INL/EXT-18-44926

Light Water Reactor Sustainability Program

Risk-Informed Systems Analysis (RISA) Dynamic Fire PRA Roadmap

March 2018

DOE Office of Nuclear Energy

INL/EXT-18-44926

Light Water Reactor Sustainability Program

Risk-Informed Systems Analysis (RISA) Dynamic Fire PRA Roadmap

Steven Prescott
Ramprasad Sampath
John Biersdorf

March 2018

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov/lwrs

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

ABSTRACT

Modeling and implementing fire safety for nuclear power plants is a costly activity. Because of the complexity of fire phenomenon and multiple operational procedures, it is difficult to computationally provide assurance that the mitigation methods are adequate for critical areas using current analysis methods. An economical method to provide more accurate modeling and optimize mitigation methods is needed to improve nuclear power viability.

This report describes the initial investigation into modeling and simulation tools for application of fire as part of the Risk-Informed Systems Analysis (RISA), formerly Risk-Informed Safety Margin Characterization (RISMC) [1]. The report provides a framework of how 3D modeling and simulation techniques could be combined with a dynamic Probabilistic Risk Analysis (PRA) to reduce compounding conservatism present in current Fire PRA methods. Electrical Power Research Institute has analyzed current Fire PRA practices and identified the most significant contributors to risk and areas for improving analysis. This framework describes how to apply dynamic PRA and simulation methods for key contributors/scenarios, credit missing factors of manual fire suppression, and applying conditional probabilities.

CONTENTS

FIGURES

ACRONYMS

3D	Three Dimensional
API	Application Programming Interface
ASME	American Society of Mechanical Engineers
ANS	American Nuclear Society
CAD	Computer-Aided Design
CAFTA	Computer Aided Fault Tree Analysis System
CFAST	Consolidated Model of Fire and Smoke Transport
CDF	Core Damage Frequency
CFD	Computational Fluid Dynamics
CUDA	Computer Unified Device Architecture
DET	Dynamic Event Tree
DNS	Direct Numerical Simulation
DPRA	Dynamic Probabilistic Risk Analysis
EMRALD	Event Model Risk Assessment using Linked Diagrams
EPRI	Electrical Power Research Institute
FDS	Fire Dynamics Simulator
FDT	Fire Dynamics Tools
FFT	Fast Fourier Transform
FIVE	Fire-Induced Vulnerability Evaluation
FVM	Finite Volume Method
EdF	Electricite de France's
GVDB	GVariant DataBase File
HRR	heat release rates
LES	Large Eddy Simulation
MPI	Message Passing Interface
NIST	National Institute of Standards and Technology
NPP	Nuclear Power Plant
NRC	Nuclear Regulatory Commission
NUREG	Nuclear Regulatory Commission technical report designation
NUREG/CR	Contractor prepared NUREG
OBST	Obstruction definitions – for FDS inputs
PRA	Probabilistic Risk Analysis
PWR	Pressurized Water Reactor
RISA	Risk-Informed Systems Analysis
RISMC	Risk-Informed Safety Margin Characterization
SAPHIRE	Systems Analysis Programs for Hands-on Integrated Reliability Evaluations
SPH	Smoothed-Particle-Hydrodynamics
SSCs	structures, systems, and components
XMPP	Extensible Messaging and Presence Protocol

1. INTRODUCTION

The 1976 fire at Brown's Ferry Nuclear Plant highlighted weakness in assessing and understanding the impacts of fire scenarios with respect to potential core damage events for nuclear reactors. The U.S. Nuclear Regulatory Commission (NRC) and Electric Power Research Institute (EPRI) created Nuclear Regulatory Commission technical report designation (NUREG) NUREG/CR-6850 which outlined guidelines for the nuclear plants to monitor risk attributed to fire events within the plant to causing core damage or a release of radionuclides. This NUREG outlines the methodology for using PRA to monitor fire risk to the plant.

There is a very strong interest in analysis of fire protection engineering practice, both domestically and worldwide. The NRC with EPRI and the National Institute of Standards and Technology (NIST) conducted a research project to verify and validate five fire models used for Nuclear Power Plant (NPP) applications. The results of this effort are documented in a seven volume NUREG report, NUREG-1824 [2], Verification & Validation of Selected Fire Models for Nuclear Power Plant Applications. More recently, NUREG-1934 [3] document provides information on the use of fire models in support of various commercial NPP fire hazard analysis applications.

Extensive work has been done in modeling fire risk using outlined methods. EPRI in conjunction with NRC have conducted workshops based on Fire PRA. Conservative and quick to implement methods are used to reduce the scope of the problem to a few major areas of concern, such as control room abandonment and switch gear room fires. In these areas, more advanced modeling methods are needed in order to better evaluate the risk. This report discusses the more advanced methods and a plan of how to couple these methods or tools with dynamic PRA (DPRA), further reducing error and uncertainty at little to no additional effort. This would provide plants additional risk reduction benefits while simultaneously allowing identification of cost savings to reductions in the robustness of the plants fire protection plan.

Experimental methods to investigate fire in a realistic environment is situationally impossible and capturing complex scenarios such as control room abandonment with current PRA methods is an intensive and still vastly incomplete process. In such cases, probabilistic and physics based simulation analysis methods are a viable and practical tool.

