DRAFT Next Generation Fuel Cycle Simulator Functions and Requirements Document

Fuel Cycle Research & Development

Prepared for
U.S. Department of Energy
Systems Analysis
Kathryn Huff
Brent Dixon
Idaho National Laboratory
July 31, 2010
FCRD-SYSA-2010-000110

Approved by :

National Technical Director, Systems Analysis

Kathy McCarthy Date

Federal Technical Manager, Systems Analysis

Brad Williams Date

ACKNOWLEDGEMENTS

This document is a result of contributions from the following:

Lori Braase, Idaho National Lab

Brent Dixon, Idaho National Laboratory

Jess Gehin, Oak Ridge National Lab

Ed Hoffman, Argonne National Lab

Kathrine Huff, Idaho National Laboratory

Temitope Taiwo, Argonne National Lab

Abdellatif Yacout, Argonne National Lab

The authors also consulted with a number of other individuals who provided important insights, including developers and maintainers of other fuel cycle simulation tools.

CONTENTS

FIGURES

TABLES

This page intentionally left blank.

EXECUTIVE SUMMARY

This report describes the major functions and initial requirements of the Advanced Fuel Cycle Simulator (FCS). The FCS is a next generation fuel cycle analysis tool proposed as the Grand Challenge for the Systems Analysis Campaign of the Department of Energy's nuclear fuel cycle technology program. The FCS will dynamically simulate the commercial nuclear fuel cycle at the discrete facility level with isotopic resolution. It seeks to be a versatile, open-platform effort, supporting user and development experiences at levels of detail appropriate for a broad and varied user and developer base interested in a range of analysis types.

The document identifies analysis missions and specifies the technical and functional requirements of a comprehensive, top-level fuel cycle simulation code. The report also proposes a programmatic design to compliment currently available fuel cycle simulation capabilities. The FCS is intended to provide a supportive framework for research across national, institutional, and disciplinary boundaries. Intended users include government laboratories, universities, and industry groups.

The current document is the initial development of the philosophy, primary functions, and initial requirements for the FCS. During this document's development, it became obvious that this is just the first iteration of many that will be needed to flesh out the primary features of FCS, determine the initial design, and engage a developer and user community. The reader should consider all statements in this document to be subject to change as the effort goes forward.

The next step will be to vet this document with stakeholders to refine the FCS features and mission requirements. Key stakeholder discussions will include DOE, and fuel cycle analysts at national laboratories, NEUP universities, and international organizations. An advisory group composed of stakeholders will be formed to provide more specific direction in further development of the functions and requirements.

Development of the FCS can then proceed, beginning with choosing the FCS framework and proceeding to design of the program core structures.

This page intentionally left blank.

ACRONYMS

ANSI	American National Standards Institute
ANL	Argonne National Laboratory
COTS	Commercial Off-The-Shelf (software)
DOE-NE	Department of Energy, Office of Nuclear Energy
FCR&D	Fuel Cycle Research and Development program
FCS	Fuel Cycle Simulator
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
INL	Idaho National Laboratory
LDRD	Laboratory Directed Research and Development
MIT	Massachusetts Institute of Technology
NRC	Nuclear Regulatory Administration
ORNL	Oak Ridge National Lab
PC	Personal Computer
PIC	Program Information Collection System (DOE)
PRPP	Proliferation Resistance and Physical Protection
PUREX	Plutonium-Uranium Extraction
PWR	Pressurized Water Reactor
SDD	Software Design Description
SESAME	Simulation Enabled Safeguards Assessment Methodology
SINEMA	Simulation Institute for Nuclear Energy Modeling & Analysis
SNF	Spent Nuclear Fuel
SPMP	Software Project Management Plan
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
SRS	Software Requirements Specification
STP	Software Test Plan
UREX	Uranium Extraction
V&V	Verification and Validation

DRAFT NEXT GENERATION FUEL CYCLE SIMULATOR FUNCTIONS & REQUIREMENTS DOCUMENT

1. INTRODUCTION

This document contains a high level description and initial requirements of the proposed Advanced Fuel Cycle Simulator (FCS), a next-generation fuel cycle simulation tool proposed to be developed for the Systems Analysis Campaign of the Department of Energy, Nuclear Energy (DOE-NE) Fuel Cycle Research and Development (FCR&D) program.

The current document is the initial development of the philosophy, primary functions, and initial requirements for the FCS. During this document's development, it became obvious that this is just the first iteration of many that will be needed to flesh out the primary features of FCS, determine the initial design, and engage a developer and user community. The reader should consider all statements in this document to be subject to change as the effort goes forward.

This document will begin with an exposition of the need for an advanced fuel cycle simulator and its place among the world's existing fuel cycle tools. Thereafter, this document will touch upon some unique aspects of the FCS concept such as a focus on extensibility and modularity, structural support for uncertainty and sensitivity analyses, and an open source platform to encourage development and collaboration. Finally, the document will detail (Section 6) the analysis and research missions that the FCS intends to support as well as the requirements driven by those mission goals (Appendix A).

1.1 Background

In FY-2009, the campaigns in the FCR&D program were each asked to identify a "Grand Challenge" within their area which, if achieved, would provide a revolutionary advancement in their respective area of research. The Systems Analysis campaign provides technical assessments of proposed fuel cycles and the consequences of various implementation scenarios. Simulation of the nuclear fuel cycle is central to the campaign's mission to "perform integrating analyses of nuclear energy and fuel cycle systems to inform fuel cycle R&D, programmatic decisions, strategy formulation, and policy development". The Grand Challenge for the Systems Analysis Campaign is to:

Develop a fuel cycle simulator as part of a suite of tools to support decision-making, communication, and education that synthesizes and visually explains the multiple attributes of potential fuel cycles. [8]

The FCS should be complementary to existing tools, at least in the near-term. One issue with existing fuel cycle simulators is their technical nature which makes communication of results difficult. Another is the complexity of codes, which makes it difficult for developers to maintain and augment them and for non-experts to use them. The underlying software architecture of the simulator also impacts the ease of implementing different types of analyses.

The FCR&D program has an existing fleet-level simulator (VISION [35]), as do the national nuclear fuel cycle research programs of several other countries (see Current Fuel Cycle Simulation Tools section). These simulators are typically complex, tracking mass at the isotopic level, and are focused on the areas of concern for the sponsoring programs. They primarily focus on the back-end of the fuel cycle and support analyses of advanced fuel cycles with different levels of recycling. Universities also have fuel cycle simulators of varying complexity, usually focused on research for advanced fuel cycles or advanced reactors, and usually developed and extended by graduate students under faculty leadership. The fuel cycle simulation of industry is quite different, typically focusing on the front end of the current once-through uranium-based fuel cycle, assessing economic decisions related to commercial operation.

The approach for the FCS is to use an architecture and software development approach that will support additional classes of analyses while improving flexibility and extensibility, reducing software development and maintenance costs, and encompassing the needs of government and university research, education and communication. One specific emphasis is to provide extensive flexibility in assessing and communicating the performance of fuel cycles to support decision-making and FCR&D program focusing by DOE-NE.

1.2 Document Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the high level functions and preliminary requirements for the Next Generation Nuclear Fuel Cycle Simulator. The SRS follows the format suggested in the ANSI/IEEE 830-1998 software engineering standard [18]. This SRS seeks to:

- Describe the functionality and potential uses of the Nuclear Fuel Cycle Simulator concept to stakeholders
- Provide the foundation for more detailed requirements development.
- Initiate collaboration among developers and users to design the system.
- Identify the pathway for development of the FCS, including the methodology for Verification and Validation.
- Provide a basis for developer use during system design.

This report is a living document which will be expanded and extended as the requirements for the FCS evolve. The current draft addresses the general objectives, functions, and approach along with an initial set of more specific requirements which will evolve as the FCS effort is focused by stakeholder feedback.

1.3 FCS Scope

The FCS will dynamically simulate the commercial nuclear fuel cycle at the discrete facility level with isotopic resolution. It seeks to be a versatile, open-platform effort, supporting user and development experiences at levels of detail appropriate for a broad and varied user and developer base interested in a range of analysis types.

2. FCS VISION

As indicated in the introduction, the FCS is the Systems Analysis Campaign Grand Challenge. The history and current state of fuel cycle analysis was an important consideration in formulating the FCS concept.

Some equivalent of fuel cycle analysis has been in existence since the beginning of the military use of nuclear energy. It has always been constrained by the human and computer resources required to conduct calculations. As computers have advanced, this constraint has evolved. We are now able to perform massive calculations within a given area, such as is performed in reactor physics codes. The primary feature of these calculations is the parallel solution of the same equation multiple times. However, the ability to link areas of calculation to model their interactions is still limited. This linkage is a key feature of the fuel cycle problem, because everything interacts with everything else. Thus the problem isn't one of solving the same equation multiple times, but solving the interaction of many different equations with their associated parameters, assumptions, and uncertainties.

Because the nature of the fuel cycle problem is different, we are considering a different architecture for the FCS. The object-based architecture is designed to support the interaction of multiple types of items, including representations of both physical objects (e.g. a reactor) and conceptual objects (e.g. a policy), including feedback loops that can provide for a dynamic, self directing system (if properly designed).

A second issue is the complexity of the fuel cycle problem, and its impact on the human/computer interface. All fuel cycle codes are limited in their scope by these complexity limits – the more detailed the treatment areas modeled by a code (depth), the more limited the number of areas included (breadth). Attempts to model both depth and breadth result in huge numbers of parameters that must be specified for an analysis, long run times on large computers, and massive amounts of output data that must be sorted through to find any meaningful insights resulting from the effort. Simplified models suffer from the opposite result; they require limited input, run quickly, and the analysis results are usually more accessible, but the limited detail usually misses key interactions and the results are often not just inaccurate but also misleading.

It is not clear how to overcome all of the problems associated with the complexity issue. The proposed approach for FCS is to employ modularity such that the used can specify the level of detail to use in modeling each area based on the problem they are assessing. For example, a user interested in the waste impacts of recycling may select detailed modules for separations facilities, waste forms, and disposal system performance, while using simple modules for the fuel cycle front end and reactors, and excluding from the analysis capabilities for modeling regional impacts, grid stability, etc.

Again, the object-based architecture should support this approach. Any object can be specialized to model it in more detail when needed. When not needed, the specialization code for that object is not loaded. Similarly, additional attributes of object behavior can be loaded as needed – so if an analysis does not require cost calculations then the code that adds an object's cost-related parameters and algorithms is not loaded.

While computers have made huge gains in capability, humans have not. One shortcoming of all fuel cycle codes is the user interface. Many of the nuclear number crunching codes still are based on the equivalent of card deck input and data table output. The fuel cycle codes typically are not much more advanced, and due to their complexity they are often limited to use only by experts or college students able to invest weeks in learning how to use them properly. The purpose of performing fuel cycle analyses is to improve our understanding of how they work, including how all the interactions affect the overall behavior and performance. This information is important not only for the researcher, but also the manager, the policy maker, the undergraduate student, etc.

Clear communication of fuel cycle behavior and analysis results is one of the primary objectives of the FCS. Part of the approach is to provide a standard set of output graphics to facilitate comparisons between different fuel cycle options, deployment assumptions, etc. Another aspect is user customizable output, where standard graphic layouts can be used to display different details of an analysis and allow the analyst to emphasize these details. Other features are also being considered such as the ability to partially "rewind" an analysis, make parameter changes, then advance the analysis again and study the impact, as well as animation of material flows, transportation links, etc.

To improve communication and learning from the code, we desire the FCS interface and data libraries to support a range of users, including users with little training who want to run simulations without specifying large numbers of parameters. The FCS must be able to prepare a simulation run by loading predefined modules and scenarios, then execute that run, including permutations requested by the user without breaking down or providing nonsensical results. This requires significant control logic within the code to make infrastructure and operational decisions, similar to the policy and business decisions that human decision makers would follow. This becomes more difficult as the number of contributing factors and options increases. Designing and applying different control logics will be an important capability to support policy analysis. The desire to support a range of users with different levels of modeling complexity also generates a requirement to support multiple hardware platforms, from PC laptops to UNIX workstations, and potentially even massively parallel supercomputers.

The other area of limitation with respect to human capital is in the development and maintenance of codes. Currently, every organization with a fuel cycle program has their own code, which results in significant duplicate development and maintenance effort. In addition, many universities have simplified codes directed by professors and developed by students that include new ideas and approaches to fuel cycle modeling, but typically never fully mature. The human capital of graduate student researchers in particular is a valuable, economical resource for code development that is inefficiently utilized. One other outcome of the proliferation of codes is that there is no "industry standard" code with a large user base. Large user bases drive improved interfaces, improved documentation, better bug identification and eradication, broader input on priorities for capability improvements, and numerous other benefits.

We have taken the idea of a Grand Challenge to heart with respect to FCS development and maintenance, proposing to use an open source approach where anyone interested in fuel cycle modeling could contribute. This approach has been very successful in other areas of software development, and would result in a revolutionary change in the fuel cycle analysis landscape. This vision includes full collaboration between the DOE fuel cycle modeling effort and that of other governments, universities, and industry, creating both user and developer communities of sufficient size to establish the FCS as the reference code used by all. However, such an approach requires avoiding any export restricted data/algorithms and waving of intellectual property rights by all participants. Discussions with additional stakeholders are needed to determine the feasibility of this approach.

Even with collaborative development, the FCS will still require significant resources. The envisioned path forward is to establish the core design and initial models while identifying the developer community. Once a critical mass of FCS capability is achieved, the developer community should blossom with rapid grow of both additional FCS capabilities and FCS users.

3. CURRENT FUEL CYCLE SIMULATION TOOLS

The FCS is intended to be a supplement to current nuclear fuel cycle simulation codes and FCR&D systems analysis tools in the near term. If fully successful, the FCS will become the industry standard tool, replacing many of the current fuel cycle codes. The FCS seeks to employ lessons learned with current and past fuel cycle simulation tools. In the near term, the FCS aims to comprehensively provide a coherent superset of the functionalities individually provided by the fuel cycle codes in existence, and in the long term to provide functionality where gaps in existing capabilities have been identified [20][22]. This report has reviewed the functions of the current available fuel cycle simulation codes listed in Table 1.

Table 1 : Fuel cycle simulation codes considered in this report.

Name	Developer	Reference(s)
CAFCA	MIT	[12]
COSI6	CEA (France)	[3][11][12][14]
DANESS	ANL	[6][12][14]
DESAE2.1	Rosatom (Russia)	[1][3][14]
EVOLCODE2	CIEMAT (Spain)	[3]
FAMILY21	JAEA (Japan)	[3]
GENIUSv1	INL	[23]
GENIUSv2	Univ. of Wisconsin	[23]
NFCSS	IAEA	[14]
NFCSim	LANL	[14]
VISION	ANL / INL	[3][12][35]

Fuel cycle codes currently in existence are capable of a range of functionalities necessary for the FCS to incorporate (e.g. multi-regional analysis, detailed isotopic tracking, cost analysis, etc.). However, while all of these codes are capable of supporting at least a subset of the range of fuel cycle analyses which are performed within government, universities, and industry, none supports the superset of these analyses. An individual code typically performs detailed analysis of a few particular fuel cycle areas, reporting detailed indicators for those aspects while neglecting others. For example, some codes that focus on economics neglect detailed isotopic tracking, while others carefully track and calculate waste disposal metrics while neglecting cost-analysis and multi-regional interactions. Furthermore, while it may seem that linking these codes to each other could provide a tool with an aggregate set of their capabilities, a self-consistent tool would not result from such a linkage. Some particular gaps of interest are summarized below.

3.1 Discrete Modeling

The ability to model discrete facilities and materials is captured to varying extent by several current fuel cycle simulation tools such as COSI6, DESAE2.1, FAMILY21, GENIUS, VISION and NFCSim. Discrete modeling of facilities enables a range of analyses, and discrete modeling is preferred when the number of facilities of a particular type is limited. As the number increases, it is preferable for some types of analysis to lump the facilities into fleets and model only the fleet behavior.

While the question of the relevance of modeling discrete materials is a subject of open debate, the FCS will be developed with a discrete architecture in order to perform a greater range of analyses. The basis for this position is the experience that many phenomena of advanced fuel cycles have system-level aggregate effects which can only be captured by discrete material tracking. Such phenomena include anisotropic fresh fuel fabrication requirements, spent fuel isotopics of a fleet of reactors with greatly varying burnups and the potentially chaotic dynamics of isotopic balance in fuel cycles involving multiple recycling passes.

3.2 Disruption Analysis

The ability to model Disruptions is captured at some level by most codes capable of tracking the operations status of discrete facilities. That is, such codes will at least alert the user to insufficient feedstock scenarios created by upstream disruptions in the fuel cycle. However, Disruption analysis also requires the specific capacity to model likelihood of facility process disruptions and graceful facility responses to insufficient feedstock (responses such as shutting down, reducing production capacity, or reordering appropriate feedstock). NFCSim, for example, discretely models facilities and materials, but in the event of an input scenario that is not self-consistent facilities fail to gracefully handle disruptions. Scenarios that involve insufficient feedstock simply end the simulation with an error [29]. DESAE, on the other hand does not force shutdown due to insufficient feedstock, and in the event of insufficient fissile material during reprocessing, borrows it from storage, leaving a negative inventory value [3]. The capability to gracefully model fuel cycle upsets due to insufficient feedstock has been achieved with GENIUSv2 at the University of Wisconsin, but is limited in its ability to model Disruptions due to other factors or define availability factors for individual facilities [23].

