenario in ORION

same electricity generation profile as the DYMOND scenario as seen in Figure 3.17. Direct comparisons of the power generated by each reactor in ORION to those in DYMOND can be found in Appendix B.1. Illustrated here, the electricity generated by each reactor type in ORION was essentially the same as that generated by reactors in DYMOND. While the DYMOND simulation maintained the net amount of separated plutonium in storage at the end of each year at zero, the corresponding ORION simulation maintained this value below approximately 10 tonnes due to the difference in time step size. The net amounts of plutonium in storage at each year in the simulation are shown in Figure 3.18. These fluctuations are acceptably small as they maintain the amount of excess Pu in the simulation well below the 100 tonne limit.

Case 2 - Dynamic Reactor Control with DYMOND profile

After verifying that the original electricity generation profile could be reproduced in ORION with nearly the exact same input parameters as those used in DYMOND, the process began to see if comparable results could be found with the DRC feature in ORION. The startup rate of fast reactors in the first few years was restricted by keeping the SFR energy demand low. Since the DRE deploys reactors based on energy demand and material availability, this effectively controlled the rate of fast reactor start-up. Once again, the reprocessing capacities were restricted as well in order to maintain the amount of separated plutonium in storage under 100 tonnes. In order to allow the DRC to fully function, the amount of plutonium in storage was not held as low as in Case 1 and so the goal here shifted to ensuring that the buildup of excess plutonium did not exceed 100 tonnes throughout the simulation. Figure 3.19 shows the electricity generation profile generated by this simulation, with a detailed look at the comparison of each reactor's contributions in ORION and DYMOND in Appendix B.2. The total amount of excess plutonium in storage at each time step is shown in Figure 3.20, with values peaking around 70 tonnes. This value is well below the 100 tonne limit and, considering the thousands of tonnes of plutonium passing through the simulation at any time, is still quite small (0.6% excess).

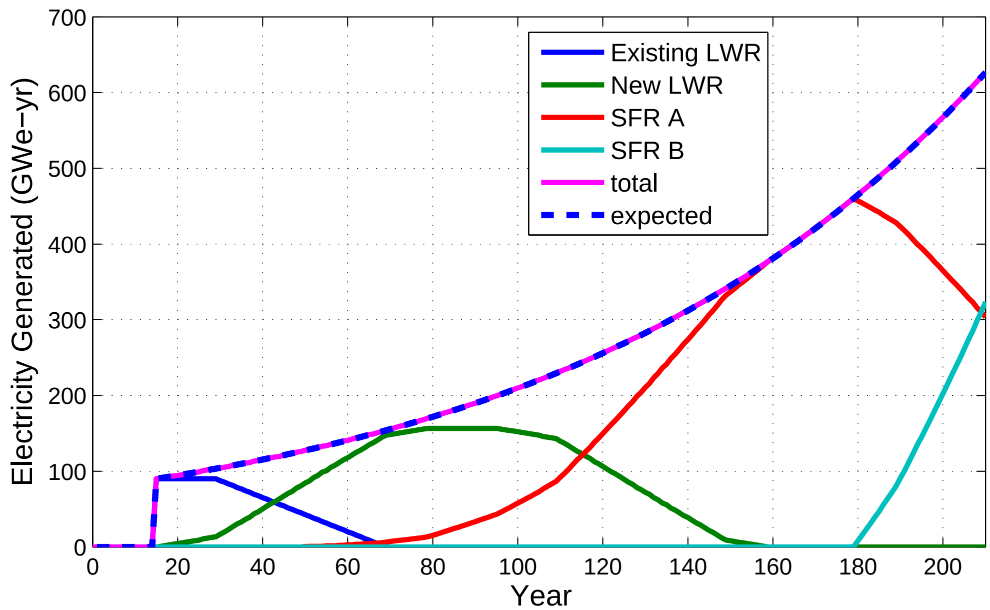

Figure 3.17: Electricity generation profile created by ORION using the DYMOND simulation input parameters for Case 1

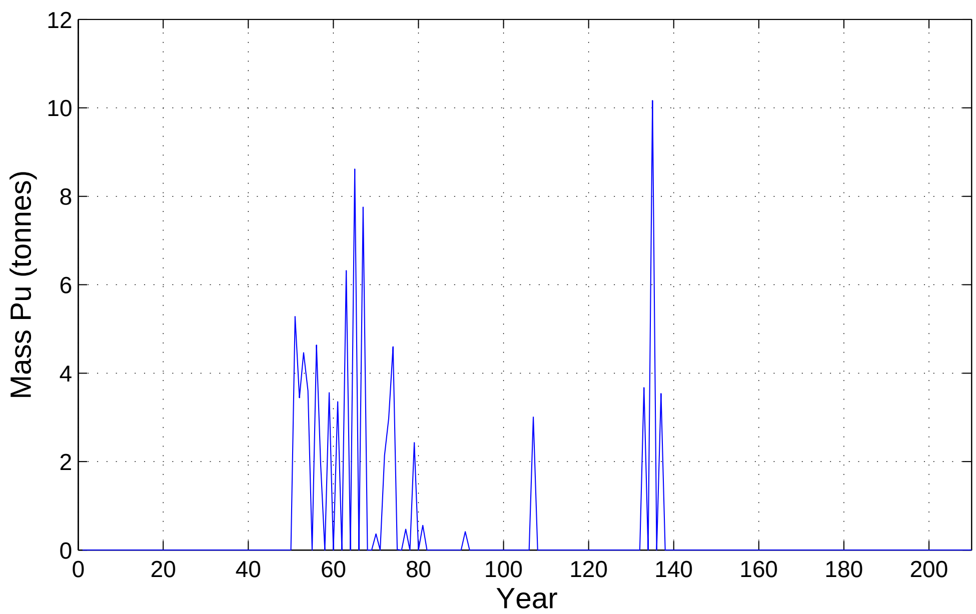

Figure 3.18: Net amount of Pu in storage at the end of each year in the ORION Case 1 transition scenario

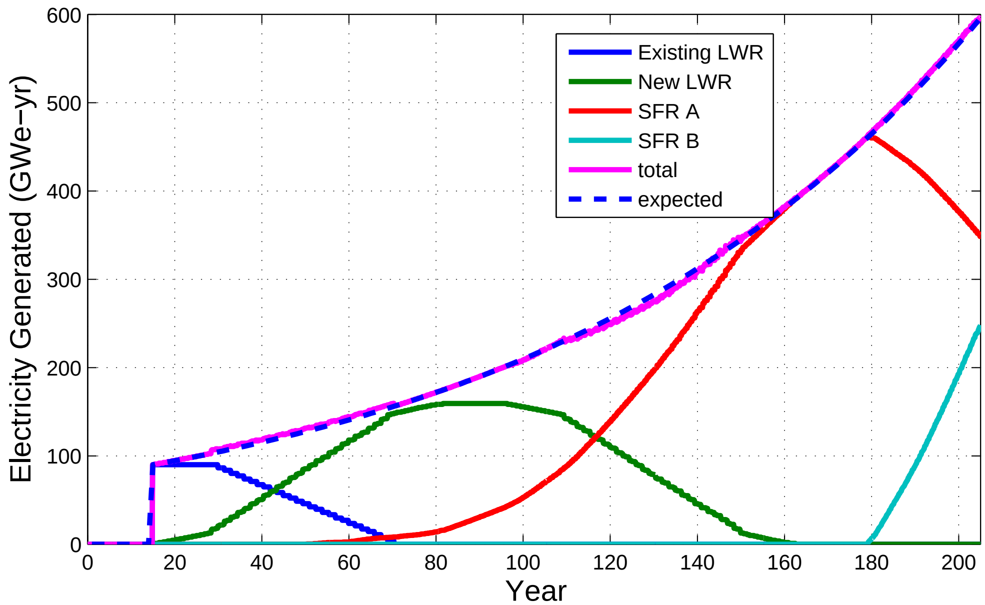

Figure 3.19: Electricity generation profile created by ORION's Dynamic Resource Control to match DYMOND results for Case 2

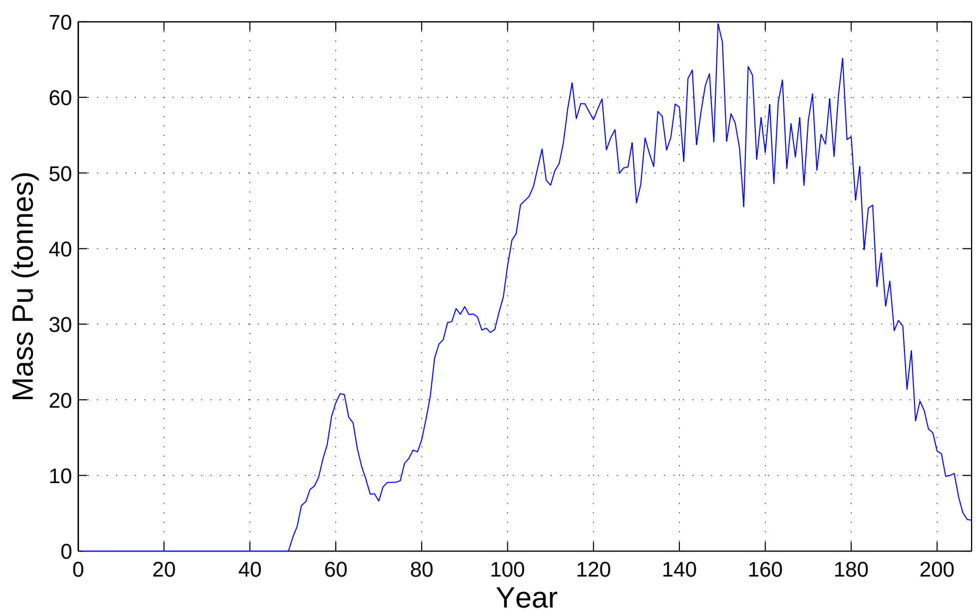

Figure 3.20: Total amount of Pu in storage at each time step in the Case 2 simulation which utilizes ORION's DRC to reproduce the DYMOND transition scenario

Case 3 - Faster SFR Start-Up Rate

After recreating the DYMOND transition scenario in ORION with two different methods, an effort to decrease the transition time began. These efforts focused primarily on adjusting the restriction on SFR startup rate in order to decrease the number of new LWRs needed, therefore shortening the amount of time that LWRs are included in the scenario. In the first iteration of this effort, the restriction of only deploying one new SFR per year for the first 20 years was reduced to affect only the first 10 years. As shown in Figure 3.21, this allowed the last new LWR to be built in year 77, permitting the transition to be completed by the year 157 instead of 161 found previously in Cases 1 and 2. This not only shortens the transition time by 4 years but also decreases the number of new LWRs that must be introduced from 177 to 139. Throughout this scenario, the total amount of Pu in storage remained below 60 tonnes in each time step, shown in Figure 3.22.