2. EXISTING STATIC FIRE PRA

Static Fire PRA is designed basically with the same general concepts as normal internal events PRA. The major changes come from what type of components are currently in scope and how scenarios are created and handled. A static PRA utilizes single failures or initiating events when beginning to analyze the plants response and capabilities to respond to said event. All structures, systems, and components (SSCs) need to be incorporated into the PRA model to realistically assess the plant response to accident situations. However, it is not practical to model everything, so reasonable assumptions must be made to simplify the modeling and still get a realistic simulation.

In creating a Fire PRA, you need a risk analysis team of experts to help identify SSCs to be included and also to help develop the scenarios. The fire risk analysis team will include individuals with expertise in four key areas:

1) Fire analysis (basic fire behavior, fire modeling, fire protection engineering, and plant fire protection regulatory compliance practices and documentation)
2) General PRA and plant systems analysis (event tree/fault tree analysis, nuclear power plant systems modeling, reliability analysis, PRA practices as applied in the internal events domain, and specific knowledge of the plant under analysis)
3) Human reliability analysis (emergency preparedness, plant operations, plant-specific safe shutdown procedures, and operations staff training practices)

4) Electrical analysis (circuit failure modes and effects analysis and post-fire safe shutdown, including plant-specific regulatory compliance strategies and documentation)

While some of this expertise is generic, much of it is specific to the plant under analysis.

2.1 Methods

The methodology for the static Fire PRA is straight forward and is built of the same event tree fault tree approach as an internal events PRA. Approved methodology for Fire PRAs for use in nuclear power plants is found in the ASME/ANS PRA Standard [4]. The major differences lie in what is to be considered in scope and scenario development. First, the in-scope components need to be concerned with fire suppression and detection equipment, and more importantly the cabling needs to be traced and identified as it can fail equipment in rooms that the main component is not currently occupying. To create the overall fire scenarios, it is important to lay out the plant boundaries and identify areas where and how fires can be created in these rooms. Once boundaries are established, scenarios need to be made that incorporate the heat release rates of fire types, fire spread from target to target as well as room to room, suppression and detection equipment, and additional operator actions. The goal of the Fire PRA is to analyze these scenarios to identify the risks to the plant, what types of fires, and where those fires would occur in the plant causing the most significant amount of risk to core damage or a release of radionuclides.

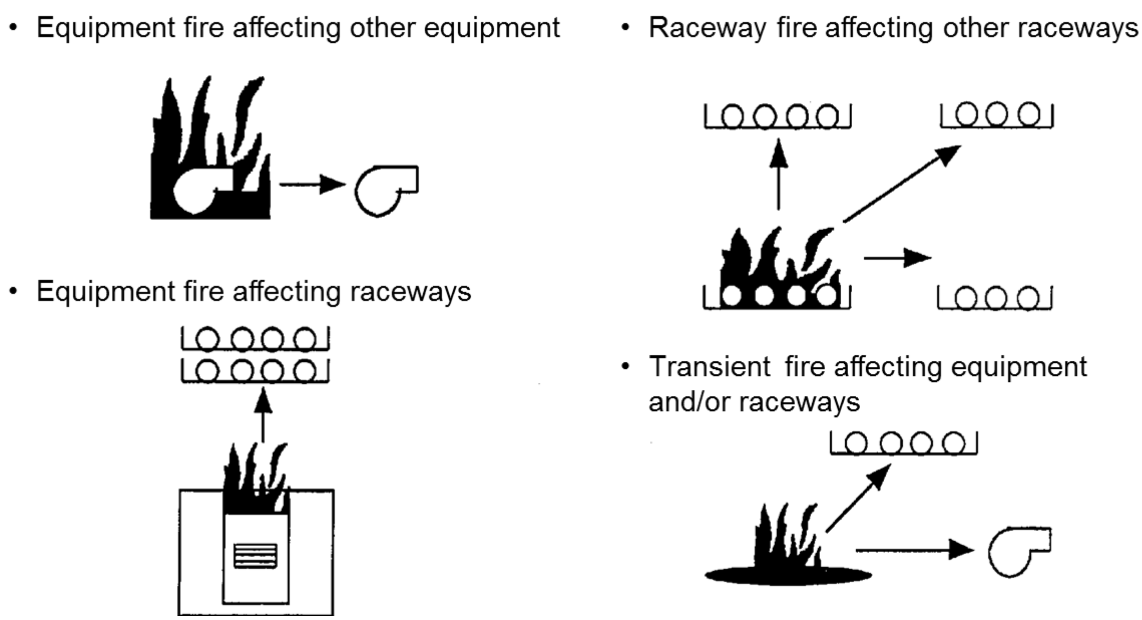

Figure 1: Sample scenarios for fire spread and impacts

The components that can cause fire (e.g., electrical cabinets, pumps, oil reservoirs, transient combustibles, cable fires) and then impact targets around them are shown in Figure 1. First, one must look in the direct impact on fires as the previous example shows. Next, you must determine if the fire can create a hot gas layer which can damage components indirectly. Finally, you need to analyze the scenarios about whether or not the fire (or hot gas layer) can spread into an adjacent compartment. These aforementioned steps provide the immediate scope of the fire and what it can directly impact the scenario. From here, it needs to be expanded to see what components are affected by cable damage, the timing for failures, and other impedances for operator actions, and incorporated into the scenarios.

