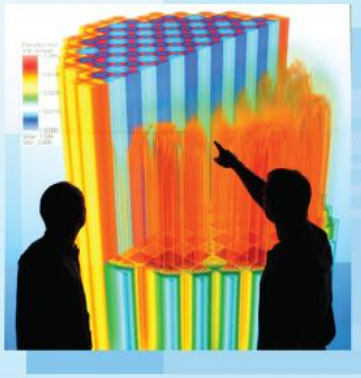
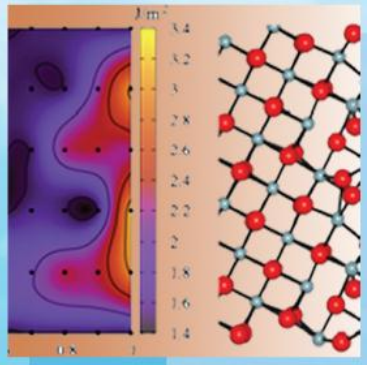
CASL-U-2011-0017-000-b

Power uprates
and plant life extension



Engineering design
and analysis

Science-enabling
high performance
computing



Fundamental science

Plant operational data

Consortium for Advanced Simulation of LWRs

L2:VRI.P2.02

John Turner
ORNL

Completed: 3/31/11

U.S. DEPARTMENT OF ENERGY | Nuclear Energy

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Technical Note

Reactor and Nuclear Systems Division
Radiation Transport Group

From: Tom Evans, Greg Davidson, Josh Jarrell
Number: RNSD-TN-11-004
Date: March 31, 2011

Subject: Design of a Neutronics Package for Multi-Physics Reactor Calculations (Rev. 1)

Executive Summary

This document lays out the code design requirements for neutronics in a coupled physics code. The fundamental methods and models are defined, a general domain model for nuclear reactor neutronics is developed, and a specific object model for CASL's VERA code is built.

1 Introduction

This document defines the general code design for a neutronics package for nuclear reactor calculations in a multi-physics environment. A multi-physics reactor simulation code requires many physics components as illustrated in Fig. 1. This document only describes the neutronics piece. Furthermore, we only consider quasi-static coupling and multigroup energy discretization. We do not attempt to provide highly detailed

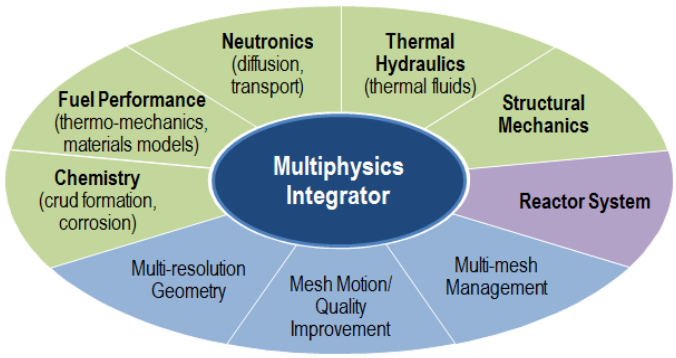


Figure 1: Multiphysics components in a reactor simulation code.

object models for each component in the neutronics package. Instead, this document is intended to serve as a roadmap that defines the interaction of the various pieces of the neutronics package at the component level. Particular emphasis is placed on defining a neutronics *domain model* that illustrates the data flow between the various parts of the calculation. Detailed class and sequence models will be provided in the documentation for each individual component.

VERA and the neutronics package documented in this report target HPC platforms. Nonetheless, except in a few cases, we do not comment on the parallel computing requirements of any piece of the package. Most of the parallel coupling in the neutronics package takes place in the transport solve. There is little

Figure 2: Input/Output data in the neutronics package.

to no processor-to-processor coupling in the cross section generation or depletion steps; thus, an overview of the code design of the neutronics package is not highly concerned with the parallel decomposition. In all that follows we assume that the space-energy decomposition of the problem is dictated by the transport component requirements and that the cross section and depletion pieces simply are applied locally to each domain.

In § 2 we define the mathematical models that are solved in the neutronics package. These equations help define the problem space and the data inputs and outputs. A domain model that solves this system of equations is postulated in § 3. The domain model is not specific to any single multi-physics framework. In § 4 we define an object model for VERA that follows the domain model. The object model explicitly defines the use of SCALE [7] for cross section generation and depletion. The transport code systems Denovo [1] and DeCART [5] are targeted for the transport component.