Case 4 - Almost Unlimited SFR Start-Up Rate

In the final iteration of ORION DRC simulations, the SFR start up restrictions were reduced again from 10 years down to only 5. This was done to see if it were possible to implement SFRs much faster without needing additional LWRs from a mass flow perspective. While this simulation is a bit of an extreme case due to the rate of introduction of SFRs as well as the large reprocessing capacities needed in the first few years, it showed that there is sufficient plutonium available to support this rate of transition. In this scenario, the final LWR was started up in year 58, allowing the transition to be complete by year 138 as opposed to year 161 in Cases 1 and 2. The number of new LWRs needed was also greatly reduced compared to the original scenario from 177 to 107. The total amount of plutonium in storage remained below 70 tonnes for the duration of this simulation. In the final years of the simulation, when the second generation of SFRs are introduced, the amount of plutonium in storage drastically decreases. This is due to a higher utilization rate of the spent fuel in the first few years so there are not as many reserves to pull from in the final years. It may be feasible to hold off on introducing this lower breeding ratio in order to sustain the reactor fleet beyond this simulation time frame.

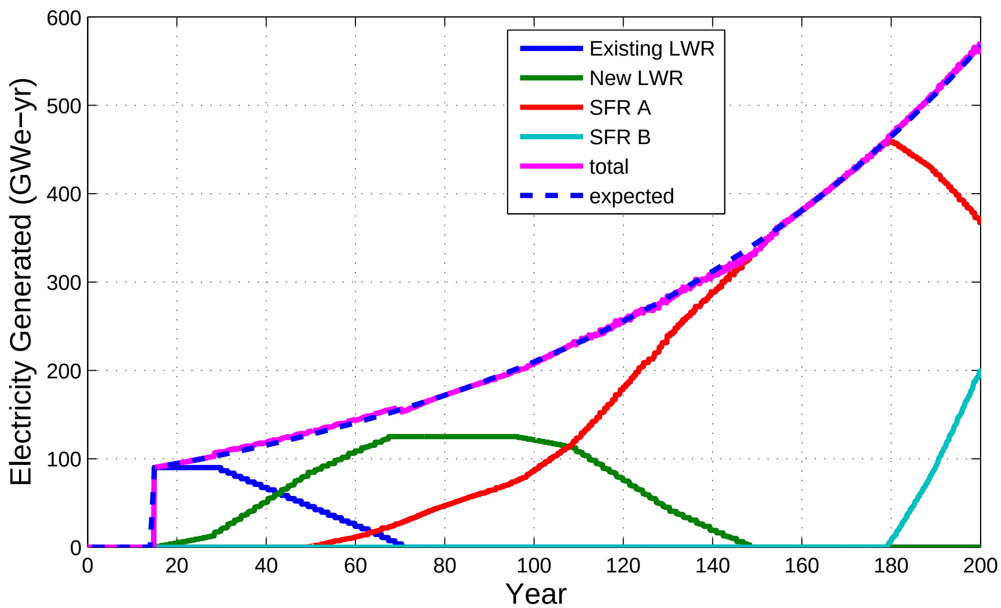

Figure 3.21: Electricity generation profile created by ORION's Dynamic Resource Control with slightly increased SFR start-up rate for Case 3

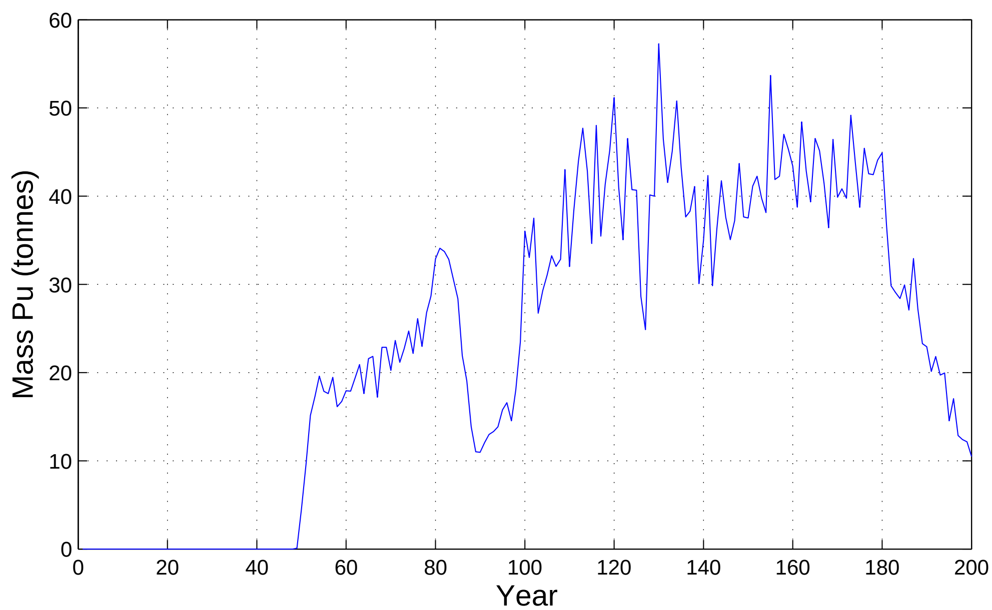

Figure 3.22: Total amount of Pu in storage at each time step in the Case 3 simulation with slightly increased SFR start-up rate

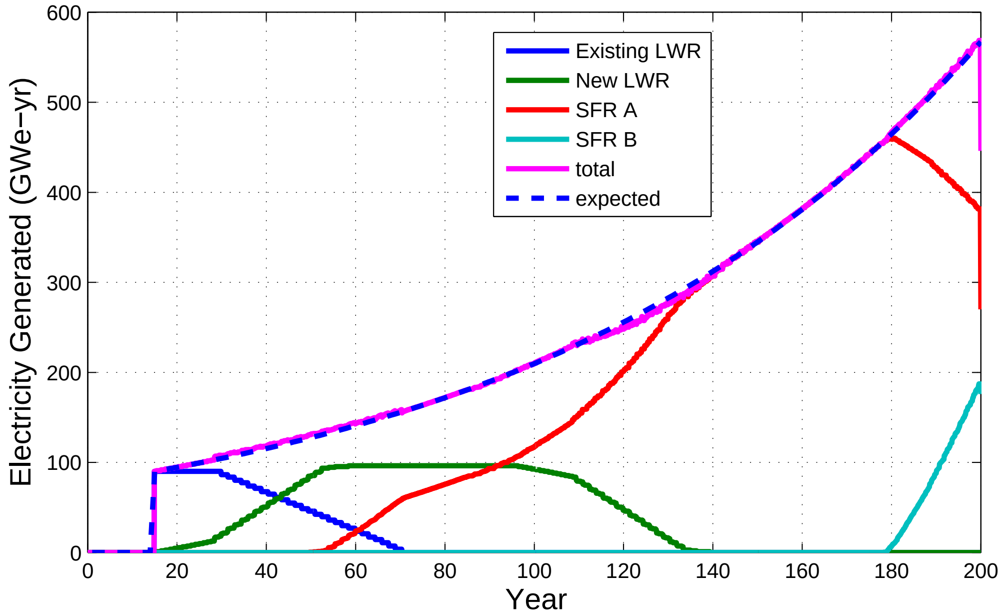

Figure 3.23: Electricity generation profile created by ORION's Dynamic Resource Control with an unconstrained SFR start-up rate for Case 4

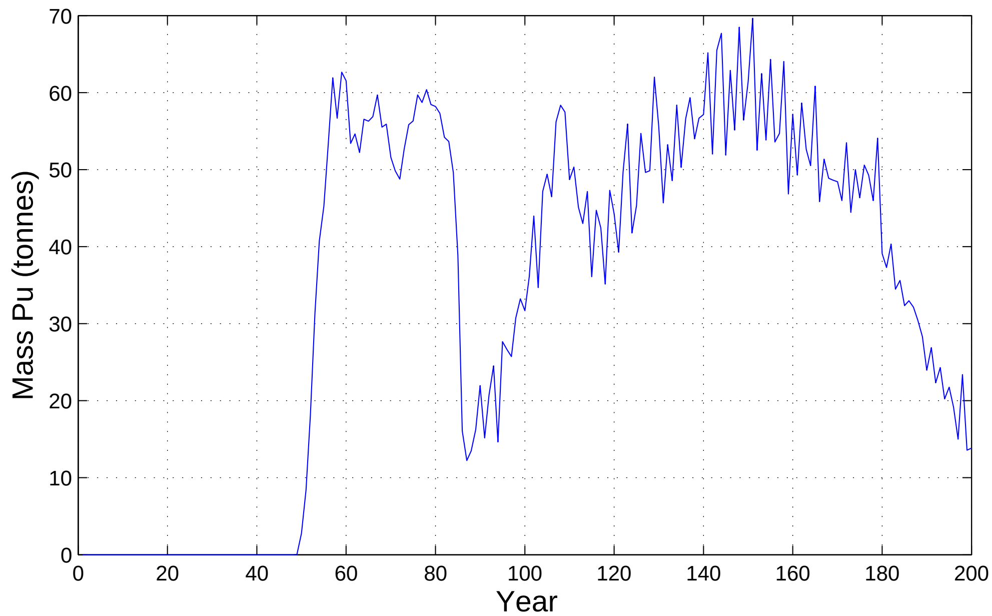

Figure 3.24: Total amount of Pu in storage at each time step in the Case 4 simulation which utilizes an unconstrained SFR start-up rate