When creating the scenarios in the PRA, there are multiple ways to incorporate the scenarios into the PRA model. You can add them directly into fault trees, or add them into event trees. A new PRA model does not need to be built for the Fire PRA; the scenarios can be incorporated directly into existing internal events PRAs. There are different types of logic modeling software such as RISKMAN, Computer Aided Fault Tree Analysis System (CAFTA), and Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE). SAPHIRE [5]the software that is used for examples and likely to be used for the DPRA research project but other tool may be used or models imported as deemed necessary.

To understand how current Fire PRA works within an internal events model, an example is shown in an event tree (Figure 2) and fault tree (Figure 3).

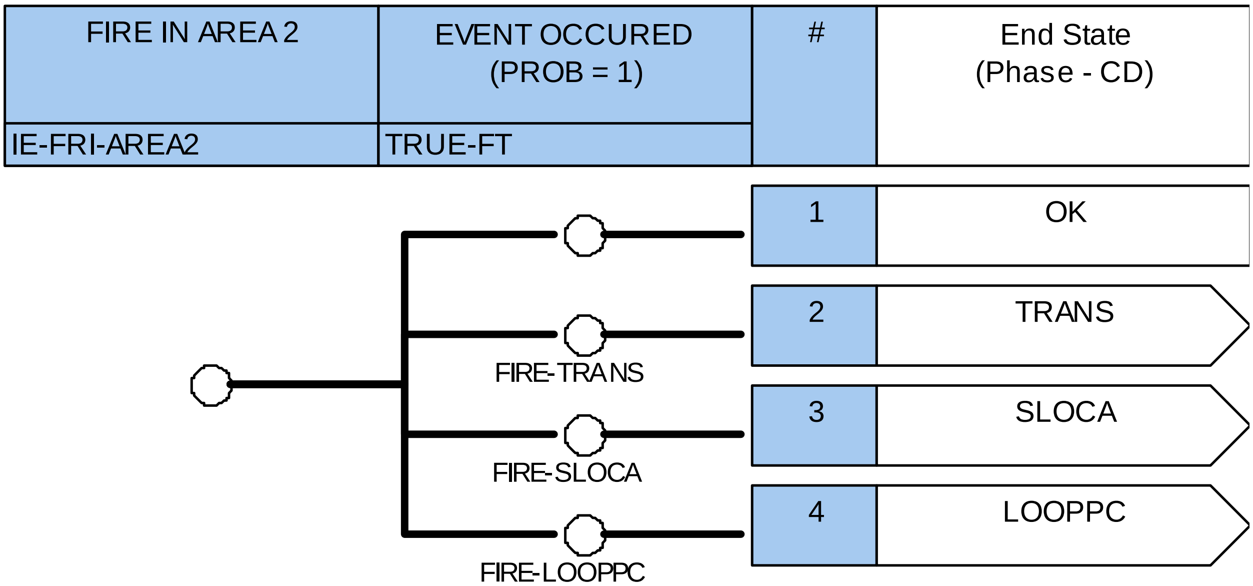

Figure 2: Example event tree with added fire branches into existing PRA

The initiators will be fires within the compartments and lead into their relevant internal-events event trees. This will check for systems to mitigate their respective initiators that can be caused by the fire. The internal events model will still analyze the individual random failures that can occur within the plant, but the fire failures must be accounted for as well. This is done by creating fire specific failures within the fault tree for that specific fire as shown in the HE-FIRE-AREA2 event in Figure 3.