3.3 Open Access

Governmental restrictions on export of potentially sensitive information, the proprietary concerns of research institutions and security constraints of data within fuel cycle codes often restrict code access. A code's use is therefore often limited to its institution of origin, necessitating effort duplication at other institutions. License agreements are required for many current codes (i.e. COSI6, DANESS, DESAE, EVOLCODE, FAMILY, and NFCSIM) [20]. For this reason, the MIT code, CAFCA provides open access to source code, and a stated priority of GENIUS and VISION design was license-free code availability. Other codes such as COSI and DANESS are available only by license. The planned approach for FCS is to provide fully free and open access to all users and developers, foreign and domestic.

3.4 Uncertainty and Sensitivity Automation

While uncertainty and sensitivity analysis of input parameters can be conducted at a medium-fidelity level by manual parametric sampling with almost any fuel cycle code [26] [33], automated sampling and propagation of uncertainties and parametric sensitivity has not been achieved by any of the fuel cycle codes discussed here. Furthermore, uncertainty distributions for included data and propagation algorithms are lacking. Many of the missions that the FCS seeks to support would be facilitated by such a capability and this need constitutes a driving factor in the development of the FCS concept.

3.5 Unconventional Reactor Modeling

Much fuel cycle simulation focus in the recent decade has been driven by specific fuel cycle policy goals. As a result, most fuel cycle models have neglected to model myriad unconventional fuel cycles and reactor models on the research stage today. The FCS seeks to offer unconstrained modeling freedom for researchers interested in the impact of unconventional fuel cycle and transmutation technologies and uses of reactor heat production on fuel cycle metrics. This flexibility may also allow modeling and analysis of non-electric uses of process heat such as for syngas production, water desalination or hydrogen production.

3.6 Dynamic Feedback for Economic Analysis

In addition to recording cost information about deployment and material movements, the simulation logic of the FCS will have the capability to simulate the dynamic responses within the system to material and process pricing, material availability, repository restrictions, etc. By allowing deployment and material routing logic to be assigned dynamically, the FCS will be capable of running in a dynamic mode in which facility deployment and material routing decisions are determined during runtime by the state of the system according to rules and constraints specified by the simulation logic and input parameters. This capability is an advanced functionality and its development will be user-driven. Currently, few codes have the ability to dynamically adjust parameters based on economic feedback and in the few cases where it is implemented, it is limited to only a portion of the fuel cycle.

4. FCS REQUIREMENTS

The FCS is intended to support the analysis needs of a wide user community. These needs generically include the ability to define a fuel cycle or fuel cycle transition scenario, assess the characteristics, behavior, and performance of the defined system, communicate the results obtained, and archive the input and output data associated with such a simulation. The implementation of these generic capabilities varies depending on the user and will require flexible performance adaptable to personal laptops and PCs, research workstations, and highly parallel computing structures alike. Furthermore, the FCS must be capable of supporting a corresponding range of user-driven simulation complexity.

Numerous specific parameters must be identified to define a fuel cycle or scenario. An expert user will need the ability to specify each of these parameters individually, while an average user will need the ability to select pre-set groups of parameters. This will require at least two different user interfaces, one for the "mechanic" who is working at the individual parameter level to manage the simulator, and one for the person desiring simply to "drive" a simulation. To support educational purposes, both in academia and for non-technical users, the simulator interface may need to be even further simplified to automate as much of the scenario building process as possible.

The FCS will need to perform a standard set of analyses across a range of mission areas. These analyses will determine the operational behavior of the scenario. The analysis must also include calculation of a standard set of performance measures. Section 7 details the range of missions the FCS should support.

Communication of simulator results can take on several forms. The Grand Challenge description includes synthesizing and visually explaining the multiple attributes of a fuel cycle system. One required capability of the simulator output is a standard set of graphs and tables that summarize key attributes, display key behavior, and provide a standard set of performance assessment results. The simulator should also provide 'restart capability'. By recording the full state of the simulation during runtime, the FCS may be paused at a point during that simulation and restarted from the simulation state file at will. This functionality could be utilized either as a feature of the sensitivity/uncertainty analysis engine, as a way to re-tune the behavior of a simulation, or to make comparative analyses of similar simulations with shared initial conditions. Finally, some simulators provide animation to help viewers better understand how the simulator works – animation requirements for the FCS have not been explored at this time.

The capability to archive simulator data is necessary for many of the same reasons as archives of the input and output data are essential for any post-simulation analysis. These include repeatability, independent verification, and reuse of simulation results. A fuel cycle simulator will generate considerable internal data during an analysis, including detailed information at multiple points in time and isotopic details. If it can be regenerated, some of this data is not required to be archived as long as repeatability remains possible. Also, the user may not require that the data be reported at the most detailed level, but instead could choose that the output be summarized (e.g. standard fuel cycle performance parameters). However, the lowest level of detail of all parameters must be accessible to the expert user during a simulator run.

The FCS will require an extensive set of libraries to support its functionality. These libraries will be a resource of information useful for purposes other than simulations. The user interface must support and facilitate access to this information.

5. FCS FUNCTIONS

The Fuel Cycle Simulator will be capable of multiple types of analysis, will possess an architecture that facilitates highly customizable use and development, performs its functions on multiple platforms, and models all the basic facilities of the nuclear fuel cycle. Figure 1 depicts the analysis, operational, performance and modeling functions that the FCS seeks to provide.

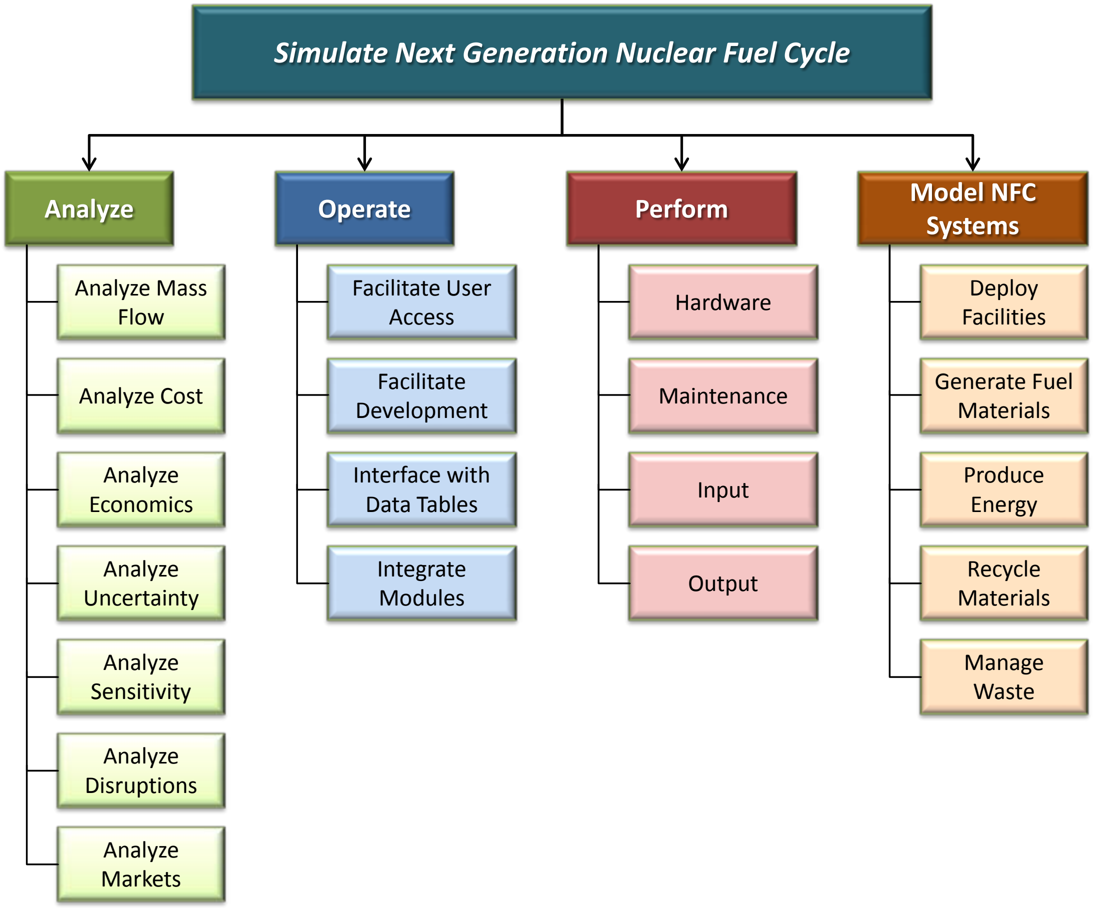

Figure 1 – Analysis, operational, performance, and modeling functions of the FCS

5.1 Analysis Functions

5.1.1 Mass Flow Analysis

The FCS will track an arbitrary number[a] of isotopes and isotope groups of potential interest in the nuclear fuel cycle in order to provide a basis to analyze fuel cycle options according to a large number of metrics [24]. Fuel cycle metrics that rely on mass flow range from simple inventories to decay heat, radiotoxicity, proliferation resistance and physical protection (PRPP) indices. Mass flow analysis will require that the FCS provide libraries of fuel composition, transmutation, separations, and waste form data, among other material data sets.

[a] For example, VISION currently tracts 81 different isotopes and isotope groups

5.1.2 Cost Analysis

The FCS will track capital, operating and other costs fundamental to fuel cycle cost analysis. Analyses of the cost competitiveness of nuclear technologies rely on comparison of cost metrics of fuel cycle options and analogous metrics for other energy sectors [31]. Cost metrics include levelized cost of electricity, total fuel cycle cost, and facility life cycle costs.

5.1.3 Economic Analysis

Facility deployments, power production, material availability, market forces, and numerous other parameters will inform material and process price effects and allow dynamic pricing and feedback simulation behavior.

5.1.4 Sensitivity Analysis

The FCS will provide a platform capable of automating input parameter perturbation and output optimization to facilitate the analysis of the magnitude of impact of an input parameter on a metric or metrics of interest in the fuel cycle.

5.1.5 Uncertainty Analysis

The FCS will provide the flexibility to replace basic data within a fuel cycle simulation with uncertainty distributions for the purpose of propagating uncertainty through the simulation. This will inform and support the legitimacy of focusing decisions and approximating assumptions.

5.1.6 Disruption Analysis

The FCS will support reliability, policy, safety, and strategic analyses with a capability to simulate disruptions of the fuel cycle and respond gracefully to uncertainty in facility availability, construction times, process accuracy, and capacity. Analysis of unexpected facility shutdown and process behavior will be necessary to inform fuel cycle robustness studies of many types.

5.1.7 Market Analysis

The FCS will at least provide the market level business logic necessary to determine deployment rates and to route material transfers between individual facilities. Customization of this logic by developers and users allows analysis of market mechanisms integral to the evolution of infrastructure and the operational movements of materials in the nuclear fuel cycle.

5.2 Operational Functions

5.2.1 Facilitation of User Access

The FCS will follow an open-source, free-software distribution paradigm. Free and open access is achieved by providing a stable distribution of the source code and current documentation via an open access online hosting platform. This unencumbered access to source code ensures transparency in the same way that access to current documentation eases use. Ease of access exposes the code to students, researchers, and citizens who benefit from a useful platform on which to conduct their analyses and who contribute by reporting bugs, critiquing code behavior, and generating valuable simulation results. Furthermore, public bug reporting, code review, and version control drives community cohesion and invites collaboration with a broad user and developer base that encourages code stability and reduces rigidity. While some users and developers may incorporate transmutation calculations or physics databases that have distribution constraints, the official stable release version of the code will contain only functions and data that have no such constraints.

5.2.2 Facilitation of Development

The FCS development and maintenance paradigm will also follow this open-source, free-software model. In this model, a complete version history of the source code and documentation is made available to developers via an open access online hosting platform. In addition to the bug reports, code behavior critiques, and simulation results also contributed by users, developers contribute code extensions and bug fixes. In this way, the user community is empowered and can drive development toward meeting mission targets relevant to its diverse research goals. Furthermore, public bug reporting, code review, and version control drives community cohesion and invites collaboration. Finally, inviting such verification and validation contributions from a broad user and developer base encourages code stability and reduces rigidity.

5.2.3 Integration of Independent Data Tables

Some simulations will require modeling one or more fuel cycle subsystems with external high precision tools or customized process data tables. FCS will support incorporation of detailed results from physics and process models through integration of independent data tables. In this way, code linkage with reactor physics codes can be conducted by supplying an FCS reactor model with fresh and used isotopic vectors from the output of core physics simulations such ORIGEN or CASMO [27]. Thus, the runtime burden associated with direct linkage to modules that dynamically call the more detailed calculations can be avoided by providing such data tables and perhaps appropriately interpolating between them.

5.2.4 Integration of Modules

In situations when tabulating detailed process data is infeasible or inadequate, the FCS must be capable of integrating the customized process models of advanced users and developers. For example, dynamic linking of storage modeling codes such as SCANS 1A [10] and REFREP [13] may be necessary since storage times are unknown before runtime. While this type of linkage will affect the execution time and memory requirements of the FCS, interfacing with user-specified models and external codes will support facility centric studies, model benchmarking, and extension to new process technologies and simulation behaviors. Shared databases in a cross-platform format such as HDF5 might serve to transport this data between the FCS and these more detailed physics codes.

5.3 Performance Functions

5.3.1 Hardware/Platform

The FCS should run on standalone PCs and laptops with current technology and running Windows (XP or greater), Linux, or UNIX operating platforms. The FCS shall support versions compatible with both laptop-level computational power and highly parallel architectures. The execution and storage requirements of the basic release version intended to be capable of running on a laptop shall not require more RAM or hard disc space than is typical of a personal home computer.

5.3.2 Maintenance

An internal team of maintainers shall maintain a source code repository with a current stable distribution version of the code and a version controlled archive of past versions. Maintainer duties will include, but are not limited to responding to bug reports, incorporating extension modules developed externally, maintaining up to date and accurate documentation and user manuals, and extending functionality to meet the goals detailed in this document.

5.3.3 Input

The FCS will allow input at various levels of detail. It will provide a graphical user interface (GUI) for 'viewers' and 'end-users' as well as access to fully customizable source code for both 'advanced users' and 'developers'. A set of data libraries detailing process data, facility data, and simulation driving data shall be provided in the input paradigm in order to facilitate a level of input specification reminiscent of an input-deck. Advanced users and users with high powered computing resources shall be capable of customizing more parameters than the end-user with the typical PC compatible version of the code. To facilitate this dual compatibility, input data at different levels of detail can be made available.

The input format shall be complemented by a comprehensive system of input checking and informative input-error exceptions in order to suggest appropriate ranges for input values and warn users against inappropriate values.

5.3.4 Output

The user should have the freedom to save the output database with a name of their choosing and in a location they specify. A typical output database generated by the end-user PC compatible version of the code shall not exceed typical PC hard disc sizes. The output data shall be in a format that allows browsing. The output data should be available at various levels of detail, providing viewers and end-users with automated graphs, tables or reports and providing advanced users and developers with access to a raw data format compatible with developer-driven scripted database analysis. Automated graphs and tables will be compliant with the requirements in this document.

5.4 Modeling Functions

For full modularity, each facility involved in the FCS must share a universal interface, so that facility modules may behave as interchangeable and substitutable black boxes. Basic facility modules and simulation behavior necessary to simulate once through, modified-open, or closed fuel cycles of interest to users of the FCS will be provided with the stable end-user release.

5.4.1 Deployment of Facilities

Nuclear fuel cycle facilities in the simulation will be deployed according to an electricity demand growth rate, a more complicated demand curve or an explicit deployment schedule specified over the length of the simulation time. The capability to model fuel cycle facilities in existence at the beginning of the simulation time as well as the capability to account for facility commissioning and decommissioning time will be necessary parts of the FCS facility deployment logic.

5.4.2 Generation of Fuel Materials

While the front end of a nuclear fuel cycle may involve many combinations and permutations of natural and recycled resource materials, conditioning processes, and fabrication schemes, a standard set of fuel cycle facilities capable of modeling mining, milling, conversion, enrichment, and fuel fabrication will be necessary for fundamental once through and some closed or modified-open fuel cycle simulations.

5.4.2.1 Mining and Milling of Natural Resources

A basic mining facility must be available which allows the user to choose the amount of finite fissile resources available to the simulation, region, or facility. The FCS must support simulations with multiple mines and capacities. To support uranium fuel cycles, the FCS should provide preconfigured optimistic and pessimistic estimates of the current state of world uranium resources. In the case of uranium, this facility will produce milled U_3O_8 “yellowcake” as well as mill tailings requiring proper disposal.

Optional modules to supplement this model may include additional individual mines and mills with an array or process times, pricing structures, and deployment schedules reflective of the array of grades and extraction methods necessary to access the world’s various ore resources (e.g. high/low-grade ore bodies, in situ mining, or seawater extraction).

5.4.2.2 Conversion

A basic conversion facility must be available in which the chemical form of the fissile material object produced by the milling process is converted into a chemical form appropriate for future fuel fabrication. In the case of uranium, for example, U_3O_8 is converted into UF_6. No isotopic changes occur in the Conversion facility.

Possible future modules to supplement this one may employ specific process details. For example, a user may provide process specifics modeling the sizing, reduction, hydrofluorination, fluorination, and distillation steps of the dry fluoride volatility conversion process in the U.S. or other processes being used around the world. Such specificity can inform studies of fuel cycle economics, material losses, low level and hazardous-non-radioactive waste generation, etc.