3.5 Summary of Transition Analyses

The scenarios created here provide important insight into the best approach for the EG01-EG23 transition. These scenarios can inform on things like reprocessing capacity needs and rate of reactor startup necessary to sustain the transition as well as meet the electricity demand. While these scenarios meet all of the initial constraints, it is important to also evaluate the realism of each approach. For example, the original DYMOND transition simulation fulfills the need to keep the separated Pu below 100 tonnes each year, as well as requires that reprocessing start no earlier than year 2050 for a realistic development and deployment time line, however the reprocessing plant capacities are required to fluctuate from year to year. While the growth of reprocessing capacity requirements is gradual, and therefore not unobtainable, it would require a reprocessing facility to operate well under its full capacity for many years until the demand caught up. Further, when this scenario is modeled in ORION with a time step size of 1 month, the reprocessing facility is simulated to only operate for two to three months each year in order to meet its demand and then it does not operate for the remainder of the year. The monthly plutonium reprocessing throughputs generated in ORION, along with their associated annual average values, for the each case are shown in Figure 3.25. Seen here, the reprocessing capacity gradually increases throughout the Case 1 simulation with a measurable decrease in the last 15 years. When the DRC is used to control the startup of reactors for this same scenario (Case 2) the reprocessing capacity is spread out slightly allowing the reprocessing facility to continuously operate, however the throughput in each month still varies greatly. This spreading of the reprocessing capacity provides a small buffer of recycled plutonium which could be used to fabricate fuel in the event of an unplanned outage of the reprocessing facility.

Either of these scenarios developed with ORION's month time step, continuous but variable throughput or biannual operation of the facility, are economically unrealistic to sustain. Additionally, the gradual increase in demand from year to year would be more realistically represented by a stepwise function at each time that a new reprocessing facility is needed. While it is important to maintain the amount of separated plutonium low for material security concerns, strictly managing the reprocessing capacities from year to year in this way may

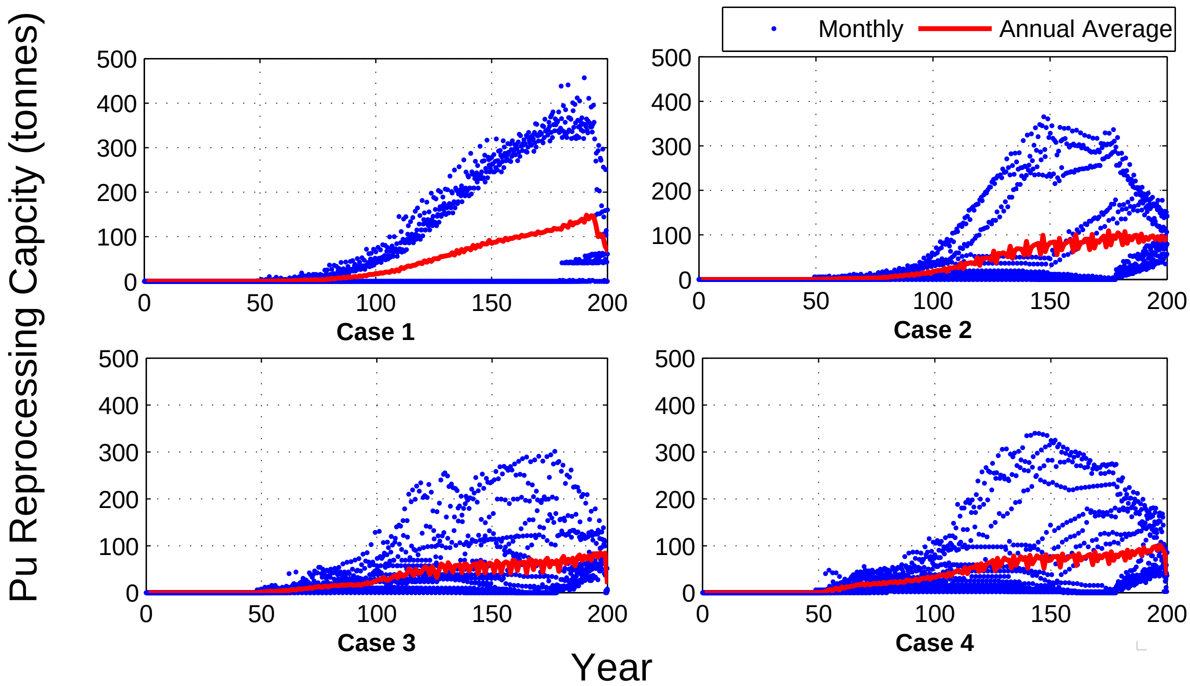

Figure 3.25: Pu reprocessing capacities at each time step in the ORION simulations with 12-month moving averages for comparison

introduce unforeseen inefficiencies within the system. In order to find alternative approaches to the transition, it may become important to allow each fuel cycle simulator to be used individually to create many unique transition scenarios. The efforts to ensure that multiple simulators can reproduce a given scenario are important for validation purposes, but it is possible that they also distract from the production of many alternate transition solutions. Since each code is different in how it takes in the fuel cycle information, attempting to match another code's specifications can prove rather difficult and time consuming in some cases. If some of this effort was able to go into producing unique results, multiple solutions could be formed which would provide a range of possible approaches from which the most advantageous could be chosen.

Chapter 4

CyBORG Development

Creation of a Cyclus module repository which integrates a stand-alone build of ORIGEN was initiated in order to introduce the ability to perform accurate depletion and decay calculations in real time during a fuel cycle simulation. Since ORIGEN tracks over 2000 nuclides, the calculations performed with it for fuel cycle analysis would be comparable to other simulators which utilize accurate isotopics, such as ORION [11]. Introducing this functionality to Cyclus not only further enhances the flexibility of the code but also allows it to be used for more sophisticated simulations. For instance, this functionality is imperative when attempting to determine small scale isotopic changes to materials which have large scale effects on the fuel cycle system, such as the effects of continuous reprocessing on plutonium fissile content. These small changes in plutonium isotopic makeup can have major impacts on the amount of fresh fuel available later in the simulation.

4.1 CyBORG Architecture

The generic archetype structure created to introduce ORIGEN into Cyclus is shown in Figure 4.1. Here, the user defines state variables which specify the mass flow and fuel cycle parameters necessary for the archetype to perform the proper calculations in the input file. Within the simulation, the archetype is responsible for communicating with the Cyclus Dynamic Resource

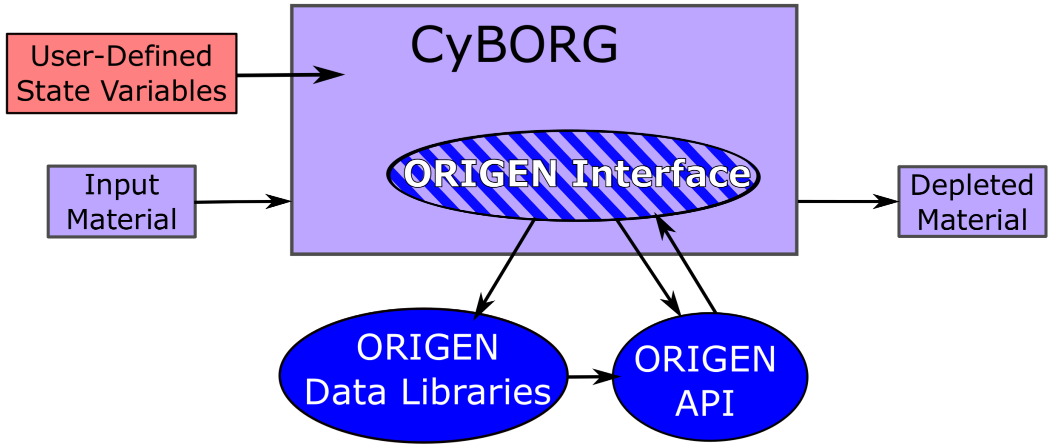

Figure 4.1: Illustration of CyBORG architecture where the purple regions represent the Cyclus pieces, blue represents the ORIGEN components, and red represents user specifications

Exchange in order to accept and offer up material trades. Once the Cyclus material information is passed into the archetype, it is converted into a form that ORIGEN can accept. From here, the state variable and material information is passed into an ORIGEN interface which defines the depletion problem for ORIGEN. Once the necessary interpolation parameters are set, the appropriate ORIGEN reactor data libraries are interpolated to create the problem specific cross section library. Following interpolation, the remaining archetype specifications are passed through the ORIGEN interface and into the ORIGEN API so that the depletion calculation can be performed. Once the depletion is complete, the material information is passed back into the archetype and converted once again into the Cyclus format. The depleted material object is then ready to be offered to the Dynamic Resource Exchange as a new commodity.

4.2 ORIGEN-Based Reactor Facility

The first archetype in development for CyBORG is a reactor facility. The goal of this reactor facility is to accurately model the depletion calculations throughout numerous reactor types and configurations based on user input. The user is required to specify the reactor data library for interpolation by defining library identifiers such as reactor type (based on ORIGEN assembly

types) and fuel type. Additionally, the interpolation parameters are set by the user by including specifications about the reactor like the enrichment, fuel temperature, or moderator density. The ORIGEN interface then has the necessary information to create a problem specific cross section library for the simulation. The remainder of the user input relates to the operating conditions of the reactor, including but not limited to reactor power capacity, cycle length, and capacity factor. More advanced parameters such as cycle by cycle power profiles are included as optional input to allow for a range of user sophistication and specificity. These variables, along with the material information provided by the Cyclus DRE and the problem-specific library, are used as input to the ORIGEN depletion calculation.

This work was divided into three major portions, the first being the Cyclus interfacing pieces that comprise the archetype itself. The second part of this work was the creation of the ORIGEN interface and development of ORIGEN interpolation methods. This second portion of the work was completed by another student in support of this project. The final step in developing the CyBORG reactor facility was the process of combining the first two pieces into a single build. This effort has been a collaborative effort with work on the Cyclus-specific unit tests coming from this project, and significant contributions to the coupled ORIGEN interface/CyBORG build coming from our academic advisor.