2 Methods and Models

This section is not meant as a comprehensive methods guide; it simply presents the fundamental equations that must be solved in a multigroup reactor neutronics package so that they may be further discussed using a common frame of reference. A reactor neutronics package can be fully time-dependent or quasi-static. In the quasi-static approximation the neutron density and/or power is assumed to be constant during the timestep. In this document we only consider requirements for quasi-static neutronics. The fully-dependent treatment has significantly different requirements for coupling the state variables as the neutron density is a state variable in this case. Additionally, rigorous time-dependence requires solution of the reactor kinetic equations, which are not considered in the quasi-static case. Therefore, we defer documentation of the fully time-dependent case to a future report.

The multigroup neutronics equations in the quasi-static approximation can be described in three categories:

Cross Section Processing Creating *macroscopic* cross sections that will be used in the transport equation.

Transport Determining the space-energy distribution of neutrons in the reactor so that power and depletion may be calculated.

Depletion Calculate burnup of fissionable material and the introduction of poisons in the reactor core.

The variable inputs and outputs for the neutronics package are illustrated in Fig. 2. Here the inputs T and N are the space-dependent temperature and atomic concentration of the isotopes (nuclides) in the problem, respectively. The outputs are updated atomic (nuclide) concentrations and power in the reactor core. These quantities get coupled to other physics in the simulation, namely flow, heat transfer, and chemistry among others. Each material in the problem has an associated physical density, ρ, in grams per cm^3, and atomic density, N, in atoms per cm^3. The atom density of a material is the sum of its constituent components,

$$N = \sum_j N_j \ . \tag{1}$$

There are many methods for calculating the N_j components depending on the input data. For nuclear reactor applications mixtures are generally defined in *weight percent* (w/o) so that the atom density is

$$N_j = 0.01(w/o)\frac{\rho N_a}{M_j} \ , \tag{2}$$

where N_a is Avogadro's number and M_j is the atomic mass of the component. Sometimes the materials are entered using *atom percent* (a/o) requiring,

$$N_j = 0.01(a/o)\frac{\rho N_a}{M} \,, \tag{3}$$

where $M = \sum_j n_j M_j$ is the total atomic mass of the mixture. Reference 6 has more details and examples.

The first step in the neutronics package is *Cross Section Processing*. The objective of the cross section processing component is to produce multigroup *Macroscopic* cross sections for the transport solver. The macroscopic cross section in group g for reaction i is defined

$$\Sigma_i^g(\mathbf{x}) = \sum_j N_j(\mathbf{x})\sigma_{i,j}^g(\mathbf{x}) \quad \left[\mathrm{cm}^{-1}\right] \,, \tag{4}$$

where $\sigma_{i,j}^g$ is the multigroup *Microscopic* cross section for reaction i of isotope j in cm^2. The multigroup microscopic cross sections are calculated by averaging continuous energy data with an appropriate weighting function,

$$\sigma_{i,j}^g = \frac{\int_{E^{g+1}}^{E^g} \sigma_{i,j}(E,T)W(E)\,dE}{\int_{E^{g+1}}^{E^g} W(E)\,dE} = \frac{\langle \sigma_{i,j} W\rangle}{\langle W\rangle} \,. \tag{5}$$

There are many methods for calculating the weight function for a variety of applications. The details of the various approaches are beyond the scope of this document; however, W is usually approximated by some expression of the angular intensity in order to capture the space-energy self-shielding effects of nuclear fuels,

$$\sigma_{i,j}^g = \frac{\langle \sigma_{i,j}\bar{\psi}\rangle}{\langle \bar{\psi}\rangle} \,. \tag{6}$$

In any case, the microscopic cross sections must be calculated each timestep because the weighting function must reflect the changing isotopic concentrations in the fuel. See Refs. 3, 4, and 9 for a more detailed discussion on calculating the multigroup microscopic cross sections.

The *Transport* step of the neutronics package is generally the most expensive part of the calculation in both memory and time. In the quasi-static approximation, we solve the eigenvalue form of the linear Boltzmann equation,

$$\begin{aligned}\hat{\mathbf{\Omega}}\cdot\nabla\psi^g(\mathbf{x},\mathbf{\Omega}) &+ \Sigma^g(\mathbf{x})\psi^g(\mathbf{x},\mathbf{\Omega}) \\ &= \frac{1}{4\pi}\sum_{g'=0}^{G}\int_{4\pi} d\mathbf{\Omega}'\,\Sigma_{\mathrm{s}}^{gg'}(\mathbf{x},\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{\Omega}}')\psi^{g'}(\mathbf{x},\hat{\mathbf{\Omega}}') + \frac{1}{k}\frac{\chi^g}{4\pi}\sum_{g'=0}^{G}\int_{4\pi} d\mathbf{\Omega}'\,\nu\Sigma_{\mathrm{f}}^{g'}(\mathbf{x})\psi^{g'}(\mathbf{x},\hat{\mathbf{\Omega}}') \,.\end{aligned} \tag{7}$$