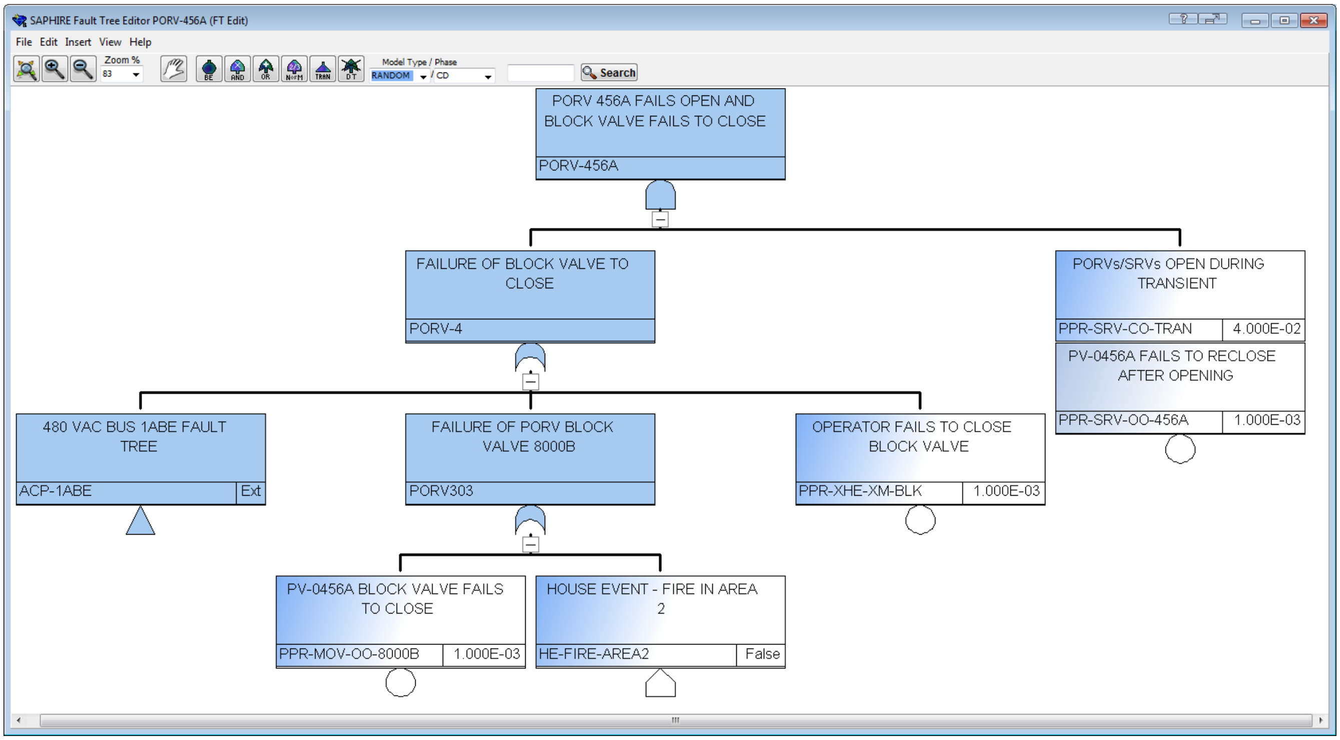

Figure 3: Example fault tree with specific fire failures added (HE-FIRE-AREA2)

This is the extent of the modeling that needs to be done for many scenarios and can be as specific as the analyst would like in terms of the extent of the impact of the fire, and what components will be impacted.

2.2 Detailed Methods

When modeling the fire scenarios the information is included at various points. For example, when deciding to model the main control room, the room is devided into different fire scenarios. The scenarios are typically devided by the type of initiator (electrical panels, motors, transients, hot work, etc.). Some of these initiators may need additional scenarios depending on the types of targets they can damage. In SAPHIRE, this would typically be done by modeling the scenarios directly as basic events in fault trees or by creating change sets that fail the components within the existing PRA model. The severity level (Figure 5) and non-suppression probabilities of the modeled fire level are directly incorporated into the ignition frequency of the initiator. This is the probability that a fire of a damaging consequence is started and maintained long enough to damage the targets that are identified in each scenario. The number of scenarios, targets, and ignition sources are going to be unique to each plant. The identification, classification, and modeling methods that are approved are found in the PRA standards document [4].

2.3 Fire Simulation

Fire modeling in a computational environment aims to predict fire behavior in different environmental conditions. These computational models need to take into account the various physical processes of combustion, fluid dynamics, and heat transfer processes. The complexity in modeling the physical process of fire propagation arises from the dynamic turbulent nature of the internal processes. Computational Fluid Dynamics (CFD) is a branch of fluid mechanics that provides methods and tools that help in modeling, prototyping, testing and analysis. NUREG-1824 documents the five software toolkits used in modeling fires in a typical NPP. They are (1) NRC's NUREG- 1805 Fire Dynamics Tools (FDT) [6], (2) EPRI's Fire-Induced Vulnerability Evaluation Revision 1, (FlVE-Rev1), (3) National Institute of Standards and Technology's (NIST) Consolidated Model of Fire Growth and Smoke Transport (CFAST) [7], (4) Electricite de France's (EdF) MAGIC [8], and (5) NIST's Fire Dynamics Simulator (FDS) [9].

FDT and FIVE-Rev1 from NRC and EPRI are algebraic models which take minimal inputs and create quick analytical calculations. The advantages are that it is fast and easy to use however this might result in being too conservative and has limited applications. CFAST and MAGIC are zone models and perform more detailed analysis than FDT. They can compute hot gas layers as well as target heat fluxes, however complex room geometries and large horizontal paths are not well treated.

FDS is broadest in scope in terms of the types of analysis which can be performed. Moreover, its dynamic in nature and simulations can be performed in three dimensional (3D) space. Chapter 3 of NUREG-1934 specifically lists seven generic scenarios which are considered for a fire at a NPP and in many of those scenarios CFD models are either recommended or necessary to provide detailed analysis with obstacles in the path of the fire. Since FDS is a computational fluid dynamics model, it allows the analyst the ability to simulate fire conditions in complex geometries with complex vent conditions. However, it takes a significant effort to properly create the model for simulation and computation times could be quite long. This modeling is used to simulate the types of fires that can cause damage to components by showing the specific heat release rates (HRR) that would damage nearby targets via direct impact or through a hot gas layer. A hot gas layer is a layer of smoke and heat that forms at the ceiling, filling the entire ceiling and slowly descends damaging additional targets as it lowers (see Figure 4).