Development of the Cyclus facing portion of this reactor facility utilized the `StubFacility` from the Cycstub repository as a starting point. This repository provides the framework for a Cyclus agent without any specific functionality in order to make the development process easier and faster for those new to Cyclus. To begin molding this generic facility into a reactor, user-defined state variables discussed above were added. Form here, the ability to interface with Cyclus' Dynamic Resource Exchange (DRE) was added in order to allow the facility to request and offer materials within a simulation. This interface with the DRE utilized the updated material buy and sell policies, rather than the original bidding and trading method, similar to what was implemented in the `Storage` facility.

Once the reactor facility was able to be included in the material flows of a simulation, the reactor functionalities were ready to be added. First, the relatively simple function meant to load the fuel into the reactor was created. With a check for empty space in the reactor, the fresh

fuel is taken out of the incoming material buffer and added to the core buffer. On the other end, the function which pulls the spent fuel out of the reactor at the appropriate time was included as well. This function also involved the interface with ORIGEN since the depletion calculations were to be performed immediately before the fuel is removed from the core. The first interaction with the ORIGEN interface required passing in the state variables which define the interpolation parameters. With the appropriate data library specified and interpolation parameters in place, the ORIGEN interface's `interpolate()` function could be called.

Once the problem-specific library was prepared, the remaining reactor operating specifications were required to be passed into the ORIGEN interface. This portion of the interface with ORIGEN required converting the Cyclus material composition data into a form that ORIGEN could accept. This included utilizing PyNE's nuclide naming module for the conversion of the nuclide IDs from the default Cyclus format (ZZAAAMMMM) to a form which ORIGEN accepts (ZZZAAAM) where Z is the atomic number of the isotope, A is the atomic mass number, and M is used to represent the the exitation state [30]. Once all of the remaining specifications were passed into the ORIGEN interface, the solver was called to utilize the ORIGEN API to complete the depletion evaluation and return an array of output fuel composition. By converting the nuclide IDs back into Cyclus acceptable format with the PyNE module once again, the material vectors were ready to be added back into the Cyclus resource exchange.

In addition to the core functionality of the CyBORG reactor, development of Cyclus-facing unit tests for this facility have begun. The framework is in place such that additional unit tests may be easily added as the reactor functionality increases. The unit tests currently included and in development are shown in Table 4.1.

4.3 CyBORG Reactor Path Forward

In its current state, the CyBORG reactor is a first iteration prototype which is primarily applicable to simulating reactors with uranium-based fuels. Immediate upcoming improvements include generalizing the processing parameters such that any reactor type included in the ORIGEN data libraries can be simulated with CyBORG. These improvements will allow the

Table 4.1: Cyclus-facing unit tests for CyBORG developed (top) and still to be implemented (bottom)

Test Name	Purpose
InitialState	Verify that the reactor facility's initial variables are declared properly
Print	Ensure that the reactor's specifications are printed appropriately
Tick	Check material loading behavior for beginning of cycle
Tock	Verify that the depletion and discharge procedures as a whole operate properly
CheckDepletionCalls	Ensure that calls to ORIGEN interface operate as expected
DischargeTime	Determine whether material is being discharged from the reactor at the appropriate time
CompositionCheck	Verify that the expected composition is produced from Cyclus-to-ORIGEN format conversion
BufferCapacity	Check that the reactor buffers (input material, core, discharge fuel, etc.) contain the appropriate amount of material
MultipleFuelSource	Check reactor functionality when multiple sources are providing small batches of fuel, ensure that they are combined appropriately
EndOfLifetime	Verify that full reactor core is discharged at the end of the reactor lifetime and that reactor discontinues operations

CyBORG reactor facility to be applicable to a wide range of fuel cycle simulation needs. Since the CyBORG reactor archetype does not actually include any ORIGEN source code, only calls on it when needed, distribution of CyBORG is much the same as Cyclus and Cycamore. CyBORG has been made available on Github at [17] so that an interested user can retrieve the most recent version at any time. This structure allows for continuous improvements on reactor features and development of new archetypes to be available to the user. In order to use CyBORG, the user must also have the ORIGEN 6.2 shared library objects installed on their machine which are available for distribution through the SCALE 6.2 release from Oak Ridge National Laboratory's Radiation Safety Information Computational Center (RSICC) and will be distributed as a standalone object in the future [31].

Chapter 5

Conclusions and Recommendations for Future Work

Fuel cycle simulators play an valuable role in the analysis of the most efficient ways to implement various nuclear fuel cycles. By quickly calculating hundreds of years of mass flow calculations between numerous fuel cycle facilities, complex problems can be completed rather quickly. This rapid evaluation is important not only for evaluating the effects of perturbations in the current fuel cycle, but also for investigating possible advanced fuel cycles and the best way to implement them. In this work, Cyclus and ORION were used for evaluating the transition from the current once-through fuel cycle to an advanced cycle which utilized continuous reprocessing and fast reactors. Before this transition was simulated in Cyclus, creation of a `Storage` facility was completed. This, along with developments outside of this project, allowed the recipe-based transition scenarios to be completely simulated using the Cycamore repository. In order to illustrate the expanded abilities of Cyclus, integration tests were performed which verified that Cyclus was ready to begin the transition analysis. The Cyclus transition simulation, as well as the initial ORION simulation, were performed to verify that a transition approach created by DYMOND was reproducible with other tools. This DYMOND approach utilized a "just-in-time" reprocessing philosophy where the reprocessing capacity in each year was specified to be only enough to sustain the reactor fleet in the following year. This approach fit

within the constraints that were placed upon the transition scenarios, however the fluctuating reprocessing capacities introduce a dimension which is only possible within the simulation scope. Reproducing this capacity profile in reality would require facilities to rarely operate at full capacity and would not leave much room for error with unexpected facility down time. After recreating this DYMOND scenario, ORION was utilized to further analyze the reactor deployment profile and attempt to shorten the transition time while maintaining the original simulation philosophy of just-in-time reprocessing. This study determined that there was sufficient fissile material to sustain a faster transition by removing the startup constraints placed upon the first wave of fast reactors.

The EG01-EG23 transition analysis is one of many that will be completed before the transition analysis portion of the Fuel Cycle Options campaign is concluded. Currently, the transition being evaluated is the EG29 case which involves continuous recycle between fast reactors and MOX LWRs. With much more complex mass flows, this case presents new challenges to be overcome by each simulator involved in the calculations. However, it is essential to continue these analyses because although the E&S Study evaluated the challenges associated with introducing advanced technologies, the transition analyses provide unique insight into potential time-lines for deployment and resource demands throughout. By performing these evaluations for each of the promising fuel cycle options outline in the E&S Study, additional information about the benefits or challenges of a particular evaluation group may come to light. It is important within these evaluations, as with any simulation-based study, that the realistic implications of each assumption be analyzed to be sure that it would hold up outside of the scope of the simulation.

In addition to performing transition analyses, this work included the development of a reactor archetype for Cyclus which utilizes ORIGEN for its depletion and decay calculations. By introducing an accurate depletion engine into Cyclus, more advanced scenarios are able to be simulated than with the recipe-based approach. This includes modeling advanced types of reactors such as molten salt reactors which require updated cross section data throughout a simulation. In addition to advanced reactors, using cross section data is the best way to determine the effects of perturbations to the reactor on the rest of the fuel cycle. To continue

this work, advanced options will be added to the CyBORG reactor in order to cater to a wider range of user sophistication. This includes adding in flexibility so that any of the ORIGEN reactor data libraries can be utilized as well as implementing an expanded database of optional interpolation parameters. In addition to continued improvements to the reactor module, additional agents for the CyBORG repository are in the planning stages. This cross section-based modeling can be utilized in any stage of the fuel cycle and will be useful for performing precise simulations for systems where small changes in resource isotopics are important.

Bibliography

[1] Science US Department of Energy Office of Nuclear Energy and Technology. The history of nuclear energy. DEO/NE-0088. 1

[2] Carbon pollution emission guidelines for existing stationary sources: Electric utility generating units. 40 CFR Part 60(64661), 2015. 1

[3] Steve Skutnik Remy Devoe, Justin Knowles. Unintended concequences lurking in EPA clean power plan, 2014. URL http://ansnuclearcafe.org/2014/08/20/unintend-consequences-lurking-in-epa-clean-power-plan/. 2

[4] Nuclear Energy Institute. Why nuclear energy?, 2016. URL http://www.nei.org/Why-Nuclear-Energy. 2

[5] Matthew L Wald. Energy policy in the age of wishful thinking. 2015. URL http://www.nei.org/News-Media/Speeches/Energy-Policy-in-the-Age-of-Wishful-Thinking. 2

[6] Nuclear Energy Institute. US nuclear energy industry even safer since fukushima. URL http://www.nei.org/News-Media/News/News-Archives/US-Nuclear-Energy-Industry-Even-Safer-Since-Fukush. 2

[7] Inderscience Publishers. Rebalancing the nuclear debate through education, September 2011. URL https://www.sciencedaily.com/releases/2011/09/110909111444.htm. 2

[8] Mimi Limbach. Transparency and proactive outreach in good nuclear energy communications, 2013. URL http://goo.gl/Ynm9F2. 2

[9] John Timmer. New nuclear plants in us may offset planned closures, February 21, 2016 2015. URL http://arstechnica.com/science/2015/11/new-nuclear-plants-in-us-may-offset-the-planned-closures/. x, 2

[10] Kathryn D. Huff, Matthew J. Gidden, Robert W. Carlsen, Robert R. Flanagan, Meghan B. McGarry, Arrielle C. Opotowsky, Erich A. Schneider, Anthony M. Scopatz, and Paul P. H.