5.4.2.3 Uranium Enrichment

A basic enrichment facility will transform the isotope vector and chemical form of a milled natural uranium material object into enriched UF6. The basic FCS enrichment model must be capable of producing any enrichment level at a rate no more than its user-specified separative work unit (SWU) capacity, according to some form of the SWU equation at least as detailed as the equation below.

(1)

Where S is the capacity in SWUs, P is product mass, W is waste mass, F is feedstock mass, x_p is the product assay, x_w is the tails assay, and x_f is the feedstock assay.

Optional models to supplement this model may include more complex calculations of SWUs and tails fractions, an economically driven model that optimizes tails enrichment based on the costs of yellowcake, conversion, and SWUs, as well as models that incorporate tails re-enrichment or process-specific behavior associated with various enrichment technologies (e.g. gaseous, centrifugal, or laser). An alternate module would allow for attainment of low enriched uranium from alternate sources such as high enriched uranium down blending.

5.4.2.4 Fuel Fabrication

A basic fuel fabrication facility model must accept enriched UF_6 provided by the Enrichment facility, or thorium provided by the mining and milling facility, and produce finished fuel assemblies at an enrichment that agrees with the received UF_6. Also, a fuel fabrication facility model using recycled materials should be able to accept separated streams of used fuel and construct viable fuel assemblies for supported reactors. A fundamental subset of reactor technologies for which the FCS should have the capability to construct viable fuel is to be established.

Optional models to supplement this model may include fuel fabrication according to fresh fuel compositions associated with additional reactor types (e.g. CANDUs, thorium reactors, etc.) and alternate fuel forms (e.g. TRISO).

5.4.3 Power Production

For power production, a variety of reactor models must be available to the end-user in the stable release distribution. Fundamental reactor types must be flexibly supported at various burnups and cycle times of interest. While in current simulation codes this is typically accomplished with exogenously calculated fresh and used pairs of isotopic 'recipes,' other models that efficiently succeed at runtime burnup calculations are encouraged [28] Also, a simple recipe-based reactor model must be available which can be customized to accept any recipe pair of fresh and discharge isotopic vectors specified by the user.

Optional extensions to this model include efficient on-the-fly transmutation approximation models, and models of new advanced reactor designs such as small modular reactors, advanced fast reactors, pebble bed reactors, or molten salt reactors. Sub-critical transmutation systems also fall within this area.

5.4.4 Material Recycling

In order to recycle used nuclear materials in the fuel cycle (e.g. depleted uranium tails or irradiated nuclear fuel), a separations facility and appropriate fuel fabrication facilities must be available. For recycling of other materials in the fuel cycle (zirconium, for example) other facilities must be provided by developer extensions.

The separations model must be capable of applying an array of separation stream matrix calculations to a used fuel isotopic vector and assigning appropriate chemical forms to a set of separated streams. This separations matrix must be customizable by the user. However, the basic model must also provide preconfigured separation matrices capable of modeling a fundamental set of aqueous, mechanical, electrochemical, and pyrolitic processes with user-specified separations efficiencies.

Supplementary models may include separations models which are driven by material demands of downstream advanced reactors, models involving more process details, etc.

5.4.5 Waste Management

Basic models of both dry and wet storage, incorporating time dependent material transmutation due to decay should be available. These models must include waste characterization, classification, waste package characteristics, and repository models of various geological settings (e.g. salt, tuff, or granite) and incorporating hydro-geological nuclide transport and decay heat effects in those environments.

6. FCS CHARACTERISTICS

6.1 Purpose

This section describes the general characteristics of the FCS, including types of users and range of modeling levels.

6.2 User Characteristics

The FCS will provide an interface appropriate for users with a broad range of computational sophistication. Target audiences include researchers at national labs, students, professors, industry representatives, policy makers, technical advisors and industry.

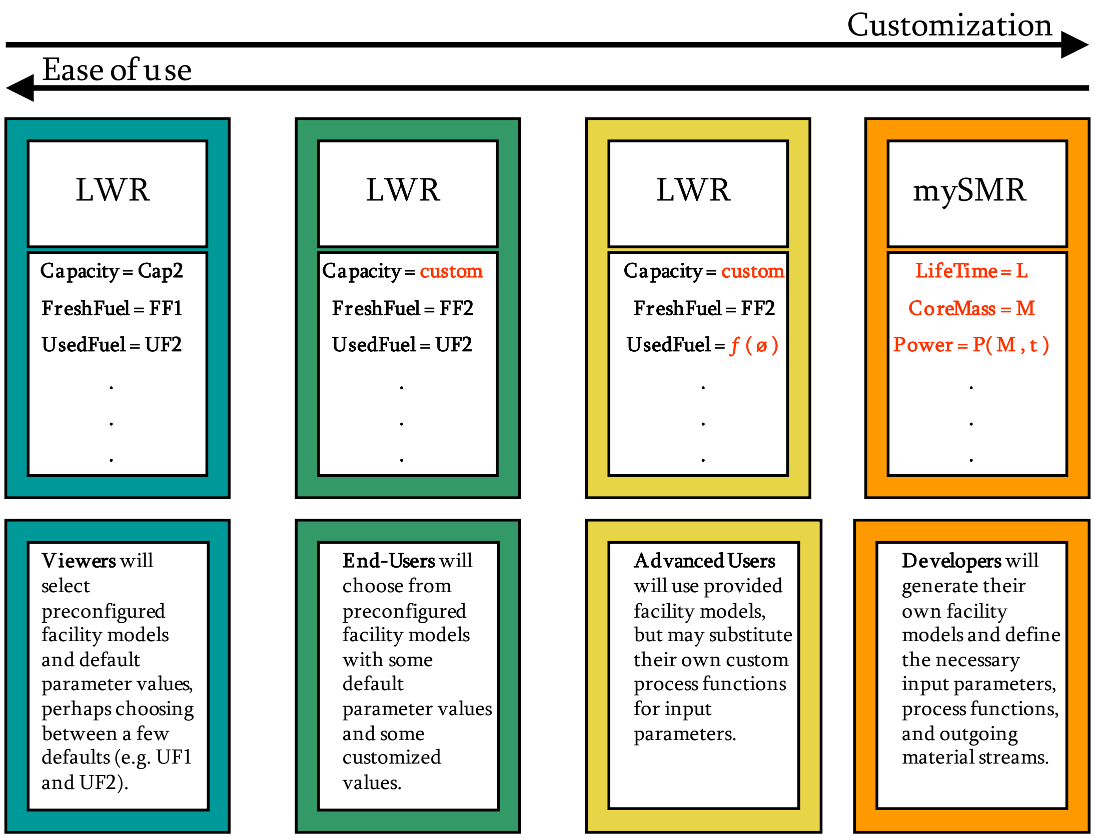

Figure 2 – Degree of customization and ease of use by user type

6.2.1 Viewers

Potential viewers of the FCS will be individuals who may or may not directly interface with the code but to whom output of the FCS may be made available for reasons of education, advocacy, and demonstration. These viewers include university students, political lobbyists, policy makers, industry leaders, and members of the general public. Since viewers will not necessarily have either a notion of the facilities that make up a nuclear fuel cycle or the concept of using computer simulations, an input/output paradigm compatible with timely, transparent, informative results is necessary for service to this group.

6.2.2 End-Users

End-users of the FCS may include additional members of the academic, scientific, commercial and general public with only a limited need for simulation customization and detail. End-users will possess a fundamental notion of the facilities that make up a nuclear fuel cycle, and will typically be familiar with engineering analysis via computer simulations. However, unlike developers, end-users have an ordinary level of computational sophistication and will rely primarily on validated data libraries and facility models provided by the verified distribution version of the FCS. End-users will have the freedom to select process behaviors from an array of facility models and specify facility and simulation parameters. While they will have the freedom to specify some aspects of simulation behavior, end-users will not modify facility process behavior or data provided to individual facility models. Such users will be interested in a stable distribution version of the FCS.

6.2.3 Advanced Users

Advanced users will be similar to end-users but will have greater need for simulation customization and detail. Advanced users such as research scientists, graduate students, engineers, and policy advisors both foreign and domestic will possess detailed knowledge of the nuclear fuel cycle and familiarity with engineering analysis via computer simulations. Potentially interested in simulations modeling the fuel cycle impacts of new technologies and systems, advanced users will have the freedom to select from a library of facility models as well as the freedom to customize those models to create new facility types within an existing facility class (e.g. a new waste form or new repository type) by specifying facility process functions and simulation data. While most advanced users will be interested in a stable distribution version of the FCS, others may be interested in a modifiable version when they are ready to develop and contribute their own extended modules. Some advanced users may volunteer to beta test the stability of new versions of the FCS prior to their formal release.

6.2.4 Developers

Potential developers include individuals and research groups from the academic, scientific, open-source, and general public interested in contributing through the open source portal. Developers will be computationally sophisticated and will possess a solid understanding of the concept of computational systems analysis. Developer contributions may include verification and validation, reporting bugs, resolving issues, and developing modular extensions. Groups of developers at national labs, universities, and other research institutions may also be involved in proposing and committing substantive changes to the FCS such as defining new classes of facilities or new performance parameters. Code changes and extensions will first be submitted to the internal group of maintainers for review and will be incorporated if they are deemed to improve the code and have been determined to satisfy FCS quality standards.

6.2.5 Code Maintainers

Maintainers of the Fuel Cycle Simulator include the core group of scientists, engineers, students, and analysts at the organization(s) directly tasked with maintaining the simulator and directing development. These developers will contribute to initial development, verification and validation, reporting and resolving issues, adding modular extensions, maintaining code revisions and documentation, and moderating external code comments from developers. Code changes and extensions reviewed by the maintainers will be incorporated if they are deemed to improve the code and have been determined to satisfy the FCS quality standards.

6.3 Depth and Breadth

6.3.1 Simulation Detail

The FCS will need to facilitate simulations specified with technical and behavioral detail at the global, national, interregional, institutional, and facility levels, encapsulating the movement and transmutation of nuclear materials from their geological extraction to their final emplacement in a repository.

By defining the modular pieces of a fuel cycle (i.e. process data, facilities, regions, institutions, and simulation modes), the user may create the simulation from the bottom up, as with building blocks. The modular architecture of the FCS will allow the user to choose and combine flexible modules (blocks) in an unconstrained manner, then to tie them together into a coherent fuel cycle by applying constraints to their behavior. Meanwhile, the FCS will facilitate many levels of user sophistication by allowing abstraction of details for the viewer and end-user while preserving the capability for advanced users and developers to alter input and output parameters as much as possible.

The FCS will support macroscopic analyses at global and national scales by encapsulating the dynamics of multiple simultaneously interacting facilities and regional fuel cycles in cumulative, system-level performance metrics. These top level analyses describe system dynamics across regions and institutions and allow the investigation of relevant policy issues at the international, multinational and national scales. Macroscopic analyses illuminate the effects of multinational waste management agreements, regional variations of electrical grid restrictions, and state level renewable portfolio standards. Since such top-level strategic comparisons are relatively insensitive to individual facility parameters, details can be approximated by default values for a simplified user experience. As shown in Figure 3, groups of preconfigured facilities can be presented to the user as ‘Front End’ or ‘Reprocessing’ modules, for example.

The FCS will also be capable of modeling multiple, simultaneously interacting non-geographical entities such as utilities or groups of nations that determine the behavior and deployment of fleets of facilities. One or more simultaneously interacting facilities may be associated with such an entity, and the deployment and behavior of these facilities may be constrained by aggregate group-wide parameters such as tax structures, capital availability, trade relationships, and technological capability. This modeling ability allows studies of synergistic institutional relationships and policy impacts of multiple fleet deployment decisions on long time scales. These studies answer questions such as where, how many, and which combinations of types of facilities result in improved fuel cycle metrics on the institutional level.

The FCS will support detailed module-level analysis as well, which takes place at a facility scale. Parametric facility and deployment comparisons, sensitivity analyses, and uncertainty analyses of institutional or regional level decisions take place at this level. At this level, the user will focus on the effects of facility and deployment performance parameters such as capacity, fuel types, process specifics, and demand growth rates. These studies answer questions such as what capacity, process methods, and facility types result in improved fuel cycle metrics for a certain fuel cycle strategy.

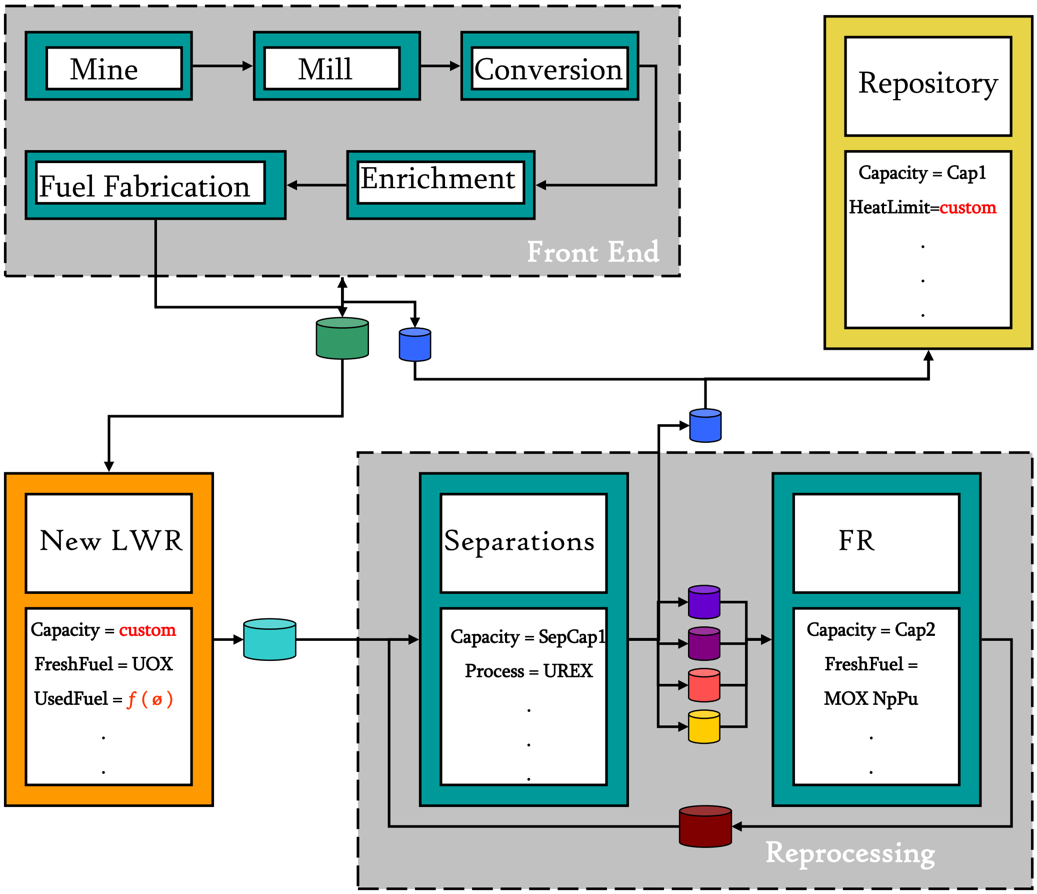

Figure 3 – Use of modularity to model different portions of the fuel cycle at different levels of detail

6.3.2 Output Breadth

The FCS output data shall be in a format that allows browsing and should be available at various levels of detail, providing viewers and end-users with automated graphs, tables, or reports (see [28]) and providing developers with access to a raw data format compatible with developer-driven database analysis. The automated set of output graphs and tables are not intended to be a comprehensive replacement for the output database but are instead intended to streamline the presentation of output metrics of common interest. Automated graphs and tables will be compliant with the requirements in this document and will support a broad range of analysis purposes. Fundamental information in the output database will include time histories of material flows at isotopic resolution, facility deployment and operating histories with all relevant facility parameters, facility capacities and capacity factors, facility capital and operating costs, material prices, calculated fuel cycle costs and calculated levelized costs of electricity for each fuel cycle and region and the data relevant to FCS.

The main unit of information in the FCS is material movement, so material flows will be automatically generated. Some graphs of interest will include material present in the system as a function of time, as well as transfers of material between facilities and between regions. These shall be customizable, allowing display of behavior such as isotope specific material flows, facility specific material flows, and regional flows. Transport of material between regions may be displayed in a variety of methods, such as on maps with arrows whose thickness indicate trade frequency, mass transferred, mass of a certain isotope transferred, etc. Finally, a table of materials, their compositions over time, and their movements between facilities will be a part of the output database.

Another important output of the FCS will be facility deployment, so graphs and tables indicating infrastructure changes such as construction of facilities by type, technology and region should be generated. A table of facilities and their parameters including all of the above data will be a part of the output database.

Finally, cost and economic information comprise a key set of fuel cycle metrics, so the FCS must be capable of generating output graphs and tables that detail cash flows vs. time for each facility and region. Material prices over time as well as calculations of fuel cycle costs and levelized costs of electricity for particular facilities and regions may also be tabulated as a part of the output database.

7. SUPPORTED MISSIONS

As the industry standard fuel cycle analysis tool, the FCS will need to support the range of studies performed by the interdisciplinary user community. The FCS will also need to supply access to a variety of input parameters and output metrics in support of these studies. This section provides a categorization of the types of studies to be supported, along with a description of how the different levels of users may interact with the FCS for each type.