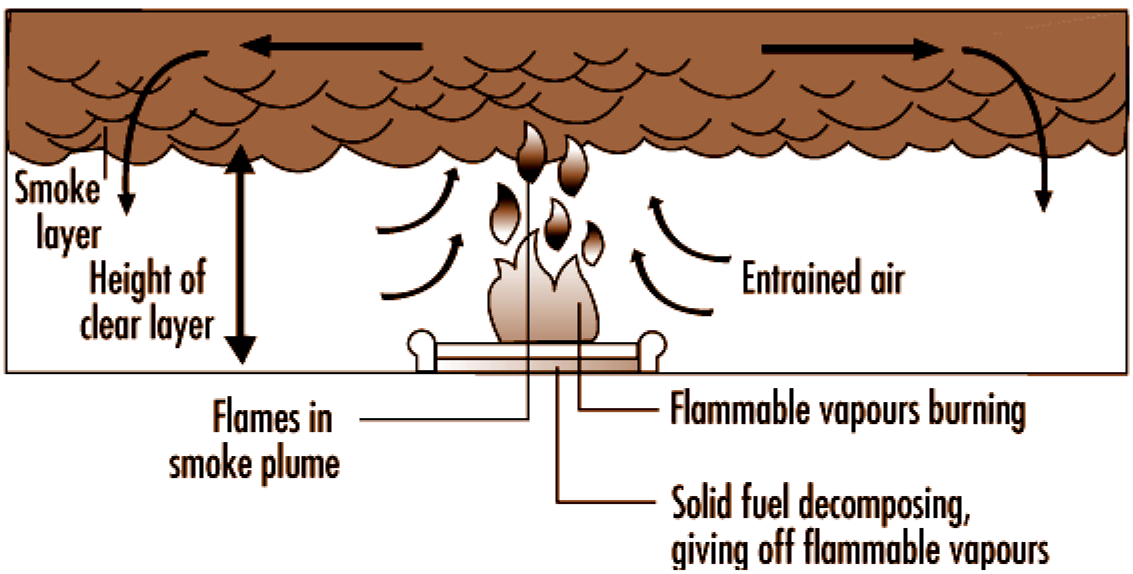

Figure 4: Smoke and hot gas layer formation and behavior.

Each scenario simulation provides an estimate of the scope of damage and what components will or will not be available to help mitigate the specific fire scenarios. The significant HRRs are incorporated though a severity factor within the ignition frequency of the fire scenarios.

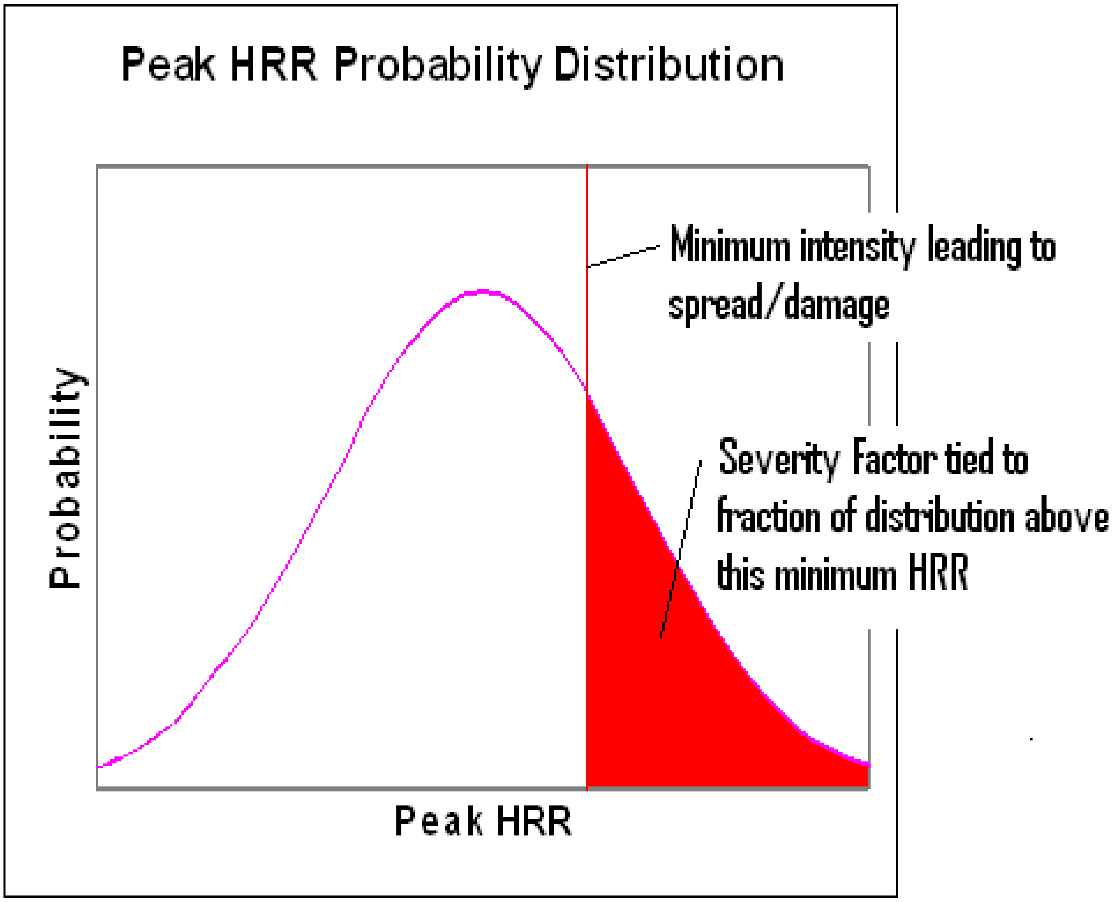

Figure 5: Example heat release rate with the severity factor for a particular component.

Figure 5 shows the percentage of fires that will be incorporated into the PRA as they can damage a specified set of additional targets and create a fire scenario that needs to be investigated. The other aspect of computer modeling is the timing of the specific fire and how long it takes to damage targets or create a hot gas layer. It additionally provides an estimate of how long it takes for suppression and detection systems to actuate, which can be incorporated with plant procedures and training times. Further, the info can be used as time measurements for fighting the fire and preventing it from damaging targets. This will help eliminate the impacts of fires within the PRA models as a non-suppression probability is incorporated into the ignition frequencies in addition to the severity factor.