Wilson. Fundamental concepts in the cyclus nuclear fuel cycle simulation framework. *Advances in Engineering Software*, 94:46–59, 2016. ISSN 0965-9978. doi: http://dx.doi.org/10.1016/j.advengsoft.2016.01.014. URL http://www.sciencedirect.com/science/article/pii/S0965997816300229. x, 3, 6, 8, 10

[11] Eva E Sunny, Robert Gregg, Timothy M Ault, Jess C Gehin, Joshua L Peterson, Jeffrey J Powers, and Andrew Worrall. Fuel cycle assessment: Evaluation and analyses using ORION for US fuel cycle options. Report, Oak Ridge National Laboratory (ORNL), 2014. 3, 6, 8, 45

[12] Ian C Gauld, Georgeta Radulescu, Germina Ilas, Brian D Murphy, Mark L Williams, and Dorothea Wiarda. Isotopic depletion and decay methods and analysis capabilities in SCALE. *Nuclear Technology*, 174(2):169–195, 2011. 3

[13] R Wigeland, TA Taiwo, H Ludewig, M Todosow, W Halsey, J Gehin, R Jubin, J Buelt, S Stockinger, and K Jenni. Nuclear fuel cycle evaluation and screening - final report. *USA: US DOE Fuel Cycle Technologies*, 8, 2014. ix, x, 3, 4, 5

[14] AM Yacout, JJ Jacobson, GE Matthern, SJ Piet, and A Moisseytsev. Modeling the nuclear fuel cycle. In *The 23rd International Conference of the System Dynamics Society Boston*. 6, 8, 9

[15] Jacob Jacobson, Abdellatif M Yacout, Gretchen Matthern, Steven Piet, David E Shropshire, and Chris Laws. Vision: Verifiable fuel cycle simulation model. *Transactions of the American Nuclear Society*, 95:157, 2006. ISSN 0003-018X. 6, 8, 9

[16] Sandia National Laboratores. Nuclear fuel cycle options catalog. URL https://connect.sandia.gov/sites/NuclearFuelCycleOptionCatalog/SitePages/a/homepage.aspx. xi, 6, 25, 26

[17] Jennifer Littell, Nicholas Sly, and Steven Skutnik. Cyborg, 2016. URL https://github.com/sskutnik/cyborg. 7, 49

[18] Lionel Boucher and Jean-Paul Grouiller. "COSI": The complete renewal of the simulation software for the fuel cycle analysis. In *14th International Conference on Nuclear Engineering*, pages 865–870. American Society of Mechanical Engineers. 8, 9

[19] Erich A Schneider, Charles G Bathke, and Michael R James. NFCSim: a dynamic fuel burnup and fuel cycle simulation tool. *Nuclear technology*, 151(1):35–50, 2005. 8

[20] Robert Gregg. ORION user guide - v4. Report, National Nuclear Laboratory, 2015. 9

[21] Päi I Davidsen. Powersim as a tool for systen dynamics modeling. *Powersim Studio*, pages 1–3, 2003. 9

[22] Christopher A Juchau, Mary Lou Dunzik-Gougar, and Jacob J Jacobson. Modeling the nuclear fuel cycle. *Nuclear technology*, 171:136–141, 2010. 9

[23] Matthew J Gidden. An agent-based modeling framework and application for the generic nuclear fuel cycle. Ph.D. Thesis - University of Wisconsin at Madison, 2015. 11

[24] Germina Ilas, Ian C Gauld, and Georgeta Radulescu. Validation of new depletion capabilities and ENDF/B-VII data libraries in SCALE. *Annals of Nuclear Energy*, 46:43–55, 2012. ISSN 0306-4549. 12

[25] S Skutnik, F Havüj, D Lago, and I Gauld. Development of an object-oriented ORIGEN for advanced nuclear fuel modeling applications. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013), Sun Valley, ID*. 12

[26] Nicholas Sly, Steven Skutnik, and William Wieselquist. Portable reactor data library interpolation via self-describing ORIGEN libraries. In *Physics of Reactors (PHYSOR 2016), Sun Valley, ID, May 2016*. 13

[27] Cyclus Developers. Cyclus additional module repository, 2016. URL `https://github.com/cyclus/cycamore`. 15

[28] Kathryn Huff. Commodconverter, 2014. URL https://github.com/katyhuff/CommodConverter. 16

[29] Google Inc. Google test, 2016. URL https://github.com/google/googletest. 16

[30] PyNE Development Team. Nuclide naming module - pyne.nucname, 2016. URL http://pyne.io/pyapi/nucname.html. 48

[31] Oak Ridge National Laboratory. Radiation safety information computational center, 2016. URL http://www.rsicc.ornl.gov/. 49

Appendix

Appendix A

Simulation Specifications

Table A.1: Parameters for use in the Reactor Deployment Unit Test

Duration	100 years [Begin year 0; end year 99]
Fabrication	There is no modeling of fabrication features in this unit test
Fuel	No specific fuel composition features are used in this test case. An infinite supply of generic reactor fuel is assumed.
Deployment	A single 1 GWe-yr generic reactor is deployed at the start of year 10. The capacity factor is 0.9, so the electricity generated is 0.9 GWe-yr. After deployment, the reactor operates at steady state throughout the duration of the simulation.
Retirement	There is no modeling of facility retirement features in this unit test.
Core Loading and Discharge	The total initial core loading of a deployed reactor is 90 t of heavy metal in year 10. The annual core charge is 20 t/yr and core discharge is 19.909 t/yr. Annual loading and discharge occur at the beginning of each year, from year 16 throughout the duration of the simulation. The generic reactor operates on 20 t/yr of an unlimited supply of heavy metal.
Wet and Dry Storage	The discharged fuel is cooled in wet storage for a total of 4 years. The first discharge enters wet storage at the beginning of year 16 and this continues throughout the duration of the simulation. After cooling, the fuel is moved to dry storage, where fuel accumulates throughout the duration of the simulation. The first batch of fuel is moved from wet storage immediately at the beginning of year 20.
Separations	There is no modeling of separations features in this unit test.

Table A.2: Parameters for use in the Growth Unit Test

Duration	200 years [Begin year 0; end year 199]
Fabrication	There is no modeling of fabrication features in this unit test
Fuel	No specific fuel composition features are used in this test case. An infinite supply of generic reactor fuel is assumed.
Deployment	A 100 GWe-yr generic reactor fleet is in existence at the beginning of the simulation. The capacity factor is 0.9, so the electricity generated is 90 GWe-yr. After deployment, the reactor operates at steady state throughout the duration of the simulation. Each year the energy demand grows by 1%. So 100 GWe-yr in year 0 becomes 101 GWe-yr in year 1, and so on. Reactors (1GWe-yr) are deployed as needed to meet the energy demand. The final energy demand is 651.9 GWe-yr in year 199.
Retirement	There is no modeling of facility retirement features in this unit test.
Core Loading and Discharge	The total initial core loading of a deployed reactor is 90 t of heavy metal. The annual core charge per reactor is 19.909 t/yr and core discharge is 19.909 t/yr. Annual loading and discharge occur at the beginning of each year, from year 0 throughout the duration of the simulation. The generic reactors operate on 19.909 t/yr of an unlimited supply of heavy metal.
Wet and Dry Storage	The discharged fuel is cooled in wet storage for a total of 4 years. The first discharge enters wet storage at the beginning of year 1 and this continues throughout the duration of the simulation. After cooling, the fuel is moved to dry storage, where fuel accumulates throughout the duration of the simulatoin. The first batch of fuel is moved from wet storage immediately at the beginning of year 5.
Separations	There is no modeling of separations features in this unit test.

Table A.3: Cyclus input parameters for use in the EG01 steady state simulation

Simulation Parameters	Duration: 1200 months (100 years) Start Month: 1 (January) Start Year: 2015
Source	All default values
Enrichment	Tails Assay: 0.0025 Max Enrichment: 0.043
Storage	Tails Storage: All default values Wet Storage: Residence Time: 48 months Dry Storage: All default values
Reactor	Power Capacity: 891 MWe/yr # of Assembleis per Batch: 20 Assembly Size: 978.5 kg # of Assemblies per Core: 91 # of Spent Assembies held on site: 91 # of Fresh Assembiles held on site: 0 Cycle Time: 18 months Refuel Time: 0 (capacity factor accounted for in Power Capacity)
Deploy Institution	112 Reactors Initialized Reactor Lifetimes: 720 months

Table A.4: Cyclus input parameters for use in the EG23 steady state simulation

Simulation Parameters	Duration: 1200 months (100 years) Start Month: 1 (January) Start Year: 2000
Natural Uranium Source	Inventory Size: 1e10 kg
Plutonium Source	Inventory Size: 1.5e6 kg
SFR Blanket	Power Capacity: 18.027 MWe/yr # of Assemblies per Batch: 1 Assembly Size: 700 kg # of Assemblies per Core: 6 # of Spent Assemblies held on site: 6 # of Fresh Assemblies held on site: 0 Cycle Time: 12 months Refuel Time: 0 (capacity factor accounted for in Power Capacity)
SFR Driver	Power Capacity: 891 MWe/yr # of Assemblies per Batch: 20 Assembly Size: 978.5 kg # of Assemblies per Core: 91 # of Spent Assemblies held on site: 91 # of Fresh Assemblies held on site: 0 Cycle Time: 18 months Refuel Time: 0 (capacity factor accounted for in Power Capacity)
Fuel Fabrication	Fill Commodities in Order of Preference: Recovered Uranium, Natural Uranium Fissile Commodities in Order of Preference: Recovered Plutonium, External Plutonium
All Storage Facilities	All Default Values
Deploy Institution	112 Reactors Initialized Reactor Lifetimes: 720 months

Table A.5: Transition analysis input parameters taken from DYMOND scenario

LWR Specifications	Thermal Power: 3000 MWth Electrical Power: 1000 MWe Capacity Factor: 90% Core Inventory: 88.7 tonnes Reactor Lifetime: 60 years (80 years for New LWR) Enrichment: 4.21% Hold-up Time: 4.5 years
SFR A Specifications	Thermal Power: 1000 MWth Electrical Power: 400 MWe Capacity Factor: 90% Core Inventory: 37.62 tonnes Reactor Lifetime: 80 years Enrichment: 7.64% Breeding Ratio: 1.18 Hold-up Time: 5.44 years
SFR B Specifications	Thermal Power: 1000 MWth Electrical Power: 400 MWe Capacity Factor: 90% Core Inventory: 37.62 tonnes Reactor Lifetime: 80 years Enrichment: 8.53% Breeding Ratio: 1.12 Hold-up Time: 4.44 years
Other Specifications	Fuel Fabrication Losses: 0.2% Reprocessing Losses: 1% Enrichment Tailings: 0.25% Initial Electricity Demand: 100 GWe-y Annual Electricity Demand Growth : 1%

