Specification of Advanced Safety Modeling Requirements
(Rev. 0)

Global Nuclear Energy Partnership

Prepared for
U.S. Department of Energy
Reactor Campaign
T. H. Fanning and T. J. Tautges
Argonne National Laboratory
April 30, 2008
ANL-AFCI-229

Prepared by:

Principal Author

4/30/08
Thomas H. Fanning Date

Reviewed by:

GNEP Work Package Manager

04/30/08
James E. Cahalan Date

GNEP WBS Level 4 Manager

4/30/08
Andrew R. Siegel Date

Approved by:

GNEP Reactor Campaign Director

4/30/08
Robert N. Hill Date

ABSTRACT

The U.S. Department of Energy's Global Nuclear Energy Partnership has lead to renewed interest in liquid-metal-cooled fast reactors for the purpose of closing the nuclear fuel cycle and making more efficient use of future repository capacity. However, the U.S. has not designed or constructed a fast reactor in nearly 30 years. Accurate, high-fidelity, whole-plant dynamics safety simulations will play a crucial role by providing confidence that component and system designs will satisfy established design limits and safety margins under a wide variety of operational, design basis, and beyond design basis transient conditions. Current modeling capabilities for fast reactor safety analyses have resulted from several hundred person-years of code development effort supported by experimental validation. The broad spectrum of mechanistic and phenomenological models that have been developed represent an enormous amount of institutional knowledge that needs to be maintained. Complicating this, the existing code architectures for safety modeling evolved from programming practices of the 1970s. This has lead to monolithic applications with interdependent data models which require significant knowledge of the complexities of the entire code in order for each component to be maintained.

In order to develop an advanced fast reactor safety modeling capability, the limitations of the existing code architecture must be overcome while preserving the capabilities that already exist. To accomplish this, a set of advanced safety modeling requirements is defined, based on modern programming practices, that focuses on modular development within a flexible coupling framework. An approach for integrating the existing capabilities of the SAS4A/SASSYS-1 fast reactor safety analysis code into the SHARP framework is provided in order to preserve existing capabilities while providing a smooth transition to advanced modeling capabilities. In doing this, the advanced fast reactor safety models will target leadership-class computing architectures for massively-parallel high-fidelity computations while providing continued support for rapid prototyping using modest fidelity computations on multiple-core desktop platforms.

CONTENTS

FIGURES

REACTOR CAMPAIGN SPECIFICATION OF ADVANCED SAFETY MODELING REQUIREMENTS (REV. 0)

1. Introduction

The U.S. Department of Energy's Global Nuclear Energy Partnership (GNEP) has lead to renewed interest in liquid-metal-cooled fast reactors for the purpose of closing the nuclear fuel cycle and making more efficient use of future repository capacity. Liquid-metal-cooled fast reactors in the form of sodium-cooled fast reactors have been successfully built and tested in the U.S. and throughout the world.[1] However, no fast reactor has operated in the U.S. for nearly fourteen years. More importantly, the U.S. has not designed or constructed a fast reactor in nearly 30 years. In addition to reestablishing the necessary industrial infrastructure, the development, testing, and licensing of a new, advanced fast reactor concept will likely require a significant base technology program that relies more heavily on modeling and simulation than has been done in the past. Accurate, high-fidelity, whole-plant dynamics safety simulations will play a crucial role by providing confidence that component and system designs will satisfy established design limits and safety margins under a wide variety of operational, design basis, and beyond design basis transient conditions over the life cycle of a plant. This paper defines the initial modeling requirements for an advanced safety modeling capability that fills this role along with a transition plan for preserving the enormous institutional knowledge represented in the current, state-of-the-art, liquid-metal fast reactor safety analysis codes.

In the sections that follow, a brief overview of a few of the past and existing safety analysis codes is presented along with a discussion of some of their limitations from the perspective of developing a significantly more advanced capability. Following this background, a selection of safety modeling requirements is presented that will form the basis of an advanced safety modeling capability. Because currently-available safety analysis codes will play an important role in the near term as higher-resolution advanced safety simulation methods are introduced, an implementation strategy is recommended that will provide a transition from existing capabilities to the high-fidelity, whole-plant simulation capabilities to be developed.

2. Background

In the late 1960s, the then U.S. Atomic Energy Commission gave development of a liquid-metal-cooled fast reactor (LMR) a high priority, and the development of the Fast Flux Test Facility (FFTF) became a cornerstone of that program. To provide adequate support for the FFTF and for the expected LMRs to follow, a major base technology program was established which provided a continuous stream of experimental information and design correlations. This experimental data would either confirm design choices or prove the need for design modifications. At the time, the "tremendous amount of data and experience pertaining to thermal design" of LMRs was recognized as providing the technical foundation for the future commercial development of LMRs.[2]

Along with the generation of experimental data came the development of safety analysis methods that used that data in correlations for mechanistic, probabilistic, or phenomenological models. These models were developed for a variety of needs ranging from individual components, such as heat exchangers, pumps, or containment barriers, to whole core or even whole-plant dynamics. Because a major portion of the overall technical effort expended within the nuclear power industry has always been allocated to safety considerations, the breadth of activities carried out since the late 1960s goes well beyond the scope of this paper. Instead, only a few of the more prominent whole-core or whole-plant dynamics safety modeling capabilities will be summarized.