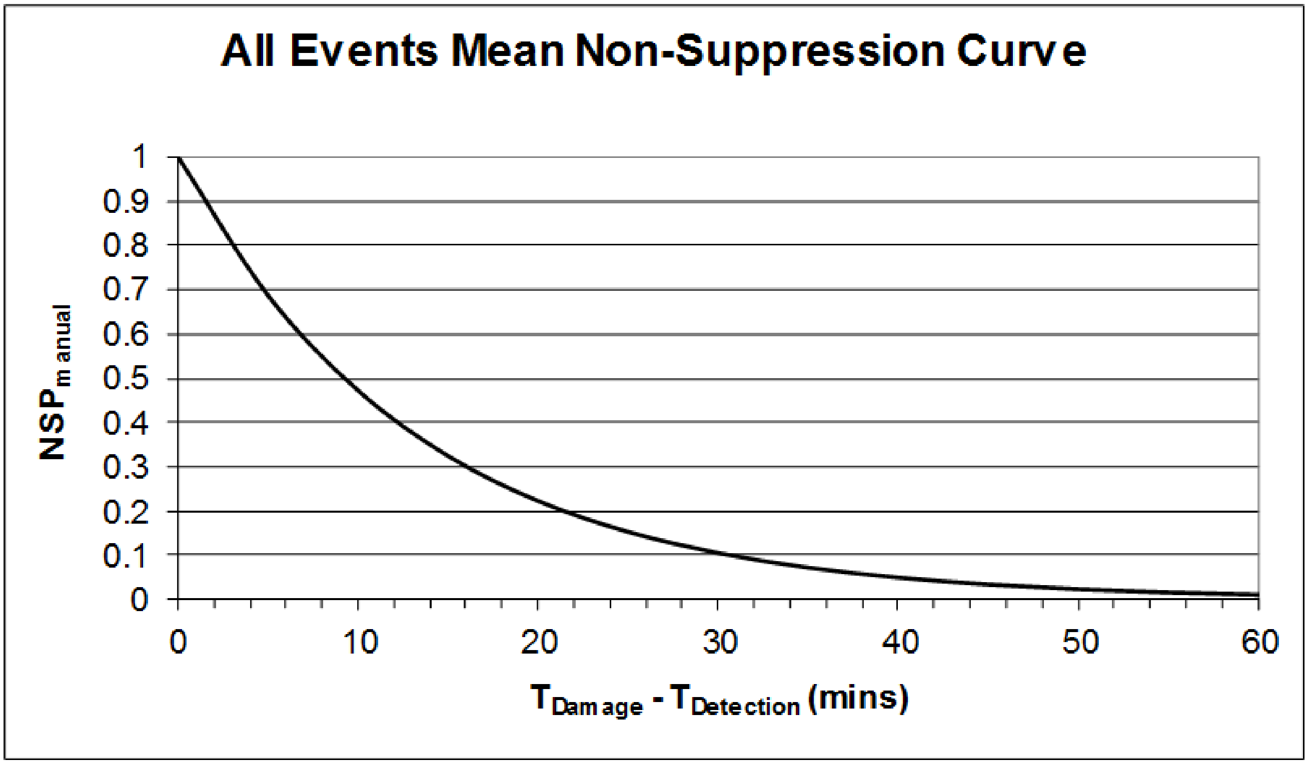

Figure 6: Example non-suppression probability over time.

The graph in Figure 6 shows how the non-suppression probability approaches 1 on the NSP axis as the time to detection approaches the time to damage. Removing the conservatism in either the time to damage and/or the time to detect can have a significant impact on the fire scenario in question.

2.4 Limitations/Issues

The primary limitations are due to the modeling assumptions and uncertainty within the model. Some are unique to fire modeling while others are common limitations for PRA in general.

- **Modeling limitations** – This is a PRA limitation where not all components or subcomponents are incorporated into the PRA. This is due to time and budget limitations for PRA staff, but it can prevent specific failure modes or actions from being performed if the component is not included into the PRA. This issue is mitigated to the extent to which one thinks an adequate representation of the plant can be made with the components required to simulate it. This limitation can be seen primarily in what is included into the model and what is selectively left out. Whether it is specific components or the detail included in scenario modeling. In the case of modeling fire scenarios, modeling conservative fire scenarios is simpler than creating more accurate individual fire scenarios. Excluding severity levels of the fires (what types of fires will damage what components), including failures of nearby equipment into scenarios, and assuming conservative failures of local fire suppression can all be examples of ways to reduce the detail in PRA modeling but introduces conservative assumptions into the modeling that increase the calculated level of risk to the plant. The limitations can be reduced by incorporating more detail into the PRA model but it is impossible to eliminate completely. It is up to the PRA developers to decide what level of assumptions and generalities are appropriate for a realistic model.
- **Operator action limitations** – The PRA currently can only take modest credit for operator actions and cannot take positive credit for actions that are implemented ahead of schedule.