An initial list of specific “use cases” is provided within each study area, supporting the initiation of development of the next level of detail of the FCS functions and requirements. Because these use cases require specific functionality (parameters, data, equations, interface features, etc.), they are the drivers for specific requirements of the FCS. The use cases are referenced by their identifying numbers in the basis statements of the more detailed requirements in Appendix A.

7.1 Facility Studies

Facility studies test the system effects of deploying a specific facility (or process, in the case of transportation for example). The capability to conduct facility studies supports the analysis of new technologies, the system-level aspects of their introduction to fuel cycles, new mathematical models of facility process dynamics, and other factors that may focus research and development goals and inform policies to incentivize first movers in new technologies.

- The FCS will support the viewer who relies on pre-configured facilities and deployment scenarios, merely substituting one predefined technology for another (e.g. trading PWRs for BWRs) in order to make comparisons between those facilities. The FCS will provide a library of interchangeable facility configurations and scenario specifications for user selection.
- The FCS will allow end-user specification of the parametric configuration of supplied facility models. This will require input error checking to constrain the user to appropriate allowed parameter ranges.
- The FCS will allow advanced user customization of process behaviors in supplied facility models. Input error checking is not possible for this use case, so the user will be expected to constrain their input to appropriate functions and values.
- The FCS will allow substitution of developer-generated facility models. That is, the developer will supply all specified databases and process information for the new facility model, but the FCS will supply databases and process information for other facilities that surround it to complete the fuel cycle. The FCS will also possess an architecture capable of interfacing with new facility models.

Initial Use Cases

The FCS will conduct investigation and analysis of:

F.1 Metrics resulting from deployment of various transmutation technologies

F.2 The “technical and economic aspects of external neutron source-driven transmutation systems to inform whether future investigation in this approach is warranted.” [7]

F.3 Uranium/thorium resource depletion as a function of facility types, deployment decisions.

F.4 Uranium/thorium utilization of various new once through fuels. [7]

F.5 Direct SNF disposal per MWh from various improved once through fuels.

F.6 Metrics resulting from deployment of various separations schema (UREX, PUREX, etc.)

F.7 Disposal metrics resulting from deployment of various partitioning and transmutation, reprocessing, and storage options.

F.8 Disposal metrics resulting from disposal of civilian and defense used nuclear fuel in "a variety of geologic environments".

F.9 Effects of waste form disposition and packaging on repository and safety metrics.

F.10 Cumulative effects of process losses in partitioning and transmutation.

F.11 Hydrogen generation capacities as a function of facility deployment.

F.12 Other waste heat utilization capacities as a function of facility deployment.

F.13 Transition dynamics for new facility types.

F.14 The need for an advanced fuel research facility and system effects of new fuels and cladding capable of irradiation to higher burnup.

F.15 Safeguards effectiveness against covert fuel cycles.[b]

F.16 Non-Radiological releases from fuel cycle facilities (e.g. SO_2 from Mine/Milling) [14].

7.2 Strategy Studies

This capability allows comparison of strategic options, using creative combinations of many reactor and facility types. These combinations may be strategic options such as once through, thermal recycling, fast reactor recycling, strategic facility redundancy etc. In such use cases, equivalent facility modules must be interchangeable and markets must support material routing capable of upstream material trades (such as the simulation logic of multipass material routing).

- The FCS will support comparison of pre-configured strategic options, with preconfigured facility deployment parameters. Such preconfigured scenarios may include a Business as Usual case, or base NE Roadmap cases. The viewer may run various base cases in order to make comparisons between those strategic deployments. The interface must support comparison of up to 5 different cases.
- The FCS will support end-user and advanced user specification of deployment schedules and facility parameters for preconfigured routing schemes, the details of which are supplied by FCS. For example, the user may want to vary the fast reactor capacities and year of first deployment in a fast reactor recycling scenario, but without perturbing the simulation logic of material routing.
- The FCS will allow substitution of new deployment logic and routing schemes generated by the developer. The developer will supply all specified databases and process information for all new business logic for material trades and routing.

Initial Use Cases

The FCS will enable the strategic investigation, analysis, and demonstration of:

S.1 Holistic impacts of proposed deployment strategies and policy options.

S.2 Uranium resource depletion as a function of demand growth rates.

S.3 Flexibility and constraints on interim storage timing.

b . That is, the FCS will allow comparison of safeguards detection uncertainty windows with the potential magnitude of material flow differences between ordinary fuel cycles and possible parallel shadow fuel cycles in which material is being diverted or covertly generated.

S.4 Flexibility and constraints on reprocessing timing.

S.5 Flexibility and constraints on dry storage timing.

S.6 Flexibility and constraints on repository emplacement or retrieval timing

S.7 The magnitude of alleviated proliferation risk from limited separations cycles [7].

S.8 System effects of thorium fuel cycles and their synergy with existing fuel cycles.

S.9 System effects of non-fuel material re-use (e.g. irradiated zirconium) [7].

S.10 Adaptation requirements for regulatory changes Waste Acceptance Criteria, Classification Bases, etc.)

S.11 Nuclear assisted reduction of fossil fuel use. (e.g. bio-mass processing)

7.3 Economic Studies

The FCS will facilitate analysis of cost and economic fuel cycle dynamics. Here, 'cost analysis' refers to financial information as an output metric only whereas 'economic analysis' refers to the use of costs and pricing as input parameters which affect simulation dynamics and even drive feedback behaviors.

- The FCS will support cost comparison of pre-configured scenarios, using FCS-calculated cost metrics. Such FCS-calculated metrics may include Fuel Cycle Costs, Levelized Cost of Electricity, Total Reactor Capital costs, etc. The viewer will rely on FCS-calculated pricing and cash flow parameters for facilities and materials, including but not limited to uranium pricing models and facility capital and operating costs.
- The FCS will support end user and advanced user specification of economic facility parameters or material pricing information as well as customized calculations of cost metrics. For example, the user may want to vary the capital costs of facilities to observe the effect on the levelized cost of electricity. In a 'cost analysis' this change in facility price will not affect facility deployment or material routing in any way. Due to the lack of feedback, much cost analysis can be conducted as a post-processing function external to the code functionality.
- The FCS will allow the developer to extend the model to incorporate new economic data (e.g. regional taxes or interest rates and capital cost depreciation schedules) in order to make cost calculations.
- The FCS will support economic analysis by being extensible enough to allow appropriately customizable system dynamics in the material routing and market structures. The developer will supply new data models and all new business logic for material trades, while the FCS will supply an architecture that can flexibly support new material routing and facility deployment logic.

Initial Use Cases

The FCS will enable the investigation and analysis of:

E.1 Sensitivity of total fuel cycle cost to individual facility costs

E.2 Sensitivity of total fuel cycle cost to reprocessing system costs

E.3 Sensitivity of total fuel cycle cost to construction and fuel material costs

E.4 Effects of various deployment parameters on levelized cost of electricity.

E.5 Investigate factors (transmutation technology, facility deployment, waste disposition, etc.) affecting transportation, reprocessing, disposal, storage costs

E.6 Economic impact of technologies to extend current fleet lifetimes [7].

E.7 Investigate factors (uranium availability, transmutation technology, deployment schedule, etc.) affecting power production costs.

E.8 Feedback effects and institutional capital limitations of facility costs on facility deployment.

E.9 Feedback dynamics of process costs on facility deployment.

E.10 Feedback dynamics of disposal costs on technology deployment.

E.11 Feedback dynamics of disposal costs on reprocessing timing.

E.12 Feedback dynamics of material pricing on facility deployment.

E.13 Feedback effects of material prices on material routing.

E.14 Feedback effects of trade relationships and material availability on regional technology deployment.

7.4 Multi-Regional Studies

The FCS will facilitate analysis of the nuclear fuel cycle on the global, national, and local scale, resolving real and fictional countries, institutions and their interactions.

- A viewer or end-user may be interested in observing regional variation in fuel cycle metrics for parametric perturbations on the Business as Usual case. Relying primarily on preconfigured scenario parameters, the user may be able to change a single parameter and observe the effect on their region-specific metric of interest (bilateral trade frequency, material availability, fuel cycle robustness, etc.). The FCS will provide a true-to-history base-case representing the history of the nuclear fuel cycle in the world, specific to the world's countries.
- An end-user may be interested in examining the effect of deployment parameters such as electricity demand or regional technology availability on fuel cycle metrics of interest. The user will be capable of defining the deployment parameters for custom regions (fictional or otherwise), as well as choosing the facility types and facility parameters available for deployment within those regions. The regional simulation response variation can then be observed as a function of those deployment constraints (designed to represent policy decisions, international relations scenarios, synergistic fuel cycle options, etc.).
- A developer conducting a multi-regional analysis may be interested in extending the model to incorporate new interregional trade rules to simulate their own political or economic models or incorporate new region specific data (such as location).

Initial Use Cases

The FCS will enable the multi-regional investigation and analysis of:

M.1 Fuel assuredness/reliability/security per region.

M.2 Movement of sensitive nuclear materials between regions.

M.3 Generation of sensitive nuclear materials in each region.

M.4 Investigate implications of international/multi-regional fuel bank scenarios.

M.5 Synergy and dynamics of fuel loaning and take-back schemes.

M.6 Implications of multi-national waste management.

M.7 Trade/technology symbiosis between regions with various indigenous technologies.

M.8 Deployment responses required by regionally differentiated demand growth.

M.9 Employ market and political models (e.g. trade data or the Affinity of Nations Index) to predict/direct interregional material and technology transfers.

M.10 Determine proliferation risk factors (e.g., capabilities and motivations) [7].

M.11 Technology and materials proliferation risks as informed by social science research in international security [7].

M.12 Feedback dynamics of international trade relationships on fuel cycle costs, regional energy security, facility deployment, etc.

M.13 Regional grid load restrictions.

M.14 Regional political deployment restrictions (e.g. state-level moratoria awaiting a federal disposal plan).

M.15 Regional electricity pricing differences (e.g. grid-level pricing dependencies).

M.16 Deployment effects of institutional (e.g. governments, corporations) variations in available capital, technology capability, etc.

7.5 Disruption Studies

A user may want to study the ripple effect (or lack thereof) experienced by a fuel cycle in the event of an unplanned facility shutdown or process disruption.

- A viewer or end-user may be interested in observing fuel cycle responses to disruption for preconfigured facilities and deployment scenarios. Relying primarily on preconfigured scenarios, the user may be able to investigate the relative disruption thresholds of scenarios by observing the effect of upset probability on fuel cycle robustness. The FCS will provide preconfigured base case fuel cycles of interest for this comparison.
- An end-user may be interested in examining the effect of upset in custom deployment scenarios. The user will be capable of defining custom deployments and facility reliability probabilities. The simulation response to disruptions can then be observed in relation to those deployment scenarios.
- A developer may be interested in altering the disruption responses of facilities and material routing logic in order to investigate strategies (e.g. redundancy alternatives, storage and staging, etc.) for improved fuel cycle robustness. The FCS will provide architecture capable of flexibly allowing modifications to facility disruption responses, deployment logic, and material routing schemes.

Initial Use Cases

The FCS will enable the disruption analysis of:

U.1 Weak links in process capacity and timing during transition to new technologies.

U.2 Fuel cycle robustness and power generation for various demand scenarios.

U.3 Effects of facility/process reliability on fuel cycle cost, power production, etc. That is, what are the ramifications of the shutdown of a facility or facilities?

U.4 Comparative benefits and drawbacks for storage and staging strategies and redundant deployment scenarios designed to promote robustness.

U.5 Reliability implications of aging and degradation of system structures and components, (reactor core internals, pressure vessels, building materials, pipes, cables etc.)

U.6 Sudden changes in resource availability or price.

U.7 Sudden shutdown of centralized reprocessing facilities.

8. PATH FORWARD

This document has provided documentation of the vision and development philosophy of the Fuel Cycle Simulator, along with initial definition of its basic functions and requirements.

The next step will be to vet this document with stakeholders to refine the FCS features. These vetting sessions are also expected to help confirm the viability of the development approach. Appendix B includes minutes of such a vetting session with internal stakeholders, illustrating the level of discussion expected.

Key stakeholder discussions will include DOE, and fuel cycle analysts at national laboratories, NEUP universities, and international organizations. An advisory group composed of stakeholders will be formed to provide more specific direction in further development of the functions and requirements. Other key enabling discussions will include legal guidance on intellectual property and export control issues.

The first step in development of the FCS will be choosing the FCS framework, including whether to confirm the use of an object-based architecture or to use some other approach and whether to pursue an open source development path. Input to this decision will involve the stakeholder advisory group along with consultation with software development experts.

Completion of the above activities will enable iteration on this functions and requirements document, involving the following items:

- Developing a more complete and consistent set of use cases to focus the mission requirements.
- Firming up user types and general performance expectations
- Transforming the mission requirements into requirements for object types, analysis algorithms, data, and user interface features
- Identification of requirements for the information backbone which will provide the universal interface that modules will plug into, including minimum data sets for modules and requirements for database development

The functions and requirements iteration will enable development of the more detailed architecture design, including the object class structure and generic modules. It will also enable selection of software for coding of FCS, including database selection. These selections will rely heavily on software experts.

At this point, the basic software framework/backbone will be developed, documentation standards and V&V procedures established and the collaborative environment for the developer community will be created.

Developers will then proceed with design and construction of the initial working version of the FCS, including:

- Development of basic modules for each stage of the fuel cycle. More detailed/specialized modules will be developed after these basic modules are in place
- Development of facility modules for each major technology in each stage (e.g. gaseous diffusion, centrifuge, and laser enrichment)
- Front end GUI development to support the range of users
- Flexible back end GUI development to support the range of module output information
- Construction of core data libraries

With the completion and V&V of FCS version 1.0, the collaborative environment for the user community will be set up and the code released to the users.

9. REFERENCES

[1] E A Andrianova, V D Davidenko, and V F Tsibul'skii. *DESAE Program for Systems Studies of Long-Term Growth of Nuclear Power*, Atomic Energy, Vol. 105, No. 6, (2008).

[2] C G Bathke and E A Schneider, *NFCSim User's Manual*, 1–80.

[3] L Boucher, A Schwenk-Ferrero, F Alvarez Velarde, E Gonzalez, B W Dixon, G Edwards, G Dick, K Ono, H Aït Abderrahim, NEA/NSC/WPFC, *Expert group on fuel cycle transitions scenarios Benchmark on Scenario codes.* March, 2010.

[4] G Palmiotti, M Salvatores, G Aliberti. *Validation of Simulation Codes for Future Systems: Motivations, Approach, and the Role of Nuclear Data.* NEMEA-4 Neutron Measurements, Evaluations and Applications, October (2008).

[5] B W Dixon, B Halsey, S H Kim, G E Matthern, S J Piet, and D E Shropshire, *Dynamic Systems Analysis Report for Nuclear Fuel Recycle*, INL/EXT08-15201, Idaho National Laboratory (2008).

[6] L G Van Den Durpel, A M Yacout, and D C Wade, D. C. *Status on developments and applications of the integrated nuclear energy system code DANESS.* Transactions of the American Nuclear Society, vol. 96, (2007), 212–14.

[7] United States Department of Energy. Nuclear Energy Research and Development Roadmap. DOE-NE, http://www.ne.doe.gov/pdfFiles/NuclearEnergy_Roadmap_Final.pdf. April 2010.

[8] United States Department of Energy. *Systems Analysis Campaign Implementation Plan*, DOE-NE Office of Fuel Cycle Technologies, March 31, 2010.

[9] E Gamma, R Helm, R Johnson, and J Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley, Reading, Mass., 1995.

[10] M A Gerhard, SCANS 1A. Shipping Cask Design Review Analysis, Lawrence Livermore National Lab, CA, (1992).

[11] G Grasso, S Monti, and M Sumini, NEA-WPFC/FCTS Benchmark for Fuel Cycle Scenarios Study with COSI6. (2009).

[12] L Guerin, L Boucher, L Van Den Durpel, B W Dixon, *A Benchmark Study of Computer Codes for System Analysis of the Nuclear Fuel Cycle.* Massachusetts Institute of Technology and Commissariat a l ´Energie Atomique, April 2009.

[13] A Hautojärvi and T Vieno, *REFREP: A near-field model for a spent fuel repository*, VTT Publications Register http://cgi.vtt.fi/progs/inf/OAI, Finland, (1988).

[14] International Atomic Energy Agency (IAEA) Guidance for the Application of an Assessment Methodology for Innovative Nuclear Energy Systems INPRO Manual-Environment, IAEA-TECDOC-1575 /Vol. 7. (2007). 1–169.

[15] International Atomic Energy Agency (IAEA), Energy, Electricity and Nuclear Power Estimates for the Period up to 2030, IAEA-RDS-1, IAEA, Vienna (2005).

[16] International Atomic Energy Agency (IAEA). Power Reactor Information System, PRIS (*http://www.iaea.org/programme/a2/*).

[17] ANSI/IEEE Std 762-1987(R2002), IEEE Standard Definitions for Use in reporting Electric Generating Unit Reliability, Availability and Productivity.

[18] IEEE (1998). IEEE Recommended Practice for Software Requirements Specifications. IEEE Std. 830-1998.

[19] J J Jacobson, Leonard Malczynski, and Vincent Tidwell, *Very Large System Dynamics Models Lessons Learned.* Idaho National Lab. 26th International Conference of the System Dynamics Society, (2008).