Table A.6: Electricity generation profile and reprocessing capacities from the DYMOND EG01-EG23 transition scenario

Year	Electricity Generated (Gwe-y)				Reprocessing Capacity (tHM/y)	
	LWR	New LWR	SFR-A	SFR-B	LWR	SFR
2010	90	0	0	0	0	0
2011	90	0	0	0	0	0
2012	90	0	0	0	0	0
2013	90	0	0	0	0	0
2014	90	0	0	0	0	0
2015	90	0	0	0	0	0
2016	90	0.9	0	0	0	0
2017	90	1.8	0	0	0	0
2018	90	2.7	0	0	0	0
2019	90	3.6	0	0	0	0
2020	90	4.5	0	0	0	0
2021	90	5.4	0	0	0	0
2022	90	6.3	0	0	0	0
2023	90	7.2	0	0	0	0
2024	90	8.1	0	0	0	0
2025	90	9	0	0	0	0
2026	90	10.8	0	0	0	0
2027	90	11.7	0	0	0	0
2028	90	12.6	0	0	0	0
2029	90	13.5	0	0	0	0
2030	87.53	16.87	0	0	0	0
2031	85.65	19.65	0	0	0	0
2032	83.02	23.18	0	0	0	0
2033	81.15	26.85	0	0	0	0
2034	78.53	30.37	0	0	0	0
2035	76.65	33.15	0	0	0	0
2036	74.02	36.68	0	0	0	0
2037	72.15	39.45	0	0	0	0
2038	69.53	43.87	0	0	0	0
2039	67.65	46.65	0	0	0	0
2040	65.02	50.18	0	0	0	0
2041	63.15	53.85	0	0	0	0
2042	60.53	57.37	0	0	0	0
2043	58.65	60.15	0	0	0	0
2044	56.02	63.68	0	0	0	0
2045	54.15	67.35	0	0	0	0
2046	51.53	70.87	0	0	0	0
2047	49.65	73.65	0	0	0	0
2048	47.02	78.08	0	0	484.85	0
2049	45.15	80.85	0	0	0	0
2050	42.53	84.37	0.69	0	85.41	0

Table A.6: Electricity generation profile and reprocessing capacities from the DYMOND EG01-EG23 transition scenario (continued)

Year	Electricity Generated (Gwe-y)				Reprocessing Capacity (tHM/y)	
	LWR	New LWR	SFR-A	SFR-B	LWR	SFR
2051	40.65	87.15	0.72	0	89.13	0
2052	38.02	91.58	0.72	0	89.12	0
2053	36.15	94.35	0.72	0	573.98	0
2054	33.53	98.77	0.72	0	89.13	0
2055	31.65	100.73	1.41	0	174.54	0
2056	29.02	105.08	1.44	0	663.11	0
2057	27.15	107.85	1.44	0	78.62	13.25
2058	24.53	111.45	2.13	0	644.56	13.83
2059	22.65	115.05	2.16	0	163.41	13.83
2060	20.02	117.74	2.85	0	733.68	13.83
2061	18.15	121.35	2.88	0	252.55	13.83
2062	15.53	124.88	3.57	0	723.17	27.09
2063	13.65	127.65	3.6	0	237.7	27.66
2064	11.02	131.25	4.29	0	807.97	27.66
2065	9.15	134.85	4.32	0	227.2	40.92
2066	6.53	137.55	5.01	0	793.13	41.49
2067	4.65	141.15	5.04	0	697.21	54.75
2068	2.02	144.75	5.73	0	778.3	55.33
2069	0.15	147.45	6.45	0	767.78	68.58
2070	0	148.5	7.2	0	1098.72	69.16
2071	0	149.4	7.92	0	845.79	82.41
2072	0	150.37	8.67	0	934.3	82.99
2073	0	151.28	9.06	0	882.94	96.24
2074	0	152.17	9.78	0	967.73	96.82
2075	0	153.08	10.5	0	957.23	110.07
2076	0	153.97	11.22	0	1184.81	123.91
2077	0	154.88	11.94	0	1408.07	138.31
2078	0	155.77	12.69	0	1639.37	152.14
2079	0	156.6	13.8	0	1668.47	166.55
2080	0	156.61	15.27	0	1794.13	174.04
2081	0	156.6	17.07	0	1912.98	187.87
2082	0	156.6	18.87	0	2031.83	201.71
2083	0	156.6	20.67	0	2150.69	215.54
2084	0	156.6	22.47	0	2269.54	229.37
2085	0	156.6	24.27	0	2384.06	243.78
2086	0	156.6	26.07	0	2446.6	265.1
2087	0	156.6	27.87	0	2457.16	293.34
2088	0	156.6	29.67	0	2662.48	327.92
2089	0	156.6	31.47	0	2382.96	362.5
2090	0	156.6	33.3	0	2349.58	397.07
2091	0	156.6	35.43	0	2595.76	431.65
2092	0	156.6	37.23	0	2316.24	466.23

Table A.6: Electricity generation profile and reprocessing capacities from the DYMOND EG01-EG23 transition scenario (continued)

Year	Electricity Generated (Gwe-y)				Reprocessing Capacity (tHM/y)	
	LWR	New LWR	SFR-A	SFR-B	LWR	SFR
2093	0	156.6	39.06	0	2525.28	500.81
2094	0	156.6	41.19	0	3013.89	535.39
2095	0	156.6	43.02	0	2980.5	569.96
2096	0	155.7	45.9	0	3077.1	604.54
2097	0	154.8	48.78	0	3169.36	639.7
2098	0	153.9	51.66	0	3460.73	680.61
2099	0	153	54.87	0	3355.75	715.19
2100	0	152.1	57.78	0	3694.15	750.35
2101	0	151.2	60.66	0	3500.67	791.26
2102	0	150.3	63.9	0	3879.93	826.42
2103	0	149.4	66.78	0	3578.14	881.74
2104	0	148.5	70.02	0	4290.63	937.07
2105	0	147.6	72.9	0	3746.43	992.39
2106	0	145.8	76.86	0	3773.11	1054.06
2107	0	144.9	80.1	0	3753.99	1109.96
2108	0	144	83.34	0	5436.18	1165.29
2109	0	143.1	86.58	0	4884.56	1227.53
2110	0	139.72	92.1	0	5879.28	1282.85
2111	0	136.95	97.23	0	6046.47	1345.09
2112	0	133.43	102.96	0	6097.48	1400.42
2113	0	129.75	109.14	0	5805.83	1476.49
2114	0	126.22	115.14	0	6807.99	1538.73
2115	0	123.45	120.27	0	6490.32	1600.97
2116	0	119.93	126	0	7701.49	1663.21
2117	0	117.15	131.46	0	6610.59	1769.25
2118	0	112.72	138.24	0	7193.97	1867.8
2119	0	109.95	143.7	0	7042.46	1977.87
2120	0	106.43	149.73	0	6896.54	2096.59
2121	0	102.75	155.94	0	6314.03	2211.85
2122	0	99.22	161.97	0	6804.56	2310.4
2123	0	96.45	167.46	0	6899.19	2420.48
2124	0	92.93	173.52	0	6618.52	2525.36
2125	0	89.25	179.73	0	6165.81	2655.61
2126	0	85.72	186.09	0	7134.39	2760.49
2127	0	82.95	191.61	0	5734.86	2876.33
2128	0	78.53	198.75	0	6848.72	2995.63
2129	0	75.74	204.21	0	5438.04	3111.46
2130	0	72.23	210.63	0	6979.99	3216.93
2131	0	69.45	216.09	0	5329.99	3333.34
2132	0	65.02	223.56	0	6812.54	3452.64
2133	0	62.25	229.05	0	5033.81	3574.81
2134	0	57.83	236.52	0	6612.27	3680.85

Table A.6: Electricity generation profile and reprocessing capacities from the DYMOND EG01-EG23 transition scenario (continued)

Year	Electricity Generated (Gwe-y)				Reprocessing Capacity (tHM/y)	
	LWR	New LWR	SFR-A	SFR-B	LWR	SFR
2135	0	55.87	241.38	0	4590.24	3880
2136	0	51.53	248.76	0	6415.72	3909.07
2137	0	48.75	254.61	0	5382.23	4045.65
2138	0	45.15	261.12	0	5395.35	4151.12
2139	0	41.55	267.93	0	5402.12	4294.61
2140	0	38.85	273.69	0	5341.32	4462.07
2141	0	35.25	280.2	0	4426.32	4529.75
2142	0	31.72	286.98	0	5676.32	4636.36
2143	0	28.95	292.8	0	4618.42	4840.7
2144	0	25.35	299.64	0	4705.47	4877.26
2145	0	21.75	306.51	0	4535.15	5077.56
2146	0	19.05	312.6	0	5113.67	5133.14
2147	0	15.45	319.47	0	3996.54	5319.03
2148	0	11.85	326.64	0	3290.09	5368.84
2149	0	9.15	332.43	0	2462.25	5574.33
2150	0	8.1	336.9	0	3710.46	5610.89
2151	0	7.2	341.22	0	1763.13	5817.53
2152	0	6.23	345.96	0	2169.59	5874.26
2153	0	5.32	350.28	0	976.14	6066.5
2154	0	4.43	354.63	0	864.67	6185.21
2155	0	3.52	359.28	0	413.63	6321.8
2156	0	2.63	363.63	0	354.76	6433.02
2157	0	1.72	368.31	0	0	6486.37
2158	0	0.83	372.99	0	0	6616.08
2159	0	0	377.58	0	0	6648.06
2160	0	0	381.21	0	0	6734.66
2161	0	0	385.14	0	0	6799.38
2162	0	0	388.77	0	0	6859.16
2163	0	0	392.73	0	0	6924.37
2164	0	0	396.69	0	0	6989.59
2165	0	0	400.65	0	0	7087.05
2166	0	0	404.61	0	0	7120.02
2167	0	0	408.6	0	0	7217.98
2168	0	0	412.89	0	0	7320.38
2169	0	0	416.88	0	0	7316.16
2170	0	0	421.2	0	0	7420.04
2171	0	0	425.49	0	0	7527.89
2172	0	0	429.48	0	0	7555.91
2173	0	0	433.8	0	0	7691.56
2174	0	0	438.12	0	0	7782.6
2175	0	0	442.47	0	0	7869.2
2176	0	0	447.12	0	0	7913.52