Gaining credit for sped up actions that would improve the likelihood of successfully completing other actions is not taken into account within the PRA. Another limitation is the ability to only model singular steps or strings of simple steps as chains of actions are difficult to create a basis for accuracy. When specifically looking at operator actions for suppressing fires being able to credit chains of manual actions or more specificity for timing of operator actions will help improve the time to detection and suppression of the fire. This will give the plant a larger credit for the number of fires that will be suppressed before the fire can actually damage plant equipment. The credit will reduce the number of fires that create fire scenarios and reduce the overall plant core damage frequency.

- **Cable tracing issues –** This is a fire specific limitation, not all cables are precisely tracked within a nuclear power plant. When developing a scenario, if you are unsure of the location of a cable and cannot say with certainty that it would be excluded from a potential fire, than it must be considered to be included within the target sets. This issue is mitigated with better cable tracing to ensure specific cables (especially high risk ones) are excluded from as many target sets as possible.
- **Fire modeling limitations** – HRR profiles for initiating fires are not as accurate as real-world experience tells us. Using guidance HRR profiles are overly conservative and assumes fires that reach a severity level of concern are more likely than is actually the case. This is driving up risk as there is no other basis for PRA engineers to utilize a reduced HRR profile for specific fire events.
- **Timing and spatial limitations –** Currently it is easier to model conservative timing issues within potential fire scenarios. For example, if a fire initiates within an area in the plant it is often far simpler to conservatively assume immediate failure of components or not taking credit for operator actions within the given area. This simplifies the modeling but reduces the precision of the model. Taking credit for timing or spatial specifics within the area can help reduce the conservative assumptions placed on the fire scenarios. Specifically if a cable or manually operated valve is located in a far corner of the room it will take a long time for the cable or valve to heat up to the point of damage or make it preventable to be operated. Using more precise fluid dynamic modeling coupled with the timing of the scenario allows for credit of the components that may be utilizing cables and actions for operators that may be prematurely failed when that may not be the case, at least for a specific amount of time as the scenario develops.

3. DYNAMIC PRA

DPRA has been recently implemented in nuclear safety to capture timing aspects that are difficult to represent in purely numerical methods currently used. Some recently DPRA can also employ physics based simulator codes to accurately model physical phenomena with system dynamics. These physics simulator codes, like FDS, are coupled with DPRA codes that monitor and control the simulation (e.g., Event Model Risk Assessment using Linked Diagrams (EMRALD)). The latter codes, in particular, introduce both deterministic (e.g., system control logic and operating procedures) and stochastic (e.g., component failures and variable uncertainties) elements into the simulation.

A typical DPRA analysis involves the following four steps:

1. Sampling values of a set of parameters from the uncertainty space of interest
2. Simulating the system behavior for that specific set of parameter values
3. Analyzing the set of simulation runs
4. Visualizing the correlations between parameter values and simulation outcome

Step 1 is typically performed by randomly sampling from a given distribution (i.e., Monte-Carlo) or selecting such parameter values as inputs from the user (i.e., Dynamic Event Tree DET). In Step 2, a simulation run is performed using the values sampled in Step 1. These values typically affect the timing and sequencing of events that occur during the simulation. The objective of Step 3 is to identify the correlations between timing and sequencing of events with simulation outcomes (such as maximum core temperature). In a classical PRA (event-tree/fault-tree based) environment, such analysis is performed by observing and ranking the minimal cut sets that contribute to a top event (e.g., core damage). In a DPRA environment however, data generated is more heterogeneous since it consists of both a) Temporal profiles of state variables and b) Timing of specific events. The clustering and visually displaying failure data over time and space is a new research topic and it is especially relevant when analyzing time or order dependent sequences such as operator actions.

3.1 Physics Based Coupling

The power behind the RISA methodology is to closely combine multiple tools addressing various aspects of the scenario, for an accurate risk informed process. For fire analysis, we propose the use of the following tools.

3.2 FDS

The NIST developed computational fluid dynamics code FDS is utilized to perform computational fire modeling and simulation. The code has been extensively validated for the types of fire scenarios encountered in both standard building, as well as nuclear environments. FDS facilitates simulation of combustion, including fire migration, of an arbitrary number of materials in geometrically complex environments. Furthermore, FDS has several interfaces and plugins for additional capabilities such as the well-known evacuation simulation software called Evac.

FDS is an efficient CFD solver which relies on an orthogonal grid and allows for efficient numerical solution schemes which leverage Fast Fourier Transform (FFT) and may also be run in parallel utilizing both shared memory and distributed parallelization strategies (e.g. Message Passing Interface MPI). In contrast to other CFD-based fire simulation tools (e.g., the *OpenFOAM* [10] and *fireFOAM* [11] solver) which rely on a boundary fitted computational domain, mesh generation and domain construction in FDS is comparatively simple. Boundary conditions may be set via explicit definition in the user input file, which consists of a user readable Fortran namelist file.

Surfaces, such as walls or building facade elements, are represented as obstructions within the FDS domain. FDS does not include a preprocessor, but such a preprocessor can be constructed using computational geometry algorithms. Effectively, an octree-based grid representation is parsed and cells which intersect with a geometric boundary are labeled as obstacles. These obstacles called *OBST* correspond to a nameless entry within the FDS input file and may consist of an arbitrary number of obstacles and wireframe or *MESH* objects. In addition to the obstacles, the file also includes the boundary conditions with a directive called SURF and its associated ID. Details of the FDS mesh, boundary, and obstacle representation can be found in the FDS User and Theory guides.