[20] C A Juchau, M L Dunzik-Gougar, and J J Jacobson, *Modeling the Nuclear Fuel Cycle.* Nuclear Technology. Vol. 171 (2010), 136-141.

[21] G. Kloss and A. Schreiber. Provenance implementation in a scientific simulation environment. In L. Moreau and I. Foster, editors, LNCS: Proceedings of the International Provenance and Annotation Workshop (IPAW'06), volume 4145. Springer-Verlag, 2006.

[22] A Miron, J Valentine, J Christenson, M Hawwari, S Bhatt, ML Dunzik-Gougar, M Lineberry. *Identification and Analysis of Critical Gaps in Nuclear Fuel Cycle Codes Required by the SINEMA Program.* DOE/ID/14839. (2009).

[23] K M Oliver, P P H Wilson, A Reveillere, T W Ahn, K Dunn, K Huff, and R Elmore, “Studying International Fuel Cycle Robustness with the GENIUSv2 Discrete Facilities and Materials Fuel Cycle Analysis Tool,” presented at Int. Conf. GLOBAL 2009, Paris, France, September 6–11, 2009.

[24] S J Piet, “Selection of Isotopes and Elements for Fuel Cycle Analysis,” Proceedings of Advances In Nuclear Fuel Management IV (ANFM-IV), Hilton Head, South Carolina, April 12-15, 2009, INL/CON-08-15050, (2009)

[25] S J Piet, G Matthern, J J Jacobson et al. Fuel Cycle Scenario, Definition, and Trade Offs. INL/EXT-06-11683. http://www.inl.gov/technicalpublications/Documents/3382970.pdf. Accessed June 18, 2010.

[26] J Preston, G Sweder, T Anderson, S Janson, M Humberstone, and J Clark, Nuclear Energy Research Initiative (NERI) Program Grant Number DE-FC07-051D14658 Uncertainty Analyses of Advanced Fuel Cycle, (2008).

[27] J Rhodes, K Smith, and D Lee, *CASMO-5 Development and Applications*, PHYSOR-2006, ANS Topical Meeting on Reactor Physics. Vancouver, BC, Canada. September 10-14 (2006).

[28] E A Schneider, Charles G Bathke, Michael R James, Los Alamos Criticality Engine (LACE) and *NFCSIM : A Dynamic Fuel Burnup and Fuel Cycle Simulation Tool*, Nuclear Technology (2004), 35–50.

[29] E A Schneider, Private Communication June 05, 2010.

[30] D E Shropshire, K A Williams, W B Boore, J D Smith, B W Dixon, M L Dunzik-Gougar, R Adams, D Gombert, and E Schneider, *Advanced Fuel Cycle Cost Basis*, Idaho National Laboratory, ID, INL/EXT-08- 11536, (2007).

[31] D E Shropshire, *Advanced Fuel Cycle Economic Tools, Algorithms, and Methodologies*, Idaho National Laboratory, ID, INL/EXT-09-15483 (2009).

[32] R M Stallman, and the Free Software Foundation. "Various Licenses and Comments about Them - GPL-Compatible Free Software Licenses". http://www.gnu.org/licenses/license-list.html. Retrieved 2010-06-14.

[33] T E Stover, Quantification of Back- End Nuclear Fuel Cycle Metrics Uncertainties Due to Cross Sections, (2007).

[34] S L Turner, Discrete Modeling : OCRWM Total System Model, (2010).

[35] A M Yacout, J J Jacobson, G E Matthern, S J Piet, D E Shropshire, C T Laws, "VISION – Verifiable Fuel Cycle Simulation of Nuclear Fuel Cycle Dynamics," *Waste Management 2006*, February 26-March 2, 2006, Tucson, Arizona.

Appendix A

TECHNICAL AND FUNCTIONAL REQUIREMENTS

This page intentionally left blank.

Appendix A - TECHNICAL AND FUNCTIONAL REQUIREMENTS

This appendix contains the initial level of technical and functional requirements for the FCS. The format used it to indicate the requirement area and subarea, then list individual requirements and their bases. The basis statements end with a list of cross-references to the high level functions in the body of the main document. These include the use cases of Section 6, as well as other functions. Use cases of potential developers and users drive the requirements of the Fuel Cycle Simulator. In addition to the overarching Analysis, Operational, Performance, and Modeling Functions, these use cases define bases for the specific requirements of the FCS.

Since the proposed audience for the FCS pursues a broad range of simulation results and relies on a variety of analysis methods, the requirements for the FCS are broad. The breadth demonstrated by the requirements that follow in this appendix seeks to be a covering set, providing a spanning foundation on which to further develop specifications as development moves forward. However, some requirements listed in this early stage of program development may have been overlooked, poorly conceived, or stated too prescriptively here. Progress on this project will treat this appendix as a living document, removing requirements as they become unnecessary and incorporating new requirements as they present themselves.

Technical requirements are listed first. These requirements address how the software is to be developed and maintained. They include the architecture, open source platform, documentation, verification and validation, user interface, and coupling with other analysis tools. Next come the functional requirements, which address what the software is to do. They include mass flow analysis, cost analysis, market analysis, sensitivity analysis, uncertainty analysis, and disruption analysis. The last section lists facility data requirements.

A-1. Architecture

A-1.1 Modularity

A-1.1.1 The FCS architecture will isolate the functionality dictating simulation behavior from functionality dictating facility modules.

Basis: Modular architecture that isolates the trunk of the code from extending modules eases development. This encapsulation allows the developer to incorporate new content without any need to understand or modify the core code behavior. This discourages fragility. Since the code core is unaffected by module additions, facility module additions by the developer do not compromise core simulation behavior. [F1, F2, F5-F8, F11-F16, S8-S11, M4, M5, M9-M11, U4, Operational Functions 5.2.2, 5.2.3 and 5.2.4]

A-1.1.2 Facility modules that model equivalent fuel cycle tasks (e.g. a laser enrichment model and a centrifuge enrichment model) must be interchangeable such that the users and developers may effectively treat the building blocks of a fuel cycle scenario as black boxes.

Basis: The flexibility of choosing any conceivable combination of available fuel cycle facility models provides user freedom in scenario specification. [F1, F2, F5-F8, F11-F16, S8-S11, M4, M5, M9-M11, U4, Operational Functions 5.2.2, 5.2.3]

A-1.1.3 The FCS will possess an object-based architecture with discrete facilities and materials.

Basis: By taking advantage of the hierarchical structure of object-based code architecture, the FCS will model discrete material objects and facilities in a manner that more realistically mimics reality. Where fleet-based models have the potential to lump and obscure fuel cycle details for the viewer, discrete models provide a transparent and straightforward imitation of reality, with well-defined facilities and individual material objects. A discrete architecture will complement existing fleet-based simulatorsPrior efforts with object-based fuel cycle simulation design (GENIUS and NFCSim [2][23][28]) indicate that taking advantage of class inheritance structures in object based design lends itself well to the hierarchical structure of information in fuel cycle simulations. [F10, F15, F16, E13, E14, M2-M5, U1-U5]

A-1.2 Dynamic Simulation Behavior

A-1.2.1 The FCS shall have the capability to perform both dynamic calculations and equilibrium calculations for mass flows and facility behaviors.

Basis: Dynamic analysis allows greater realism and accuracy than static analysis [5][25].

A-2. Open Source Platform

A-2.1 User Accessibility

A-2.1.1 The FCS source code repository shall be mirrored in an open access online hosting platform.

Basis: Open, free access on the internet to the code and its revision history ensures transparency, encourages use, and supports research done by students, researchers, and citizens. Ease of access exposes the code to new users who contribute by reporting bugs, critiquing code behavior, and generating valuable simulation results. A broad user base encourages code stability. Potential open access hosting platforms include Google Code, Launchpad, and Bit Bucket, to name a few. [Operational Functions 5.2.1 and 5.2.2]

A-2.1.2 Any official user distribution version of FCS must comply with Validation and Verification criteria listed in this document.

Basis: While the code may be distributed in its pre-validation state to developers contributing to content and validation, viewers, end-users, and advanced users should be provided only code that has been vetted, is deemed to be stable, and can be trusted to perform as expected. [Operational Function 5.2.1]

A-2.1.3 The FCS shall maintain current (compatible with the latest stable distribution version) documentation online.

Basis: Current, accessible documentation facilitates code use, review, and transparency. [Operational Functions 5.2.1, 5.2.2, Performance Function 5.3.2]

A-2.1.4 *All data, comments, and documentation in the public distribution must be DOE export control compliant.*

Basis: The FCS will be accessible by all interested people, foreign and domestic. [Operational Functions 5.2.1, 5.2.2]

A-2.2 Developer Accessibility

A-2.2.1 *The FCS development repository, consisting of all source, configuration, and build files, shall be hosted at an open access, online location under version control.*

Basis: Open, free access on the internet to the code and its revision history ensures transparency and encourages development by students, researchers, and the public. A version controlled repository is essential for development. [F1, F2, F5-F8, F11-F16, S8-S11, M4, M5, M9-M11, U4, Operational Function 5.2.2, Performance Function 5.3.2]

A-2.2.2 *The FCS will have a standardized comment format in the source code that is informative for each module, function, and data element of importance.*

Basis: In large, distributed development software development projects, standardized code comments facilitate inter-developer communication, increase transparency of code functionality, and enable automated code documentation generation. [F1, F2, F5-F8, F11-F16, S8-S11, M4, M5, M9-M11, U4, Operational Functions 5.2.1, 5.2.2, 5.2.3, 5.2.4, Performance Function 5.3.2]

A-2.2.3 *The FCS will facilitate online bug reporting with a publicly available bug report and issue resolution history.*

Basis: Online, public bug reporting focuses development on areas of weakness, and reduces the maintenance burden by inviting contribution from a broad external developer base with a wide variety of problem solving skills. [Operational Functions 5.2.1 and 5.2.2, Performance Function 5.3.2]

A-2.2.4 *A unit test suite shall be developed in parallel with code developments and module extensions.*

Basis: That is, each function or module unit of the FCS should have one or more associated behavioral tests executed during the configuring/compiling step of building the executable. These tests assert the accuracy of each function for every build, assisting validation and verification during development. An example unit test might be as small as checking that a summation function takes the two integers 1 and 1 and returns the integer 2. More complicated unit tests might check that a copying method results in two identical objects or that performing a decay function on a mass of some isotope results in an expected population of the expected daughters and that the resultant mass of the products is equal to the original mass within the target mass conservation tolerance (and accounting for alpha decay). [F1, F2, F5-F8, F11-F16, S8-S11, M4, M5, M9-M11, U4, Operational Functions 5.2.1, 5.2.2, and 5.2.4, Performance Function 5.3.2]

A-2.2.5 *An online forum for communication about development collaboration shall be established.*

Basis: A forum for communication about development collaboration is necessary for cohesive development efforts. A developer listhost or wiki page online for distribution of communications concerning code changes, protocols, progress, goals, and milestones is desirable. [Operational Functions 5.2.1 and 5.2.2, Performance Function 5.3.2]

A-2.3 Open Source License Compatibility

A-2.3.1 The FCS will be licensed under a free and open source license appropriate for free and universal access and distribution. The New Free BSD license is a good example of such a license [32].

Basis: Open source access exposes the code to new developers and advanced users who contribute by generating content, driving design, and reporting bugs. The broad developer base made possible by free and open source access discourages code rigidity. The New Free BSD license is a good example of such a license [32]. For clarity, "copyleftness"[32] is not a necessary quality of this license. However, necessary qualities of this license include four main freedoms. The FCS license should preserve the freedom to run the program, for any purpose. The FCS license should preserve the freedom to study and alter how the program works. (Access to source code is a necessary precondition for this.) The FCS license should preserve the freedom to redistribute copies of the original source code. The FCS license should preserve the freedom to distribute copies of modified versions [32]. [Operational Functions 5.2.1, 5.2.2, and 5.2.4, Performance Function 5.3.2]

A-2.3.2 All libraries upon which the FCS depends must also be distributed with open source licenses compatible with the FCS license.

Basis: A code is only as distributable as the libraries upon which it depends. [Operational Functions 5.2.1, 5.2.2, and 5.2.4, Performance Function 5.3.2]

A-3. Documentation

A-3.1 In-Source Documentation

A-3.1.1 Both source code and extension modules shall be documented within the code itself.

Basis: For ease and clarity of development, in source documentation illuminates code behavior, facilitates bug-checking, and improves inter-developer communication. [Operational Functions 5.2.1 and 5.2.2, Performance Function 5.3.2.]

A-3.1.2 Automatic compilation of in-source commentary into a documentation format is desirable.

Basis: While Commercial-Off-The-Shelf software (COTS) often provide this capability in their GUIs, tools also exist for more basic programming environments which convert in-source comments to formatted documentation for web pages and distribution. For an example of such a tool, see Doxygen (http://www.doxygen.org). [Operational Functions 5.2.1 and 5.2.2, Performance Function 5.3.2.]

A-3.2 User Manual

A-3.2.1 A user manual should be generated, maintained, and freely distributed which documents access, installation, use, development, model design and development validation history.

Basis: Ease and accessibility of code use and development relies heavily on up to date, informative, easily accessible instructions, design explanation, and testing history. [Operational Functions 5.2.1 and 5.2.2]

A-3.2.2 A user manual design that provides the reader with a choice between documentation at various levels of detail from a global overview to detailed model specifics is desirable.

Basis: The TSM experience with their user manual design indicates that a single user manual with a structure facilitating users interested in varying levels of detail is desirable [34]. The design suggested by lessons learned from TSM indicates that a modular, hierarchical user manual will achieve this. [Operational Functions 5.2.1 and 5.2.2]

A-4. Verification and Validation

A-4.1 Manual Accuracy Review

A-4.1.1 Process equations, predictive algorithms, and transmutation matrices are a few examples of the equations that must be individually checked for accuracy.

Basis: Verification is the process of investigating the equations and business logic of the simulation and asserting their accuracy [4]. In order to trust that the FCS behaves as expected each method in the simulation must individually behave as expected. [Performance Function 5.3.2]

A-4.1.2 The FCS must provide and rely on only those data sets that are confirmed to be accurate and standardized by the literature.

Basis: Validation addresses the accuracy of the parametric data with which we solve the equations and business logic of the simulation [4]. In order to assert that the FCS behaves as expected the parameters supplied to the methods in the simulation must be as accurate as possible. Validation is typically based on experiments. [Performance Function 5.3.2]

A-4.1.3 All sources of data used in the FCS must be cited in the source code comments where they appear and in the documentation. This ensures that all data shall be traceable to the source

Basis: Referencing sources where they appear in the code facilitates user/developer error checking and citing them in the documentation describing the models improves code transparency. [Performance Function 5.3.2]

A-4.2 Integrated Unit Testing

A-4.2.1 Expected behavior for each model and modular extension to the FCS should be confirmable with pre-constructed unit tests.

Basis: A unit is a specific, small aspect of the code that can be tested. An example of a simple unit test might determine whether a finite mine facility model produces exactly the finite amount requested of it, or if each facility gracefully responds to a feedstock disruption. [Operational Function 5.2.2, Performance Function 5.3.2]

A-4.2.2 Unit tests should be packaged in such a way as to be integral to the developer distribution of the build system.

Basis: Such a system confirms expected performance of the modular components as the executable is built, alerting the user or developer to unexpected behavior in the simulator before simulations are run. [Operational Function 5.2.2, Performance Function 5.3.2]

A-4.3 Benchmarks

A-4.3.1 Standard results of base-case scenarios that have been simulated and confirmed by comparable fuel cycle codes must be compared to results generated by the FCS. Base cases should be run against CAFCA-SD, VISION[35], DANESS [6], COSI6[11] to the extent that their functional capabilities overlap those of FCS. An example timeline of base cases follows in Table 2.

Basis: In order to rigorously test for standard, expected behavior and fundamental functionality the FCS must compare favorably to accepted simulation results by codes with similar functionality. Notably, while the literature indicates a spattering of such benchmarks for fuel cycle simulation codes, the field of such codes lacks capability uniformity and standard benchmarks cover only the shared basic functions of each code. An example set of base cases follows in Table 2.

Table 2 - Example set of cases for FCS Benchmarks

Type	Description	Benchmark Reference
Once Through	A single UOX PWR with standard isotopic recipes.	DYMOND case 1 [25] [11]
	One new UOX PWR every five years for one hundred years.	Explicit Deployment Base Case
	Ten initial UOX PWRs and new response to demand growth at a 2% growth rate for one hundred years.	Demand Response Base Case
	One PWR region and one PHWR region simultaneously responding to similar demand curves.	INPRO [14]
	Many simultaneous regions, each responding to different demand curves with various reactor types.	Multi-region base case
	120 years, constant 60 GWe, PWR open cycle	OECD/NEA scenario 1 [3] [11]
Thermal Recycle	Thermal recycle in an LWR with blended ¾ UOX and ¼ Inert Matrix Np-Pu-Am fuel.	DYMOND case 2a.1 [25]

Type	Description	Benchmark Reference
	Thermal recycle in LWR with a full core of MOX Np-Pu-Am fuel.	DYMOND case 2a.2 [25]
	120 years, constant 60 GWe, PWR UOX simultaneous with PWR MOX.	OECD/NEA scenario 2 [3]
	LWR-UOX to LWR-UOX/MOX (30% MOX loading) to FR burner (CR=0.5),	MIT case 2 [12] [11]
Mixed Thermal and Fast Recycle	Np-Pu IMF LWR with recycling in CR 0.25 fast reactor	DYMOND case 2b.1 [25]
	Np-Pu MOX LWR and CR 0.25 fast reactor	DYMOND case 2b.2 [25]
	120 years, constant 60 GWe, PWR UOX until year 109, PWR MOX until year 72, FR begins in year 80.	OECD/NEA scenario 3 [3]
Fast Recycle	UOX LWR recycled in a CR 0.25 fast reactor	DYMOND case 2c [25]
	UOX LWRs to FR burners (CR=0.5, starting in year 40)	MIT case 1 [12]
	Gen III UOX LWRs to FR (CR=1.0, starting in year 40)	MIT case 3 [12]

Economic analysis could be benchmarked against MARKAL, according to the AEO 2006 reference case. The Advanced Fuel Cycle Economic Basis gives a good economic benchmark according to this data with further macroeconomic and demographic assumptions from 2005 to 2050.