The independent variables are space $(\mathbf{x})$, angle $(\hat{\mathbf{\Omega}})$, and energy (E) that is represented in the multigroup approximation by the discrete index g. This equation is more recognizable as an eigenvalue equation when it is written in (continuous) operator form,

$$(\hat{L} - \hat{S})\Psi = \lambda\hat{F}\Psi \,, \tag{8}$$

where $\lambda = 1/k$. Here, the notation $\hat{(\cdot)}$ indicates a continuous operator. The fundamental unknown in Eq. (7) is the angular radiation intensity, $\psi = vN_p$, where N_p is the particle density in particles per cm^3 and v is the particle velocity in cm per second[1]. The first angular moment of the angular intensity gives the scalar intensity,

$$\phi^g(\mathbf{x}) = \int_{4\pi} d\mathbf{\Omega}\,\psi^g(\mathbf{x},\mathbf{\Omega}) \,, \quad \Phi = \hat{D}\Psi \,. \tag{9}$$

[1] ψ is often called the *flux* in nuclear engineering literature; however, this terminology can be confusing when dealing with multiphysics CFD coupling so we will continue to use intensity.

The scalar intensity can be used to calculate the power in the reactor core,

$$P = \sum_{g=0}^{G} \int_V d\mathbf{x}\, \kappa^g \Sigma_{\mathrm{f}}^g(\mathbf{x}) \phi^g(\mathbf{x}) \,. \tag{10}$$

The remaining terms in Eqs. (7) through (10) are:

Σ^g	total cross section for group g (cm^{-1})
$\Sigma_{\mathrm{s}}^{gg'}$	scattering from group g' to g through angle $\hat{\mathbf{\Omega}} \cdot \hat{\mathbf{\Omega}}'$ (cm^{-1})
Σ_{f}^g	fission cross section for group g (cm^{-1})
χ^g	fission spectrum for group g
ν	number of neutrons released per fission (neutrons per fission)
κ^g	energy per fission event (MeV per fission)

The pertinent state quantities calculated during transport are power and scalar intensity. The principal edit quantity is the eigenvalue of the system, k, that indicates reactor criticality.

The final step in the quasi-static neutronics package is to deplete the fuel and calculate the fission products that are generated during the timestep. The *Depletion* step requires solving the following equation [2],

$$\begin{aligned} \frac{dN_j}{dt} &= \sum_k \underbrace{\left(\gamma_k \bar{\sigma}_{\mathrm{f},k} \bar{\phi} N_k + \bar{\sigma}_{\mathrm{c},k} \bar{\phi} N_k + \lambda_k N_k\right)}_{\text{production}} - \underbrace{\left(\bar{\sigma}_{\mathrm{f},j} \bar{\phi} N_j + \bar{\sigma}_{\mathrm{c},j} \bar{\phi} N_j\right)}_{\text{destruction}} - \underbrace{\lambda_j N_j}_{\text{decay}} \\ &= \mathbf{A} N \,. \end{aligned} \tag{11}$$

The individual terms in Eq. (11) are defined

$\gamma_k \bar{\sigma}_{\mathrm{f},k} \bar{\phi} N_k$	production of N_j from fission of nuclide N_k
$\bar{\sigma}_{\mathrm{c},k} \bar{\phi} N_k$	transmutation into N_j resulting from neutron capture by nuclide N_k
$\lambda_k N_k$	production of N_j resulting from radioactive decay of nuclide N_k
$\bar{\sigma}_{\mathrm{f},j} \bar{\phi} N_j$	destruction of N_j from fission
$\bar{\sigma}_{\mathrm{c},j} \bar{\phi} N_j$	destruction of N_j from all reactions other than fission
$\lambda_j N_j$	radioactive decay of N_j

In the quasi-static formulation, we assume that the space-energy integrated scalar intensity, $\bar{\phi}$, is constant over the timestep. The same approximation is used for the total microscopic cross sections, $\bar{\sigma}$. A constant power approximation can also be applied to Eq. (11).

We have now defined the fundamental equations that are required for multigroup, quasi-static neutronics. To summarize, the general steps of a neutronics calculation are

1. Calculate cross section weighting parameters ($W(E)$) for the current timestep,
2. Calculate multigroup microscopic cross sections using Eq. (5),
3. Calculate multigroup macroscopic cross sections using the current atomic densities using Eq. (4),
4. Solve the transport equation, Eq. (7),
5. Calculate the reactor power using Eq. (10),
6. Deplete the fissionable nuclides in the reactor core using Eq. (11),
7. Return the new nuclide concentrations and power.