Perhaps the strongest factor that influenced early fast reactor safety analysis was the concern over the possibility of core compaction followed by an energetic core disassembly — the so-called Bethe-Tait accident.[3] In the late 1960s, the Hanford Engineering Development Laboratory (HEDL) began developing the MELT code[4,5] to evaluate the initiating phase of hypothetical core disruption accidents (HCDA) as part of the FFTF project. The MELT series of codes has the capability to model the transient behavior of several representative fuel pins (channels) within a reactor core to allow for incoherency in the accident sequence. By 1978 MELT had evolved into the MELT-IIIB code.[5]

Around the same time that development on MELT began, Argonne National Laboratory began developing the SAS series of codes.[6–10] Like MELT, SAS has the capability to model the transient behavior of several representative channels to evaluate the initiating phase of HCDAs. SAS originated from a sodium boiling model and includes single- and two-phase coolant flow dynamics, fuel and cladding thermal expansion and deformation, molten fuel dynamics, and a point kinetics model with reactivity feedback. By 1974, SAS evolved to the SAS2A computer code[7] which included a detailed multiple slug and bubble coolant boiling model which greatly enhanced the ability to simulate the initiating phases of loss-of-flow (LOF) and transient overpower (TOP) accidents up to the point of cladding failure and fuel and cladding melting.

The SAS3A code[8] added mechanistic models of fuel and cladding melting and relocation. This version of the code was used extensively for analysis of accidents in the licensing of FFTF. In anticipation of LOF and TOP analysis requirements for licensing of the Clinch River Breeder Reactor Plant (CRBRP), new fuel element deformation, disruption, and material relocation models were written for the SAS4A version of the code,[9] which saw extensive validation against TREAT M-Series test data. In addition, a variant of SAS4A, named SASSYS-1, was developed with the capability to model ex-reactor coolant systems to permit the analysis of accident sequences involving or initiated by loss of heat removal or other coolant system events. This allows the simulation of whole-plant dynamics feedback for both shutdown and off-normal conditions, which have been validated against EBR-II Shutdown Heat Removal Test (SHRT) data and data from the FFTF LOF tests. Version 2.1 of the SAS4A/SASSYS-1 code was distributed to Germany, France, and Japan in the late 1980s, and served as the starting point for international oxide fuel model developments.

For HCDAs, SAS and MELT are limited to modeling the initiating phase of the accident sequences up to and including non-energetic failures. Argonne also developed the VENUS code to evaluate the energy released during an energetic disassembly phase.[11,12] In the disassembly phase the core materials are treated as a homogeneous, isotropic fluid, which may not be valid during milder excursions where structural considerations become important. To partially address this, the VENUS-II code[12] had the option to use pressure thresholds to constrain motion and simulate structural influence. Nevertheless, the importance of certain effects (e.g. retained fission gas expansion and coolant vaporization) results in significant conservatisms in evaluating accident energetics.

During accident sequences for which there are insufficient intrinsic negative feedbacks to terminate the hypothetical excursion, it is possible that large segments of the core could melt. Because of the conservatisms required in evaluating the disassembly phase, Los Alamos National Laboratory developed the somewhat higher-fidelity SIMMER code[13,14] that could evaluate the slowly-developing transition phase of disrupted core geometry leading up to core disassembly.

Beyond these early developments, revisions to SAS4A/SASSYS-1 continued into the Integral Fast Reactor (IFR) program at Argonne, causing a shift in modeling emphasis towards metallic fuel and accident prevention by means of inherent safety mechanisms. The whole-plant dynamics capability of the

SASSYS-1 component plays a vital role in predicting passive safety feedback. With the termination of the IFR program in 1994, SAS4A/SASSYS-1 saw continued developments with additions for heavy liquid metal coolants (lead and lead-bismuth eutectic), steam generator modeling updates and support for spatial kinetics. The most recent updates include capabilities for whole-core subchannel analysis.[10]

In 2002, under the plutonium disposition program, a version of SAS4A/SASSYS-1 was exported to Russia and training was provided in its use. Under the GNEP program in the U.S., SAS4A/SASSYS-1 continues to be a means for evaluating the safety performance of advanced nuclear reactor design features and serves as a focal element in international collaboration in fast reactor safety analysis. As an example of this latter role, France is currently restoring its version of SAS4A/SASSYS-1 to re-establish a fast reactor safety studies capability.

