SAND2009-6304P

Waste Forms and Systems Integrated Performance and Safety Codes

System Design Specification

Advanced Fuel Cycle Initiative

Prepared for
U.S. Department of Energy

Advanced Modeling and Simulation Campaign

Sandia National Laboratories

September 2009
AFCI-XXX-XXX-XX-XX-2009-XXXXXX

Reviewed by:

Acting Director, Fuel Cycle Research and Development

	Robert Price	**Date**

Concurred by:

Director, AFCI Technical Integration Office

	Phillip Finck	**Date**

Approved by:

Deputy Assistant Secretary, Fuel Cycle Management

(AFCI Program Manager)	**Paul Lisowski**	**Date**

SUMMARY

The objective of the Fuel Cycle Research and Development (FCR&D) Waste Forms (WF) Integrated Performance and Safety Codes (IPSC) is to provide an integrated suite of computational modeling and simulation capabilities for simulation of the performance of waste forms in the engineered environment of a long-term disposal repository or waste storage facility. This suite will include first-principles codes for property characterization, high-fidelity modeling of coupled transport phenomena, and a set of efficient surrogate models that have confirmed accuracy in well-specified performance assessment regimes. These surrogate models will be based on abstractions of the higher fidelity models. The surrogate models will enable production of simulation results with which quantified predictions can be made. The ultimate goal is to support predictive simulation-based, risk-informed decision making about managing future US nuclear waste.

The WF IPSC will be developed using state-of-the-art software quality engineering practices leading to high-confidence software components. A special purpose thermal-hydrological-chemical-mechanical (THCM) multi-physics framework will be developed to support coupling of high-fidelity models and/or surrogate models for simulation of key phenomenological processes.

This report provides initial phenomena identification and ranking tables (PIRTs) and use cases based upon Sandia National Laboratories' (SNL's) extensive repository simulation and analysis experience with the Yucca Mountain Project (YMP) and Waste Isolation Pilot Plan (WIPP). A preliminary THCM framework architecture is developed from these requirements, use cases, and survey of similar or applicable software packages. A high-level overview of the planned software engineering environment is given based upon SNL's rigorous experience with software quality engineering within the Advanced Simulation and Computing (ASC) program. Finally, a summary list existing software components for potential use in the THCM framework is given.

CONTENTS

1. Introduction 1

2. Programmatic Drivers and Requirements 4

3. Phenomena Identification and Ranking 6
   3.1 Disposal System Designs and Scenarios 6
   3.2 Phenomena Identification 8
   3.3 Importance Ranking 27
   3.4 State of Knowledge Ranking 28

4. Verification, Validation, and Uncertainty Quantification Strategy 31
   4.1 The strategy: requirements to VV-UQ to feedback 31
   4.2 VV-UQ through the modeling hierarchy 34
   4.3 Sub-continuum materials properties characterization 35
   4.4 Upscaling from subcontinuum to continuum models 37
   4.5 The high-fidelity THCM layer in the hierarchy 38
   4.6 Upscaling from high-fidelity continuum models to the surrogate models 40
   4.7 The surrogate models: The Integrated assessment code 40
   4.8 Summary 41

5. Use Cases and Utilization Methodology 42
   5.1 Performance Assessment with Surrogate Models 42
   5.2 Continuum Analysis and Abstraction to Surrogate Models 44
   5.3 Sub-continuum Analysis and Upscaling to Continuum Models 46

6. THCM Framework Architecture 51
   6.1 Inter-fidelity Coupling Model Database 52
   6.2 Multi-physics coupling framework 53
   6.3 Workflow framework 65

7. Software Engineering Environment 76
   7.1 Project Management 77
   7.2 Requirements and Design 78
   7.3 Configuration Management 80
   7.4 Build and Test 82
   7.5 Internal Software Integration 83
   7.6 External Software Support and Collaboration Models 83
   7.7 Release and Distribution 86
   7.8 Training 87
   7.9 User support 91

8. References 92

A. Phenomena Identification and Ranking Table (PIRT) 95

B. Detailed Use Cases 128

C. Potential Frameworks, Tools, and Simulations 167
C.1 Salome 167
C.2 SIERRA 169
C.3 Trilinos 170
C.4 Dakota 174
C.5 ITAPS 174
C.6 CUBIT 175
C.7 CCA 175
C.8 GoldSim 175
C.9 WIPP Baseline Software List 175
C.10 YMP Baseline Software List 178

FIGURES

Figure 1. Simulation Domain of Waste Forms and Systems Integrated Performance and Safety Codes 1

Figure 2. Three Levels of Fidelity for WF IPSC Simulations and their Interrelationships 2

Figure 3. Key Disposal System Components 7

Figure 4. Construction of a performance assessment (PA) system from surrogate models. The relationship of surrogate models to high-fidelity and sub-continuum models are also indicated. The PA system repeatedly reads the input parameter values from and saves the simulation results to the databases during the model execution. 44

Figure 5. The THCM framework architecture is partitioned into analysis workflow, multi-physics coupling, and inter-fidelity coupling component frameworks. 51

Figure 6: A few examples of different mathematical problems that are supported by the ModelEvaluator interface. 62

Figure 7: ModelEvaluator as an interface between solvers (MOOCHO, Rythmos, NOX, LOCA) and applications/models (Charon, Aria/SIERRA) as well as relationship to linear solver and preconditioning capabilities in Trilinos exposed through the Stratimikos package. 63

Figure 8 Generalized conceptual model for workflow including user interaction, pre-processing, component computations, storage, and post-processing 66

Figure 9 A high-level collaboration diagram between workflow framework components supporting an analysis workflow 67

Figure 10 Anticipated architectural components of the workflow framework 70

Figure 11 Example architecture to illustrate potential utilization of existing analysis workflow and data management services 74

Figure 12: ModelEvaluator interfaces, adapters, and related software in Trilinos. 171

Figure 13: A COMPOSITE ModelEvaluator subclass for a multi-physics set of models. 172

Figure 14: Basic design of Rythmos for solving transient models. 173

TABLES

Table 1. YMP FEPs Not Relevant to the WF IPSC 9

Table 2. YMP FEPs Relevant to the WF IPSC 10

Table 3. Preliminary List of Phenomena 11

Table 4 Importance Ranking Scheme 27

Table 5. State of Knowledge Ranking Scheme 29

Table 6. Model Adequacy Ranking Scheme 29

Table 7. Data Adequacy Ranking Scheme 29

Table 8. Likelihood of Obtaining Additional Information Ranking Scheme 30

Table 9 Workflow framework requirements mapped to steps in an analysis workflow 68

ACRONYMS

AD	Automatic Differentiation
AFCI	Advanced Fuel Cycle Initiative
AFCI QAPD	Advanced Fuel Cycle Initiative Quality Assurance Program Document
ASC	Advanced Simulation and Computing
CCA	Common Component Architecture
CS&E	Computational Science & Engineering
DA	Decision Analysis
DAE	Differential Algebraic Equations
FCR&D	Fuel Cycle Research & Development
FEP	Features, Events, and Processes
FMM	Fundamental Methods and Models
IPSC	Integrated Performance and Safety Codes
LHS	Latin Hypercube Sampling
NEAMS	Nuclear Energy Advanced Modeling and Simulation
ODE	Ordinary Differential Equations
PA	Performance Assessment
PCE	Polynomial Chaos Expansion
PDE	Partial Differential Equations
PIRT	Phenomenon Identification and Ranking Table
SA	Sensitivity Analysis
SC	Stochastic Collocation
SE	Software Engineering
SQE	Software Quality Engineering
THCM	Thermal, Hydrologic, Chemical, and Mechanical
UQ	Uncertainty Quantification
VV or V&V	Verification and Validation
WF	Waste Form
WIPP	Waste Isolation Pilot Plant
YMP	Yucca Mountain Project

FUEL CYCLE RESEARCH AND DEVELOPMENT MODELING AND SIMULATION CAMPAIGN WASTE FORMS AND SYSTEMS INTEGRATED PERFORMANCE AND SAFETY CODES

1. Introduction

The objective of the Fuel Cycle Research and Development (FCR&D) Waste Forms (WF) Integrated Performance and Safety Codes (IPSC) is to provide an integrated suite of computational modeling and simulation capabilities for simulation of the long-term performance of waste forms in the engineered environment of a waste storage or disposal repository (Figure 1). This suite will include first-principles codes for property characterization, high-fidelity modeling of coupled degradation and transport phenomena, and a set of efficient surrogate models that have confirmed accuracy in well-specified performance assessment regimes. These surrogate models will be based on abstractions of the higher fidelity models. The surrogate models will enable production of simulation results with which quantified predictions can be made. The ultimate goal is to support predictive simulation-based, risk-informed decision making about managing future US nuclear waste.

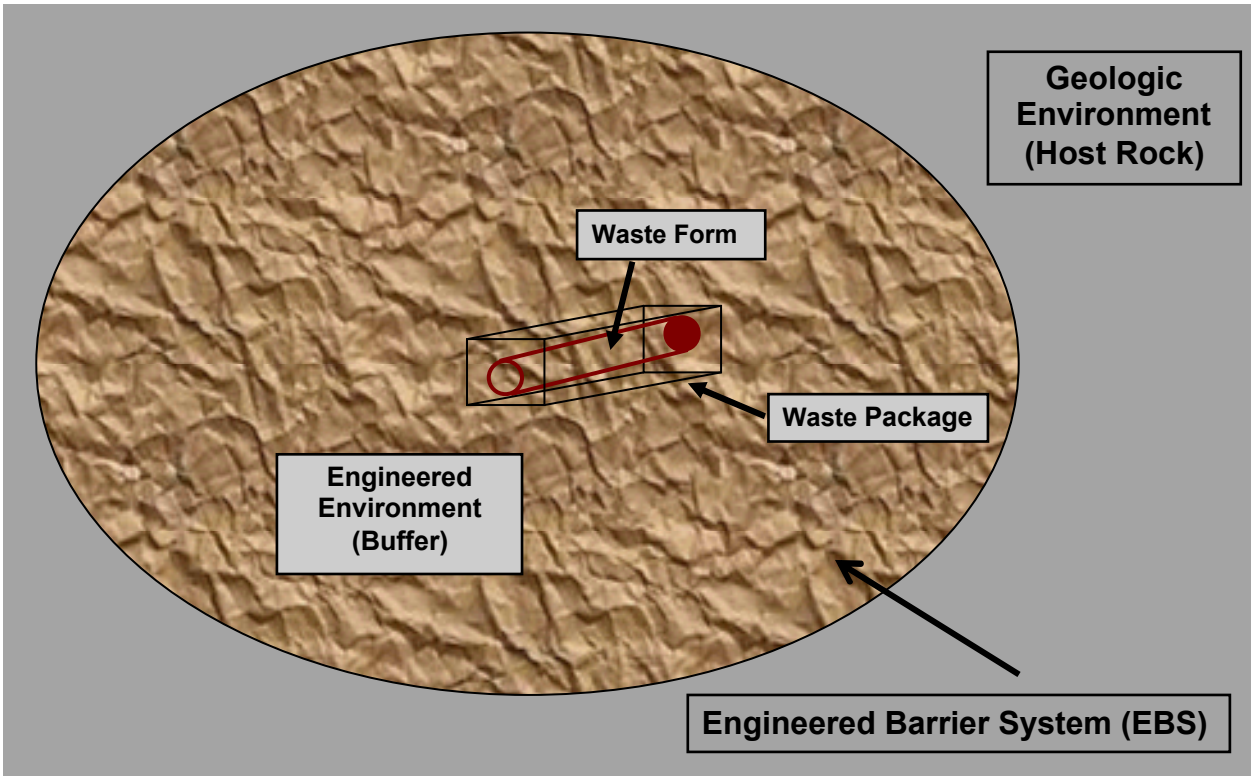

Figure 1. Simulation Domain of Waste Forms and Systems Integrated Performance and Safety Codes

Three levels of simulation fidelity will be used in the WF IPSC, as illustrated in Figure 2. Maximum-fidelity sub-continuum simulations will be used in conjunction with experimental data to characterize material properties and mechanistic processes. It is anticipated that the Fundamental Methods and Models (FMM) component of the FCR&D Modeling and Simulation Campaign will provide many required sub-continuum simulation capabilities. Results of coordinated sub-continuum simulations and experimental investigations will be used to develop and verify high-fidelity continuum physics models. High-fidelity continuum physics models will be integrated to investigate coupled multi-physics (i.e., thermal-hydrologic-chemical-mechanical (THCM)) processes. Surrogate simulation components are abstracted from the high-fidelity simulations to be "robust and fast" for performance and design assessment analyses over large numbers of waste forms and environment realizations. These surrogate simulations will be verified against the corresponding high-fidelity simulations.

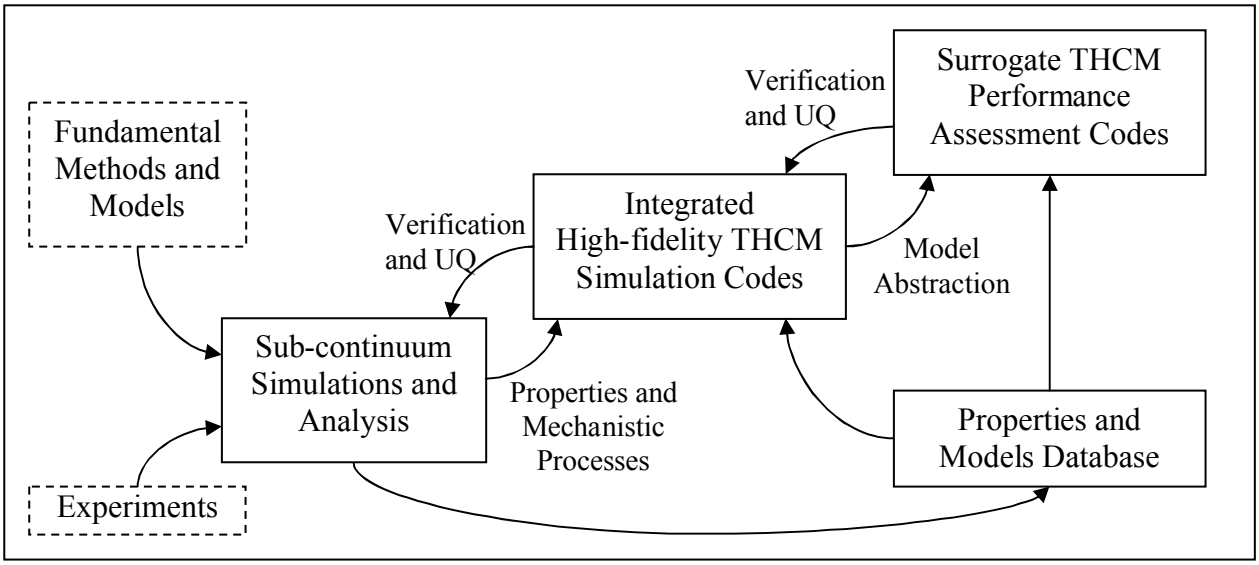

Figure 2. Three Levels of Fidelity for WF IPSC Simulations and their Interrelationships

Sub-continuum simulations and experiments are used to characterize material properties and mechanistic processes significant to waste forms in repository environments. In particular, to predict key state and transport properties is needed to specify constitutive relationships for high-fidelity continuum models. These properties are needed to model the evolving isotopic composition within the waste form matrix and along repository environment pathways.

The integrated high-fidelity THCM simulation codes are used to analyze multi-physics coupling of macro-scale models for thermal, hydrological, chemical, and mechanical processes of waste forms in repository environments. The results of these simulations will be used to identify the relevant governing equations, constitutive models, and accuracies required for abstracting waste form surrogate performance assessment models.

The surrogate THCM performance assessment simulations are expected to have a collection of alternative modeling approaches, ranging from using high-fidelity continuum models with reduced-dimension realizations of the waste form and environment to using abstracted models that are simple calibrated response-surface functions. The objective of the surrogate models is to be self-contained modules that are flexibly linked together to simulate specified waste forms and repositories to a needed accuracy. A primary computational efficiency objective of performance assessment codes is to efficiently simulate hundreds or thousands of waste form and repository realizations.

Both the integrated high-fidelity THCM simulation codes and the surrogate THCM performance assessment codes will have embedded sensitivity analysis capabilities to support verification, validation, uncertainty quantification (UQ), and design optimization analyses. Development of embedded sensitivity analysis capability will be in collaboration with the Verification, Validation, and Uncertain Quantification (VU) component of the FCR&D Modeling and Simulation Campaign.

A special purpose thermal-hydrological-chemical-mechanical (THCM) multi-physics framework will be developed to support coupling of high-fidelity simulation components and/or surrogate simulation components for analysis of key phenomenological processes. Modular simulation components will plug in to the THCM framework through well-defined interfaces. It will be essential for this framework to ensure transparent, traceable, reproducible, and retrievable simulation results in order to satisfy regulatory compliance requirements such those associated with the Yucca Mountain Project (YMP) and Waste Isolation Pilot Plant (WIPP).

The WF IPSC will use software quality engineering (SQE) best-practices to develop high-confidence software components, coordinate large distributed development teams, and respond to evolving requirements.

2. Programmatic Drivers and Requirements

The FCR&D program integrates small-scale experiments, theory development, and advanced modeling and simulation to provide a more complete understanding of underlying science supporting the development of advanced fuel cycle technologies and waste management options and thereby provide a sound basis for future decision making. Long-term objectives of the FCR&D program include:

- improving waste storage and disposal options,
- promoting the safe and secure management of nuclear fuel and waste products,
- minimizing the proliferation risk of the civilian nuclear fuel cycle, and
- reducing the timeframe for managing waste from many hundreds of thousands of years (geologic time-scales) to centuries (engineering time-scales).

The WF IPSC is one of several IPSCs that comprise the Advanced Modeling and Simulation (AMS) Campaign of the FCR&D program. The broad objective of the AMS Campaign is to apply state-of-the-art computing capabilities to develop simulation tools for addressing the behavior of nuclear technologies in realistic situations. Specific objectives of the AMS Campaign that are relevant to the WF IPSC include:

- developing code architectures and methods to model the performance of advanced waste forms in adverse geological environments for very long-term storage and disposition,
- delivering Fundamental Models and Methods that will allow the understanding of performance of materials on the lower length scales needed to simulate the performance of integrated systems,
- developing a set of experimental data needs and requirements over the entire spectrum of time and length-scale for the models, and
- developing the set of validation techniques necessary for demonstrating the quality of the modeling tools and for defining requirements for further development of these tools.

The WF IPSC also overlaps with the Waste Forms Campaign of the FCR&D program. The Waste Forms Campaign builds on recommendations outlined by Peters et al. [1] in support of the Global Nuclear Energy Partnership (GNEP) and the Advanced Fuel Cycle Initiative (AFCI). The broad objective of the Waste Forms Campaign is to develop waste forms suitable for disposal in a future geologic repository that meet U.S. environmental requirements for future systems. Certain long-lived fission products can be significant contributors to the long-term environmental effects of used fuel in specific geological environments, and separation of these elements for incorporation into new waste forms for safe disposal is needed. In order to decrease the volume of high level wastes, while maintaining durability, research is also needed in advanced glasses and metal waste form compositions and waste loadings.

The combination of FCR&D, AMS, and Waste Forms objectives described above provides long-term programmatic guidance for the WF IPSC, leading to the WF IPSC broad objective stated in Section 1, which is to provide an integrated suite of computational modeling and simulation capabilities for simulation of the performance of waste forms in the engineered environment of a long-term disposal repository or waste storage facility. The development of these advanced modeling and simulation capabilities of the WF IPSC provides a unifying approach to evaluate the complex behavior of waste forms over a range of conditions and settings within the context of total system performance.

Total system performance includes consideration of (1) current and potential future waste forms that provide a source of radionuclides, (2) the engineered and natural barriers that will influence migration of the radionuclides to human receptors, (3) processes and associated uncertainties that occur over a broad range of time scales (from nanoseconds to millions of years) and distances (from angstroms to

kilometers), and (4) the highly coupled nature of THCM multi-physics processes. These considerations are largely interrelated and require close coupling between theory, experiment, and modeling efforts, which necessitates co-ordination with other FCR&D campaigns.

The remainder of this document outlines the initiation of a multi-year plan to develop the WF IPSC suite of modeling and simulation capabilities to satisfy the above requirements. Section 3 describes the preliminary identification of the range of potential waste forms, repository designs, geologic settings, and relevant phenomena that define the scope of total system to be assessed by the WF IPSC. Section 4 describes the requirements and plan for uncertainty quantification (UQ) and verification and validation (VV) to demonstrate quantified confidence in the results across the full hierarchy of WF IPSC computer codes and in all data flow between the different hierarchies of simulations. Section 5 describes preliminary use cases for each of the three levels of simulation fidelity. The use cases define the preliminary requirements of the software system and modeling framework, based on the relevant phenomena and identified model scope (from Section 3) and the UQ and VV requirements (from Section 4). Section 6 describes the requirements and plan for a model framework architecture, based on the VV-UQ requirements and use cases, that includes consideration of inter-fidelity coupling, multi-physics coupling, and a workflow framework. Section 7 provides the requirements and a high-level overview for a software engineering environment for development of the WF IPSC.

3. Phenomena Identification and Ranking

As stated in Section 1, the objective of the WF IPSC is to provide an integrated suite of computational modeling and simulation capabilities for simulation of the performance of waste forms in an engineered environment. The suite of capabilities includes three levels of simulation fidelity: sub-continuum scale characterization of material properties and mechanistic processes; continuum-scale high-fidelity coupled THCM models; and surrogate THCM models for performance assessment (PA) analyses. This section documents the development of a preliminary Phenomena Identification and Ranking Table (PIRT) to identify the relevant phenomena necessary for the WF IPSC model suite to have the flexibility to simulate the performance of a range of current and potential future HLW waste forms, design and storage options, and geologic settings over very long timeframes.

The PIRT approach [27] is an iterative process that evolves and is updated as new information (new research, data, and/or model results) becomes available. In addition to identifying fundamental technical issues that should be addressed, the PIRT approach is also useful for determining and prioritizing which technical issues would benefit from additional research before developing final solutions. The WF IPSC PIRT draws upon Sandia National Laboratories' (SNL's) extensive experience developing and evaluating features, events, and processes (FEPs) for the Yucca Mountain Project (YMP) [31] and Waste Isolation Pilot Plant (WIPP) [32] and being involved with the International FEP Database [30]. The preliminary PIRT represents an initial identification of potentially relevant phenomena, based on selected design options. As the associated theory, experiment, and modeling efforts mature, the PIRT will be continually refined.

The preliminary PIRT was developed using the following steps:

1. Identify disposal system designs and scenarios (including temporal phases and spatial components)
2. Identify phenomena (including associated processes and parameters, where applicable)
3. Identify the importance ranking for each phenomena
4. Identify the state of knowledge for models and data and the likelihood of obtaining new information

3.1 Disposal System Designs and Scenarios

A schematic representation of a generic long-term geologic repository is shown in Figure 3. The engineered barrier system (EBS) is a subsurface excavation that generally contains waste forms, surrounded by waste packages, surrounded by a buffer region. The EBS is in turn surrounded by the host rock geologic setting. The buffer region may contain a number of engineered features such as backfill, excavation liners, seals, and waste package support structures.

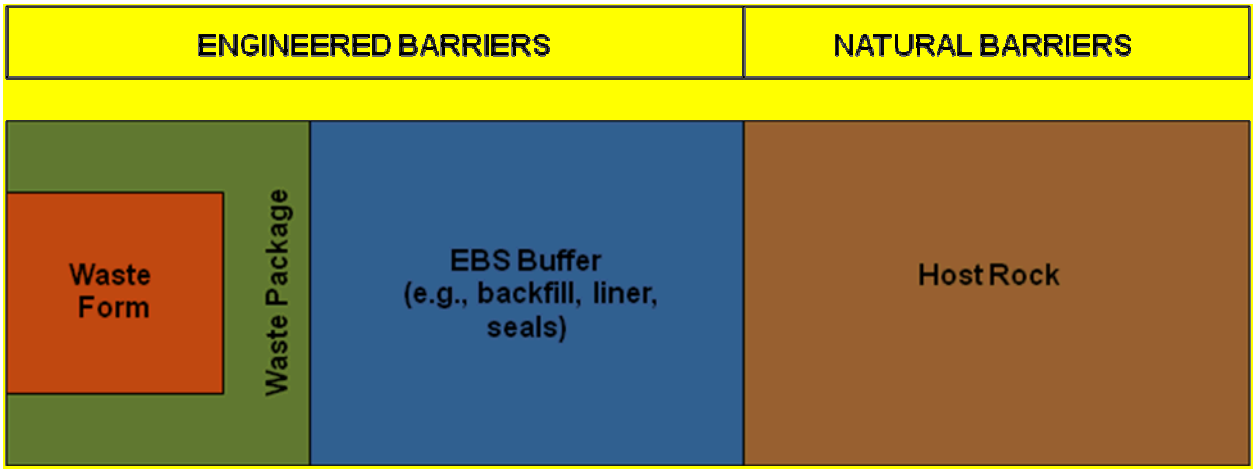

Figure 3. Key Disposal System Components

The performance of a geologic repository system typically relies upon multiple barriers (i.e., defense in depth) to limit radionuclide release through the engineered and geologic environments to a human receptor over timeframes as long as a million years. The four disposal system components shown in Figure 3 are the most significant contributors to limiting radionuclide transport. While the FCR&D program is focused on waste form composition and performance, the WF IPSC must also consider the performance of the other three components.

Features of the four key disposal system components that are likely to have the greatest effect phenomena identification and ranking are as follows:

- Waste Form Material
  - HLW borosilicate glass (fission products and trace actinides)
  - HLW crystalline ceramic (fission products and trace actinides)
  - HLW metal alloy (transition metal fission products)
  - SNF with cladding (actinides and fission products) [SNF is not considered as a WF option for FY09]
- Waste Package Material
  - Corrosion resistant with some structural strength
  - No corrosion resistance or structural strength
- EBS Buffer Material
  - Clay (e.g., bentonite) backfill
  - Host rock backfill
  - No backfill
  - Some cementitious materials present (e.g., liner or grout)
- Host Rock Geologic Environment
  - Host rock type (crystalline, clay, salt)
  - Local exposure conditions (chemically reducing - below the water table, or chemically oxidizing – above the water table)

The performance of a waste disposal system is strongly dependent on the EBS design features and on the local exposure conditions. These design features and exposure conditions are in turn affected by the interface regions between the features, the coupled THCM processes, and the geologic setting. Therefore, the identification and ranking of phenomena is dependent upon the selected disposal scenario (the combination of EBS design and geologic setting). At the same time, the WF IPSC must maintain the flexibility to simulate a wide range of scenarios.

To simplify the preliminary phenomena identification and ranking, a single reference scenario was selected, consisting of borosilicate glass waste forms, corrosion resistant waste packages, clay backfill, and crystalline host rock below the water table with chemically reducing conditions. However, design-specific phenomena and/or ranking considerations that may not be apparent in the reference scenario are still noted in the preliminary PIRT. In future years, additional details of the phenomena for all scenarios will be identified and evaluated.

For FY09, the WF IPSC considers only the EBS (waste forms, waste packages, and buffer region). The geologic environment (including the excavation disturbed zone (EDZ)) contributes only in the form of boundary conditions to the EBS. However, the framework of the WF IPSC is flexible enough to accommodate explicit modelling of the geologic environment (and SNF) if necessary in future years. The WF IPSC framework is also flexible enough to simulate interim storage in an above ground container/cask.

3.2 Phenomena Identification

The preliminary identification of WF IPSC phenomena was based on information gleaned from decades of FEP development for long-term geologic disposal of HLW and SNF. A FEP is typically a process or event acting on a feature. Phenomena in a PIRT are similar to FEPs. An international FEP database [30], which catalogs FEP lists from radioactive waste disposal programs in several countries, contains several thousand entries relevant to a number of waste forms, EBS designs, and geologic settings. The Yucca Mountain Project (YMP) FEP list [31] was developed from the international FEP database, and is therefore a comprehensive summary of potential waste disposal phenomena that implicitly captures the thousands of the FEPs from the international database.

The preliminary list of WF IPSC phenomena was developed by (1) examining the list of 374 YMP FEPs, (2) reviewing FCR&D and AFCI planning documents, and (3) brainstorming by WF IPSC subject matter experts. Preliminary phenomena identification was performed at a coarse level-of-detail, consistent with the level-of-detail that might be simulated with the surrogate PA models. Finer levels-of-detail, such as might be simulated with the high-fidelity models are identified as associated processes to the high-level phenomena. The details of the development of the preliminary phenomena and associated processes generally applicable to simulations with high-fidelity models and surrogate PA models are described in Section 3.2.1. A discussion of some potentially important sub-continuum processes, having an even finer level-of-detail, is provided in Section 3.2.2.

3.2.1 High-Fidelity and Surrogate Model Phenomena

The first step in phenomena identification was to identify those YMP FEPs that were not relevant to the preliminary scope of the WF IPSC.

Table 1 identifies classes of FEPs that were considered beyond the scope of the preliminary WF IPSC, totaling 148 not-relevant FEPs. If the scope of the WF IPSC is expanded (e.g., to include the host rock and/or alternate designs) then the not-relevant FEPs will be re-evaluated for relevance.

Table 1. YMP FEPs Not Relevant to the WF IPSC

Designator	Number of YMP FEPs	Rationale
N – FF	97	FEPs associated with far-field (FF) processes and characteristics (i.e., in the geosphere and biosphere). The WF IPSCs are currently limited to EBS and near-field processes.
N – DE	11	FEPs associated with igneous and/or seismic disruptive events (DE) that only impact far-field (FF) processes and characteristics.
N – Design	4	FEPs associated with far-field (FF) processes that are specific to features unique to the YMP geologic setting.
N – Design: Drip Shield	15	FEPs associated with an EBS feature (drip shield) that is unique to the YMP design.
N – Design: Pallet	2	FEPs associated with an EBS feature (waste package emplacement pallet) that is unique to the YMP design.
N – HI	8	FEPs associated with specific details of human intrusion (HI) not required for the WF IPSCs. Generic human intrusion phenomena are included in the WF IPSC list.
N – SYS	11	FEPs associated with system-level (SYS) details of repository system as whole not required for the WF IPSCs.
Total	**148**	

The second step in phenomena identification was to categorize the remaining 226 YMP FEPs that were relevant to the WF IPSCs. The categories are identified in Table 2. Table 2 indicates that many of the relevant FEPs are not directly modeled by the WF IPSC, but must be considered in the specification of boundary conditions. As with the not-relevant FEPs, the categorization of these relevant boundary-condition FEPs will be re-evaluated if the scope of the WF IPSC is expanded. Table 2 also indicates the number of WF IPSC phenomena derived from the relevant FEPs. Some of the YMP FEPs were combined to create a set of phenomena with a consistent level-of-detail; this accounts for the smaller number of phenomena. However, the 92 WF IPSC phenomena represent the same associated processes as the YMP FEPs from which they derived.

Table 2. YMP FEPs Relevant to the WF IPSC

Designator	Number of YMP FEPs	Number of WF IPSC Phenomena	Rationale
Y	131	84	FEPs directly applicable to the WF IPSCs.
Y – BC	34	0	FEPs associated with THCM processes in the near-field (e.g., temperature, mechanical alteration, rate and chemistry of inflowing water). These FEPs are not directly modeled by the WF IPSCs, but must be considered in the specification of boundary conditions.
Y – IC	8	0	FEPs associated with THCM processes during the pre-closure in the EBS and near-field. These FEPs are not directly modeled by the WF IPSCs, but must be considered in the specification of initial conditions.
Y – DE	15	2	FEPs associated with igneous or seismic disruptive events (DE) that impact the EBS and near-field spatial domains.
Y – Design	6	5	FEPs associated with specific design features (seals, rock reinforcement materials, copper) that may or may not be part of the design scenarios.
Y – NF	10	0	FEPs associated with THCM processes in the EBS that have impacts on the near-field (NF). These NF impacts are not directly modeled by the WF IPSCs, but may result in changes to the boundary conditions.
Y – FF	9	0	FEPs associated with THCM processes in the EBS that have impacts on the far-field (FF). These FF impacts are not directly modeled by the WF IPSCs, but must be considered if the WF IPSCs are expanded or coupled to a far-field model.
Y – HI	4	1	FEPs associated with human intrusion (HI) that impact the EBS and near-field spatial domains.
Y – SYS	9	0	FEPs associated with system-level (SYS) impacts on the EBS and near-field. These FEPs are not directly modeled by the WF IPSCs, but must be considered in the specification of scenarios and/or initial conditions.
Total	**226**	**92**	

Note: Classes of FEPs that translated to “0” phenomena, may be translated in future years as the scope of the WF IPSC increases.

The preliminary set of 92 WF IPSC phenomena and associated processes, derived from the FEPs in Table 2, is listed in Table 3. Rankings are discussed in Section 3.3.1.

Table 3. Preliminary List of Phenomena

Phenomena Number	Phenomena	Associated Processes	Related FEP Number
0.0.00.00	ASSESSMENT BASIS		
1.0.00.00	EXTERNAL FACTORS		
1.1.00.00	1. REPOSITORY ISSUES		
1.2.00.00	2. GEOLOGICAL PROCESSES AND EFFECTS		
1.2.03.01	Seismic activity impacts EBS and/or EBS components	- Mechanical damage to EBS (from ground motion, rockfall, drift collapse, fault displacement) [see also Mechanical Impacts in 2.1.07.04, 2.1.07.05, 2.1.07.06, 2.1.07.07, 2.1.07.08, and 2.1.07.10]	1.2.02.03.0A 1.2.03.02.0A 1.2.03.02.0B 1.2.03.02.0C
1.2.04.01	Igneous activity impacts EBS and/or EBS components	- Mechanical damage to EBS (from intrusion intrusion) - Chemical interaction with magmatic volatiles - Transport of radionuclides (in magma, pyroclasts, vents) [see also Mechanical Impacts in 2.1.07.04, 2.1.07.05, 2.1.07.06, 2.1.07.07, and 2.1.07.08]	1.2.04.03.0A 1.2.04.04.0A 1.2.04.04.0B 1.2.04.05.0A 1.2.04.06.0A
1.3.00.00	3. CLIMATIC PROCESSES AND EFFECTS		
1.4.00.00	4. FUTURE HUMAN ACTIONS		
1.4.02.01	Human Intrusion - Deliberate - Inadvertent		1.4.02.01.0A 1.4.02.02.0A 3.3.06.01.0A
1.5.00.00	5. OTHER		
2.0.00.00	DISPOSAL SYSTEM FACTORS		
2.1.00.00	1. WASTES AND ENGINEERED FEATURES		
2.1.01.00	1.01. INVENTORY		
2.1.01.01	Waste Inventory - Radionuclides - Non-Radionuclides	- Composition - Enrichment / Burn-up	2.1.01.01.0A
2.1.01.02	Radioactive Decay and Ingrowth		3.1.01.01.0A
2.1.01.03	Heterogeneity of Waste Inventory - Waste Package Scale - Repository Scale	- Composition - Enrichment / Burn-up - Damaged Area	2.1.01.03.0A 2.1.01.04.0A
2.1.01.04	Interactions Between Co-Located Waste		2.1.01.02.0A 2.1.01.02.0B
2.1.02.00	1.02. WASTE FORM		

Phenomena Number	Phenomena	Associated Processes	Related FEP Number
2.1.02.01	SNF (Commercial, DOE) Degradation - Alteration / Phase Separation - Dissolution / Leaching - Radionuclide Release	Degradation is dependent on: - Composition - Geometry / Structure - Enrichment / Burn-up - Surface Area - Gap and Grain Fraction - Damaged Area - THC Conditions [see also Mechanical Impact in 2.1.07.06 and Thermal-Mechanical Effects in 2.1.11.06]	2.1.02.02.0A 2.1.02.01.0A 2.1.02.28.0A 2.1.02.07.0A
2.1.02.02	HLW (Glass, Ceramic, Metal) Degradation - Alteration / Phase Separation - Dissolution / Leaching - Cracking - Radionuclide Release	Degradation is dependent on: - Composition - Geometry / Structure - Surface Area - Damaged / Cracked Area - Mechanical Impact - THC Conditions [see also Mechanical Impact in 2.1.07.07 and Thermal-Mechanical Effects in 2.1.11.06]	2.1.02.03.0A 2.1.02.05.0A
2.1.02.03	Degradation of Organic/Cellulosic Materials in Waste	[see also Complexation in EBS in 2.1.09.17]	2.1.02.10.0A
2.1.02.04	HLW (Glass, Ceramic, Metal) Recrystallization		2.1.02.06.0A
2.1.02.05	Pyrophoricity or Flammable Gas from SNF or HLW		2.1.02.08.0A 2.1.02.29.0A
2.1.02.06	SNF Cladding Degradation and Failure	- Initial damage - General Corrosion - Microbially Influenced Corrosion - Localized Corrosion - Enhanced Corrosion (silica, fluoride) - Stress Corrosion Cracking - Hydride Cracking - Unzipping - Creep - Internal Pressure - Mechanical Impact	2.1.02.11.0A 2.1.02.12.0A 2.1.02.13.0A 2.1.02.14.0A 2.1.02.15.0A 2.1.02.16.0A 2.1.02.17.0A 2.1.02.18.0A 2.1.02.27.0A 2.1.02.21.0A 2.1.02.22.0A 2.1.02.23.0A 2.1.02.25.0A 2.1.02.25.0B 2.1.02.19.0A 2.1.02.26.0A 2.1.02.20.0A 2.1.02.24.0A 2.1.09.03.0A
2.1.03.00	1.03. WASTE CONTAINER		
2.1.03.01	Early Failure of Waste Packages	- Manufacturing defects - Improper sealing	2.1.03.08.0A
2.1.03.02	General Corrosion of Waste Packages	- Dry-air oxidation - Humid-air corrosion - Aqueous phase corrosion - Passive film formation and stability	2.1.03.01.0A
2.1.03.03	Stress Corrosion Cracking (SCC) of Waste Packages	- Crack initiation, growth and propagation - Stress distribution around cracks	2.1.03.02.0A

Phenomena Number	Phenomena	Associated Processes	Related FEP Number
2.1.03.04	Localized Corrosion of Waste Packages	- Pitting - Crevice corrosion - Salt deliquescence	2.1.03.03.0A 2.1.09.28.0A
2.1.03.05	Hydride Cracking of Waste Packages	- Hydrogen diffusion through metal matrix - Crack initiation and growth in metal hydride phases	2.1.03.04.0A
2.1.03.06	Microbially Influenced Corrosion (MIC) of Waste Packages		2.1.03.05.0A
2.1.03.07	Internal Corrosion of Waste Packages Prior to Breach		2.1.03.06.0A
2.1.03.08	Flow In and Through Waste Packages	- Saturated / Unsaturated flow - Movement as thin films or droplets [see also Flow in EBS in 2.1.08.01]	2.1.03.10.0A 2.1.03.11.0A
2.1.03.09	Evolution Flow Pathways in Waste Packages	- Evolution of physical form of waste package - Plugging of cracks in waste packages [see also Evolution of Flow Pathways in EBS in 2.1.08.02, Mechanical Impact on Waste Packages in 2.1.07.05]	2.1.03.10.0A 2.1.03.11.0A
2.1.04.00	1.04. BUFFER / BACKFILL		
2.1.04.01	Evolution and Degradation of Backfill	- Alteration - Thermal expansion / Degradation - Swelling / Compaction - Erosion / Dissolution - Evolution of backfill flow pathways [see also Evolution of Flow Pathways in EBS in 2.1.08.02, Mechanical Impact on Backfill in 2.1.07.04, Thermal-Mechanical Impact in 2.1.11.08, Chemical Interaction 2.1.09.06]	2.1.04.05.0A 2.1.04.03.0A
2.1.04.02	Flow in Backfill	- Fracture / Matrix flow [see also Flow in EBS in 2.1.08.01]	2.1.04.01.0A
2.1.05.00	1.05. SEALS		
2.1.05.01	Degradation of Seals	- Alteration / Degradation / Cracking - Erosion / Dissolution [see also Mechanical Impact in 2.1.07.04, Thermal-Mechanical Impact in 2.1.11.09, Chemical Interaction 2.1.09.08]	2.1.05.03.0A
2.1.05.02	Flow Through Seals	[see also Flow in EBS in 2.1.08.01]	2.1.05.01.0A
2.1.06.00	1.06. OTHER EBS MATERIALS		
2.1.06.01	Degradation of Liner / Rock Reinforcement Materials in EBS	- Alteration / Degradation / Cracking - Corrosion - Erosion / Dissolution / Spalling [see also Mechanical Impact in 2.1.07.08, Thermal-Mechanical Impact in 2.1.11.09, Chemical Interaction 2.1.09.07]	2.1.06.02.0A
2.1.06.02	Flow Through Liner / Rock Reinforcement Materials in EBS	[see also Flow in EBS in 2.1.08.01]	2.1.06.04.0A

Phenomena Number	Phenomena	Associated Processes	Related FEP Number
2.1.07.00	1.07. MECHANICAL PROCESSES		
2.1.07.01	Rockfall	- Dynamic loading (block size and velocity)	2.1.07.01.0A
2.1.07.02	Drift Collapse	- Static loading (rubble volume) - Alteration of seepage - Alteration of EBS flow pathways - Alteration of EBS thermal environment [see also Evolution of Flow Pathways in EBS in 2.1.08.02, Chemical Effects of Drift Collapse in 2.1.09.12, and Effects of Drift Collapse on TH in 2.1.11.04]	2.1.07.02.0A 1.2.03.02.0D
2.1.07.03	Mechanical Effects of Backfill	- Protection of other EBS components from rockfall / drift collapse	2.1.04.04.0A
2.1.07.04	Mechanical Impact on Backfill	- Rockfall / Drift collapse - Hydrostatic pressure - Internal gas pressure [see also Degradation of Backfill in 2.1.04.01 and Thermal-Mechanical Effects in 2.1.11.08]	2.1.04.05.0A
2.1.07.05	Mechanical Impact on Waste Packages	- Rockfall / Drift collapse - Waste package movement - Hydrostatic pressure - Internal gas pressure - Swelling corrosion products [see also Thermal-Mechanical Effects in 2.1.11.05]	2.1.03.07.0A 2.1.07.04.0A 2.1.09.03.0B
2.1.07.06	Mechanical Impact on SNF Waste Form	- Drift collapse - Swelling corrosion products [see also Thermal-Mechanical Effects in 2.1.11.06]	2.1.07.02.0A 2.1.09.03.0B
2.1.07.07	Mechanical Impact on HLW Waste Form	- Drift collapse - Swelling corrosion products [see also Thermal-Mechanical Effects in 2.1.11.06]	2.1.07.02.0A 2.1.09.03.0B
2.1.07.08	Mechanical Impact on Other EBS Components - Seals - Liner/Rock Reinforcement Materials - Waste Package Support Materials	- Rockfall / Drift collapse - Movement - Hydrostatic pressure - Swelling corrosion products [see also Thermal-Mechanical Effects in 2.1.11.09]	2.1.07.02.0A 2.1.09.03.0C
2.1.07.09	Mechanical Effects at EBS Component Interfaces	- Component-to-component contact (static or dynamic)	2.1.06.07.0B
2.1.07.10	Mechanical Degradation of EBS	- Floor buckling - Fault displacement - Consolidation of EBS components - Degradation of waste package support structure - Alteration of EBS flow pathways [see also Evolution of Flow Pathways in EBS in 2.1.08.02, Degradation in 2.1.04.01, 2.1.05.01, and 2.1.06.01]	2.1.06.05.0B 2.1.07.06.0A 1.2.02.03.0A 2.1.08.15.0A

Phenomena Number	Phenomena	Associated Processes	Related FEP Number
2.1.08.00	1.08. HYDROLOGIC PROCESSES		
2.1.08.01	Flow Through the EBS	- Saturated / Unsaturated flow - Preferential flow pathways [see also Flow in Waste Packages in 2.1.03.08, Flow in Backfill in 2.1.04.02], Flow through Seals 2.1.05.02, Flow through Liner in 2.1.06.02, Thermal Effects on Flow in 2.1.11.10, Effects of Gas on Flow in 2.1.12.02]	2.1.08.09.0A 2.1.08.07.0A 2.1.08.05.0A
2.1.08.02	Alteration and Evolution of EBS Flow Pathways	- Drift collapse - Degradation/consolidation of EBS components - Plugging of flow pathways - Formation of corrosion products - Water ponding [see also Evolution of Flow Pathways in WPs in 2.1.03.09, Evolution of Backfill in 2.1.04.01, Drift Collapse in 2.1.07.02, and Mechanical Degradation of EBS in 2.1.07.10]	2.1.08.12.0A 2.1.08.15.0A 2.1.03.10.0A 2.1.03.11.0A 2.1.09.02.0A
2.1.08.03	Condensation Forms in Repository - On Tunnel Roof / Walls - On EBS Components	- Heat transfer (spatial and temporal distribution of temperature and relative humidity) - Dripping [see also Heat generation in EBS in 2.1.11.01, Effects on EBS Thermal Environment in 2.1.11.03 and 2.1.11.04]	2.1.08.04.0A 2.1.08.04.0B
2.1.08.04	Capillary Effects in EBS	- Wicking	2.1.08.06.0A
2.1.08.05	Influx (Seepage) Into the EBS	- Water influx rate (spatial and temporal distribution) [BOUNDARY CONDITION FOR WF IPSC]	2.1.08.01.0A
2.1.09.00	1.09. CHEMICAL PROCESSES - CHEMISTRY		
2.1.09.01	Chemistry of Water Flowing into the Repository	- Chemistry of influent water (spatial and temporal distribution) [BOUNDARY CONDITION FOR WF IPSC]	2.2.08.12.0A 2.1.08.01.0A
2.1.09.02	Chemical Characteristics of Water in Waste Packages	- Water composition (radionuclides, dissolved species, …) - Initial void chemistry (air / gas) - Water chemistry (pH, ionic strength, pCO2, ..) - Reduction-oxidation potential - Reaction kinetics - Influent chemistry (from tunnels and/or backfill) [see also 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels] - Evolution of water chemistry / interaction with waste packages	2.1.09.01.0B 2.1.02.09.0A 2.2.08.12.0B 2.1.09.06.0A 2.1.09.07.0A

Phenomena Number	Phenomena	Associated Processes	Related FEP Number
2.1.09.03	Chemical Characteristics of Water in Backfill	- Water composition (radionuclides, dissolved species, …) - Water chemistry (pH, ionic strength, pCO2, ..) - Reduction-oxidation potential - Reaction kinetics - Influent chemistry (from tunnels and/or waste package) [see also 2.1.09.02 Chemistry in Waste Packages, 2.1.09.04 Chemistry in Tunnels] - Evolution of water chemistry / interaction with backfill	2.1.04.02.0A 2.1.09.01.0A 2.1.09.06.0B 2.1.09.07.0B
2.1.09.04	Chemical Characteristics of Water in Tunnels	- Water composition (radionuclides, dissolved species, …) - Water chemistry (pH, ionic strength, pCO2, ..) - Reduction-oxidation potential - Reaction kinetics - Influent chemistry (from near-field host rock) [see also 2.1.09.01 Chemistry of Water Flowing in, 2.1.09.02 Chemistry in Waste Packages, 2.1.09.03 Chemistry in Backfill] - Evolution of water chemistry / interaction with seals, liner/rock reinforcement materials, waste package support materials	2.1.09.01.0A 2.1.09.06.0B 2.1.09.07.0B
2.1.09.05	Chemical Interaction of Water with Corrosion Products- In Waste Packages- In Backfill- In Tunnels	- Corrosion product formation and composition (waste form, waste package internals, waste package)- Evolution of water chemistry in waste packages, in backfill, and in tunnels [contributes to 2.1.09.02 Chemistry in Waste Packages, 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	2.1.09.02.0A
2.1.09.06	Chemical Interaction of Water with Backfill - On Waste Packages - In Backfill - In Tunnels	- Backfill composition and evolution (bentonite, crushed rock, ...) - Evolution of water chemistry in backfill, and in tunnels - Enhanced degradation of waste packages (crevice formation) [contributes to 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels, 2.1.03.04 Localized Corrosion of WPs]	2.1.04.02.0A
2.1.09.07	Chemical Interaction of Water with Liner / Rock Reinforcement and Cementitious Materials in EBS - In Backfill - In Tunnels	- Liner composition and evolution (concrete, metal, ...) - Rock reinforcement material composition and evolution (grout, rock bolts, mesh, ...) - Other cementitious materials composition and evolution - Evolution of water chemistry in backfill, and in tunnels [contributes to 2.1.09 .03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	2.1.06.01.0A

Phenomena Number	Phenomena	Associated Processes	Related FEP Number
2.1.09.08	Chemical Interaction of Water with Other EBS Components - In Waste Packages - In Tunnels	- Seals composition and evolution - Waste Package Support composition and evolution (concrete, metal, ...) - Other EBS components (other metals (copper), ...) - Evolution of water chemistry in backfill, and in tunnels [contributes to 2.1.09 .03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	2.1.06.05.0D 2.1.03.09.0A
2.1.09.09	Chemical Effects at EBS Component Interfaces	- Component-to-component contact (chemical reactions) - Consolidation of EBS components	2.1.06.07.0A 2.1.08.15.0A
2.1.09.10	Chemical Effects of Waste-Rock Contact	- Waste-to-host rock contact (chemical reactions) - Component-to-host rock contact (chemical reactions)	2.1.09.11.0A
2.1.09.11	Electrochemical Effects in EBS	- Enhanced metal corrosion	2.1.09.09.0A 2.1.09.27.0A
2.1.09.12	Chemical Effects of Drift Collapse	- Evolution of water chemistry in backfill and in tunnels (from altered seepage, from altered thermal-hydrology) [contributes to 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	1.2.03.02.0E
2.1.09.13	Radionuclide Speciation and Solubility in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Dissolved concentration limits - Limited dissolution due to inclusion in secondary phase - Enhanced dissolution due to alpha recoil [controlled by 2.1.09.02 Chemistry in Waste Packages, 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	2.1.09.04.0A 2.1.09.10.0A 2.1.02.04.0A
2.1.09.00	1.09. CHEMICAL PROCESSES - TRANSPORT		
2.1.09.14	Advection of Dissolved Radionuclides in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Flow pathways and velocity - Advective properties (porosity, tortuosity) - Saturation [see also Gas Phase Transport in 2.1.12.02]	2.1.09.08.0B 2.1.04.09.0A 2.1.09.27.0A
2.1.09.15	Diffusion of Dissolved Radionuclides in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Gradients (concentration, chemical potential) - Diffusive properties (diffusion coefficients) - Flow pathways and velocity - Saturation	2.1.09.08.0A 2.1.04.09.0A 2.1.09.27.0A
2.1.09.16	Sorption of Dissolved Radionuclides in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Surface complexation properties - Flow pathways and velocity - Saturation	2.1.09.05.0A 2.1.04.09.0A 2.1.09.27.0A

Phenomena Number	Phenomena	Associated Processes	Related FEP Number
2.1.09.17	Complexation in EBS	- Formation of organic complexants (humates, fulvates, organic waste) - Enhanced transport of radionuclides associated with organic complexants	2.1.09.13.0A
2.1.09.18	Formation of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Formation of intrinsic colloids - Formation of pseudo colloids (host rock fragments, waste form fragments, corrosion products, microbes) - Formation of co-precipitated colloids - Sorption/attachment of radionuclides to colloids (clay, silica, waste form, FeOx, microbes)	2.1.09.15.0A 2.1.09.16.0A 2.1.09.17.0A 2.1.09.18.0A 2.1.09.25.0A
2.1.09.19	Stability of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Chemical stability of attachment (dependent on water chemistry) - Mechanical stability of colloid (dependent on colloid size, gravitational settling)	2.1.09.23.0A 2.1.09.26.0A 2.1.09.21.0A
2.1.09.20	Advection of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Flow pathways and velocity - Advective properties (porosity, tortuosity) - Saturation - Colloid concentration	2.1.09.19.0B 2.1.04.09.0A
2.1.09.21	Diffusion of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Gradients (concentration, chemical potential) - Diffusive properties (diffusion coefficients) - Flow pathways and velocity - Saturation - Colloid concentration	2.1.09.24.0A 2.1.04.09.0A
2.1.09.22	Sorption of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Surface complexation properties - Flow pathways and velocity - Saturation - Colloid concentration	2.1.09.19.0A 2.1.04.09.0A
2.1.09.23	Sorption of Colloids at Air-Water Interface in EBS		2.1.09.22.0A
2.1.09.24	Filtration of Colloids in EBS	- Physical filtration (dependent on flow pathways, colloid size)- Electrostatic filtration	2.1.09.20.0A 2.1.09.21.0A
2.1.09.25	Radionuclide Transport Through Seals	- Advection - Diffusion - Sorption	2.1.05.02.0A
2.1.10.00	1.10. BIOLOGICAL PROCESSES		
2.1.10.01	Microbial Activity in EBS - Natural - Anthropogenic	- Effects on corrosion - Formation of complexants - Formation of microbial colloids - Formation of biofilms - Biodegradation - Biomass production - Bioaccumulation [see also Microbiallly Influenced Corrosion in 2.1.03.06, Complexation in EBS in 2.1.09.17, Radiological Mutation of Microbes in 2.1.13.03]	2.1.10.01.0A
2.1.11.00	1.11. THERMAL PROCESSES		

Phenomena Number	Phenomena	Associated Processes	Related FEP Number
2.1.11.01	Heat Generation in EBS	- Heat transfer (spatial and temporal distribution of temperature and relative humidity) [see also Waste Inventory in 2.1.01.01]	2.1.11.01.0A 2.1.11.02.0A
2.1.11.02	Exothermic Reactions in EBS		2.1.11.03.0A
2.1.11.03	Effects of Backfill on EBS Thermal Environment	- Thermal blanket - Condensation	2.1.04.04.0A
2.1.11.04	Effects of Drift Collapse on EBS Thermal Environment	- Thermal blanket - Condensation	1.2.03.02.0D
2.1.11.05	Effects of Influx (Seepage) on Thermal Environment	- Temperature and relative humidity (spatial and temporal distribution) [BOUNDARY CONDITION FOR WF IPSC]	2.1.08.01.0B 2.1.08.01.0A
2.1.11.06	Thermal-Mechanical Effects on Waste Form and In-Package EBS Components	- Alteration - Cracking - Thermal expansion / stress	2.1.11.05.0A
2.1.11.07	Thermal-Mechanical Effects on Waste Packages	- Thermal sensitization / phase changes - Cracking - Thermal expansion / stress / creep	2.1.07.05.0A 2.1.11.06.0A 2.1.11.07.0A
2.1.11.08	Thermal-Mechanical Effects on Backfill	- Alteration - Cracking - Thermal expansion / stress	2.1.11.07.0A 2.1.04.04.0A
2.1.11.09	Thermal-Mechanical Effects on Other EBS Components - Seals - Liner / Rock Reinforcement Materials - Waste Package Support Structure	- Alteration - Cracking - Thermal expansion / stress	2.1.11.07.0A
2.1.11.10	Thermal Effects on Flow in EBS	- Altered saturation / relative humidity - Condensation	2.1.11.09.0A
2.1.11.11	Thermally-Driven Flow (Convection) in EBS	- Convection	2.1.11.09.0B 2.1.11.09.0C
2.1.11.12	Thermally-Driven Buoyant Flow / Heat Pipes		2.2.10.10.0A
2.1.11.13	Thermal Effects on Chemistry and Microbial Activity in EBS		2.1.11.08.0A
2.1.11.14	Thermal Effects on Transport in EBS	- Thermal diffusion (Soret effect) - Thermal osmosis	2.1.11.10.0A
2.1.12.00	1.12. GAS SOURCES AND EFFECTS		
2.1.12.01	Gas Generation in EBS	- Repository Pressurization - Mechanical Damage to EBS Components - He generation from waste from alpha decay - H2 generation from waste package corrosion - CO2, CH4, and H2S generation from microbial degradation	2.1.12.01.0A 2.1.12.02.0A 2.1.12.03.0A 2.1.12.04.0A
2.1.12.02	Effects of Gas on Flow Through the EBS	- Two-phase flow - Gas bubbles [see also Two-Phase Buoyant Flow in 2.1.11.12]	2.1.12.06.0A 2.1.12.07.0A

Phenomena Number	Phenomena	Associated Processes	Related FEP Number
2.1.12.03	Gas Transport in EBS	- Gas phase transport - Gas phase release from EBS	2.1.12.07.0A 2.1.12.06.0A 2.2.10.10.0A
2.1.12.04	Gas Explosions in EBS		2.1.12.08.0A
2.1.13.00	1.13. RADIATION EFFECTS		
2.1.13.01	Radiolysis - In Waste Package - In Backfill - In Tunnel	- Gas generation - Altered water chemistry	2.1.13.01.0A
2.1.13.02	Radiation Damage to EBS Components - Waste Form - Waste Package - Backfill - Other EBS Components	- Enhanced waste form degradation - Enhanced waste package degradation - Enhanced backfill degradation - Enhanced degradation of other EBS components (liner/rock reinforcement materials, seals, waste support structure)	2.1.13.02.0A
2.1.13.03	Radiological Mutation of Microbes		2.1.13.03.0A
2.1.14.00	1.14. NUCLEAR CRITICALITY		
2.1.14.01	Criticality In-Package	- Formation of critical configuration	2.1.14.15.0A 2.1.14.16.0A 2.1.14.21.0A 2.1.14.22.0A
2.1.14.02	Criticality in EBS or Near-Field	- Formation of critical configuration	2.1.14.17.0A 2.1.14.23.0A
2.2.00.00	2. GEOLOGICAL ENVIRONMENT		
2.3.00.00	3. SURFACE ENVIRONMENT		
2.4.00.00	4. HUMAN BEHAVIOR		
3.0.00.00	RADIONUCLIDE / CONTAMINANT FACTORS (BIOSPHERE)		
3.1.00.00	1. CONTAMINANT CHARACTERISTICS		
3.2.00.00	2. RELEASE / MIGRATION FACTORS		
3.3.00.00	3. EXPOSURE FACTORS		

The numbering scheme for the WF IPSC phenomena derives from the common numbering scheme used in the international FEP database, which group phenomena according to spatial domain (similar to the four regions in Figure 3) or dominant physical process (i.e., T,H,C, or M). The preliminary phenomena listed in Table 3 also include "heading" entries used to categorize the FEPs in the international database. The grouping of the phenomena in accordance with the international database heading entries helps to demonstrate high-level completeness of the list of phenomena.