Table A.6: Electricity generation profile and reprocessing capacities from the DYMOND EG01-EG23 transition scenario (continued)

Year	Electricity Generated (Gwe-y)				Reprocessing Capacity (tHM/y)	
	LWR	New LWR	SFR-A	SFR-B	LWR	SFR
2177	0	0	451.47	0	0	8024.42
2178	0	0	456.12	0	0	8135.52
2179	0	0	459.93	0.54	0	8181.89
2180	0	0	457.02	8.13	0	8317.9
2181	0	0	454.14	15.69	0	8426.29
2182	0	0	450.9	23.61	0	8536.16
2183	0	0	448.02	31.2	0	8645.16
2184	0	0	444.78	39.45	0	8815.87
2185	0	0	441.9	47.04	0	8885.99
2186	0	0	437.94	56.01	0	9016.76
2187	0	0	434.7	63.96	0	9116.83
2188	0	0	431.46	72.24	0	9416.78
2189	0	0	428.22	80.52	0	9459.12
2190	0	0	422.7	91.08	0	9637.85
2191	0	0	417.57	101.28	0	9767.9
2192	0	0	411.84	112.38	0	9854.49
2193	0	0	405.66	123.63	0	9923.66
2194	0	0	399.66	135.03	0	10152.16
2195	0	0	394.53	145.56	0	10201.99
2196	0	0	388.8	156.69	0	10473.06
2197	0	0	383.34	167.55	0	10427.74
2198	0	0	376.56	179.76	0	10664.89
2199	0	0	371.1	190.95	0	10778
2200	0	0	365.07	202.41	0	10899.42
2201	0	0	358.86	214.38	0	10974.16
2202	0	0	352.83	226.17	0	11201.7
2203	0	0	347.34	237.42	0	11316
2204	0	0	341.28	249.27	0	11440.57
2205	0	0	335.07	261.57	0	11553.44
2206	0	0	328.71	273.72	0	11802.66
2207	0	0	323.19	285.36	0	11772.17
2208	0	0	316.05	298.62	0	12109.58

Appendix B

Additional Output

B.1 ORION Transition Output - Case 1

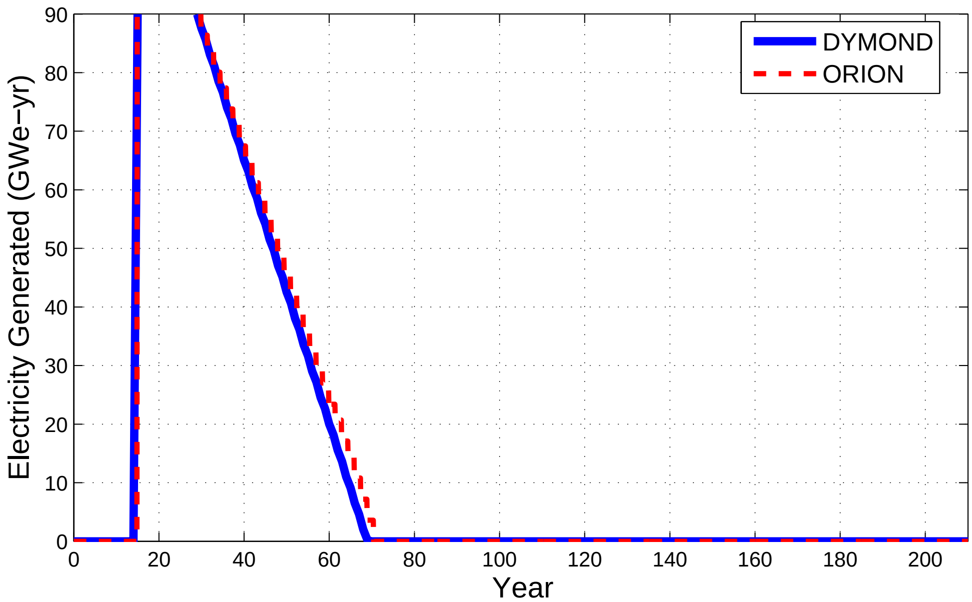

Figure B.1: Power generated by legacy Light Water Reactors in EG01-EG23 transition scenarion in ORION Case 1

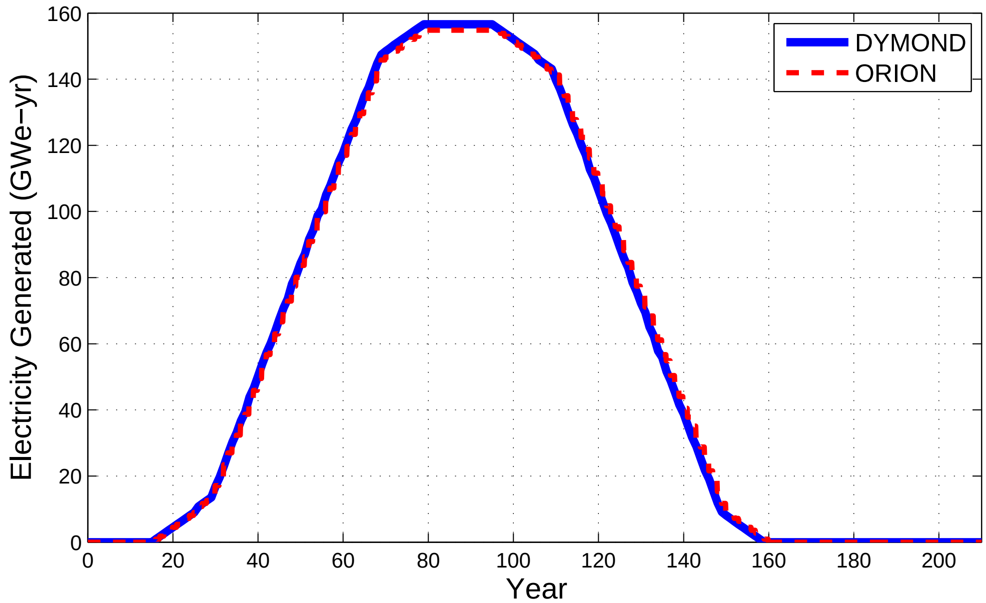

Figure B.2: Power generated by new Light Water Reactors in EG01-EG23 transition scenarion in ORION Case 1

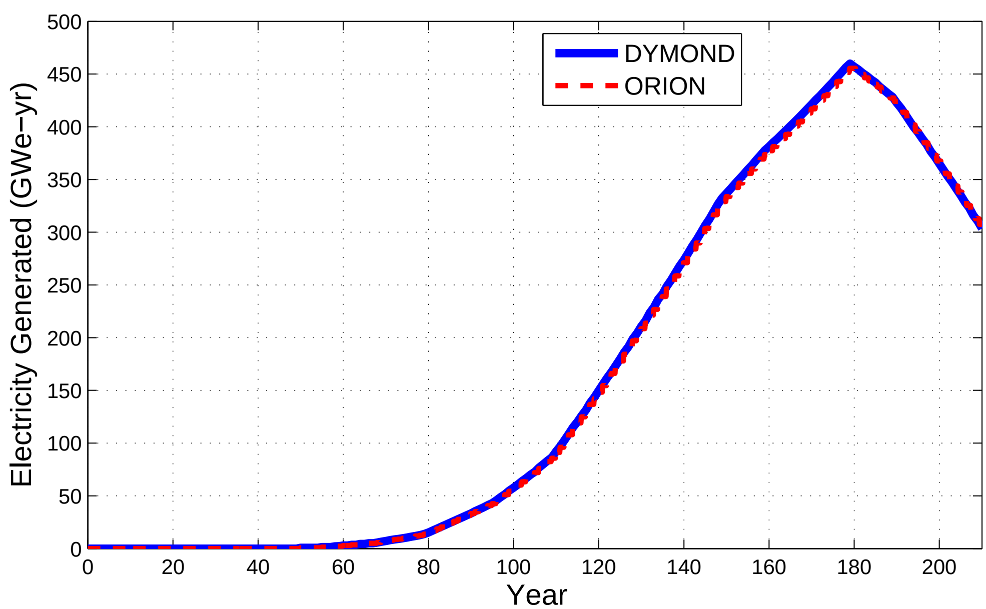

Figure B.3: Power generated by first generation Sodium-cooled Fast Reactors in EG01-EG23 transition scenarion in ORION Case 1

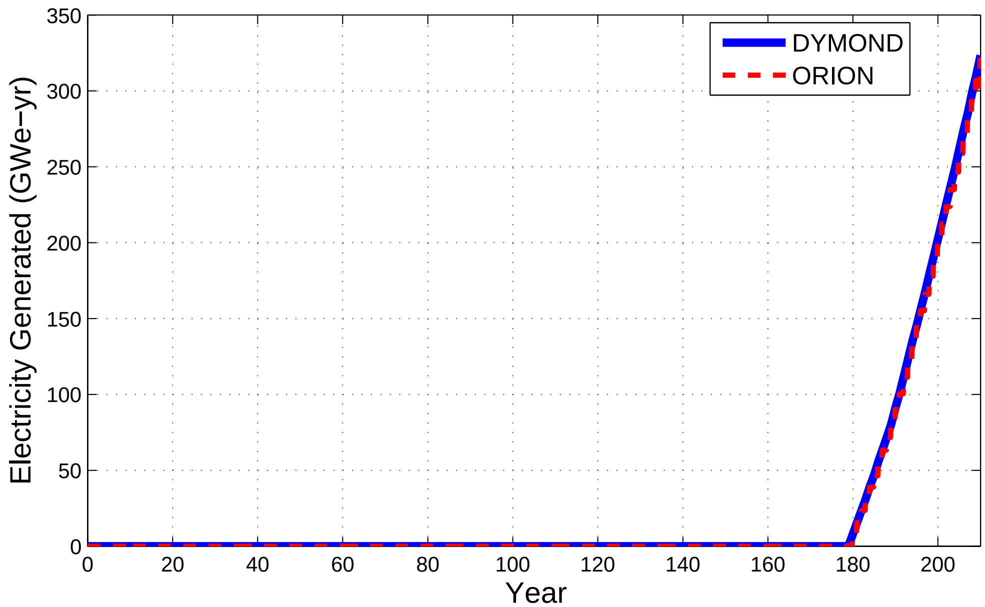

Figure B.4: Power generated by second generation Sodium-cooled Fast Reactors in EG01-EG23 transition scenarion in ORION Case 1