3. Current Modeling Capabilities and Limitations

The SAS4A/SASSYS-1 code continues to be maintained under active development and represents the current state-of-the-art in fast reactor safety analysis codes. SAS4A/SASSYS-1 contains extensive modeling capabilities that represent several hundred person-years of code development effort supported by experimental validation. These capabilities include

- Multiple channel and subchannel modeling of core thermal hydraulics limited only by available computing memory.
- Point kinetics and spatial kinetics capabilities including decay heat and reactivity feedback models for fuel Doppler; fuel, cladding, and coolant density variations; coolant voiding; core radial expansion; control-rod driveline expansion; and primary vessel expansion.
- Detailed mechanistic models for oxide fuel and cladding that characterize porosity migration, grain growth, fission gas release, fuel cracking with crack healing, fission-gas-induced swelling, irradiation-induced steel swelling, gas plenum pressurization, fuel-cladding gap conductance changes, fuel and cladding mechanical behavior, thermal expansion, and cladding failure.
- Detailed models of metallic fuel cladding transient behavior, metal fuel pre-transient thermophysical properties characterization, and pre-failure transient behavior models for fuel element mechanics, central cavity formation, extrusion, fission-gas-induced swelling, plastic flow, fuel-cladding eutectic formation, and fuel element failure detection.
- Two-phase coolant thermal hydraulics model to characterize low-pressure sodium boiling with the ability to track the formation and collapse of multiple bubbles and the ejection of liquid slugs from coolant channels.
- Intra-pin oxide fuel melting and relocation; cladding failure; molten cladding dynamics including melting, relocation, and freezing; fuel-coolant interactions in flooded channels including fission gas release, cladding perforation, molten fuel flow, and fuel freezing and plating; and fuel, fission gas, cladding, and coolant vapor dynamics in voided coolant channels.
- Primary and intermediate loop reactor coolant systems models for compressible volumes (with or without cover gas), pipes, intermediate heat exchangers, centrifugal pumps, electromagnetic pumps, valves, bypass channels, annular flow elements, reactor vessel auxiliary cooling systems (RVACS), air-dump heat exchangers, and steam generators.
- Balance of plant thermal hydraulics modeling capabilities including component models for deaerators, steam drums, condensers, reheaters, turbines, and several other components.
- Reactor control system models that are driven by user-defined mathematical operators controlled by simulation variables.

Because most of these modeling capabilities originated on very early computing architectures, the current code demonstrates fast execution times on common desktop computing resources available today. However, this origin has also resulted in significant limitations on continued code development. In the current code, fast execution times are due in part to the use of coarse-mesh, one-dimensional or even quasi one-dimensional models for components and systems. These lower-fidelity models perform very well in capturing overall plant dynamic responses to various transient conditions, but they do not capture the detailed flow and temperature distributions that evolve within and between components in response to transient conditions. It is therefore a priority to implement higher-fidelity models that can be coupled into a whole-plant dynamics simulation capability.

SAS4A/SASSYS-1, like many codes that originated predominantly in the 1970s, represents a monolithic application with an archaic code architecture. This resulted from then-modern programming practices which targeted scalar computing resources with limited memory and CPU speed. Each component in the SAS4A/SASSYS-1 code relies on an interdependent data model which requires significant knowledge of the complexities of the entire code in order for each component to be maintained. In addition, the lack of modularity impedes the ability to develop higher-fidelity, multi-dimensional component models that can take advantage of leadership-class computing resources. This is made more difficult by the fact that SAS4A/SASSYS-1 has been under continuous development for approximately 40 years, with significant attrition of code development knowledge during its more recent history.

4. Advanced Safety Modeling Requirements

In order to develop an advanced safety modeling capability, the limitations of the existing code architecture must be overcome. Nevertheless, a fundamental requirement is to preserve the existing capabilities while translating the knowledge of fast reactor physics, fluid dynamics, structural mechanics, and fuel performance phenomena that is embedded in the current tools into a more accessible code structure. A second fundamental requirement is for the safety modeling capability to be developed under an appropriate quality assurance plan as defined by the GNEP program.

Additional requirements for an advanced safety modeling capability include

- Modularity
  - Develop component models using modern, modular programming practices to support updating, modifying, or replacing component models with little to no modifications to the underlying simulation framework.
  - Define functional interfaces for each component model, specifying required input and output generated by the component. Where appropriate, define or use data abstractions and models for data that is common to several components. Well-defined interfaces and data models facilitate connecting components together for multi-physics, system, or mixed-fidelity models.
  - Support regression testing to verify or validate individual component models through the use of standalone drivers.
- Fidelity
  - Target both leadership-class computing (and foreseeable architectures) for massively-parallel high-fidelity computations along with continued support for modest-fidelity computations on multiple-core desktop computing platforms.
  - Support simultaneous coupling of high-fidelity component models running on many processors with low-fidelity or lumped parameter component models running on a single processor.

- Framework
  - Support definition of part or all of physical domain using modern CAD or mesh functionality, and reading of the physical domain definition from the simulation framework.
  - Communicate with other components through the simulation framework, using geometry or mesh for the physical domain definition, and variables on the geometry or mesh for physics quantities.
  - Support coupling at a variety of granularities, from coarse-grained coupling (communication only at start or end of run) to fine-grained coupling (communication every time step).
  - Support deformation of physical domain as a result of structural mechanical modeling to provide feedback to other components.
- Overall Integration
  - Support a wide range of transient conditions, from short term "prompt" transients to long-term operational transients or transients involving decay-heat removal.
  - Support an efficient steady-state initialization before the onset of the transient calculation to minimize the expense of null-transient calculations.
  - Support the coordination of time-step sizes at the coupling interfaces, but component models may select smaller steps for internal calculations.

The above requirements specify a modular based code structure supported by a flexible simulation framework. By eliminating unnecessary dependencies like those that exist in current codes, safety modeling capabilities can expand far beyond their current limitations to support higher-fidelity simulations that are applicable to a wider variety of applications.