While a reference scenario was identified in Section 3.1, the high-level phenomena and associated processes in Table 3 are generally applicable to all scenarios and across all time and spatial domains. For example, "Drift Collapse" can represent the rubble infill from crystalline rock or the creep closure of salt. As the phenomena become more detailed in future PIRT iterations, they will likely become more scenario-specific.

3.2.2 Sub-continuum Phenomena

As described in Section 1, sub-continuum analyses will be used to characterize some of the material properties and mechanistic processes simulated with the continuum-scale high fidelity THCM models. Therefore, identification of potentially important sub-continuum processes is necessary. These sub-continuum phenomena derive from the phenomena and associated processes listed in Table 3, but generally have an even finer level-of-detail. At the lowest sub-continuum scale, the behavior of atoms, ions, groups of atoms, and/or molecules at short time and length scales define the potential phenomena that are modeled with the high fidelity models at the larger continuum scale over longer time scales. Sub-continuum, atomistic processes contribute to the importance of radionuclide release and transport in all three of the key engineered barrier system components identified in Figure 3. Some potentially important sub-continuum phenomena include:

- Waste Form Material
  - Chemical and physical evolution of the radionclide-bearing solid state waste form (chemical and physical alteration, degradation)
  - Transport of radionuclides through the waste form to the surface
- Waste Form and Waste Package Surface
  - Corrosion and formation of surface layers in the interface between the waste form or waste package surface and the surrounding physical environment, which may be a solid (e.g., corrosion product rind, cladding), liquid, or gas)
  - Transport of radionuclides through the surface layer
  - Radiation and microbial effects associated with the corrosion
  - Temperature-dependent kinetics and solid-liquid interface chemistry, dissolution of the solid phase and transport of the radionuclide into the aqueous environment
  - Sorbed species and surface site activities on the waste form and waste package
  - Species behavior within micropores and on charged surfaces, electrochemical corrosion
- Waste Form and Waste Package Aqueous Environment
  - Aqueous properties of the liquid, including the thermodynamic activity of all dissolved species in the aqueous phase
  - Chemical effects of degradation of the waste form, waste package, and other EBS components and their degradation products on the chemistry of the aqueous solution
  - Nucleation of secondary phases, rate of co-precipitation and colloid formation from the chemical components released from the WF and EBS
  - Mass transport to and from the waste from or waste package surface: advection, molecular diffusion, electrochemical diffusion, and diffusion of radionuclides through the aqueous environment as influenced by chemical potential gradients
- EBS Buffer Material
  - Transport (advection, diffusion, sorption) of radionuclides and radionuclide-bearing species through the EBS buffer materials (e.g., backfill, liner, seals)

There have been a large number of experimental studies on selected aspects of these topics, but much remains elusive and uncertain. Reaction mechanisms and the thermodynamics of the incorporation and adsorption of radionuclides on minerals and secondary phases have only been investigated using experimental and field approaches. Fortunately, the computational capability to simulate waste form properties has expanded greatly, and there are now many new opportunities to use computational approaches to mechanistically interpret experimental and field observations and predict thermo-chemical

behavior. Furthermore, computational approaches, especially at the quantum mechanical and atomistic level, are a complementary tool that can provide detailed insight into reaction pathways, thermodynamics, and kinetics during the degradation of EBS components and subsequent mobilization and transport of radionuclides within the EBS.

3.2.2.1 Phenomena in the Waste Form Materials

Within the waste form, there are a number of fundamental sub-continuum phenomena that relate to the evolution of the waste form. One of the fundamental phenomena endemic to the WF is radioactive decay, whereby a radionuclide decays into a daughter product and emits another particle and/or radiates energy (e.g., $^{235}U \rightarrow {}^{231}Th + \alpha$). This changes the chemical identity of the radionuclide (parent to daughter) and also transfers energy into other atoms of the lattice, with elevated temperatures and structural disruptions through recoil cascades causing lattice damage. A second fundamental process, more general in nature, is diffusive movement of constituents (radionuclides, other WF structural elements, vacancies) through the WF and underpins a number of processes at the continuum scale. Damage creates vacancies (and interstitials), whose migration facilitates transport of other constituents via site exchange, aggregation within the structure, or annihilation at the solid surface. Radionuclides may migrate to grain boundaries or surfaces where they may be available for dissolution into an aqueous phase. Chemical alteration or damage, enabled by constituent transport or incorporation of external constituents from the external environment, facilitated by elevated temperatures from radionuclide decay may nucleate secondary phases within the waste-form itself. Gases (e.g. from radiation products) diffuse and nucleate bubbles, which grow and add additional stresses to the structural integrity of the WF, leading to swelling and cracking. The basic nature of the phenomena is mostly independent of the WF (glass, ceramic, or metallic); the differences in composition can be thought of as "boundary effects" for the solid state phenomena and entail little fundamental distinction in the description of the processes. Many of the similar processes also apply to cladding, waste package canister and containment structural barrier components. Phenomena in the waste form of specific interest at the sub-continuum level include:

- structure and phase stability, volumetric and conductivity changes - as a function of composition (including transmutation effects), temperature, and pressure
- nucleation and growth of bubbles, secondary phases, and cracks within the waste form
- cascade damage, amorphization and recrystallization
- point defect formation energies (vacancies, interstitials, anti-site, Frenkel, Schottky)
- point defect migration energies
- volume diffusion of chemical constituents (especially radionuclides) within the WF to exposed surfaces
- grain boundary formation energies, dislocation motion energies
- grain structure and microstructural evolution
- diffusion rates of constituent elements (esp. radionuclides) along grain boundaries
- out-diffusion (or in-diffusion) energies and rates of chemical species at bulk interfaces
- surface free energies for exposed surfaces and cracks
- surface charge along exposed surfaces and cracks and the migration rate of charged ions through potentially overlapping electrical double layers within microcracks formed within the waste form
- migration of water into microcracks in the waste form, followed by chemical reaction and associated mechanical effects (shrinkage or swelling)

Example of Borosilicate Glass Corrosion

One particular example is provided by the corrosion of borosilicate glass, which has been the focus on numerous studies in recent years. The key objective in these studies has been to determine the long-term rate of glass corrosion and the mechanism by which this takes place. The mechanism is important for demonstrating that predictions over geological time scales (up to 10,000 years) are reliable. The phenomenology of glass dissolution as presently understood (Van Iseghem et al., 2007) is described in Figure 4. Upon reaction with water, the glass is first hydrated, and ion exchange between the hydrated glass and the adjacent solution takes place. While the hydration front moves further into the glass, the outer part of the hydrated glass transforms itself to a porous gel-like material. The transition between hydration and gel formation may be related to the solubility of water in the glass, with hydrated glass defined as the case in which water is soluble in glass, while the gel represents a case in which a phase separation between water and the residual glass network occurs. Finally, the reaction rate slows down once the hydrated glass becomes stabilized by high Si concentrations in the adjacent solution (i.e., in the pores of the gel). In certain cases, transport in the gel may be so slow that it becomes rate limiting, but most reductions in rate appear to be due to the affinity term and not to slow diffusive transport through the gel.

To simulate these effects requires a microscopic (μm) to mesoscopic (cm) model for glass dissolution in 1D and 3D, including the formation of a leached gel layer with possible diffusion control of reactive constituents to and from the dissolving fresh glass. The aim is to reproduce the observed glass dissolution rates, especially the rate reduction by up to a factor of 10,000 occurring over long time periods. Effects of glass composition on the rates and on the chemistry and physics of the evolving gel layer will also be considered. The model will consist of a mechanistic surface reaction-controlled glass dissolution rate law that incorporates "affinity" effects (the slowing of rates due to high silica concentrations) coupled to μm to cm scale multi-component diffusive transport through the gel layer and the hydrated glass. The goal of the 3D modeling is to capture the formation and evolution of fractures that penetrate the outer glass surface.

The model that will be developed will be capable of representing the coupling of two diffusion steps, both water and silica diffusion in the gel and the hydrated glass, on the nanometer to micron scale. Porosity evolution as a function of reaction progress and the phase transition of hydrated glass (a single phase) to gel plus water (two phases) will be included in the model as well. Modeling of water transport and glass corrosion in the glass fracture network will make it possible to obtain a relation between fracture aperture and extent of corrosion.

The microscopic continuum scale modeling requires input from molecular scale models that capture the behavior of solutes along charged surfaces within nanopores within the evolving gel layer, or within the waste form itself. Fundamentally, this is a problem of describing the nature of waste form surface in contact with water and electrolytes, or with gas, and then to understand how this special microscopic environment affects transport to and from the surface.

Coupling of the glass dissolution process with near field constraints is essentially done by controlling the transport and the rate of release and of consumption of silica dissolved either from the glass or from near field materials. Sorption of Si on corrosion products of the container like magnetite will in this way increase glass dissolution rates because Si concentrations at the interface of the hydrated glass are diminished by this sorption process. Similar processes occur on clay (bentonite).

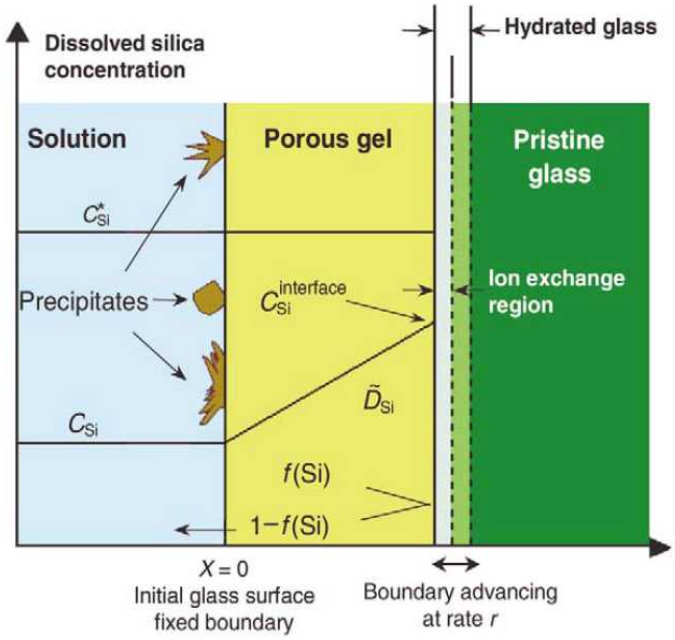

Figure 4. Schematic representation of formation of a gel layer during glass corrosion.

3.2.2.2 Phenomena at the Waste Form and Waste Package Surface

Phenomena occurring at the waste form surface control the release of radionuclides from the waste form into the surrounding physical environment. The interface at the waste form surface may be another solid (e.g., a designed material such as a clad, or a degradation generated corrosion product such as a rind or barrier layer) in which case it is a particular condition of the solid state problem listed above, or the waste form surface may be contact with an aqueous or gaseous environment. Phenomena occurring at the waste package surface are important to the degradation of the waste package and the subsequent development of breaches in the waste package material that permit fluid movement into and radionuclide transport out of the waste package. The waste package surface may be in contact with a solid (e.g., a corrosion product rind), aqueous, or gaseous environment.

The temperature-dependent kinetics of the reactions at the interfaces of the waste form and waste package surface with aqueous solutions of specified composition needs to be determined. Chemical reactions at the interfaces (by H_2O, O_2, or chemical products of dissolution of waste form or waste package corrosion), availability, identity and density of surface sites, and transport of chemical species to and from the interfaces lead to formation of surface complexes and determine in large part the rate of transport of radionuclides from the surface and into the surrounding environment. Driven by diffusion of water or species from within the waste form or waste package, additional solid or gel phases can nucleate and precipitate on the surface, thus determining the rate of barrier layer growth or surface dissolution. Out-diffusion of chemical species, e.g. leaching of alkali ions, contributes to the overall dissolution of the waste form. Ionization of species from the WF surface can allow those species to be released into solution (dissolve) resulting in changes not only to elemental abundances (i.e. composition) at the surface of the waste form, but also to the solution composition adjacent to it. All these process are influenced by protonation or deprotonation rates, depending on reducing or oxidizing geologic environment, and electrochemically-driven chemistry. Sub-continuum modeling can identify crucial reaction paths and estimate relative rates of processes at short time and length scales, quantities of interest for continuum scale simulations. Phenomena at the interface at the sub-continuum scale of concern include:

- surface/interface structure free energies, surface site energies and densities as a function of (bulk material and aqueous) composition and structure, temperature, oxidizing or reducing conditions.
- energies of surface chemical processes, elemental attacks (waste form degradation products, H_2O, O_2, H^+, OH^-, H_2O_2, CO_2, He, H_2, CH_4, H_2S, acids and bases, sulfates, phosphates, Na, Ca, and other geological species): protonation/de-protonation, hydration/hydroxylation of surface sites, ligand or gas molecule attachment, formation of surface and solution complexes, oxidation of metal atoms
- migration/diffusion of chemical species along surfaces
- out-diffusion (or in-diffusion) energies and rates of chemical species at bulk interfaces: water, bulk vacancies, alkali atoms, ion exchange, redox of radionuclides.
- surface layer precipitation or surface dissolution: nucleation, formation and rates
- microbial effects in surface layers
- electrochemical and radiolysis effects on the chemistry
- fugacities (partial pressures) of species in gas phase
- kinetic and equilibrium treatments of each of these

3.2.2.3 Phenomena in the Waste Form and Waste Package Aqueous Environment

The chemical activities at the waste form and waste package surface are directly affected by the nature of the aqueous environment, as is the migration of radionuclides and radionuclide complexes away from the surface. The composition of the aqueous environment is a dynamic function of the geological environment, and the degradation of the waste form, waste package, and other EBS components. The mobilized radionuclides can form chemical complexes, altering their interactions with (sorption to) solid surfaces or colloids. Mass transport to and from the surface is mediated by the nearby aqueous environment, through molecular diffusion, advection, and behavior of species micropores and charged surfaces. Electrochemical effects and formation of electric double layers influence the chemistry near a waste form or waste package surface. A special case of phenomena in the aqueous phase is the formation of secondary phases or colloids from the chemical components released the surface of the degrading waste form or waste package. These colloids can trap and release radionuclides, and thereby inhibit transport through the aqueous environment. Sub-continuum phenomena of potential interest within the aqueous environment include:

- free energies of solution for chemical constituents
- chemical activity coefficients, speciation
- bulk aqueous diffusion rates, advective mixing and transport
- electrochemical free energies and diffusion rates
- pH and temperature effects
- alpha radiolysis effects on local chemistry and dissolution reaction kinetics - equilibrium and kinetic treatments of each of these
- nucleation and growth of secondary phases and colloids
- transport and behavior of colloids

3.2.2.4 Phenomena in the EBS Buffer Materials

An important phenomena in the EBS buffer is radionuclide transport, such as advection and diffusion through backfill. As the distance from the degrading waste form increases, the chemistry becomes more

dominated by the EBS buffer environment (clay backfill, cementitious materials) and the surrounding host rock geologic environment. Transport of radionuclide and radionuclide complexes through the EBS buffer environment takes on many of the same concerns described for the aqueous environment in Section 3.2.2.3, with a focus on the following:

- complexation and sorption on mineral surfaces, especially iron hydroxide and clays
- diffusion through "dense" nanoporous EBS materials like bentonite backfill where overlapping double layers can restrict movement of anions (anion exclusion) and modify the diffusion rate of cations

3.2.2.5 Particular Challenges at the Sub-continuum Scale

The sub-continuum analyses are likely to encompass certain phenomena that will pose particular challenges for predictive simulations, and for which data and computational techniques are likely to be challenging to acquire. These challenges include:

- *Actinides*. Experimental data for the chemical and solid state properties of actinide-bearing materials will be scarce – radioactive materials are difficult and expensive to handle. Quantum-mechanical treatments of actinides are problematic (inaccurate) due to importance of relativistic effects that are poorly treated in existing first-principles computational methods. This inadequacy extends to classical molecular dynamics methods, where developing accurate inter-atomic potentials is problematic due to lack of sufficient data concerning actinide chemistry. New actinide-capable quantum chemistry methods, and new inter-atomic potentials that abstract crucial aspects of actinide chemistry, will need to be developed. The lack of extensive experimental data will make validation of the new methods challenging.
- *Electrochemistry*. Much of the chemistry at the solid-aqueous interfaces will be driven by electrochemical effects, particularly for corrosion of metallic waste forms (and structural components of the engineered barrier), but also for all waste form surfaces exposed to aqueous environments. Proper electrochemical boundary conditions (and therefore computation of chemical activity and free energies of reactions) is problematic for first principles quantum methods, and the necessary chemistry is beyond the accuracy of classical force field methods. New methods for computing chemical properties in electrochemical environments will be needed.
- *Nucleation-driven processes*. Much of the physics and chemistry of the WF degradation and radionuclide transport is dependent on stochastic events that are difficult to predict. Formation of gas bubbles in the waste form, or the formation of secondary phases in the WF, at the WF surface, or in the aqueous environment, and particularly the formation of colloids play prominent roles in the degradation of waste forms and the transport of radionuclides. The state of modeling for the prediction of nucleation behavior is primitive.
- *Time scales*. The time scales accessible to sub-continuum simulations is very limited, the order of nanoseconds for classical molecular dynamics, picoseconds for first principles density functional theory methods. Accelerated dynamics techniques extend these time scales somewhat, but candidate waste forms have degradation and dissolution rates potentially on geologic time scales, years or hundreds of years. Methods for computing long-time scale and rare processes are challenging, and develop the problem of how to validate the results.
- *Length scales.* The processes important about a waste form are highly heterogeneous, with extensive disorder, encompassing amorphous systems, complex interfaces/surfaces, and aqueous environments. The sheer size of the necessary calculations require gross simplifications to enable first-principles quantum chemistry calculations, and challenge even the much more computationally efficient classical molecular dynamics simulations. The refinement and verification of the simplifications in

the computational models will entail careful exploration. This is related to the upscaling problem: what simplifications can one make and still retain the essential quantitative aspects of the process?

3.3 Importance Ranking

3.3.1 High-Fidelity and Surrogate Model Phenomena

A preliminary importance ranking was performed on the phenomena identified in Table 3. . Separate rankings were performed for the high-fidelity models (See PIRT Table A-1) and the surrogate models (see PIRT Table A-2). The preliminary importance rankings were based on the reference scenario identified in Section 3.1, but were generally applicable to most scenarios. Specific exceptions are identified in the PIRTs.

The phenomena were ranked as having low, medium, or high significance (Table 4). The significance was based on a subjective consideration of evaluation criteria, generally based on some combination of performance metrics, such as:

- Contribution to dose
- Cumulative release and release rate from EBS (total and by radionuclide)
- Interim measures such as waste form degradation rate, radionuclide mobilization rate, waste package degradation rate, transport rate through EBS, etc.

Table 4. Importance Ranking Scheme

Rank	Value	Definition
High (H)	3	Phenomenon has a controlling impact on one or more of the evaluation criteria
Medium (M)	2	Phenomenon has a moderate impact on one or more of the evaluation criteria
Low (L)	1	Phenomenon has a minimal impact on one or more of the evaluation criteria
Uncertain (U)		

Source: NUREG 1918 [29]

When more resolution in the importance rankings is required, each of these categories can be split into three subdivisions giving a nine-level scale, or the high and low categories can be split into two subdivisions, giving a five-level scale. For some phenomena, there will be a lack of available knowledge, resulting in a need for additional information.

The preliminary importance rankings in Tables A-1 and A-2 will be continually evaluated as new information becomes available during the course of this multi-year effort. The importance rankings will be used to help focus research and model development on the key phenomena. In particular, there are expected to be a large number of sub-continuum phenomena requiring prioritization, as discussed in Section 3.3.2.

3.3.2 Sub-Continuum Phenomena

The range and scope of phenomena at the sub-continuum scale potentially relevant to modeling the performance of long-term waste disposal options is vast, as can be deduced from the very cursory listing of the classes of sub-continuum phenomena presented in Section 3.2.2. It will not be possible, nor is it

desirable to compute quantities for every possible physical phenomenon that might contribute to radionuclide release from a specific waste form. The preliminary list of sub-continuum phenomena does not incorporate any importance ranking or requirements. The requirements for any sub-continuum simulations will dictated by the actual needs determined through a propagation of requirements for the constitutive models from the high-fidelity simulations. Some phenomena may prove to be unnecessary or to be of peripheral importance, or the quality of a constitutive model for a particular process may be perfectly adequate with a very phenomenological (empirical, irreducible) model for a particular physical process, or the necessary data may be available or readily obtained from experimental observations. In each of these cases sub-continuum simulations may be unnecessary. Additionally, the computational techniques needed to compute a quantity of interest may be inadequate or entirely lacking, in which case quantitatively predictive simulations may not be possible, unless significant effort is invested in developing new theoretical methods. In any event, the prioritization of sub-continuum phenomena for further analysis will be based on the following considerations:

- Is a quantitative description of the phenomena is required by a constitutive model? If not, then no further sub-continuum analysis is necessary because the model does not need this quantity.
- Is an existing phenomenological model (constitutive model) adequate (i.e., use of the model for the desired range of high-fidelity simulations is proven adequate in validation tests)? If so, then no further sub-continuum analysis is necessary because the details of the sub-continuum processes are not needed.
- Is experimental data of sufficient quality (within uncertainties) available, or readily obtainable? If so, then no further sub-continuum analysis is necessary because the experimental data makes simulations redundant.

Only if the phenomenology is inadequate and experimental data are lacking, are sub-continuum simulations called for, and if existing simulation methods can be used to obtain the quantities of interest, then simulations are pursued. If the simulations methods are inadequate, either because of computational limitations or because of the inherent inaccuracy of the physical approximations used in the simulation methods, development of new methods is appropriate (or a rethinking of the nature of the phenomenology used in the high-fidelity simulations).

The strategy for the prioritization will also need to account for the fact that the quantities of interest are not just the parameters that describe fundamental physical processes, but also the identification and enumeration of the physical and chemical processes important for the high-fidelity modeling, i.e., the form of the physical abstraction embodied by the constitutive model. The need for validation studies will also contribute to the prioritization strategy.

3.4 State of Knowledge Ranking

Each phenomena will be represented in a model, in which it may be characterized by one or more parameters. Therefore the state of knowledge and the adequacy of the models and the data are relevant to the ability of the WF IPSC to ultimately simulate long-term behavior of a disposal system.

The state of the knowledge ranking (Table 5) is based on:

- Existing modelling tools,
- Available supporting data, and
- Likelihood of obtaining additional information.

Table 5. State of Knowledge Ranking Scheme

Rank	Value	Definition
Known (K)	3	Sufficient understanding to make assessment of practical ramifications; small uncertainty exists.
Partially Known (PK)	2	Partial knowledge and understanding; moderate to large uncertainty remains.
Unknown (UK)	1	Totally unknown or very limited knowledge; uncertainty cannot be characterized.

Source: NUREG 1918 [29]

The state of knowledge rankings may be different for various scenarios and model fidelities. The ranking schemes for adequacy of models (Table 6) and data (Table 7) provide additional considerations for determining the state of knowledge.

Table 6. Model Adequacy Ranking Scheme

Rank	Value	Definition
High (H)	3	At least one mature physics-based or correlation-based model is available that is believed to adequately represent the phenomenon over the full parameter space of the applications
Medium (M)	2	Significant discovery activities have been completed. At least one candidate model form or correlation form has emerged that is believed to nominally capture the phenomenon over some portion of the application parameter space.
Low (L)	1	No significant discovery activities have occurred and model form is still unknown or speculative.
Uncertain (U)		The existing state of modeling tools with respect to this phenomenon is not known.

Source: NUREG 6798 [28]

Table 7. Data Adequacy Ranking Scheme

Rank	Value	Definition
High (H)	3	A high resolution database (e.g., validation grade data set) exists, or a highly reliable assessment can be made based on existing knowledge. Data needed are readily available.
Medium (M)	2	Existing database is of moderate resolution, or not recently updated. Data are available but are not ideal due to age or questions of fidelity. Moderately reliable assessments of models can be made based on existing knowledge.
Low (L)	1	No existing database or low-resolution database in existence. Assessments cannot be made with even moderate reliability based on existing knowledge.

Source: NUREG 6798 [28]

Where the state of knowledge is limited, it is important to determine the likelihood of obtaining new information (Table 8) to improve the state of knowledge.

Table 8. Likelihood of Obtaining Additional Information Ranking Scheme

Rank	Value	Definition
High (H)	3	Data needed are readily obtainable based on existing experimental capabilities.
Medium (M)	2	Data would be obtainable but would require moderate, readily attainable extensions to existing capabilities.
Low (L)	1	Data are not readily obtainable and/or would require significant development of new capabilities.

Source: NUREG 6798 [28]

4. Verification, Validation, and Uncertainty Quantification Strategy

Various programs of national significance are pushing the development of numerical simulation to new levels. Among them is the DOE program to assess the long-term safety of the nation's first underground high-level radioactive waste repository. An enabling technology common to all of these programs is the ability to compute the reliability of complex, large-scale systems with high confidence.

The goal of the NEAMS IPSC WF program is to develop an integrated suite of simulation capabilities for modeling the long-term performance of fabricated waste forms placed in an engineered environment for waste storage or disposal. This system will be used to generate computational evidence regarding the performance of waste forms in possible nuclear waste disposal scenarios, in support of risk-informed decision-making about options for the safe and permanent disposal of nuclear waste. Quantified confidence must be established in the full chain of modeling techniques and computer codes used to generate that numerical evidence. The NEAMS IPSC WF system will enable simulations with quantified uncertainties. Verification and validation, in addition to uncertainty quantification, (VV-UQ) will be used to demonstrate quantified confidence in the results of the NEAMS IPSC WF modeling suite. Uncertainty quantification (UQ) is an increasingly important aspect of many areas of computational science, where the challenge is to make accurate predictions with a quantified level of confidence about the performance of complex physical systems in the absence of complete or reliable data. For purposes of the following discussion, we envision UQ in the context of certification, as a tool for deciding whether a system is likely to perform safely and reliably within design specifications.

The term “NEAMS IPSC WF modeling suite,” as used herein, will be used to encompass all of the sub-models in the system, in the full hierarchy of fidelity from properties characterization at the sub-continuum, through high-fidelity modeling of various coupled phenomena in the THCM framework, culminating in a set of efficient surrogate models with confirmed accuracy in well-delineated performance assessment regimes. The term “Integrated Assessment Code” will be used to identify the software implementing the surrogate models, to be used for generating statistical information needed for performance assessment (PA) and decision analysis (DA). The ultimate output of the system will be the statistical analyses of the Integrated Assessment Code, with quantified uncertainties. However, the fidelity of these analyses is predicated on the fidelity of every element of the hierarchy that contributed to this analysis. The VV-UQ process within the NEAMS IPSC WF modeling suite will apply to the full hierarchy of computer codes and all data flow between the different hierarchies of simulations.

The complexity of the long-term performance assessments of waste forms in geological repository environments is reflected in the hierarchy of simulation capabilities envisioned in the NEAMS IPSC WF modeling suite. Each element in the hierarchy of simulations will have requirements and circumstances specific to that component, and it is expected that the means to establish quantitative confidence will depend on those specific circumstances—the state of knowledge, maturity of the computational methods, accuracy of the available approximations—but all simulations feeding ultimately to the PA code will hew to a VV-UQ process, which will be integrated into an overall system VV-UQ across hierarchies, with quantified uncertainties.

4.1 Overview

Requirements

Use of every component in the NEAMS IPSC WF modeling suite begins with satisfying a requirement of the application: with identification of the phenomena of interest, prioritized within the PIRT, and the “customer” for that quantity of interest. If simulations are required to obtain the desired quantity—as decided via a requirements triage in developing the PIRT—a simulation process is set in motion. The

NEAMS IPSC WF PIRT documentation will be used to determine the needs for simulations. The requirements will include not only the need for an identification of the quantities of interest in a phenomenon, but also a minimum accuracy and maximum level of uncertainty in the quantities of interest needed for the simulation to satisfy those requirements.

Verification

A prerequisite for any discussion of validation is qualification of the computer codes to be used in simulations. Verification is the demonstration that a computer code correctly and reproducibly does what it is designed to do, in the regime for which it is to be applied, sufficiently "converged" to give a correct representation of the conceptual description for which it was intended [3]. Verification makes no assertion concerning the accuracy of the physical approximations implemented in a code, only that the software correctly implements in computer code the mathematics that express a specific conceptual description. Qualification is related to the use of computer codes for solving real physical problems: a code is "qualified" for a particular application if it has been verified and the combination of solution techniques, constitutive equations, geometric discretization, and initial/boundary conditions, all consistent with the limitations of the code, lead to an acceptable solution to the physical problem [3]. The answer to what is an acceptable solution is a relative one that may depend on the complexity of the problem being solved, and often constitutes the prominent source of uncertainty in a simulation that needa to be captured in an uncertainty analysis.

Procedures to verify a computer code will depend on the particular code, and include unit testing, methods of manufactured solutions, internal consistency checks, and comparison to analytic solutions, or some combination of these. The verification evidence must satisfactorily prove the correct operation of the computer code across the entire span of its intended use in the simulations. To assess the predictive capability of a code, "benchmarking" is often used. Benchmarking, in this context, is defined as the comparison of predictions obtained with one code to those obtained with other codes having presumably similar capabilities. Benchmarking does not itself constitute verification and at best is weak evidence that a given computer code performs as well (or as badly) as a different code. It is only through a formalized discrepancy-resolution process among the various codes participating in such a benchmarking that significant confidence can be established in any particular code [8,9]. While benchmarking has often been used for either verification or qualification, herein, benchmarking will be associated with qualification rather than verification because prior to performing any benchmark calculations discussed herein, it is intended that every code will be subjected to extensive verification.

An immediate corollary of verification is reproducibility and traceability. The software used in every simulation will be of established provenance, with a known recorded version that can be reconstituted upon demand for use in additional verification tests and new simulations. The software specifically developed within the NEAMS IPSC WF modeling suite will be subject to documented software quality engineering (SQE) processes with full version control. It will be able to record (and recover on demand) the exact state of the software used for each simulation, along with the minimum input and control data necessary to reproduce each test. It is anticipated that the NEAMS IPSC WF modeling suite will also make extensive use of preexisting software from outside parties. Second party software used for simulations within the NEAMS IPSC WF process will be subject to the same requirements of verification and reproducibility/traceability.

Validation

With the prerequisite of a qualified code satisfied, the fidelity with which the physical model implemented in the computer code represents reality will be assessed. Validation is achieved when acceptable agreement is demonstrated between the predictions of the simulations and the observation of the physical process or system for which it was intended. This means that is that there is a quantified

assurance that the simulation model embodied in the computer code is a sufficiently accurate representation of the physical phenomenon or system to which it is applied [4]. Sufficiency of a model, or validation, is measured against the requirements for the simulation, that the prediction have a specified accuracy. Inadequate validation requires reassessing the simulation model, and is a trigger for an investigation of the physical model at a lower scale of the modeling hierarchy. This might involve either a reevaluation of the parameters that are the realization of the physical model or a construction of an entirely new physical model that invokes additional or different phenomena.

Validation is achieved through direct comparison of simulation predictions with observed reality, for all the quantities of interest in the system application, within the regime of interest. It is not a computer code, but a circumscribed range of simulations of a computer code that is validated. Evidence of validation will be provided or cited for each simulation of a new application area.

The nature of validation will vary dependent upon the application and the level in the hierarchy. Typical model validation compares a model's results with experimental measurements and/or field observations. At the subcontinuum level, only a small amount of discrete experimental data may be available for certain phenomena, and the intrinsic accuracy of the physical models may not be well known. Validation measurements will be impossible to obtain for the Integrated Assessment Code at the temporal and spatial scales of interest for post-closure repository performance. Therefore, the overall NEAMS IPSC WF modeling suite will be validated using corroboration, technical review, and natural analogues.

On occasion, the necessary data will be available from the literature or project reports, but it is anticipated that the validation will involve physical systems for which data will be lacking. For the success of the NEAMS IPSC WF modeling suite, obtaining adequate validation evidence will require close interaction with a robust experimental program. This will require strong coordination between the simulation teams and the experimental program to assure that such data necessary to achieve validation, where it is within practical reach, is actually obtained.

Uncertainty Quantification

Uncertainty quantification, as used herein, will be understood to be the quantitative characterization of uncertainties in our computational modeling of the waste-form degradation and near-field environment. The uncertainties in a model can arise from: the model's structure, or the accuracy with which a mathematical model describes the true system to which it is applied; the numerical approximation, or how appropriate the numerical method that is used for an application may be in approximating the operation of what is being modeled; the initial and/or boundary conditions, namely, how well the data or information needed for specifying them is known; and the data for input to the model itself and/or the model's various parameters.

Various types of uncertainty can be identified and include aleatory uncertainties, which are irreducible variabilities inherent in nature, and epistemic uncertainties, which are reducible uncertainties resulting from a lack of knowledge. For the waste-form/near-field system under study, both types of uncertainties are anticipated to be present.

Part of evaluating the uncertainties through a calculation is to propagate through a simulation the uncertainties stemming from how well the parameters specifying the physical model underlying the simulation are known. A byproduct of a comprehensive uncertainty analysis will be a sensitivity analysis (SA) over those parameters. The SA serves to identify the relative importance of various aspects of the simulation, and serves as a quantitative measure of the requirements on the accuracy and uncertainty on the parameters that are the realization of the physical model. Hence, while the UQ is passed upwards in the modeling to propagate simulation uncertainties higher in the modeling hierarchy, the SA is passed downwards to refine the accuracy and uncertainty requirements on the phenomena simulated at lower levels of the WF modeling hierarchy. A simulation may reveal that certain sub-scale phenomena are less

important, are needed only imprecisely, while revealing that other sub-scale phenomena take on great importance and must be refined to greater accuracy and smaller uncertainties. The role of the UQ/SA will serve two duties, one looking up the hierarchy, the other looking down the hierarchy.

Upscaling Between Levels in the Model Hierarchy

The information passed between levels in the hierarchy will be captured in the constitutive models. Here, a "constitutive model" is meant as a generalized object: the form of the abstraction of the physical model and the parameters that are the specific realization of that model for a simulation at any level of the WF hierarchy. The nature of the "upscaling", or the communication of sub-scale information into the abstracted physical model of the next scale is, in general, an unsolved problem, and is dependent on the specific case. There is rarely an exact one-to-one correspondence of the parameters directly computed at a sub-scale and the parameters used as input at the next scale. The nature of a constitutive model can be an interpolation of the results of a higher fidelity simulation model results, such as the development of a surrogate model for performance assessment from results of a high-fidelity THCM simulations or the development and refinement of analytic interatomic potentials from the results of dynamical quantum mechanics simulations. A constitutive model can incorporate evaluations of specific sub-scale unit processes, such as a diffusion constant computed with a dynamical atomistic simulation used to populate a THCM constitutive model or the population of the site energies and activation energies of a kinetic Monte Carlo model with the results of atomistic simulation. Often, constitutive model parameters will be "calibrated" by the upper scale application for validation purposes to correct for systematic errors introduced by the abstraction of a simplified physical model. The abstraction of the physics and the consequent calibration of the parameters will need to be accounted for in propagating uncertainties up the hierarchy, and on occasion may result in sub-scale uncertainties being not directly propagated as a consequence of a calibration that subsumes the uncertainties in a parameter. In addition to the physics challenge of fashioning a suitable physical abstraction and populating that abstraction, the numerical challenge of how to propagate the uncertainties in that abstraction and a specific realization of that abstraction will require generalization and refinement of existing techniques for UQ.

4.2 Model Hierarchy

The approach to verification and validation outlined here is similar to that which has been used previously [5]. Namely, the NEAMS IPSC WF modeling suite verification activities will be designed to establish confidence that the calculated results are achieved properly using the modeling tools, sub-models, and a given set of controlled inputs. NEAMS IPSC WF modeling verification will include verification of the Integrated Assessment Code, verification of the NEAMS IPSC WF modeling suite, verification of any dynamically-linked library implementations within the NEAMS IPSC WF modeling suite, verification of model inputs entered into the NEAMS IPSC WF input database, and verification of the implementation of the sub-model abstractions within the Integrated Assessment Code. Coupling between sub-models within the Integrated Assessment Code will be verified by ensuring that information generated by one sub-model is fed correctly to successive sub-models.

The NEAMS IPSC WF modeling suite inputs will be checked, controlled, and documented to maintain traceability and transparency. Confidence in the methodology of and inputs to the NEAMS IPSC WF modeling suite will be provided through:

- selection of input parameters and/or input data from validated supporting analysis model reports,
- model calibration activities and/or evaluation of the initial/boundary conditions for the NEAMS IPSC WF modeling suite, establishing model convergence, and
- evaluation of the impacts of uncertainties on model results.

These three activities should demonstrate that: (1) the NEAMS IPSC WF modeling suite's input parameter values from source documents, as well as those parameter values that are calculated by the NEAMS IPSC WF modeling suite, are correctly propagated throughout the Integrated Assessment Code; (2) the Integrated Assessment Code is stable in terms of the number of realizations, the length of model time steps, and spatial discretization; and (3) that the uncertainty in model inputs is propagated through and correctly accounted for in the Integrated Assessment Code.

The following post-development methods will be used to demonstrate the Integrated Assessment Code validation with respect to intended use and desired level of confidence:

- Corroboration of Integrated Assessment Code results with analogue studies or other relevant observations not previously used to develop or calibrate the model
- Confidence building through incorporation of the recommendations and observations provided by technical reviews conducted by external experts regarding a preliminary version of the NEAMS IPSC WF modeling suite
- Corroboration of abstraction or sub-model results to the results of the validated mathematical models from which the abstraction or sub-model was derived
- Corroboration of NEAMS IPSC WF modeling suite results with the results obtained from auxiliary analyses (including benchmarking) as a means of providing additional confidence

4.3 Sub-continuum Materials Properties Characterization

The product of sub-continuum materials properties characterization is the development, maintenance, and updating of the constitutive models that are input to the high-fidelity THCM simulations in NEAMS IPSC WF modeling suite. The goal is science-informed engineering simulations that are more broadly predictive (extrapolative) rather than interpolative within a narrow circumscribed regime. The range of sub-continuum physical phenomena for which quantified descriptions might be required is extremely broad, spanning chemistry, physics and materials science. The sub-continuum research tools that are required to simulate many of these phenomena are deterministic, "first principles" methods with few free parameters, methods whose accuracy is fundamentally limited by the fidelity of the physical model rather than the parameterization of that model. Research tools at the sub-continuum scale are often in a continuous state of development, implementing improved understanding of physical approximation that underlie the methods. The development and application of methods to simulate sub-continuum phenomena is a highly dynamic and institutionally distributed enterprise. This poses special challenges for NEAMS IPSC WF, where the process requires detailed traceability with documented reproducibility and propagating verified, validated quantified uncertainties.

It is anticipated that WF IPSC will coordinate with the Fundamental Methods and Models (FMM) thrust area of the NEAMS project to identify crucial gaps in simulation capabilities at the sub-continuum scale. Any significant code and method development needed at the sub-continuum scale will be undertaken in coordination with FMM. A wider scientific community already uses an extensive computational infrastructure of codes, methods, and simulation techniques to model many of the phenomena potentially of interest to waste forms, and the sub-continuum properties characterization needed for the WF IPSC simulation system will leverage this prior art whenever possible.

Many of the computer codes used in sub-continuum simulations relevant to waste forms will be commercial or otherwise proprietary codes, where the source code may not be directly accessible, or open source tools codes without formalized software quality practices, or "research" codes developed and used by individual investigators without any formal distribution. This heterogeneous software environment must be integrated into the materials properties characterization needed for the NEAMS IPSC WF modeling suite, but it poses a daunting challenge for a controlled overall simulation system required to

propagate uncertainties through a hierarchy of simulations. Regardless of the origins of the simulation code, all simulation results that enter the data flow of the NEAMS IPSC WF process will be required to demonstrate qualification: documented verification evidence of suitability, establish reproducibility and traceability, and validation for application of the code to the computation of the quantities of interest.

Requirements

Requirements for the sub-continuum phenomena will be communicated through a PIRT process, in response to an inadequacy in a quantity of interest within a constitutive model for the high-fidelity THCM level in the hierarchy.

Verification

All sub-continuum materials properties simulations which feed into the data flow of the NEAMS IPSC WF, using codes either developed within the auspices of the overall NEAMS project or from external sources, will demonstrate verification evidence to qualify them for the target application. Code verification will often be *ex post facto*, on existing code, for which the source may not be available. This verification may take the form of the simulation satisfying internal consistency checks (e.g. a variational principle), giving satisfactory solutions to solved problems (e.g., comparisons to analytic solutions), or convergence tests against computational model parameters (e.g., numerical quadratures in codes), and combinations of these. Code comparisons, or benchmarking, may be useful to establish equivalency between a two codes, but alone do not demonstrate verification of any code. Any code used to produce simulation data that feed into the data flow of the NEAMS IPSC WF will be traceable and reproducible. At minimum, this entails archiving the versions of the software used in the simulations, along with sufficient documentation of the input and computational model such that a given simulation could be recovered and repeated at a later date—traceability, reproducibility, and transparency.

Validation

Simulations that feed into the data flow of the NEAMS IPSC WF will present validation evidence, sufficient to credibly estimate the magnitude of the errors of the simulation in predicting the quantities of interest. This error analysis usually takes the form of comparisons to experimental data. Directly comparable experimental data available in the literature is often sparse, and adequate validation may entail coordinating with a robust experimental program to acquire additional data to for validation. This validation evidence will be documented and archived along with the simulation predictions.

Uncertainty Quantification

Sub-continuum simulation are frequently “first principles” or deterministic, in the sense that there are few or no free parameters to manipulate. The fundamental accuracy is limited by the fidelity of the choice of the underlying physical model (e.g., a specific flavor of density functional in quantum simulations, or a particular form of interatomic potential in an atomistic molecular dynamics simulation). An estimate of the sensitivity of the simulation predictions to the form of this physical approximation should be provided. Numerical uncertainties arise in the construction of computational models, such as integration grids and reciprocal space sampling for solid state quantum code, or length and time scales for dynamical simulations. Simulations will document convergence against such boundary conditions and computational model definitions, and estimate uncertainties in the predictions from these aspects of the simulations.

4.4 Upscaling from Sub-continuum to Continuum Models

It is recognized that upscaling—communicating information between different temporal and physical scales—is central to the success of a science-based program for prediction of WF performance. Current upscaling techniques to bridge between scales are mostly *ad hoc*, application-specific, and are generally not adequate for coupled non-linear process in heterogeneous media. New computational tools and methodologies are needed for linking different scales and representing processes to obtain high-fidelity predictions beyond the range of conditions and scale at which models and parameterizations were developed. Success stories are few.

It is not possible to use sub-continuum materials models for large scale "waste form to near-field scale" simulations due to the length and time scales involved. The sub-continuum models will help provide parameterization for constitutive models used by the high-fidelity THCM software and, moreover, will identify and give improved understanding of the phenomena that are incorporated into the physical abstraction represented by a constitutive model. Hence, the fidelity of a constitutive model is dependent on both an abstracted physical model and the numerical realization of that model.

Development of many required intermediate length-scale models for simulating WF performance is incomplete, both in the form of the model chemistry and in the parameterization of that model chemistry. Many pertinent parameters, such as aqueous activity coefficients, have yet to be reliably computed. Molecular level calculations have a more direct impact on understanding relative rates of physical processes. These calculations will have associated errors and uncertainties associated with them, which, in principle should be propagated into the next length scale. Complicating this simple propagation of uncertainties is that the next higher scale will usually recalibrate the parameters, fitting the model parameters to achieve validation or internal consistency (e.g. thermodynamic continuity between different regimes) at the next scale. This is the principal means by which sub-scale mechanistic information is propagated into the next scale, while allowing the flexibility for obtaining quantitative agreement (validation) with experimental observations at the next scale. This form of upscaling has proven to be the most generally successful approach and is an iterative process between the simulation scales (e.g., a candidate chemistry and initial parameterization is tested and calibrated, if validation is inadequate for a desired level of confidence, a new chemistry or parameterization is proposed, and the constitutive model refined until adequate validation is achieved). The identification of new phenomena, and the quantitative requirements will be updated in the PIRT as this iterative refinement of the model progresses toward a validated model.

The degree to which (1) a parameter evaluation at a sub-scale will be valid in the abstracted model of the next scale and (2) the uncertainty in that parameter is determined by the uncertainty of the evaluation of that parameter at the sub-continuum scale, depends on the degree to which the parameter value is determined by the recalibration at the higher scale rather than the computed value at the lower scale. This recalibration is necessitated by the inability to do a complete simulation using complete physics within a sub-scale model and the physical abstraction of a constitutive model discarding many of the less significant phenomena. The remaining constitutive model parameters must compensate for the absence of the hidden phenomena, and the parameter of interest loses some of its unique identity as a quantitative representation of the nominal phenomenon of interest. Different constitutive models will have differing levels of fidelity with which their parameters quantitatively represent the phenomena from which they are nominally composed.

In the limiting case where there is a direct and close correspondence between the phenomena computed at the sub-scale and the dominant phenomena in the constitutive model, the computed sub-scale uncertainties might be propagated mostly unchanged. Conversely, if an abstracted model requires significant recalibration, indicating a weak quantitative link between the uncertainties in the quantity of interest evaluated at the sub-scale and the ultimate values of the parameter in the constitutive model, the

uncertainties in a parameter computed at the sub-scale might need to be discounted in favor of the uncertainties in the parameter determined from calibration tests. Current best practice requires judgment of experienced subject matter experts. Methods for propagating uncertainties across a scale boundary are lacking, and need further development.

4.5 High-Fidelity Continuum Models

The central high-performance part of the NEAMS IPSC WF code suite is the high-fidelity THCM layer in the modeling hierarchy. The high-fidelity THCM code will model non-linear, highly coupled physical phenomena, and be composed of multiple interacting software components. The exceeding complexity of the phenomena and software used to simulate it and the numerical challenges of modeling a non-linear, highly coupled environment place stringent demands for detailed verification and validation. It is anticipated that much if not most of this code will need to be developed and deployed under the auspices of NEAMS IPSC WF, and thereby directly benefit from use of sophisticated software quality engineering practices.

Verification

The development of the high-fidelity THCM software will follow sophisticated software engineering practices, with version control, comprehensive test suites, unit testing, regular regression testing, build tests, etc., so that to the extent possible verification will be built in. An infrastructure will be constructed for the development of the software that will enforce these practices. Verification testing will follow well-established practices used previously [5]. Verification activities will be designed to establish confidence that the calculations are correct, such as using the modeling tools, sub-models, and a well-defined set of inputs in comparisons with numerical predictions of analytically known solutions or manufactured solutions. Modularity in the software will be leveraged to verify components in the software and associated libraries, e.g., mesh generation software or solvers, independent of the larger THCM software package. Verification activities will include verification of model inputs entered into the NEAMS IPSC WF input database and verification of the implementation of the sub-model abstractions within the Integrated Assessment Code. The NEAMS IPSC WF modeling suite inputs will be checked, controlled, and documented to maintain traceability, reproducibility, and transparency.

Validation

Typical model validation compares a model's results with experimental measurements and/or field observations. However, such measurements will be impossible to obtain for the high-fidelity THCM software and the Integrated Assessment Code at the (geologic) temporal and spatial scales of interest for post-closure repository performance. Validation for short-term WF performance will be demonstrated using all available data for short time scales, and will entail coordination with a robust experimental program. Predictions of the NEAMS IPSC WF modeling suite at geological scales will be extrapolated from this foundation and will be validated using corroboration, technical review, and natural analogues.

Calibration activities and evaluation of the initial/boundary conditions for the NEAMS IPSC modeling suite will be performed to establish model convergence, and demonstrate validation to the desired/required confidence. Validation inadequacies may indicate either a suboptimal parameterization within the constitutive models, or an insufficient model that requires additional phenomena. Detection of a validation inadequacy that is not remedied through calibration indicates that sub-scale phenomena need to be reinvestigated, and will propagate requirements to the sub-continuum materials properties characterization effort to identify and better quantify the phenomena of interest.

Uncertainty Quantification

There are numerous techniques for UQ analysis, including: (1) sampling-based methods; (2) reliability-based methods; or (3) methods based on stochastic expansions. The first class of methods includes standard (Monte Carlo) sampling, importance sampling, and Latin hypercube sampling (LHS). The advantage of these methods is that they are simple to implement, readily provide error estimates caused by limited sampling, and converge at the same rate regardless of the number of underlying uncertain variables. However, convergence rates are slow and some kind of model reduction technique is usually necessary. Latin hypercube sampling is very popular for use with computationally demanding models because its efficient stratification properties allow for the extraction of a large amount of uncertainty and sensitivity information with a relatively small sample size; LHS has been applied to previous studies of this type with great success [6].

The second class of methods is specifically designed to predict probabilities of failure with minimal computational effort. Reliability-based methods solve an optimization problem to locate the most probable point of failure, and then quantify the system reliability based on its location and an approximation to the shape of the limit state at this point. Gradient-based solvers are commonly used to solve this optimization problem, which may fail to converge for non-smooth response functions with unreliable gradients or may converge to only one of several solutions for response functions that possess multiple local optima. In addition, the evaluated probabilities can be adversely affected by low-order limit state approximations that may be inaccurate [7].

Stochastic expansion methods can be viewed as an extension of traditional techniques for approximating the solution to deterministic differential equations (e.g., finite element analysis) to the case where the underlying set of equations exhibits some uncertainty. Specific techniques include polynomial chaos expansion (PCE) and stochastic collocation (SC). Algorithms for PCE and SC are closely related in that both seek to capture a functional representation of the relationship between random variable inputs and key response outputs. Whereas PCE forms coefficients for known orthogonal polynomial basis functions, SC forms interpolants for known coefficients. Under certain conditions, these methods exhibit similar, if not identical, exponential convergence rates in the statistics of interest. This class of methods is relatively new and still under development; accurate assessment of approximations errors are not yet well quantified.