The Total System Model (TSM) of the Yucca Mountain Project provides some benchmarking for potential repository and transportation models. Included waste package characteristics should be validated by the VB code in the TSM. [34]

A-5. User Interface

A-5.1 Input: Appropriate Levels of Detail

A-5.1.1 The FCS will provide a default configuration for each facility model it provides, providing all necessary data and standard behavioral, economic, and material parameters.

Basis: Some use cases will focus on the broad strokes of facility combinations rather than the finer details of facility model parameters and behaviors. Also, some other use cases may be interested in the detailed behavior for some but not all facilities. To facilitate these users and use cases, preconfigured standard facility behaviors and parameters must be available. [F1-F16, S1-S11, E1-E14, M1-M16, U1-U5, Operational Function 5.2.1 and Performance Function 5.3.3]

A-5.1.2 The FCS will facilitate user specification of all appropriate behavioral, economic, and material parameters.

Basis: Some use cases will require the freedom to define specific facility parameters. For example, a technical study of the effect of separations processes on repository metrics will require detailed modification of separations facility model parameters such as efficiencies and process types. [F1-F16, S1-S11, E1-E14, M1-M16, U1-U5, Operational Function 5.2.1 and Performance Function 5.3.3]

A-5.1.3 The FCS will provide optimistic, pessimistic, and business as usual demand curves to use in demand driven simulations.

Basis: End-users, advanced users, and developers may be equally interested in relying on standard demand curves in order to focus on the other aspects of fuel cycle behavior. Providing default demand curves helps to standardize simulation results and reduce potential user error. Such future projections have been made by an array of industry, international, and consulting institutions [15]. [F1-F16, S1, S3-S11, E1-E14, M1-M7, M9-M16, U1-U5, Operational Function 5.2.1, 5.2.2, and Performance Function 5.3.3]

A-5.1.4 The FCS will accept user specified demand curves for demand driven simulations.

Basis: Some use cases will focus on the response of various technology scenarios to an array of electricity demand models. For example, a user may be interested in probing fuel cycle robustness in the event of drastic increases or decreases in demand. Such users will require the ability to define custom demand curves. [S2, M8, Operational Function 5.2.1, 5.2.2 and Performance Function 5.3.3]

A-5.1.5 The FCS will provide a true to history deployment scheme for explicit deployment driven simulations.

Basis: While for some use cases, a static present state of the system is sufficient, other use cases and benchmarks require the capability to simulate the known history of the nuclear fuel cycle, both domestic and foreign. This historical data may rely on databases such as the IAEA Power Reactor Information System (PRIS) [16]. [F6, F7, F13, F14, S1, S3- S11,M6, M8-M11, M13-M16, U4, U5, Operational Function 5.2.1 and Performance Function 5.3.3]

A-5.1.6 The FCS will accept user specified deployment schemes for explicit deployment driven simulations.

Basis: Some use cases will be particularly interested in explicitly specifying custom facility deployment schedules in order to compare and contrast those deployment scenarios according to one or many fuel cycle metrics. Deployment true to history, for example, will be useful in benchmarking and validation. [S2, M8, Operational Function 5.2.1 and Performance Function 5.3.3]

A-5.2 Output: Appropriate Levels of Detail

A-5.2.1 The FCS should be equipped to automate the generation of a set of informative output graphs in standard units illuminating typical fuel cycle metrics for all major portions of the fuel cycle.

Basis: Many potential FCS use cases share output of interest. Graphs of total material flows, facility specific material flows, time histories of the locations of each isotope, etc. are interesting results to a variety of use cases. To avoid redundant human effort, these shared results of interest should be identified and a post-process system to automate their generation should be available to the user. [F1-F10, F13, F14, F16, S1, S2, S11, E1-E13, M2, M3, Operational Function 5.2.1, Performance Function 5.3.4]

A-5.2.2 The FCS should be equipped to automate the generation of key results tables in standard units illuminating typical fuel cycle metrics for all major portions of the fuel cycle.

Basis: Many potential FCS use cases share output data of interest. Tables of total facility deployments for each facility type, total time histories of the locations of each isotope, etc. are interesting results to a variety of use cases. To avoid redundant human effort, these shared results of interest should be identified and a post-process system to automate their generation should be available to the user. [F1-F10, F13, F14, F16, S1, S2, S11, E1-E13, M2, M3, Performance Function 5.3.4]

A-5.2.3 A complete description of the input parameters that led to the output should be included in the output database.

Basis: Maintenance of provenance is essential to the scientific effort [21]. Inclusion of a complete notion of the input specifications in the output database allows for result reproduction and facilitates knowledge sharing. Reproducibility is not only a sacred tenant of scientific work, but facilitates post-process analysis and error-checking. [Performance Functions 5.3.2, 5.3.3, 5.3.4]

A-5.2.4 Materials present in the system, their isotopic compositions, their locations (at least to facility resolution), and their movements between facilities shall be recorded as a function of time in the output database.

Basis: Material flow data is the fundamental unit of information for analysis of many types. [F1-F16, S1-S11, E7, E9-E14, M1-M7, M10-M12, U4, U5, Performance Requirement 5.3.4]

A-5.2.5 Graphs of material flows which describe material present in the system as a function of time shall be automatically generated. These shall be customizable, allowing display of isotope specific material flows (e.g. ^{99}Tc vs. time), facility specific material flows (e.g. ^{99}Tc in the repository vs. time), and regional flows (e.g. ^{99}Tc in Canada vs. time).

Basis: Material flow data is the fundamental unit of information for analysis of many types, so graphs of this information will be informative to viewers, end-users, advanced users, and developers alike. Graph customizability eases verification and validation by developers, post-simulation analyses by end users, and demonstration to viewers. [F1-F16, S1-S11, E7, E9-E14, M1-M7, M9-M12, U4, U5, Performance Requirement 5.3.4]

A-5.2.6 Material flow graphs concerning transfers of material between facilities and between regions shall be automatically generated along with appropriate transit details such as time and distance. These shall be customizable, allowing display of transfers of a specific isotope (e.g. ^{239}Pu in transit vs. time), inter-facility transfers (e.g. ^{239}Pu leaving a separations facility vs. time), and inter-regional transfers (e.g. ^{239}Pu between France and Germany vs. time).

Basis: Material transfer data supports an array of system analyses including those concerned with transportation dynamics, multi-regional policies, and nonproliferation metrics. Transport of material between regions may be best displayed on maps with arrows whose thickness indicate trade frequency, mass transferred, or mass of a certain isotope transferred. [F1-F7, F13, F14, S3-S6, S8-S10, E5, E8-E14, M1, M2, M4-M7, M9-M12, U1-U5, Performance Requirement 5.3.4]

A-5.2.7 Facility deployment data indicating construction, operation, and decommissioning of facilities by type (e.g. Enrichment Facilities vs. time), by subtype (e.g. Gaseous, Laser, etc. vs. time), and by region (e.g. Enrichment Facilities in the southeastern US vs. time) shall be recorded in the output database.

Basis: Facility deployment data supports system analyses concerning strategic timing, inter-facility dependencies, policy dynamics, and economics.

A-5.2.8 The FCS must be capable of generating output graphs and tables that detail cash flows vs. time for each facility and region. Material prices over time as well as calculations of fuel cycle costs and levelized costs of electricity for particular facilities and regions may also be tabulated as a part of the output database.

Basis: Cash flow data and cost information provide information necessary for economic and cost analysis. [S1, E1-E14, Performance Function 5.3.4]

A-5.3 Database Format

A-5.3.1 Material flow, facility deployment, and other raw output data indicated above must be recorded to a cross-platform database format accessible and manipulable by users in Windows, Unix, and Linux.

Basis: The output database should enable full flexibility for post-process, user-driven data analysis by all potential users. While a majority of potential users from DOE national labs operate on Windows platforms, the scientific university population includes an appreciable number of researchers who operate on UNIX (apple/mac) systems. Finally, some advanced users and developers interested in linking physics codes will be restricted to the Linux systems on which those codes are capable of compiling. Experience with GENIUS and VISION have led to some experience with three tools. A database in a basic comma-separated-value (.csv) file format will be compatible with an array of text editors, Open Office, and Microsoft Excel as long as no advanced formatting features are introduced. Faster-writing, and similarly compatible database formats include SQL and HDF5, both of which have free, open access editors and viewers available online and compatible with all major operating systems. Of these two, HDF5 is a more memory efficient database format, but is less flexible for editing and conversion. [Operational Functions 5.2.1 and 5.2.2, Performance Function 5.3.4]

A-5.3.2 The FCS database must facilitate memory savings.

Basis: Some databases are more memory efficient than others and the possible appreciable size of the output database will create a need for memory savings. [Analysis Functions 5.1.2 and 5.1.4, Operational Functions 5.2.1 and 5.2.2, Performance Function 5.3.4]

A-5.3.3 The FCS database must have a database interfacing tool that allows database creation and alteration as well as data browsing.

Basis: These tasks are essential for post processing and analysis, and the typical user must not be required to perform complex queries on the database. [Analysis Functions 5.1.2 and 5.1.4, Operational Functions 5.2.1 and 5.2.2, Performance Function 5.3.4]

A-6. Coupling with Other Tools

A-6.1 Interface

A-6.1.1 The FCS will be capable of supporting linkages to physics codes such as ORIGEN.

Basis: Potential use cases involving coupling of FCS fuel cycle modeling with modules that call the more detailed calculations of other codes. This may require interfacing hooks enabled by an intermediate scripting language such as Python or XML that has translational capability between programming environments such as FORTRAN, C/C++, MatLab and Visual Basic. Alternatively, shared databases in a cross-platform database format such as HDF5 might serve to transport data between the FCS and more detailed physics codes. [F1, F2, F13, F14, S8, S11, Operational Function 5.2.2]

A-6.1.2 The FCS shall support the capability to focus on the detailed behavior of a portion of the fuel cycle while neglecting the details of other areas of the system.

Basis: These 'imbalanced' simulations will help minimize runtimes for users interested in modeling fuel cycle subsystems with high precision tools. They also provide a framework for module testing during development and benchmarking against other fuel cycle codes. [F1-F14, S3-S7, Operational Functions 5.2.1, 5.2.2, 5.2.3, 5.2.4, Performance Function 5.3.2]

A-6.2 Model Size

A-6.2.1 The FCS should be capable of a fuel cycle model that deploys just one explicit facility model and summarizes the rest with a front and back end material source and sink.

Basis: The validation and verification process will require examination of the behavior of a single facility model. During development of a facility model, its behavior within the simulation must be observable independent from other facility model. For this to be possible, all facility models upstream should be collapsible into a finite material source facility module and all facility models downstream should be similarly collapsible into a finite material sink facility module. [F1-F14, S3-S7, Operational Functions 5.2.1, 5.2.2, 5.2.3, 5.2.4, Performance Function 5.3.2]

A-6.2.2 The FCS should be capable of recording material and facility histories for time horizons extending to hundreds of years.

Basis: Political planning horizons and facility lifetimes are on the order of at least 50 to 100 years. Important activities of the nuclear fuel cycle, such as waste internment are operations with forecasted processes a few hundred years into the future. Though certainly radioisotope halflives well exceed this time scale, basic social and technical assumptions by which this simulation is driven become untrustworthy a few hundred years into the future. [F13, F14, F1-F4, F8, F9, S1-S8, S10, S11, E2, E5, E6, E9, E11, E14, M4-M6, M8-M12, U5]

A-6.2.3 The FCS shall be capable of tracking and recording deployment and behavior for tens of thousands of simultaneously interacting facilities.

Basis: Current economic and technical growth rates indicate that the few hundred reactors and fuel cycle facilities in the world will number in the thousands on the time scale of the simulation (hundreds of years, see above). However, there are also "high case" projections for global nuclear capacities as high as 10,000 GWe by 2100. The FCS will need to support tens of thousands of instantiations. It's worth noting that this is a problem for the object-based approach, and a method to collapse objects would be useful –

when a single country has 1000+ reactors, lumping (or implementation of a flyweight design pattern [9]) would probably improve comprehension. [F1, F2, F13, F14, S1, S8, S11, E1, E2, E6, E12-E14, M4-M7, M9-M16, U1-U5]

A-6.2.4 The FCS should be capable of recording and tracking the composition histories of hundreds of thousands of material objects.

Basis: In order to accurately model tens of thousands of fuel cycle facilities and their associated material flows, the simulation must be capable of modeling at least a factor of fifty more material objects than facilities. That is, if a material object is the size of a batch of fuel, a reactor with an 18 month cycle will pass over fifty fuel batches in an 80 year lifetime. [F1, F2, F13, F14, S1, S8, S11, E1, E2, E6, E12-E14, M4-M7, M9-M16, U1-U5]

A-7. Mass Flow Analysis

A-7.1 Isotopic Tracking

A-7.1.1 The basic unit of information in the FCS shall be nuclear material movement and transmutation.

Basis: Mass flows can be translated in post-processing into many metrics of interest to fuel cycle analysis. Facility inventories, radio toxicity, decay heat, waste volumes, PRPP indices, etc. can all be found from mass flow data. [F1-F10, F13-F16, S1-S11, E2, E3, E7-E14, M1-M8, M10, U1, U3-U5, Analysis Function 5.1.1]

A-7.1.2 The FCS must be capable of tracking the mass of element groups, individual elements, isotope groups and individual isotopes

Basis: Currently, 81 mass items have been determined to be of interest in maintaining mass balances and calculating metrics based on experience with VISION. These include groups of elements, individual elements, groups of isotopes and individual isotopes.

A-7.1.3 The FCS must be capable of extending isotope tracking to include (or exclude) any specific isotope of interest (or disinterest).

Basis: As new fuel cycle technologies progress, the effects of previously unimportant isotopes on fuel cycle metrics of interest often become non-negligible [24]. [F1-F10, F13-F16, S3-S11, E2, E3, E7, E9-E14, M1-M3, M6 M10, U1, U4, U5, Analysis Function 5.1.1]

A-7.2 Mass Tracking

A-7.2.1 Mass must be conserved in the FCS.

Basis: Conservation of the fundamental unit of information in a simulation provides transparency to the developer and the user. A fundamentally sacred unit also provides a useful tool in error checking during development. Mass flows can be translated in post-processing into many metrics of interest to fuel cycle analysis. Facility inventories, radiotoxicity, decay heat, waste volumes, PRPP indices, etc. can all be found from mass flow data. [F1-F10, F13-F16, S1-S11, E2, E3, E7-E14, M1-M8, M10, U1, U3-U5, Analysis Function 5.1.1]

A-7.2.2 The FCS must be able to convert mass between mass groups due to transmutation and decay.

Basis: Lossless mass tracking requires accounting for mass changes between mass groups. Furthermore, a fundamentally unit provides a good basis for error checking during development. [F1-F10, F13-F16, S1-S11, E2, E3, E7-E14, M1-M8, M10, U1, U3-U5, Analysis Function 5.1.1]

A-7.3 Material Data

A-7.3.1 The FCS shall provide validated preconfigured modules for each stage of the fuel cycle.

Basis: Enough data should come pre-packaged with the FCS such that an end-user or viewer may run a broad range of scenarios without providing any data to the simulation. Only validated, quality controlled data should be made available for the user to select when designing a scenario. [Analysis Function 5.1.1, Operational Function 5.2.1 and Performance Function 5.3.3]

A-7.3.2 The FCS must provide comprehensive and validated decay data.

Basis: This supports all mission objectives involving the use of preconfigured storage and repository models, which will rely on decay data in order to perform appropriate transmutations of material objects at runtime. [F1, F3-F10, F12, F14-F16, S2-S10, E2, E3, E6-E13, M2, M3, M6, M7, M14, U1, U3-U5, Operational Functions 5.2.1 and Performance Function 5.3.3, Analysis Function 5.1.1]

A-7.3.3 Validated libraries of standard data concerning input and output fuel compositions must be provided.

Basis: This supports benchmarking and verification objectives as well as all mission objectives involving the use of standard reactor facility types and burnups. [F1-F4, F6-F10, F12, F13, F15-F16, S1-S11, E2, E3, E6-E13, M1-M6, M7, U1, U3-U5, Analysis Function 5.1.1, Operational Function 5.2.1 and Performance Function 5.3.3]

A-7.3.4 Validated libraries of standard data concerning transmutation by irradiation must be provided.

Basis: This supports all mission objectives involving the use of preconfigured nuclear systems, which will rely on libraries of externally calculated core physics isotopics in order to perform appropriate transmutations of material objects at runtime. [F1-F4, F6-F10, F12, F13, F15-F16, S1-S11, E2, E3, E6-E13, M1-M7, U1, U3-U5, Analysis Function 5.1.1, Operational Function 5.2.1 and Performance Function 5.3.3].