5. Implementation Strategy

As previously stated, the effort to develop an advanced safety modeling capability must be carried out in a way that preserves the existing capabilities while satisfying the requirements defined in the previous section. The modular approach based on a flexible framework will allow new capabilities to take advantage of modern and future computing architectures while being easier to maintain. This approach is consistent with the overall SHARP reactor simulation project at Argonne.

In the following subsections, the SHARP framework design will be described, and the plans for incorporating safety modeling capabilities into the SHARP framework will be given. For a more detailed description of the framework and its support of reactor simulation, see Reference 15.

5.1 SHARP Framework Design

Physical processes in a nuclear reactor are inherently coupled, depending on reactor physics, thermal-hydraulics, and structural mechanics. Simulations can vary from detailed, first-principles studies of channel flow, done on the largest supercomputers available, to parametric design studies performed on workstations in minutes. Results from the detailed calculations are often used to construct models for the less detailed codes. Ideally, a modeling framework must support the various simulation types as well as the exchange of data and models between them.

There are several process-based requirements which guide the design of the SHARP framework and which are relevant to advanced safety modeling (and which are represented in the requirements in the previous section):

- *Minimally intrusive*: A variety of codes and code modules have already been developed for use in reactor simulations, each with tens or even several hundreds of person-years invested in their development, qualification, and validation. It is infeasible to expect that these applications be entirely re-written to fit into a new code coupling framework. Therefore, the effort required to attach a new code or physics module to the framework must be minimally intrusive to the original code.

- *Compatibility with standalone development*: There are many simulation methods which, while not developed originally for that purpose, are well suited to reactor simulation. Furthermore, these codes will continue to be developed outside the framework used for reactor simulation. The coupling framework must be compatible with standalone development of physics modules, so that updates to these modules can be made available as they are developed outside the framework. This capability also supports validation and regression testing on individual components independent of the coupled code.
- *Utility services*: There are many services which are needed by most parallel simulation codes, including parallel I/O, mesh generation, and visualization. Advanced techniques like adaptive mesh refinement and dynamic load balancing are also of interest to some codes. The coupling framework must provide utility services like these to avoid duplicate implementations in individual physics modules. The framework must also include a mechanism to add other services as they become available.
- *Integrated multi-physics*: Finally, there is a need to integrate physics modules of various types to perform coupled simulation for some problems. The coupling framework should simplify the addition of physics modules and their coupling to other modules in the framework.

The SHARP framework has been designed to meet these requirements while also providing important new simulation capabilities early in the design cycle. Thermal-hydraulic and neutronics modules are currently being interfaced to this framework. Section 5.2 describes the initial plans for how safety modeling capabilities will be integrated into this framework.

5.1.1 The ITAPS Mesh and Geometry Interfaces and Supporting Tools

The SHARP framework is being designed around function interfaces developed as part of ITAPS, a project supported by the DOE Scientific Discovery for Advanced Computing (SciDAC) program. The primary objective of the ITAPS center is to simplify the use of multiple mesh and discretization strategies within a single simulation on petascale computers. This is accomplished through the development of common functional interfaces to geometry, mesh, and other simulation data. These interfaces are referred to as the iMesh, iGeom, and iField interfaces, respectively.

The iMesh interface is implemented in the MOAB library. MOAB provides memory- and CPU-time-efficient storage and access to mesh data using array-based storage. MOAB also provides element types encountered in most finite element codes, as well as polygons and polyhedra. Tools included with MOAB provide parallel partitioning, parallel I/O, and commonly used functions like finding the outer skin of a contiguous mesh. Since the bulk of data in a PDE-based simulation is either the mesh or data associated with the mesh, the memory and cpu time performance of this component is critical to the overall performance of the coupled simulation.

The data model used by the mesh interface defines the data types used to pass information through the interface and how those types are used to express various mesh-based simulation data. The ITAPS and MOAB data models are the same and provide four fundamental data types:

- **Entity:** A typical finite element topological entity such as vertex, edge, triangle, or hexahedron.
- **Entity Set:** An arbitrary collection of entities and other entity sets. Entity sets also support parent/child relationships between sets, which are distinct from a set containing another set.

- **Interface:** An instance of the data model object through which all interface functions are called and within which entity handles are unique (i.e. if two entities have the same handle they are the same entity).
- **Tag:** Application-defined data which can be assigned to any of the other three types. Tags have specified size and name, and can have either a specified data type (within integer, double or entity handle) or an unspecified ("opaque") type.

Although relatively simple, this data model is capable of representing most types of data encountered in typical PDE-based simulation codes. For example, material types, processor partitions, and geometric topology used to generate a mesh can all be represented using a combination of sets and tags. This data model is also being used by the SHARP framework to define boundary conditions and subsets of a mesh to be coupled to subsets in another mesh for multi-physics simulations.