There are two approaches/frameworks to UQ: a non-intrusive (sampling-based) approach and intrusive (direct or embedded) approach. In a non-intrusive approach to UQ, simulations are used as black boxes and the calculation of response metrics of interest is based on a set of simulation response evaluations. Hence non-intrusive approaches have been traditionally used when very complex and extensive applications already exist and retrofitting them to gain access to internal variables in the code is not feasible. The intrusive approach is termed as such because it requires new solvers/codes designed for the reformulated system of equations. The advantage of the intrusive approach is that it finds, for example in the PCE method discussed above, the representation of model outputs by a one-time solution of the reformulated model [44]. Because we have the opportunity to develop new solvers/code for computational modeling of the waste-form and near-field environment from the ground up under this program, our intent is to research, evaluate, and pursue the embedded approach.

A byproduct of uncertainty quantification will be sensitivities of the simulation results to the various input parameters. The sensitivity analyses will be used to identify important phenomena/parameters and help refine the PIRT, and the SA also will be used to quantify the requirements for sub-continuum scale evaluations of the quantities of interest within the constitutive models.

4.6 Upscaling From High-Fidelity Continuum Models to Surrogate Models

Upscaling from high-fidelity continuum models to the surrogate models that will be used in the Integrated Assessment Code will make use of and, where appropriate, extend approaches similar to those used previously for other nuclear waste programs [5,10]. It is envisioned that these approaches can range all the way from relatively simple cases on the lower end to the more sophisticated at the higher end. At the simplest end of the range, the surrogate model may use something as straight-forward as a lookup table and interpolation of results that are provided by the Hi-Fi continuum code (e.g., the porosity surface method used by WIPP to indirectly couple mechanical closure with two-phase fluid flow calculations [10]). At the more sophisticated end, full-up system-scale coupled continuum analyses may be performed by the surrogate model albeit with significantly coarser discretizations (and perhaps different partitioning of materials to represent the system and different solution techniques) than those used for the Hi-Fi continuum model.

Much of the discussion on uncertainty and its propagation in the previous subsection "Upscaling from subcontinuum to continuum models" is also germane to the issue of continuum-to-surrogate upscaling, having analogous threads throughout, and will not be repeated. Suffice it to say that the propagation of uncertainties in this upscaling from continuum to surrogate will be dependent on the nature of the upscaling that is used in the generation of the various surrogate models across the range discussed above. Different surrogate models will have differing levels of fidelity with which their parameters will quantitatively represent the phenomena from which they are nominally derived. As such it will be necessary to develop a flexible way of handling and adapting the uncertainty quantification and propagation appropriately to accommodate the range of approaches anticipated in the upscaling.

4.7 Surrogate Models: Integrated Assessment Code

The Integrated Assessment Code is intended to execute very quickly for the purpose for generating statistical information needed for the performance assessment and decision analysis. The surrogate models contained within it are meant to mimic the essential features of the high-fidelity THCM simulations with sufficient accuracy to satisfy certification requirements to a desired confidence.

Requirements

The requirements will be defined by Use Cases flowing from the certification requirements imposed on NEAM WF IPSC from the licensing authority. The results of the performance assessment and decision analysis will be the output used for certification.

Verification

The NEAMS IPSC WF modeling suite inputs will be checked, controlled, and documented to maintain traceability and transparency. Confidence in the methodology of and inputs to the NEAMS IPSC WF modeling suite will be provided through:

- selection of input parameters and/or input data from validated supporting analysis model reports,
- model calibration activities and/or evaluation of the initial/boundary conditions for the NEAMS IPSC WF modeling suite, establishing model convergence, and
- evaluation of the impacts of uncertainties on model results.

These three activities should demonstrate that: (1) the NEAMS IPSC WF modeling suite's input parameter values from source documents, as well as those parameter values that are calculated by the NEAMS IPSC WF modeling suite, are correctly propagated throughout the Integrated Assessment Code;

(2) the Integrated Assessment Code is stable in terms of the number of realizations, the length of model time steps, and spatial discretization; and (3) that the uncertainty in model inputs is propagated through and correctly accounted for in the Integrated Assessment Code.

Validation

Validation concerns for the Integrated Assessment Code follow the same line of reasoning as for the Hi-Fi codes. The following post-development methods will be used to demonstrate the Integrated Assessment Code validation with respect to intended use and desired level of confidence:

- Corroboration of Integrated Assessment Code results with analogue studies or other relevant observations not previously used to develop or calibrate the model
- Confidence building through incorporation of the recommendations and observations provided by technical reviews conducted by external experts regarding a preliminary version of the NEAMS IPSC WF modeling suite
- Corroboration of abstraction or sub-model results to the results of the validated mathematical models (Hi-Fi) from which the abstraction or sub-model was derived
- Corroboration of NEAMS IPSC WF modeling suite results with the results obtained from auxiliary analyses (including benchmarking) as a means of providing additional confidence.

Uncertainty Quantification

The discussion on uncertainty quantification in the Section 4.5 is also applicable to the surrogate models discussed here.

4.8 Summary

Every component of the NEAMS IPSC WF system will use qualified software that is verified and validated for the simulation of the quantities of interest. The verification will include traceability and reproducibility. Requirements for simulations will be managed through the PIRT, a dynamic assessment of the state of knowledge and requirements for the NEAMS IPSC WF to successfully predict WF performance to a specified confidence.

5. Use Cases and Utilization Methodology

Use cases describe the outwardly visible requirements of a software system (Schneider and Winters, 1997). They are used in various software development phases. The use cases described in this section and Appendix B are intended to provide a high-level description of the NEAMS waste form IPSC system planned for development. These use cases are developed according to a top-down approach that starts with the top-level requirements for a performance assessment system (surrogate models) and then propagates these requirements down scale to the high-fidelity continuum and sub-continuum models. The development of use cases is an iterative process, with high-fidelity continuum and sub-continuum models influencing performance assessment. The use cases documented here will be revised and refined the project progresses.

As an integral part of the use case development, we have also identified risk factors related to the development of the NEAMS waste form IPSC system. The major risk factors are:

- funding continuity,
- integration among software developers, PA experts, and process modelers, and
- accommodation to a diverse group of users (repository license applicants, regulators, stakeholders, decision makers, researchers, etc.).

5.1 Performance Assessment with Surrogate Models

Long-term performance assessments (PA) of waste forms in geologic repository environments require consideration of coupled thermal, hydrologic, chemical and mechanical processes that span multiple spatial and temporal scales. A PA calculation generally requires a large number (hundreds to thousands) of model simulations to quantify the effect of model parameter uncertainties on the predicted repository performance. A surrogate model – a simplified version of the corresponding high fidelity (hi-fi) model - is designed to speed up PA calculations while still capturing the essential behavior of the hi-fi model. For this purpose, a surrogate model must be sufficiently robust and fast in terms of code execution.

A surrogate model can be as simple as a response surface (e.g., look-up table or neural network) or can be a model with reduced dimensionality and couplings among different processes. It is perceived that surrogate models combined with appropriate integration/analysis tools will ultimately constitute a waste form PA system, in which self-contained surrogate models (or code modules) will be linked with high flexibilities to accommodate specific repository designs (

Figure 5). A PA system constructed in this manner will be sufficiently flexible to handle different disposal environments (e.g., salt bed, volcanic tuff, clay, granite, deep borehole disposal, etc.) and various waste forms (e.g., glasses, metallic alloys, and ceramics). This PA system will provide multiple alternative models for a specific set of physical/chemical processes, so that the users can choose various levels of modeling complexity based on their modeling needs.

Unlike the high-fidelity model, this PA system will be mainly limited to handling unidirectional linking among code modules at each time step (i.e., there will be no iterative feedback between processes) and it will run model simulations on coarse grids. In each code module, however, full couplings among various physical/chemical processes are expected; therefore, the degree of model granulation is a key factor that must be considered in surrogate model development. Finally, for regulatory compliance applications, the PA system must be designed to ensure the transparency, traceability, reproducibility, and retrievability of simulation results.

The detailed use cases for the surrogate model are provided in Appendix B. These use cases are grouped into two general categories: the system level and the subsystem level. At the system level, the use cases describe possible scenarios for the application of a PA system:

- Set up access permissions for various categories of users
- Define simulation domains for a disposal system (e.g., waste panels or drifts)
- Define the types of waste forms, the types of waste containers and their distributions in the disposal system
- Define state variables of each simulation domain and associate each simulation domain with a set of model operations (i.e., process models or code module)
- Determine model parameters by fitting a model to experimental data
- Determine the statistical properties of the parameter distribution or probability density function to be used in the simulations
- Perform multiple realizations for a given scenario
- Run single realization calculations by using the means, medians, specified constants (or flow fields), or specified percentile values for the model parameters
- Perform uncertainty analyses and construct statistical results for regulatory compliance
- Identify important parameters that control total system performance through sensitivity analyses
- Determine the uncertainty related to the surrogate model abstraction and simplification
- Verify and validate a code module or a linked set of code modules against a set of testing cases
- Perform regression tests against a set of established testing cases
- Visualize the temporal evolution of the state of each individual simulation domain the user selects
- Provide a graphic interface to wrap and execute a self-contained code
- Run multiple independent codes sequentially according to a specified data flow among them

A performance assessment system constructed from surrogate models consists of six subsystems. The use cases of the subsystem level are formulated according to the functionalities of each subsystem and each use case provided is corresponding to a code model that needs to be developed:

- Basic operations/functions/integration tools: Provide miscellaneous functions and tools for model linking
- Thermal processes: Provide necessary code modules for modeling heat generation and conduction in the near-field environment of a disposal system
- Hydrologic/transport processes: Provide necessary code modules for calculating flow fields and the related radionuclide transport in the disposal system
- Mechanical processes: Provide necessary code modules for simulating rock damage and deformation around the excavation of the repository and their impacts on the integrity of an engineered barrier system
- Chemical processes: Provide necessary code modules for modeling the chemical evolution of the near-field environment
- Databases: Provide a data warehouse for storing input parameter data, PA results, and the other data supporting the derivation of the input parameter values and PA calculations (Rechard, 1995)

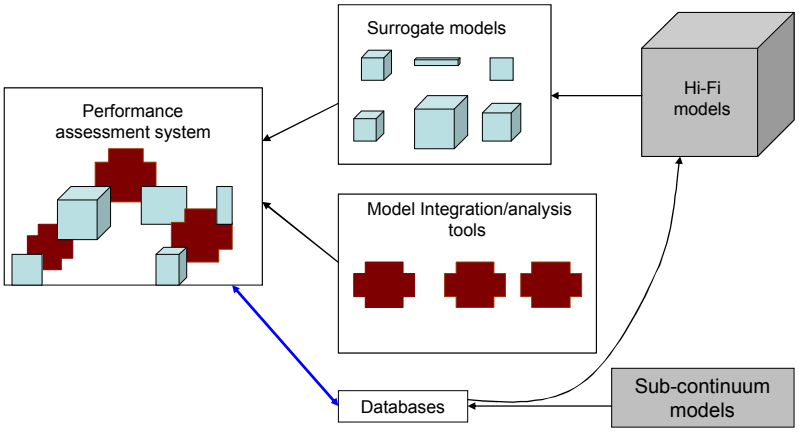

Figure 5. Construction of a performance assessment (PA) system from surrogate models. The relationship of surrogate models to high-fidelity and sub-continuum models are also indicated. The PA system repeatedly reads the input parameter values from and saves the simulation results to the databases during the model execution.

5.2 Continuum Analysis and Abstraction to Surrogate Models

Continuum models, or high-fidelity models—that range from waste form to waste package to EBS buffer / emplacement drift / near-field domain—are designed to evaluate the integrated coupled thermal, hydrologic, mechanical, and chemical processes involved in waste isolation. Simulations of these processes typically involve large three-dimensional meshes, simulate time periods of tens of thousands of years, and require coupling between the THCM processes. The coupling between THCM processes may be fully integrated, two-way coupling contained in a single code, or one-way coupling that requires periodic communication between codes developed to model the different processes. Additionally, the continuum models need to be sufficiently flexible to handle different geologic disposal environments (e.g., salt, volcanic tuff, granite, etc.), engineered barrier design options (e.g., waste form and waste package types, backfill, deep boreholes, etc.), and various waste forms (e.g., glasses, metallic alloys, and ceramics). For a regulatory compliance application, this system must be designed to ensure the transparence, traceability, reproducibility, and retrievability of simulation results.

The continuum model use cases described in Appendix B identify the individual analyses required to model some aspect of the interaction between the waste form and the emplacement environment. Each case is tied directly to one or more specific phenomena identified in Table 3. For each use case, a description of the required input information, analytical decision made by the modeler, and expected results are listed. In addition, each use case has a list of coupled parameters, along with an explanation of their importance: input parameters that will be required as the simulation progresses through time; and output parameters that will be required by other process simulation codes during the progress of their simulations.

Unlike the surrogate models, the use cases for the high-fidelity models are focused on the couplings among different physical/chemical processes. Each use case thus has a list of coupled parameters, both input and output, that will be required as the simulation evolves through time. The use cases for the high fidelity models are categorized into the following three groups:

Thermal-Hydrological Processes

Two use cases are formulated for the continuum (or high-fidelity) models for thermal-hydrological processes. Here flow through variably-saturated porous media implies both gas and liquid flow. We mention flow through variably saturated porous media as this is a superset of the needed capability. This use case also encompasses flow through the saturated zone and flow through the unsaturated zone. Gas generation models from boiling, microbial activity, and chemical reactions will be incorporated as necessary.

The first use case assumes that the simulation domain geometry is fixed and the only necessary coupling is to the chemistry code. This coupling can be one-way, for which a flow field is given from the thermal-hydrological code and radionuclide release and transport are modeled accordingly. We also include two-way coupling as a possibility, for which the geochemical evolution in the domain could affect the flow field via thermo-physical property changes.

The second use case captures the first, but also includes the effects of evolving domain geometry from various processes such as drift collapse, precipitates affecting pore openings, rock falls, waste package corrosion breach and degradation, flow through breach openings, etc. This use case requires coupling of the thermal-hydrological code to both the chemistry and mechanical codes. Again a variety of coupling procedures will need to be available from one-way coupling to two-way coupling either loosely or tightly.

Mechanical Processes

All the mechanical cases at the continuum model level have direct input into one of the following processes critical to waste isolation:

- Mechanical failure of the waste form or waste package - such a failure would introduce water or contaminants to the waste form
- Mechanical change to the porosity or permeability of the host rock - such a change would affect the hydrologic flow and transport of radionuclides from the near field to the far field environment

The use cases in this group describe the requirements for modeling mechanical processes related to:

- Closure of the drift around the waste package due to salt creep or clay deformation
- Thermal-mechanical behavior of the drift and the potential for rock fall event with sufficient energy to cause mechanical failure in the waste package or waste form
- Thermal-mechanical behavior of the drift and the change in fracture apertures and permeability in the near field
- Hydroscopic swelling of bentonite backfill and the resulting stress changes on the WP/WF, closure of fractures/interfaces in backfill (i.e., change in permeability)
- Effect of seismic event on waste form and waste package, determine if mechanical failure can occur
- Closure of the drift around the waste package due to salt creep, and the effect of that creep on the salt backfill around the waste package and waste form

Chemical Processes

Use cases for chemical processes at the continuum level are focused on the couplings among various chemical reactions in waste forms, waste packages and the near field environment as well as the couplings of these reactions with thermal, hydrological, and mechanical processes. These use cases describe the requirements for simulating the following processes:

- Evolution with time of (1) waste form composition and (2) radionuclide isotopic composition and distribution within the waste form and inside the waste package
- Chemistry of incoming water into the emplacement drift
- Evolution of water chemistry from interaction with ground support and other introduced materials in the emplacement drift
- Evolution of water chemistry from interaction with backfill around the waste package in the emplacement drift
- Evolution of water chemistry from interaction with rockfall rubble around the waste package
- Uniform corrosion process and penetration of waste package wall
- Localized corrosion (pitting and crevice corrosion) process and penetration of waste package wall
- Stress corrosion cracking (SCC) process and penetration of waste package wall
- Corrosion degradation of waste package internal structural materials upon initial breach of waste package
- Corrosion degradation of waste form canister upon initial breach of waste package
- Waste form degradation, radionuclide release and mobilization from waste form, and in-package chemical environment inside breached waste package
- Radionuclide release from breached waste package and transport in the EBS

5.3 Sub-continuum Analysis and Upscaling to Continuum Models

The sub-continuum regime refers to materials properties characterization performed in support of developing, maintaining, and updating materials properties databases, or constitutive models, needed as input quantities to the high-fidelity continuum analysis. The range of materials phenomena defined to be "sub-continuum" include atomistic processes (chemistry and migration of molecular species) through meso-scale (phase stabilities, dislocation dynamics, cracking) and could include nominally "continuum"-scale phenomena that are sub-scale to the continuum models used in the high-fidelity continuum analysis. The producst of the sub-continuum models are mechanistic models of materials behavior upscaled into constitutive models—the physical models and their parameterization—that describe material behavior in the regimes required for the high-fidelity simulations.

The relationship of the sub-continuum scale analysis to the continuum scale analysis is of hierarchical model-passing. The constitutive models that are "passed" take both a functional form (a physical model), and a numerical realization (parameters) for that model.

It is expected that these materials models can and often will be refined and calibrated (fit) at the continuum-scale to achieve internal consistency (e.g. thermodynamic consistency) or to achieve validation for continuum scale simulations with respect to continuum-scale experimental observations. The development of constitutive models is an iterative process, involving feedback between the continuum scale simulations and the sub-continuum level. These properties might be thermal-hydrological (e.g., diffusion constants of chemica(radionuclide-bearing) species through porous media or aqueous activity coefficients), or mechanical (e..g., deformation in the waste form as a result of physical processes), or chemical (e.g. temperature- and pH-dependent kinetics for the dissolution or reprecipitation of a waste form in an aqueous solution).

Requirements for sub-continuum analyses are propagated downwards from the requirements of the high-fidelity continuum simulations. Sub-continuum analyses are triggered by a lack of a constitutive model

for a given physical or chemical process, or a demonstrated inadequacy in an existing constitutive model. The inability to achieve adequate validation in a high-fidelity continuum simulation of a physical system is an indication of an inadequacy in a constitutive materials model. The inability to achieve a unique solution for a given physical system indicates that the constitutive/conceptual model needs additional constraints from the sub-continuum analyses. The inability to reduce uncertainties due to the materials model is an indication that the uncertainties in parameters of the materials model need to be reduced. In the latter case, the sub-continuum analyses will be focused on improving the quality of the information concerning the quantity of interest. A validation inadequacy will entail either developing an alternate parameterization of a physical model (improving the quality of the information for quantities of interest), or a refinement of the conceptual model for the physical phenomena (a change in the abstraction of the physical/chemical processes) included within the model. Alteration of the physical constitutive model will usually entail modification of the associated PIRT, to include additional phenomena, or to alter the importance and state of knowledge of an existing phenomenon. This process will iterate between the continuum and sub-continuum scales until a constitutive model is developed that achieves adequate validation for the high-fidelity continuum simulations.

It is recognized that "upscaling" is central to the success of the program. Upscaling is the process of propagating sub-continuum mechanistic descriptions into continuum models. Current upscaling techniques to bridge between scales are mostly *ad hoc*, application-specific, and generally not adequate for coupled non-linear processes in heterogeneous media, such as those encountered in the immediate vicinity of the degrading waste form. It is not possible, nor desirable to incorporate all plausible sub-continuum physical phenomena into a computational "waste form to near-field scale" simulation. Rather, the goal is to achieve the simplest, most compact model consistent with achieving the goal of a predictive high-fidelity continuum simulation of the physical system. The resulting physical abstractions will span the spectrum from strictly phenomenological, with weak or no correlation with individual sub-scale mechanistic processes, to dominantly physics-based, with strong correlation with sub-scale unit processes. Any practical model will represent a compromise between this computational efficiency and greater physical fidelity. Validation inadequacies at the continuum scale will drive development of more physics-based models that have greater predictive (extrapolative) capabilities .More phenomenologically –based models havie a more interpolative nature with limited predictive capabilities beyond the range of conditions and scale at which the models and associated parameterizations were developed. New computational methodologies and tools will be needed to bridge between scales with better fidelity.

The need for sub-continuum simulation of phenomena will be adjudicated through a requirements triage. If an adequate constitutive model exists, it will be used. If the parameterization of the material model can be obtained from existing or readily performed experiments, the experimental data will be used to populate the model. If the phenomenology is insufficient and the experiments are inadequate or unavailable to provide the necessary data, then appropriate sub-continuum simulations wil; be used to compute the quantities of interest, within the conditions specified by the needs of the physical system targeted by the high-fidelity continuum simulation.

Properties characterization at the sub-continuum level will coordinate closely with the Fundamental Methods and Models (FMM) program element and with other IPSC teams involved in characterization of related phenomena, particularly within the Nuclear Fuels and Safeguards and Separations IPSC. Simulation results will be incorporated from:

- existing literature or project reports (subject to satisfaction of VV-UQ requirements),
- new university research or wider scientific community (also subject to satisfaction of VV-UQ requirements),
- NEAMS FMM program element commissioned studies of phenomena shared by other (non-WF) IPSC(s), and

- specifically commissioned studies for quantities of interest for phenomena (a) not shared by another IPSC, (b) deemed sufficiently crucial to WF IPSC mission success to require intimate control, or (c) needing simulations repeated with adequate VV-UQ for results reported elsewhere (in the above) without sufficient documentation.

The quantities of interest from the simulation will be recorded along with their source, the manner in which the results are incorporated into constitutive models, and sufficient documentation to satisfy transparency, traceability, reproducibility, and retrievability of results.

The use cases for sub-continuum processes are driven by the requirements of the high-fidelity continuum simulations. The sub-continuum materials characterization level of the hierarchy is distinguished from the surrogate or high-fidelity levels by (1) its extreme modularity, dictated by the varied chemical and physical phenomena, (2) the distributed nature of the enterprise, dictated by programmatic limitations and the range of expertise required, and (3) use of continuously evolving scientific techniques, due to ever-improving understanding of the fundamental chemistry and physics, along with improved computational capabilities. The required materials behavior corresponding to the quantities of interest in a constitutive model is decomposed into fundamental physical or chemical processes. If simulations are required, requirements and boundary conditions are specified and the appropriate tool(s) or hierarchy of tools is identified. The general categories of use cases are summarized below.

- *Bulk solid unit processes* - For quantifying unit processes (bulk structure energies, defect energies, migration barriers, grain boundary energies, surface energies, etc.) of defect chemistry in the solid state waste form or barrier elements (WF wall or cladding, containment vessel), quantum chemistry (QC) simulations based on density functional theory (qDFT) methods will be used. Computed quantities can either be directly upscaled into constitutive models or be used to parameterize atomistic simulations of materials behavior: interatomic potentials for classical molecular dynamics (cMD) simulations or kinetic Monte Carlo (kMC). Crucial input conditions: composition, pressure, temperature (and gradients thereof).
- *Bulk solid dynamic response (microscopic)* - The phase stabilities of WF as a function of composition, contaminant loading, temperature and pressure, thermal and chemical expansion, diffusion/transport of chemical species (esp. radionuclide), cracking, nucleation of phases and gas bubbles will be addressed with cMD simulations (some qDFT-MD) and kMC methods. Computed quantities can be either directly upscaled into constitutive models, or used to inform higher scales in a sub-continuum hierarchy, as in grain growth or dislocation dynamics simulations. Crucial input conditions: composition, pressure, temperature (and gradients thereof),
- *Bulk solid dynamic response (mesoscale to macro)* - Deformation rates and plasticity, heat transport, grain/microstructure evolution and growth, bubble growth and interactions will be addressed with dislocation dynamics, rate theories, and phase field methods. Quantities will typically be directly upscaled into constitutive models. This step involves homogenization from discrete physical processes into internal state variables. Crucial input conditions: composition, texture/structure/scale, pressure, temperature, chemical evolution (radiolytic).
- *Fundamental processes at solid surfaces* - Fundamental processes at a solid waste form surface, such as surface structure driven by bulk phase stabilities and sorption of simple chemical species, will be addressed with quantum chemical methods. Computed quantities will provide input to interface chemistry studies. Crucial input conditions: composition.
- *Chemistry at solid-liquid interfaces (microscopic)* - Corrosion of WF (or barrier) surface, dissolution of WF, rind (barrier layer) formation in the presence of an aqueous environment will be decomposed into unit chemical processes and modeled with either QC, qDFT or reactive-cMD, with solvation models to incorporate effects of aqueous environment. Chemical processes included elemental attacks by chemically active species, water (H^+ and OH^-), H_2O_2, O_2, CO_2, H_2, Na, Ca, and other

geological species, leading to alteration of the chemistry of the surface through reduction/oxidation, addition or removal of chemical species, formation of surface complexes, corrosion rind and barrier layer formation, and dissolution. Diffusion rates and electrochemical migration will be determined from cMD simulations. Quantitative information from chemical processes will be used to inform coarser-grained simulations of sub-continuum phenomena, and key chemical processes may be upscaled directly into constitutive models. Crucial input conditions: bulk composition and surface structure, aqueous constituents and pH, redox, temperature), transport of chemical species to and from the interface (from in the bulk or through aqueous environment), temperature and thermal gradients, electrochemical properties.

- *Transport to/from solid-liquid interface (meso-scale)* - Thermodynamics and kinetics of dissolution, corrosion and saturation effects, will be evaluated using free energy calculations to compute surface site densities, kMC, grand canonical MC, classical density functional theory, and continuum approaches. This includes evaluating the effect of pores, advection, electrochemistry, surface site densities and charged species. Simulation results awill be upscaled into constitutive models for the high-fidelity continuum simulations. Crucial input conditions: WF surface composition, geological environment, temperature and chemical gradients.
- *Aqueous chemistry* - Thermodynamic activity of all dissolved species in aqueous solutions, including high ionic strength solutions at elevated temperatures will be addressed with solvated molecular simulations for bulk aqueous speciation with selected solvated qDFT calculations. Aqueous complexation, diffusive and advective transport of solutes will be addressed with cMD. Thermodynamic calculations for. Simulation results will be upscaled into constitutive models for high-fidelity, coupled physics continuum simulations. Crucial input conditions: aqueous constituents, pH, temperature gradients, water flow.
- *Precipitation and formation of secondary phases* - Nucleation, growth rates and transport of colloids will be addressed with molecular simulations, kMC, to compute relative stabilities for input into mixing models. The results of continuum simulations of nucleation and Ostwald rule of stages (thermodynamically driven) and continuum-scale simulation of kinetic solid solutions will ultimate be directly imported into the constitutive models for the coupled-physics high-fidelity continuum simulations. Crucial input conditions: chemistry of the degrading WF surface, transport from the interface, temperature and temperature gradients, pH, aqueous chemistry.
- *Transport through porous media* - Transport of radionuclides and radonuclide complexes through porous media, such as a clay, will be addressed through molecular simulations, either cMD or cDFT. Diffusion constants will be imported directly into constitutive models. Input conditions: backfill material composition, near-field environment, temperature, aqueous chemistry.

It is noted that simulation capabilities for modeling many of these phenomena are limited. Theories, methods and codes may be lacking, understanding of physical processes may be inadequate. For example, current quantum chemical methods are inaccurate for actinide chemistry involving relativistic *f*-electrons. This shortcoming is particularly unfortunate, as there is expected to be very limited experimental data for actinide chemistry. Quantitative treatment of solvation and especially of electrochemical effects is rudimentary, and requires additional development. Extrapolating to geologic time scales is problematic for methods computationally targeted for much shorter time scales, and the inability to run validation experiments at geological time scales. The use case for these is to foster development of better methods, and using the best data achievable, and refining the constitutive models in a calibration phase.

Above all is the particular challenge of upscaling. Upscaling requires development of new ideas and techniques, often narrowly targeted to specific phenomena in given constitutive model. Much of the efforts at the molecular level can have impact on understanding the relative rates of competing physical

processes, and on understanding the molecular-level structures and mechanisms that influence events at the continuum scale. The enumeration of the important processes and their ranking has value, even if the actual rate constants computed at the molecular scale are not numerically accurate at the next higher physical scale. The enumeration of the important physical processes helps define the physical abstraction used in the constitutive model, and the computed mechanistic quantities are useful to populate the parameters of the constitutive model for use in a calibration step.

6. THCM Framework Architecture

The THCM Framework architecture illustrated in Figure 6 addresses three distinct THCM analysis perspectives: analysis workflow, multi-physics model coupling, and inter-fidelity model coupling.

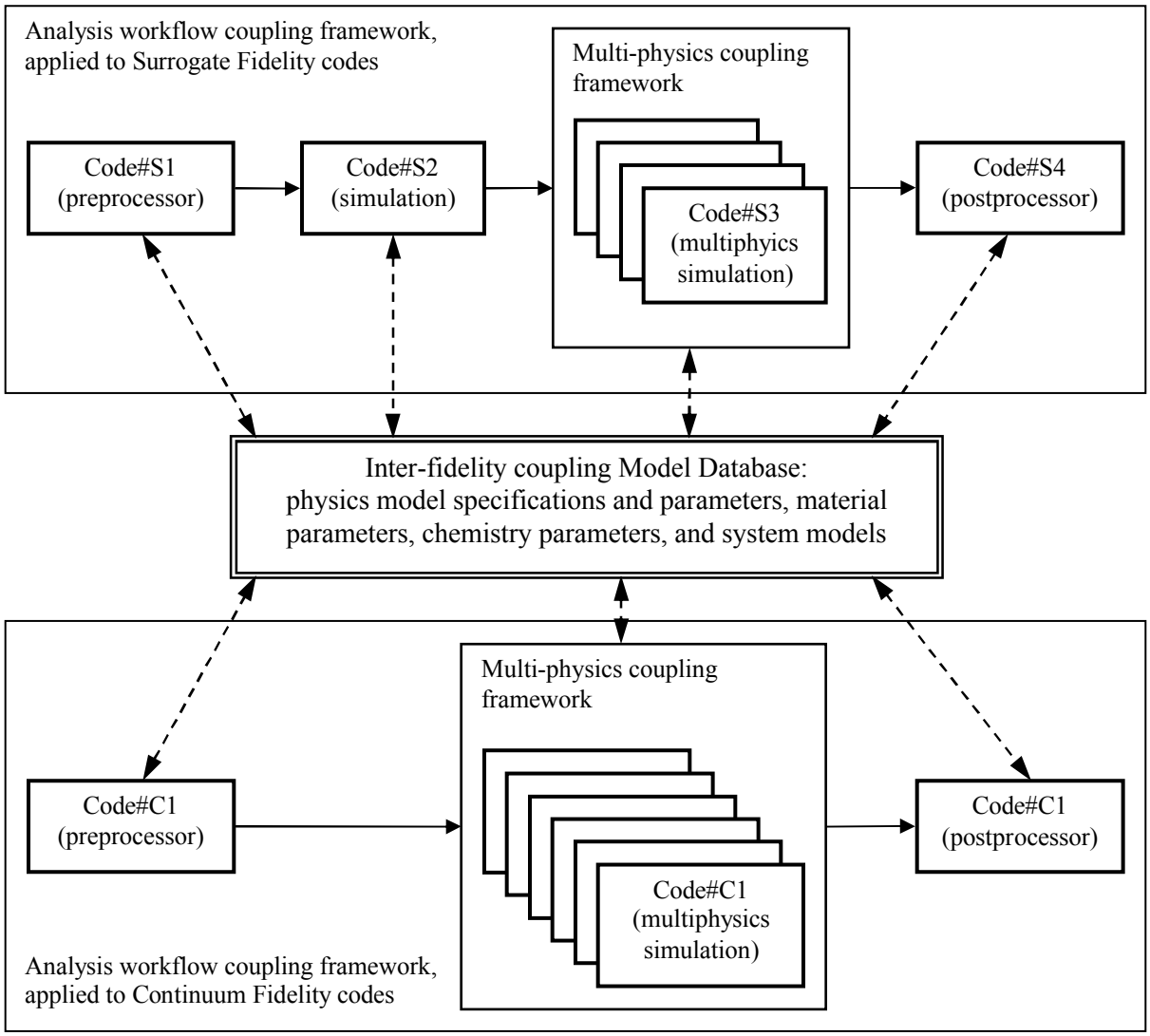

Figure 6. The THCM framework architecture is partitioned into analysis workflow, multi-physics coupling, and inter-fidelity coupling component frameworks.

Analysis Workflow

From the analysis workflow perspective an end-user runs a sequence of codes to carry out a particular analysis or assessment activity. These codes typically include simulations, preprocessors for problem setup, postprocessors for analyzing results, and data manipulation utilities. An analysis workflow framework supports the planning, performing, tracking, and reproducing of a sequence of steps for an analysis activity. Each step potentially involves selecting and configuring a code to run, generating new inputs for the current step, selecting inputs from results generated by previous steps in the workflow, executing the code with these inputs, and archiving output results. Sufficient information must be

maintained to track and reproduce the workflow step. In addition, workflow tracking information should document the end-users intent for each step in the workflow.

Multi-physics Coupling

From the multi-physics model coupling perspective an end-user or code developer creates a particular multi-physics simulation by configuring and integrating physics models. The need for integrated physics models is apparent from the diversity of coupled phenomenon identified in Section 3, Phenomena Identification and Ranking. A multi-physics coupling framework supports configuration of a simulation code by selecting component physics models and specifying how those models are coupled. Physics model coupling can be as weak as a simple feed-forward transfer of output from one model to the input of another model, as strong as solving the models' fully integrated set of equations, or an operator-splitting solution strategy. The models selected and coupling method used impact the fidelity / accuracy of the simulation results as well as the runtime resources (time and memory) required by the simulation.

Inter-fidelity Coupling

From the inter-fidelity coupling perspective simulations and analyses performed at the subcontinuum fidelity, continuum fidelity, and surrogate fidelity are correlated to support verification, validation, and uncertainty quantification (Section 4). An inter-fidelity coupling data database supports traceability of physics models, material properties, chemistry parameters, system models, and other simulation components between levels of fidelity. This repository is also anticipated to support traceability to experimental and system design data. Inter-fidelity correlation information could be as simple as identifying that two different codes model the same physics at different fidelities, or as involved as a higher fidelity model generating parameter sets that are used to approximate the corresponding physics in a lower fidelity model.

Physics models' specifications and parameters represent ubiquitous knowledge required to support analysis workflows, multi-physics coupling, and inter-fidelity coupling, A *model database* is defined to manage this knowledge to support all three coupling perspectives. The concept of a model database is common to workflow management frameworks. This concept is further developed within multi-physics coupling frameworks. For the NEAMS WF IPSC the model database concept must be further extended to include inter-fidelity coupling.

6.1 Inter-fidelity Coupling Model Database

The verification, validation, and uncertainty quantification (VV-UQ) strategy (Section 4) requires VV-UQ of simulations and analyses at lower fidelities to be supported by VV-UQ of simulations and analyses at higher fidelities. In particular, constitutive models integrated within high-fidelity continuum simulations are developed, verified, and validated with respect to sub-continuum simulations and analyses. Likewise surrogate models integrated within performance assessment simulations are developed, verified, and validated with respect to integrated high-fidelity continuum simulations and analyses. The inter-fidelity coupling framework is responsible for supporting traceability between models at these different levels of fidelity.

Inter-fidelity coupling occurs solely through a model database, as illustrated in Figure 6. This database maintains specifications for physics models, parameters for these models, and correlation / traceability between analogous models and parameters at different levels of fidelity. Model specifications convey the phenomenon or physics modeled, potential capabilities such as direct computation of sensitivities for quantities of interest, domain of intended use and anticipated accuracy over that domain, and references to verification and validation evidence.

6.1.1 Physics Model Specifications

The domain of intended use and anticipated accuracy may be quantified with respect to bounds on model parameters and problem domain. Within this domain of intended use a model may also quantify the accuracy of results produced. This quantification could be simple bounds that reflect aleatory uncertainties, or complex bounds with problem-specific dependencies. For example, the accuracy of a model may depend upon the spatial resolution of the problem discretization and accordingly have a verified rate of convergence for the solution with respect to this spatial resolution.

A physics model may have an embedded UQ capability, where sensitivities of quantities of interest to parameter or problem inputs are directly computed by the model. A model's specification includes identification of which quantities of interest support sensitivity computations. Sensitivities to quantities of interest that are not directly computed by the model may be indirectly computed through a sampling method. Results from these computations can become part of the model's specifications.

Sensitivity analysis of a particular workflow or multi-physics simulation can be used to assess the relative importance and suitability of the models integrated into that workflow or simulation. Results of such a sensitivity analysis will indicate where higher or lower accuracy / fidelity models are needed to support a particular NEAMS WF analysis. This knowledge will also provide guidance for prioritizing model development, mechanistic process modeling, and properties characterization efforts.

6.1.2 Physics Model Parameters

Physics models' parameters are derived from experimental data, first-principles simulations and analyses, and derivations from other parameters. It is expected that many of these parameters will not be exact, that they will have uncertainties which should be quantified. Parameters, their uncertainties, and their association with models will evolve as models are developed, simulations performed, and experiments analyzed. For traceability and reproducibility this model parameter data is configuration managed within the model repository. Parameters derived from other parameters must have sufficient information to reproduce that computation. Examples of such computations could be as simple as a curve fit to experimental data or as involved as carrying out a sampling-based UQ analysis over a range of input parameters to a simulation to determine the uncertainty of a given output parameter.

6.1.3 Version Controlled Database

New models will be developed, existing models modified, and model parameters revised. Analyses performed with earlier versions of models and parameters must be reproducible and traceable to support VV-UQ activities. Thus configuration management of the model database must provide version control for models' specifications and parameters.

6.2 Multi-physics coupling framework

An important type of coupling in the NEAM WF ISPC system is multi-physics coupling between different modules with the same basic level of fidelity (i.e. continuum-to-continuum or surrogate-to-surrogate model coupling). These mathematical models are approximations of the various physical processes in the WF systems environment. Many of these models are expresses as steady-state or transient partial differential equations (PDEs). These PDE models are discretized on a computational mesh [36] often resulting in square sets of nonlinear equations (in the steady-state case), or sets of ordinary differential equations (ODEs) or differential algebraic equations (DAEs) (in the transient case) [37].

In the next section, we analyze the multi-physics coupling domain. This is followed in Section 6.2.2 with an analysis of the multi-physics coupling framework with respect to UQ and optimization. Given the completed analysis of the multi-physics coupling domain, the specific requirements for the basic models and solvers for the multi-physics coupling framework are spelled out in 6.2.3. These requirements form the foundation for a basic multi-physics coupling framework architectural plan described in Section 6.2.4.

6.2.1 Analysis of the multi-physics coupling model and solver domain

The issues in multi-physics coupling are similar in the continuum and surrogate domains. The primary difference would be that the models in the surrogate domain would tend to be courser and would rely more on pre-computed data (supplied by the continuum simulations). Other than that, we can largely address the continuum and surrogate multi-physics coupling domains using the same overall multi-physics coupling architecture that is described here.

A model for transient WF physics is abstractly represented by Equation (6.1).

$$\begin{aligned} f(\dot{x}, x, p, t) &= 0, t \in [t_0, t_f] \\ x(t_0) &= x_0 \end{aligned} \quad . \tag{6.1}$$

The state equation in represents a discrete form of a single model for the transient problem in an implicit ODE or DAE form where $x \in R^{n_x}$ are the discretized state variables, $\dot{x} \in R^{n_x}$ are the variables of time state derivatives, $p \in R^{n_p}$ are parameters defining the model, t is time (with t_0 and t_f being the initial and final times the model is represented over, respectively), and $f(...) \in R^{2n_x+n_p+1} \rightarrow R^{n_p}$ is the state residual function. Once the parameters p are specified, the state equation (6.1) can then be solved for the solution of the state variables $x(t)$ as a function of time t. The parameters p can represent several different types of inputs into the model such as:

- uncertain inputs with perhaps an associated probability distribution function,
- general parameters for coupling to other models, or
- general design parameters used in sensitivity studies or design optimization.

Note that we could also specifically discuss steady-state models as well but we instead just focus on the transient problem because the issues related to the steady-state problem are typically a subset of the issues faced by the transient problem.

At an abstract level, the model in (6.1) can either represent a single compact physics model or can represent the entire set of multi-physics models integrated into a single large set of unknowns and equations. When considering a multi-physics set of equations, one can abstractly think of them as one large coupled system as shown in (6.1) or one can consider the coupled equations separately. For example, a general formulation for a set of two coupled physics models can be represented as:

$$\begin{aligned} f_0(\dot{x}_0, x_0, p_0, t) &= 0, t \in [t_0, t_f] \\ f_1(\dot{x}_1, x_1, p_1, t) &= 0, t \in [t_0, t_f] \\ h_0(p_1, x_0) &= 0 \\ h_1(p_0, x_1) &= 0 \end{aligned} \quad . \tag{6.2}$$

Here, the two sets of physical models are represented by the single-physics state residual equations $f_0(...)$ and $f_1(...)$ and the coupling between these two models is given by the coupling equations $h_0(...)$ and $h_1(...)$. In the most general case, the two physical models $f_0(...)$ and $f_1(...)$ and their sets of variables can be represented on different meshes of the same physical domain or different domains. The coupling equations $h_0(...)$ and $h_1(...)$ therefore can embody mesh transfer operations and mathematical equations needed to define the coupling. Note that in many cases, the coupling equations are simply function evaluations of the form $h_0(p_1, x_0) = p_1 - r_0(x_0)$ so typically no real "solve" is required to perform the coupling transfer calculations.

As already stated, the different multi-physics models can be represented abstractly as a single set of equations. For the dual coupled multi-physics model in (6.2), this leads to the following abstract combined DAE model.

(6.3)
$$f(x, \dot{x}, p, t) = \begin{bmatrix} f_0(\dot{x}_0, x_0, p_0, t) \\ f_1(\dot{x}_1, x_1, p_1, t) \\ h_0(p_1, x_0) \\ h_1(p_0, x_1) \end{bmatrix}, x = \begin{bmatrix} x_0 \\ x_1 \\ p_0 \\ p_1 \end{bmatrix}, \dot{x} = \begin{bmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{p}_0 \\ \dot{p}_1 \end{bmatrix}$$

These different physical models can vary in a number of important ways in a single physics or multi-physics setting. The different physical models may be strongly coupled or weakly coupled. The different physical models can represent dynamics on radically different type scales. The models may be representable as smooth continuous functions or may have significant discontinuities in the model functions. These and other factors affect how the discretization and the solution of these problems must be approached in order to be able to efficiently and accurately solve the underlying sets of multi-physics equations. Related to these issues is the ability to define and compute basic efficient forward and adjoint sensitivities as described in Section 6.2.2.

There are a wide range of strategies for solving transient multi-physics models. In some cases, the bi-directional coupling of the models may be so weak that we can fully solve one set of physics over the entire time and space domain and then we can use the converged solution from the first physics to feed into the solution of the second physics model. We will call this "feed forward" coupling and is the easiest type of coupling to implement. A general feed forward formulation for a set of two physics models can be represented by Equation (6.4).

(6.4)
$$\begin{aligned} f_0(\dot{x}_0, x_0, p_0, t) = 0, t \in [t_0, t_f] \\ h_0(p_1, x_0) = 0 \\ f_1(\dot{x}_1, x_1, p_1, t) = 0, t \in [t_0, t_f] \end{aligned}$$

In the basic feed forward model shown in (6.4), the only coupling equation between $f_0(...)$ and $f_1(...)$ is $h_0(...)$. Therefore, the first single physics equation $f_0(...)$ can be solved for completely independently of the second equation $f_1(...)$. Even in this feed forward case, there can be advantages to considering them to be a single set of equations in the analysis and in the simulation software. For example, time integration software may step the two equations together to avoid having to store the first solution $(\dot{x}_0, x_0)$ over the entire time horizon $[t_0, t_f]$.

When the models are weakly coupled or have radically different dynamic time scales and cannot be fully decoupled, it can be advantageous split up the different disparate models and solve them with different solution strategies and only keep the models in sync in less rigorous ways. This approach is known as the "operator split" approach where for example one set of physics models may be solved with an explicit

time integration method and the other physics model may be solved with an implicit time integration scheme where the two models only exchanged updated state infrequently [33].

The other end of the extreme of the extreme from the ideal "feed forward" coupling problems are multi-physics problems where the models are so strongly coupled that any attempt to decouple them in the basic nonlinear and transient solution methods will result in divergence or in substantial degradation in the performance of the numerical method [38]. In many cases, the more efficient approach to integrate the transient equations is to use a fully implicit time integration method. The classic problem with fully implicit methods is that off-the-shelf preconditioning approaches and software for solving the linear systems using iterative methods can be very inefficient when dealing with a challenging multi-physics problem. More recently, the growing trend in many of research groups for addresses these multi-physics problems with fully implicit time integration methods is to use operator split ideas to instead build physics-based preconditioners. Such an approach has proven to be very computational efficient and yet very robust for many multi-physics problems [34].

The coupling equations $h_0(...)$ and $h_1(...)$ in (6.2) may involve the transfer of data from one model to another where the models may live on different computational meshes and/or may use different basis representations for the same or related qualities. It is desirable to implement these different-discretization couplings such that smoothness of the coupling equations is preserved and basic derivatives can be computed. For example, we would like the derivatives $\partial h_0 / \partial x_0$, $\partial h_0 / \partial p_1$, $\partial h_1 / \partial x_1$, and $\partial h_1 / \partial p_0$ to all be well defined. However, if the coupling equations are not smooth, then it is not be possible to compute efficient and accurate sensitivities discussed in the next section.

6.2.2 Sensitivities for UQ and Optimization for Multi-physics Coupling

Basic forward and adjoint sensitivity computations are needed by the most efficient and effective methods for sensitivity analysis, parameter estimation, design optimization, and uncertainty quantification. In order to compute the forward and adjoint sensitivities described in this section, the underlying models must be smooth enough and well enough defined so that these sensitivities exist in the first place (See Section 6.2.3.1).

Before previewing the basic forward and adjoint sensitivity methods, first consider a general form of an observed forward problem.

(6.5)
$$f(\dot{x}, x, p, t) = 0, t \in [t_0, t_f]$$
$$x(t_0) = x_0(p)$$

(6.6)
$$d(x, p) = \int_{t_0}^{t_f} g(x(t), p, t)dt + h(x(t_f), p)$$

The form of the composite response function $d(...)$ in (6.6) shows both a distributed (i.e. integrated) response $g(...)$ and a terminal response $h(...)$. Also shown is a parameterized initial condition $x(t_0) = x_0(p)$. Using the state equation in (6.5) to (abstractly) eliminate the state variables x in (6.6) leads to a reduced response function $\hat{d}(p)$. This formulation covers the majority of different types of problems that are used in practice. A more detailed treatment of this formulation and the derivation of the forward and adjoint sensitivity computations are given in [41].

A "discretize then differentiate" [41] approach to sensitivity computations is planned because automatic differentiation can be utilized at the source code level (see Section 6.2.3.1). However, one can also derive

the forward and adjoint sensitivities at the continuous PDE level and then discretized and solve the resulting equations using any approach that seems reasonable, including operator split methods. Depending on what the sensitivities will be used for, issues of discrete consistency may or may not be important but we will not discuss these issues any further here.

6.2.2.1 Transient Forward Sensitivities

The forward sensitivity method involves differentiating (6.5) and (6.6) with respect to the parameters p which yields:

$$\frac{\delta f}{\delta \dot{x}}\dot{S} + \frac{\delta f}{\delta x}S + \frac{\delta f}{\delta p} = 0, t \in [t_0, t_f] \qquad \textbf{(6.7)}$$

$$S(t_0) = \frac{\delta x_0}{\delta p}$$

$$\frac{\delta \hat{d}}{\delta p} = \int_{t_0}^{t_f} \left(\frac{\delta g}{\delta x}S + \frac{\delta g}{\delta p} \right) dt + \left(\frac{\delta g}{\delta x}S + \frac{\delta g}{\delta p} \right)\Bigg|_{t=t_f} S(t_0) = \frac{\delta x_0}{\delta p}\Bigg|_{t=t_0} \qquad \textbf{(6.8)}$$

where $S = \partial x / \partial p$ is the integrated sensitivity of the state x with respect to the parameters p. The forward sensitivity equations in (6.7) are integrated right along with the forward state equations in (6.5). The integral in (6.8) is also accumulated while the state and forward sensitivities are being integrated.

Computing forward sensitivities scales linearly with the number of parameters n_p and therefore becomes computationally intractable for very large parameter sets. However, solving the sensitivity equations can reuse the pieces of the state solve and can be very efficient for small numbers of parameters. In addition, the forward sensitivity method can be much more efficient than using finite differencing across the entire transient simulation and is much more accurate. For example, in Sandia's finite-element code Charon, the sensitivities to 40 model parameters was computed at $1/10^{th}$ the cost of doing finite differences and yielded much more accurate sensitivities [42].

6.2.2.2 Transient Adjoint Sensitivities

The other basic approach for computing sensitivities is to use the adjoint approach. A general but complete derivation of transient adjoint sensitivities is given in [41]. Given the form of the observed forward problem in (6.5)-(6.6), the adjoint approach for computing the sensitivities (for $\partial f / \partial \dot{x}$ not a function of t) is given by:

$$\frac{\delta f}{\delta \dot{x}}^T \dot{\lambda} - \frac{\delta f}{\delta x}^T \lambda + \frac{\delta g}{\delta x}^T = 0, t \in [t_0, t_f] \qquad \textbf{(6.9)}$$

$$\left(\frac{\delta f}{\delta \dot{x}}^T \lambda \right)\Bigg|_{t=t_f} = \frac{\delta h}{\delta x}^T\Bigg|_{t=t_f}$$

$$\frac{\delta \hat{d}}{\delta p}^T = \int_{t_0}^{t_f} \left(\frac{\delta g}{\delta p}^T - \frac{\delta f}{\delta p}^T \lambda \right) dt + \frac{\delta h}{\delta p}^T\Bigg|_{t=t_f} + \frac{\delta x_0}{\delta p}^T \left(\frac{\delta f}{\delta \dot{x}}^T \lambda \right)\Bigg|_{t=t_0}. \qquad \textbf{(6.10)}$$

The adjoint equations are solved backward in time after the forward state equations have been solved. While the cost of the forward sensitivity approach scales with the number of parameters n_p, the adjoint approach scales mostly independently with the number of parameters and instead scales linearly with the number of response functions n_g. The adjoint approach is therefore the method of choice when there are

many parameters but few response functions. In addition to basic sensitivity computations, the adjoint can also be used for post-priori error estimation and subsequent adaptively (of both the spatial and temporal discretizations) [43].

When the forward problem is nonlinear, the state solution $(\dot{x}, x)$ at different points t must be accessed to compute the derivative operators that form the adjoint. The need to store or recomputed the entire forward state history can make the adjoint approach expensive and even intractable in some extreme cases.

6.2.3 Multi-physics Coupling Framework Requirements and Goals

Now that an overview of the multi-physics coupling problem and sensitivity computations have been presented, we now clearly state the requirements for the NEAMS THCM multi-physics coupling framework. Here we differentiate true requirements from goals that we have that are not requirements per say.

Requirements for the multi-physics coupling framework

Configurable coupling of independent models
Allow the specification and handling of different physics models
Allow each physics model to use a different discretization
Allow each physics model to be represented on a different mesh (or region of a mesh)
Allow each physics model to be solved with its own specialized solver or using a more generic solution method
Support the rapid development and evaluation of different solution strategies from operator split through fully implicit
Support the development and application of customized physics-based preconditioners
Support the specification of general nonlinear functions for the coupling of different physics domains
Allow coupling with external physics models at the time-step level (nonlinear Gauss Seidel or fully implicit)
VU derived requirements
Support the computation of transient adjoints to enable global error estimation
Support the needs for embedded UQ and sensitivity analysis
Runtime environments
Portability of all dependent software to the major ASC-type MPPs (affects selection of external software dependencies)
Run on parallel computers with MPI
Integration with other architectural elements
Hook into the overall workflow framework for inputs, outputs, and driving calculations.
Handle input of common parameters and data consistent with the workflow framework and database system

Other goals for the multi-physics coupling framework

Reuse existing software for the expression and discretization of the various physics models
Reuse existing software for numerical algorithms
Decouple the basic models from the solvers used to solve the problem (to allow for different solution options)

6.2.3.1 Model Smoothness

In order to be able to use derivative-based methods, the underlying models must possess a certain degree of smoothness. In order for the derivatives produced by the model to be useful, the underlying functions must be at least zeroth- and first-order continuous. This can be accomplished with carefully chosen piece-wise continuous functions (e.g. cubic splines for instance). Even when the underlying model is inherently non-smooth, one can often still come up with useful smooth approximations that can be solved with gradient-based methods and still achieve solutions to engineering accuracy.

Examples of reformulating inherently non-smooth models as smooth models abound in the literature and in production applications. For example, the inequality constrained optimization problem yields a feasible set that is inherently non-smooth as the set of constraints in the active set change. However, interior point approaches reformulate the inequality constrained problem to a smooth approximation using a weighted log-barrier term [56]. The smoothed approximate optimization problem is then solved in a sequence of inner and outer iterations where the log-barrier weighting constant is progressively decreased (in a type of continuation method) until the solution with the desired engineering accuracy is achieved.

Another example of a smooth approximation to an inherently non-smooth model is the reformulation of an absolute value using a piece-wise function with a smooth quadratic or cubic minimum near zero. Many other examples like this exist in the literature.

Even when the underlying model is technically smooth, the second derivatives can be so high that the change in the first derivatives can appear essentially discontinuous in float point arithmetic. This, for example, is the motivation for continuation methods for solving sets of equations using an inner derivative-based Newton method. In such a continuation method, an initial value of the continuation parameter(s) is chosen such that the problem function and derivatives are well behaved, allowing for an efficient approximate solution using a derivative-based inner loop. Then, the continuation parameter(s) are adjusted some and the problem is approximately reconverged and so on until the desired final values of the continuation parameters are achieved or engineering accuracy is achieved. For example, it is common to do continuation on Reynolds number when solving difficult flow problems in CFD.