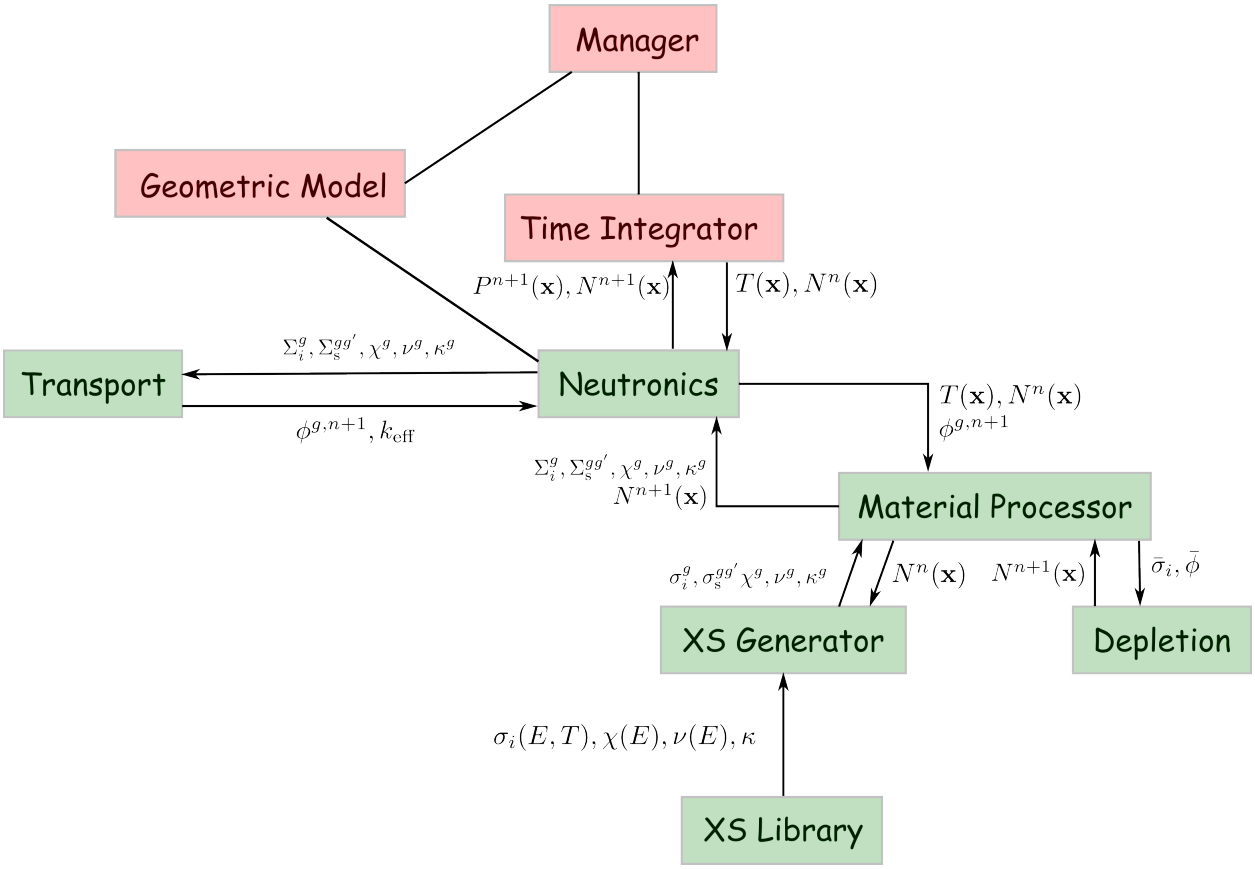


Figure 3: Domain model for a neutronics package in a multiphysics reactor simulation. Red abstractions represent pieces external to the Neutronics component. Green abstractions are part of the Neutronics component.

3 Domain Model

In § 2 the model equations for multigroup neutronics were defined. Now, we can proceed to defining a Domain Model (DM) for a general neutronics package. The DM provides human readable abstractions for the code. The DM is **not** directly transferable to code; instead, the DM provides a bridge between the methods and algorithms and the code model. The code model does have a one-to-one (or nearly) relationship with the eventual code base. In this particular case, the DM provides a valuable mechanism to represent data flow between the constituent parts of the code architecture.

The DM is shown in Fig. 3. As mentioned above, the objective of the neutronics package is to take as input the problem temperatures and atomic densities at t^n and return the power and atomic densities at time t^{n+1}. The DM shows the flow of data through the various components of the neutronics package. We have not shown the supporting data that will be required; this information is encapsulated in the Geometric Model abstraction and includes material ids, problem dimensions, mesh, and other parameters related to the geometry-material model. All components in the neutronics package will have access to the Geometric Model abstraction.

The Neutronics component manages the data flow between the Material Processor and Transport pieces. It also processes return data to the Time Integrator. This includes integrating the scalar intensities to calculate power according to Eq. (10).