Using a relatively simple but common data model allows the construction of related tools based on generic geometric constructs without the need for unique data processing or modifications to dependent codes. Several tools are being or have been developed based on this data model that will be crucial to high-performance reactor simulation. These tools include:

- Zoltan: a static partitioning and dynamic load balancing tool from Sandia National Laboratories. Zoltan has been modified to interact with a mesh read through iMesh and embed partition data as sets and/or tags in iMesh. Other partitioners like Metis and Jostle can also be called through Zoltan.
- VisIt: a visualization tool developed at Lawrence Livermore National Laboratory. VisIt has been modified to read mesh data through iMesh, and is being enhanced to allow visualization of and interaction with set and tag data.
- CUBIT/MOAB: In addition to its role as the data conduit for SHARP simulation, MOAB is also able to read mesh data written by the CUBIT mesh generation toolkit developed at Sandia National Laboratories. If mesh is saved in CUBIT's ".cub" file format, most data (including geometric topology) can be restored when read into MOAB.
- Data Interpolation: Interpolation of data from one mesh to another is being implemented on top of MOAB. Various interpolation algorithms will be implemented, starting with a first-order point-in-element-based algorithm.

The iGeom interface is implemented in the Common Geometry Module (CGM) component. This component allows creation and evaluation of solid model-based geometry. The CUBIT mesh generation toolkit uses CGM, and meshes read into MOAB can be re-associated to CGM geometry. This capability is useful for adaptive mesh refinement on curved surfaces. Currently, CGM depends on the commercial ACIS solid modeling engine. However, development of an engine based on the open source Open Cascade modeling engine is underway, and should be ready before the end of FY08.

5.1.2 SHARP Framework Applications

As stated earlier, the SHARP framework is designed to simplify the construction of multi-physics applications. A variety of application types are supported, as shown in Figure 1. In all cases, code developers must separate physical models into a driver and a code library. Most of the physics modeling capabilities are put into the library, and therefore are available to both stand-alone and coupled codes. This separation is easily accomplished for most modern applications. After this is accomplished, developers can connect the physics models (through the library) into the framework in one of several ways, depending on the degree of reliance on the framework that is desired. This is illustrated by the following.

Physics Model A: Stand-Alone Model with Mesh Adapter. Existing massively parallel physics codes already implement mesh handling and other capabilities needed for high parallel performance. In these codes it may be difficult to replace the mesh implementation with that of the framework, since performance is very sensitive to the mesh implementation. In cases like this, writing a mesh adapter is most appropriate. The adapter works in both directions, reading the mesh from the framework into the native representation of the physics model and writing physics quantities to be coupled to other modules from the physics module back into the framework. Adapters like this can be included or not based on compile-time directives, and therefore are only used when the module is built in coupled mode. While there is some duplication of data and some extra work to copy the data back and forth, this work is not substantial, and much of it occurs during startup and is amortized over the entire execution. This approach is being used by SHARP with the Nek Thermal/Hydraulics module. At the other end of the spectrum, existing code tools such as SAS4A/SASSYS-1 may employ coding practices or solution methods that are incompatible with the framework. In this case, the mesh adapter may work across a surface, rather than a volume, and the physics model essentially becomes a black box on the other side of the surface.

Phyiscs Models B and C: Stand-Alone Models without Mesh Adapter. Developers may commit to using the framework directly for mesh and field data storage. This allows developers to also take advantage of the mesh services shown in Figure 1, including parallel I/O and mesh partitioning. If the physics module is written in C or Fortran, it will access the mesh through the iMesh interface. However, if the module is written in C++ and is particularly sensitive to memory performance, the data can be accessed directly in underlying MOAB implementation. Since the iMesh interface carries little state, the module will see the same exact mesh and data through both interfaces. This approach of using the iMesh interface is implemented by the UNIC neutronics code in SHARP.

Physics Models A-C: Multi-Physics Driver. Physics modules are coupled together through the mesh and field data stored in the framework. Since these modules are already connected with the framework in some way, no extra development work is needed to move data between the module and the framework. Data is communicated between physics modules by mapping from the mesh used by one module to the mesh used by the other, or directly if modules use the same mesh. Mapping is accomplished using a data coupler module, which is implemented as a mesh service on top of the framework. This approach centralizes the development of various mapping algorithms and allows mapping to be used by a variety of modules.

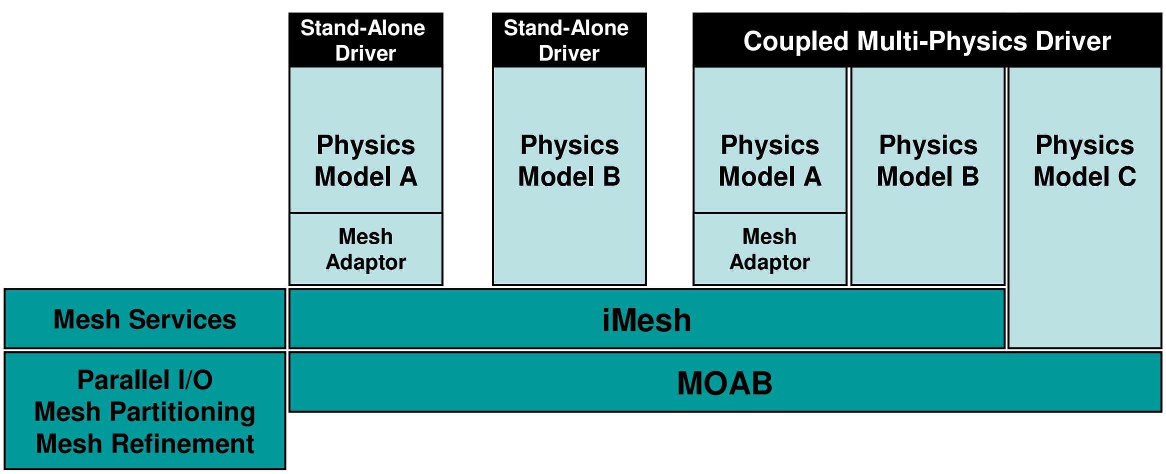

Figure 1: Relationships between Physics Modules and the SHARP Framework.