A-7.3.5 A library of verified and validated separations matrices, standard reprocessing method data and process details must be provided.

Basis: This supports all mission objectives involving the use of preconfigured reprocessing facilities, which will rely on separations matrices modeling standard aqueous (i.e. PUREX, UREX and electrochemical), pyrolitic (e.g. electrolysis, voloxidation, or fluoride volatility), and other standard reprocessing methods to perform appropriate transmutations of material objects at runtime. [F1-F10, F12-F16, S1-S10, E1-E3, E8-E14, M1-M7, M10, M16, U1-U5, Analysis Function 5.1.1, Operational Function 5.2.1 and Performance Function 5.3.3].

A-7.3.6 A verified library of data concerning material chemical forms, waste forms, and material packaging must be provided.

Basis: This supports all mission objectives concerned with waste characterization which rely on preconfigured repository, storage, and transportation models. [F1-F10, F12, F13, F15, F16, S1, S3-S10, E1-E3, E6, E7, E9, E11, M2, M3, M6, M7, U3-U5, Analysis Function 5.1.1, Operational Function 5.2.1 and Performance Function 5.3.3]

A-8. Cost Analysis

A-8.1 Timeline Data

A-8.1.1 The FCS must either calculate a levelized electricity cost or its output database must include basic timeline data that includes dates of facility deployment, power production, fuel orders, and decommissioning.

Basis: These four basic factors can be used to analyze (in post-processing) many economic metrics. [S1, E1-E14, M1-M7, M10, M15, U3, U5, Analysis Function 5.1.2]

In particular, the levelized cost of electricity ($\bar{e}$ in \$/kWh) for a facility can be derived from the average annual operational cost ($\bar{O}$ in \$), and construction, decommissioning, and power production (production of K, the facility capacity and L, the capacity factor) schedules over a number of years N. Operational cost calculations rely on fuel ordering and power production histories. Using $\bar{O}$ and some user provided data (depreciation schedule (d_n), tax rate (τ in %), facility capital cost (I_0 in \$), property tax and insurance rate (p %), and A/P and P/F which can be calculated from interest rate (x %)) this important metric can be calculated in post processing.

$$\bar{e} = \frac{\bar{O}}{8.766K\bar{L}} + \frac{I_0}{8.766K\bar{L}}\left(p + \frac{(A/P,x,N)}{1-\tau} - \frac{\tau(A/P,x,N)}{1-\tau}\sum_{n=1}^{N} d_n (P/F,x,n) \right) \quad (2)$$

A-8.1.2 The FCS must provide default capital and operating costs for each facility.

Basis: Enough data should come pre-packaged with the FCS such that an end-user may run economic analysis without providing any data to the simulation. [S1, E1-E14, M1-M7, M10, M15, U3, U5, Analysis Function 5.1.2, Operational Function 5.2.1 and Performance Function 5.3.3]

A-8.1.3 The FCS must allow user-specification of capital and operating costs for each facility.

Basis: This supports mission objectives concerned with cost analyses (e.g. sensitivity of fuel cycle costs to capital and operating costs of facilities) as well as economic analyses (e.g. the response of the system to institutional capital availability limits). [F1, F2, S1, E1-E14, M1-M7, M10, M15, U3, U5, Analysis Function 5.1.2]

A-8.1.4 The FCS must allow user specification of pricing schemes for uranium resources.

Basis: This supports mission objectives concerned with cost analyses (e.g. sensitivity of fuel cycle costs to uranium pricing) as well as economic analyses (e.g. the response of the system to uranium pricing). [F1-F4, S1, S2, E1-E7, E10, E14, M1-M7, M9, M10, M12-M15, U3, U4, Analysis Function 5.1.2]

A-8.1.5 The FCS must provide an optimistic and a pessimistic pricing scheme for uranium resources.

Basis: Enough data should come pre-packaged with the FCS such that an end-user may run cost analysis without providing any data to the simulation. [F1-F4, S1, S2, E1-E7, E10, E14, M1-M7, M9, M10, M12M15, U3, U4, Analysis Function 5.1.2, Operational Function 5.2.1 and Performance Function 5.3.3]

A-9. Market Analysis

A-9.1.1 The FCS must be extendible to incorporate market models which determine matching between material orders and products.

Basis: Purchases and material transfers are not made arbitrarily and often rely on contracts, competitive pricing, and other market phenomena. Extensibility to incorporate market models and transportation functionality is necessary to accurately model fuel cycle behavior. [F6. F7, S1, S3-S5, S8-S11, E4, E5, E8-E14, M1-M16, U1-U5, Analysis Function 5.1.7, Operational Function 5.2.2]

A-9.1.2 The FCS market model that dictates material trade between facilities must be extensible to allow future models incorporating regional behavior and relations.

Basis: Financial, regulatory and policy relations play a pivotal role in the movement of nuclear materials. Thus extensibility for modeling regional behavior and international trade relationships is necessary to address questions of policy, energy security, fuel availability, demand growth, etc. [S1, S8, E14, M1-M16, Analysis Function 5.1.7, Operational Function 5.2.2]

A-9.1.3 The FCS must robustly incorporate developer extensions that describe transportation model packaging, distances, routes, and travel time between facilities.

Basis: Use cases concerned with the economic, security, and safety concerns of material transportation in the nuclear industry will require customizable transportation dynamics. The details of material packaging and aging during transportation are also necessary to accurately model the chemical and isotopic composition of material during a fuel cycle scenario. [F7, S1, S3-S10, E9-E11, E14, M1-M16, U1-U5, Analysis Function 5.1.7, Operational Function 5.2.2]

A-10. Sensitivity Analysis

A-10.1 I/O Automation

A-10.1.1 The capability to streamline input parameter variation and code execution with a script, code wrapper, or built-in option is necessary.

Basis: Sensitivity analyses must be executable in an automated fashion in order to produce results in wall clock time without repetitive human effort. [F1, F2, F5-F13, F15, E1-E14, M1-M3, M8, M9, M11-M16, U1-U5, Analysis Function 5.1.4, Operational Function 5.2.1]

A-10.1.2 The FCS will have the capability to neglect recording of individual metrics and datasets.

Basis: The output database may be of a non-trivial size, so minimization of hard disk space usage during sequential simulation executions is desirable. Many sensitivity analyses will be concerned with only a few of the myriad output metrics and generated data, so neglecting uninteresting records saves space. Also, the process of recording data to the database may be a non-trivial contributor to runtime and should be similarly minimized. [F1, F2, F5-F13, F15, E1-E14, M1-M3, M8, M9, M11-M16, U1-U5, Analysis Function 5.1.4, Operational Functions 5.2.1 and 5.2.2]

A-10.2 Run Time Constraints

A-10.2.1 Base case simulations modeling upwards of 1000 facilities and upwards of 100 years must have run times on the order of a few minutes.

Basis: In order to informatively explore parameter fields on reasonable time scales, an automated perturbation of a parameter consisting of possibly thousands of full length simulations must be possible in wall-clock time. [F1, F2, F5-F13, F15, S11, E1-E14, M1-M3, M8, M9, M11-M16, U1-U5, Analysis Function 5.1.4, Operational Function 5.2.1 and Performance Function 5.3.3]

A-11. Uncertainty Analysis

A-11.1 Uncertainty Data

A-11.1.1 FCS shall support validated uncertainty distributions for primary data supplied to the model.

Basis: Uncertainty analysis informs and supports model decisions. As parameters are shown to have low or high impact on simulation results, the appropriate approximations and model design decisions are made clear. A parameter field can be probed in a Monte Carlo fashion if appropriate data is sampled from probability distributions. For models of adequate sample size, uncertainty analysis can be conducted if uncertainty is propagated through the model. [F1, F3-F10, F12, F14-F16, S2-S10, E1-E3, E6-E13, M2, M3, M6-M8, M14, U1-U5, Analysis Function 5.1.5, Operational Function 5.2.1 and Performance Function 5.3.3]

A-11.1.2 The FCS shall facilitate customization of uncertainty distributions for data that is customizable by the user.

Basis: Uncertainty analyses should be possible on data introduced by the user as well. In fact, error propagation of possibly un-validated data provided by the user, or by the output of external codes coupled to the FCS is arguably more valuable than uncertainty analyses on validated, static data provided by the FCS. [F1, F3-F10, F12, F14-F16, S2-S10, E1-E3, E6-E13, M2, M3, M6-M8, M14, U1-U5, Analysis Function 5.1.5, Operational Function 5.2.1 and Performance Function 5.3.3]

A-11.2 I/O Automation

A-11.2.1 The FCS should be capable of automating the sampling of probability distributions.

Basis: Many sequential simulation runs are required by uncertainty analysis for Monte-Carlo type uncertainty analyses as well as variational-propagation-type uncertainty analyses where appropriate. [F1, F3-F10, F12, F14-F16, S2-S10, E1-E3, E6-E13, M2, M3, M6-M8, M14, U1-U5, Analysis Function 5.1.5, Operational Function 5.2.1 and Performance Function 5.3.3]

A-11.2.2 The FCS shall have the capability to produce 'uncertainty band' type graphs.

Basis: The fundamental result of an uncertainty analysis is best displayed in an 'uncertainty band' graph style. Automated production of such a visualization will encourage and facilitate code use by users. [F1, F3-F10, F12, F14-F16, S2-S10, E1-E3, E6-E13, M2, M3, M6-M8, M14, U1-U5, Analysis Function 5.1.5, Operational Function 5.2.1 and Performance Function 5.3.3]

A-12. Disruption Analysis

A-12.1 Simulation Control Options

A-12.1.1 The FCS will allow customizable specification of process performance uncertainty for each facility.

Basis: Economic, safeguards, and policy use cases analyzing scenario robustness may be interested in the effect of various levels of performance uncertainty of one or more facilities, facility types, or groups. Process performance uncertainty refers to the likelihood of a facility to perform at its capacity at any given time. For the mine/mill this may reflect resource availability while for facilities such as the reactor this parameter reflects the availability factor less the unplanned outage factor. These contribute to the overall 'capacity factor' of a facility [17]. [M1-M16, U1-U5, Analysis Function 5.1.6, Operational Function 5.2.2]

A-12.1.2 Simulation behavior in the FCS will facilitate modeling of possible political or transportation disruptions.

Basis: Economic, safeguards, and policy use cases analyzing scenario robustness may be interested in these non-facility processes of the fuel cycle. Markets will require some notion of political affinity and transportation should be customizable input parameters. [M1-M16, U1-R5, Analysis Function 5.1.6, Operational Function 5.2.2]

A-12.2 Disruption Response Realism

A-12.2.1 Each facility must appropriately handle material composition changes in the event of a process disruption.

Basis: In disruption analysis use cases, each facility must gracefully experience disruption in order for the rest of the fuel cycle to appropriately reflect the effects of that disruption. Graceful facility disruption must include appropriate material composition changes (e.g. time dependent decay, modification to reactor core discharge burnup, or the isotopics of fresh fuel batch orders). Such isotopic changes are necessary in order to accurately characterize isotopic compositions downstream of that facility in the simulation. [M1-M16, U1-R5, Analysis Function 5.1.6]

A-12.2.2 Each facility must appropriately handle internal economic effects resulting from a process disruption within that facility or upstream of that facility (e.g. insufficient feedstock).

Basis: In disruption analysis use cases, each facility must gracefully experience disruption in order for the rest of the fuel cycle to appropriately reflect the effects of that disruption. Graceful facility disruption must reflect appropriate economic implications (e.g. increased maintenance costs, modified process costs, loss in revenue), and other appropriate market effects (e.g. feedstock reordering or product shipment rescheduling). Such changes are necessary in order to affect responses both upstream and downstream of that facility in the simulation. For example, an unexpected power reactor shutdown has nontrivial implications for power production, fresh fuel order schedules, etc. [E4-E8, E12-E14, M1-M16, U1-U5, Analysis Function 5.1.6]

A-13. Facility Data

A-13.1.1 Cost related data for each facility shall be provided to support cost analyses.

- ***Capital Costs –*** The initial investment for licensing, construction, and commissioning of the nuclear facility.
- ***Operation & Maintenance Costs –*** The predicted annual investment for operating and maintaining the nuclear facility.
- ***Material Prices –*** A pricing structure for uranium resources shall be made available. The red book estimate
- ***Construction Time –*** The date the facility is commissioned will vary for each facility and will be determined by the deployment schedule, as determined by the simulation logic.
- ***Decommissioning Time –*** The date the facility is decommissioned will vary for each facility and could just as easily be replace by the combination of construction time and facility lifetime. Thus, this too is determined by the deployment schedule, as determined by the simulation logic.
- ***Capacity –*** The amount of material, product, storage space, power, or SWUs that this facility can process in a given time.
- ***Availability Factor –*** The amount of time this facility operates at capacity. That is, a facility may characteristically have one month of unexpected shut down time during a year. In that case, they have an availability factor of 11/12 for that year. A default value for this parameter should be available for each facility.

A-13.1.2 Basic process data for each fundamental facility model in the fuel cycle must be provided.

- ***Mine / Mill –*** A mine/mill model shall include standard default values and allowed input parameter ranges defining uranium resource availability, resource disposition (ore grade and time and cost of recovery), chemical form, process time, and mill tailing and other waste streams.

- ***Conversion*** **–** A simulation of the conversion model shall include default values and allowed input parameter ranges defining resultant chemical form, process time, and waste streams.
- ***Separations*** – A separations model shall provide standard separations matrices that support separations into multiple product and waste streams
- ***Fuel Fabrication*** – A fuel fabrication model shall be equipped with a library of fresh fuel recipes for each reactor model suggested by the supported reactor types.
- ***Reactor*** – A reactor model must be capable of modeling the full range of nuclear systems, including reactors and sub-critical systems. It must also be capable of utilizing multiple fuel types appropriate for the reactor type.
- ***Interim Storage*** – An interim storage model shall be capable of performing decay calculations on material objects of arbitrary isotopic composition.
- ***Repository*** – A repository model shall have finite but customizable volumetric, heat, and toxicity limits. It shall also be capable of performing decay calculations on material objects of arbitrary isotopic composition.

This page intentionally left blank.

Appendix B

REVIEW MEETING NOTES

This page intentionally left blank.

Appendix B - FCS Review Meeting Notes

A meeting was held at Argonne National Lab on July 15th to review the draft of the FCS Functions & Requirements. In attendance were Brent Dixon, Temi Taiwo, J'Tia Taylor, T K Kim, Ed Holloway, Latif Yacout, Jess Gehin, Katy Huff, Lori Braase, and Robert Hill (only during the working lunch). This appendix documents the minutes of the meeting.

8:30 WELCOME (LORI)

What should be in this simulator?

What should be in this simulator? There are some use cases that may need to be in there that aren't yet. Sonny Kim's interest in the effect of carbon taxes on the demand for nuclear power may be within the scope, or not, for example. It would be nice to discuss whether we have all the areas we want to support, and how far that scope goes.

High level of discussion on the areas of requirements - We're interested in high level discussion of the broad ideas, the big changes that need to be made, etc. rather than the details.

8:45 BACKGROUND (BRENT)

Grand Challenge - Each of the campaigns were given the task of defining a grand challenge that, if successful, would provide a "game-changing" revolutionary improvement in their area. We thought about a next generation fuel cycle simulator that could do "everything". We didn't want to replace VISION, but rather to compliment it. In the long run, potentially it could replace all of the fuel cycle tools that IAEA has, NEA, the US, the French, all the labs. To us, this is the grand challenge.

Comments: It's not such a bad idea to start fresh, actually (Ed and Jess Gehin).

Efficiency - Maintaining one code is easier for everyone, and having an open architecture gives this potential. The amount that we use a fuel cycle simulator ourselves is a lot less than the potential uses from graduate students that would be possible with an open architecture.

Open Architecture - Providing a fuel cycle simulator might be an export control problem. When a developer wants to add a module, the internal developers would be able to develop

- Is it bug free?
- Does it add an absent functionality?
- Does it work with the code?

Jess Gehin: It may be really hard to get the go-ahead from the labs to make it open-source.

Brent: One problem found in the VISION review includes a question about getting breeder recipes released.
We were able to get all standard recipes released, but not Breeders.

J'Tia Most recipes are going to have a problem.

Jess: Being able to plug in codes like Arsic or ORIGEN or something would avoid the problem of being able to release the recipes.

Brent: The problem there is a run-time issue.

Jess: You can allow people to decide for themselves how much of a runtime constraint. Systems analysis might be very different. However, it seems like a lot of people are locked into the TSA idea. It's not like Yucca in which everything is very linear (in yucca, something gets past one barrier, the next barrier tries to catch it, etc.). However, a performance assessment tool is necessary, and the need is out there. A first round of pre-screening by January for the fuel cycle options. There was some notion with GNEP of broad fuel cycle decisions, then more detailed PRAs for each reactor design.

Ed: Defining the end point is nice. If you start with a module, you can get things like fuel demand, fuel discharge, annual capital, etc. Playing with an unconstrained set of input parameters will give information. Then, you can increase the model to many linked modules.

Brent: There's more than just this too, when you start adding logic for material routing, recycling, deployment to match a growth curve, etc. The important part of these plug and play modules allows you to have levels of detail appropriate for each user.

Temi: In the report, you put in place some criteria on like, the size of the file, etc. This may not be a good thing to have a requirement for.