The Material Processor abstraction performs two tasks: (1) generates multigroup cross sections for each nuclide, and (2) depletes the reactor fuel and generates fission products. The Transport abstraction uses the macroscopic cross sections generated by the Material Processor and calculates the space-energy distribution of neutrons in the reactor. From this distribution the reactor power generated during the timestep is calculated. The scalar intensity is then passed to the Material Processor so that depletion can be calculated.

The Material Processor contains two abstractions, the XS Generator and Depletion components. Both of these components will require disk I/O to read in fine-group or continuous energy cross sections and material data that are used to calculate the multigroup cross sections. The Depletion component uses both default tabular data and problem-specific data calculated by the XS Generator. The reason that the XS Generator and Depletion components are grouped together under the Material Processor, instead of begin peer-level components with Transport, is that both cross section generation and depletion key off of material id. Thus, these operations naturally fit together under a common material processing component.

The abstractions in the DM can be correlated with the neutronics computational sequence listed at the end of § 2 to give a complete picture of the neutronics package. The resulting flow-of-control is:

1. Neutronics receives temperatures and atomic concentrations (densities) from the Time Integrator,
2. Neutronics passes this data to the Material Processor,
3. The Material Processor passes the atomic densities and temperatures to the XS Generator,
4. The XS Generator reads continuous energy data from the XS Library and calculates multigroup data,
5. Material Processor takes the microscopic multigroup cross sections from the XS Generator and calculates macroscopic multigroup data,
6. The multigroup data is returned to the Neutronics component by the Material Processor,
7. Neutronics passes the multigroup data to the Transport component,
8. The Transport component calculates the space-energy scalar intensities and returns these to Neutronics,
9. Neutronics calculates the reactor power and sends the scalar intensities to the Material Processor for depletion,
10. The Material Processor integrates the scalar intensities and microscopic multigroup cross sections and passes these to the Depletion component,
11. The Depletion component calculates new atomic concentrations and returns these to the Material Processor that returns them to the Neutronics component,
12. Neutronics returns the power and atomic concentrations at t^{n+1} back to the Time Integrator.

These steps are illustrated in the activity diagram shown in Fig. 4. We point out that there may be two different parallel decompositions in the neutronics package, one for cross section generation and one for transport and depletion.

A general DM has now been defined for the Neutronics component. This model is not specific to any single code implementation. However, it can be used as the basis for implementation in a specific code system. We provide a model in the following section for VERA in the CASL project.

4 VERA Code Model

VERA is the virtual reactor code system in the CASL project. It is built on the LIME multi-physics code framework[2]. The objective is to convert the concepts of the DM described in § 3 into a code object model for

[2]See draft report, R. Pawloski et al. "A Theory Manual for Multi-physics Code Coupling in LIME," Version 1.0, Sandia Report, February, 2011