5.2 Safety Modeling in the SHARP Framework

To the extent possible, it is desirable to preserve and maintain the capabilities of existing codes such as SAS4A/SASSYS-1 as high-fidelity whole plant dynamics safety simulation methods are developed. This will ensure that a safety simulation capability is available for both short-term and long-term needs. In terms of the standalone codes depicted in Figure 1, SAS4A/SASSYS-1 is analogous to "Physics Model A". However, the various capabilities and models that are part of SAS4A/SASSYS-1 share a strongly interdependent data model that results in a monolithic code that can not be easily separated into modular components. This is depicted in Figure 2, where the main components of SAS4A/SASSYS-1 are collected into a single box.

One of the goals of advanced safety simulations is to provide simulation capabilities that do not currently exist for fast reactor analyses. It is not sufficient to simply rewrite existing capabilities using modern code practices. For example, with the forthcoming international benchmark to evaluate flow patterns and thermal stratification in the Monju upper plenum during shutdown conditions, a high-fidelity plenum modeling capability will be used. While various capabilities exist in the form of standalone computational fluid dynamics codes, none are currently coupled to a whole plant dynamics simulation capability. Another example is the need to develop, test, and validate a high-fidelity modeling capability to support design and safety evaluations for decay heat removal systems or other safety components. If that capability utilizes the SHARP framework, it now becomes a candidate for inclusion in the whole-plant dynamics simulation capability. These two standalone capabilities are also shown in Figure 2. Using these standalone capabilities as examples, an approach for incorporating them into a whole plant dynamics model while maintaining existing capabilities can be described.

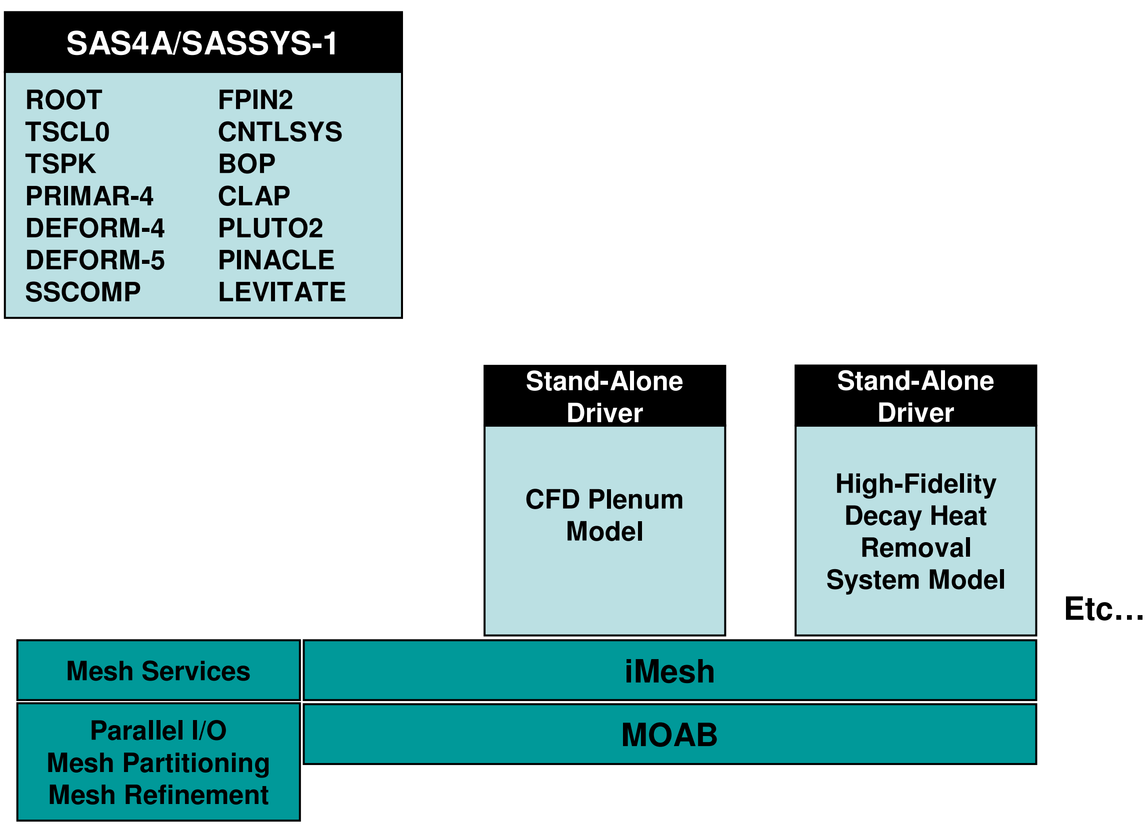

Figure 2: SAS4A/SASSYS-1 is a Monolithic Code Compared to Modular, High-Fidelity Modeling Capability Developed within the Framework.