B.2 ORION Transition Output - Case 2

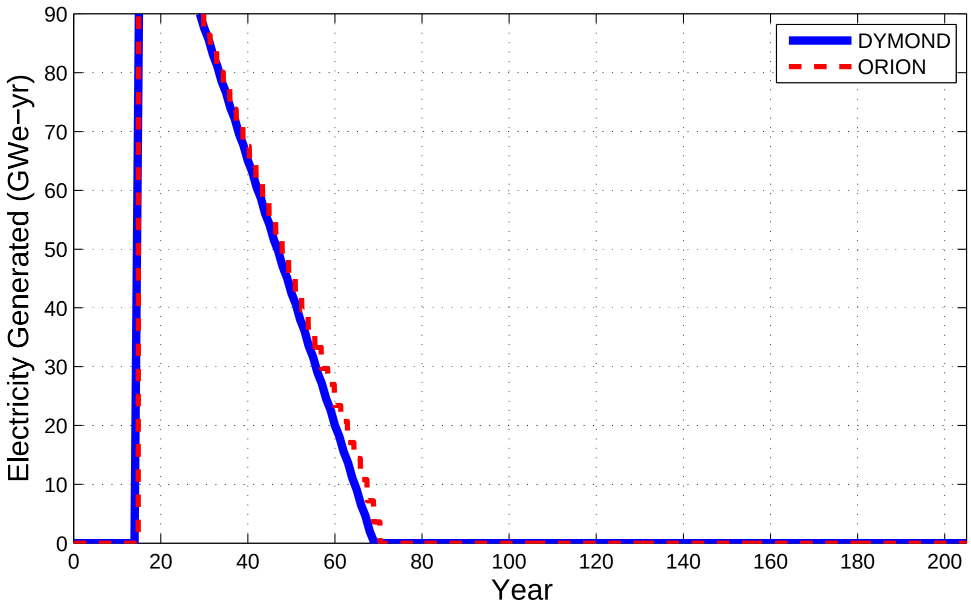

Figure B.5: Power generated by legacy Light Water Reactors in EG01-EG23 transition scenarion in ORION Case 2

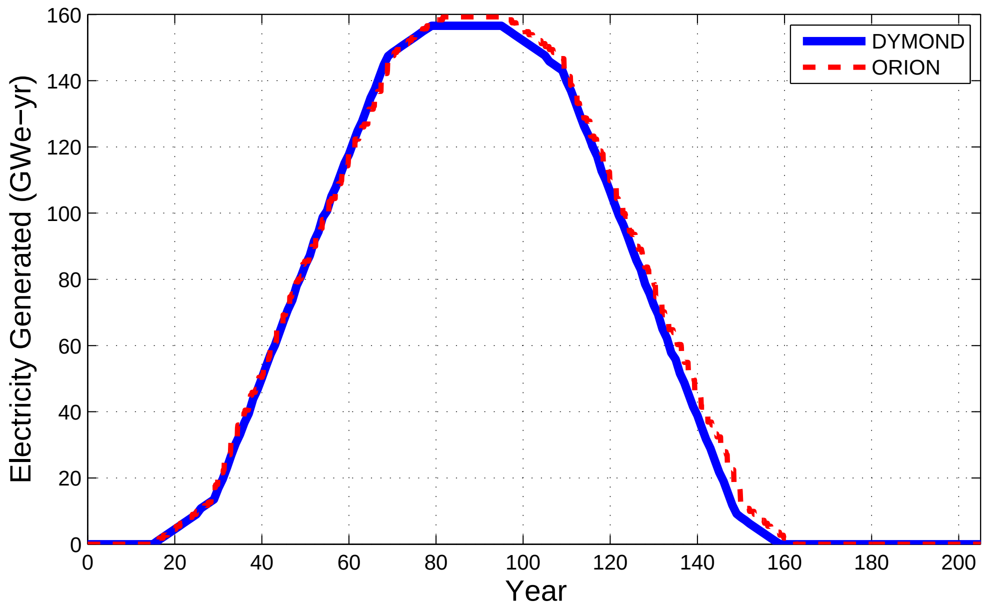

Figure B.6: Power generated by new Light Water Reactors in EG01-EG23 transition scenarion in ORION Case 2

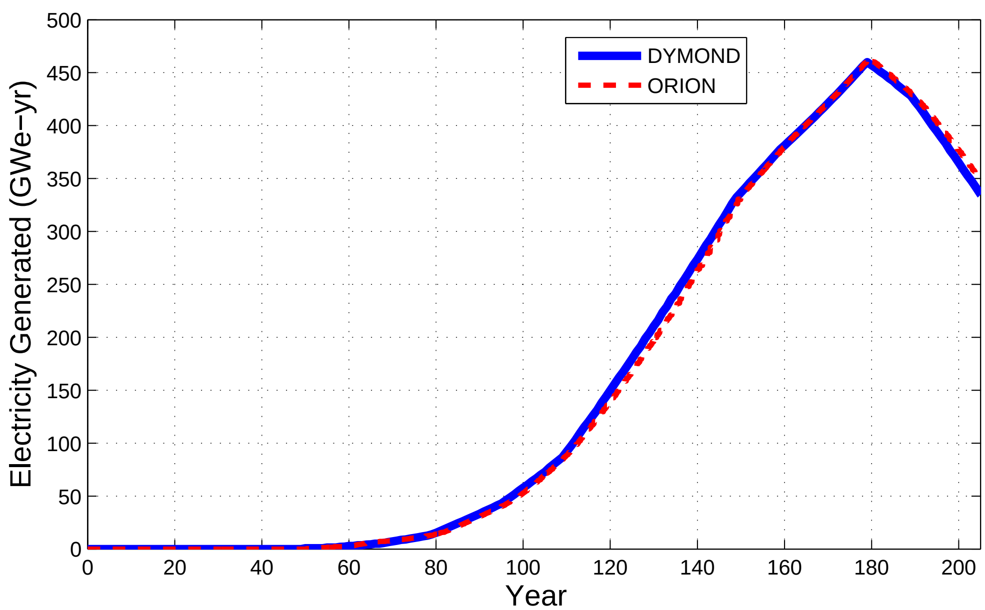

Figure B.7: Power generated by first generation Sodium-cooled Fast Reactors in EG01-EG23 transition scenarion in ORION Case 2

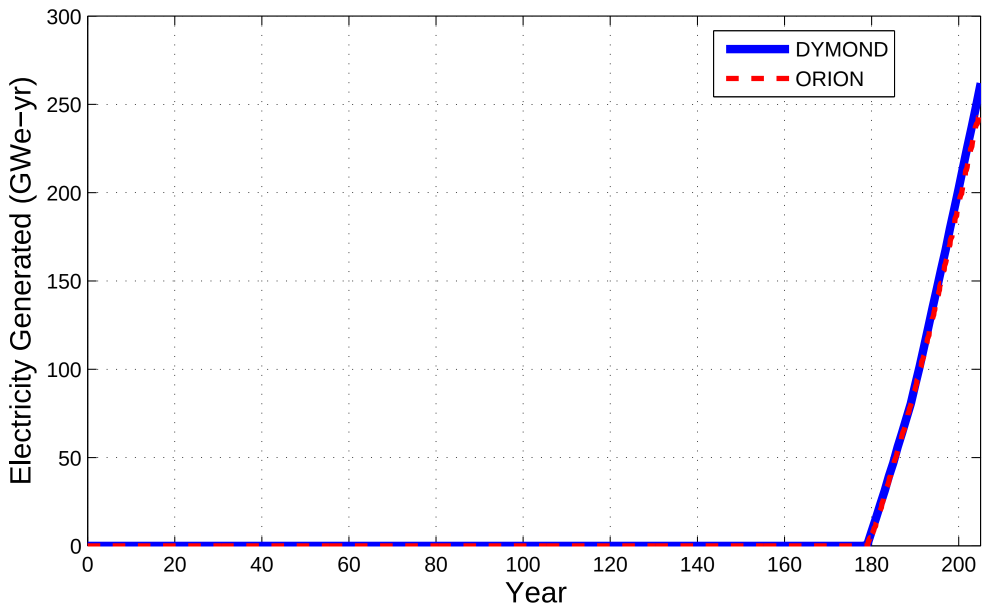

Figure B.8: Power generated by second generation Sodium-cooled Fast Reactors in EG01-EG23 transition scenarion in ORION Case 2

Vita

Jennifer Littell was born in Tracy City, TN. She attended Jasper Adventist school and continued on to Highland Academy in Portland, TN. After graduation in 2009, she attended Southern Adventist University for one year studying Engineering Science. Jennifer transferred to the University of Tennessee, Knoxville in 2010 (UTK) to pursue a degree in Nuclear Engineering with a concentration in Radiological Engineering. During her time as an undergraduate at UTK she worked as an Undergraduate Research Assistant under Dr. Eric Lukosi. She finished her Bachelor of Science in Nuclear Engineering degree in May 2014 and determined to stay at UTK for graduate school. She completed her graduate research under Dr. Steven Skutnik at UTK, as well as working for Andy Worrall at Oak Ridge National Laboratory. Jennifer plans to graduate with a Master of Science Degree in May 2015 and pursue a career which will allow her to contribute to the nuclear community with the skills she has learned throughout her education and research opportunities.