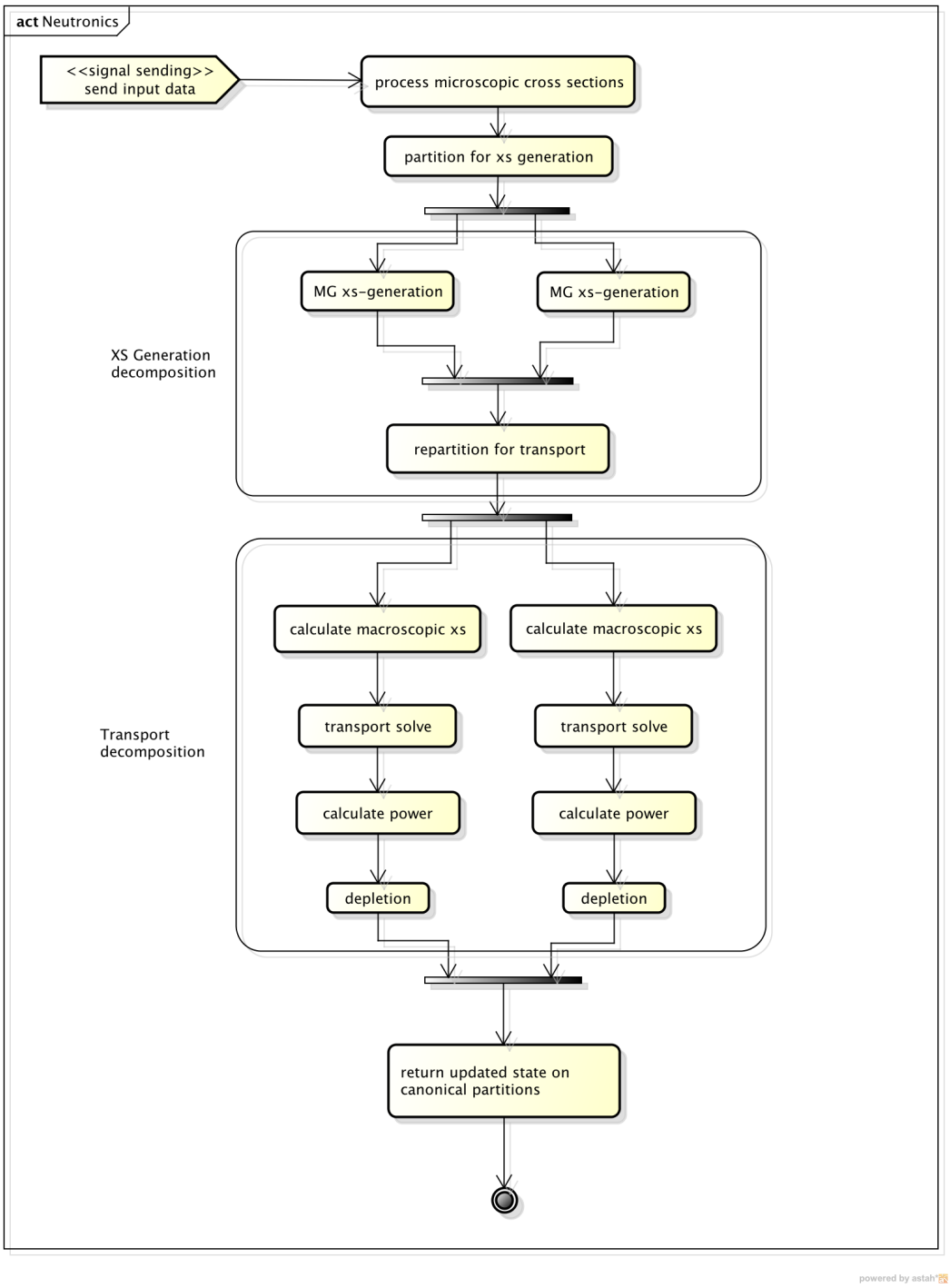


Figure 4: Activity diagram for the DM shown in Fig. 3. The parallel regions show only 2 paths of concurrency, but this is for illustrative purposes only. There can be many concurrent paths in a simulation.

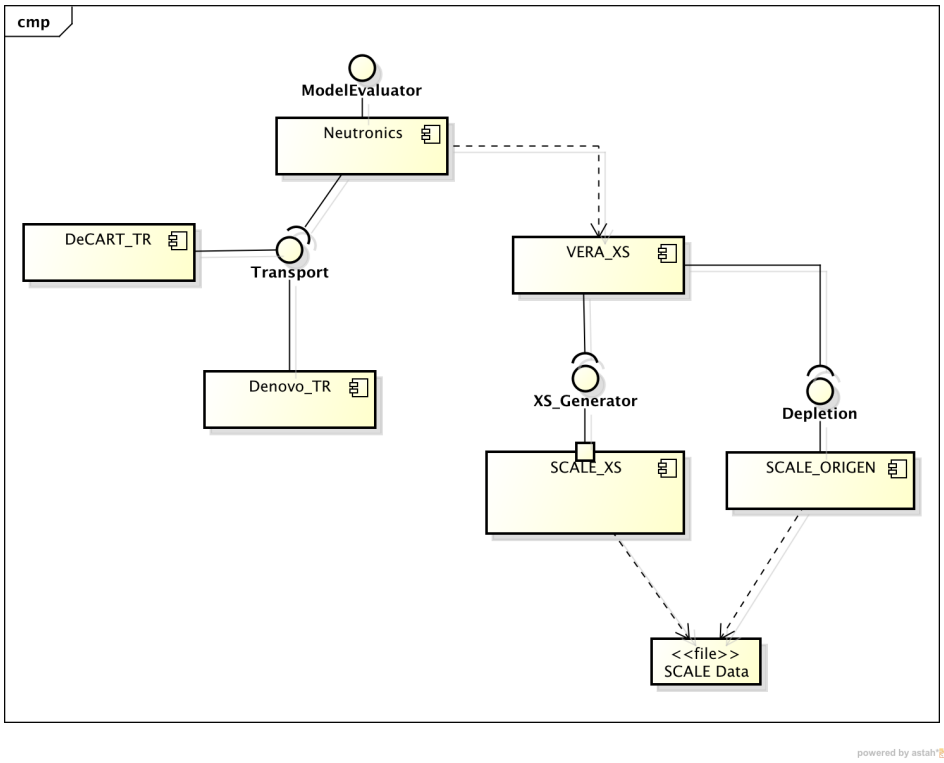


Figure 5: Component diagram of the VERA-Neutronics package.

incorporation into LIME. The models shown below are defined at the *Component*-diagram level. Individual class diagrams for the components will be developed in the component-level documentation that arises from the design process for each constituent component.