Almost any non-smooth model where one has access to the model equations can be appropriately approximated with a piece-wise smooth reformulation and continuation can be used to solve the original non-smooth problem to engineering accuracy using a derivative-based method on the smooth form of the model. Note that this typically does not apply to black-box approaches where the model equations are hidden and are solved by an unknown iterative method. The only hope we have for addressing such problems is to expose nearly every implicit equation to the numerical methods to allow such issues to be controlled. I general, if one knows the source of a discontinuity and can access the underlying equations, then a smooth reformulation is typically possible. However, the relative effectiveness of the smooth reformulation and continuation approach will vary on a case-to-case basis.

In summary, if one does not take steps to address discontinuities in the model (coming from either the underlying physics or an artifact of the implementation), then one will be stuck using numerical approaches that do not use any derivative information and therefore do not allow for the use of the efficient sensitivity methods described in Section 6.2.2.

6.2.3.2 Model Derivatives

Once we have been able to create models with smooth functions, as a practical matter, one must consider the implementation and computation of model derivatives as shown in Equations (6.7) and (6.8). Manual approaches for deriving and implementing model derivatives can too easily result in incorrect derivatives and, in many cases, overly expensive derivative computations. A recent approach for computing discrete function derivatives is to use automatic differentiation (AD) [47]. When using AD, the developers only need to code the basic function evaluation in a programming language like Fortran or C++ and then an AD tool automatically produces the desired derivatives to machine precision (AD it is not a finite differencing approach). In Fortran codes, a source transformation tool like ADIFOR is commonly used. For C++, there are no source transformation tools and instead the operator-overloading approach must be used. One such library for the operator-overloading AD in C++ is Sandia's Sacado[a] package within Trilinos. Such AD tools can also efficiently and automatically compute second derivatives which are needed by some optimization methods, and needed to propagate sensitivities from data through optimal parameter values in parameter estimation optimization problems using in upscaling.

Producing the derivatives automatically using AD avoids extra programming and code maintenance work and can be very efficient in many cases. In fact, in one study the C++ AD tool Sacado has been shown to compute machine precision derivatives at 1/3rd the cost of a finite difference method [55].

6.2.4 Multi-physics Model Coupling and Solution Framework Architecture

The multi-physics coupling framework architecture includes the physics models' interfaces and solution framework. This architecture must accommodate the range of coupling and solution strategies, facilitate embedded sensitivity computations, and support embedded UQ.

The approach being taking in recent projects is to expose the basic model unknowns and equations of each physics model (i.e. as in (6.1)) and then construct different solution strategies in a plug-and-play way with existing high performance algorithmic building blocks. A separation between models and solvers is fundamental to the multi-physics coupling framework. To clearly articulate the separation between models and solvers we will define these as:

Models: A *model* is defined to be a set of coupled discretized equations of the form (6.1) which includes its discretization, representation of the discrete vectors for the states x and the parameters p, and the evaluation of the state function f. Also, the functionality and data-structures needed to compute the various derivatives shown in Section 6.2.2 also need to be embodied in a *model*. For example, the use of AD would be encapsulated within the implementation of a *model*. Note that there is a significant amount of foundational capabilities needed to fully implement a *model* including parallel mesh and discretization software, parallel vector and matrix objects, and many other categories of software. In general, we will require that the evaluation of a model's functions have no side-effects and therefore are ignorant of the particular state of any solution algorithm. Making the model's evaluations stateless (i.e. not affecting the observed state of object) is a key requirement needed to decouple models and solvers and enhance basic composeability of these objects.

Solvers: A *solver* defines or carries out a specific numeric algorithm using a mathematical formulation with functions provided by one or more models. Solvers can represent anything from basic linear solver

[a] http://trilinos.sandia.gov/packages/sacado/

and preconditioning strategies, through nonlinear solvers, up to through and beyond transient solvers and optimization and even UQ methods. Individual component solvers should also be composeable so that higher-level solver objects can be created from more basic building blocks [42].

Given this basic separation between models and solvers and the need for the flexible composeability of these objects, we now go into a little more detail into the architectural specification of these two types of software objects in the THCM Framework.

6.2.4.1 Physics Model Component Interfaces

Here we describe a general approach to abstracting single physics and multi-physics model components such that they allow for the flexible composition with different solution approaches and also support the needs for sensitivity analysis, UQ, and optimization. This description is illustrated through an existing research code: the ModelEvaluator package within Trilinos (Appendix C.3.1).

The ModelEvaluator is an abstract object-oriented interface expressed in C++ to represent basic model equations that take the general form

$$\begin{aligned} & f\left(\dot{x}, x, \{p_l\}, t\right) = 0, t \in [t_0, t_f] \\ & g_j\left(\dot{x}, x, \{p_l\}, t\right), j = 0 \ldots N_g - 1 \end{aligned} \tag{6.11}$$

where $f(...)$ is the state residual model as shown in (6.11) and $g_j(...)$ are different sets of auxiliary response functions. Here, the parameters $\{p_l\}$ represent a set of $l = 0 \ldots N_p\text{-}1$ of different parameter sub-vectors that can be mapped into different types of qualities such as design parameters, uncertain variables, coupling variables with other physics models, and other types of inputs. The ModelEvaluator interface uses a stateless model evaluation where the values of all of the inputs $(\dot{x}, x, \{p_l\}, t)$ are passed into the model evaluation and all of the output functions f and $\{g_j\}$ computed are passed out. Insisting on a stateless model evaluation greatly simplifies the development of the implementation of the model in the ModelEvaluator subclasses and also greatly simplifies the development of powerful numerical solver algorithms built on top of the ModelEvaluator.

In addition to the basic function outputs f and $\{g_j\}$,a ModelEvaluator object can also sign up to support the evaluation of various derivative objects such as:

$$\begin{aligned} & W = \alpha \frac{\partial f}{\partial \dot{x}} + \alpha \frac{\partial f}{\partial x} \in R^{n_x \times n_x} \\ & \frac{\partial f}{\partial p_l} \in R^{n_x \times n_{p,l}}, l = 0 \ldots N_p - 1 \\ & \frac{\partial g_j}{\partial \dot{x}} \in R^{n_{g,j} \times n_x}, j = 0 \ldots N_g - 1 \\ & \frac{\partial g_j}{\partial x} \in R^{n_{g,j} \times n_x}, j = 0 \ldots N_g - 1 \\ & \frac{\partial g_j}{\partial p_l} \in R^{n_{g,j} \times n_{p,l}}, j = 0 \ldots N_g - 1, l = 0 \ldots N_p - 1 \end{aligned} \tag{6.12}$$

The composite derivative operator W is used in a great number of implicit time integration methods. The scalar constants $\alpha \in R$ and $\beta \in R$ are chosen by the specific time integration method. The other derivatives operators are used in basic forward and adjoint sensitivity computations as described in Section 6.2.2.

Figure 7 shows just some of the mathematical problems that the ModelEvaluator interface supports in addition to the transient problems being discussed here. There are concrete examples of all of these problems in production codes such as Charon and Aria/SIERRA using the ModelEvaluator [42]. In addition, the ModelEvaluator interface supports mixed problems types such as adding continuation to an optimization problem to make it easier to solve or adding uncertain parameters to a set of nonlinear equations [44]. Nearly all of these problem types will be presented in the NEAMS WF IPSC at some stage.

Nonlinear equations:	Solve $f(x) = 0$ for $x \in \mathbf{R}^n$
Stability analysis:	For $f(x,p) = 0$ find space $p \in \mathcal{P}$ such that $\frac{\partial f}{\partial x}$ is singular
Explicit ODEs:	Solve $\dot{x} = f(x,t) = 0, t \in [0,T]$, $x(0) = x_0$, for $x(t) \in \mathbf{R}^n, t \in [0,T]$
DAEs/Implicit ODEs:	Solve $f(\dot{x}(t), x(t), t) = 0, t \in [0,T]$, $x(0) = x_0$, $\dot{x}(0) = x'_0$ for $x(t) \in \mathbf{R}^n, t \in [0,T]$
Explicit ODE Forward Sensitivities:	Find $\frac{\partial x}{\partial p}(t)$ such that: $\dot{x} = f(x,p,t) = 0, t \in [0,T]$, $x(0) = x_0$, for $x(t) \in \mathbf{R}^n, t \in [0,T]$
DAE/Implicit ODE Forward Sensitivities:	Find $\frac{\partial x}{\partial p}(t)$ such that: $f(\dot{x}(t), x(t), p, t) = 0, t \in [0,T]$, $x(0) = x_0$, $\dot{x}(0) = x'_0$, for $x(t) \in \mathbf{R}^n, t \in [0,T]$
Unconstrained Optimization:	Find $p \in \mathbf{R}^m$ that minimizes $g(p)$
Constrained Optimization:	Find $x \in \mathbf{R}^n$ and $p \in \mathbf{R}^m$ that: minimizes $g(x,p)$ such that $f(x,p) = 0$
ODE Constrained Optimization:	Find $x(t) \in \mathbf{R}^n$ in $t \in [0,T]$ and $p \in \mathbf{R}^m$ that: minimizes $\int_0^T g(x(t),p)$ such that $\dot{x} = f(x(t),p,t) = 0$, on $t \in [0,T]$ where $x(0) = x_0$

Figure 7: A few examples of different mathematical problems that are supported by the ModelEvaluator interface.

An important feature of the ModelEvaluator design is that it provides a single interface between nonlinear solvers and application models as shown in Figure 8. The ModelEvaluator interface provides a single base class from which different subclasses for each application model can be derived. In a multi-physics setting, a different ModelEvaluator subclass may be developed for each set of physics if different codes are used to implement each of the models. Or, if a single code infrastructure is used to implement the models (such as with Aria/SIERRA), then a single subclass of the ModelEvaluator can be used and then

different single-physics ModelEvaluator objects can be constructed (one for each set of physics model) from this one subclass. This is the case in Aria/SIERRA.

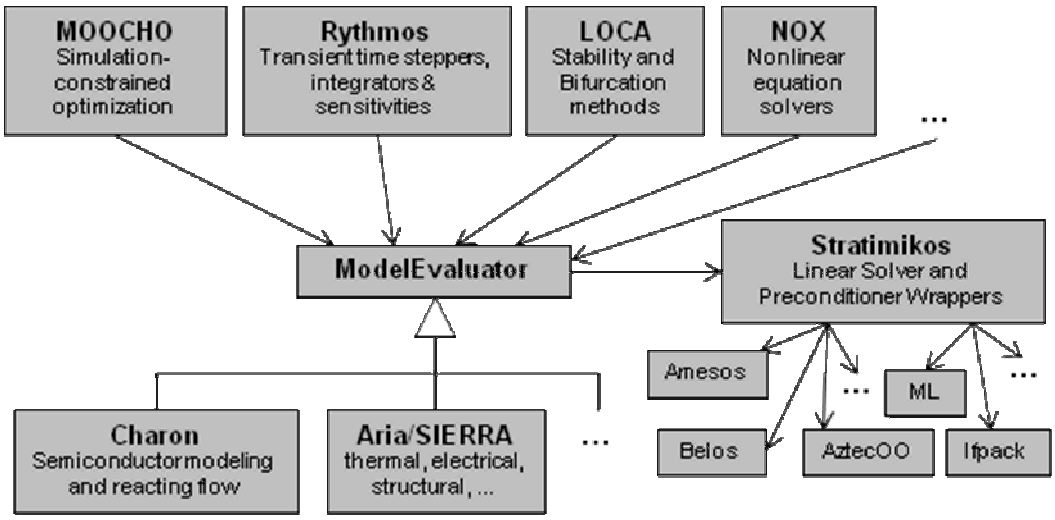

Figure 8: ModelEvaluator as an interface between solvers (MOOCHO, Rythmos, NOX, LOCA) and applications/models (Charon, Aria/SIERRA) as well as relationship to linear solver and preconditioning capabilities in Trilinos exposed through the Stratimikos package.

The ModelEvaluator design allows for a pay-as-you-go approach where an application can first start by exposing only a basic set of steady nonlinear equations $f(x) = 0$ with only the basic residual evaluation to get started. Then, the model can be incrementally expanded to add parameters p_l, response functions (used for sensitivity analysis and optimization), and various derivative operators as they are needed.

The concept of having a independent model object that is driven by external solvers is not unique to the Trilinos ModelEvaluator. One can find this basic concept in every major type of numerical library that supports nonlinear and transient solves (e.g. PETSc, SUNDIALS, etc.). The CCA has consider solvers and models but in terms of the CCA toolkit this does not appear to be much more than CCA wrappers for the concrete packages PETSc, CVODE (Sundials), and TOPS but there is little available to look at (see https://www.cca-forum.org/wiki/tiki-index.php?page=Toolkit). The ModelEvaluator approach in Trilinos and being advocated here is to raise the level of abstraction of a model and to generalize the types of inputs and outputs the model can support.

6.2.4.2 Multi-physics Solution Algorithm Toolkit

There are a variety of possible design approaches to constructing a solution algorithm infrastructure for multi-physics coupling. Here we describe a basic approach where independent general solver component can be combined with the different physics models to rapidly construct and evaluate different solution strategies. The composeable solver toolkit approach will support a variety of different schemes ranging from basic feed-forward coupling, through various operator-split approaches all the way up to fully implicit solution methods using specialized physics-based preconditioners. The specific solver components and interfaces will not be described here in any detail, but a candidate design in the Rythmos package that is part of Trilinos is mentioned and is described in more detail in Appendix C.3.2.

On one extreme, one can consider a basic feed-forward model coupling approach. Here, different time integration methods can be used for each physics model using a different solver algorithm component. In such an approach, only a limited amount of solution buffer space is needed to communicate solution data (through interpolation) from one model solution to the next. The same type of basic approach also applies to operator-split method that use time lagging and therefore do not require any type of subcycling. Alternatively, the individual solver components can be synced up to force the same time-steps, eliminating the need for flexible buffer space.

In the more general operator-split case, some form of subcycling between the model solutions within a single time-temp can be performed. Here, a different nonlinear time-step component (implicit or explicit) can be used for each different physics model and then a fixed-point iteration can be used to converge the coupled equations to solve degree. In this type of approach, the solver components for each individual physics model would only have to communicate solution data and accept coupling parameters from the other models as input. In this type of approach, the coupling equations or transfers would also need to be incorporated in the subcycle workflow.

At the other extreme would be a fully implicit method where all of the model equations are presented as a single abstract model as shown in (6.3). In this case, a single solver component would be used to drive the time integration method. Here, a specialized physics-based nonlinear solver component for the coupled time-step equations could be developed and used. Or, a generic Newton-type nonlinear solver component could be used and instead a specialized physics-based preconditioner component could be developed and used inside of the standard iterative linear solver component.

Hopefully the above examples have given a sense of the type of flexible composable solver component toolkit that the NEAMS THCM Framework is going to seek to develop. As mentioned earlier, the Trilinos collection has adopted this solver toolkit approach as described in [42] and the Rythmos design, based on the foundation of solver objects in Trilinos, discussed in Appendix C.3.2 is being designed to natively support all of the approaches discussed above but concrete examples of all of these do not yet exit. As mentioned before, other efforts (e.g. CCA, PETSc, SUNDIALS) have seen the value of this basic approach and have incorporated some of the needed aspects into their software. It is not clear what aspects of these various existing software collections will be adopted by the NEAMS WF THCM Framework or what level of software development would be needed in order to incorporate their use.

The last issue to consider with respect to basic simulation and multi-physics coupling is the middle technology to hook together the various simulation objects. The ModelEvaluator and Trilinos approach assumes basic computer language coupling with basic object-oriented C++ being the primary object interoperability approach. This may be wholly sufficient for the core multi-physics coupling and solution framework for the THCM framework software. However, there may also be some advantages to considering adopting a component architecture to facilitate the coupling of the different model and software components. Two such component architectures are the Common Component Architecture (CCA) and Salome (See Appendix C).

6.3 Workflow Framework

The THCM Workflow Framework manages sequences of complex analysis calculations performed with integrated high-fidelity THCM simulation codes and surrogate THCM performance assessment codes. For high consequence calculations, there is a need to manage many different elements.

- Version control of source code and reproducibility of executable code.
- Retention and retrieval of minimal input and output data associated with specific studies needed to reproduce computations.
- Specification and reproducibility of the exact sequence of computations, i.e. the workflow.

In this section, issues related to the specification of the workflow specification and management of the key input and output files is described. Version control of source code and reproducibility of executable code is addressed in Sections Configuration Management7.3 and 7.4.

6.3.1 Currently Identified Requirements

The following THCM analysis workflow framework requirements are defined to meet verification and validation objectives and support requirements identified in the AFCI QAPD [2]. In addition, the analysis workflow framework must be accessible, interoperable, and capable of handling large amounts of generated data. The following requirements describe the needs of the analysis workflow framework, not any single simulation code or tool. They may imply requirements on specific tools.

Accessible (RQ1): This effort spans many laboratories across the current DOE complex; as such, any supplied services must be accessible from all other sites. This implies acceptable performance on this scale as well. Furthermore, the program is likely to expand to other currently unknown physical locations over time.

Interoperable (RQ2): Just as services need to be accessible across multiple sites, exposed services must be able to interoperate. Without this kind of interoperation, new services will be unable to extend previously offered services leading to unnecessarily limited framework growth.

Recreateable (RQ3): The deployed framework must be able to support extensive result review. To support this, the results of any simulation must be easily obtainable. Furthermore, we need to be able to effectively recreate any given step of an analysis workflow. This requires that we maintain all data used as input and the outputs of any simulation work and periodically test the recreateability of workflow steps. We also need to maintain tools used to create input data, including commercial tools and simulation codes. This extends into subcontinuum, continuum, and surrogate work.

Accurate (RQ4): Results need to be able to be trusted, and must be known to be as accurate as possible within acceptable engineering tolerances. This requirement implies the ability to arbitrarily and easily verify and validate final and intermediate results.

Verifiable (RQ5): In order to have confidence in generated results, stakeholders must be able to review, verify, and validate any generated product at any given time. Possibilities include checking output from the computation of a single supercomputer node given the original input, checking a material property set against experimental results or formal reviews of results as per AFCI QAPD [2]. The specific form of a review, verification, or validation activity is arbitrary, but any and all steps in a given run must be able to supply data and tools as needed.

Secure (RQ6): The system needs to be designed with security in mind at the outset. Furthermore, the security policies will need to support different levels of rigor depending on where implemented. For example, individuals and groups from different laboratories must be authorized to use and be able to be

authenticated to access cross-laboratory resources. Furthermore, system events that may indicate operational failures or intrusion must be securely logged and handled appropriately depending on event severity.

Reporting (RQ7): Personnel must be able to generate reports of arbitrary activities. This would include reports over system or security logs, use of supercomputing resources, or the use of a specific material property set. The initial group of reports is currently undefined.

Data Magnitude (RQ8): We currently generate large volumes of data during supercomputing simulations, both at intermediate steps of a run and the beginning and end of a run. The workflow framework must support the ability to retain and version these files, reliably, for long periods of time (on the order of 10 years).

Data Fragmentation (RQ9): When running a single simulation, work is generally partitioned over some number of nodes. Those nodes then generate output data from node-specific input data. This data needs to be versioned and maintained.

Traceable (RQ10): The workflow framework itself as well as work product must support backward and forward traceability to enable analysts to understand exactly how a given system performed and why certain results may have been generated. This requirement is implied by the need for extensive verification. Furthermore, all associations will need to be bidirectional and must be able to support the propagation of uncertainty information.

6.3.2 General Workflow

Analysis of workflow requirements is based upon the generalized conceptual model of a workflow presented in Figure 9. In this figure the components in purple are optional and the "component computation" cycle can execute an arbitrary number of times.

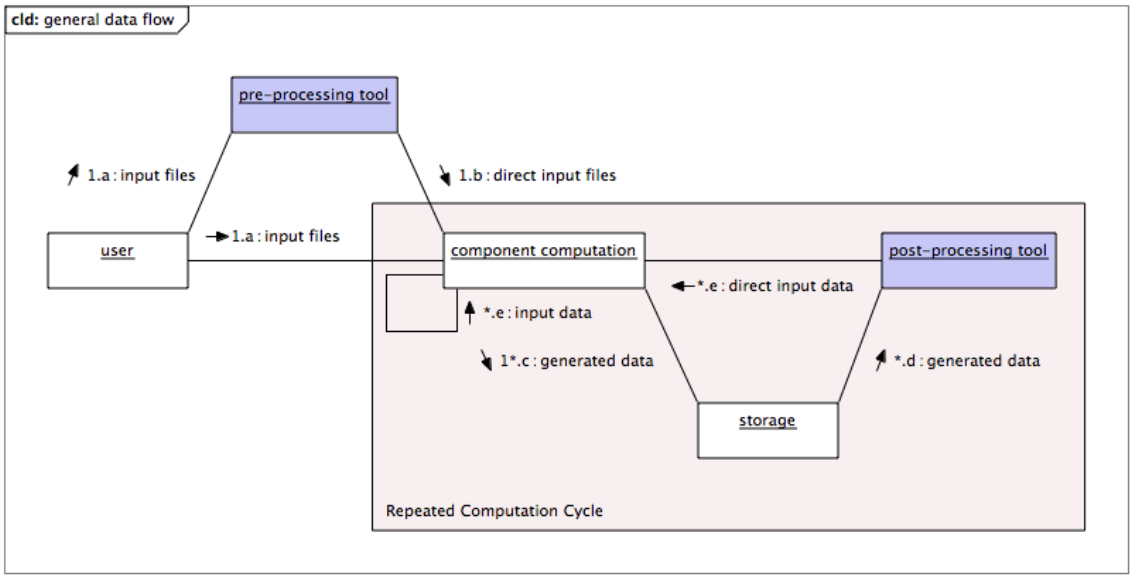

Figure 9 Generalized conceptual model for workflow including user interaction, pre-processing, component computations, storage, and post-processing

In this conceptual model a user provides basic input files for a component computation. These input files may be optionally processed by some tool prior to use or submitted directly to the computation engine

(steps denoted by arrows 1.a and 1.b). The component calculation runs, writing produced data to output files or other data storage. An optional postprocessing tool can then extract data from storage to pass to downstream computations in an iterative process (steps 1.c through 1.e). Note that overall the executable code, scripts, and input data from step 1.a, and the intermediate data files from the component computation cycle (steps *.e), must be archived.

6.3.3 Requirements Mapped into a Use-context

Requirements are mapped into the context of expected system use to show where specific services are needed within an expected continuum analysis or performance assessment workflow. This context is illustrated with the high-level collaboration diagram (Figure 10) between the components of a workflow framework. This diagram is presented to highlight architectural concerns of the workflow framework. Note that a user or group of users is typically involved in the analysis workflow, although they are not explicitly shown in this diagram.

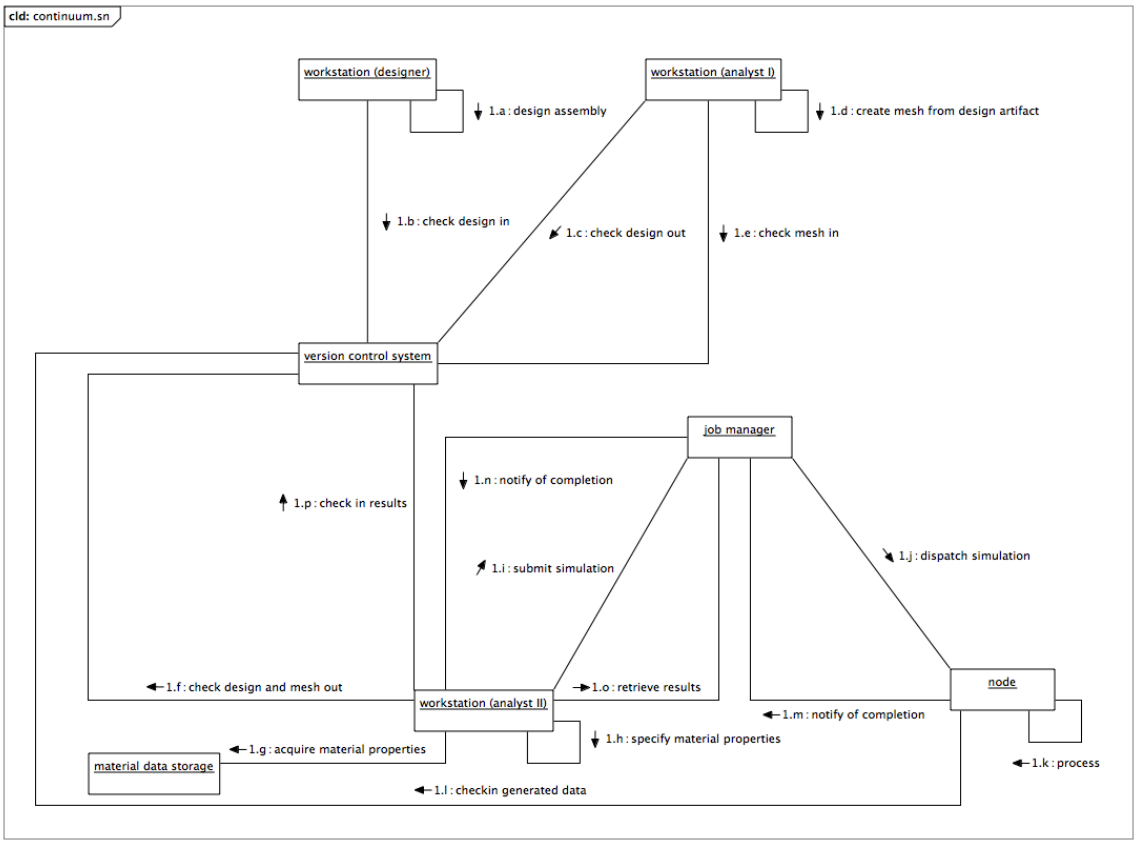

Figure 10 A high-level collaboration diagram between workflow framework components supporting an analysis workflow

This collaboration diagram does not include potential relationships between a version control system and the models database (materials data storage). Requirements for version control of materials data sets and other parameters are addressed in Section 6.1, and are not included as step in this analysis workflow. Three distinct workstation components are identified in the collaboration diagram. This is a logical (not necessarily physical) distinction made to group workflow steps performed through a user's workstation. These workflow steps could be performed through a single workstation, or through multiple workstations by different users.

Workflow framework requirements given in Section 6.3.1 are mapped to the analysis workflow steps in Figure 10 in the following table.

Table 9 Workflow framework requirements mapped to steps in an analysis workflow

Workflow Step		Mapped Requirements
1.a	design model assembly of waste form and environment	Secure (RQ6), Recreateable (RQ3), Verifiable (RQ5)
1.b	check in designed model assembly	Accessible (RQ1), Interoperable (RQ2), Recreateable (RQ3), Verifiable (RQ5), Secure (RQ6), Reporting (RQ7), Traceable (RQ10)
1.c	check out designed model assembly	Accessible (RQ1), Interoperable (RQ2), Recreateable (RQ3), Verifiable (RQ5), Secure (RQ6), Reporting (RQ7), Traceable (RQ10)
1.d	create mesh from design artifact	Secure (RQ6), Recreateable (RQ3), Verifiable (RQ5)
1.e	check in meshed model assembly	Accessible (RQ1), Interoperable (RQ2), Recreateable (RQ3), Verifiable (RQ5), Secure (RQ6), Reporting (RQ7), Traceable (RQ10)
1.f	check out designed and meshed model assembly	Accessible (RQ1), Interoperable (RQ2), Recreateable (RQ3), Verifiable (RQ5), Secure (RQ6), Reporting (RQ7), Traceable (RQ10)
1.g	acquire material properties and other parameters	Accessible (RQ1), Interoperable (RQ2), Recreateable (RQ3), Accurate (RQ4), Verifiable (RQ5), Secure (RQ6), Reporting (RQ7), Traceable (RQ10)
1.h	specify material properties and other parameters for selected physics models	Secure (RQ6), Recreateable (RQ3), Verifiable (RQ5)
1.i	submit simulation to job manager	Accessible (RQ1), Interoperable (RQ2), Recreateable (RQ3), Accurate (RQ4), Verifiable (RQ5), Secure (RQ6), Reporting (RQ7), Traceable (RQ10)
1.j-k	job manager dispatches simulation to be processed on computational node(s)	Secure (RQ6), Recreateable (RQ3), Verifiable (RQ5)
1.m-n	notify of simulation completion	Secure (RQ6)
1.o	retrieve simulation results	Accessible (RQ1), Interoperable (RQ2), Recreateable (RQ3), Accurate (RQ4), Verifiable (RQ5), Secure (RQ6), Reporting (RQ7), Traceable (RQ10)
1.p	check in simulation results	Accessible (RQ1), Interoperable (RQ2), Recreateable (RQ3), Accurate (RQ4), Verifiable (RQ5), Secure (RQ6), Reporting (RQ7), Traceable (RQ10)

Performing an Analysis

An end-to-end analysis workflow spans all identified steps from designing the model to checking in analyzed simulation results. An analysis activity may span only a subset of these steps. For example, many simulations may be run (Steps 1.f through 1.o) with different parameters for the same designed and meshed model assembly. Such an analysis activity should begin with data that is checked out from the version controlled repository so that the analysis activity is recreatable.

Rerunning an Analysis

A previous analysis activity may be recreated and rerun with a modified model, parameters, or other inputs. For example, subcontinuum analysis may revise a material property or a software defect is found that may impact analysis results. Rerunning an analysis consists of retrieving previous, unmodified inputs, combining these with revised inputs, and repeating the analysis workflow. In this scenario the rerun analysis produces a new version of results which is traceable to the revised inputs.

Examining Simulation State

If a code is altered or a material property set changed, an analyst may compare the results of a previous simulation with a new simulation using the new data or code. As all results from the previous simulation have been saved, all the analyst need to is run the new simulation and then compare the results from both simulation runs, noting and analyzing any changes.

6.3.4 Anticipated Architectural Components

A preliminary analysis of the modeled workflows and mapped requirement suggest the architectural components identified in Figure 11. These include the model database (Section 6.1) for material properties and other parameters, simulation services including physics models and multi-physics coupling framework, version controlled repository for waste form simulation models and results, workflow orchastraction services, pre- and post-processing capabilities, job management services, and other ubiquitous / foundational services.

Pre and post-processing Services: Various common pre-processing services include development an initial model, meshing, and graphical user interfaces (GUI) for creating and editing complex files. Common examples of post-processing services traditionally include visualization, analysis, and result reporting.

Simulation Services: The physics models and multi-physics coupling framework required to run a given simulation.

Workflow Services: A system that supports automation of simulations and simulation recreations to increase the reliability and repeatability of a given run and supporting work. Ideally, it would allow for control of all tools used in a given analysis workflow. This implies the need for such workflow services to be both distributed and operating system agnostic.

Data (configuration) Management Services: Version controlled data management services are required to manage waste form model design assemblies and meshes, simulation inputs and results, and traceability among these artifacts. These services need to be accessible from workstations and supercomputing nodes. They should also be available program-wide to enable collaboration. If this is in fact used between analysts to share work product however, the ability to quickly examine files for changes, to branch and merge, and to tag specific versions of work items will also be needed.

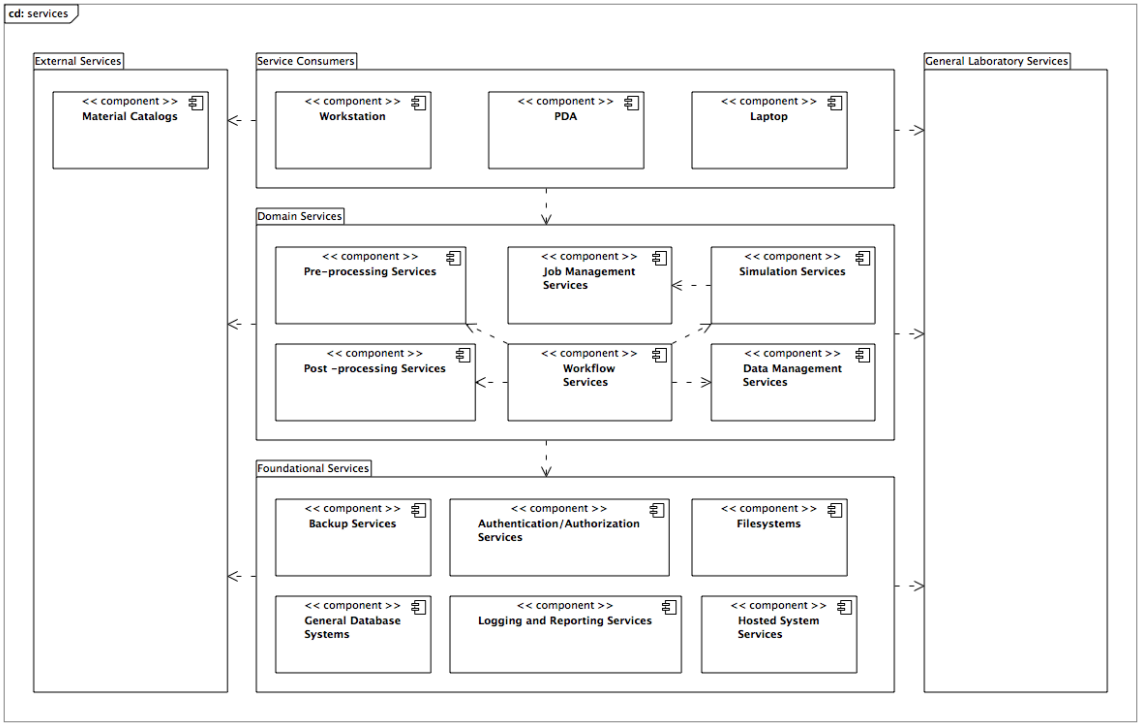

Figure 11 Anticipated architectural components of the workflow framework

Job Management Services: Most supercomputing facilities have some kind of master node that users access to submit work across all needed nodes. Program personnel program wide would need to have access to any of these kinds of nodes controlling resources the program may use.

Backup Services: Data needs to be retained for an extended period of time to support validation and recreation scenarios. Ideally, these data sets would be maintained in a way that's easy to use and interoperates across the various laboratories. This should also be part of a larger disaster recovery plan.

Authentication and Authorization: Access to shared resources should be limited to approved personnel only. Traditional role-based authentication and authorization should suffice. This system would need to interoperate across the program however, and would need to be robust enough to withstand attack from outside (or inside) entities.

Filesystems: Local and distributed filesystem services.

General Database Systems: General database systems including relational databases and possibly non-traditional data storage approaches (e.g. Hadoop or BigTable).

Logging and Reporting Services: In order to appropriately manage this kind of a system, operators and management need to be able to access reports covering metrics of interest regularly and easily. These reports could include information covering anything from user access to simulation running time. These reports should be easy to create, use, edit, save, remove, and print, and may include notifications to subscribed users of specific events. Logging services could include everything from protected security event logging to billing information retention.

Hosted System Services: General systems administration of supercomputing, virtualized, or desktop computers.

General laboratory services: Any kind of general services a laboratory location may provide. This can include anything ranging from accounting verification of job numbers to high-performance backup, storange, and retrieval to product data management (PDM) software packages. Domain and foundational services may extensively use these services if appropriate.

6.3.5 Analysis Workflow Services

Analysis workflow services support automation and recreation of workflow steps such as checkpointing, data archiving, associating data artifacts, etc. These services are closely tied to the domain of engineering simulation and analysis. Requirements which must be statisfied by these services include recreateability, traceability, verifiability, and accessibility. Several commercial, open source, and other packages may be able to provide analysis workflow services.

Option 1: Commercial Packages

Companies providing similar functionality at differing scales include MSC Software, Dassault Systems and Phoenix Integration via SimManager, ENOVIA, and PHX ModelCenter products, respectively.

- SimManager is part of MSC Software's suite of simulation and simulation management products. Essentially intended to be a collaborative environment for distributed simulation and development teams, SimManager manages access to all related MSC Software simulation tools as well as tools that have been integrated into MSC's simulation framework.

- ENOVIA, formerly known as Matrix One, is primarily a product lifecycle management system rather than a simulation management system. ENOVIA Portfolio though provides functionality similar to that accessible via SimManager.

- PHX ModelManager is specifically targeted to creating engineering workflows. It comes with prepackaged adapters to allow users to create workflows with commonly used engineering codes. It also allows users to create their own adapters for arbitrary tools with command line interoperability. These workflows are then saved in files that can then be submitted to a revision control system or other data storage system.

- With respect to SimManager and ENOVIA, neither one will provide all functionality required in a technical workflow system, but they may provide some key components. Both products are part of a large, integrated suite of systems, and claim to be extensible enough to integrate into virtually any enterprise. Realistically, though they likely can be integrated, it may take significant effort and the interfaces used may not be rigorously controlled. Also, as they are both large companies, we can expect them both to be stable partners, though they may not be optimally responsive to functionality requests. This would lead to partial lock-in to the vendors and their development cycles.

- PHX ModelManager is a purpose-built engineering workflow tool, but is backed by a smaller company, Phoenix Integration. That said, they do have an extensive client list and have been in business since 1995.

Option 2: Open Source Projects

Three open source projects that attempt to address these issues with different levels of focus are the Salome Platform, the Kepler Project, and the Common Component Architecture. All three projects seem stable currently.

- Salome is a project dedicated to providing generic pre and post processing services for simulation work. Physically headquartered in France, it provides a component model and well as various

semantic models, and can manipulate data and tools remotely via CORBA. Salome is currently distributed under the LGPL license.

- The Kepler project distributes Kepler under the BSD license Kepler is specifically targeted to creating executable workflows composed of webservices, C, R, Matlab, or other general tools accessible via a command line. Kepler supports various grid computational technologies natively and has parallel processing support built into the product.

- Finally, the Common Component Architecture project is more of a component standard for scientific computation rather than a workflow system. It does however provide for remote service execution and standard component architectures that can be extended into a scripted workflow.

Each package will require varying levels of customization to apply in the NEAMS environment. For example, Salome and Kepler will both require extensions to handle codes developed specifically at DOE facilities, while the Common Component Architecture needs to be integrated with some kind of scripting system in order to represent a workflow. Of the three products, Salome has been used in nuclear reactor research, and so is likely to be the best initial fit. Kepler, on the other hand, has been used specifically for scientific workflow management and seems more tailored for workflow maintenance than Salome, but would need more tailoring to fit into the expected computational environment. The common component architecture would likely be the most expensive as it is more of a standard than a usable tool currently.

With either project we would need to incorporate any of our changes into the original project in order to keep future integration costs as low as possible. Caution is the order of the day in this regard; code that shows areas of interest may be sensitive.

Finally, licensing may be an issue as well forcing us to expose code we would like to keep proprietary or limiting the distribution of changes.

Option 3: ASC Packages

DART has extended what is now ENOVIA to support simulation workflow in the nuclear weapons community. The current workbench supplies revision control, job submission, and data archival, and the ability to define and execute workflow via Ant. This work could potentially be extended into the NEAMS domain as appropriate as it is a close match to currently known to be required functionality. Currently however, ENOVIA is in the process of being supplanted by Windchill PDMLink at Sandia National Laboratories.

Option 4: Develop

The final option is to develop a new system to support analysis workflow. In order to implement a new system to support our known workflow requirements, we would extend existing infrastructure and software as much as possible to constrain costs and effort. Possible viable solutions could include extending Salome or other current open source workflow solutions and integrating with current infrastructure like PDMLink or Filenet.

In general, product data management systems like PDMLink automatically supply the ability to establish and manage relationships between tracked artifacts. Traditional software version control systems, on the other hand, are generally either file or revision based, and do not track associations in such a way without extension. In our current domain, PDM systems are a more natural fit than source control systems, but both should be rigorously evaluated for suitability.

6.3.6 Data Management Services

In order to support analysis workflow a data management system is required that can handle extremely large files, provide a variety of traceability over those files, and operate over a variety of domains from handling model design and mesh files to archiving interim simulation input files.

For ease of initial analysis, we can divide our data requirements into two areas, materials data and simulation data, both of which can be handled via commercial off-the shelf packages and potentially integrated with current open source software if needed.

Option 1: Commercial Packages

Commercial PDM systems supply much of the needed archival functions required. With respect to managing materials information, the current market leader is Granta Design with their Granta MI product. This product supplies traceability between experimental data and generated material property sets, robust data management, and custom authentication. They also have a variety of programming interfaces that can be used to extend the product, and a solid reputation for responding to customer demands. At Sandia, Windchill PDMLink has been rolled out to the engineering community. Phoenix International also provides similar services that can integrate with their PHX ModelManager product.

Option 2: Open Source Packages

Currently, no open source solutions exist for materials management, nor are any open source PDM systems available. That said, a variety of open source revision control systems are available, but they would require extension in order to handle associations correctly.

Option 3: ASC Packages

DART has produced both a materials management system as well as a data management and archival system. The data management system is now integrated into the DART workbench as was covered in the previous section. The materials management system is in the process of being phased out and has proved to be too expensive to maintain when compared to similar commercial offerings.

Option 4: Develop

Again, as with analysis workflow, any data management services development efforts should be constrained as much as possible via integrating with current commercial and open source packages.

6.3.7 Example of a Potential Technical Architecture

Analysis workflow has been analyzed from an application-centric perspective, where functional elements have not been mapped to particular products, technologies, or protocols. The example architecture presented in Figure 12 does not promote the adoption of any particular products or technology; rather, it's an example of one possible heterogeneous environment which could be pursued.

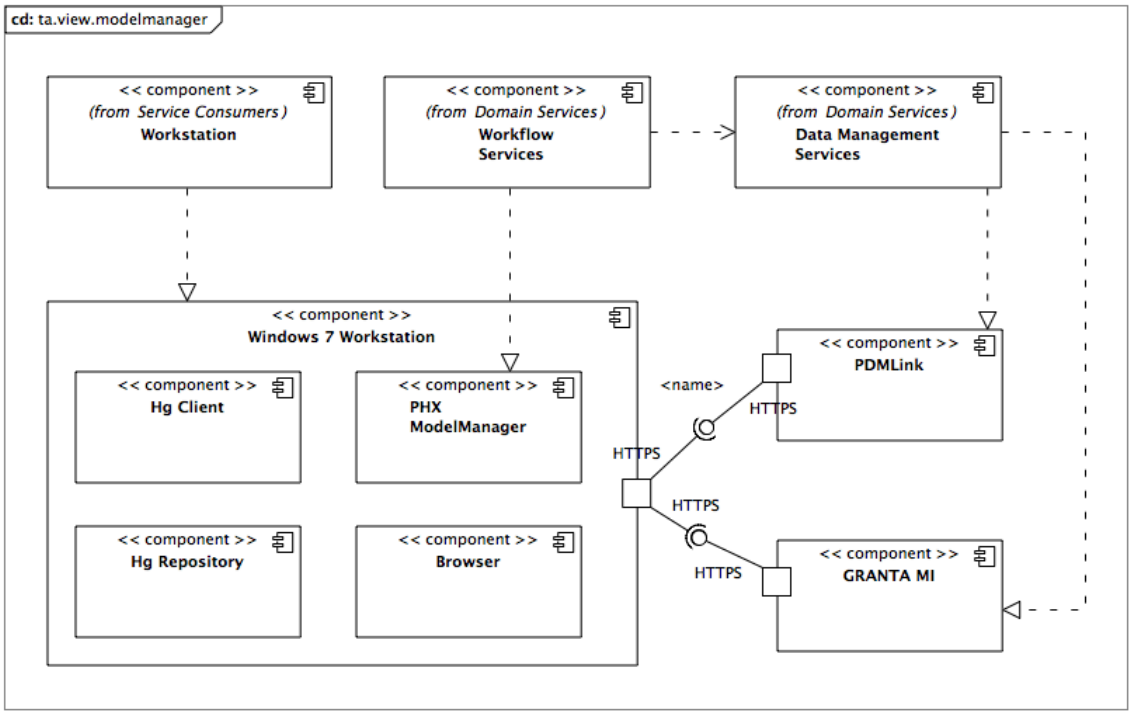

Figure 12 Example architecture to illustrate potential utilization of existing analysis workflow and data management services

This example architecture includes the following components.

- *PDMLink:* PDMLink is used as an archival tool at a fairly coarse level of granularity. Specifically, users wouldn't depend on PDMLink for revision and change control when collaborating, but rather would use it to check in repositories that other users can check out to merge changes or update with new work.
- *Granta MI:* Granta MI is a commercial, off-the-shelf materials management product and is also a realization of a data management service.
- *Mercurial (Hg):* Mercurial is a distributed revision control system. Similar to Git, it supports merging and branching of repositories on the repository level. The master repository to which everyone collaborating would merge and from which they would update is stored within PDMLink.
- *PHX ModelManager*: PHX ModelManager is a commercial tool for creating engineering workflows and a realization of a workflow service.
- The primary network protocol in this example is HTTPS used via a web browser. Other protocols and access tools could certainly be used, but HTTPS is ubiquitous and reasonably secure.

In this example, requirements for revision control at the collaboration level are more stringent than they would be for other areas of interest. For example, potential requirements to create different branches of a given workflow and to track all revisions and changes to a given workflow file are assumed. To support this, the data management repository is implemented with the PDM system. When a user needs access to the repository, that user checks it out from the PDM system and creates a repository clone. After checking the master repository back in, that user can check any and all changes into that local repository,

create repository branches, rollback changes, and so on. Here, that user would be required to include extensive comments with any repository changes to maintain a semantic context other engineers can review later if needed. Finally, when that user hits a point where the changes should be checked into the master repository, the user checks the master out again, merges into the master, and checks the master back into the PDM system. Those changes are now available to the rest of the team.

Walk-through of a collaboration use case

Assume two users, user A and user B, are collaborating on project team X. User B has been working with project team X for six months, but user A is new and is working on completing a new simulation workflow. Both user A and user B are using workstations with the same toolset installed.

User A begins by directing a browser to PDMLink and logs in. Then, having navigated to the project repository, he then checks out the current Mercurial master repository. Then, he immediately clones the repository and checks it back in.

Now that user A has a copy of the repository, he opens PHX ModelManager and then opens the version controlled workflow description file. Over the next few days, he tweaks that workflow, checking all changes into his local repository copy. At this point, user A decides he's finished and needs user B to review his work. While building the workflow, user A logs into the Granta MI materials repository to ensure that the correct materials property sets have been checked in by the experimental team and that both the property sets themselves and the experimental data is available. User A then checks the master repository out, merges his changes into that master repository, and checks it back in.

User A notifies user B by email of the repository being checked back into PDMLink. User B then checks out the master repository and updates his local copy with any changes checked in by user A, immediately checking the master repository in when finished. User B looks over the current state of the workflow within PHX ModelManager, but finds something he doesn't understand. In order to clarify the final result, he goes back to his local copy of the repository and examines the revision history that was updated to his local copy from the master repository. The revision history is extensive, and due to the detailed comments, he now understands more clearly why user A made the changes he did.

The review finished, user B cleans the file up, adds some comments, and checks the changes into his local copy. He then checks out the master copy and merges his local repository into the master, checking the master back in when completed.

7. Software Engineering Environment

High-level requirements and plans for the software engineering environment are developed based upon SNL's rigorous experience implementing SQE within numerous software development projects, especially those within the Advanced Simulation and Computing (ASC) program. It is expected that eventually some WF IPSC will be required to satisfy Quality Rigor Level 1 Requirements defined in the Advanced Fuel Cycle Initiative (AFCI) Quality Assurance Program Document (QAPD) [1]. As such software engineering environment practices and tools are planned to enable software development at this quality rigor level. However, development at lower levels of rigor will also be supported.

Many SQE best practices have evolved for efficient development of high quality software; i.e., software that provides the expected functionality; is reliable, usable, efficient, and portable; and is maintainable and flexible when incorporating new requirements. These software quality characteristics are desirable regardless of the required Quality Rigor Level. As such the software engineering environment practices and tools will emphasize efficient development of quality software (functional, reliable, usable, efficient, portable, maintainable, and flexible) as well as enabling support for AFCI QAPD requirements.

Software quality engineering considers the total cost from requirements gathering through support of delivered software. This includes man-hours expended as well as the indirect cost of poor-quality software; e.g., consequences of erroneous results, unreliable performance, and numerous debugging and patching efforts. It is "common knowledge" in the SQE community that inadequate investment in understanding software requirements, evaluating software design choices, and testing software implementation can lead to out-of-control indirect costs.

Strong software engineering (SE) must be at the foundation of any complex software intensive endeavor. There are a number of unique challenges in producing high credibility software in a computational science & engineering environment (CS&E). The issues described here will seek to provide the highest quality environment for creating the advanced simulation software, will make the work more productive, and will provide high quality software from the very beginning. Principles from the modern Lean/Agile SE community will be adapted to suit our unique CS&E environment.

One of the principles from the modern Lean/Agile movement is that a software development team will always use appropriate high quality processes and practices to develop and maintain the code at a high level of quality from the instant that it is created [13,19,23,16,21,24,25,14]. The Fundamental Principle of Software Quality [13] is that: upfront investment in developing software with a high quality is more than paid for on the backend by avoiding expensive debugging and code reworking. Therefore, investing in quality SE processes and procedures is, in total, free for deployed and supported software.

In some CS&E projects the software is not deployed or supported, and only needs to provide a few isolated calculations to produce "first of the kind" and "demonstrative" results. The quality control of the research CS&E journal peer-review process is such that these types of low-credibility and incorrect calculations can be routinely published [52]. It has been argued that the CS&E community in general needs to mature if the results from CS&E simulations will become a routinely trusted tool in important decision making [51]. Software quality can be perceived as too expensive in a competitive research-driven environment where success is measured by the number of publications and not by the credibility and reproducibility of the results. In an environment where requirements for software correctness is low, the General Principle of Software Quality (i.e. that quality is free in the end) may not apply.

The purpose of this section is to highlight some of the SQE issues that will be important for the NEAMS WF IPCS project and discuss SE issues specific to our CS&E environment not routinely discussed in the larger literature.

7.1 Project Management

Project management is the systematic management of the projects' mission, organization, resources, requirements, tasks, schedules, risks, and practices or processes. A project's processes are defined to accommodate volatility in the mission, requirements, organization, and resources. Processes may be revised to improve software quality or development efficiency.

7.1.1 Stakeholder Relationships

Management of stakeholder relationships includes identification of stakeholders and their roles and responsibilities, and intentional communication with those stakeholders. Stakeholders include members of the THCM application and framework development team, THCM application and framework end-users, DOE funding organizations, peer AFCI campaigns/projects/teams, and suppliers of hardware and software to this project. Intentional communication with stakeholders is necessary to manage expectations, constraints, and collaborations. The scope of content, level of formality, and means of communication must be clear among stakeholders.

Stakeholder expectations will have priority depending upon the role and authority of the stakeholder. An applicable regulatory expectation is a non-negotiable constraint that must be satisfied. Programmatic expectations are negotiable with respect to the classical SQE-constraint of capability, cost, and time. Peer-project collaboration expectations could be subject to mutually beneficial negotiation.

It is anticipated that stakeholder priorities and expectations will change or be clarified. Such changes introduce a significant risk in a project's ability to satisfy customer expectations. Regular dialogue with stakeholders is essential to be responsive to these changes or clarifications. Note that the Agile software development methodology mitigates this systemic risk by including frequent dialogue with stakeholders into the software development process.

Given the scope and anticipated funding for the NEAMS WF IPSC, it is expected that the project team will include numerous engineers and scientists who participate with a fraction of their overall time. In this organizational environment intra-team communication becomes a risk to the success of the project. A mitigation strategy which has already been implemented is to support intra-team communication with the http://neams.sandia.gov collaboration website. This collaboration website provides project information, programmatic and stakeholder information, and task tracking.

7.1.2 Lifecycle Processes

Software development lifecycle processes are an integral part of the software engineering environment. The goal of these lifecycle processes is to promote efficient development, maintenance, and support of high quality software which meets the projects' mission and requirements and to deliver increasing levels of value at regular intervals. Additional lifecycle process requirements may exist to satisfy stakeholder constraints such as the AFCI QAPD [1] and SNL Corporate Process Requirements.

Software engineering tools are deployed within an organization to support implementation of lifecycle processes. The quality of these tools and their suitability to the organization will significantly impact the effectiveness of these lifecycle processes. As such lessons learned from implementations of the Advanced Simulation and Computing (ASC) Software Quality Engineering Practices [3] at Sandia National Laboratories will be leveraged when defining WF IPSC lifecycle processes and selecting software engineering tools.

7.1.3 SQE Metrics and Improvement

Sustaining and improving both software quality and lifecycle process effectiveness requires objective measurement and evaluation. Definition, collection, and analysis of appropriate metrics helps direct project resources to address software quality or lifecycle process problems, or to take advantage of opportunities for improvement. However, the use of inappropriate metrics can lead to the diversion of valuable project resources without improving either software quality or lifecycle process effectiveness.

Metrics which directly measure software quality are highly likely to be appropriate. For example, the number of code bugs discovered by users and the amount of code covered by successfully passing unit tests are potential direct measures of software quality. Metrics for lifecycle process effectiveness are more challenging in that they attempt to measure software quality versus the cumulative cost of resources allocated to that software. For example, counting the number of team hours dedicated to software design reviews is likely be correlated to software quality; however, variables such as the knowledge, skills, and dedication of the team members involved in such a design review can significantly impact the affectivity of such a metric.

7.1.4 Project Planning, Risk Management, and Project Oversight

A project plan documents the scope, assumptions and constraints, roles and responsibilities, inter-dependencies with external projects, expected budget and resource allocations, identified risks and mitigation plans, as well as task plans. Project planning and risk analysis includes developing and evolving the content of the project plan to an appropriate level of detail and formality. Project oversight and risk monitoring includes measuring actual project performance against the project plan, analyzing significant performance deviations or risk events, and implementing corrective or risk mitigation actions.

This NEAMS WF IPSC report documents the project scope, strategic requirements, use cases, verification and validation strategy, THCM framework architecture, and software engineering environment. The http://neams.sandia.gov project website includes other components of a project plan including identification of roles, responsibilities, anticipated inter-dependencies with other projects, and tasks.

7.2 Requirements and Design

Sources of NEAMS WF IPSC requirements currently include programmatic stakeholders and the WF IPSC teams' experience with similar programs (e.g., YMP, WIPP, and ASC). This report represents a preliminary baseline of requirements and THCM framework architectural design. The requirements will grow, change, and be re-prioritized throughout the lifespan of the program to meet the stakeholders' evolving needs. The design will similarly evolve as requirements change and software quality improvements are introduced.