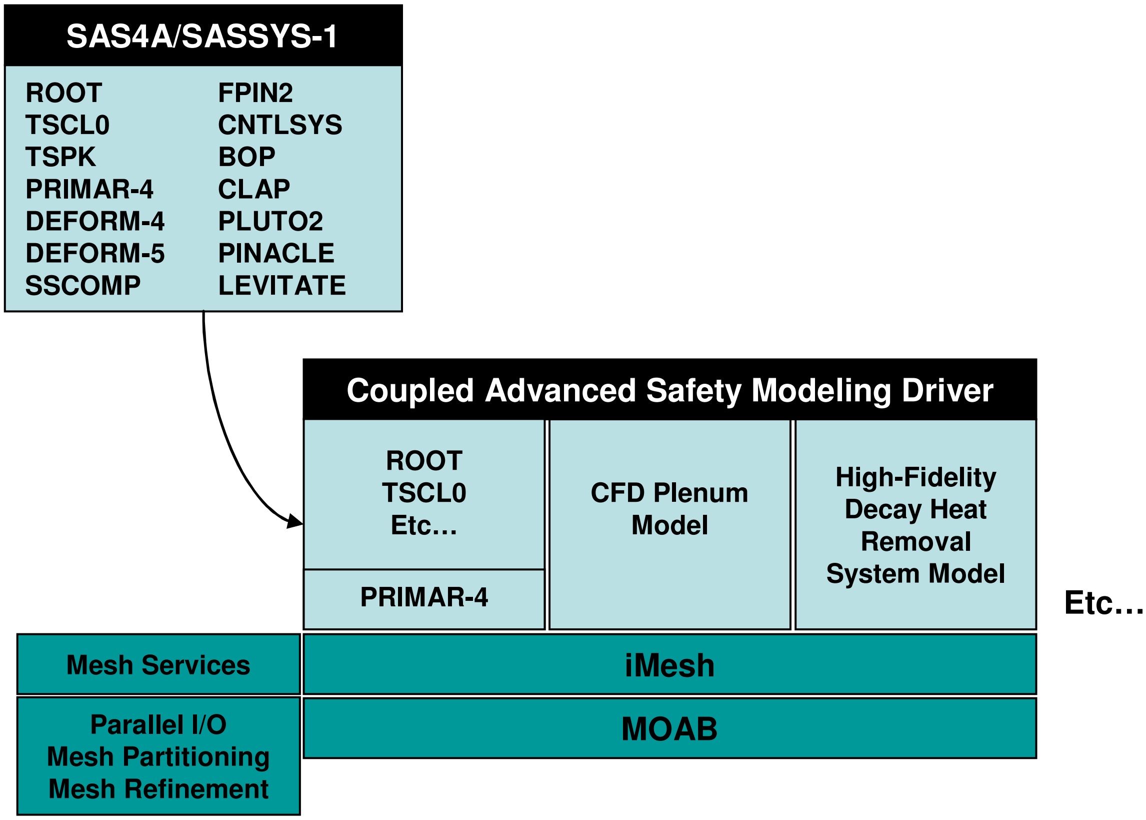

Figure 3: Relationship between SAS4A/SASSYS-1 and Advanced Modeling Capabilities when Coupled through iMesh.

Because of increased attention to passive safety and inherent feedback mechanisms in current advanced LMR designs, the first phase of development will focus on single-phase dynamic modeling of intact geometry and the important role that reactor coolant systems play in the overall passive response. Within SAS4A/SASSYS-1, the PRIMAR-4 primary and intermediate loop reactor coolant systems model has been identified as a potential point of interface into the SHARP framework. As a result, PRIMAR-4 becomes the "mesh adapter" identified in Figure 1 while the remainder of SAS4A/SASSYS-1 provides the initial whole-plant modeling capability. This relationship is shown in Figure 3.

PRIMAR-4 contains low-fidelity component models for compressible volumes (with or without cover gas) and elements such as pipes, intermediate heat exchangers, centrifugal pumps, electromagnetic pumps, valves, bypass channels, annular flow elements, reactor vessel auxiliary cooling systems (RVACS), air-dump heat exchangers, and steam generators. For coupling with the reactor core, PRIMAR-4 utilizes an extrapolated boundary condition, or estimate, of the thermal-hydraulic conditions at the core inlet and outlet. A similar type of approach can be used to couple PRIMAR-4 with the SHARP framework. In effect, the framework becomes a new element of PRIMAR-4. However, rather than a shared data model, the approach will require a functional interface between the framework and PRIMAR-4.

For subsequent phases of development, new modeling capabilities will be added to the framework in the form of modular components. The choice of components will depend on availability of models that have been developed for the SHARP framework and need. Over time, as new models are coupled into the safety simulation framework, the existing capabilities of SAS4A/SASSYS-1 will gradually become less important over time. However, the modeling capabilities of SAS4A/SASSYS-1 can continue to be maintained as a low-fidelity option. For example, the component models in PRIMAR-4 may compose a module for low-fidelity modeling of the associated reactor components, or they may be separated into individual physics modules for more flexibility. In this way, a smooth transition to an advanced, whole-plant dynamics simulation capability is provided while preserving and maintaining the capabilities that currently exist.