Ed: Since not everyone is interested in dynamic simulations, being able to model just one facility is good. It will allow students just to model one facility or something.

9:00 INTRODUCTION

Ed: The other thing he noticed that was important is that we need to define the fundamental equations in parametric form.

9:15 PROGRAM DESCRIPTION

Lori: On the agenda, we want to focus most of our time on a lot of the stuff between the break and lunch, but for now, let's talk about the intro and background.

Temi: There should be references on page 1 where there is a reference to isotopic tracking etc.

Latif: We want to define in the Introduction exactly how the reader is supposed to understand the place of this tool, it's differences and similarities to current tools, etc.

Katy: I hope this meeting will include some brainstorming on exactly what that should be, too.

Ed: What would be interesting is a philosophy section.

Latif: Also, some parts of this document are way too detailed. Some notion of the basic functions and philosophies are probably more appropriate to the current mission.

Brent: We tried to put it in the appendix. It would be nice for the front matter of this document to lead into the details later on.

J'Tia: It would be nice if this were re-ordered. So, something more like an inverted triangle would be good. What is a fuel cycle code? What fuel cycle codes exist? What this fuel cycle code should do? How exactly it should do it, etc.

Brent/Ed : This document needs to be seen as a living document starting with mission requirements, leading into a whole host of similar documents, etc. Development will improve upon this.

Temi: In Chapter 2, you try to define the audience for the users of the system, but is the audience of the document is a different group?

Lori: Yes, is this for people to review or is this for generating excitement, making pitches, etc?

Brent: A different report would pitch this idea, but the point we're at right now is to have discussions about what is in this document, so we can inform a higher level of document, for excitement and pitching purposes.

Temi: How is this different from SINEMA?

Brent/Jess : It's really not. Possibly, it's just a current iteration of the Sinema vision. We may be borrowing heavily from Sinema, but we'll reference it where appropriate.

Jess: What I would encourage is to develop the requirements for the perfect tool, and we'll figure out how to get there (or whether it already exists even).

Brent: Discussed the evolution of GENIUS and Cyclus

Latif: This is where we want to focus. Systems dynamics was good for a while, but things like uncertainty analysis, sensitivity analysis, etc. are limited by this.

Ed: If the goal is to do some optimization, then aren't we required mathematically to calculate the adjoint?

Brent: Optimization may be possible for individual parts, but isn't really possible for fuel cycle systems. There are too many uncertainties, etc. to do real optimization.

9:30 PROGRAM REQUIREMENTS

Latif: Industry should be included explicitly in the potential users.

J'Tia: We need some more explanation of the Figure 1.

Jess: If you just throw a code at a user, you can have this problem in which new users are discouraged by early bugs. We often have these use cases that drive the development, and we have a user rather than a developer who tries to ensure the code can accomplish the driving use cases. We need to be careful not to make the users beta testers. So, we need to make sure that benchmarking is done by the maintainers.

Brent: If we ever have 100 users, the stable version of the code needs to be bug free.

TK: Why did we define all these users?

Katy: This will help define the way the code needs to provide an interface with different levels of detail for each type of user. Not everyone is very sophisticated.

Ed: In the 2.2.* sections it should talk about the classes of analyses you expect different analyses you expect each group to do. What specific questions you might want to ask if you were that user.

Brent: Part of that is in Section 5. We may want to restructure the order much like J'Tia suggested. There's this very big difference between people who would use this. Some would be interested in subcritical transmuters, but some will be interested in much simpler analyses.

Jess: You need to make it possible to plug in modules that a physicist could write, possibly Python.

10:15 DEPTH AND BREADTH

Ed: The lowest level of analysis might not be low enough. Just picking one block in an unconstrained input/output model of one process or facility should be possible. A more microscopic analysis should be possible.

Brent: So let's try to find some bounds here so that we avoid octopi. If I'm plugging in a separations facility, does detail mean I'm modeling a facility, or each stage of a cascade, or what?

Latif: The input of the facility is only interesting insofar as the output interacts with the rest of the code. Why would you want to do so much detail if there is not a way for this much information coming out.

Ed: Initially when you start writing code, you want to be able to build small blocks like facility components. Needs and time will tell whether the user needs to use that level of detail.

Latif: How do you use the level of detail in the model if you never put it in the output database?

Brent: There's some analysis that is out of scope here too, like the analysis of costs of co-located plants that are interested in the costs relieved by shared resources, but maybe that's not in the scope of this analysis. I think you have a really good point, Latif,

Brent: To summarize: 1) We should make the architecture extendible enough to add infinite detail (stages of cascades) but 2) only include in the core release models that go to a level of detail that has a ripple effect in the rest of the fuel cycle.

Jess: You may want to add detailed constraints that have real implications for the model feedback but don't have any kind of output parameter. You may want to see different fuel cycle wide effects of a different reagent in the separations facility.

Latif: You want to be sure to make this model as flexible in terms of input and output for each model as is reasonably possible.

Ed: Agreed. We put so much into a VISION run that it's often hard to reduce the information in the output to the causes in the input. It helps to be able to run a single facility in order to determine what's happening on a piece by piece basis before you can roll the pieces together.

Katy: Something important you can use this for is Validation and Verification, so running a single facility in isolation may be useless for analysis, but for validation is important.

Ed: Really, each piece may be important analysis for itself.

Jess: That's right. A lot of power users will want to use this feature of single facility analysis to investigate the effects of sub-parameters, etc. This could help with driving reactor design, for example.

Brent: So, what you're saying here is that you might want to investigate models to find the right characteristics of a process model in order to fit with the rest of the fuel cycle.

Latif: The word scenario is a little funny since you never use it again. Maybe simulation is a better word.

Ed: Breadth: In my mind, a fuel cycle has the scope everywhere between ground and ground.

Brent: Depth: Right now, the way the document is written, some analyses involve the whole analysis of simultaneous nuclear fuel cycles. But, what about the way nuclear interacts with other electricity production. Should we restrict it to electricity, or should we include thermal? Then, should we limit it to nuclear or include it's sympathetic interaction with other sources?

Latif: It should probably be by itself. There are some really huge codes already.

Ed: But it's supposed to be the mother of all codes.

Jess: Maybe it just needs to be compatible with the big mothers of all climate model codes and economics models.

Temi: Yes, I think we should probably think about making a coupling capability for sharing that information through the demand driver.

Ed: In the near term, it's not likely we'll get anywhere near the rest of the energy production in the world. However, maybe we can have very simple, basic parametric models that suggest nuclear growth rate vs. other energy sources.

Jess: There are some compatibility issues that might be considered in terms of time steps (related to Sonny Kim's models, for example).

Brent: Do we want this to be dynamic or static?

Temi: How do other codes do that?

Latif: Simple high level wrappers that allow this FCS module to interact via economics, electricity demand, etc.

Brent: What if we address this by specifying that the demand curve be dynamic rather than static. That is, responding to feedback. Then, some simple model that defines parameters that affect the demand curve.

Temi: In the case of plugging in some module from a more complicated physics code, we need to be able to have dynamic and static options for modules like demand.

Latif: Generally, the requirements should say something like , "The code can interface with other codes, etc." In 2.3.2 there is maybe too much detail.

J'Tia: What does Program refer to in Program requirements?

Brent: It's complicated. We'll have to rearrange a lot of this stuff.

Latif: This statement about running backward in time, what's that?

Brent/Katy: Sometimes you realize halfway through the simulation that you want to change some parameter in the input.

Temi: That comes with a lot of overhead.

Jess: This is a big difference. There is post-analysis and there's running something you interact with while you're doing it

Latif: This is often called restart capability.

Ed: We may want to make it possible to extend the function to extend past the point at which you can run it on a pc.

Jess: Can we make conflicting requirements?

Ed: I think we have two requirements. One: extensibility to enormo-exoflop machines, but also Two: the ability to run on basic platforms.

Latif: You need to make this code capable of running on parallel machines.

Jess: The code then needs to reach out to other machines? There are nontrivial aspects of machine architecture that probably going to limit parallel functions on the module level.

Temi: It's probably best to make sure that the basic functions not be limited to huge machines. Are some users limited to different classes of problems then?

Latif: Maybe if you keep in mind that classes of analysis like multi-regions and uncertainty analyses willbe limited to massively parallel machines.

Jess: So, all basic functions, with whatever low fidelity versions of the complicated modules, should be capable of running on basic PCs.

Brent: There's an important point at which the library of known true-to-history might be useful independent of the simulator. Thus, we might want to require that the user be able to interact with the library all by itself, so if people start populating the library with useful stuff it's available for browsing.

Temi: It would be nice to make sure that the groups who update this information be the ultimate authority on that data rather than the FCS.

Latif: The architecture needs to have a separation between the simulation and the databases.

Brent: Think of putting all the information in the Cost Basis report. It would be nice to have a library of this information for different types of data (econ, deployment, isotopes, geography, power demand).

Brent: The library needs to have selectable versions, etc.

Jess: I know there's some work going on in Knowledge Management. Maybe that stuff could be used in developing this library structure.

Latif: Are the uses for this library outside of the scope of this simulator?

Brent: Maybe not, but we should acknowledge it.

Jess: Also, this database might be huge. Maybe you need to be able to select types of libraries.

J'Tia: It'll be helpful to summarize the requirements in the FCS Requirements discussion. It's important for these things to stand out. (Section 3)

Jess: There are other things we can do, too, in terms of animation, caves, demonstration, etc. Animation and other advanced animation techniques might be possible to add in the future.

Brent: So, it's a possibility now, to add animation, but we'll leave it open ended.

11:25 CURRENT CODES

Temi: Why does it say FCS needs to be a top level supplement? Maybe this is misleading?

Brent: Let's say complementary or supplementary.

Ed: We're clearly trying to avoid saying that we're going to try to replace them.

Brent: Are we really replacing VISION?

Ed: We can just say incorporate. We're going to do this lessons learned thing. In the near term we'll complement what we have and potentially incorporate what we have into a bigger thing in the long term.

Temi: The last phrase involving critical gaps needs to be cited. So does Table 1.

Latif: Discrete materials, what's the philosophy there?

Brent: There are phenomena like the input feeds to your Separations plants, etc. that require discrete tracking, but the notion that batch tracking because of burnup is a little silly.

Jess: The question arises whether you need to model our current fuel cycle at the batch level.

Latif: We should use the word modeling instead of materials. Discrete modeling covers discrete facilities, materials, etc. Agent based models are an example of this. If you have a passive reactor, it just moves a process in and out. If you have an agent it will use message passing. You want to make these things possible. This has been done at ANL in terms in the electrical grid.

Brent: If the government decides to convert toward MOX, the reactor decides whether or not it becomes a MOX plant and this requires agency in the reactor.

Brent: Let's move on to this word upset. Is it a good word? What about reliability analysis? If you're doing a reliability analysis you give each facility probabilities and have a system brittleness metric. It's a particular type of analysis.

Latif: Disruption Analysis is a good thing to call it.

Ed: For uncertainty, you need to calculate things like an adjoint or a residual. It's really important to make it clear exactly how huge this requirement is. The customer of this requirements document needs to see that it's a million dollar task.

Brent: The capability of propagating the uncertainty through the system is probably a requirement in itself.

Jess: If it's a Monte Carlo approach, then it's something you can do with a wrapper. If it's something else, then it's complicated.

Brent: There's something called DPL that can wrap an uncertainty propagator like this.

Brent: Let's talk about this table. Is this informative? Is this helpful?

Latif: No.

Temi: Yes.

Brent: No. A vague discussion of these things is probably useful.

All: The table is not complete enough.

Latif: There's got to be just a little mention in each of the codes in the paragraphs.

Jess: Reference the studies and make some comments, but move past.

Latif: Add unconventional reactor modeling, multiregional analysis, etc.

- unconventional reactor modeling
- multiregional analysis
- optimization compatibility
- target analysis - you have some goal output metric and try to let the code get the fuel cycle there.
- economic analysis - with feedback

SECTION 5

Brent: Add waste management to the diagram.

All: Change Fabricate Fresh Fuel to Fabricate Fuel. Change Fabricate Recycled Fuel to Reprocess.

Brent: Deploy Facilities, Generate Fuel Materials, Fabricate Fuels, Produce Energy, Recycle Materials, Manage Wastes. Add caveat.

Jess: What about non-radioactive mass flows? What about groups of materials.

Ed: Maybe this should be something more specific under the Analysis functions part that discusses how the analysis functions are served by the discreteness of the code.

Latif: Deployment analysis is probably another type of analysis we'll want, like how many reactors we need to meet demand, etc.

Ed: Uncertainty should be modeled, maybe not analysis. Where does it actually fit?

Latif: Market analysis probably isn't necessary.

Ed: After Operational Functions in section 5.2 you should have a little description of what we mean by operational functions.

Latif: Modules that rely on proprietary models will not be included in the stable distribution version of the code.

Ed: The challenge with the simple user includes propagating simple parameter specifications through completely.

Latif: The run time needs to be added to the specifics.

Brent: Get rid of existing HEU, say things like "secondary sources."

Jess: Don't be so specific with uranium, get rid of specificity or include Thorium or something like that.

11:45 SUPPORTED MISSIONS

Temi: Get rid of use cases and replace it with use-cases.

Latif: Transportation needs to be included as a type of study.

J'Tia This is a little detailed, but it is perhaps better up front. Take the paragraphs and put them in the beginning. Maybe this needs to be in the front with the user descriptions.

Temi: The bullets in the Facility studies section. . . need to think about text on V&V of contributor modules.

Latif: Be sure to say that these use cases are just examples. The idea is that this code needs to be very flexible, so you need to make it clear that people can add use cases they're interested in.

Ed: There is a category of cases that's missing. Compare single reactor A vs. single reactor B. Alternatively, compare three small reactors with one big reactor.

Brent: Particular uncertainty analyses might be a type of study? What about sensitivity? Is upset studies fit if uncertainty doesn't fit?

Ed: Maybe this should be Facility, System, Macro, etc. . .

J'Tia: What about perturbation instead of upset or disruption. It seems like facility studies are more specific, but strategy studies are combinations of facility deployments.

Latif: Economic analysis and cost analysis need to be specified much earlier. Local multiregional analyses need to be added to 'global' multiregional.

12:30 TECHNICAL AND FUNCTIONAL REQUIREMENTS

Brent: Before we look at individual requirements, let's make sure we get the scope, structure, etc. What are the thoughts here?

Latif: Start with the architecture. This is the most distinctive. It will make more sense if we put the modularity and such first rather than the specific analysis types. Coupling with other tools, user interface, etc, are probably part of the architecture. Maybe you should make a separate category out of architecture. Have technical requirements with architecture etc. first, then functional requirements (models, analyses, etc.) later.

Ed: In a real top down system you need to have some more generic types. Make sure that everything in the high level according to the philosophy in the beginning needs to be a requirement. Needs to have some more generic stuff that isn't so specific right away. Try to insert a middle level. 7.1 for example should say something like 'The FCS shall be able to calculate mass flow.' Everything you've already said up in the front matter needs to be addressed in the requirements.

Latif: Be sure to add to these everything we've added in section 5.

Temi: The appendix needs to be referred to in the front matter. You need to be less specific about processes. We need to be careful about this discussion of export control. It might be annoying to do a review every time you add a module.

Ed: Make sure each module is export controlled individually. Maybe there should also be a section that explains how this is a grand challenge exactly. Summarize how you think the big challenges present themselves.

Temi: The requirement of export control compliance is incredibly important. I think we need to state it non-ambiguously. In fact, it should be first, and so does intellectual property.

Temi: Is SQL free?

Katy: Yes.

Ed: Be careful about specifics in terms of data in FCS requirements and their references. Too specific.

Latif: Also, should not say source code when you mean data libraries.

Temi: Need an acronym table. Also, what is copyleftness.

J'Tia: To avoid the repeated use of models and uses of model might be avoided by the word simulations.

All: Some of these specifics need to be moved into basis statements in the facility data section. There is too much data there.

Jess: Is there a fundamental subset of separations processes or fuel cycles that we should require in this basic functionality?

Brent/Ed: What about a set of separations facilities which you can set up the separations facility as a set of stages of separator black boxes. That is, limit a separations black box to like 10 streams, and then chain them together.

Jess: Maybe there should be a requirement that says there shall be no hard limits in this code.

Latif: Do we want the ability to draw from scratch an entire fuel cycle?

Brent: This is related to level of user sophistication. This may require some specific input checking.

Jess: This gets back to a question we've dealt with. There's some need to do some material routing with a full knowledge of the facilities available.

Ed: How much lead time do we need to have in this model.

Brent: What if the code had a mode that relied on the user to answer some questions to help with simulation logic?

2:30 PATH FORWARD

Temi: But, we just got started!

Brent: We have some amount of funding that will help address some of these details. Then, we'll see if we can get the momentum to get the code to start driving this. This will take a lot of careful thought, well more than a $100,000 investment. This discussion has been really helpful, and I think we'll be able to deliver something much more impressive than anything they expect.

Temi: Is there no need to write something about when this grand challenge should be met?

Brent: We'll have to determine phases of development, define core audiences, etc. first. Laying out this path forward should be discussed, but putting in a timeline is probably infeasible.

Temi: Is it 20 years? Tomorrow?

Jess: Some phases and whatnot need to be indicated in an implementation section, but timelines wouldn't fit here. Some priorities need to be defined specifically.

J'Tia: Grand challenges require grand funding.

Ed: Some actual examples of GUIs that currently work in some other area might be good.