Requirements are categorized into long term strategic requirements and short term tactical requirements. Strategic requirements are assumed to be stable, guide long-term project planning, and drive the system / software architecture. Tactical requirements and their priorities are assumed to change more frequently, guide short term tasks, and may perturb existing software design. Tactical requirements should be correlated to strategic requirements and the system / software architecture.

In addition to basic requirements, some discussion of reasonable strategies for driving short-term and long-term activities is also discussed in this document. These discussions of probable approaches, or any of the more concrete aspects of this document, should not be seen as set in stone but instead should be seen as best guesses for where to start.

7.2.1 Requirements Management System

A Requirements Management (RM) system shall be deployed to organize, maintain, and track accumulating and changing requirements including the PIRT. Organization of requirements shall include categorization, project-specific attributes, and traceability between derived or interrelated requirements. It is expected that multiple project members will update requirements; therefore, the RM capability shall provide multi-user configuration management. A candidate commercial product that satisfies these RM configuration management requirements is IBM® Rational® DOORS®.

7.2.2 Strategic Requirements and Software Architecture Development

Strategic requirements reflect the mission and scope of a project, identify stakeholders, provide a foundation for project planning, and drive software architecture. Intentional development of software architecture is essential for software quality. "Brittle monolithic systems, silo applications, and long and unpredictable development times, are symptomatic of architectural decay which causes huge organizational drag. To break the chains of our corporate legacy and build systems that fit the environment, and adapt with the environment as it changes, we need architecture." [12]

A software architecture partitions a large complex system into smaller, more manageable, components with well-defined roles, responsibilities, and interfaces. Strategic requirements and software architecture are derived from analysis of the problem domain and are inextricably bound together in the stakeholders' conceptual model(s) of the problem domain. When stakeholders have different conceptual models of the problem domain then they will have different interpretations of requirements, and thus their expectations will be in conflict. The modern domain driven design [15] approach to integrated requirements and design development emphasizes the need for stakeholders to develop and maintain a single shared conceptual model of the problem domain (referred to as a *domain model*).

The domain model for a large complex system has hierarchical partitioning of the problem domain, with the outermost components reflecting the architectural view and inner components reflecting successively detailed views of the software design. This system specification document includes the current architectural-level of the NEAMS WF IPSC domain model. Each stakeholder's need to understand interfaces and internal details of a particular architectural component will vary with that stakeholder's roles and responsibilities. However, all stakeholders need to understand the context of their components within the hierarchy.

The architectural view of a domain model will change as the project scope, strategic requirements, or stakeholders' understanding of the problem domain evolves. Note that in a research setting, the stakeholders' understanding of the problem domain may rapidly evolve even if the scope or strategic requirements remain unchanged. When changes occur it is essential for impacted stakeholders to be involved so that a single shared understanding of the domain model is maintained. It is also essential that changes to the domain model be propagated into software implementations and documentation to avoid misunderstandings and resulting breakdowns in software interfaces.

7.2.3 Tactical Requirements and Software Design Development

Tactical requirements and software design are differentiated from strategic requirements and software architecture by their scope and rate of change. It is expected that rate at which strategic requirements and software architecture change decreases as the software matures; however, it is also expected that such changes can and will occur as long as the project continues active development. Tactical requirements and software design changes are those which do not impact the architectural-level of the domain model, and are expected to occur frequently over the lifetime of the project.

Effective response to frequent tactical requirements and software design changes has both software quality and development process considerations. The development process must enable agile response to changing requirements and priorities, without sacrificing software quality. Similarly, the THCM framework and analysis codes must be sufficiently extensible and flexible to accommodate changing software design. The need for extensibility and flexibility must be balanced with the understanding that "all possible changes" cannot be accurately anticipated. As such the introduction of software design features to specifically accommodate extensibility within components should be deferred until they are needed and understood (the "fool me once" design principle [14]). This approach favors as-needed refactoring of software design and implementation over an up-front "over-engineering" of the software design, which can lead to an unnecessarily complex design and implementation.

A modern Agile software development process like Scrum [24,25,16] was realized by software development practitioners to effectively respond to changing requirements and design. This software development process is being deployed by many ASC software projects at SNL. It is expected that software development for NEAMS WF IPSC at SNL will be integrated with, and conform to, the in-place Agile/Scrum process.

An important process and practice for any technical effort are technical reviews. In software development, technical reviews of the requirements, architecture, design, code, and tests have all been shown to improve software quality, reduce defects, and reduce the cost of developing and maintaining software [13]. Such reviews processes are not common in research-oriented CS&E organization but it is critical to the NEAMS WF IPSC effort to develop a culture where these types of reviews are regularly conducted and become ingrained in day to day work.

7.3 Configuration Management

Configuration management refers to storing software products and other artifacts so they can be communicated among a development community, to facilitate access to previous versions, to record and track changes (the when, why, how), and to back up the products and artifacts to avoid loss.

7.3.1 Collaborative Development Environment

There are a few reasonable products that integrate the component tools useful for a large-scale software development project. The Trac product is an open source solution which is actively being used at Sandia. Its web site is http://trac.edgewall.org/. It provides a usable and capable wiki, a few version control plugins, and flexible issue tracking.

Its strength is in flexibility, which derives from that fact that low level building blocks are accessible to the users. This also means that many advanced capabilities have to be "programmed" at this lower level. Therefore, some development and support will be necessary to provide the necessary capability to NEAMS projects.

A concern with Trac is that the document management support is less than ideal. That is, it can attach documents to wiki pages and issue trackers, but does not version control these. One would assume that Trac plugins could be written (or are already available) to address these issues.

Another advantage for considering Trac is that there are a several projects here at Sandia that are already using the Trac product, which allows leveraging of existing experience and access to more advanced, "programmed" capabilities.

Note that the Trac product (and other collaborative development products) does not manage mailings lists in an integrated way. However, this capability could easily be achieved by using a separate mailing list

management product, such as Mailman at http://www.gnu.org/software/mailman/index.html or could be developed as an additional plugin for Trac.

7.3.2 Version Control

In this section, version control refers to the tool used to store and provide complete history of individual files, such as source code, tests, documentation files, and other important software artifacts [13].. It should be noted that binary files can be stored this way and, therefore, be used as a form of general document management.

By far, the mostly widely used version control tools are CVS, Subversion, and Git, in that order. CVS should not be considered due to significant drawbacks with binary files and lack of atomic operations (among others problems). There are a number of projects at Sandia that use Subversion and a few that use Git.

At a very high level, Subversion is easier to use while Git provides some additional features. Git has considerable flexibility for collaborating among development groups and individuals without requiring a central repository. However, the Git interface is nontrivial, which requires training and expert knowledge to be at hand. The Subversion interface is fairly straightforward with a more standard working concept. However, branching support in Subversion is not as clean and solid as Git and communicating among peers with Subversion requires the use of a central repository.

7.3.3 Issue Tracking

There are really two categories for users of an issue tracking system: project developers and end users. The end users require an easy to use interface and may also require access external to Sandia. The issue tracking tool can also be used to track the development and maintenance effort and can be used to accumulate valuable project data that can later be used to perform various types of analysis (such as number of defects found, percentage of time fixing defects, etc.).

The product for collaborative development should include an issue tracking system sufficient for project developers. For end users, it may be worthwhile to examine the issue tracking products available, such as Bugzilla or commercial software. Note that the early phases of the NEAMS program should not need an external, end customer focused issue tracking mechanism. Also, the Trac product could be used for end customers if sufficient efforts are made to develop more advanced tracker interface using lower level constructs.

7.3.4 Backup and Recovery

The most cost effective solution for backup and recovery is to utilize the expertise and existing mechanisms of the network and machine system administrators. An installation of Trac or some other collaboration product will be on one machine and that machine must be on a network that provides daily or even hourly backup. All work product artifacts are then backed up.

Recovery consists of either restoring the machine by the system administrators or obtaining access to the backup files and extracting the needed data.

Backup and recovery is also closely related to version control. The version control databases (e.g. Git or Subversion) provide the means to track and reproduce software product but the backup and recovery infrastructure is a key foundation to product and support this. Note that one of the advantages of Git is that backup and recovery is built into the tool with the ability to clone repositories across multiple machines and then merge them back together again. In fact, if all software artifacts where controlled with

Git, then one could construct a low cost, low tech backup and recovery system without more substantial sysadmin support.

Note that no backup and recovery system will be sufficient unless it is routinely tested through an automated testing process. The automated testing process needs to access the backed-up data (e.g. could just be accessing a cloned set of Git repositories) and then verify that the recovery process is able to reproduce the software artifacts.

7.4 Build and Test

Building the code means compiling the software source code into one or more executables for a given platform. These executables can then be launched with appropriate arguments to perform their intended function. The executables are tested by running through a suite of command line arguments and data inputs and comparing to verified results [53, 50]. Performing this process of build and test quickly and flexibly is crucial to efficient software development and for delivering a quality product.

The main considerations of a build system are capabilities for operation in a large, complex code base, ease of configuration and maintenance, and usability in an external installation environment. There are a plethora of build systems in use in the scientific community; however, the CMake product currently stands out as an obvious choice. It is open source, it is gaining momentum in the larger software world, Sandia has connections with Kitware (the company that develops CMake), it is being used by a handful of Sandia projects, and it appears to have the capabilities needed for a large code project.

Testing tools significantly impact development efficiency and code quality. Developers of larg complex codes, such as NEAMS WF IPSC, must be able to manage thousands of tests on a continual basis. Tests must be easy to add, easy to determine why failures occur, easy to filter and choose tests to be run or rerun, run on standalone platforms as well as batch based parallel machines, and allow convenient verification-type testing. However, no test harness will automatically produce high-quality tests and that is the subject of much frustration in the CS&E community [51].

Currently, no test support tool exists that can satisfy all these needs. The CMake family of tools provides CTest which is functional and is undergoing further enhancements. There are also existing test harnesses that have been home grown in Sandia projects and are fairly effective. If CMake is chosen for the build system, then probably CTest would be reasonable if the needed enhancements are made.

Finally, previous lessons learned and knowledge from the general software engineering community indicate that a focus on release and distribution testing must be done early in the project's development [21]. Release and distribution testing includes collecting the product from version control, packaging it for external and internal installation, and actually performing mock installations. Effective installation ensures flexible development models as well as the ability to quickly get the product to the customer with new features.

More detailed practices that improve software quality related to testing include the following.

- Develop (as much as possible) self verifying automated test suites
- For all new code, develop with strong unit tests that achieve near 100% line coverage and near 100% unit feature coverage.
- Change legacy code by first breaking dependencies, getting unit tests in place, refactoring code to get ready for new functionality, and then adding new functionality along with new unit testing code [22]

- Failing tests will be fixed, modified, or disabled in short order. Tests will not be allowed to fail for long periods of time. [21]
  Regression tests should be based on the foundation of automated verification tests.

- Verification tests should be based on quantitative, verifiable criteria. Manual inspect as the basis for verification tests should be minimized [53, 50]

7.5 Internal Software Integration

Keeping software components integrated as they are developed is critical to software productivity, quality, and risk management. The gold standard approach to software integration for modern Lean/Agile software development is *Continuous Integration* (CI) [16,21,23]. There are two main approaches to full CI; synchronous CI, and asynchronous CI. *Synchronous CI* (SCI) requires developers to fully integrate and test their changes before each check-in to the configuration managed repository. *Asynchronous CI* (ACI) involves developers doing much less thorough testing before each check-in. After each such check-in, a CI server automatically detects the check-in and then proceeds to checkout, build, and run a more substantial test suite, and then informs developers if anything fails. With SCI, the code is not committed unless all of it builds and all of the tests pass. With ACI, if the build or any test fails on the CI server then it is flagged and fixed by the team with "Stop the Line" priority [16]. SCI is the premiere CI method in terms of code stability, but ACI can scale to larger projects at the cost of greater code instability. In projects where ACI starts to produce failing builds too often, other CI-like methods may be considered [48]. Using CI requires that the software build and regression testing to be completely automated and this was described above.

At some point the size of a project will become too large to realistically apply any reasonable single CI method and other less-than-full CI methods must be considered. It is likely that the NEAMS WF project will eventually become too large to develop under a single blanket of full CI. When this happens, the best approach is to partition the code base into distinct pieces with carefully architected interfaces and then to define appropriate less-than-full CI methods to keep the software integrated on a reasonable (but not continuous) schedule. Eric Evans in [15] describes a number of different code partitioning and staged integration strategies. The strategy that is most applicable to the type of CS&E software environment being considered in the NEAMS WF effort is the Customer/Supplier relationship. A less-than full CI method being called Almost CI is described in [49] and has been used to successfully keep SIERRA and Trilinos integrated together in short time windows and would be the most idea chose for less-than-full CI methods.

7.6 External Software Support and Collaboration Models

Integrating externally developed software into the NEAMS WF IPSC has an associated set of potential benefits along with a set of additional risks [15]. While it is clear that the NEAMS WF IPSC cannot and should not develop all of its core capabilities from scratch and will have to incorporate various externally developed software products, at the same time we need to be mindful of the risks and apply the appropriate planning and due diligence to mitigate the risks of depending on externally developed software.

When considering the incorporation of an externally developed software package (referred to here as Package X) into the NEAMS WF IPSC, there are several different integration and upgrade models to choose from, listed from the loosest to the tightest integration include:

1. Absorb the sources for Package X and never upgrade: The source for a specific version of Package X would be assimilated into the sources for the NEAMS WF IPSC and no upgrades

would ever be expected. In these case, the burden of maintenance and support of this code fully falls on the NEAMS WF IPCS team.

2. Perform periodic punctuated upgrades of Package X: The NEAMS WF IPSC accepts new upgrades of Package X as they are released or as needed with no testing in between major releases of Package X. This is the most common way that most projects handle integration with an external software package but it is also the most risky.

3. Keep Package X working with the NEAMS WF IPSC through Daily Integration testing: In order to ensure that changes to the development version of Package X do not break the NEAMS WF IPSC, the development versions of Package X and the NEAMS WF IPSC are checkout out and built together every day. When a build or test fails, the problem is immediately addressed. When a new version of Package X is released, the NEAMS WF IPSC is upgraded in a safe manner. This mode of integration works very well for most situations but requires that Package X be developed in a stable manner (i.e. consistent with modern Lean/Agile software engineering standards) and keeps good backward compatibility.

4. Enable co-development of the NEMAS WF IPSC and Package X through Almost CI: When new capabilities in Package X are being developed in order to support the NEAMS WF IPSC then a more aggressive model of integration may be needed. The Almost Continuous Integration (Almost CI) approach would have the NEAMS WF IPSC frequently upgrade the sources for Package X as they are developed. This allows for co-development of Package X and the NEAMS WF IPSC. This is the closest from of integration possible next to full CI.

The first integration strategy of just absorbing the sources and never upgrading is no integration strategy at all but is just a code seeding approach. The last three integration strategies listed above are discussed in the context of CS&E software in [49]. Each of these various integration strategies will be appropriate for a given external software package and the best strategy will likely change during the life of the NEAMS WF IPSC development and maintenance effort.

When considering whether to incorporate an externally developed piece of software, Package X, and what integration strategy to use, there are a number of factors to contemplate including:

- Level of dependence on Package X: Does the NEAMS WF IPSC strongly depend on Package X in a unique way or it is just used for optional functionality? Is the surface area of interaction with Package X high or low? If the surface area is high, then the client NEAMS WF IPSC is placed at greater risk.
- Level of duplication of functionality in Package X with other external packages: Is the functionality in Package X unique or is there similar overlapping functionality in other available software packages? If there is no duplication, then the client NEAMS WF IPSC is placed at greater risk.
- Level of sophistication of Package X: Are the software/algorithms in Package X very sophisticated or can basic satisfactory versions of the capabilities be developed from scratch if needed? If the software/algorithms in Package X are very sophisticated then the risk to the client NEAMS WF IPSC is higher.
- Ease or difficulty of independent verification of Package X: Is it relatively easy to independently verify that Package X is working correctly or is independent verification more difficult? If independent verification will be difficult, the NEAMS WF IPSC project will need to accept the verification claims provided by the providers of Package X and this increases the risk especially when upgrading versions of Package X.
- Level of active development of Package X: Is Package X being actively developed or is it in maintenance mode? If Package X is being actively developed, then there is greater risk that changes

will be made that will break backward compatibility or break critical functionality being used by the NEAMS WF IPSC when upgrades of Package X are accepted.

- Need for new functionality and upgrades of Package X: Is the functionality currently present in Package X likely to be sufficient for the NEAMS WF IPSC for many years to come or are there key bits of functionality that will need to be added? If the current functionality in Package X is not sufficient, then future upgrades will be needed and the risk to the NEAMS WF IPSC is elevated.
- Interdependence of Package X on other external software packages: Does Package X have dependencies on other software packages that are also used directly or indirectly by the NEAMS WF IPSC? The greater the web of dependencies, the greater the risk to the NEAMS WF IPSC, especially when upgrades are considered.
- Level of quality needed for Package X and associated Quality Rigor Level: Is the functionality being provided by Package X going to be critical to the highest Quality Rigor Level computations that the NEAMS WF IPSC will be used for? If Package X functionality is critical to high Quality Rigor Level computations, then the risk is elevated.
- Level of Software Quality Engineering used to produce Package X: What is the level of skill and discipline used by the primary developers to implement and maintain Package X? Is Package X developed at a high level of quality throughout the development cycle in a modern Lean/Agile consistent way or are lower quality and ill-defined processes used? If Package X is developed with stable sources, then the daily integration strategies described above will not work and this limits the level of dependence on Package X that would be advisable.
- Release schedule for Package X: How often are releases of Package X put out? Is Package X released on fixed frequent intervals (i.e. consistent with modern Lean/Agile methods) or is the release schedule ill-defined or are official releases only made once a year or even less frequently? The more irregular the release schedule and the further between releases of Package X, the greater the risk. For example, if new capabilities are added to Package X that the NEAMS WF IPSC needs but Package X does not put out frequent releases, then there is a risk that NEAMS WF IPSC deliverables may be at risk of not delivering sufficient capabilities in Package X.
- Level of relationship and pull between the NEAMS WF IPSC and the developers of Package X: Does the NEAMS WF IPSC team have a lot of pull with the main development group of Package X or will the developers of Package X be unresponsive to the needs of the NEAMS WF IPSC team? If the main development team for Package X is not responsive to the needs of the NEAMS WF IPSC, then the risk is significantly elevated.
- Stability of the organization that develops and supports Package X: How stable is the organization that develops, maintains and supports Package X? If Package X is developed as a shorter term research effort, the risk of depending on that package is much elevated. However, if Package X is developed by a very stable organization and Package X is used internally within that organization in significant ways, then it is likely that Package X will continue to be developed and supported for many years to come.
- Usage of Package X by other NEAMS IPSCs: Will Package X be used by other NEAMS IPSC efforts or will it only be used by the NEAMS WF IPSC? If several NEAMS IPSC efforts will depend on Package X, then the resources needed to support Package X can be pooled together and the risk of depending on Package X will be reduced. However, if the NEAMS WF IPSC goes it alone in using Package X, then the risk is higher.

These are just some of the issues to consider when deciding on what external software packages to incorporate and then deciding which if the integration strategies described above to use. While these are some of the important issues to consider, there is no precisely defined decision tree that will determine the

best integration strategy. However, we can consider a few different examples to see how these issues can be used to select an integration strategy and address the risks.

First, let's consider a case where performing periodic punctuated upgrades of Package X is perfectly acceptable and low risk. As an example, consider a dependence on BLAS (Basic Linear Algebra Subroutines). Depending on the BLAS for low-level dense linear algebra computations is attractive because it is a standard interface and there are several high performance and even platform-specific optimized implementations to improve accuracy and speed. The risk of depending on BLAS is very low when considering the issues outlined above. First, there are many different implementations of BLAS including the reference BLAS that provides the basic source code. It is trivial to provide the basic (but suboptimal) algorithm implementations. It is easy (but perhaps tedious) to independently verify the correct behavior of the BLAS routines. There have been no changes being made to the BLAS interface for over 30 years. There is no need for new functionality in the BLAS. Lastly, BLAS has no significant dependence on any other software package. Any one of these would make the risk of depending on the BLAS low.

Now, let's consider an example of a package dependence that is much higher risk. A particularly high risk type of dependency is the dependence on an application framework, such as the NEAMS WF IPSC depending on the Salome and/or SIERRA frameworks. An application framework defines the overall structure for some significant piece of the software and typically defines various base classes that are specialized for the specific application. The surface area of exposure to a framework is usually very high and it is typically very difficult to phase out the use of a framework or change frameworks. If the framework does not provide critical capabilities the risk is reduced. However, if the framework does provide complex and critical capabilities and if it will need to provide more capabilities for the NEAMS WF IPSC then the risks become very significant. Larger pieces of software like complex frameworks also tend to have a lot of other package dependencies that may conflict with other dependencies. For example, external package X may depend on SuperLU version A which may not be compatible with SuperLU version B that is used in another dependent external package.

For these types of higher risk dependences, the daily integration approach or the Almost CI approach will be needed to mitigate the risks. However, these tighter daily integration approaches require Package X to be developed with a high level of stability and quality which is not common in the CS&E community. Actually, if you think about it, if the team developing Package X is highly skilled and disciplined (e.g. keeping very stable high quality development sources), if the Package X is released very frequently on a fixed schedule, and if there is strong commitment for the needs of the NEAMS WF IPSC, then all of the other risk factors mostly fall away. New capabilities can be added to Package X and co-developed with the NEAMS WF IPSC and low-risk releases of the NEAMS WF IPSC with upgrades to Package X can go out on a fixed schedule to meet NEAMS deliverables.

7.7 Release and Distribution

Release and distribution are at the heart of delivering capabilities to customers. Modern Agile methods involve putting out new high quality releases on relatively frequent fixed time schedules [14, 16, 23, 24, 25]. Some projects only put out releases when the development sources have reach sufficient "maturity" or when some given number of new features are ready. Modern Agile methods instead fix the release schedule and then the Scrum process (or a related Agile process) works to deliver the maximum value possible with each new release given a fixed set of development resources. There are many advantages to releasing software on short fixed intervals. First, it gets the development team into the habit of creating completed working software. It makes the release process lower risk and go more smoothly. Finally, it reduces schedule risks for customers since they can pick up a new release whenever they need an upgrade to meet their deliverables.

There are several key elements needed to effectively put out frequent high-quality releases on short fixed increments. The development process needs to be conducted in fixed-time iterations (e.g. Scrum Sprints). Every new capability that gets added needs to be quickly completed, ready for release (i.e. the Scrum idea of "Done"). Existing capabilities need to be maintained at the highest quality and defect introduction needs to be minimized. In addition, the release process needs to be well defined and low overhead.

Putting out a release should involve minimum extra overhead and low risk. The build and testing infrastructure used to drive the development process should be exactly the same infrastructure used to create and test release distributions. When the development sources are ready for a release, the sources should be tagged and branched in the version control repository, the software should be built and tested on platforms as similar as possible to the final customer platforms, and the full test suites (including the customer's acceptance tests) should be run. All tests must pass before putting out the release or failing tests should be disabled and the problems should be added to the "known issues" section in the release notes. The test suites should all be automated strong tests with pass/fail. Manual inspection tests should be kept to a minimum. This is made easier if the main development platform is made to mimic the final customer distribution platform as much as possible.

Only a skilled and knowledgeable development team can create and release software in this manner. Therefore, the issue of developer skill sets and training is discussed in the next section.

7.8 Training

There are a variety of different types of training that are needed for the NEAMS WF ISPC project. There is developer and software project management training that is needed and typical user training that is needed. These types of issues will be similar to any long-lived software development project that is expected to produce complex high-quality software. However, the challenges in the computational science & engineering (CS&E) environment may be more challenging because of the general lack of software development background in many CS&E organizations.

Achieving the level of software quality needed for high-credibility Quality Rigor Level 1 computations will require a certain level of knowledge and skill in the software development teams at all levels. The level of skill and knowledge will be higher than many in the more research-dominated CS&E communities may be accustomed to. While some of the tasks and aspects of producing complex highly integrated software will require significant knowledge and skill from a subset of the developers, a more basic set of knowledge and skills will be required by everyone that touches the code. Here we will describe several different levels of software knowledge and skills needed to create complex high-quality CS&E software and identify training targets and requirements to achieve the critical mass needed to produce high credibility CS&E software.

7.8.1 Common Developer Skills and Development Themes

Here, we point out several different sets of skills and knowledge that are important for developing high quality software that are not universally common in the CS&E community. These skills will be referred to when discussing the various levels of software development in Section 7.8.2. All of the relevant skills and knowledge sets needed to create high quality software are not mentioned below; only the more significant items are mentioned. However, we also only mention issues that require the assimilation of new skills and the development of new habits and are not just simple practices that are immediately picked up by most people. A short listing of some of the required knowledge and skill sets include:

1) *Minimizing complexity*: Software's Primary Technical Imperative is the management of complexity [13]. All of the skills and issues discussed in this document and in the general software engineering literature are either directly or indirectly related to the management of software complexity. Leaning

the manage complexity is not really a specific skill or knowledge set but instead is a recognition of the importance of managing complexity and in learning specific skills and techniques to do so.

2) *Naming*: The number one tool for managing complexity and improving understandability and maintainability of software is the careful naming of software entities. There are good conventions and strategies for naming software entities [13].

3) *Elimination of duplication*: One of the most important considerations in software development is the ruthless elimination of duplicate code [13, 19, 20, 23]. Duplicate code makes refactoring very difficult and encourages bugs when incompatible changes are made in the code. Eliminating duplication without decreasing software comprehensibility and increasing complexity requires skill and good judgment but is critical to software developed in an Agile method or any software effort that will have a long life cycle.

4) *Structured unit testing*: Testing of code to achieve high line coverage, high branch coverage, and high date flow coverage falls under the area of structural testing [13]. High quality code developed in an Agile way should be able to achieve near 100% line coverage and very high, if not complete, levels of branch and data flow coverage. Being able to write high quality code with full unit testing requires discipline and skill.

5) *Test-driven development (TDD)*: Developing functional unit tests before writing the code is known as test-driven (or test-first) development [19, 13, 23, 25]. Studies have shown a positive correlation in productivity and software quality with TDD. Using TDD, achieving high quality and high coverage unit tests comes almost automatically. Getting into the habit of doing TDD requires some conditioning before it becomes second nature for many people and learning how to do TDD effectively requires some effort.

6) *Pair Programming*: Collaborative software development approaches have proven to improve the quality and productivity of software development and to achieve higher defect removal rates higher more cheaply than by only functional testing [13]. In Agile methods, pair programming (where two developers work together to write code on one computer) have proven nearly as effective in productivity and defect removal rates as formal code reviews [13]. Programming in pairs and knowing when it is better to code alone involves a learning curve in order to achieve sustainable effective work.

7) *Basic object-oriented design*: The basic object-oriented concepts of abstraction, encapsulation, and polymorphism are critical to improving software quality and managing complexity. This requires making the OO paradigm shift [14].

8) *Structured refactoring*: Agile software development methods that create well factored and maintainable software are impossible without a systematic approach to software refactoring [15, 23]. It takes time, knowledge, and discipline to learn how to refactor code safely and effectively. While there are refactoring tasks that can take place within a single subroutine, more typically refactoring involves the manipulation of code between subroutines and classes and therefore requires a basic proficiency in basic object-oriented design.

9) *Intermediate and advanced object-oriented design*: Constructing larger more feature full software in a larger group setting while effectively addressing complexity requires going past basic object-oriented knowledge and skills. More advanced OO includes design patterns [18] and collaborative communication (i.e. UML [17]) are needed.

10) *Basic software engineering*: Effectively developing and delivering high-quality software for a single smaller team requires a set of skills and knowledge base that includes version control management,

build and test infrastructure, functional testing, continuous integration (CI), and basic software deployment. Best practices related to this are described in [13, 16, 21].

11) *Large-scale software integration, life-cycles, and deployment.*: Integrating different software development efforts with larger collections of separately developed software requires even a higher level of knowledge and skills. Software life cycle issues such as release schedules, and less than CI approaches must be known and understood [21, 49].

7.8.2 Levels of Software Development

In the prior section, different software knowledge and skill sets that comprise some the best practices of software development and engineering were described. In this section, we consider the different levels of software development and what skills apply. In general, each level of software development requires the full set of knowledge and skills of all of the lower levels.

1) *Laying out low-level statements and control structure within routines*: The ability to write sufficient quality low-level code within a single routine would seem to come naturally to any developer of average or better intelligence but experience has shown this not to be true. Studies have shown that routines with high complexity tend to have more defects. Testability, understandability, changeability, and general maintainability are rooted in high quality code at the statement and control-structure level. Excellent guidance on writing high-quality coded at subroutine level is found in [13]. The basic skills required include *managing complexity*, *naming*, *elimination of duplication*, *structured unit testing*, *TDD,* and *pair programming*. Every developer touching the IPSC code base needs to have these basic skill sets and knowledge base (or pair program with someone who does).

2) *Basic layout and development of basic classes and subroutines*: Going beyond writing high-quality code within subroutines, the next level of software development is designing and implementing basic classes and laying out data-structures and subroutines. The skills and knowledge base at this level includes all of those for writing lower-level statements and control structures but now also includes the skills *basic object-oriented design*, and *structured refactoring*. The majority, but not all, of individuals who contribute code to the NEAMS WF IPSC will write software at this level and will need to acquire the needed skills and knowledge base to perform this task well.

3) *Development of interoperating classes in a team environment*: Going beyond the development of basic classes, data-structures, and subroutines developed by single developers is the development of interoperable software written by different developers in the same team. This level of software development requires the skills and knowledge base inherent in *intermediate and advanced object-oriented design*. In a team of 8 developers, for instance, only about two of the developers need to have this level of skill and these developers would be considered to be *basic software architects*. The other team members will come to and consult these individuals when issues addressing larger scale interoperability and design are prevalent. An individual with this skill set and knowledge (including all of the lower-level skills and knowledge) would naturally act as a technical team lead. This level of software development would typically be all that is necessary for single SCRUM-like software development team.

4) *Integration of software between different teams in the same general organization*: It takes even more skills and knowledge in order to coordinate different development teams and keep software developed by the different teams integrated. The skills required in order to perform this type of work additionally include *basic software engineering*. At least one individual from each software team needs to have this set of knowledge and skills consistent with modern (Agile) software engineering best practices.

5) *Integration of larger collections of software from different organizations and delivering software to end customers*: The highest level of software engineering considered in the NEAMS WF IPSC is the coordination of larger-scale software development efforts from different teams in different organizations and managing releases of production software to end customers. The additional skills and knowledge set needed for this type of work are *large-scale software integration, life-cycles, and deployment*. This level of software engineering knowledge and skills will only be possessed by a handful of individuals related to the NEAMS WF IPSC and related efforts.

The purpose of listing out the various levels of software development above is to clearly identify the different types of roles that developers of the NEAMS WF IPSC must take on and what types of skills and knowledge sets must be obtained. As described above, there are a basic core set of skills and knowledge that everyone associated with developing the NEAMS WF IPCS must possess. This gives a baseline for the evaluation of individual development team members and helps target specific training activities which are discussed in the next section.

7.8.3 Training, Mentorship and Collaborative Development

There are several different types of people when it comes to acquiring software development skills: a) Individuals that will pick up the necessary knowledge and skills from reading and self study, b) individuals that will obtain the necessary knowledge and skills through formal classes, c) individuals who will only learn through one-on-one interactions with a knowledgeable and skilled mentor, and d) individuals who will never obtain the necessary skills and knowledge to produce high-credibility high-quality software. Achieving a critical mass of knowledge and skills will require a multi-faceted effort.

An effective training strategy will most likely have to begin with some formal software development courses taught by professional software instructors (such as provided by Construx). After taking the basic software training course, those with the initiative will then be able to go off on their own and obtain the rest of the knowledge and skills they will need. Having a recommended list of books and other articles along with perhaps setting up reading groups will be sufficient for this group of people.

For other individuals, more one-on-one instruction will be needed with qualified mentors. An effective way to mentor such individuals is through Pair Programming [13, 23]. Pair Programming involves two people sitting behind a computer writing code together. Pair Programming is most effective from a variety of perspectives when a less experienced developer is teamed up with a more experienced developer. In these pair programming sessions, knowledge and experience from the more experienced developer/mentor will naturally and organically flow down to the less experienced developer. The pace of development may be quite slow at first when a very inexperienced developer is teamed with a much more knowledgeable developer but that is fine as long as it is recognized that one of the primary goals of the pair programming process is training as much as it is about writing actual software. As time goes on, the various pairs of programmers will start to reach a similar level of knowledge and skills and the pace and quality of the development effort will go up and stabilize. Once a pool of developers have reached similar levels of skill and knowledge, the need to pair program all code will diminish and more code can be written individually (but will still require some form of code review process). However, some level of pair programming should always continue in order to keep the team well jelled.

A final group of individuals will never pick up the necessary software knowledge and skills to develop high-quality software. However, many of these people are skilled experts in their non-software domains and need to be intimately involved in the project and are critical to the project's success. There have been cases in other projects where these people were alienated from a project because of a software skills/interests gap and it came at the expense of the project's success [50]. A possible way to address this problem and keep these people involved at the deepest levels of the software is to have them always pair program with a more skilled and knowledgeable software developer to write all code. In this way, these

individual's extensive and critical domain knowledge and experience can be exploited and still result in high quality code. By working in such pairs, the more experienced software developer will be constantly reviewing and insuring basic software quality while at the same time the domain non-software savvy domain expert will ensure that the software is written in a way that is consistent with the domain itself.

7.9 User support

As the NEAMS WF IPSC effort progresses and begins to put out regular releases, issues of user support and training will need to be considered. User support includes the creating of user-level documentation, setting up a user support infrastructure, providing targeted user training, coordinating upgrades of the software, and addressing defects and supplying patches.

The NEAMS program as a whole needs to develop a user support plan that encompasses the various individual areas for which the NEAMS WF IPSC is just one. For example, some clusters of users might be using more than one of the NEAMS IPCS codes and therefore a higher level of user support infrastructure may be justified.

User support and training we become more of an issue once the basic scientific challenges have been sufficiently addressed.

8. References

1. Peters M, Ewing R, and Steefel C (2008) GNEP Waste Form Campaign Science & Technology and Modeling & Simulation Program: Roadmap with Rational and Recommendations, GNEP-M50-3040-303, Global Nuclear Energy Partnership Waste Form Campaign, U.S. Department of Energy, Office of Nuclear Energy, Washington D.C.
2. *Advanced Fuel Cycle Initiative Quality Assurance Program Document (AFCI-TIO-PMO-QA-PL-2009-000042)*, Idaho National Laboratory
3. Morgan, H. S, R. D. Krieg, and R. V. Matalucci, *Comparative Analysis of Nine Structural Codes Used in the Second WIPP Benchmark Problem*, SAND81-1389, Sandia National Laboratories, Albuquerque, New Mexico, November 1981.
4. Silling, S. A., *Final Technical Position on Documentation of Computer Codes for High-Level Waste Management*, NUREG-0856, U. S. Nuclear Regulatory Commission, Office of Nuclear Material Safety and Safeguards, 1983.
5. Sandia National Laboratories. "Total System Performance Assessment Model/Analysis for the License Application." MDL-WIS-PA-000005 REV 00 AD 01, 2008.
6. J. C. Helton and C. J. Sallaberry. Illustration of sampling-based approaches to the calculation of expected dose in performance assessments for the proposed high level radioactive waste repository at Yucca Mountain, Nevada. Sandia National Laboratories Technical Report SAND2007-1353, April, 2007.
7. M. S. Eldred. Strategic Plan for DAKOTA Research, Development, and Deployment, Sandia Internal Technical Memo, December 3, 2008.
8. D. E. Munson and H. S. Morgan, "Methodology for Performing Parallel Design Calculations (Nuclear Waste Repository Application)," SAND85-0324, Sandia National Laboratories, Albuquerque, NM, May 1986.
9. H. S. Morgan, M. Wallner and D. E. Munson, "Results of an International Parallel Calculations Exercise Comparing Creep Responses Predicted with Three Computer Codes for Two Excavations in Rock Salt," SAND87-2125, Sandia National Laboratories, Albuquerque, NM, November 1987.
10. U. S. Department of Energy (DOE), Title 40 CFR Part 191 Compliance Recertification Application for the Waste Isolation Pilot Plant, 10 vols., DOE/WIPP 2004-3231, Carlsbad Field Office, Carlsbad, NM, March 2004.
11. Minana M, Turgeon J, Pilch M, and Hackney P. *(2009) Sandia National Laboratories Advanced Simulation and Computing (ASC) Software Quality Plan: ASC Software Quality Engineering Practices Version 3.0 (SAND2008-5517)*, Sandia National Laboratories, Albuquerque, NM
12. Bredemeyer Consulting Resources for Software Architects (2009) http://www.bredemeyer.com
13. McConnell, Steve. *Code Complete (Second Edition)*. Microsoft Press, 2004
14. Martin, Robert C. *Agile Software Development (Principles, Patterns, and Practices)*. Prentice Hall, 2004
15. Evans, Eric. *Domain-Driven Design. Addison Wesley, 2004*
16. Poppendieck, Mary and Tom. *Implementing Lean Software Development*. Addison Wesley, 2007
17. Fowler, Martin. *UML Distilled (Third Edition)*. Addison Wesley, 2004
18. Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns*. Addison Wesley, 1995
19. Beck, Kent. *Test Driven Development*. Addison Wesley, 2003
20. Fowler, Martin. *Refactoring (Improving the Design of Existing Code)*. Addison Wesley, 1999
21. Duvall, Paul, et al. *Continuous Integration*. Addison Wesley, 2007
22. Feathers, Michael. *Working Effectively with Legacy Code*. Prentice Hall, 2005
23. Beck, Kent. *Extreme Programming (Second Edition)*. Addison Wesley, 2005

24. Schwaber, Ken and Mike Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002
25. Kniberg, Henrik. *Scrum and XP from the Trenches*. Info, 2007
26. Brooks, Frederick. *The Mythical Man Month: 20th Anniversary Edition*. Addison Wes*ley, 1995*
27. Wilson, G.E. and Boyack, B.E., 1998. The role of the PIRT process in experiments, code development and code applications associated with reactor safety analysis, Nuclear Engineering and Design, Vol. 186, pp. 23-37.
28. Olivier, T.J. and Nowlen, S.P., 2008. A Phenomena Identification and Ranking Table (PIRT) Exercise for Nuclear Power Plant Fire Modeling Applications, NUREG/CR-6978, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, Washington, DC.
29. Tregoning, R.T., Apps, J.A., Chen, W., Delegard, C.H., Litman, R., and MacDonald, D.D, 2009. Phenomena Identification and Ranking Table Evaluation of Chemical Effects Associated with Generic Safety Issue 191, NUREG-1918, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, Washington, DC.
30. International FEP Database
31. Yucca Mountain Project FEP Database
32. WIPP Database
33. Shadid, John, David Ropp. Stability of operator splitting methods for systems with indefinite operators: reaction-diffusion systems Source. Journal of Computational Physics. Volume 203 , Issue 2, March 2005, Pages: 449 – 466
34. Knoll, D., V.Mousseau1, L. Chacón, and J. Reisner. Jacobian-free newton-krylov methods for the accurate time integration of stiff wave systems. Journal of Scientific Computing. Springer. Volume 25, Numbers 1-2, November, 2005
35. M. Heinkenschloss, M. B. Hribar, and M. Kokkolaras. *Acceleration of multidisciplinary analysis solvers by inexact subsystem simulations*. In 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1998
36. Eric B. Becker, Graham F. Carey, and J. Tinsley Oden. *Finite Elements: An Introduction, volume 1*. Prentice Hall, London, second edition, 1981
37. K. E. Brennan, S. L. Campbell, and L. R. Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*, volume 14 of Classics in Applied Mathematics. SIAM, Philadelphia, PA, first edition, 1996
38. Hooper, Russell, Matthew M. Hopkins, Roger P. Pawlowski. *Enabling Newton-based Coupling within a Multi-physics Environment Using NOX - An Object-Oriented Nonlinear Solver Library*, Conference Paper, International Conference on Computational Methods for Coupled Problems in Science and Engineering, Santorini Island, Greece, May 2005
39. Cacuci, Dan G., Mihaela Ionescu-Bujor, Michael Navon. *Sensitivity And Uncertainty Analysis: Applications to Large-Scale Systems (Volume II)*, Chapman & Hall, 2005.
40. Tjoa, Bhieng, Lorenz T. Biegler. *Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems*. Ind. Eng. Chem. Res. 30 (2), pp 376-385, 1991
41. Bartlett, Roscoe. *Derivation of forward and adjoint sensitivities for ODEs and DAEs*, SAND2007-6699, Sandia National Laboratories. October 2007
42. Bartlett, Roscoe, Scott Collis, Todd Coffey, David Day, Mike Heroux, Rob Hoekstra, Russell Hooper, Roger Pawlowski, Eric Phipps, Denis Ridzal, Andy Salinger, Heidi Thornquist, and Jim Willenbring. *ASC Vertical Integration Milestone*. SAND2007-5839, Sandia National Laboratories, 2007
43. Michael Park, *Adjoint-Based, Three-Dimensional Error Prediction and Grid Adaptation*. 32nd AIAA Fluid Dynamics Conference and Exhibit, AIAA-2002-3286, 2002
44. Phipps, Eric, Mike Eldred, Andy Salinger, and Clayton Webster. *Capabilities for Uncertainty in Predictive Science*, SAND2008-6527, 2008

45. Bartlett, Roscoe. *Thyra Linear Operators and Vectors: Overview of Interfaces and Support Software for the Development and Interoperability of Abstract Numerical Algorithms*. SAND2007-5984, Sandia National Laboratories, 2007
46. Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns*. Addison Wesley, 1995
47. A. Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia, PA, 2000
48. D. Poole. *Multi-stage continuous integration*. Dr. Dobb's Journal, December 2008
49. Bartlett, Roscoe. *Integration Strategies for Computational Science & Engineering Software*. SAND2004-3268, Second International Workshop on Software Engineering for Computational Science and Engineering, 2009
50. Post, D., R. Kendall. *Software project management and quality engineering practices for complex, coupled multiphysics, massively parallel computational simulations: lessons learned from asci*. The International Journal of High Performance Computing Applications, Vol 18, No. 4, pp. 399416, 2004
51. Post, D. and L. Votta. *Computational Science Demands a New Paradigm*. American Institute of Physics, S-0031-9228-0501-020-3, 2005
52. Miller, Greg. *A Scientist's Nightmare: Software Problem Leads to Five Retractions*. Science, p. 1856-1857, 2006
53. Trucano, Timothy G., Douglass E. Post, Martin Pilch, William L. Oberkampf, *Software Engineering Intersections with Verification and Validation (V&V) of High Performance Computational Science Software: Some Observations*, White Paper, June 2005
54. Bartlett, Roscoe, Bart van Bloemen Waanders, and Michael Heroux. *Vector Reduction/Transformation Operators*. ACM Transactions on Mathematical Software. Vol. 30, No. 1, p. 62–85, 2004
55. Bartlett, Roscoe A., David M. Gay, Eric T. Phipps, *Automatic Differentiation of C++ Codes for Large-Scale Scientific Computing*, Conference Paper, Third International Workshop on Automatic Differentiation: Tools and Applications at ICCS 2006, February 2006
56. J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Verlag, Berlin, Heidelberg, New York, 1999

Appendix A

A. Phenomena Identification and Ranking Table (PIRT)

A preliminary importance ranking was performed on the phenomena identified in Table 3. . Separate rankings were performed for the high-fidelity models (Table A-1) and the surrogate models (Table A-2). The preliminary importance rankings were based on the reference scenario identified in Section 3.1, but were generally applicable to most scenarios. The ranking schemes are described in Sections 3.3 and 3.4.

Table A-1. Preliminary Phenomena Identification and Ranking Table (PIRT) for the High-Fidelity Continuum Models

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
0.0.00.00	ASSESSMENT BASIS					
1.0.00.00	EXTERNAL FACTORS					
1.1.00.00	1. REPOSITORY ISSUES					
1.2.00.00	2. GEOLOGICAL PROCESSES AND EFFECTS					
1.2.03.01	Seismic activity impacts EBS and/or EBS components	- Mechanical damage to EBS (from ground motion, rockfall, drift collapse, fault displacement) [see also Mechanical Impacts in 2.1.07.04, 2.1.07.05, 2.1.07.06, 2.1.07.07, 2.1.07.08, and 2.1.07.10]	2	1	1	2
1.2.04.01	Igneous activity impacts EBS and/or EBS components	- Mechanical damage to EBS (from intrusion intrusion) - Chemical interaction with magmatic volatiles - Transport of radionuclides (in magma, pyroclasts, vents) [see also Mechanical Impacts in 2.1.07.04, 2.1.07.05, 2.1.07.06, 2.1.07.07, and 2.1.07.08]	3	1	1	1
1.3.00.00	3. CLIMATIC PROCESSES AND EFFECTS					
1.4.00.00	4. FUTURE HUMAN ACTIONS					
1.4.02.01	Human Intrusion - Deliberate - Inadvertent		N/A. Not evaluated for high-fidelity continuum models			
1.5.00.00	5. OTHER					
2.0.00.00	DISPOSAL SYSTEM FACTORS					

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.00.00	1. WASTES AND ENGINEERED FEATURES					
2.1.01.00	1.01. INVENTORY					
2.1.01.01	Waste Inventory - Radionuclides - Non-Radionuclides	- Composition - Enrichment / Burn-up	3	3	2	2
2.1.01.02	Radioactive Decay and Ingrowth		3	3	3	3
2.1.01.03	Heterogeneity of Waste Inventory - Waste Package Scale - Repository Scale	- Composition - Enrichment / Burn-up - Damaged Area	2	3	2	2
2.1.01.04	Interactions Between Co-Located Waste		1	2	2	2
2.1.02.00	1.02. WASTE FORM					
2.1.02.01	SNF (Commercial, DOE) Degradation - Alteration / Phase Separation - Dissolution / Leaching - Radionuclide Release	Degradation is dependent on: - Composition - Geometry / Structure - Enrichment / Burn-up - Surface Area - Gap and Grain Fraction - Damaged Area - THC Conditions [see also Mechanical Impact in 2.1.07.06 and Thermal-Mechanical Effects in 2.1.11.06]	N/A. Not part of current WF IPSC scope.			
2.1.02.02	HLW (Glass, Ceramic, Metal) Degradation - Alteration / Phase Separation - Dissolution / Leaching - Cracking - Radionuclide Release	Degradation is dependent on: - Composition - Geometry / Structure - Surface Area - Damaged / Cracked Area - Mechanical Impact - THC Conditions [see also Mechanical Impact in 2.1.07.07 and Thermal-Mechanical Effects in 2.1.11.06]	3	1	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.02.03	Degradation of Organic/Cellulosic Materials in Waste	[see also Complexation in EBS in 2.1.09.17]	1	1	1	2
2.1.02.04	HLW (Glass, Ceramic, Metal) Recrystallization		1	2	1	1
2.1.02.05	Pyrophoricity or Flammable Gas from SNF or HLW		N/A. Not evaluated for high-fidelity continuum models			
2.1.02.06	SNF Cladding Degradation and Failure	- Initial damage - General Corrosion - Microbially Influenced Corrosion - Localized Corrosion - Enhanced Corrosion (silica, fluoride) - Stress Corrosion Cracking - Hydride Cracking - Unzipping - Creep - Internal Pressure - Mechanical Impact	N/A. Not part of current WF IPSC scope.			
2.1.03.00	1.03. WASTE CONTAINER					
2.1.03.01	Early Failure of Waste Packages	- Manufacturing defects - Improper sealing	N/A. Not evaluated for high-fidelity continuum models	2	1	1
2.1.03.02	General Corrosion of Waste Packages	- Dry-air oxidation - Humid-air corrosion - Aqueous phase corrosion - Passive film formation and stability	3	2	1.5	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.03.03	Stress Corrosion Cracking (SCC) of Waste Packages	- Crack initiation, growth and propagation - Stress distribution around cracks	3	2	1.5	2
2.1.03.04	Localized Corrosion of Waste Packages	- Pitting - Crevice corrosion - Salt deliquescence	3	2	1.5	2
2.1.03.05	Hydride Cracking of Waste Packages	- Hydrogen diffusion through metal matrix - Crack initiation and growth in metal hydride phases	2	1	1.5	2
2.1.03.06	Microbially Influenced Corrosion (MIC) of Waste Packages		2	2	1.5	2
2.1.03.07	Internal Corrosion of Waste Packages Prior to Breach		2	2	2	2
2.1.03.08	Flow In and Through Waste Packages	- Saturated / Unsaturated flow - Movement as thin films or droplets [see also Flow in EBS in 2.1.08.01]	3	2	1	2
2.1.03.09	Evolution Flow Pathways in Waste Packages	- Evolution of physical form of waste package - Plugging of cracks in waste packages [see also Evolution of Flow Pathways in EBS in 2.1.08.02, Mechanical Impact on Waste Packages in 2.1.07.05]	2	2	1	2
2.1.04.00	1.04. BUFFER/BACKFILL					
2.1.04.01	Evolution and Degradation of Backfill	- Alteration - Thermal expansion / Degradation - Swelling / Compaction - Erosion / Dissolution - Evolution of backfill flow pathways [see also Evolution of Flow Pathways in EBS in 2.1.08.02, Mechanical Impact on Backfill in 2.1.07.04, Thermal-Mechanical Impact in 2.1.11.08, Chemical Interaction 2.1.09.06]	3	1.5	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.04.02	Flow in Backfill	- Fracture / Matrix flow [see also Flow in EBS in 2.1.08.01]	2	2	1	2
2.1.05.00	1.05. SEALS					
2.1.05.01	Degradation of Seals	- Alteration / Degradation / Cracking - Erosion / Dissolution [see also Mechanical Impact in 2.1.07.04, Thermal-Mechanical Impact in 2.1.11.09, Chemical Interaction 2.1.09.08]	2	1	1	2
2.1.05.02	Flow Through Seals	[see also Flow in EBS in 2.1.08.01]	2	2	1	2
2.1.06.00	1.06. OTHER EBS MATERIALS					
2.1.06.01	Degradation of Liner / Rock Reinforcement Materials in EBS	- Alteration / Degradation / Cracking - Corrosion - Erosion / Dissolution / Spalling [see also Mechanical Impact in 2.1.07.08, Thermal-Mechanical Impact in 2.1.11.09, Chemical Interaction 2.1.09.07]	2	1	1	2
2.1.06.02	Flow Through Liner / Rock Reinforcement Materials in EBS	[see also Flow in EBS in 2.1.08.01]	2	2	1	2
2.1.07.00	1.07. MECHANICAL PROCESSES					
2.1.07.01	Rockfall	- Dynamic loading (block size and velocity)	1 (2: no backfill)	3	2	2
2.1.07.02	Drift Collapse	- Static loading (rubble volume) - Alteration of seepage - Alteration of EBS flow pathways - Alteration of EBS thermal environment [see also Evolution of Flow Pathways in EBS in 2.1.08.02, Chemical Effects of Drift Collapse in 2.1.09.12, and Effects of Drift Collapse on TH in 2.1.11.04]	3	2	2	2
2.1.07.03	Mechanical Effects of Backfill	- Protection of other EBS components from rockfall / drift collapse	3	3	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.07.04	Mechanical Impact on Backfill	- Rockfall / Drift collapse - Hydrostatic pressure - Internal gas pressure [see also Degradation of Backfill in 2.1.04.01 and Thermal-Mechanical Effects in 2.1.11.08]	3	2	2	2
2.1.07.05	Mechanical Impact on Waste Packages	- Rockfall / Drift collapse - Waste package movement - Hydrostatic pressure - Internal gas pressure - Swelling corrosion products [see also Thermal-Mechanical Effects in 2.1.11.05]	3	2	1.5	2
2.1.07.06	Mechanical Impact on SNF Waste Form	- Drift collapse - Swelling corrosion products [see also Thermal-Mechanical Effects in 2.1.11.06]	2	2	2	2
2.1.07.07	Mechanical Impact on HLW Waste Form	- Drift collapse - Swelling corrosion products [see also Thermal-Mechanical Effects in 2.1.11.06]	2	2	2	2
2.1.07.08	Mechanical Impact on Other EBS Components - Seals - Liner/Rock Reinforcement Materials - Waste Package Support Materials	- Rockfall / Drift collapse - Movement - Hydrostatic pressure - Swelling corrosion products [see also Thermal-Mechanical Effects in 2.1.11.09]	1	2	2	2
2.1.07.09	Mechanical Effects at EBS Component Interfaces	- Component-to-component contact (static or dynamic)	1	2	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.07.10	Mechanical Degradation of EBS	- Floor buckling - Fault displacement - Consolidation of EBS components - Degradation of waste package support structure - Alteration of EBS flow pathways [see also Evolution of Flow Pathways in EBS in 2.1.08.02, Degradation in 2.1.04.01, 2.1.05.01, and 2.1.06.01]	2	2	2	2
2.1.08.00	1.08. HYDROLOGIC PROCESSES					
2.1.08.01	Flow Through the EBS	- Saturated / Unsaturated flow - Preferential flow pathways [see also Flow in Waste Packages in 2.1.03.08, Flow in Backfill in 2.1.04.02], Flow through Seals 2.1.05.02, Flow through Liner in 2.1.06.02, Thermal Effects on Flow in 2.1.11.10, Effects of Gas on Flow in 2.1.12.02]	3	2	1	2
2.1.08.02	Alteration and Evolution of EBS Flow Pathways	- Drift collapse - Degradation/consolidation of EBS components - Plugging of flow pathways - Formation of corrosion products - Water ponding [see also Evolution of Flow Pathways in WPs in 2.1.03.09, Evolution of Backfill in 2.1.04.01, Drift Collapse in 2.1.07.02, and Mechanical Degradation of EBS in 2.1.07.10]	3	2	1	2
2.1.08.03	Condensation Forms in Repository - On Tunnel Roof / Walls - On EBS Components	- Heat transfer (spatial and temporal distribution of temperature and relative humidity) - Dripping [see also Heat generation in EBS in 2.1.11.01, Effects on EBS Thermal Environment in 2.1.11.03 and 2.1.11.04]	2.5	2.5	1.5	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.08.04	Capillary Effects in EBS	- Wicking	1	2	2	2
2.1.08.05	Influx (Seepage) Into the EBS	- Water influx rate (spatial and temporal distribution) [BOUNDARY CONDITION FOR WF IPSC]	3	2	2	2
2.1.09.00	1.09. CHEMICAL PROCESSES - CHEMISTRY					
2.1.09.01	Chemistry of Water Flowing into the Repository	- Chemistry of influent water (spatial and temporal distribution) [BOUNDARY CONDITION FOR WF IPSC]	3	2	1	3
2.1.09.02	Chemical Characteristics of Water in Waste Packages	- Water composition (radionuclides, dissolved species, …) - Initial void chemistry (air / gas) - Water chemistry (pH, ionic strength, pCO2, ..) - Reduction-oxidation potential - Reaction kinetics - Influent chemistry (from tunnels and/or backfill) [see also 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels] - Evolution of water chemistry / interaction with waste packages	3	2	1	3
2.1.09.03	Chemical Characteristics of Water in Backfill	- Water composition (radionuclides, dissolved species, …) - Water chemistry (pH, ionic strength, pCO2, ..) - Reduction-oxidation potential - Reaction kinetics - Influent chemistry (from tunnels and/or waste package) [see also 2.1.09.02 Chemistry in Waste Packages, 2.1.09.04 Chemistry in Tunnels] - Evolution of water chemistry / interaction with backfill	3	1	1	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.09.04	Chemical Characteristics of Water in Tunnels	- Water composition (radionuclides, dissolved species, …) - Water chemistry (pH, ionic strength, pCO2, ..) - Reduction-oxidation potential - Reaction kinetics - Influent chemistry (from near-field host rock) [see also 2.1.09.01 Chemistry of Water Flowing in, 2.1.09.02 Chemistry in Waste Packages, 2.1.09.03 Chemistry in Backfill] - Evolution of water chemistry / interaction with seals, liner/rock reinforcement materials, waste package support materials	3	2	1	3
2.1.09.05	Chemical Interaction of Water with Corrosion Products- In Waste Packages- In Backfill- In Tunnels	- Corrosion product formation and composition (waste form, waste package internals, waste package)- Evolution of water chemistry in waste packages, in backfill, and in tunnels [contributes to 2.1.09.02 Chemistry in Waste Packages, 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	3	2	1	3
2.1.09.06	Chemical Interaction of Water with Backfill - On Waste Packages - In Backfill - In Tunnels	- Backfill composition and evolution (bentonite, crushed rock, ...) - Evolution of water chemistry in backfill, and in tunnels - Enhanced degradation of waste packages (crevice formation) [contributes to 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels, 2.1.03.04 Localized Corrosion of WPs]	3	1	1	3