6. Summary

Current modeling capabilities for fast reactor safety analyses have resulted from several hundred person-years of code development effort supported by experimental validation. The broad spectrum of mechanistic and phenomenological models that have been developed represent an enormous amount of institutional knowledge that needs to be maintained. Complicating this, the existing code architectures for safety modeling evolved from programming practices of the 1970s. This has lead to monolithic applications with interdependent data models which require significant knowledge of the complexities of the entire code in order for each component to be maintained.

In order to develop an advanced fast reactor safety modeling capability, the limitations of the existing code architecture must be overcome while preserving the capabilities that already exist. A set of advanced safety modeling requirements has been defined, based on modern programming practices, that focuses on modular development within a flexible coupling framework. An approach for integrating the existing capabilities in the SAS4A/SASSYS-1 fast reactor safety analysis code into this framework is also provided in order to preserve existing capabilities while providing a smooth transition to advanced modeling capabilities. In doing this, the advanced fast reactor safety models will target both leadership-class computing architectures while providing continued support for modest fidelity computations on multiple-core desktop platforms.

References

1. *Fast Reactor Database, 2006 Update*, IAEA-TECDOC-1531, International Atomic Energy Agency, December 2006. (Available on-line via http://www-pub.iaea.org/MTCD/publications/PDF/te_1531_web.pdf.)
2. Y. S. Tang, R. D. Coffield, Jr., R. A. Markley, *Thermal Analysis of Liquid Metal Fast Breeder Reactors*, Westinghouse Electric Corporation, Madison, Pennsylvania, American Nuclear Society Publications, 1978
3. H. A. Bethe and J. H. Tait, *An Estimate of the Order of Magnitude of the Explosion When the Core of a Fast Reactor Collapses*, RHM-56-113, April 1956.
4. A. E. Waltar, A. Padilla, Jr., and R. J. Shields, *MELT-II, A Two-Dimensional Neutronics-Heat Transfer Program for Fast Reactor Safety Analysis*, HEDL-TME 72-43, Hanford Engineering Development Laboratory, 1972.
5. K. K. Tabb, C. H. Lewis, L. D. O'Dell, A. Padilla, Jr., D. E. Smith, and N. P. Wilburn, *MELT-IIIB: An Updated Version of the MELT Code*, HEDL-TME 78-108, Hanford Engineering Development Laboratory, April 1979.
6. J. C. Carter, et al., *SAS1A, A Computer Code for the Analysis of Fast-Reactor Power and Flow Transients*, ANL-7607, Argonne National Laboratory, October 1970.
7. F. E. Dunn, et al., *The SAS2A LMFBR Accident-Analysis Computer Code*, ANL-8138, Argonne National Laboratory, October 1974.

8. M. G. Stevenson, W. R. Bohl, F. E. Dunn, T. J. Heames, G. Hoppner, and L. L. Smith, "Current Status and Experimental Basis of the SAS LMFBR Accident Analysis Code System," *Proc. Fast Reactor Safety Meeting*, American Nuclear Society, CONF-740401-P3, pp. 1303–3121, Beverly Hills, CA, April 2–4, 1974.

9. J. E. Cahalan, A. M. Tentner, and E. E. Morris, "Advanced LMR Safety Analysis Capabilities in the SASSYS-1 and SAS4A Computer Codes," *Proceedings of the International Topical Meeting on Advanced Reactors Safety*, American Nuclear Society, Vol. 2, pp. 1038–1045, Pittsburgh, PA, April 17–21, 1994.

10. F. E. Dunn, J. E. Cahalan, D. Hahn, H. Y. Jeong, "Whole Core Sub-Channel Analysis Verification with EBR-II SHRT-17 Test," *Proc. 2006 Int'l Congress on Advances in Nuclear Power Plants*, Reno, NV, June 6–8, 2006.

11. W. T. Sha and T. H. Huges, *VENUS: A Two-Dimensional Coupled Neutronics-Hydrodynamics Computer Program for Fast-Reactor Power Excursions*, ANL-7701, October 1970.

12. J. F. Jackson and R. B. Nicholson, *VENUS-II: An LMFBR Disassembly Program*, ANL-7951, 1972.

13. L. L. Smith, J. E. Boudreau, C. R. Bell, P. B. Bleiweis, J. F. Barnes, and J. R. Travis, "SIMMER-I, An LMFBR Disrupted Core Analysis Code," *Proc. of the International Meeting on Fast Reactor Safety and Related Physics*, Chicago, IL, CONF-871001, October 1976.

14. L. L. Smith, et al., *SIMMER-II: A Computer Program for LMFBR Disrupted Core Analysis*, LA-7615-MS, NUREG/CR-9453, Los Alamos Scientific Laboratory, 1978.

15. A. Caceres, T. J. Tautges, J. Lottes, P. Fischer, C. Rabiti, M. A. Smith, A. Siegel, W. S. Yang, G. Palmiotti, *Status Report on SHARP Coupling Framework*, ANL-AFCI-215, Argonne National Laboratory, September 2007.