A component diagram for the VERA-Neutronics package is illustrated in Fig. 5. In this model, two transport components are shown, DeCART [5] and Denovo [1]. Each implements a transport-level interface that can be called from the Neutronics component. The Neutronics component is a model of the ModelEvaluator concept that is defined by LIME. Thus, in this design there is a single ModelEvaluator for the multigroup neutronics physics. The dynamic behavior of the VERA-Neutronics package is illustrated in Fig. 6.

The VERA_XS components will use the SCALE [7] code system for cross section processing and depletion. The ORIGEN [2] module will be used for depletion calculations. The latest version of ORIGEN (6.1+) has a defined API that allows it to be called inside of other applications. ORIGEN depletion calculations are performed in each problem zone (computational cell); therefore, there are no parallel issues that need to be resolved as no space-dependent coupling is involved in the depletion calculations. ORIGEN will require disk I/O as it needs to load cross sections and decay coefficients for each material.

There are three prospective APIs being developed in SCALE for cross section generation. These utilize the BONAMI-IR [3] and CENTRM [9] modules and an as-yet undeveloped Embedded Self-Shielding Method (ESSM) [8]. Front-ends will be developed for each module that correspond to the XS_Generator interface specification as illustrated in Fig. 7. The new APIs do not write the generated multigroup cross sections to disk; all calculated cross sections are kept in dynamic memory. Thus, no file I/O is required to communicate data between the components. The only required file I/O is done when reading the continuous energy microscopic cross sections from the master library. The BONAMI component will be the first cross section generator integrated into Neutronics.