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.09.07	Chemical Interaction of Water with Liner / Rock Reinforcement and Cementitious Materials in EBS - In Backfill - In Tunnels	- Liner composition and evolution (concrete, metal, ...) - Rock reinforcement material composition and evolution (grout, rock bolts, mesh, ...) - Other cementitious materials composition and evolution - Evolution of water chemistry in backfill, and in tunnels [contributes to 2.1.09 .03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	3	1	1	3
2.1.09.08	Chemical Interaction of Water with Other EBS Components - In Waste Packages - In Tunnels	- Seals composition and evolution - Waste Package Support composition and evolution (concrete, metal, ...) - Other EBS components (other metals (copper), ...) - Evolution of water chemistry in backfill, and in tunnels [contributes to 2.1.09 .03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	3	1	1	3
2.1.09.09	Chemical Effects at EBS Component Interfaces	- Component-to-component contact (chemical reactions) - Consolidation of EBS components	3	1	1	3
2.1.09.10	Chemical Effects of Waste-Rock Contact	- Waste-to-host rock contact (chemical reactions) - Component-to-host rock contact (chemical reactions)	3	2	1	3
2.1.09.11	Electrochemical Effects in EBS	- Enhanced metal corrosion	2	1	1	2
2.1.09.12	Chemical Effects of Drift Collapse	- Evolution of water chemistry in backfill and in tunnels (from altered seepage, from altered thermal-hydrology) [contributes to 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	1 (2: unsat.)	1	1	3

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.09.13	Radionuclide Speciation and Solubility in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Dissolved concentration limits - Limited dissolution due to inclusion in secondary phase - Enhanced dissolution due to alpha recoil [controlled by 2.1.09.02 Chemistry in Waste Packages, 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	3	2	2	3
2.1.09.00	1.09. CHEMICAL PROCESSES - TRANSPORT					
2.1.09.14	Advection of Dissolved Radionuclides in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Flow pathways and velocity - Advective properties (porosity, tortuosity) - Saturation [see also Gas Phase Transport in 2.1.12.02]	3	3	2	2
2.1.09.15	Diffusion of Dissolved Radionuclides in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Concentration gradients - Diffusive properties (diffusion coefficients) - Flow pathways and velocity - Saturation	3	3	2	2
2.1.09.16	Sorption of Dissolved Radionuclides in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Sorptive properties (distribution coefficients) - Flow pathways and velocity - Saturation	3	2	2	3
2.1.09.17	Complexation in EBS	- Formation of organic complexants (humates, fulvates, organic waste) - Enhanced transport of radionuclides associated with organic complexants	2	2	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.09.18	Formation of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Formation of intrinsic colloids - Formation of pseudo colloids (host rock fragments, waste form fragments, corrosion products, microbes) - Formation of co-precipitated colloids - Sorption/attachment of radionuclides to colloids (clay, silica, waste form, FeOx, microbes)	2	2	2	3
2.1.09.19	Stability of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Chemical stability of attachment (dependent on water chemistry) - Mechanical stability of colloid (dependent on colloid size, gravitational settling)	3	1	1	2
2.1.09.20	Advection of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Flow pathways and velocity - Advective properties (porosity, tortuosity) - Saturation - Colloid concentration	3	2	1	2
2.1.09.21	Diffusion of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Concentration gradients - Diffusive properties (diffusion coefficients) - Flow pathways and velocity - Saturation - Colloid concentration	2	2	1	2
2.1.09.22	Sorption of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Sorptive properties (distribution coefficients) - Flow pathways and velocity - Saturation - Colloid concentration	2	1	1	2
2.1.09.23	Sorption of Colloids at Air-Water Interface in EBS		1	1	1	2
2.1.09.24	Filtration of Colloids in EBS	- Physical filtration (dependent on flow pathways, colloid size)- Electrostatic filtration	1	1	1	2
2.1.09.25	Radionuclide Transport Through Seals	- Advection - Diffusion - Sorption	2	2	1	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.10.00	1.10. BIOLOGICAL PROCESSES					
2.1.10.01	Microbial Activity in EBS - Natural - Anthropogenic	- Effects on corrosion - Formation of complexants - Formation of microbial colloids - Formation of biofilms - Biodegradation - Biomass production - Bioaccumulation [see also Microbiallly Influenced Corrosion in 2.1.03.06, Complexation in EBS in 2.1.09.17, Radiological Mutation of Microbes in 2.1.13.03]	2	2	2	2
2.1.11.00	1.11. THERMAL PROCESSES					
2.1.11.01	Heat Generation in EBS	- Heat transfer (spatial and temporal distribution of temperature and relative humidity) [see also Waste Inventory in 2.1.01.01]	3	3	2	2
2.1.11.02	Exothermic Reactions in EBS		1	1	1	2
2.1.11.03	Effects of Backfill on EBS Thermal Environment	- Thermal blanket - Condensation	3	3	2	2
2.1.11.04	Effects of Drift Collapse on EBS Thermal Environment	- Thermal blanket - Condensation	2	2	2	2
2.1.11.05	Effects of Influx (Seepage) on Thermal Environment	- Temperature and relative humidity (spatial and temporal distribution) [BOUNDARY CONDITION FOR WF IPSC]	3	3	2	2
2.1.11.06	Thermal-Mechanical Effects on Waste Form and In-Package EBS Components	- Alteration - Cracking - Thermal expansion / stress	2	2	1	3
2.1.11.07	Thermal-Mechanical Effects on Waste Packages	- Thermal sensitization / phase changes - Cracking - Thermal expansion / stress / creep	2.5	2.5	1.5	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.11.08	Thermal-Mechanical Effects on Backfill	- Alteration - Cracking - Thermal expansion / stress	3	2	1.5	2
2.1.11.09	Thermal-Mechanical Effects on Other EBS Components - Seals - Liner / Rock Reinforcement Materials - Waste Package Support Structure	- Alteration - Cracking - Thermal expansion / stress	1.5	2	1.5	2
2.1.11.10	Thermal Effects on Flow in EBS	- Altered saturation / relative humidity - Condensation	3	3	2	2
2.1.11.11	Thermally-Driven Flow (Convection) in EBS	- Convection	3	3	2	2
2.1.11.12	Thermally-Driven Buoyant Flow / Heat Pipes		2	2	2	2
2.1.11.13	Thermal Effects on Chemistry and Microbial Activity in EBS		3	2	1	2
2.1.11.14	Thermal Effects on Transport in EBS	- Thermal diffusion (Soret effect) - Thermal osmosis	1	2	1	1
2.1.12.00	1.12. GAS SOURCES AND EFFECTS					
2.1.12.01	Gas Generation in EBS	- Repository Pressurization - Mechanical Damage to EBS Components - He generation from waste from alpha decay - H2 generation from waste package corrosion - CO2, CH4, and H2S generation from microbial degradation	3	2	1	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.12.02	Effects of Gas on Flow Through the EBS	- Two-phase flow - Gas bubbles [see also Two-Phase Buoyant Flow in 2.1.11.12]	1	2	1	2
2.1.12.03	Gas Transport in EBS	- Gas phase transport - Gas phase release from EBS	2	2	1	2
2.1.12.04	Gas Explosions in EBS		1	2	1	1
2.1.13.00	1.13. RADIATION EFFECTS					
2.1.13.01	Radiolysis - In Waste Package - In Backfill - In Tunnel	- Gas generation - Altered water chemistry	2	2	2	2
2.1.13.02	Radiation Damage to EBS Components - Waste Form - Waste Package - Backfill - Other EBS Components	- Enhanced waste form degradation - Enhanced waste package degradation - Enhanced backfill degradation - Enhanced degradation of other EBS components (liner/rock reinforcement materials, seals, waste support structure)	2	2	2	2
2.1.13.03	Radiological Mutation of Microbes		1	1	1	1
2.1.14.00	1.14. NUCLEAR CRITICALITY					
2.1.14.01	Criticality In-Package	- Formation of critical configuration	2	2	2	2
2.1.14.02	Criticality in EBS or Near-Field	- Formation of critical configuration	2	2	2	2
2.2.00.00	2. GEOLOGICAL ENVIRONMENT					
2.3.00.00	3. SURFACE ENVIRONMENT					
2.4.00.00	4. HUMAN BEHAVIOR					

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
3.0.00.00	RADIONUCLIDE/CONTAMINANT FACTORS (BIOSPHERE)					
3.1.00.00	1. CONTAMINANT CHARACTERISTICS					
3.2.00.00	2. RELEASE/MIGRATION FACTORS					
3.3.00.00	3. EXPOSURE FACTORS					

Table A-2. Preliminary Phenomena Identification and Ranking Table (PIRT) for the Surrogate PA Models

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
0.0.00.00	ASSESSMENT BASIS					
1.0.00.00	EXTERNAL FACTORS					
1.1.00.00	1. REPOSITORY ISSUES					
1.2.00.00	2. GEOLOGICAL PROCESSES AND EFFECTS					
1.2.03.01	Seismic activity impacts EBS and/or EBS components	- Mechanical damage to EBS (from ground motion, rockfall, drift collapse, fault displacement) [see also Mechanical Impacts in 2.1.07.04, 2.1.07.05, 2.1.07.06, 2.1.07.07, 2.1.07.08, and 2.1.07.10]	2 (3: no backfill)	2.5	2.5	2
1.2.04.01	Igneous activity impacts EBS and/or EBS components	- Mechanical damage to EBS (from intrusion intrusion) - Chemical interaction with magmatic volatiles - Transport of radionuclides (in magma, pyroclasts, vents) [see also Mechanical Impacts in 2.1.07.04, 2.1.07.05, 2.1.07.06, 2.1.07.07, and 2.1.07.08]	3	2	2	1.5
1.3.00.00	3. CLIMATIC PROCESSES AND EFFECTS					
1.4.00.00	4. FUTURE HUMAN ACTIONS					
1.4.02.01	Human Intrusion - Deliberate - Inadvertent		3	3	2	1
1.5.00.00	5. OTHER					
2.0.00.00	DISPOSAL SYSTEM FACTORS					
2.1.00.00	1. WASTES AND ENGINEERED FEATURES					

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.01.00	1.01. INVENTORY					
2.1.01.01	Waste Inventory - Radionuclides - Non-Radionuclides	- Composition - Enrichment / Burn-up	3	3	3	3
2.1.01.02	Radioactive Decay and Ingrowth		3	3	3	3
2.1.01.03	Heterogeneity of Waste Inventory - Waste Package Scale - Repository Scale	- Composition - Enrichment / Burn-up - Damaged Area	2	3	3	3
2.1.01.04	Interactions Between Co-Located Waste		2	2.5	2.5	2
2.1.02.00	1.02. WASTE FORM					
2.1.02.01	SNF (Commercial, DOE) Degradation - Alteration / Phase Separation - Dissolution / Leaching - Radionuclide Release	Degradation is dependent on: - Composition - Geometry / Structure - Enrichment / Burn-up - Surface Area - Gap and Grain Fraction - Damaged Area - THC Conditions [see also Mechanical Impact in 2.1.07.06 and Thermal-Mechanical Effects in 2.1.11.06]	N/A. Not part of current WF IPSC scope.	2	2	2
2.1.02.02	HLW (Glass, Ceramic, Metal) Degradation - Alteration / Phase Separation - Dissolution / Leaching - Cracking - Radionuclide Release	Degradation is dependent on: - Composition - Geometry / Structure - Surface Area - Damaged / Cracked Area - Mechanical Impact - THC Conditions [see also Mechanical Impact in 2.1.07.07 and Thermal-Mechanical Effects in 2.1.11.06]	3	2	2	2
2.1.02.03	Degradation of Organic/Cellulosic Materials in Waste	[see also Complexation in EBS in 2.1.09.17]	3	2	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.02.04	HLW (Glass, Ceramic, Metal) Recrystallization		1	1	1	1
2.1.02.05	Pyrophoricity or Flammable Gas from SNF or HLW		1	1	2	2
2.1.02.06	SNF Cladding Degradation and Failure	- Initial damage - General Corrosion - Microbially Influenced Corrosion - Localized Corrosion - Enhanced Corrosion (silica, fluoride) - Stress Corrosion Cracking - Hydride Cracking - Unzipping - Creep - Internal Pressure - Mechanical Impact	N/A. Not part of current WF IPSC scope.	2	2	2
2.1.03.00	1.03. WASTE CONTAINER					
2.1.03.01	Early Failure of Waste Packages	- Manufacturing defects - Improper sealing	2.5	2.5	2.5	2
2.1.03.02	General Corrosion of Waste Packages	- Dry-air oxidation - Humid-air corrosion - Aqueous phase corrosion - Passive film formation and stability	3	2.5	2	2
2.1.03.03	Stress Corrosion Cracking (SCC) of Waste Packages	- Crack initiation, growth and propagation - Stress distribution around cracks	3	3	2	2
2.1.03.04	Localized Corrosion of Waste Packages	- Pitting - Crevice corrosion - Salt deliquescence	3	2	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.03.05	Hydride Cracking of Waste Packages	- Hydrogen diffusion through metal matrix - Crack initiation and growth in metal hydride phases	2	1	1	2
2.1.03.06	Microbially Influenced Corrosion (MIC) of Waste Packages		2	2	2	2
2.1.03.07	Internal Corrosion of Waste Packages Prior to Breach		1	2	2	2
2.1.03.08	Flow In and Through Waste Packages	- Saturated / Unsaturated flow - Movement as thin films or droplets [see also Flow in EBS in 2.1.08.01]	3	2	2	2
2.1.03.09	Evolution Flow Pathways in Waste Packages	- Evolution of physical form of waste package - Plugging of cracks in waste packages [see also Evolution of Flow Pathways in EBS in 2.1.08.02, Mechanical Impact on Waste Packages in 2.1.07.05]	3	2	2	2
2.1.04.00	1.04. BUFFER/BACKFILL					
2.1.04.01	Evolution and Degradation of Backfill	- Alteration - Thermal expansion / Degradation - Swelling / Compaction - Erosion / Dissolution - Evolution of backfill flow pathways [see also Evolution of Flow Pathways in EBS in 2.1.08.02, Mechanical Impact on Backfill in 2.1.07.04, Thermal-Mechanical Impact in 2.1.11.08, Chemical Interaction 2.1.09.06]	3	3	2	2
2.1.04.02	Flow in Backfill	- Fracture / Matrix flow [see also Flow in EBS in 2.1.08.01]	3	3	2	2
2.1.05.00	1.05. SEALS					

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.05.01	Degradation of Seals	- Alteration / Degradation / Cracking - Erosion / Dissolution [see also Mechanical Impact in 2.1.07.04, Thermal-Mechanical Impact in 2.1.11.09, Chemical Interaction 2.1.09.08]	3	3	3	2
2.1.05.02	Flow Through Seals	[see also Flow in EBS in 2.1.08.01]	3	3	3	2
2.1.06.00	1.06. OTHER EBS MATERIALS					
2.1.06.01	Degradation of Liner / Rock Reinforcement Materials in EBS	- Alteration / Degradation / Cracking - Corrosion - Erosion / Dissolution / Spalling [see also Mechanical Impact in 2.1.07.08, Thermal-Mechanical Impact in 2.1.11.09, Chemical Interaction 2.1.09.07]	1	3	3	2
2.1.06.02	Flow Through Liner / Rock Reinforcement Materials in EBS	[see also Flow in EBS in 2.1.08.01]	1	3	2	2
2.1.07.00	1.07. MECHANICAL PROCESSES					
2.1.07.01	Rockfall	- Dynamic loading (block size and velocity)	2 (3: no backfill)	3	2	1.5
2.1.07.02	Drift Collapse	- Static loading (rubble volume) - Alteration of seepage - Alteration of EBS flow pathways - Alteration of EBS thermal environment [see also Evolution of Flow Pathways in EBS in 2.1.08.02, Chemical Effects of Drift Collapse in 2.1.09.12, and Effects of Drift Collapse on TH in 2.1.11.04]	3	3	3	1.5
2.1.07.03	Mechanical Effects of Backfill	- Protection of other EBS components from rockfall / drift collapse	3	3	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.07.04	Mechanical Impact on Backfill	- Rockfall / Drift collapse - Hydrostatic pressure - Internal gas pressure [see also Degradation of Backfill in 2.1.04.01 and Thermal-Mechanical Effects in 2.1.11.08]	3	3	2	2
2.1.07.05	Mechanical Impact on Waste Packages	- Rockfall / Drift collapse - Waste package movement - Hydrostatic pressure - Internal gas pressure - Swelling corrosion products [see also Thermal-Mechanical Effects in 2.1.11.05]	3	2	2	2
2.1.07.06	Mechanical Impact on SNF Waste Form	- Drift collapse - Swelling corrosion products [see also Thermal-Mechanical Effects in 2.1.11.06]	N/A. Not part of current WF IPSC scope.	2	2	2
2.1.07.07	Mechanical Impact on HLW Waste Form	- Drift collapse - Swelling corrosion products [see also Thermal-Mechanical Effects in 2.1.11.06]	2	2	2	2
2.1.07.08	Mechanical Impact on Other EBS Components - Seals - Liner/Rock Reinforcement Materials - Waste Package Support Materials	- Rockfall / Drift collapse - Movement - Hydrostatic pressure - Swelling corrosion products [see also Thermal-Mechanical Effects in 2.1.11.09]	2	2	2	2
2.1.07.09	Mechanical Effects at EBS Component Interfaces	- Component-to-component contact (static or dynamic)	1	2	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.07.10	Mechanical Degradation of EBS	- Floor buckling - Fault displacement - Consolidation of EBS components - Degradation of waste package support structure - Alteration of EBS flow pathways [see also Evolution of Flow Pathways in EBS in 2.1.08.02, Degradation in 2.1.04.01, 2.1.05.01, and 2.1.06.01]	3	2	2	2
2.1.08.00	1.08. HYDROLOGIC PROCESSES					
2.1.08.01	Flow Through the EBS	- Saturated / Unsaturated flow - Preferential flow pathways [see also Flow in Waste Packages in 2.1.03.08, Flow in Backfill in 2.1.04.02], Flow through Seals 2.1.05.02, Flow through Liner in 2.1.06.02, Thermal Effects on Flow in 2.1.11.10, Effects of Gas on Flow in 2.1.12.02]	3	3	2	2
2.1.08.02	Alteration and Evolution of EBS Flow Pathways	- Drift collapse - Degradation/consolidation of EBS components - Plugging of flow pathways - Formation of corrosion products - Water ponding [see also Evolution of Flow Pathways in WPs in 2.1.03.09, Evolution of Backfill in 2.1.04.01, Drift Collapse in 2.1.07.02, and Mechanical Degradation of EBS in 2.1.07.10]	3	2	2	2
2.1.08.03	Condensation Forms in Repository - On Tunnel Roof / Walls - On EBS Components	- Heat transfer (spatial and temporal distribution of temperature and relative humidity) - Dripping [see also Heat generation in EBS in 2.1.11.01, Effects on EBS Thermal Environment in 2.1.11.03 and 2.1.11.04]	1 (3: unsat.)	3	3	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.08.04	Capillary Effects in EBS	- Wicking	1 (3: unsat.)	3	3	2
2.1.08.05	Influx (Seepage) Into the EBS	- Water influx rate (spatial and temporal distribution) [BOUNDARY CONDITION FOR WF IPSC]	3	3	3	2
2.1.09.00	1.09. CHEMICAL PROCESSES - CHEMISTRY					
2.1.09.01	Chemistry of Water Flowing into the Repository	- Chemistry of influent water (spatial and temporal distribution) [BOUNDARY CONDITION FOR WF IPSC]	3	3	3	2
2.1.09.02	Chemical Characteristics of Water in Waste Packages	- Water composition (radionuclides, dissolved species, …) - Initial void chemistry (air / gas) - Water chemistry (pH, ionic strength, pCO2, ..) - Reduction-oxidation potential - Reaction kinetics - Influent chemistry (from tunnels and/or backfill) [see also 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels] - Evolution of water chemistry / interaction with waste packages	3	3	2.5	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.09.03	Chemical Characteristics of Water in Backfill	- Water composition (radionuclides, dissolved species, …) - Water chemistry (pH, ionic strength, pCO2, ..) - Reduction-oxidation potential - Reaction kinetics - Influent chemistry (from tunnels and/or waste package) [see also 2.1.09.02 Chemistry in Waste Packages, 2.1.09.04 Chemistry in Tunnels] - Evolution of water chemistry / interaction with backfill	2.5	3	2.5	2.5
2.1.09.04	Chemical Characteristics of Water in Tunnels	- Water composition (radionuclides, dissolved species, …) - Water chemistry (pH, ionic strength, pCO2, ..) - Reduction-oxidation potential - Reaction kinetics - Influent chemistry (from near-field host rock) [see also 2.1.09.01 Chemistry of Water Flowing in, 2.1.09.02 Chemistry in Waste Packages, 2.1.09.03 Chemistry in Backfill] - Evolution of water chemistry / interaction with seals, liner/rock reinforcement materials, waste package support materials	3	3	2.5	2
2.1.09.05	Chemical Interaction of Water with Corrosion Products- In Waste Packages- In Backfill- In Tunnels	- Corrosion product formation and composition (waste form, waste package internals, waste package)- Evolution of water chemistry in waste packages, in backfill, and in tunnels [contributes to 2.1.09.02 Chemistry in Waste Packages, 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	3	3	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.09.06	Chemical Interaction of Water with Backfill - On Waste Packages - In Backfill - In Tunnels	- Backfill composition and evolution (bentonite, crushed rock, ...) - Evolution of water chemistry in backfill, and in tunnels - Enhanced degradation of waste packages (crevice formation) [contributes to 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels, 2.1.03.04 Localized Corrosion of WPs]	2.5	3	2	2
2.1.09.07	Chemical Interaction of Water with Liner / Rock Reinforcement and Cementitious Materials in EBS - In Backfill - In Tunnels	- Liner composition and evolution (concrete, metal, ...) - Rock reinforcement material composition and evolution (grout, rock bolts, mesh, ...) - Other cementitious materials composition and evolution - Evolution of water chemistry in backfill, and in tunnels [contributes to 2.1.09 .03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	2	3	2	2
2.1.09.08	Chemical Interaction of Water with Other EBS Components - In Waste Packages - In Tunnels	- Seals composition and evolution - Waste Package Support composition and evolution (concrete, metal, ...) - Other EBS components (other metals (copper), ...) - Evolution of water chemistry in backfill, and in tunnels [contributes to 2.1.09 .03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	2.5	3	2	2
2.1.09.09	Chemical Effects at EBS Component Interfaces	- Component-to-component contact (chemical reactions) - Consolidation of EBS components	2	2	2	2
2.1.09.10	Chemical Effects of Waste-Rock Contact	- Waste-to-host rock contact (chemical reactions) - Component-to-host rock contact (chemical reactions)	2.5	3	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.09.11	Electrochemical Effects in EBS	- Enhanced metal corrosion	2 (2.5: metal HLW)	2	2	2
2.1.09.12	Chemical Effects of Drift Collapse	- Evolution of water chemistry in backfill and in tunnels (from altered seepage, from altered thermal-hydrology) [contributes to 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	1	2	2	2
2.1.09.13	Radionuclide Speciation and Solubility in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Dissolved concentration limits - Limited dissolution due to inclusion in secondary phase - Enhanced dissolution due to alpha recoil [controlled by 2.1.09.02 Chemistry in Waste Packages, 2.1.09.03 Chemistry in Backfill, 2.1.09.04 Chemistry in Tunnels]	3	3	2	2
2.1.09.00	1.09. CHEMICAL PROCESSES - TRANSPORT					
2.1.09.14	Advection of Dissolved Radionuclides in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Flow pathways and velocity - Advective properties (porosity, tortuosity) - Saturation [see also Gas Phase Transport in 2.1.12.02]	3	3	2	2
2.1.09.15	Diffusion of Dissolved Radionuclides in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Concentration gradients - Diffusive properties (diffusion coefficients) - Flow pathways and velocity - Saturation	3	3	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.09.16	Sorption of Dissolved Radionuclides in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Sorptive properties (distribution coefficients) - Flow pathways and velocity - Saturation	3	3	2	2
2.1.09.17	Complexation in EBS	- Formation of organic complexants (humates, fulvates, organic waste) - Enhanced transport of radionuclides associated with organic complexants	3	3	2	2
2.1.09.18	Formation of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Formation of intrinsic colloids - Formation of pseudo colloids (host rock fragments, waste form fragments, corrosion products, microbes) - Formation of co-precipitated colloids - Sorption/attachment of radionuclides to colloids (clay, silica, waste form, FeOx, microbes)	3	2	2	2
2.1.09.19	Stability of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Chemical stability of attachment (dependent on water chemistry) - Mechanical stability of colloid (dependent on colloid size, gravitational settling)	3	3	2.5	2
2.1.09.20	Advection of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Flow pathways and velocity - Advective properties (porosity, tortuosity) - Saturation - Colloid concentration	3	2	2	2
2.1.09.21	Diffusion of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Concentration gradients - Diffusive properties (diffusion coefficients) - Flow pathways and velocity - Saturation - Colloid concentration	2	2	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.09.22	Sorption of Colloids in EBS - In Waste Form - In Waste Package - In Backfill - In Tunnel	- Sorptive properties (distribution coefficients) - Flow pathways and velocity - Saturation - Colloid concentration	3	2.5	2	2
2.1.09.23	Sorption of Colloids at Air-Water Interface in EBS		2	2	2	1
2.1.09.24	Filtration of Colloids in EBS	- Physical filtration (dependent on flow pathways, colloid size)- Electrostatic filtration	3	3	2	2
2.1.09.25	Radionuclide Transport Through Seals	- Advection - Diffusion - Sorption	3	3	2	2
2.1.10.00	1.10. BIOLOGICAL PROCESSES					
2.1.10.01	Microbial Activity in EBS - Natural - Anthropogenic	- Effects on corrosion - Formation of complexants - Formation of microbial colloids - Formation of biofilms - Biodegradation - Biomass production - Bioaccumulation [see also Microbiallly Influenced Corrosion in 2.1.03.06, Complexation in EBS in 2.1.09.17, Radiological Mutation of Microbes in 2.1.13.03]	3 (1.5: unsat)	2	2	2
2.1.11.00	1.11. THERMAL PROCESSES					
2.1.11.01	Heat Generation in EBS	- Heat transfer (spatial and temporal distribution of temperature and relative humidity) [see also Waste Inventory in 2.1.01.01]	3	3	3	2
2.1.11.02	Exothermic Reactions in EBS		1	3	3	3
2.1.11.03	Effects of Backfill on EBS Thermal Environment	- Thermal blanket - Condensation	3	3	2	2
2.1.11.04	Effects of Drift Collapse on EBS Thermal Environment	- Thermal blanket - Condensation	3	2	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.11.05	Effects of Influx (Seepage) on Thermal Environment	- Temperature and relative humidity (spatial and temporal distribution) [BOUNDARY CONDITION FOR WF IPSC]	3	2	2	1.5
2.1.11.06	Thermal-Mechanical Effects on Waste Form and In-Package EBS Components	- Alteration - Cracking - Thermal expansion / stress	2.5	2	2	2
2.1.11.07	Thermal-Mechanical Effects on Waste Packages	- Thermal sensitization / phase changes - Cracking - Thermal expansion / stress / creep	2.5	2	2	2
2.1.11.08	Thermal-Mechanical Effects on Backfill	- Alteration - Cracking - Thermal expansion / stress	3	2	2	2
2.1.11.09	Thermal-Mechanical Effects on Other EBS Components - Seals - Liner / Rock Reinforcement Materials - Waste Package Support Structure	- Alteration - Cracking - Thermal expansion / stress	2	2	2	2
2.1.11.10	Thermal Effects on Flow in EBS	- Altered saturation / relative humidity - Condensation	3	2.5	2	2
2.1.11.11	Thermally-Driven Flow (Convection) in EBS	- Convection	3	3	2	2
2.1.11.12	Thermally-Driven Buoyant Flow / Heat Pipes		3	3	2	2
2.1.11.13	Thermal Effects on Chemistry and Microbial Activity in EBS		3	3	2	2
2.1.11.14	Thermal Effects on Transport in EBS	- Thermal diffusion (Soret effect) - Thermal osmosis	1	3	3	3

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.12.00	1.12. GAS SOURCES AND EFFECTS					
2.1.12.01	Gas Generation in EBS	- Repository Pressurization - Mechanical Damage to EBS Components - He generation from waste from alpha decay - H2 generation from waste package corrosion - CO2, CH4, and H2S generation from microbial degradation	3 (1: unsat.)	3	3	2
2.1.12.02	Effects of Gas on Flow Through the EBS	- Two-phase flow - Gas bubbles [see also Two-Phase Buoyant Flow in 2.1.11.12]	3	3	3	2
2.1.12.03	Gas Transport in EBS	- Gas phase transport - Gas phase release from EBS	3	3	2	2
2.1.12.04	Gas Explosions in EBS		1	2	2	1
2.1.13.00	1.13. RADIATION EFFECTS					
2.1.13.01	Radiolysis - In Waste Package - In Backfill - In Tunnel	- Gas generation - Altered water chemistry	2.5	3	2	2
2.1.13.02	Radiation Damage to EBS Components - Waste Form - Waste Package - Backfill - Other EBS Components	- Enhanced waste form degradation - Enhanced waste package degradation - Enhanced backfill degradation - Enhanced degradation of other EBS components (liner/rock reinforcement materials, seals, waste support structure)	2	2	2	2
2.1.13.03	Radiological Mutation of Microbes		1	1	1	1
2.1.14.00	1.14. NUCLEAR CRITICALITY					
2.1.14.01	Criticality In-Package	- Formation of critical configuration	3	3	2	2

Phenomena Number	Phenomena	Associated Processes	Importance (3=High, 2=Med, 1=Low)	State of Knowledge - Model	State of Knowledge - Data	Likelihood of New Info
2.1.14.02	Criticality in EBS or Near-Field	- Formation of critical configuration	2.5	3	2	2
2.2.00.00	2. GEOLOGICAL ENVIRONMENT					
2.3.00.00	3. SURFACE ENVIRONMENT					
2.4.00.00	4. HUMAN BEHAVIOR					
3.0.00.00	RADIONUCLIDE/CONTAMINANT FACTORS (BIOSPHERE)					
3.1.00.00	1. CONTAMINANT CHARACTERISTICS					
3.2.00.00	2. RELEASE/MIGRATION FACTORS					
3.3.00.00	3. EXPOSURE FACTORS					

Appendix B

B. Detailed Use Cases

Use Cases for a Performance Assessment System (Surrogate Models)

SYSTEM-LEVEL USE CASES

UC #S1: LOGIN
Purpose: Set up access permissions for various categories of users.
Actors: User
Steps:
1. User starts the application by providing his/her username and password.
2. The system verifies the information.
3. The use selects a function as needed.
4. The system performs the function selected

User interface: GUI

UC #S2: DEFINING SIMULATION DOMAINS FOR A DISPOSAL SYSTEM
Purpose: Define simulation domains for a disposal system (e.g., waste panels or drifts). Each simulation domain can be further meshed into sub-cells. Each domain can be associated with one or more meshes, for example, a fine mesh for flow field calculations and a coarse mesh for chemical reaction simulations. In this case, a grid interpolation or extrapolation is needed for transferring data between the two meshes.
Actors: User
Steps:
1. The user selects the function defining simulation domains for the disposal system.
2. The user adds a physical domain to the screen.
3. The user specifies the geometry (e.g. shape, volume) of the domain.
4. The user specifies whether the domain needs to be meshed.
5. The system meshes the domain as the user has specified.
6. The user repeats steps 2 through 5 until all domains are specified.
7. The system displays the topologic layout of the disposal system.

User interface: GUI

UC #S3: DEFINING TYPES OF WASTE FORMS
Purpose: Define the types of waste forms.
Actors: User
Steps:
1. The user selects the function defining waste forms.
2. The user adds a new waste form.
3. The user specifies basic physical/chemical properties of the waste form (e.g., density, chemical composition).
4. The user repeats steps 2 through 3 until all waste form are included.

User interface: GUI

UC #S4: DEFINING TYPES OF WASTE CONTAINERS & THEIR DISTRIBUTIONS
Purpose: Define the types of waste containers and their distributions in the disposal system.
Actors: User
Steps:
1. The user selects the function defining waste container types.

2. The user adds a new type of waste containers.
3. The user specifies the amount (volume or mass) of each waste form inside a waste container of the type added.
4. The user repeats steps 2 through 3 until all types of waste containers are included.
5. The user specifies the distributions of waste containers in the disposal system, i.e., the numbers of each specific type containers in each simulation domain of the disposal system.

User interface: GUI

UC #S5: DEFINING SIMULATION DOMAINS FOR ENGINEERED BARRIERS
Purpose: Define simulation domains for engineered barriers (e.g., waste containers or backfill materials). Each simulation domain can be further meshed into sub-cells as needed.
Actors: User
Steps:

1. The user selects the function defining simulation domains for waste containers (e.g., layers of waste package materials).
2. The user selects the type of waste containers.
3. The user adds a physical domain to the selected container type.
4. The user specifies the geometry (e.g. shape, volume) of the domain.
5. The user specifies whether the domain needs to be meshed.
6. The system meshes the domain as the user has specified.
7. The user repeats steps 2 through 6 until all domains are specified.
8. The user repeats steps 2 through 7 until all types of waste containers are considered.
9. The user selects the function defining simulation domains for other components of the engineered barrier system (e.g., backfill, invert, etc.).
10. The user adds a physical domain to the engineered barrier system.
11. The user specifies the geometry (e.g. shape, volume) of the domain.
12. The user specifies whether the domain needs to be meshed.
13. The system meshes the domain as the user has specified.
14. The user repeats steps 10 through 13 until all domains are specified.
15. The system displays the topologic layout of the engineered barrier system.

User interface: GUI

UC #S6: DEFINING STATE VARIABLES
Purpose: Define state variables of each simulation domain. The physical and chemical states of a simulation domain can be described by a set of state variables (e.g., temperature, pressure, pH, mass, etc.)
Actors: User
Steps:

1. The user selects the function defining state variables.
2. The user adds a state variable to the list. A state variable can be a scalar variable, a vector or even a tensor. For example, the concentrations of all radionuclides in an aqueous solution constitute a vector.
3. The user repeats step 2 until all relevant state variables are defined.
4. The user selects the function of displaying simulation domains.
5. The user associates each domain with appropriate state variables.

User interface: GUI

UC #S7: MODEL LINKAGE
Purpose: Associate each simulation domain with a set of model operations (i.e., process models or code module).
Actors: User
Steps:

1. The user selects the function displaying simulation domains.
2. The user highlights a group of simulation domains.
3. The user selects a set of process models/basic operations (e.g., a chemical equilibrium calculation, a time delay operation, etc.).
4. The user links the models with unidirectional arrows indicating data flows through the association of input and output model parameters. The user can add a time step shift to break a full loop linkage.
5. The user associates the simulation domains with the process models.
6. The user links the simulation domains by specifying appropriate boundary conditions, either the first type or the second type. A boundary condition is specified by linking a state variable in one domain to that in another domain or simply to a model input parameter, which the user can introduce as needed during model linkage.

User interface: GUI

UC #S8: MAKING MODEL FILE
Purpose: Initialize a model system and make a model file. A model file includes both the information on the model system and the associated input parameter values.
Actors: User
Steps:

1. The user selects a linked model system.
2. The system displays a complete list of input parameters. Input parameters are identified as the parameters that are not calculated within the model system.
3. The user separate aleatory uncertain parameter from epistemic uncertain parameters.
4. The user graphically links the input parameters to the parameter values (or distributions) in the database or manually specifies the parameter values (or distributions). The user can also impose correlation between two input parameters.
5. The system warns if a specified parameter value is outside the validated range of the parameter. The validated range of a model parameter is specified in UC #S18.
6. The system saves a record of the parameters used as inputs or identifies the parameters in the database using the scenario identifier and provides time-date stamping in the record.

User interface: GUI

UC #S9: MULTIPLE MODEL REALIZATIONS
Purpose: Perform multiple realizations for a given scenario.
Actors: User
Steps:

1. The user loads a model file.
2. The user specifies the simulation environment (time duration, time step, number of realizations, etc.).
3. The user specifies which intermediate results need to be saved.
4. The system samples epistemic uncertain parameters upfront of each realization or reads a table of sampled values.
5. The system runs each realization with aleatory parameters sampled or specified.
6. The system displays the status of the simulation.
7. The system checks mass conservation across all simulation domains.
8. The system checks if any model parameter has the calculated value outside its validated range. The validated ranges of model parameters are specified in UC #S15.
9. The system saves the simulation results in the database with a unique version identifier indicating a specific data-model association.

User interface: GUI. The system displays the progress of the simulation.

UC #S10: SINGLE REALIZATION SIMULATIONS
Purpose: Run single realization calculations by using the means, medians, specified constants (or flow fields), or specified percentile values for the model parameters.
Actors: User
Steps:
1. The user loads a model file.
2. The user specifies values (e.g., the means, medians, specified constants, or specified percentile values) for uncertain parameters.
3. The user specifies the simulation environment (time duration, time step, etc.).
4. The user specifies which intermediate results need to be saved.
5. The system runs the simulation.
6. The system saves the simulation result with the model file.

User interface: GUI. The system displays the progress of the simulation.

UC #S11: UNCERTANITY QUANTITIFICATION
Purpose: Perform uncertainty analyses and construct statistical results for regulatory compliance.
Actors: User
Steps:
1. The user loads model files for one or more modeling scenarios.
2. The system runs multiple realizations (UC #S9)
3. The user selects the output variables for uncertainty quantification.
4. The system displays horsetail plots for each selected variable for each time-dependent output variable for each scenario.
5. The system calculates and displays the means and the percentiles of the variables.
6. The system keeps traceability from a single realization back to the input parameters.
7. The system synthesizes and displays the composite results for all the scenarios.
8. The user can export the results as a text file that contains time/date stamping and traceability to the input files.

User interface: GUI

UC #S12: SENSITIVITY ANALYSIS
Purpose: Identify important parameters that control total system performance.
Actors: User
Steps:
1. The user loads a model file.
2. The user selects a set of output variables.
3. The user selects a set of uncertain input parameters.
4. The system calculates correlation coefficients (e.g., stepwise regression correlation coefficients, partial rank correlation coefficients, etc.) between the output variables and the input parameters. The sensitivity analysis also uses other methods including stepwise regression and partial correlation coefficients.
5. The system displays the correlations between any pair of variables (e.g., using scattering plots).

User interface: GUI

UC #S13: OPTIMIZATION/DATA FITTING
Purpose: Determine model parameters by fitting the model to experimental data. One application of this UC is to constrain waste degradation parameters from waste form leaching experiments.
Actors: User
Steps:
1. The user sets up a modeling system (e.g., a flow through column) as suggested in UC #2 – UC #8.

2. The user specifies the parameters to be fitted.
3. The user imports the experimental data.
4. The system fits the model to the experimental data.
5. The system outputs and displays the fitting results.

User interface: GUI. The system displays in real time the fitting process.

UC #S14: COMPARING SURROGATE MODEL WITH HI-FI MODEL
Purpose: Determine the uncertainty related to the surrogate model abstraction and simplification.
Actors: User
Steps:

1. The user chooses and runs a surrogate model.
2. The user imports the corresponding hi-fi model results.
3. The system compares the results of both models.
4. The system determines the accuracy of the surrogate model relative to the hi-fi model.
5. The system displays the graphical comparison between the surrogate model and the hi-fi model in terms of model predictions.

User interface: GUI.

UC #S15: MODEL VERIFICATION & VALIDATION
Purpose: Verify and validate a code module or a linked set of code modules against a set of testing cases.
Actors: User
Steps:

1. The user makes a model file by linking a selected set of code modules (UC #S2-S6).
2. The user specifies the expected model result (i.e., the data that are not used to constrain model parameters).
3. The system runs the model file.
4. The system displays both the model result and the expected result.
5. The user decides whether the testing is successful.
6. If successful, the system saves all testing information.

User interface: GUI.

UC #S16: REGRESSION TESTS
Purpose: Perform regression tests against a set of established testing cases. This use case is needed, for example, for operating system changes.
Actors: User
Steps:

1. The user loads the model files of the established testing cases.
2. The system runs all the testing cases.
3. The system compares the results with the previous results.
4. The system displays the difference if there any.
5. The user decides whether the testing is successful.
6. If successful, the system saves all testing information.

User interface: GUI.

UC #S17: VISUALIZATION
Purpose: Visualize the temporal evolution of the state of each individual simulation domain the user selects.
Actors: User
Steps:

1. The user selects a model file from the file archive.
2. The user uploads the model file and the associated model result.

3. The user selects the state variables for visualization.
4. The user specifies the graphic representations of the initial state and the end state of each variable.
5. The user specifies the settings for visualization (the speed for displaying each time step).
6. The system graphically displays the states of simulation domains the user selects.
7. The user pauses and reverses the visualization as needed.

User interface: GUI.

UC #S18: SPECIFYING VALIDATED RANGES OF MODEL PARAMETERS
Purpose: Specify the validated ranges of individual model parameter in each code module (sub-process model). The system warns if any model parameter has its input or calculated value outside its validated range.
Actors: User **(**code developer)
Steps:

1. The user selects a specific code module he or she has developed.
2. The system displays the list of model input parameters of the code module.
3. The user specifies the validated range for each input model parameter.
4. The system saves the specified ranges for each code module.

User interface: GUI.

UC #S19: UNIT CONVERSION
Purpose: The system performs automatic unit conversions for data transfer between two code modules.
Actors: User/system
Steps:

1. The user specifies the units of model parameters.
2. The system automatically converts units for data transfer between two code modules.

User interface: GUI.

UC #S20: UNIFIED DATA FORMAT
Purpose: A data file may be transferred from a Hi-Fi model to a surrogate model. To facilitate data transfer, a unified data format must be enforced. This unified data format also facilitates post-processing of model simulation results.
Actors: User
Steps:

1. The system provides a selection of unified data formats.
2. The user selects a data format for a specific set of model outputs.
3. The system saves the model outputs to a file in the selected data format.

User interface: GUI.

UC #S21: RUNNING AN INDEPENDENT CODE
Purpose: An independent code is a self-contained code with its own input and output formats. Such a code can be an alternative model that needs to be evaluated for a performance assessment. In this case, there is no intention to intrusively modify the original source code. Therefore, a graphic interface must be provided to wrap and execute the code.
Actors: User
Steps:

1. The user specifies the list of code input parameters in a database.
2. The user specifies the information on simulation domain meshing as in UC #S2 and UC #5.
3. The user imports the template of the input file of the code.
4. The user graphically relates the input parameters in the database to the text of the template of the input file.

5. The system generates an input file for the code.
6. The user specifies the list of model output parameters in the database.
7. The user imports the template of the output file of the code.
8. The user relates the output parameters to the text of the template of the output file.
9. The system executes the code and saves the output file as a temporary file.
10. The system extracts the output parameter values from the output file and save them to the database.

User interface: GUI.

UC #S22: BATCH PROCESSING OF MULTIPLE INDEPENDENT CODES
Purpose: Run multiple independent codes sequentially according a specified data flow among them. Unlike use case UC #S8, where each code modules are linked at each time step, each independent code is run for a whole simulation time period and data transfer takes place only at the end of each simulation through a database.
Actors: User
Steps:

1. The user wraps each individual code as in UC #21.
2. The user specifies the data flow among the codes by associating the code input and output parameters in the database. The data flow is limited to be unidirectional, and all data are transferred through a database.
3. The system executes the code sequentially according to the user-specified data flow.

User interface: GUI.

SUBSYSTEM-LEVEL USE CASES:

BASIC OPERATIONS

UC #B1: MATHEMATICAL FUNCTIONS
Purpose: The subsystem "Basic Operations" provides basic mathematical functions for model linking in UC #S4.
Actors: System
Steps:

1. The system invokes the mathematical functions as specified in a model file during simulations. The basic operation includes numerical integration or averaging of a variable over a space or a time interval.

User interface: None.

UC #B2: TIME-DELAY FUNCTION
Purpose: The subsystem "Basic Operations" provides a time-delay function to break a full loop coupling among code modules within one time step as needed.
Actors: System
Steps:

1. The system takes the previous time step values from the upstream code module.
2. The system feeds these values to the downstream code modules.

User interface: None.

UC #B3: DATA TRANSFER BETWEEN TWO MESHES
Purpose: Transfer data from one mesh to another. The surrogate model system allows sub-process models to run on different spatial grids. For example, a flow model is run on a fine grid while a chemical model is run on a coarse grid; and a data transfer between the two meshes is thus required.

Actors: System
Steps:
1. The system transfers the data from one mesh to another through interpolation or averaging.

User interface: None.

THERMAL PROCESSES

UC #T1: THERMAL OUTPUT OF WASTE FORM

Purpose: Calculate the thermal output of a given volume of waste form as a function of radionuclide inventory and burn up.
Actors: System
Steps:
1. The system imports radionuclide inventory from the previous time step.
2. The system imports radionuclide decay parameters from the database.
3. The system calculates the thermal output of waste form for the next time step.

User interface: None.

UC #T2: HEAT TRANSFER

Purpose: Calculate the heat transfer and temperature distribution in a simulation domain at each time step.
Actors: System
Steps:
1. The system updates thermal properties of the relevant materials in a simulation domain.
2. The system imports the physical configuration of materials.
3. The system imports the fluid flow field;
4. The system calculates the heat transfer through conduction, convection, and radiation.
5. The system calculates the corresponding temperature distribution through the domain.

User interface: None

UC #T3: THERAML IMPACTS OF VOLCANIC FLOWS

Purpose: Simulate the thermal impacts of volcanic flows on waste forms and waste containers.
Actors: System
Steps:
1. The system imports the dynamic properties and boundary conditions of a volcanic flow.
2. The system imports the thermal properties of waste forms and container materials.
3. The system calculates the number of waste container affected by volcanic flows.
4. The system calculates the damage of the container and the waste forms caused by a volcanic thermal event. The damage could be caused by thermal stress and mineral phase transition.

User interface: None

HYDROLOGIC/TRANSPORT PROCESSES

UC #H1: MULTIPHASE FLOW IN POROUS MEDIA

Purpose: Provide the flow fields and pressure distribution for calculating advective radionuclide transport in porous media including fractured porous media
Actors: System
Steps:
1. The system selects a specific flow model (e.g., Darcy flow, two-phase flow, etc.) as the user specified in the model file.

2. From the previous time step, the system imports the hydrologic properties and boundary conditions for the simulation domain to be modeled.
3. From the previous time step, the system imports the total mass, the flow field, and temperature distribution.
4. The system updates the flow fields and the pressure distribution for the current time step by solving a set of multiphase (liquid and gas) flow equations or simply importing a pre-generated response surface.

User interface: None

UC #H2: MULTIPHASE FLOW IN OPEN CHANNELS
Purpose: Provide the flow fields and pressure distribution for calculating radionuclide transport in open channels. This module can apply to a flow in an open borehole or a waste emplacement drift.
Actors: System
Steps:

1. From the previous time step, the system imports the hydrologic properties and boundary conditions for the simulation domain to be modeled.
2. From the previous time step, the system imports the total mass, the flow field.
3. The system updates the flow field and the pressure distribution for the current time step by solving a set of multiphase flow equations or simply importing a pre-generated response surface.

User interface: None

UC #H3: TRANSPORT OF DISSOLVED/GASEOUS CHEMICAL COMPONENTS
Purpose: Simulate the transport of dissolved/gaseous chemical components (including radionuclides) across a simulation domain.
Actors: System
Steps:

1. The system imports the flow field.
2. The system updates the mass of each dissolved/gaseous chemical component in the simulation domain. The relevant processes include molecular diffusion, mechanical dispersion, flow advection, and chemical sorption onto rock matrix.

User interface: None

UC #H4: TRANSPORT OF COLLOIDS
Purpose: Simulate colloid transport across simulation domains.
Actors: System
Steps:

1. The system imports the flow field.
2. The system updates colloid particle distributions in each simulation domain. The relevant processes include diffusive and advective transport, colloid particle attachment/detachment, and colloid filtration in porous media, and colloid stability.

User interface: None

UC #H5: COLLOID-FACILITATED RADIONUCLIDE TRANSPORT
Purpose: Simulate the transport of radionuclides attached to colloidal particles. This use case can be combined with UC #H3.
Actors: System
Steps:

1. The system imports the flow field.
2. The system imports the coefficients of radionuclide partitioning between colloids and solution.
3. The system imports the flux of colloid movement.
4. The system calculates radionuclide transport by colloid particles.

User interface: None

UC #H6: CAVING EFFECT
Purpose: Simulate the radionuclide transport through caving around a borehole.
Actors: System
Steps:
1. The system imports the flow field and the properties of the fluid (e.g., viscosity).
2. The system imports the physical properties of degraded wastes (e.g., particle size distribution).
3. The system calculates the volume of waste particles entrained by the flow in a borehole.

User interface: None

MECHANICAL PROCESSES

UC #M1: SALT CREEP
Purpose: Simulate salt creep around a disposal system. Salt creep may enhance encapsulation and isolation of waste forms emplaced in the repository.
Actors: System
Steps:
1. The system imports salt creep mechanical properties.
2. The system imports temperature and moisture distribution in the domain to be simulated.
3. The system calculates salt deformation and creeping as a function of time and space.

User interface: None

UC #M2: FORMATION OF DISTURBED ROCK ZONE (DRZ)
Purpose: Simulate the formation of DRZ around a waste disposal room and its impact on mechanical and hydrologic properties. Rock fall may cause fracturing in waste containers. Rock permeability may also change as fractures open/close in DRZ.
Actors: System
Steps:
1. The system imports rock mechanical properties.
2. The system imports boundary conditions.
3. The system simulates rock falls if needed.
4. The system calculates the changes in mechanical, hydrologic, and thermal property (e.g., porosity and permeability) due to the formation of DRZ.

User interface: None

UC #M3: MECHANICAL DAMAGE WASTE CONTAINERS BY SALT CREEP
Purpose: Stress salt exerts on waste containers may cause failure of the containers. In this use case, the system calculates the failure rate and the area of failure openings of a waste container as a function of salt creep.
Actors: System
Steps:
1. The system imports mechanical properties of waste container materials.
2. The system imports the extent and the geometry of salt creep from UC #M1.
3. The system calculates the failure rate and the area of failure openings of a waste container as controlled by salt creep.

User interface: None

UC #M4: SEISMIC DAMAGE OF WASTE CONTAINERS

Purpose: Calculate the failure rate and the area of failure openings of a waste container as controlled by rock falls and seismic ground motion. The mechanisms for failure include stress corrosion.
Actors: System
Steps:
1. The system imports mechanical properties of waste container materials.
2. The system imports the model parameters for rock falls or seismic ground motion.
3. The system calculates the failure rate and the area of failure openings of a waste container as controlled by seismic ground motion.

User interface: None

UC #M5: CLAD UNZIPPING
Purpose: Clad unzipping due to the volumetric expansion of degraded fuel inside the clad sleeves causes further exposure of waste forms to disposal environments. In this use case, the system calculates the area of failure opening as the inside waste form degrades.
Actors: System
Steps:
1. The system imports mechanical properties of waste container materials.
2. The system calculates the solid volume change inside the clad.
3. The system calculates the stress created volume expansion of waste form corrosion products.
4. Calculate the area of clad unzipping.

User interface: None

UC #M6: EXPANSION OF CLAY MATERIALS
Purpose: Simulate the expansion of clay materials as a physical barrier. The porosity and permeability of clay may change as it expands.
Actors: System
Steps:
1. The system imports the boundary conditions for the clay barrier.
2. The system imports pore-water compositions (e.g., ionic strength, pH).
3. The system calculates the expansion of the clay material and the pressure created due to the volume expansion as a function of temperature and pore water chemistry.
4. The system calculates the porosity and permeability changes of the material.

User interface: None

UC #M7: WASTE FORM INTERACTIONS WITH VOLCANIC FLOWS
Purpose: Simulate the mechanical interactions waste forms and waste containers with volcanic flows.
Actors: System
Steps:
5. The system imports the dynamic properties and boundary conditions of a volcanic flow.
6. The system imports the mechanical properties of waste forms and container materials.
7. The system calculates the portion of waste forms entrained by the volcanic flow.
8. The system calculates the amount of fine waste particles generated by the volcanic flow.

User interface: None

CHEMICAL PROCESSES

UC #C1: KINETICS OF WASTE FORM DEGRADATION
Purpose: Calculate the degradation rate of each type of waste form (e.g., ceramic, glass, etc) as a function of environmental parameters.
Actors: System

Steps:
1. The system imports the type of waste form and the related kinetic parameters for waste form degradation.
2. The system imports the environmental parameters (e.g., temperature, pH, etc.).
3. The system calculates the quantity of waste form degraded at each time step. The relevant processes include waste form dissolution, secondary mineral precipitation, and diffusion across possible coating layers. A chemical affinity term should be accounted for.
4. The system updates the quantities of waste form remaining and corrosion products produced.

User interface: None

UC #C2: KINETICS OF HOST ROCK MINERAL DISSOLUTION
Purpose: Calculate the dissolution rates of minerals in the host rock.
Actors: System
Steps:
1. The system imports the name of a mineral of interest and the related kinetic parameters for dissolution.
2. The system imports the environmental parameters (e.g., temperature, pH, etc.).
3. The system calculates the quantity of mineral dissolved at each time step. The relevant processes include mineral dissolution/precipitation. A chemical affinity term should be accounted for.

User interface: None

UC #C3: CHEMICAL EQUILIBRIUM CALCULATION
Purpose: Calculate a chemical equilibrium at given total mass, temperature and pressure.
Actors: System
Steps:
1. The system imports environmental parameters.
2. The system imports the equilibrium constants for relevant chemical reactions.
3. The system imports the total mass for each chemical component.
4. The system calculates the concentrations and activity coefficients of relevant chemical species (dissolved, gaseous, and solid) according to appropriate models based on bulk solution chemistry. The calculation must be performed on basis of chemical elements.

User interface: None

UC #C4: PARTIAL CHEMICAL EQUILIBRIUM CALCULATION
Purpose: Calculate a partial chemical equilibrium at given total mass, temperature and pressure. This is an alternative use case to use cases UC #C1 through UC #C3. In this use case, the chemical equilibrium calculation is fully coupled with mineral/waste form dissolution calculation through numerical iterations.
Actors: System
Steps:
1. The system imports environmental parameters.
2. The system imports the equilibrium constants for relevant chemical reactions.
3. The system calculates the amounts of waste forms/rock-forming minerals dissolved as suggested in UC # C1 and UC #C2.
4. The system calculates the concentrations and activity coefficients of relevant chemical species (dissolved, gaseous, and solid) according to appropriate models based on bulk solution chemistry. The calculation must be performed on basis of chemical elements.

User interface: None

UC #C5: REACTIVE TRANSPORT
Purpose: Solve reactive transport for all relevant chemical components in a simulation domain. The reactive transport model is solved using the implicit scheme to avoid possible numerical instability that

may occur if the model is solved explicitly by sequentially applying use cases UC #H3, UC #C1, UC #C2 and UC #C3. The model calculation for this use case is limited to 1-D or 2-D reactive transport.
Actors: System
Steps:

1. The system imports environmental parameters, the flow field, and the equilibrium constants for relevant chemical reactions.
2. The system solves transport equations and updates the mass of each dissolved/gaseous chemical component in the simulation domain.
3. The system calculates the amounts of waste forms/rock-forming minerals dissolved as suggested in UC # C1 and UC #C2.
4. The system calculates the concentrations and activity coefficients of relevant chemical species (dissolved, gaseous, and solid) as suggested in UC #3.
5. The system iterates steps 2 through 4 until the concentrations of all species converge within a precision specified by the user.
6. The system updates the contraction distributions in the domain.

User interface: None

UC #C6: SURFACE SORPTION
Purpose: Calculate radionuclide partitioning between an aqueous solution and a solid surfaces available for sorption.
Actors: System
Steps:

1. The system imports environmental parameters.
2. The system imports the surface properties for each solid.
3. The system imports aqueous speciation information (e.g., pH and ionic strength).
4. The system imports the equilibrium constants for relevant surface reactions.
5. The system calculates the amount of each radionuclide adsorbed to each surface.

User interface: None

UC #C7: SOLID SOLUTION
Purpose: Calculate the amount of a radionuclide incorporated in secondary mineral structures.
Actors: System
Steps:

1. The system imports environmental parameters as needed.
2. The system imports aqueous speciation information as needed.
3. The system imports the partitioning coefficient of a radionuclide.
4. The system calculates the amount of the radionuclide incorporated.

User interface: None

UC #C8: PRESSURE SOLUTION
Purpose: Calculate the solubility of a mineral as a function of the stress applied.
Actors: System
Steps:

1. The system imports environmental parameters (e.g., stress, temperature, moisture).
2. Calculate the solubility of a mineral as a function of the stress applied.

User interface: None

UC #C9: COLLOID STABILITY
Purpose: Evaluate the stability of a colloidal suspension.
Actors: System
Steps:

1. The system imports environmental parameters (e.g., pH and ionic strength).
2. The system evaluates the stability of a colloidal suspension.

User interface: None

UC #C10: MICROBIAL REACTIONS
Purpose: Evaluate the consumption and production of constituents by microbial reactions.
Actors: System
Steps:

1. The system imports environmental parameters (e.g., temperature, pH, etc.)
2. The system imports nutrient information.
3. The system determines appropriate reaction pathways (e.g., aerobic respiration, denitrification, Mn reduction, Fe reduction, sulfate reduction and methanogensis).
4. The system calculates the progress of each reaction pathway.
5. The system calculates the amount of nutrients consumed.
6. The system calculates the amount of gas generated.
7. The system calculates the amount of organic acids produced.
8. The system calculates the amount of biomass produced.
9. The system calculates the amount of microbial colloids produced.
10. The system calculates the stability of colloidal suspensions.

User interface: None

UC #C11: RADIOLYSIS
Purpose: Calculate the amounts of chemical species generated by radiolysis.
Actors: System
Steps:

1. The system imports the quantity and geometry of each radiolytic material including water.
2. The system imports the G values for each material and each type of radiation (gamma, beta, alpha).
3. The system imports the rate constants for non-radiolytic reactions.
4. The system calculates the energy deposited on each material.
5. The system calculates the production of chemical species by radiolysis.

User interface: None

UC #C12: METAL CORROSION
Purpose: Calculate the amount of metal corroded and the quantities of corrosion products generated. The relevant metallic materials include metallic waste forms, waste container materials, and other introduced metals.
Actors: System
Steps:

1. The system imports environmental parameters (e.g., pH, ionic strength, etc.).
2. The system calculates the quantity of metal corroded at each time step through both localized and general corrosion mechanisms. The relevant processes include dissolution, oxidation, secondary mineral precipitation, diffusion across possible coating layers, possible Galvanic effect, and microbially influenced corrosion.
3. The system updates the quantities of metal remaining and corrosion products produced
4. The system calculates the amounts of hydrogen gas generated and oxygen gas and water consumed.

User interface: None

UC #C13: WASTE PACKAGE FAILURE DUE TO CORROSION

Purpose: Calculate the amount of metal corroded and the quantities of corrosion products generated. The relevant metallic materials include metallic waste forms, waste container materials, and other introduced metals.
Actors: System
Steps:

1. The system imports environmental parameters (e.g., pH, ionic strength, etc.).
2. The system imports the corrosion properties and physical configuration data of container materials.
3. The system uses the corrosion model developed for UC #C12 to estimate the extents of both general corrosion and localized corrosion.
4. The system calculates the rate of container failure and the area of failure openings.

User interface: None

UC #C14: EFFECT OF RADIATION DAMAGE
Purpose: Predict the effect of radiation damage on material stability and durability.
Actors: System
Steps:

1. The system calculates the accumulated dose of each radiation.
2. The system calculates the radiation-induced changes in mineral stability and dissolution kinetics.

User interface: None

UC #C15: RADIOACTIVE DECAY & INGROWTH
Purpose: Track radioactive decay and ingrowth of isotopes.
Actors: System
Steps:

1. The system imports of the information on half lives of radionuclides and the related decay chains.
2. The system updates the total mass of each isotope.

User interface: None

DATABASES

UC #D1: LOGIN
Purpose: Set up access permissions for various categories of users.
Actors: User
Steps:
1. The user starts the application by providing his/her username and password.
2. The system verifies the information.
3. The user selects a function as needed.
4. The system performs the function selected.
5. The user case ends.

User interface: GUI

UC #D2: DATA INPUT/UPDATE
Purpose: Add and update any entries in a controlled manner.
Actors: User
Steps:
1. The user logs in the database system.
2. The system displays all the entries in the database as permitted.
3. The user browses and selects an entry. An entry represents a one model parameter. The user can associate one model parameter with multiple values or distributions.
4. Or the user adds/updates a model parameter value.
5. For each parameter value entered, the user provides the supporting information as needed.
6. The user specifies the quality level of the datum.
7. The system tracks any changes made to the database.

User interface: GUI
UC #D3: DATA INQUIRY/DOWNLOAD
Purpose: Inquire the data (both input and output data) and download them in appropriate formats.
Actors: User
Steps:
1. The User logs in the database system.
2. The system displays all the entries in the database as permitted.
3. The user browsers and selects entries.
4. The user specifies a template of the format for data downloading.
5. The system downloads the data.

User interface: GUI

UC #D4: THERMODYNAMIC/KINETIC DATA ANALYSIS
Purpose: Estimate thermodynamic/kinetic parameters (e.g., activity coefficients of aqueous species) for elevated temperature and pressure and high ionic strength conditions.
Actors: User
Steps:
1. The user logs in the database system.
2. The user selects a thermodynamic/kinetic parameter and calculation scheme.
3. The system interpolates or extrapolates the parameter value to the conditions the user specifies according to a specific modeling scheme.
4. The system outputs or saves the result.

User interface: GUI

UC #D5: CONSTRUCTION OF PARAMETER DISTRIBUTION
Purpose: Construct a parameter value distribution from experimental data.

Actors: User
Steps:
1. The user logs in the database system.
2. The user specifies the model parameter to be constrained.
3. The user inputs experimental data.
4. The system calculates the statistical parameters of the data and suggests a possible distribution that best fits the data.
5. The user selects the distribution. The user can also choose an empirical distribution.
6. The system displays both the data and the theoretical distribution.
7. Repeat steps 4 and 5 until a satisfied data distribution is obtained.
8. The system saves the result.

User interface: GUI

UC #D6: MODEL-DATA LINKAGE
Purpose: Link model parameters to the data in the database for UC #S5.
Actors: System
Steps:
1. The system imports data from the database.
2. The system associates the data with model input parameters specified in a model file.

User interface: None

UC #D7: ARCHIVING SIMULATION RESULT
Purpose: Save a model simulation result.
Actors: System
Steps:
1. The system saves the model result after each simulation completes.
2. Associated with the result, the system also saves the model file and the input parameter values to ensure the transparency, traceability, reproducibility, and retrievability of simulation results.

User interface: None

Use Cases for Continuum Models (High-Fidelity Models)

THERMAL-HYDROLOGICAL PROCESSES

CUC #TH1: Flow, Heat, and Radionuclide Transport
Purpose: Simulate heat, gravity, production/injection well, and precipitation driven flow and transport including appropriate boundary conditions.
Actors: System thermal-hydrologic code(s) that a) can be either Lagrangian or Eulerian; b) simulates either porous media flow, Navier-Stokes flow, or gas generation and pressurization; c) includes effect of temperature on fluid properties; and d) may include coupled chemical generation and transport, or ability to input/output such parameters to independent TC code.
Steps:

4. The system imports a geostratigraphic model of waste repository region.
5. Analyst determines key geologic features of repository and creates a mesh.
6. The system imports thermophysical properties.
7. The system imports initial conditions such as temperature, radionuclide concentration, moisture content, waste inventory.
8. The system imports boundary conditions for field variables and concentrations.
9. The system calculates thermal and flow fields as a function of time and space.
    a. Flow through variably saturated porous media
    b. Navier-Stokes flow in cavities (e.g. waste package) or drift
        i. Free surface flows if necessary (e.g. film flow, dripping)
    c. Gas generation and pressurization models

Coupled parameters:
Transfer flow and thermal field to chemistry solver => chemistry solver updates thermophysical and geologic properties.
    a. Couple as necessary
        i. One-way coupling
        ii. Two-way coupling
            1. Loose coupling
            2. Tight coupling

User interface: None

CUC #TH2: Flow, Heat, and Radionuclide Transport with Evolving Geometry
Purpose: Simulate heat, gravity, production/injection well, and precipitation driven flow and transport including appropriate boundary conditions when the geometry is evolving (e.g. subsidence, drift closure, corrosion, stress fractures, seismic events etc.).
Actors: System thermal-hydrologic code(s) that a) can be either Lagrangian or Eulerian; b) simulates either porous media flow, Navier-Stokes flow, or gas generation and pressurization; c) includes effect of temperature on fluid properties; and d) may include coupled chemical generation and transport, or ability to input/output such parameters to independent transport-chemical (TC) code.
Steps:

1. The system imports a geostratigraphic model of waste repository region.
2. Analyst determines key geologic features of repository and creates a mesh.
3. The system imports thermophysical properties.
4. The system imports initial conditions such as temperature, radionuclide concentration, moisture content, waste inventory.
5. The system imports boundary conditions for field variables and concentrations.
6. The system calculates thermal and flow fields as a function of time and space.
    a. Flow through variably saturated porous media
    b. Navier-Stokes flow in cavities(e.g. waste package) or drift

i. Free surface flows if necessary (e.g. film flow, dripping)

c. Gas generation and pressurization models

Coupled parameters:

Transfer flow and thermal field to chemistry solver => chemistry solver updates thermophysical and geologic properties.

a. Couple as necessary
   i. One-way coupling
   ii. Two-way coupling
      1. Loose coupling
      2. Tight coupling

Transfer flow, thermal, and chemistry fields to mechanical solver to update geometry, porosity, permeability etc.

a. Couple as necessary
   i. One-way coupling
   ii. Two-way coupling
      1. Loose coupling
      2. Tight coupling

User interface: None

MECHANICAL PROCESSES

CUC #M1: Closure of drift

Purpose: Simulate the closure of the drift around the waste package due to salt creep or clay deformation.

Actors: System continuum mechanics code, in particular a Lagrangian thermal-mechanical simulation code (e.g., JAS3D)

Steps:

1. The system imports the computational mesh of the WP/drift geometry.
2. The system imports either a) salt creep mechanical properties; or b) clay deformation properties.
3. The system imports temperature and moisture distribution in the domain to be simulated.
4. The system calculates salt/clay deformation and creeping as a function of time and space.
5. The system calculates resulting stress changes to WP/WF.
6. The system imports the mechanical and failure properties of waste forms and container materials.
7. The system determines if WF/WP stresses exceed failure criteria, mode of failure.

Coupled parameters:

Input: Temperature (from TH code; calculate thermal expansion, creep rates, thermally-dependent mechanical properties of WP/WF)
Corrosion of WP/WF (from TC code; for corrosion-induced failure)

Output: Stress profiles in WP/WF as function of time
Fracture of WP/WF (to TH code; use to identify new flow path)

User interface: Analysis-specific input for the particular computational code, i.e., input deck including material properties and computational parameters, user-developed subroutines for initial and boundary conditions.

CUC #M2: Rock fall in drift

Purpose: Simulate the thermal-mechanical behavior of the drift, and determine the potential for rock fall event with sufficient energy to cause mechanical failure in the waste package or waste form.

Actors: Three possible components: 1) Standard keyblock analysis using known or calculated stress levels in the drift; 2) System continuum mechanics code, in particular a Lagrangian thermal-mechanical simulation code (e.g., JAS3D); 3) Impact dynamics code (e.g., ALEGRA) to simulate fracture process in WP/WF.

Steps:

1. The system imports the computational mesh of the WP/drift geometry.
2. The system imports host rock mechanical properties and failure criteria.
3. The system imports temperature and moisture distribution in the domain to be simulated.
4. The system calculates stress changes to host rock due to thermally expansion.
5. The system compares stresses in host rock to rock failure criteria.
6. If rock failure is predicted, the system imports the mechanical and failure properties of waste forms and container materials.
7. The system predicts velocity and mass of falling rock striking the WP/WF.
8. The system determines if WP/WF stresses exceed failure criteria, mode of failure.

Coupled parameters:

Input: Temperature (from TH code; calculate thermal expansion, creep rates, thermally-dependent mechanical properties of WP/WF)
Corrosion of WP/WF (from TC code; for corrosion-induced failure)

Output: Stress profiles in WP/WF as function of time
Fracture of WP/WF (to TH code; use to identify new flow path)

User interface: Analysis-specific input for the particular computational code, i.e., input deck including material properties and computational parameters, user-developed subroutines for initial and boundary conditions.

CUC #M3: Closure of fractures in drift

Purpose: Simulate the thermal-mechanical behavior of the drift, and determine the change in fracture apertures and permeability in the near field.

Actors: System continuum mechanics code, in particular a Lagrangian thermal-mechanical simulation code (e.g., JAS3D)

Steps:

1. The system imports the computational mesh of the WP/drift geometry.
2. The system imports host rock mechanical properties and failure criteria.
3. The system imports initial fracture spacings and permeability for the DRZ in the host rock.
4. The system estimates initial fracture aperture thicknesses based on spacings and permeability.
5. The system imports temperature and moisture distribution in the domain to be simulated.
6. The system calculates stress changes to host rock due to thermally expansion.
7. The system predicts changes in fracture apertures or permeability.

Coupled parameters:

Input: Temperature (from TH code; calculate thermal expansion, creep rates, thermally-dependent mechanical properties of WP/WF)

Output: Changes in fractures in host rock, fracture permeability as f(x,y,z) (to TH code; use to modify hydrologic flow parameters)

User interface: Analysis-specific input for the particular computational code, i.e., input deck including material properties and computational parameters, user-developed subroutines for initial and boundary conditions.

CUC #M4: Swelling of clay (bentonite) backfill

Purpose: Simulate the hydroscopic swelling of bentonite backfill and the resulting stress changes on the WP/WF, closure of fractures/interfaces in backfill (i.e., change in permeability).

Actors: System coupled THM code

Steps:

1. The system imports the computational mesh of the WP/drift geometry.
2. The system imports host rock mechanical properties.
3. The system imports backfill mechanical and hydrologic properties, including those as a function of temperature.

4. The system imports temperature and moisture distribution in the domain to be simulated.
5. The system calculates stress volume changes in backfill due to hydroscopic swelling.
6. The system predicts changes in stress to WP/WF.
7. The system calculates change in porosity/permeability of backfill..

Coupled parameters:

Input: Temperature (from TH code; calculate thermal expansion, creep rates, thermally-dependent mechanical properties of WP/WF)
Moisture content in drift, gas and liquid (from TH code; use to calculate volume changes in backfill due to swelling)
Corrosion of WP/WF (from TC code; for corrosion-induced failure)

Output: Changes in porosity/permeability of backfill as f(x,y,z) (to TH code; use to modify hydrologic flow parameters)
Stress profiles in WP/WF as function of time
Fracture of WP/WF (to TH code; use to identify new flow path)

User interface: Analysis-specific input for the particular computational code, i.e., input deck including material properties and computational parameters, user-developed subroutines for initial and boundary conditions.

CUC #M5: Seismic activity in drift

Purpose: Simulate effect of seismic event on WP/WF, determine if mechanical failure can occur.

Actors: Three possible components: 1) Code required to develop wave function of specified seismic event; 2) System continuum mechanics code, in particular a Lagrangian thermal-mechanical simulation code (e.g., JAS3D); 3) Dynamic mechanics code (e.g., ALEGRA) to simulate fracture process in WP/WF.

Steps:

1. The system imports the computational mesh of the WP/drift geometry.
2. The system imports characterization of seismic event (wave form, duration, etc.).
3. The system calculates static stress state at time prior to event.
4. The system predicts velocity and mass of WP/WF.
5. The system determines if WP/WF stresses exceed failure criteria, mode of failure.

Coupled parameters:

Input: None (short-term calculation, thermal environment is “steady-state”)

Output: Stress profiles in WP/WF as function of time
Fracture of WP/WF (to TH code; use to identify new flow path)

User interface: Analysis-specific input for the particular computational code, i.e., input deck including material properties and computational parameters, user-developed subroutines for initial and boundary conditions.

CUC #M6: Crushed backfill due to drift creep

Purpose: Simulate the closure of the drift around the waste package due to salt creep, and the effect of that creep on the salt backfill around the WP/WF.

Actors: System continuum mechanics code, in particular a Lagrangian thermal-mechanical simulation code (e.g., JAS3D), plus calculate change in porosity/permeability of salt backfill to export to TH calculation.

Steps:

1. The system imports the computational mesh of the WP/drift geometry.
2. The system imports salt creep mechanical properties of the drift, and crushed salt properties of the salt backfill.
3. The system imports temperature and moisture distribution in the domain to be simulated.
4. The system calculates salt deformation and creeping as a function of time and space.
5. The system calculates the change in stress to the backfill.

6. The system calculates the change in porosity/permeability of the backfill, and outputs information to TH code.
7. The system calculates resulting stress changes to WP/WF.
8. The system imports the mechanical and failure properties of waste forms and container materials.
9. The system determines if WF/WP stresses exceed failure criteria, mode of failure.

Coupled parameters:

Input: Temperature (from TH code; calculate thermal expansion, creep rates, thermally-dependent mechanical properties of WP/WF)
Corrosion of WP/WF (from TC code; for corrosion-induced failure)

Output: Changes in porosity/permeability of backfill as f(x,y,z) (to TH code; use to modify thermal/hydrologic flow parameters)
Stress profiles in WP/WF as function of time
Fracture of WP/WF (to TH code; use to identify new flow path)

User interface: Analysis-specific input for the particular computational code, i.e., input deck including material properties and computational parameters, user-developed subroutines for initial and boundary conditions. Also, output computed change of porosity/permeability of backfill to TH code.

Coupled Chemical Processes

CUC #C01: Waste Inventory
Purpose: Simulate evolution with time of waste form composition and radionuclide isotopic composition and distribution within the waste form and inside waste package
Actors: System
Steps:

1. The analyst specifies the in-package domain to be simulated, and the system imports the computational mesh.
    a. Waste package type
        i. Waste package dimension and materials
        ii. Internal structural components dimensions and materials
    b. Initial environment inside intact waste package
        i. Thermal condition
        ii. Gas composition
    c. Waste form initial condition property
        i. Waste form type, quantity and dimension
        ii. Waste form canister dimension and materials
        iii. Initial waste form phase composition
        iv. Radionuclide isotopic composition and distribution within the waste form
2. The system imports physicochemical properties of waste form and radionuclides related to transport and re-distribution of radionuclides within the waste form.
    a. Solid state diffusion
    b. Grain boundary diffusion
    c. Radioactive decay and in-growth
3. The system imports thermal environment inside waste package from thermal-hydrological model
4. The system imports waste form physical condition (e.g., porosity, cracks, surface area, etc.) from mechanical model
5. The system calculates at each grid point for each time step:
    a. Gas composition inside waste package
    b. Waste form phase composition
    c. Radionuclide isotopic composition.

   d. Update for waste form materials properties related to transport of mobile radionuclides within the waste form
   e. Transport and re-distribution of mobile radionuclides within the waste form (e.g., matrix, grain boundaries, outer surface, etc.)
   f. Release of volatile radionuclides from waste form, and their composition and distribution in the space between the waste form and waste package.
6. The system stores simulation results for use by other use cases.

Coupled parameters:

Input: - Thermal environment inside intact, and, when occurs, breached waste package from thermal-hydrological model
- Waste form physical condition (e.g., porosity, cracks, surface area, etc.) from mechanical model

Output: - Waste inventory and radionuclide composition changes with time and location

User interface: None.

CUC #C02: Chemistry of Incoming Water to Emplacement Drift

Purpose: Simulate chemistry of incoming water into the emplacement drift.

Actors: System

Steps:

1. The analyst specifies the domain to be simulated, and the system imports the computational mesh.
2. The analyst specifies initial and boundary conditions
   a. Initial ambient water and gas chemistry in the near-field host rock
   b. Initial ambient water and gas chemistry in the emplacement drift before waste emplacement.
3. The system imports thermal-hydrological condition in the emplacement drift and near-field host rock from thermal-hydrological model
   a. Thermal and water flow fields
      i. Dry-out zone development
      ii. Re-wetting of dry-out zone
   b. Gas flow field (for unsaturated condition)
      i. Relative humidity (RH)
      ii. CO_2 partial pressure
      iii. O_2 partial pressure
   c. Seepage into emplacement drift (for unsaturated condition)
4. The analyst specifies chemical reaction network and database for water-rock interaction, and the system imports them as specified.
   a. Thermodynamic property models and parameters
   b. Activity coefficient models and parameters
   c. Dissolution and precipitation kinetic models and parameters
   d. Sorption kinetic models and parameters
   e. Ion exchange kinetic models and parameters
   f. Microbial metabolism and growth model and parameters
5. The system calculates water and gas chemistry in the near-field host rock at each grid point for each time step
   a. Mineral phase assemblage evolution
   b. Matrix and fracture porosity changes
   c. Microbial activity effects
6. The system exports simulation results to thermal-hydrologic model for updates of the thermal-hydrologic properties in the near-field host rock
   a. One-way coupling

   b. Loose coupling
   c. Tight coupling
7. The system stores simulation results for use by other use cases.

Coupled parameters:

Input: - Thermal-hydrological condition in emplacement drift and near-field host rock from thermal-hydrological model

Output: - Water and gas chemistry changes in emplacement drift with time and location
- Porosity changes with time and location to thermal-hydrological model

User interface: None.

CUC #C03: Chemical Interactions of Water with Ground Supports and Other Introduced Materials in Emplacement Drift

Purpose: Simulate evolution of water chemistry from interaction with ground support and other introduced materials in the emplacement drift.

Actors: System

Steps:

1. The analyst specifies types, dimensions, materials, quantities, and initial compositions of ground supports and other introduced materials
   a. Emplacement drift liners
   b. Invert on the drift floor
   c. Waste package supports
   d. Grouts
   e. Steel mesh
   f. Rock bolts
   g. Mature cementitious material compositions, if used
2. The analyst specifies the domain to be simulated, and the system imports the computational mesh.
3. The analyst specifies initial and boundary conditions
   a. Initial ambient water and gas chemistry in the emplacement drift
4. The analyst specifies chemical degradation models and parameter for ground support and other introduced materials
   a. Corrosion for steel components
   b. Physical and chemical degradation of cementitious material components
5. The system imports thermal-hydrological condition from thermal-hydrological model
   a. Thermal and water flow fields
      i. Dry-out zone development
      ii. Re-wetting of dry-out zone
   b. Gas flow field (for unsaturated condition)
      i. Relative humidity (RH)
      ii. CO_2 partial pressure
      iii. O_2 partial pressure
   c. Seepage into emplacement drift (for unsaturated condition)
6. The system imports mechanical damage and failure of ground supports and other introduced materials from mechanical model
   a. Thermal and static stress
   b. Stress from swelling pressure of steel corrosion products
   c. Structural strength reduction from thinning by corrosion, leaching and other degradation mechanisms
   d. Cracking
   e. Spalling

7. The analyst specifies chemical reaction network and database for chemical interactions with water, and the system imports them as specified.
    a. Thermodynamic property models and parameters
    b. Activity coefficient models and parameters
    c. Dissolution and precipitation kinetic models and parameters
    d. Sorption kinetic models and parameters
    e. Ion exchange kinetic models and parameters
    f. Microbial metabolism and growth model and parameters
8. The system imports chemistry of water and gas coming into emplacement drift
9. The system calculates water and gas chemistry from interaction with ground support and other introduced materials at each grid point for each time step
    a. Corrosion of steel components
    b. Dissolution, leaching and other degradation of cementitious materials
    c. Precipitation and re-dissolution of secondary phases
    d. Microbial activity effects
10. The system exports simulation results to thermal-hydrological model for update of the thermal-hydrologic condition in emplacement drift
    a. One-way coupling
    b. Loose coupling
    c. Tight coupling
11. The system exports simulation results to mechanical model for updates of the mechanical condition of ground supports
    a. One-way coupling
    b. Loose coupling
    c. Tight coupling
12. The system stores simulation results for use by other use cases.

Coupled parameters:

Input: - Thermal-hydrological conditions in emplacement drift from thermal-hydrological model
- Mechanical damage and failure of ground supports and other introduced materials from mechanical model

Output: - Thermal-hydrologic property changes to thermal-hydrologic model
- Physical condition changes of ground support or introduced material to mechanical model

User interface: None.

CUC #C04: Chemical Interactions of Water with Backfill

Purpose: Simulate evolution of water chemistry from interaction with backfill around waste package in the emplacement drift.

Actors: System

Steps:

1. The analyst specifies backfill types, materials, dimensions, quantities, and initial compositions
    a. Clay
    b. Crushed rocks
    c. Introduced contaminants
    d. Introduced microbes and organic matters
2. The analyst specifies the domain to be simulated, and the system imports the computational mesh.
3. The analyst specifies initial and boundary conditions
    a. Initial ambient water and gas chemistry in the backfill
    b. Initial ambient water and gas chemistry in the emplacement drift

4. The system imports thermal-hydrological condition for the backfill from thermal-hydrological model
    a. Thermal and water flow fields
        i. Dry-out zone development
        ii. Re-wetting of dry-out zone
    b. Gas flow field (for unsaturated condition)
        i. Relative humidity (RH)
        ii. CO_2 partial pressure
        iii. O_2 partial pressure
5. The system imports mechanical condition for the backfill from mechanical model
    a. Thermal stress, and static stress from its own weight
    b. Dynamic and static stress from rockfall and rubble accumulation over the backfill
    c. Stress from swelling pressure of expansive clays
    d. Cracking
    e. Fracturing
6. The analyst specifies chemical reaction network and database for chemical interaction of backfill materials with water, and the system imports them as specified.
    a. Thermodynamic property models and parameters
    b. Activity coefficient models and parameters
    c. Dissolution and precipitation kinetic models and parameters
    d. Sorption kinetic models and parameters
    e. Ion exchange kinetic models and parameters
    f. Microbial metabolism and growth model and parameters
7. The system imports chemistry of water and gas coming into the backfill
8. The system calculates water and gas chemistry in the backfill at each grid point at each time step
    a. Dissolution and leaching of backfill materials
    b. Precipitation and re-dissolution of secondary phases
    c. Water and gas chemistry in matrix and fractures
    d. Microbial activity effects
    e. Chemistry of water and gas exiting backfill and contacting waste package surface
9. The system exports simulation results to thermal-hydrologic model for updates of the thermal hydrologic condition for the backfill
    a. One-way coupling
    b. Loose coupling
    c. Tight coupling
10. The system exports simulation results to mechanical model for updates of the mechanical condition for the backfill
    a. One-way coupling
    b. Loose coupling
    c. Tight coupling
11. The system stores simulation results for use by other use cases.

Coupled parameters:

Input: - Thermal-hydrological conditions in the backfill from thermal-hydrological model
- Mechanical condition of the backfill from the continuum mechanical model

Output: - Thermal-hydrologic property changes of the backfill to thermal-hydrologic model
- Physical condition changes of the backfill to mechanical model

User interface: None.

CUC #C05: Chemical Interactions of Water with Rockfall Rubbles

Purpose: Simulate evolution of water chemistry from interaction with rockfall rubbles around the waste package. This use case is for an EBS design option with no engineered backfill in the emplacement drift.
Actors: System
Steps:

1. The analyst specifies the domain to be simulated, and the system imports the computational mesh.
2. The analyst specifies initial and boundary condition
    a. Initial ambient water and gas chemistry in emplacement drift
    b. Initial pore water composition and mineral assemblage of host rock
3. The system imports mechanical degradation of emplacement drift from mechanical model
    a. Geometry of accumulated rockfall rubbles around waste package
    b. Quantity and layer thickness of accumulated rockfall rubbles
    c. Rockfall rubble size distribution
4. The system refines the simulation domain mesh for degraded drift and accumulated rubble around waste package
    a. If necessary, mineral assemblage changes of rockfall rubbles from water-rock interactions prior to rockfall
5. The system imports thermal hydrological condition for degraded drift and accumulated rockfall rubbles from thermal hydrological model
    a. Thermal and water flow fields
        i. Dry-out zone development
        ii. Re-wetting of dry-out zone
    b. Gas flow field (for unsaturated condition)
        i. Relative humidity (RH)
        ii. CO_2 partial pressure
        iii. O_2 partial pressure
6. The analyst specifies chemical reaction network and database for chemical interaction of rockfall rubbles with water, and the system imports them as specified.
    a. Thermodynamic property models and parameters
    b. Activity coefficient models and parameters
    c. Dissolution and precipitation kinetic models and parameters
    d. Sorption kinetic models and parameters
    e. Ion exchange kinetic models and parameters
    f. Microbial metabolism and growth
7. The system imports chemistry of water and gas coming into the rockfall rubbles
8. The system calculates water and gas chemistry in the rubbles at each grid point for each time step
    a. Dissolution and leaching of rock rubbles
    b. Precipitation and re-dissolution of secondary phases
    c. Microbial activity effects
    d. Chemistry of water and gas exiting rubbles and contacting waste package surface
9. The system exports simulation results to thermal-hydrologic model for updates of the thermal-hydrologic conditions for rockfall rubbles
    a. One-way coupling
    b. Loose coupling
    c. Tight coupling
10. The system imports updates for drift mechanical degradation and rockfall rubble accumulation from mechanical model
    a. One-way coupling
    b. Loose coupling
    c. Tight coupling
11. The system stores simulation results for use by other use cases

Coupled parameters:

Input: - Thermal-hydrological conditions in degraded drift and accumulated rockfall rubbles from thermal-hydrological model.
- Drift degradation and rockfall rubble accumulation from mechanical model.

Output: - Thermal-hydrologic property changes of rockfall rubbles to thermal hydrologic model

User interface: None.

CUC #C06: Uniform Corrosion of Waste Package

Purpose: Simulate uniform corrosion process and penetration of waste package wall.

Actors: System

Steps:

1. The analyst specifies the domain to be simulated, and the system imports the computational mesh.
    a. Waste package dimensions and materials
    b. Waste package manufacturing data
2. The analyst specifies initial conditions of waste package
    a. Initial physical condition of waste package surface
        i. Oxide film condition and thickness
        ii. Scratches and other surface damages
    b. Initial ambient water and gas chemistry contacting waste package surface
3. The system imports thermal-hydrological conditions on waste package surface from thermal hydrological model
    a. Thermal and water flow fields
        i. Seepage dripping onto waste package (for unsaturated condition)
    b. Gases in the drift and contacting waste package surface (for unsaturated condition)
        i. Relative humidity (RH)
        ii. CO_2 partial pressure
        iii. O_2 partial pressure
4. The analyst specifies chemical reaction network and database for chemical environments on the waste package surface, and the system imports them as specified.
    a. Thermodynamic property models and parameters
    b. Activity coefficient models and parameters
    c. Dissolution and precipitation kinetic models and parameters
    d. Sorption kinetic models and parameters
    e. Ion exchange kinetic models and parameters
    f. Microbial metabolism and growth model and parameters
5. The system imports chemistry of water and gas contacting waste package
    a. CUC C#04 (Chemical Interactions of Water with Backfill) for a backfill design option
    b. CUC #C05 (Chemical Interactions of Water with Rockfall Rubbles) for a no-backfill design option
6. The system calculates uniform corrosion damage of waste package at each grid point for each time step
    a. Water and gas chemistry, temperature and electrochemical corrosion potential at each grid point on the waste package surface
    b. Distribution of anodic and cathodic sites and their sizes on the waste package surface
    c. Passive film stability
    d. Passive current density distribution
    e. Galvanic coupling effect if waste package contacts with other metal
    f. Effects of microbial metabolism and biofilm growth on passive film stability and passive current density
    g. Advance of passive corrosion front into metal matrix (or uniform corrosion penetration)
        i. Thinning of waste package wall by uniform corrosion

ii. Corrosion product formation and their quantity
h. Time for initial breach of waste package by uniform corrosion
i. Number and sizes of breach openings by uniform corrosion

7. The system exports simulation results to thermal-hydrologic model for updates of thermal-hydrologic conditions on waste package surface
   a. One-way coupling
   b. Loose coupling
   c. Tight coupling
8. The system stores simulation results for use by other use cases

Coupled parameters:

Input: - Thermal-hydrological conditions on waste package surface from thermal-hydrological model (See Step xx).

Output: - Thermal-hydrological property changes on waste package surface to thermal hydrologic model

User interface: None.

CUC #C07: Localized Corrosion of Waste Package

Purpose: Simulate localized corrosion (pitting and crevice corrosion) process and penetration of waste package wall.

Actors: System

Steps:

1. The analyst specifies the domain to be simulated, and the system imports the computational mesh.
   a. Waste package dimensions and materials
   b. Waste package manufacturing data (welds, etc.)
2. The analyst specifies initial conditions of waste package
   a. Initial physical condition of waste package surface
      i. Oxide film condition and thickness
      ii. Scratches and other surface damages
   b. Initial ambient water and gas chemistry contacting waste package surface
3. The analyst specifies localized corrosion model and parameters
   a. Pitting corrosion
      i. Initiation model and parameters
      ii. Penetration model and parameters
   b. Crevice corrosion
      i. Initiation model and parameters
      ii. Penetration model and parameters
4. The system imports thermal-hydrological conditions on waste package surface from thermal hydrological model
   a. Thermal and water flow fields
      i. Seepage dripping onto waste package (for unsaturated condition)
   b. Gas phase contacting waste package surface (for unsaturated condition)
      i. Relative humidity (RH)
      ii. CO_2 partial pressure
      iii. O_2 partial pressure
5. The analyst specifies chemical reaction network and database for chemical environments on the waste package surface, and the system imports them as specified.
   a. Thermodynamic property models and parameters
   b. Activity coefficient models and parameters
   c. Dissolution and precipitation kinetic models and parameters
   d. Sorption kinetic models and parameters

e. Ion exchange kinetic models and parameters
f. Microbial metabolism and growth model and parameters

6. The system imports chemistry of water and gas contacting waste package
   a. CUC C#04 (Chemical Interactions of Water with Backfill) for a backfill design option
   b. CUC #C05 (Chemical Interactions of Water with Rockfall Rubbles) for a no-backfill design option
7. The system calculates pitting corrosion damage of waste package at each time step
   a. Water and gas chemistry, temperature and electrochemical corrosion potential at each grid point on the waste package surface
   b. If pitting corrosion initiation condition is met, refine the mesh for pit geometry
   c. Initiate and grow pits
      i. Number of incipient pits
      ii. Electrochemical corrosion potentials inside and outside growing pits
      iii. Transport of reacting and product species into and out of pits
      iv. Corrosion products and other solid phase formation inside and outside pits
   d. Passive film damage and repair processes at pit bottom
      i. Sustained pit growth
      ii. Pit growth rate decay
      iii. Pit stifling
   e. Effects of galvanic coupling if waste package contacts with other metals
   f. Effects of microbial metabolism and biofilm growth
      i. Pit initiation and growth
      ii. Passive film stability
   g. Time for initial breach of waste package by pitting corrosion
   h. Number and sizes of breach openings by pitting corrosion
8. The system calculates crevice corrosion damage of waste package at each time step
   a. Water and gas chemistry, temperature and electrochemical corrosion potential at each grid point on waste package surface
   b. Potential crevice corrosion sites on waste package surface
      i. Contacts with backfill material particles
      ii. Contacts with rockfall rubble particles
      iii. Under Mineral deposits
      iv. Contacts with waste package supports and/or with other engineered materials
      v. Under microbial films or colonies
   c. Refine the mesh for potential crevice corrosion sites for crevice geometry
   d. Water chemistry inside crevice and electrochemical corrosion potentials inside and outside crevice
      i. Transport of reacting and product species into and out of crevice
      ii. Corrosion products and other solid phase formation inside and outside crevice
   e. If crevice corrosion initiation condition is met, grow crevice
   f. Passive film damage and repair processes inside crevice and at growing crevice bottom
      i. Sustained crevice growth
      ii. Crevice growth rate decay
      iii. Crevice stifling
   i. Effects of galvanic coupling if waste package contacts with other metals
   g. Effects of microbial metabolism and biofilm growth
      i. Crevice formation
      ii. Crevice initiation and growth
      iii. Passive film stability
   h. Time for initial breach of waste package by crevice corrosion
   i. Number and sizes of breach openings by crevice corrosion

9. The system exports simulation results to thermal-hydrological model for updates of thermal-hydrologic conditions on waste package surface
    a. One-way coupling
    b. Loose coupling
    c. Tight coupling
10. The system stores simulation results for use by other use cases

Coupled parameters:

Input: - Thermal-hydrological conditions on the waste package surface from thermal-hydrological model (See Step xx).

Output: - Thermal-hydrological property changes on the waste package surface to thermal-hydrologic model

User interface: None.

CUC #C08: Stress Corrosion Cracking of Waste Package

Purpose: Simulate stress corrosion cracking (SCC) process and penetration of waste package wall.

Actors: System

Steps:

1. The analyst specifies the domain to be simulated, and the system imports the computational mesh.
    a. Waste package dimensions and materials
    b. Waste package manufacturing data (welds, etc.)
2. The analyst specifies initial conditions of waste package
    a. Initial physical condition of waste package surface
        i. Oxide film condition and thickness
        ii. Scratches and other surface damages
    b. Initial ambient water and gas chemistry contacting waste package surface
3. The analyst specifies SCC-associated models and parameters
    a. SCC-related mechanical properties of waste package materials
        i. Young's modulus
        ii. Yield strength
        iii. Threshold stress intensity factor
    b. Number and size of surface and embedded flaws
    c. Number and size of incipient cracks
    d. Threshold stress for crack coalescence and initiation
    e. Crack growth and arrest
4. The system imports stress profiles in the waste package from mechanical model
    a. Weld stress
    b. Manufacturing stress
    c. Static stress from its own weight
    d. Static stress from swelling pressure of clay backfill if used
    e. Bending stress at contact points at the waste package bottom
    f. Dynamic and static stress from rockfall and rubble accumulation on waste package
5. The system imports thermal-hydrological conditions on waste package surface from thermal-hydrological model
    a. Thermal and water flow fields
        i. Seepage dripping onto waste package (for unsaturated condition)
    b. Gases contacting waste package surface (for unsaturated condition)
        i. Relative humidity (RH)
        ii. CO_2 partial pressure
        iii. O_2 partial pressure

6. The analyst specifies chemical reaction network and database for chemical environments on the waste package surface, and the system imports them as specified.
    a. Thermodynamic property models and parameters
    b. Activity coefficient models and parameters
    c. Dissolution and precipitation kinetic models and parameters
    d. Sorption kinetic models and parameters
    e. Ion exchange kinetic models and parameters
    f. Microbial metabolism and growth model and parameters
7. The system imports chemistry of water and gas contacting waste package
    a. CUC C#04 (Chemical Interactions of Water with Backfill) for a backfill design option
    b. CUC #C05 (Chemical Interactions of Water with Rockfall Rubbles) for a no-backfill design option
8. The system calculates SCC damage of waste package at each time step
    a. Water and gas chemistry, temperature, and electrochemical corrosion potential at each grid point on waste package surface
    b. Potential SCC sites on waste package surface
        i. Residual tensile stress exceeding threshold stress for crack coalescence and growth
    c. Refine the mesh for potential SCC sites for crack geometry
    d. Initiate and grow cracks if tensile stress at the crack tip exceeds the threshold stress for crack initiation
    e. Crack growth rate
        i. Stress intensity factor at the crack tip
        ii. Stress relaxation at the crack tip as crack propagates
        iii. Crack growth rate decay
        iv. Crack arrest
    f. Water chemistry and electrochemical corrosion potentials inside and outside crevice
        i. Transport of reacting and product species into and out of crack
        ii. Corrosion products and other solid phase formation inside and outside crack
    g. Passive film damage and repair processes at the crack tip and crack walls
    h. Effects of galvanic coupling if waste package contacts with other metals
    i. Effects of microbial metabolism and biofilm growth
    j. Time for initial breach of waste package by SCC
    k. Number and sizes of breach openings by SCC
9. The system imports updates for stress profiles in waste package from mechanical model
    a. One-way coupling
    b. Loose coupling
    c. Tight coupling
9. The system exports simulation result to thermal-hydrologic model for updates of thermal hydrologic conditions on waste package surface
    d. One-way coupling
    e. Loose coupling
    f. Tight coupling
10. The system stores simulation results for use by other use cases

Coupled parameters:

Input: - Thermal-hydrological conditions on the waste package surface from thermal hydrological model
- Stress profiles in waste package from the continuum mechanical model

Output: - Thermal-hydrological property changes on the waste package surface to thermal hydrologic model

User interface: None.

CUC #C09: Waste Package Internal Structure Degradation
Purpose: Simulate corrosion degradation of waste package internal structural materials upon initial breach of waste package.
Actors: System
Steps:

1. The analyst specifies the domain to be simulated, and the system imports the computational mesh.
    a. Initial environment inside intact waste package
    b. Waste package internal structure dimensions and materials
2. The system imports waste package corrosion degradation profiles (initial breach time, and breach opening types and extents)
3. The system imports in-package water chemistry.
4. The analyst specifies corrosion degradation models and parameters for internal structural materials.
5. The analyst specifies chemical reaction network and database for chemical condition changes associated with internal structural materials corrosion, and the system imports them as specified.
    a. Thermodynamic property model and parameters
    b. Activity coefficient model and parameters
    c. Dissolution and precipitation rate kinetic models and parameters
    d. Sorption rate kinetic models and parameters
    e. Ion exchange rate kinetic models and parameters
    f. Microbial metabolism and growth model and parameters
6. The system imports thermal-hydrological environments inside breached waste package from thermal-hydrological model
    a. Temperature
    b. Relative humidity (RH) (unsaturated condition)
    c. CO_2 partial pressure, and O_2 partial pressure (unsaturated condition)
    d. Seepage into waste package
    e. Flow field inside waste package
    f. Saturation of porous degradation products and other mineral precipitates inside waste package (unsaturated condition)
7. The system calculates degradation of internal structure materials at each grid point for each time step, upon initial waste package breach
    a. Dry-air oxidation damage (RH less than a humid-air corrosion threshold RH).
    b. Humid-air corrosion damage (RH greater than a humid-air corrosion threshold RH, but less than an aqueous corrosion threshold RH).
        i. Uniform corrosion
        ii. Localized corrosion (pitting and crevice corrosion)
        iii. Stress corrosion cracking
        iv. Microbiologically influenced corrosion (MIC)
    c. Aqueous corrosion damage (contact with liquid water, or RH greater than an aqueous corrosion threshold RH)
        i. Uniform corrosion
        ii. Localized corrosion (pitting and crevice corrosion)
        iii. Stress corrosion cracking
        iv. Microbiologically influenced corrosion (MIC)
    d. Corrosion products and other mineral precipitates build-up
        i. Precipitation and re-dissolution of corrosion product and other mineral phases on, near and away from the corroding internal structural components.

ii. Transport of reacting and product species into and out of the corroding surface through corrosion product and other mineral precipitate phases

e. Chemistry of water contacting internal structure materials (pH, redox potential, ionic strength, chloride ion concentration, nitrate ion concentration, etc.)

8. The system exports simulation results to thermal-hydrologic model for updates of thermal-hydrologic conditions inside breached waste package
   a. One-way coupling
   b. Loose coupling
   c. Tight coupling
9. The system exports corrosion damage conditions of internal structures to mechanical model for structural analysis
   a. Thinning damage of internal structural components from corrosion
   b. Stress corrosion cracking damage
   c. Mechanical model determines time for internal structure failures and slump of waste form canisters
   d. Couple as necessary
      i. One-way coupling
      ii. Loose coupling
      iii. Tight coupling
10. The system stores simulation results for use by other use cases

Coupled parameters:

Input: - Thermal-hydrological conditions inside breached waste package from thermal-hydrological model (See Step xx).
- Internal structure mechanical failure and re-arrangement of waste form canisters from mechanical model.

Output: - Thermal-hydrological property changes inside breached waste package to thermal-hydrologic model
- Corrosion damage conditions of internal structures to mechanical model for structural analysis

User interface: None.

CUC #C10: Waste Form Canister Corrosion Degradation

Purpose: Simulate corrosion degradation of waste form canister upon initial breach of waste package.

Actors: System

Steps:

1. The analyst specifies the domain to be simulated, and the system imports the computational mesh.
2. The analyst specifies initial and boundary conditions
   a. Initial environment inside intact waste package
   b. Waste form canister dimensions and materials
3. The system imports waste package corrosion degradation profiles (initial breach time, and breach opening types and extents)
4. The system imports in-package water chemistry
5. The analyst specifies corrosion degradation models and parameters for waste form canister materials
6. The analyst specifies chemical reaction network and database for chemical condition changes associated with waste form canister corrosion, and the system imports them as specified.
   a. Thermodynamic property model and parameters
   b. Activity coefficient model and parameters
   c. Dissolution and precipitation kinetic models and parameters
   d. Sorption kinetic models and parameters

e. Ion exchange kinetic models and parameters
f. Microbial metabolism and growth model and parameters

7. The system imports thermal-hydrological environment inside breached waste package from thermal-hydrological model
   a. Temperature
   b. Relative humidity (RH) (unsaturated condition)
   c. CO_2 partial pressure, and O_2 partial pressure (unsaturated condition)
   d. Seepage into waste package
   e. Flow field inside waste package
   f. Saturation of porous degradation product media inside waste package (unsaturated condition)
8. The system calculates degradation of waste form canister materials at each grid point for each time step, upon initial waste package breach:
   a. Dry-air oxidation (RH less than a humid-air corrosion threshold RH).
   b. Humid-air corrosion (RH greater than a humid-air corrosion threshold RH, but less than an aqueous corrosion threshold RH).
      i. Uniform corrosion
      ii. Localized corrosion (pitting and crevice corrosion)
      iii. Stress corrosion cracking
      iv. Microbiologically influenced corrosion (MIC)
   c. Aqueous corrosion (contact with liquid water, or RH greater than an aqueous corrosion threshold RH).
      i. Uniform corrosion
      ii. Localized corrosion (pitting and crevice corrosion)
      iii. Stress corrosion cracking
      iv. Microbiologically influenced corrosion (MIC)
   d. Corrosion products and other mineral precipitates build-up
      i. Precipitation and re-dissolution of corrosion product and other mineral precipitate phases on, near and away from waste form canister
      ii. Transport of reacting and product species into and out of the corroding surface through corrosion product and other mineral phases
   e. Radiolysis effect
   f. Radiation damage effect
   g. Chemistry of water contacting waste form canister (pH, redox potential, ionic strength, chloride ion concentration, nitrate ion concentration, etc.)
   h. Breach of waste form canister
      i. Time of initial breach
      ii. Breach type, opening size, and numbers from different corrosion attack modes
9. The system exports corrosion damage conditions of waste form canisters to thermal-hydrological model
   a. Water flow through corrosion breach openings in waste form canister
   b. Water contact with exposed waste form
   c. Thermal and water flow field inside waste form canister
   d. Couple as necessary
      i. One-way coupling
      ii. Loose coupling
      iii. Tight coupling
10. The system exports corrosion damage conditions of waste form canister to mechanical model for structural analysis
    a. Thinning damage from corrosion
    b. Stress corrosion cracking damage

c. Mechanical model determines the time for waste form canister structure failure and loss of barrier function to water flow into the canister
d. Couple as necessary
    i. One-way coupling
    ii. Loose coupling
    iii. Tight coupling

11. The system stores simulation results for use by other use cases

Coupled parameters:

Input: - Thermal-hydrological conditions inside breached waste package and breached waste form canister from thermal-hydrological model.

Output: - Thermal hydrological property changes inside breached waste package to thermal hydrologic model
- Corrosion damage conditions of waste form canister to mechanical model for structural analysis

User interface: None.

CUC #C11: Waste Form Degradation and In-Package Chemical Environment

Purpose: Simulate waste form degradation, radionuclide release and mobilization from waste form, and in-package chemical environment inside breached waste package.