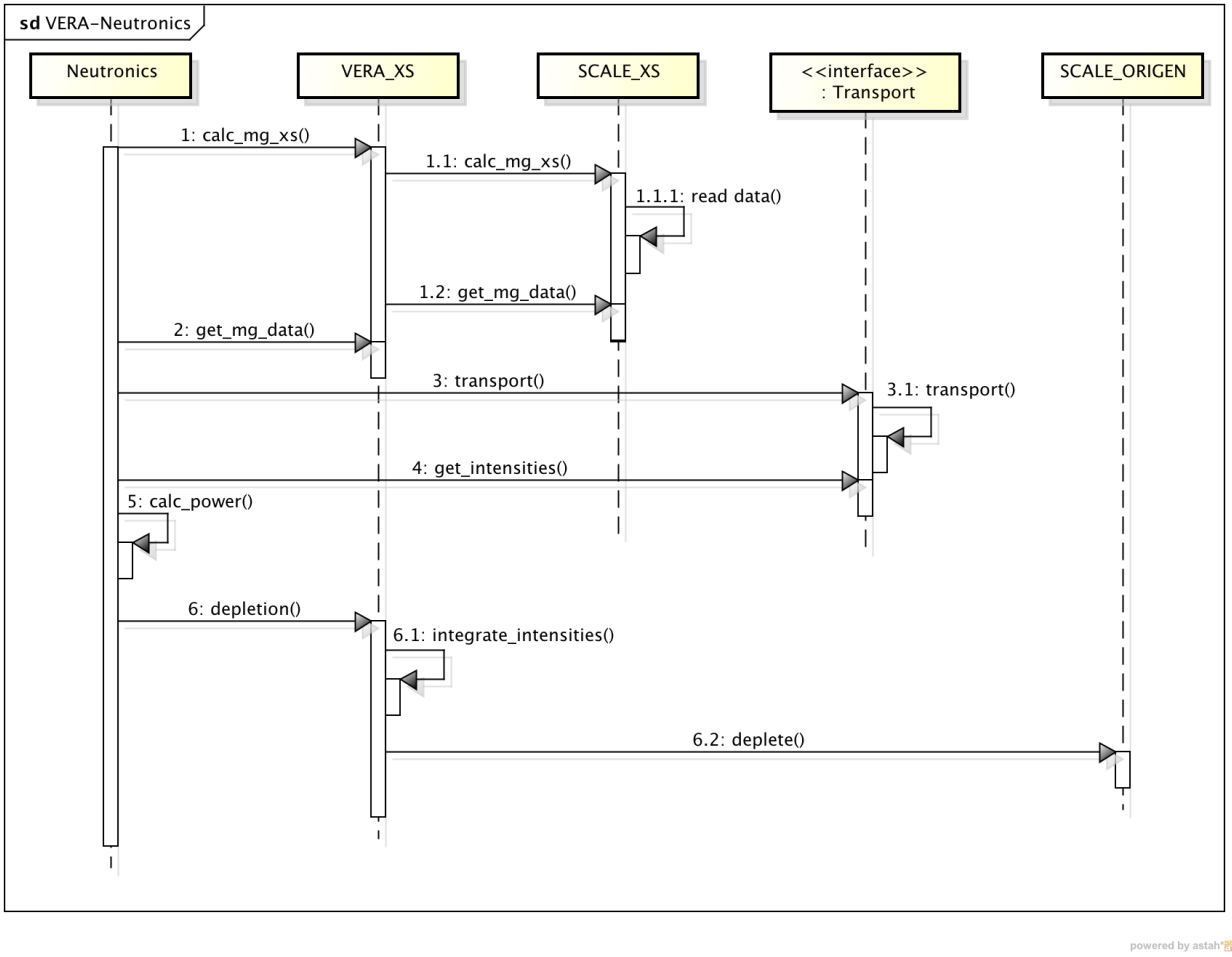


Figure 6: Sequence diagram of the VERA-Neutronics package.

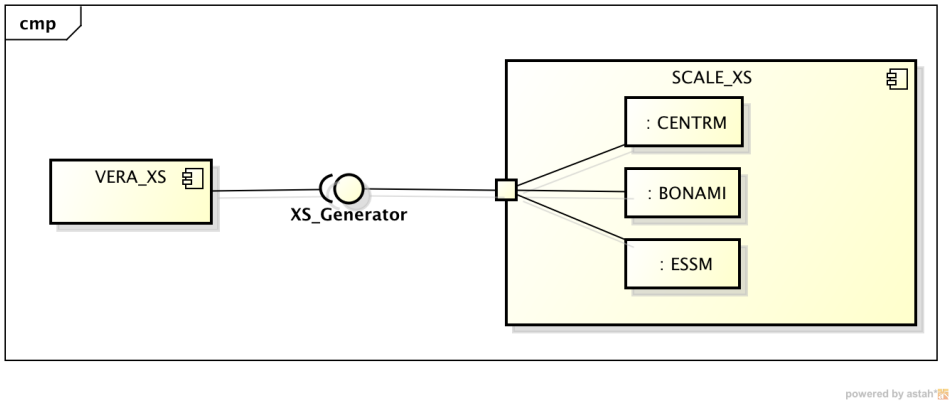


Figure 7: Detailed view of the SCALE_XS component.

References

[1] T. Evans, A. Stafford, R. Slaybaugh, and K. Clarno, "Denovo: A new three-dimensional parallel discrete ordinates code in SCALE," *Nuclear Technology*, vol. 171, 2010.

[2] I. Gauld, O. Hermann, and R. Westfall, "ORIGEN-S: SCALE system module to calculate fuel depletion, actinide transmutation, fission product buildup and decay, and associated radiation source terms," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, January 2009.

[3] N. Greene, "BONAMI: Resonance self-shielding by the Bondarenko method," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, January 2009.

[4] N. Greene, L. Petrie, and R. Westfall, "NITAWL: SCALE system module for performing resonance shielding and working library production," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, January 2009.

[5] M. Hursin, B. Kochunas, and T. Downar, "DeCART v2.05 theory manual," tech. rep., University of Michigan, November, 2008.

[6] J. Lamarsh and A. Baratta, *Introduction to Nuclear Engineering*. Upper Saddle River, NJ: Prentice-Hall, 3rd ed., 2001.

[7] Oak Ridge National Laboratory, *SCALE: A Modular Codes System for Performing Standardized Computer Analyses for Licensing Evaluation*, January 2009. Version 6, ORNL/TM-2005/39.

[8] M. Williams, "Self-shielding methods for SCALE-7 and CASL." White paper on cross-section processing methods for CASL., February 2011.

[9] M. Williams, M. Asgari, and D. Hollenbach, "CENTRM: A one-dimensional neutron transport code for computing pointwise energy spectra," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory, January 2009.

Distribution

Chris Baker <bakercg@ornl.gov>
Roscoe Bartlett <rabartl@sandia.gov>
Samuel Bays <Samuel.Bays@inl.gov>
Jay Billings <billingsjj@ornl.gov>
Kevin Clarno <clarnokt@ornl.gov>
Bob Grove <grovere@ornl.gov>
Brendan Kochunas <bkochuna@umich.edu>
Bill Martin <wrm@umich.edu>
Roger Pawlowski <rppawlo@sandia.gov>
Brad Rearden <reardenb@ornl.gov>
Rod Schmidt <rcschmi@sandia.gov>
John Turner <turnerja@ornl.gov>
John Wagner <wagnerjc@ornl.gov>

CAUTION

This document has not been given final patent clearance and is for internal use only. If this document is to be given public release, it must be cleared through the site Technical Information Office, which will see that the proper patent and technical information reviews are completed in accordance with the policies of Oak Ridge National Laboratory and UT-Battelle, LLC.