Actors: System

Steps:

1. The analyst specifies the domain to be simulated, and the system imports the computational mesh.
2. The analyst specifies initial and boundary conditions
    a. Initial environment inside intact waste package
    b. Waste form property
        i. Waste form type, quantity and dimensions
        ii. Waste form canister dimensions and materials
    c. Waste package property
        i. Waste package container dimensions and materials
        ii. Internal structure dimensions and materials
3. The analyst specifies waste form degradation models and their parameters, and the system imports them.
4. The system imports waste inventories with time and location
    a. Waste form phase compositions
    b. Radionuclide isotopic compositions
    c. Distribution of radionuclides within waste form and in the space between waste form and waste package.
5. The system imports waste package degradation profiles (initial breach time, and breach opening types and extents)
6. The analyst specifies chemical reaction network and database for chemical condition changes associated with waste form degradation, and the system imports them as specified
    a. Thermodynamic property model and parameters
    b. Activity coefficient model and parameters
    c. Dissolution and precipitation kinetic models and parameters
    d. Sorption kinetic models and parameters
    e. Ion exchange kinetic models and parameters
    f. Microbial metabolism and growth model and parameters
7. The system imports thermal-hydrological condition inside breached waste package from thermal-hydrological model
    a. Temperature

b. Relative humidity (RH) (unsaturated condition)
c. CO_2 partial pressure, and O_2 partial pressure (unsaturated condition)
d. Seepage into waste package
e. Flow field inside waste package
f. Saturation of porous degradation product media inside waste package (unsaturated condition)

8. The system imports physical conditions of waste form from mechanical model
    a. Crack properties (frequency or density, length, width, etc.)
    b. Surface area
9. The system calculates water and gas chemistry inside waste package at each grid point for each time step, upon initial breach of waste package
    a. The system imports corrosion degradation profiles of internal structure materials
        i. Internal structure corrosion products and other mineral phase formation, and their effects on in-package water chemistry
        ii. Time for internal structure failure causing slump of waste form containers
    b. The system imports degradation of waste form canisters
        i. Waste form canister corrosion breach and corrosion damages
        ii. Water flow into breached/damaged waste form canisters
        iii. Canister corrosion products and other mineral phase formation, and their effects on in-package water chemistry
    c. Chemistry of water and gas exiting waste package.
    d. Dissolved radionuclide concentrations in water
10. The system calculates waste form degradation at each grid point for each time step, upon initial waste form canister breach
    a. Chemistry of water contacting waste form (pH, redox potential, ionic strength, etc.)
    b. Mobilization of gaseous radionuclide from waste form.
    c. Mobilization of radionuclides from waste form surface and in the space between waste form and waste package.
    d. Dry-air oxidation of waste form matrix (RH less than a humid-air oxidation threshold RH).
    e. Humid-air oxidation of waste form matrix (RH greater than a humid-air oxidation threshold RH, but less than an aqueous-phase oxidation threshold RH).
    f. Aqueous phase oxidation of waste form matrix (contact with liquid water, or RH greater than an aqueous-phase oxidation threshold RH).
    g. Waste form matrix dissolution, and release of dissolved radionuclides and dissolved waste form matrix species.
    h. Release of radionuclides deposited on the grain boundaries.
    i. Waste form grain spalling/detachment from the matrix.
    j. Precipitation and re-dissolution of radionuclide solid phases near and away from waste form.
    k. Precipitation and re-dissolution of secondary phases near and away from waste form.
    l. Incorporation (sorption, ion exchange, etc.) of radionuclide into secondary phases.
    m. Update for waste form surface area due to matrix dissolution and alteration.
    n. Transport of reacting and product species into and out of corroding waste form surface through corrosion product and other mineral phases
    o. Radiolysis effects
    p. Radiation damage effects
    q. Microbial metabolism and growth effects
11. The system exports simulation results associated with thermal-hydrological property changes to thermal-hydrological model.
    a. Couple as necessary

i. One-way coupling
ii. Loose coupling
iii. Tight coupling

12. The system stores simulation results for use by other use cases.

Coupled parameters:

Input: - Thermal-hydrological conditions inside breached waste package from thermal-hydrological model.
- Internal structure failure from mechanical model

Output: - Thermal-hydrological property changes inside breached waste package to thermal-hydrologic model
- Corrosion damage conditions of internal structures and waste form canisters to mechanical model for structural analysis

User interface: None.

CUC #C12: Radionuclide Transport in EBS

Purpose: Simulate radionuclide release from breached waste package and transport in the engineered barrier system (EBS).

Actors: System

Steps:

1. The analyst specifies the domain to be simulated, and the system imports the computational mesh.
2. The analyst specifies initial and boundary conditions
    a. Initial chemical environment in emplacement drift
3. The system imports thermal-hydrological condition in the drift and inside breached waste package from thermal-hydrological model
    a. Temperature
    b. Relative humidity (RH) (for unsaturated condition)
    c. CO_2 partial pressure, and O_2 partial pressure (for unsaturated condition)
    d. Seepage into waste package
    e. Flow fields in the drift and inside breached waste package
    f. Saturation of porous media in the drift and inside waste package (for unsaturated condition)
    g. Permeability
    h. Porosity
    i. Tortuosity
4. The system imports waste package corrosion degradation profiles (initial breach time, and breach opening types, numbers and sizes)
5. The system imports in-drift chemical environments
6. The system imports in-package chemical environments
    a. Dissolved radionuclide concentration inside breached waste package
7. The analyst specifies chemical reaction network and database for chemical condition changes associated with radionuclide transport, and the system imports them as specified
    a. Thermodynamic property model and parameters
    b. Activity coefficient model and parameters
    c. Dissolution and precipitation kinetic models and parameters
    d. Sorption kinetic models and parameters
    e. Ion exchange kinetic models and parameters
    f. Microbial metabolism and growth model and parameters
8. The system calculates radionuclide transport out of breached waste package and in the drift at each grid point for each time step, upon breach of waste package and initial breach of waste form canister

a. Colloid generation and stability in the drift and inside waste package
    i. Natural colloid introduced by site groundwater
    ii. Colloids derived from corrosion and degradation products
    iii. Microbe-entrained and/or organic matter-entrained colloids
b. Sorption of radionuclides onto colloids
    i. Reversible sorption
    ii. Irreversible sorption
    iii. Competitive sorption and desorption kinetics for sorption sites on colloids by radionuclides
    iv. Amounts of radionuclides sorbed onto each type of colloids
c. Concentration of stable (or suspended in water) colloids
    i. Mass balance for stationary (or non-mobile) and stable (or mobile) colloids
    ii. Mass balance for sorbed radionuclides between stationary and stable colloids
d. Diffusive transport
    i. Dissolved radionuclides
    ii. Radionuclides sorbed (reversibly and irreversibly) on colloids
e. Advective transport
    i. Dissolved radionuclide
    ii. Radionuclide sored (reversibly and irreversibly) on colloids
f. Radioactive decay and in-growth of radionuclides

9. The system exports simulation results associated with thermal-hydrological property changes in the drift and inside breached waste package to thermal-hydrological model, and imports updates for thermal-hydrologic conditions from thermal-hydrologic model
    a. One-way coupling
    b. Loose coupling
    c. Tight coupling
10. The system stores simulation results for use by other use cases.

Coupled parameters:

Input: - Thermal-hydrological conditions inside breached waste package and in the drift from thermal-hydrological model.

Output: - Thermal-hydrological property changes inside breached waste package and in the drift to thermal-hydrologic model

User interface: None.

Appendix C

C. Potential Frameworks, Tools, and Simulations

There are a number of existing software collections that should be evaluated and potentially used to implement the NEAMS WF IPSC. Each of these software collections provide significant current and potential capabilities and complement each other to a large degree. There is very little significant overlap in core functionality between most of these software collections. In cases where there is overlap in functionality, that would need to be resolved in some way.

There are a few different categories of software that are important to the NEAMS WF IPCS. Some of the high-level categories are:

- Workflow management
- Multi-physics coupling
- PDE modeling and solvers
- Pre- and post-processing
- Numerical solvers
- Sub-continuum
- SQE software
- VU tools

Many of the various software collections described below contain software related to more than one of these categories. In this case, if we choose to use software Package X, we can choose to which aspect of that software we are interested in reusing. For example, both Salome and CCA contain support for multi-physics coupling but we may chose to ignore that capability in Salome and instead use CCA for multi-physics coupling and only use Salome for workflow management.

Each of these different software collections has different types of licenses. Everything from open-source LGPL to commercial and export controlled software are listed. Licensing issues must be carefully considered early on when considering what software the NEAMS WF IPCS will produce and release and to what customers. It is up to the NEAMS program elements to collectively decide what licensing requirements will be. For example, if it is expected that the majority of the non-export controlled NEAMS software will be developed and released as open source (e.g. LGPL), then it cannot have any mandatory dependencies on software with licenses that are incompatible with the chosen open source license. For example, a GPL external software dependency (such as Dakota) cannot be shipped with an LGPL software release. Other types of license incompatibilities also exist.

C.1 Salome

SALOME is open source (LGPL) software, developed jointly by the EDF and CEA in France, which provides a generic platform for pre- and post-processing for numerical simulations. It is based on an open architecture (based on CORBA) that incorporates a number of existing software integrated as software components. It is open-source (LGPL) can be downloaded in source code form.

At its most basic level, Salome is a framework for hooking together different components implemented through modules and is a workflow framework (as described in Section 6.3). Each module has its own GUI specification but uses the Salome GUI tools (built on Qt). It provides built-in support for driving the workflow through python scripting. At this basic level, Salome really is not biased for any particular type of simulation or tool.

Salome ships with a number of useful modules including a geometry module (GEOM, based on Open Cascade), a meshing module (SMESH, using their own internal mesh data structure MED), a component-coupling and driver module (YACS), and a post-processing modules (VISU, based on VTK). There are also a number of "solver" modules that have been hooked into Salome.

GEOM Module: The geometry module GEOM has a GUI to create geometries and looks pretty impressive. The GEOM module is based on Open Cascade. There are a number of primitives and operations supported to quickly create complex geometries. There is also a general python scripting interface to automate the generation of geometries. This looks very powerful. The GEOM module also supports several different geometry file formats to import and export geometries as files. I don't know how it compares to the CUBIT geometry GUI and engine but from what I have looked at, it would seem that the GEOM module in Salome is more powerful and flexible than the geometry GUI and engine in CUBIT. The GEOM module does not support any type of parametric sensitivity computations so shape sensitivities would not be well supported.

SMESH Module: Closely related to the GEOM module is the meshing module SMESH. The SMESH module includes a number of adapters that incorporate several meshing tools. They have to write adapters that translate the geometry description in the GEOM module into the format used by the native meshers. They then have written converters that will convert from the native mesher data structure into their common MED mesh data structure. The MED data structure is a concrete file format and an internal serial-only mesh data structure implementation. The MED mesh data structure appears to be pretty general but does not support everything. For example, MED does not support the handling of face curvature that is needed for high accuracy in higher order discretization methods. There is a GUI front end that uses VTK for visualization that can be used to both view the mesh and to manipulate the mesh in various ways. Again, I don't know how this compares to the GUI in CUBIT. However, it was expressed that the meshing algorithms in CUBIT are better than any existing open source meshers and there is great desire to get an open source version of the CUBIT meshers and create the SMESH adapters to incorporate it.

VISU Module: The built-in Salome VISU module is based on VTK and can be used to view meshes, solutions etc. It only works in serial. There are plans to create a new visualization module based on ParaView to allow viewing parallel data and data on remote machines.

YACS Module: The new (in Salome 5) YACS module has a GUI for hooking together different modules to compute results and flow data from one module to another. This can be used for things such as multi-physics coupling. For simple feed-forward configurations this could be helpful. However, for complex configurations the diagrams are too complex and a simple python-based description based on objects would be much more clear and easier to program and handle. Some non-programming users might like the GUI connections but that is difficult to judge.

Studies: Salome saves files in groups called 'Studies'. Every module that hooks into Salome is required to implement a 'save' function. Version control of Study files is not implemented. However, it would be easy to use another version control system to manage the versioning of these files be we would have to do this ourselves. These workflow issues are discussed in Section 6.3.

Portability: Salome depends on a fairly large stack of 30+ other software packages. The core Salome developers and users are almost exclusively running on Linux and Debian Linux at that. Some people at the June Salome short course expressed that it was hard to install Salome on other Linux distributions. There is some concern about the cost of porting Salome to various platforms.

Integration of new software components: Adding new modules that are CORBA-enabled is amazingly easy (using YACSGEN). Modules that implement components can be coded in C, C++, Fortran, or Python.

Support for MPI-style parallel computations: Salome does not directly support MPI-style parallel computations. The mesh data structure in SMESH is serial and CORBRA will not scale well with the numbers of processors. However, there is nothing stopping individual Salome CORBA components from running MPI. Consider a scenario where a user could be running the Salome GUI front end on their laptop and the over heavy duty meshing and computational components could be running on an MPP server. The user could create geometries locally with the GEOM module and then could invoke a mesher component on MPP server and run the MPI parallel application and solvers on the MPP. CORBRA would be used to communicate between the client laptop and the MPP server but the heavy duty computational components could communicate with CCA or directly with MPI in the same address space. YACS would have to be extended to allow other forms of communication between components but this should be feasible.

Sensitivity calculations: Salome does not seem to any support sensitivity computations needed to support fast and robust UQ and optimization methods. The GEOM and SMESH module have no support for any type of shape sensitivity. It would likely require significant development to enable shape sensitivities within Salome using the existing components. However, shape computations in the underlying application components could be handled no problem if those components supported it.

Software quality: The development version of Salome 5 used at the June Salome short course seemed to have quite a few bugs. You had to save frequently as the code would segfault and other errors would occur. Some of this was due to the incompatibly with older graphics drivers on the IBM Linux machines that were being used at IBM Watson but some of the other errors were not due to the graphics driver. This suggests that Salome is not developed with modern Lean/Agile ideas of highly stable code, the concept of 'Done' and keeping defects low by not putting defects in the software in the first place.

Software life cycle processes: In order be able to rely on Salome in a significant way, to affect its development for our use, and to ensure that we can meet our obligations to our customers, development and release lifecycle issues are critical (see Section 7.6). Currently, it would seem that Salome is being developed in a less than modern Lean/Agile way. When new features are added, they do not immediately finish the work to remove defects. Instead, they seem to add many new features with bugs and then work to debug the code before putting out a release. This is a very standard way to develop code but is not up to modern Lean/Agile standards. Because of this, release dates are not fixed and are few and far in between (a year or more) and it will be difficult to do shared co-development with unstable sources. This is the most significant issue since all other technical functionality can be addressed if there are solid software engineering processes and practices in place. However, the main Salome developers at the June Salome short course expressed interest in working with others to improve Salome and their software engineering processes and practices.

Software Categories: Workflow management, Multi-physics coupling, Pre- and post-processing

C.2 SIERRA

SIERRA is a large ASC-driven effort to develop high-end PDE solvers to run on large-scale parallel computers. It is both a framework for developing parallel finite-element (and related discretization) codes and a collection of concrete applications. The more recent SIERRA toolkit (STK) effort seeks to generalize and externalize some of the more significant bits of functionality needed to construct massively parallel discretization simulations. There is a large software foundation in SIERRA that needs to be considered for the NEAMS continuum modeling efforts.

Software Categories: Multi-physics coupling, PDE modeling and solvers, Pre- and post-processing

C.3 Trilinos

The Trilinos project at http://trilinos.sandia.gov is active in research, development, and production software for solving linear systems, nonlinear systems, automatic differentiation, data partitioning for load balance, time integration methods, and others. The software structure is component in nature with controlled dependencies among the pieces. It has a large user base, including large applications at SNL.

Some of the more relevant parts of Thyra related to the architecture and high-level design of the TCHM multi-physics coupling framework are described below.

Software Categories: Multi-physics coupling, PDE modeling and solvers, Pre- and post-processing, Numerical solvers, SQE software, VU tools

C.3.1 Thyra ModelEvaluator Model Interface

A brief description of the ModelEvaluator design and representation is appropriate. The primary ModelEvaluator interface is built on the Thyra C++ interface layer in Trilinos. Thyra is the offical abstract interface layer in Trilinos to facilitate the development of complex ANAs and to define the highest level of interoperability. Thyra is founded on a set of operator/vector interface classes which will not be discussed here [45]. Built on the foundation of the basic Thyra operator/vector interfaces are the Thyra operator solve interfaces which include the Thyra::LinearOpWithSolveBase and the Thyra::LinearOpWithSolveFactoryBase interfaces shown in Figure 13. A Thyra::LinearOpWithSolveBase object is simply a Thyra::LinearOpBase object that contains a solve() function and is therefore equivalent to a (iterative) linear solver. A Thyra::LinearOpWithSolveFactoryBase object takes basic forward Thyra::LinearOpBase objects and creates Thyra::LinearOpWithSolveBase objects. Therefore, Thyra::LinearOpWithSolveFactoryBase is basically a factory for linear solvers. These basic Thyra classes, along with the Thyra::ModelEvaluator, form the foundation for algorithms development for the Trilinos packages NOX, LOCA, Rythmos, MOOCHO, and other Trilinos packages.

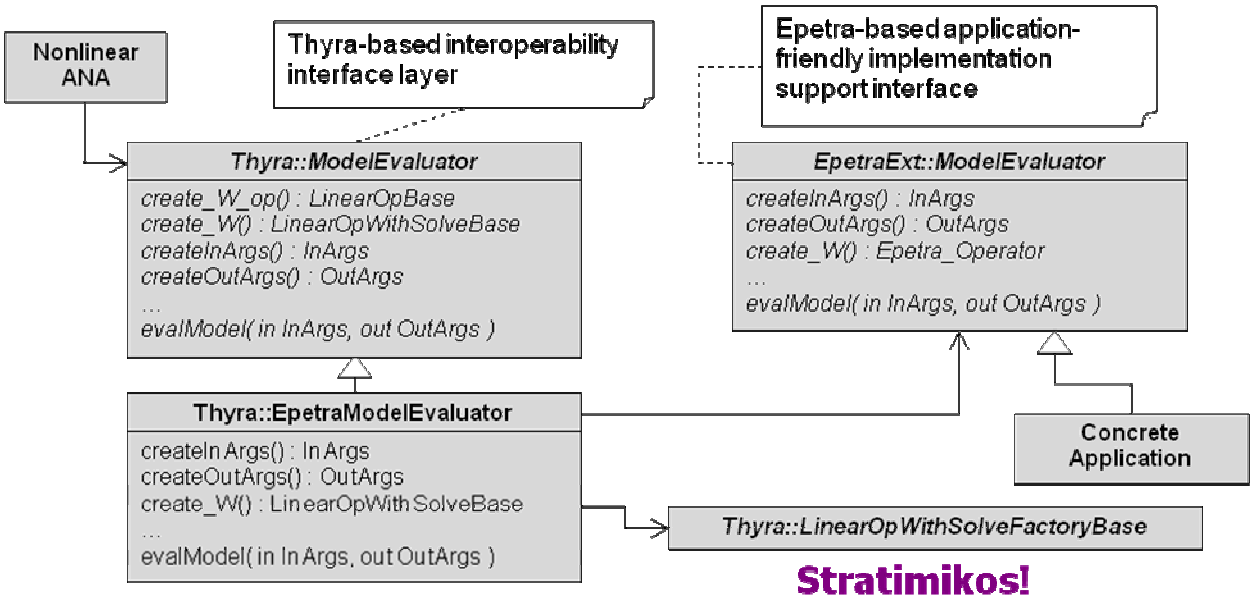

Figure 13: ModelEvaluator interfaces, adapters, and related software in Trilinos.

Figure 13 shows the two ModelEvaluator interfaces and related software that are being developed in Trilinos. The Thyra::ModelEvaluator interface is an interoperability layer that is designed to be used by nonlinear abstract numerical numerical algorithms (ANAs) [54] to drive the solution process. The ModelEvaluator acccepts inputs and outputs in a flexible way through InArgs and OutArgs objects and the evaluation of outputs given inputs in done in a "stateless" way through the abstract evalModel(…) function. By "stateless" we mean that calling evalModel(…) does not change the observered behavior of the ModelEvaluator object that is it is called on. Constructing numerical algorithms using a stateless evaluations results in much simplier and more powerfull algorithms (see [15] for a discussion of side-effect-free functions).

The EpetraExt::ModelEvaluator interface is almost idential to the Thyra::ModelEvaluator interface except it deals with raw Epetra objects (maps, vectors, and matrices) instead of abstract Thyra objects. The EpetraExt::ModelEvaluator is designed to be used to create concrete subclasses for specific applications/models. For example, Charon and Aria both have concrete EpetraExt::ModelEvaluator subclasses.

The Thyra::EpetraModelEvaluator class is a standard ADAPTER [46] subclass that takes any arbitrary EpetraExt::ModelEvalautor object and turns it into a basic Thyra::ModelEvaluator object. In addition, it accepts a Thyra::LinearOpWithSolveFactoryBase object and can therefore create linear solvers associated with the state-state derivative W.

The ModelEvaluator has already been extended to support stocastic Galerkin UQ methods [44]. In a stocastic Galerkin method, a probability polynomial expansion is used to represent the uncertain paraemeters which in turn leads to the computation of a similar polynomical expension for the state varaibles. These methods can be used even in a multi-physics environment.

From a software perstective, the ModelEvalutor is strongly typed in terms of the kinds of objects that it supports. For example, the vector of state unknowns x is strongly typed in that it has to be set as a Vector object and not any other type of object. However, from a mathematical prespective, the ModelEvaluator is weakly typed in terms of the problem formulation that it represents. The unknowns and the functions

supported by a ModelEvalutor object must be interpreted by the algorithms it is used with. For example, a model that supports the evaluation of a state residual *f(x)* could be interpreted as a set of steady-state nonlinear equations *f(x) = 0* or could be interpreted as the right-hand-side of an explicit ODE of the form $\dot{x} = f(x)$. It is up to the users in setting up the various objects and solvers to make sure that ModelEvaluator's unknowns and functions are interpreted correctly in the creation of the solvers. Such is the burden of weak typing.

To demonstrate the power and flexibility of the Thyra::ModelEvaluator design, consider the strongly coupled multi-physics problem in **Error! Reference source not found.Error! Reference source not found.**. One can use the COMPOSITE [46] design pattern to create an abstract combined model as shown in Figure 14.

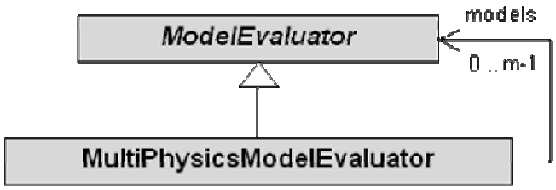

Figure 14: A COMPOSITE ModelEvaluator subclass for a multi-physics set of models.

Each arbitrary individual ModelEvaluator 'model' object in Figure 14 represents a single-physics or coupling set of model equations such as $f_0(...)$, $f_1(...)$, $h_0(...)$, and $h_1(...)$ shown in **Error! Reference source not found.Error! Reference source not found.** while the MultiPhysicsModelEvaluator class represents the combined abstract model shown in **Error! Reference source not found.Error! Reference source not found.**. This approach also allows each individual model to present its own specialized single-physics specific preconditioner and various strategies for the overall multi-physics preconditioner can be configured for each specific category of MultiPhysicsModelEvaluator object related to each specific category of multi-physics problem. Examples of approaches for constructing these COMPOSITE multi-physics preconditioning strategies are given in [38].

The ModelEvaluator is being adopted by a number of different research and development efforts including a system of systems UQ framework, and various multi-physics coupling projects.

The goal of adopting the ModelEvaluator for the different basic physics models is to provide the flexibility to compose and combine them in various ways using a toolbox of numerical methods that are designed to interact with models expressed as ModelEvaluator objects. This would therefore allow the rapid development of various operator-split methods as well as different fully implicit methods using physics-based preconditioners. The set of solvers in Trilinos that can accept ModelEvaluator objects includes NOX, LOCA, Rythmos, and MOOCHO. A particular toolbox of algorithmic approaches for solving transient multi-physics problems that are being developed in the Trilinos package Rythmos are described in the next section.

C.3.2 Rythmos transient solver and sensitivity toolkit

Once the different single-physics models have been implemented and exposed as ModelEvaluator objects, one can then quickly compose a variety of solution methods given a toolbox of solver components. One should be able to quickly and efficiently experiment with everything from operator split approaches to

fully implicit methods with off-the-shelf and custom physics-based preconditioners. The Trilinos package Rythmos is being developed to create such a toolbox for solving forward state, forward sensitivity, and adjoint approaches. The design of Rythmos allows for great flexibility in how transient models are solved. Rythmos is built on the foundation of Thyra and the ModelEvaluator. Figure 15**Error! Reference source not found.Error! Reference source not found.** shows part of the design of Rythmos. The fundamental building block is the StepperBase interface which is used for taking single time steps. There are several different implementations of steppers in Rythmos (e.g. implicit and explicit RK, implicit BDF, etc.) and it is relatively straightforward to add new stepping algorithms. Implicit stepper classes allow for overriding the nonlinear solvers and linear solvers used to the maximum extent. Many of the stepper implementations include automatic local-truncation-type error control algorithms for variable time steps. There is a very flexible DefaultIntegrator subclass that allows for customization of how the time steps are selected and how the output for the algorithm is handled.

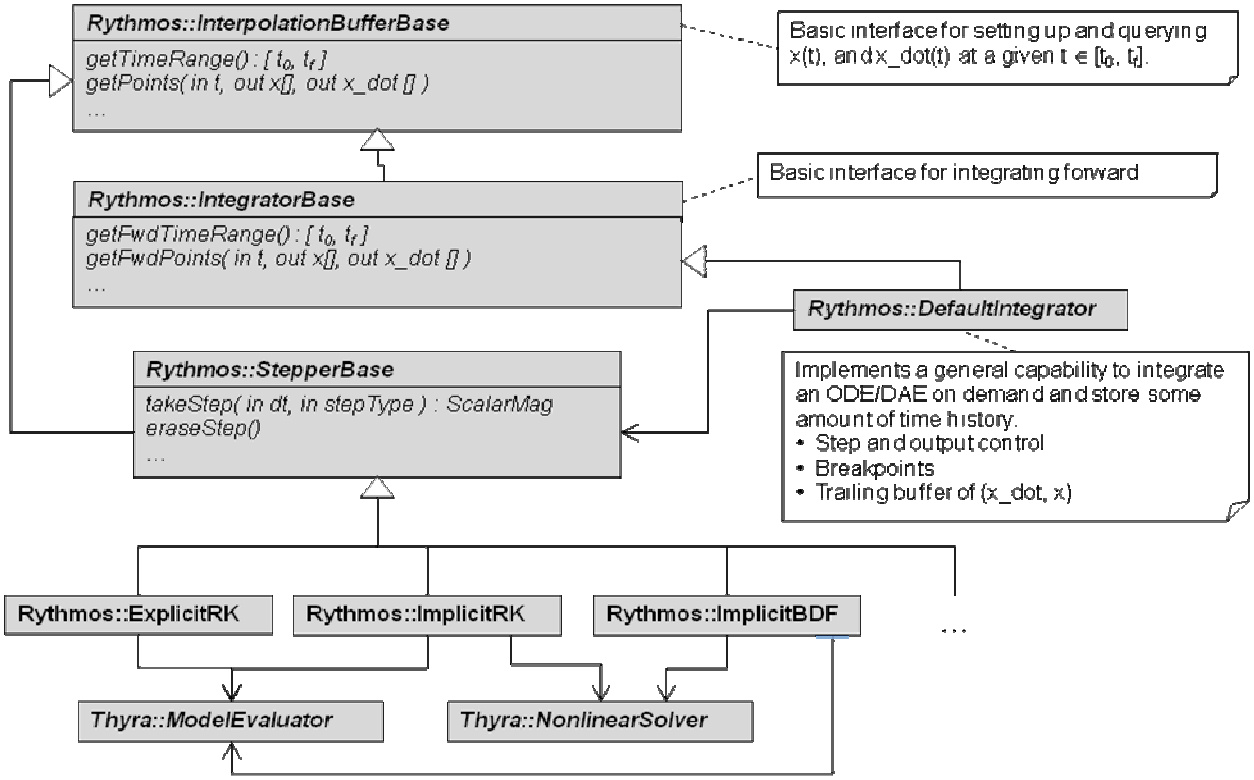

Figure 15: Basic design of Rythmos for solving transient models.

Rythmos continues to undergo active development and has been or is being incorporated in many production simulation codes at Sandia National Labs including Charon, Xyce, and SIERRA.

The design of Rythmos provides a toolbox for constructing various time integration methods including operator-split methods with mixes of explicit and implicit time integration solves.

Another framework for constructing multi-physics transient solvers is embodied in SIERRA's Solution Control module. In this case, the solution of systems is constructed using a more procedural approach where a single-physics model solve is initiated and then other single-physics solves are initiated and cycled as needed until there is converge in a time step. Time lagging of different physics model solves to

decouple the physics is also supported. The SIERRA Solution Control approach does not directly provide any support for fully implicit solves for multi-physics problems. Therefore, the current SIERRA approach is limited to nonlinear Gauss Seidel methods which are known to perform poorly and difficult to converge for many strongly coupled problems [35].

The Rythmos approach is to expose the mathematical structure of the models being solved and therefore allows for the potential of constructing fully implicit multi-physics solves using any number of preconditioning strategies and computing forward and adjoint sensitivities. Embedded in the model-based Rythmos approach is the concept of GS types of Solution Control approaches but Rythmos is not limited to this.

C.4 Dakota

DAKOTA stands for Design Analysis Kit for Optimization and Terascale Applications and provides algorithms for optimization, uncertainty quantification with sampling, and sensitivity analysis, among others. Their website is http://www.cs.sandia.gov/DAKOTA/. The most common and supported use model is to run a controller DAKOTA process that communicates parameters to a separate simulation program through a file, and the response metrics are communicated through a file back to DAKOTA.

Software Categories: VU tools

C.5 ITAPS

ITAPS stands for Interoperable Technologies for Advanced Petascale Simulations and is part of the DoE SciDAC program [???]. Their web site is http://www.tstt-scidac.org/. The charter is to deliver interoperable and interchangeable mesh, geometry, and field manipulation services that are of direct use to SciDAC applications. Interfaces are being refined and developed to provide data structure neutral access to mesh, geometry, and field information, which is key to interoperability.

The primary interface is iMesh, which defines an interface for accessing mesh geometry and topology of an arbitrary mesh implementation. Algorithms built to use the iMesh access functions can then operate on multiple mesh implementations. Interface functions are being refined and developed to include solution field data and parallel processing capabilities.

Concrete component services most relevant here include:

1. iMeshIO Library: Provides access to a wide variety of scientific data by using the iMesh interface to populate an in-memory mesh implementation or to write an in-memory mesh to a data format.

2. Dynamic Parallel Data: Provides parallel applications the ability to distribute or redistribute the finite elements onto the available processors so that the work load on each processor is approximately equal (balanced). The implementation is built on Sandia's Zoltan load balancing toolkit.

3. VisIt Plugins: Provides the capability to visualize data from iMesh implementations by plugging into the VisIt visualization application.

Software Categories: PDE modeling and solvers

C.6 CUBIT

The CUBIT tool suite is licensed to SNL and has a small fee for its use. The web site is http://cubit.sandia.gov. They have a large customer base and the team is actively involved in research as well as production coding. The main product provides a GUI for constructing three dimensional (unstructured) geometries and for mesh generation of those geometries. There are also tools for mesh verification, adaptive mesh generation, and others.

Software Categories: PDE modeling and solvers, Pre- and post-processing

C.7 CCA

The Common Component Architecture is a standard for component-based software engineering used in high performance computing. The designers of CCA are members of the Common Component Architecture Forum, with site http://www.cca-forum.org/.

It is worth noting that the Center for Component Technology for Terascale Simulation Software (CCTTSS) is dedicated to the development of a component-based software development model suitable for the needs of high-performance scientific simulation, particularly the CCA. This effort includes development of a suite of scientific components.

Software Categories: Workflow management, Multi-physics coupling, PDE modeling and solvers, Pre- and post-processing, Numerical solvers

C.8 GoldSim

GoldSim is a commercial, Windows only GUI program which was used by the Yucca Mountain project at Sandia for combining models and running Monte Carlo simulations. Their web site is http://www.goldsim.com/. Indications from the Yucca Mountain team indicate that it may not be sufficient for the demands of NEAMS WF IPSC.

C.9 WIPP Baseline Software List

Code Name	Code Version	Description
ALGEBRACDB	2.35	The software performs most of the algebraic manipulations used to setup data transferred between modeling codes. This normally entails changing units, decomposing vectors to appropriate components, integrating over-time results at specified boundaries, and deleting redundant data. With ALGEBRACDB, an analyst can generate pertinent data external to a code by combining data already stored in a CAMDAT database (CDB) rather than by modifying a code, thereby avoiding the need for a new quality assessment on the code.
BRAGFLO	6.0	The software is used to model two-phase (brine & gas), three-dimensional isothermal flow in porous media.

Code Name	Code Version	Description
CCDFGF	5.02	The software combines all the various releases, for all the scenarios, vectors, and so forth, to construct complementary, cumulative-distribution functions (CCDFs). The software assembles results obtained from calculations from the principal physical models into CCDFs. It scales BRAGFLO_DBR's and SECOTP2D's results to match radionuclide outputs calculated by NUTS and PANEL. The code combines all the calculated release data to simulate many different repository histories and produces the CCDF plots that summarize the WIPP's predicted performance.
CUTTINGS_S	6.02	The software is used to estimate the quantity (in Curies) of waste brought to the surface as a result of an inadvertent borehole drilled directly over the WIPP repository so as to penetrate a waste panel.
DRSPALL	1.10	The software is to calculate the volume of waste subject to material failure and transport during an inadvertent drilling intrusion of WIPP. The code uses text-formatted input and output files, and calculates coupled repository and wellbore transient compressible fluid flow before, during, and after the drilling intrusion process. Mathematical models are included of bit penetration, multi-phase flow in the well, fluid expulsion at the surface, coupling of the well and the repository, repository spalling (tensile) failure associated with fluidized bed transport, and repository internal gas flow. The wellbore model is one-dimensional linear, and the repository model is one-dimensional either spherical or cylindrical.
EPAUNI	1.15A	The software is a modeling code used to estimate the spatial probability distribution of radioactivity (expressed in EPA Units) for the transuranic waste in the WIPP. It builds a data set for the probability distribution for the volumetric EPA Unit (the EPA Unit divided by the total volume of the waste being considered) for each of the Contact Handled-Transuranic (CH-TRU) waste streams and for the WIPP-Scale Remote Handled-Transuranic (RH-TRU) identified in the Transuranic Waste Baseline Inventory Database. The program also generates auxiliary output data files that are used as a quality check on the computations performed by the subroutine. The purpose of the program was to concur with EPA standards that address the management and disposal of transuranic radioactive waste.
FMT	2.4	The program is used to calculate chemical equilibrium in high-ionic-strength geochemical systems at 25 degrees C.
GENMESH	6.08	GENMESH is the principal mesh-generation code used in WIPP performance assessment (PA). The program produces rectilinear, finite-element or finite-element difference meshes that are right-handed, Cartesian, rectangular, and one-, two-, or three-dimensional. User input files define the exact geometry. In addition to setting the node coordinates and mesh connectivity, the code sets material regions, identifies (flags) nodes or elements for boundary conditions, and sets the elevation of elements. The program outputs its results in a computational database (CDB) file.
ICSET	2.22	The software sets initial conditions, specifically the database analysis array variables (history, global, nodal, and/or element variable values) at the first time step. It obtains the values from a user file. In addition, any nodal or element variables (existing or new) can be linearly interpolated by specifying interpolation tables in the program input text file.

Code Name	Code Version	Description
LHS	2.42	The software generates samples from the distributions developed in the first step of a Monte Carlo analysis. Latin hypercube sampling (LHS) ensures a representative coverage of the full range of each variable. Once cumulative distribution functions have been developed for each parameter to be sampled, LHS methods are used to generate a sample. LHS has a number of desirable properties:·1) Full coverage (stratification) across the range of each variable (extremes as well as midpoints), 2) Relatively small sample sizes, 3) Direct estimation of means, variances, and distribution functions, 4) Availability of a variety of techniques for sensitivity analysis, and·5) Possible to determine the effects of different distribution for the input variables on the estimated distribution for and output variable without rerunning the model
MATSET	9.10	The software sets material names to specified regions (e.g., defined by GENMESH), sets material property values, and sets attribute values into the performance-assessment computational database. Both property and attribute values are obtained from either the property secondary database, or the user-supplied MATSET input text file.
NUTS	2.05C	The Nuclides Transport System (NUTS) code is a multidimensional, multicomponent radionuclide/metal contaminant transport, single-porosity (SP), dual-porosity (DP), and dual-permeability (DPM) five-point finite-difference simulation model. The model simulates first-order radioactive chain decay during radionuclide transport. The program is not limited to radioactive material transport, and any non-radioactive materials can be included. Three types of sorption isotherms are considered to represent ion exchange between the waste and the surrounding formation: linear, Freundlich, and Langumir equilibrium isotherms. Hydrodynamic dispersion is modeled with the assumption that the dispersivity corresponds to an isotropic porous medium.
PANEL	4.03	The software is a radionuclide mobilization and decay code. Its principal functions are to decay the inventory and to estimate the quantity of all modeled radionuclides that are transported.
PCCSRC	2.21	The software is used to evaluate parameter importance by reporting the partial correlation coefficients (PCC's) and standardized regression coefficients (SRCs) on either the raw or ranked data.
RELATE	1.43	The software is used to interpolate data from one coordinate grid to a different coordinate grid that overlies it, and to combine two data files that are defined over the same grid.
SANTOS	2.1.7	The software is a finite element program designed to compute the quasi-static, large deformation, inelastic response of two-dimensional planar or axisymmetric solids or engineering structures. The code is derived from the transient dynamic code PRONTO 2D. The solution strategy used to compute the equilibrium states is based on a self-adaptive dynamic relaxation solution scheme, which is based on explicit central difference pseudo-time integration and artificial mass proportional damping. The element used in SANTOS is a uniform strain 4-node quadrilateral element with an hourglass solution scheme to control the spurious deformation modes. Finite strain constitutive models for many common engineering materials are available. A robust master-slave contact algorithm for modeling sliding contact is implemented. An interface for coupling to an external code is also provided.
SECOTP2D	1.41A	The software performs single- or multiple-component radionuclide transport calculations in fractured or granular aquifers. Fractured porous media are represented through a dual-porosity model.
STEPWISE	2.21	The software is used for stepwise regression analysis.

Code Name	Code Version	Description
SUMMARIZE	3.01	The software is used to combine multiple CAMDAT databases (CDB) (potentially hundreds) into one output file used by CCDFGF. The files are saved and then combined in the CCDFGF preprocessor.

C.10 YMP Baseline Software List

Code Name	Code Version	Description
2KGRID8.FOR	1.0	The software generates dual-permeability grids for the TOUGH2 family of codes and allows adjustment of input file field sizes.
AMESH	1.0	The software is used to generate discrete 1-D, 2-D or 3-D grids for numerical modeling of flow and transport problems in which the formulation is based on the integral finite difference method.
ANSYS	5.6.2-01	The software is used to perform thermal, mechanical, seismic and coupled thermo-mechanical analyses for waste package.
ASHPLUME_DLL_LA	2.1-01	The software is used to estimate the distribution of ash and waste fuel released into the atmosphere during a hypothetical volcanic event intercepting the Repository.
CWD	2.0-00	The software routine calculates the probability distributions for the occurrence and size manufacturing defects in the closure weld of waste packages given the probability for the non-detection of defects and the fraction of defects to be considered.
DIFFCELL	2.0	The software provides a numerical solution to an equation describing one-dimensional diffusive transport through a rock wafer with time-dependent concentration boundary conditions.
DIRECT	4.0	The software is used to estimate the numbers of waste packages damaged by both intrusive and extrusive igneous events under stochastically developed parameter inputs.
DISCRETE_TF	1.10	The software is a FEHM output data post processor used to generate transfer functions curves from discrete fracture simulation data generated by FEHM simulations of solute breakthrough recorded in the ".trc" file.
DRKBA	3.31	The software performs an analysis of the possible formation of key rock blocks based on the orientations of discontinuities inherent in the rock mass in waste emplacement drift.
EARTHVISION	5.10	The software is used to create 3-D models of geologic features.

Code Name	Code Version	Description
EMSA	1.0	EMSA (Explicit Multiple Scale Algorithm) software simulates in two dimensions, the vertical propagation of a dike driven by incompressible or compressible magma. The emphasis is on a compressible magma containing dissolved volatiles because a hypothetical magmatic intrusion at Yucca Mountain is expected to have a H_20 and CO_2 content such that these volatiles would exsolve from the magma during its ascent. EMSA also addresses horizontal confining stresses that may vary with depth.
EQ3-6	8.10	This software is used for geochemical modeling of equilibrium interactions between aqueous species and minerals, specifically to model the compositional evolution of fluids in the in-drift environment under various relative humity (RH) conditions.
EXDOC_LA	2.0	The software is used to analyze the TSPA_LA model and support the associated documentation. This updated version provides the following additional features: 1) ability to calculate CCDF via a Monte Carlo method, 2) improve interpolation, 3) ability to address the nominal early WP failure case, and 4) ability to add uncertainty to seismic hazard.
FAR	1.20	The software is used to evaluate the redistribution of volcanic ash and associated radionuclides within the Fortymile Wash drainage area away from the Yucca Mountain repository. It is also used to model redistribution of radionuclides in soil at the location of a reasonably maximally exposed individual (RMEI) on the Fortymile Wash fan.
FEHM	2.26	The software is based on finite-element heat and mass-transfer code which simulates non-isothermal multiphase, multicomponent flow and solute transport in porous media. The FEHM code is also used for parameter sensitivity studies in the design and specification of field tracer and flow experiments and the interpretation for those field experiments.
FEHM2POST	1.0	The software is a set of perl scripts used to automate the repetitive series of steps required to make multiple runs of FEHM and post-process the output data.
FLAC	4.04	The software is used for performing ground control design analyses in order to assess the stability of both emplacement and non-emplacement openings and needs of ground supports.
FLAC3D	2.14	The software is used to simulate the behavior of three-dimensional structures built of soil, rock or other materials subjected to various loads.
FLOW_CON	1.0	The software is used to convert TOUGH2 flow input files into files readable by FEHM (with 5-character element).
FRACMAN	2.512	The software is used to provide model of geometry of discrete features, including faults, fractures, paleochannels, karsts, and stratigraphic contacts.
FRACWORKSXP	1.0	This software is used to simulate and analyze discrete fracture networks. Characterization of the repository lithostratigraphic units within the repository footprint involves determination and estimating uncertainties of fracture network properties. This software aids in these determinations.

Code Name	Code Version	Description
GENHSUMODELDATA	1.0	This software takes distributions for solid thermal conductivity, matrix porosity and Hsu model parameter, and determines the mean and standard deviation for the porosity and dry and saturated matrix thermal conductivities.
GET_TEMP_V0.F	1.0	The software is used to process data for thermal site scale model for Yucca Mountain.
GETEQDATA	1.0.1	The software is a post-processor macro that greatly enhances EQ3/6 user productivity, data reliability, and reproducibility through the automation of output information extraction.
GETEQPHASES	1.0	This software is an excel macro that scans large PHREEQC outputs, extracts the relevant information (saturation index of selected minerals containing selected chemical elements), and processes the results (computes statistics).
GETTHK_LA	1.0	The software runs in Yucca Mountain Repository TSPA GoldSim models. The software extracts remaining thickness for the waste package inner barrier from the THK file produced by the WAPDEG code. The thickness information is returned to the calling GoldSim model as statistics for each time step across all packages.
GMFIX	1.61	The software is to model multiphase, high-speed flow resulting from eruption of magma accompanying future volcanism at Yucca Mountain. The code is used to simulate the expansion of magma from a feeder intrusion dike into repository drift.
GOLDSIM	9.60.300	GoldSim is an acquired 32-bit Windows-based program that: 1) Quantitatively addresses the inherent variability and uncertainty that is present in real-world systems using Monte-Carlo simulations; 2) Superimposes the occurrence and consequences of discrete events onto continuously varying systems; 3) Builds top-down models using hierarchical containers that facilitate simulation of large, complex systems while keeping them easy to understand and navigate; and 4) Dynamically links external programs or spreadsheets directly to the GoldSim model.
HAZUHS	1.0	This software reads in a suite of seismic hazard curves at specified structural frequencies and computes corresponding uniform hazard spectra (UHS) for given annual probabilities of exceedance. The software interpolates the annual number of seismic events and ground motion values.
INFIL	2.2	The software performs surface water flow routing and simulation of daily mean discharge, estimates snow fall, snow melt, and sublimation. The code estimates root-zone layering and the extension of the root-zone into the bedrock based on soil thickness.
INTERPZDLL_LA	1.0	The software is used to provide interpolation capabilities for the Physical & Chemical Environment submodel of the total system performance assessment License Application (TSPA LA) model for the Yucca Mountain Repository. It implements a 4-point linear interpolations that can accommodate any one-point missing.
ITOUGH2	5.0	The software performs inverse modeling for TOUGH2 - a numerical code for multiphase flow and transport through automatic calibration.

Code Name	Code Version	Description
KAPPAAH	1.0	This software is used to estimate stress drip, moment and other parameters related to spectral analysis of seismic data. This software extracts seismic waveforms, measures, and tabulates spectral fitting parameters.
KM_IMPACTS_PP	1.0	This post-processor processes output from kinematic analyses of waste packages (WP) and the emplacement pallet. and their interactions with the drip shield and invert for the Yucca Mountain repository analysis, as determined through the use of the LS-DYNA software code. Because a large number of analyses are generally performed (e.g. 1000 realizations for a given scenario) the software is used to automate the calculation of waste package damage estimates.
KREG	1.1	The software is used to create or modify some of the thermodynamic data incorporated into the thermodynamic database of the reactive transport code TOUGHREACT. This version has flexible input formats. The software is used to calculate regression coefficients of log K data as a function of the temperature.
KSWITCH	1.1	This utility routine is used to upgrade the thermodynamic database of TOUGHREACT. TOUGHREACT simulations are used to validate the Drift-Scale Test (DST) THC Model against water chemistry, gas chemistry, and mineralogical data collected during the heating and cooling phases of the DST.
LAGRIT	1.1	The software is used for 3-dimensional finite element and finite volume mesh generation. Specifically, it is used to create some of the 3-D computational meshes for saturated and unsaturated zone flow and transport calculations at the Yucca Mountain site.
LHS	2.51	The software performs Latin Hypercube and Monte Carlo sampling. This activity increases the number of points for the density function from 1000 to a user defined number.
MCNP5	1.4	MCNP5 is a general purpose Monte Carlo radiation transport code used to simulate neutron, photon, electron, or coupled neutron/photon/electron transport, and includes the capability to calculate eigen values for critical systems. The code models an arbitrary three-dimensional configuration of materials in geometric cells bounded by first- and second-degree surfaces and fourth-degree elliptical torii.
MINACC	1.0	The software is an excel macro that extracts relevant information (precipitation history of each mineral) from PHREEQC output files and processes the results (computes volumes).
MKTABLE_LA	1.0	The software is a DLL that runs in the total system performance assessment (TSPA) model for the Yucca Mountain repository. It processes the environment history files that are input to the waste package degradation simulation model, WAPDEG DLL.
Modflow-2000	1.17.02	The software is used to simulate steady state and transient conditions for the Death Valley regional scale ground-water flow system encompassing Yucca Mountain, Nevada and steady-state flow for the defined site scale region.

Code Name	Code Version	Description
MSTHAC	7.0	This software routine develops time histories of variables of interest at various locations within the Engineered Barrier System. This is done by developing relationships between TH models with varying levels of detail, model domain, and processes models.
MULTRAN	1.0	The software is a two-dimensional numerical model that uses an implicit-in-time, alternating-direction finite-difference method to solve the equations describing multi-component transport of sorbing and non-sorbing solutes in a dual-porosity medium.
MVIEW	4.0	The software is used to transform test data describing numeric model geometry and numeric model output into 2D and 3D visual representations.
NPHF2D	1.0	The software performs numerical modeling supporting analysis of magmatic dike propagation and analysis of the dike-drift interaction where magma enters a drift.
NUFT	3.0S	The software is used to provide thermal-hydrological modeling of unsaturated zone systems.
PARTICLE_STAT	1.0	This software is used to (1) compute statistics of travel time (to water table) for particles released at all repository nodes, including statistics as functions of number of particles, and particle statistics for all nodes in each 5 bins and normalized concentration computed from travel time distribution; and (2) compute the travel time statistics for particles released at any particular node with a number of Kd and Dm values.
PHREEQC	2.11.01	The software performs a wide-variety of aqueous geochemical calculations including speciation and saturation-index; reaction-path and advective-transport involving specified irreversible reactions and inverse solution pathway modeling.
PHREEQC_POST	1.10	The software is an excel macro that processes PHREEQC outputs, extracts relevant information (precipitation history of each mineral), and computes yearly mineral accumulations.
POST_RASCAL	1.0	The software is used to develop seismic inputs for preclosure design and analysis and postclosure analyses of the Yucca Mountain repository. This software includes a suite of post-processing modules (PARINP, LOGNORM, NORM, SMRATIO, and SPMEAN) that compute various statistical parameters from the output of the software RASCAL SET V1.0. This code is acquired software from Pacific Engineering & Analysis, El Cerrito, CA. The code is written in FORTRAN 77 and runs on an IBM compatible PC with: DOS V6.22, QEMM V9.0 and RUN386.EXE (Phar Lap run time DOS extender).
PPPTRK	1.0	The software is a FEHM output data post-processor used to produce breakthrough curves from particle tracking simulation data.
PREWAP_LA	1.1	The software extracts data from various tables and reformats it as output that is then used as input to the waste package degradation simulation software WAPDEG. This version allows input of percolation rate data for an additional climate state necessary for peak dose evaluation, and provides output of the data associated with the additional climate State.

Code Name	Code Version	Description
RADPRO	4.0	The software routine is GUI driven and used to calculate radiation coefficients between grid locks for a 2 dimensional or 3 dimensional grid and output this information in a format compatible with the NUFT.
RASCAL_SET	1.0	The software is used to develop seismic inputs for preclosure design/analysis and postclosure analyses. This software includes a suite of modules (RASCALS, RASCALP, RANPAR, VELAVG and SCP) that computes seismic design site response factors.
RASCALP	2.0.2	The program generates synthetic time histories (acceleration, velocity, and displacement) by computing the Fast Fourier Transform (FFT), by extracting the phase of an input accelerogram and combining this phase with the computed and filtered source Fourier amplitude spectrum to generate the output time histories. Site response for inclined P-SV waves may be computed for a single- or multilayer profile using either linear or equivalent-linear frequency-domain RVT.
RASCALS	5.4	The software calculates a source Fournier amplitude spectrum and acceleration and response spectral velocity by using random vibration theory (RVT) techniques. The code addresses horizontal ground motions.
RATEDENS	1.0	The software is used to generate calculations of rate density of potential future volcanic events near the proposed Yucca Mountain nuclear waste repository as part of a probabilistic volcanic hazard assessment (PVHA-U).
RELAP	2.0	The software models tracer transport by convoluting a Laplace-domain transfer function for transport through dual-porosity media with transfer functions that describe tracer injection, mixing in the injection and production wellbores (or flow manifolds in laboratory experiments) and recirculation of the product fluid (in field experiments only). It also performs curve-fits to field or laboratory tracer test data to obtain the best-fitting transport parameter values.
RETRAN	2.0	The software models reactive transport in dual-porosity media with a general, nonlinear sorption isotherm and with time-varying flow rates.
SAC	00.46	The software is used as a general purpose processing program for seismological data analysis. It provides filtering, spectral calculations, signal timing, and manipulation of 3-component recordings.
SAPHIRE	7.26	This (acquired) software is a probabilistic risk and reliability tool used for probabilistic risk assessment (PRA). The software is used to construct and analyze fault trees and event trees based on waste package design basis events for the risk and criticality analysis.
SCALE	5.1	This software performs light water reactor fuel depletion analyses, reactivity sensitivity analyses and radiation transport calculations. This software is used for designing benchmark experiments, performing fuel depletion evaluations and radiation calculations.
SCCD	2.01	The software is used to calculate uncertainty in stress and stress intensity versus depth in waste package.

Code Name	Code Version	Description
SEEPAGEDLL_LA	1.3	This software calculates seepage into the drifts across the repository. This version provides the same features/functionality as the previous version; it also allows input of percolation rate data for an additional climate state necessary for peak dose evaluation and provides output of the data associated with the additional climate state.
SOILHAZ_SET	1.0	The software is used to develop seismic inputs for preclosure design and analysis and postclosure analysis. This software includes a suite of modules (SOULUHS, SOILUHSI, FRACTILE, SUHSINP, and HCSCP) that computes soil hazard curves and uniform hazard spectra from rock hazard curves and strain-compatible soil properties.
SUPCRT92	1.0	The software calculates the standard molal thermodynamic properties of minerals, gases, aqueous species and reactions from 1 to 5,000 bars and 0 to 1,000 degrees Centigrade.
SZ_CONVOLUTE	3.10.01	The software is used in the total system performance analysis (TSPA) analyses for the proposed Yucca Mountain repository to calculate the radionuclide mass flux at the water table beneath the repository.
SZ_POST	3.0	The software is used to translate the output files from the saturated zone (SZ) site-scale model into the format used by the SZ_Convolute software code, which is used in the total system performance assessment (TSPA) analysis for the proposed Yucca Mountain repository. SZ_Post reads the output files from the FEHM software code and writes the breakthrough curve data for radionuclide transport in the SZ.
SZ_PRE	2.0	The software is used to generate the input files for the saturated zone (SZ) site-scale model, which simulates radionuclide transport for use in the total system performance assessment (TSPA) analysis for the proposed Yucca Mountain repository. SZ_Pre reads the uncertain parameter values for key SZ site-scale model parameters and writes these parameters in the appropriate file format for the FEHM software code.
T2FEHM	4.0	The software is used to convert TOUGH2 files into files readable by FEHM.
T2R3D	1.4	The software is used as a radionuclide transport program based upon TOUGH2 (unsaturated zone model). It was developed to handle transport of sorbing and non-sorbing tracers in fractured media.
TOUGH2	1.6	The software is used as an integral finite difference numerical simulator for non-isothermal flows of multi-component, multiphase fluids in porous and fracture media. Changes were made to the code parameter file (tough2.prm) in order to accommodate simulations with a larger number of elements.
TOUGHREACT	3.1.1	This software is a numerical simulator for non-isothermal flow and transport coupled with reactive chemistry. This version incorporates vapor-pressure lowering effects on TH and THC calculations, and provides additional improvements for mineral precipitation in boiling fractures.
TRANSL	2.0	The software is written in C and performs the translation of the EQ6 thermodynamic database from the EQ6 format to the PHREEQC format.

Code Name	Code Version	Description
WAPDEG	4.07	The software is used to stochastically simulate waste package and drip shield degradation for use in the total system performance assessment (TSPA) for the proposed Yucca Mountain repository.
WINGRIDDER	2.0	The software is used to generate 1-D, 2-D or 3-D grids for numerical modeling of flow and transport problems based on the integral finite difference method.
XTOOL	10.1	The software is used to perform post-processing of NUFT output. It also provides a graphical visualization capability to NUFT output.
YMESH	1.5	This software is an interactive program developed as an input generator for Nonisothermal Unsaturated-saturated Flow and Transport (NUFT) model.