FDS includes combustion and pyrolysis models suitable for the simulation of combustible solid materials and fire migration. Combustion products are tracked and transported throughout the simulation domain, and may be utilized to calculate important derived quantities (e.g., visibility). Expanded development of Evac like user-agents may also allow for customized operator procedures that react, dynamically, to a fire scenario. Quantities such as levels of obscuration may affect the predicted path, reaction time, and suppression for a given location and fire.

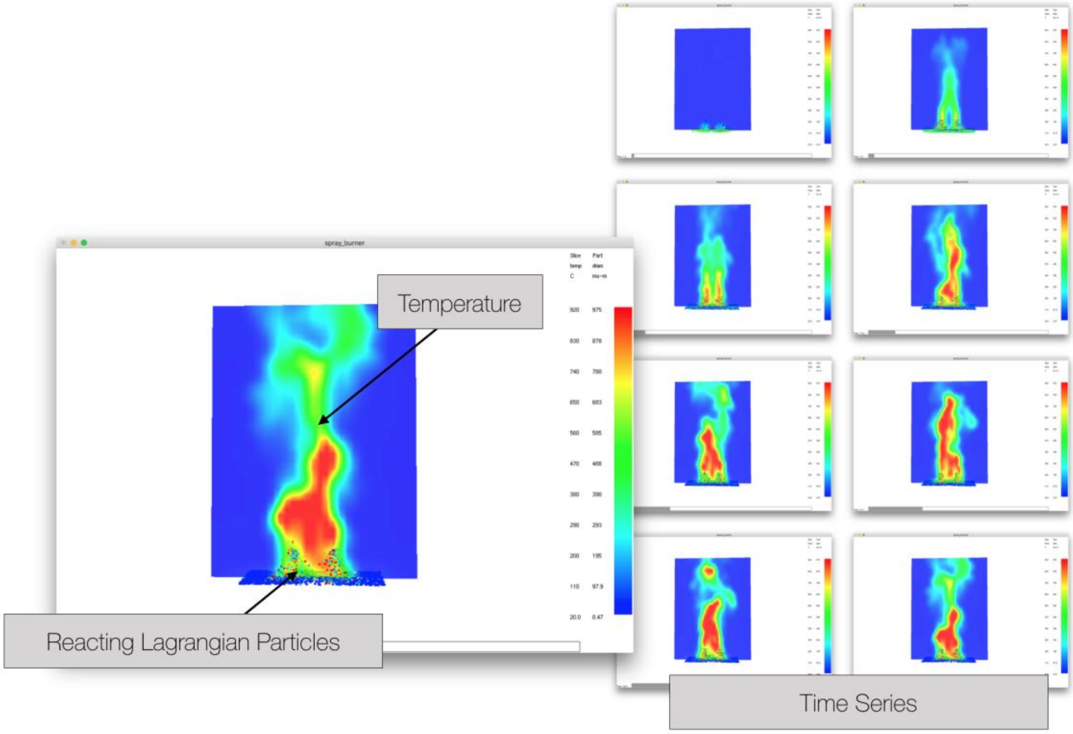

Figure 7: A standard FDS validation case which includes reacting Lagrangian particles interacting with an Eulerian flow field.

FDS solves equations that describe the evolution of fire. FDS solves a form of the Navier-Stokes equations appropriate for a low-speed, thermally-driven flow numerically, with an emphasis on smoke and heat transport from fires. Figure 7 shows the calculated temperature on a 2D slice through the domain is shown. Lagrangian particles (fuel droplets) which react with air. Turbulence is treated by means of Large Eddy Simulation (LES). It is possible to perform a Direct Numerical Simulation (DNS) if the underlying numerical mesh is fine enough. LES is the default mode of operation. Combustion Model For most applications, FDS uses a single step, mixing-controlled chemical reaction which uses three lumped species (a species representing a group of species). These lumped species are air, fuel, and products. By default, the last two lumped species are explicitly computed. Options are available to include multiple reactions and reactions that are not necessarily mixing-controlled.

Radiation Transport Radiative heat transfer is included in the model via the solution of the radiation transport equation for a gray gas, and in some limited cases using a wide band model. The equation is solved using a technique similar to finite volume methods for convective transport, thus the name given to it is the Finite Volume Method (FVM).

FDS approximates the governing equations on a rectilinear mesh. Rectangular obstructions are forced to conform to the underlying mesh. Any model which is presented to FDS would have to be discretized in the form of a set of rectilinear meshes.

It is possible to prescribe more than one rectangular mesh to handle cases where the computational domain is not easily embedded within a single mesh. FDS also employs OpenMP, a programming interface that exploits multiple processing units on a single computer. For clusters of computers, FDS employs MPI.

All solid surfaces are assigned thermal boundary conditions, and information about the burning behavior of the material. Heat and mass transfer to and from solid surfaces is usually handled with empirical correlations, although it is possible to compute directly the heat and mass transfer when performing a Direct Numerical Simulation (DNS).

The above listed features of FDS are quite important to model the physics behind fire accurately and predictably with minimum uncertainties and make it an appropriate tool for use in DPRA. In addition to accurate physics modeled by FDS, it is also possible to add extinguishers and specify exact material properties and the combustion temperatures of objects which could potentially ignite and cause a cascade of fire. The primary quantities of interest, Heat Release Rate, Temperature, Pressure, Density, and Velocity, are tracked by FDS. These measurements are important quantities to describe conditions of failure for critical equipment monitored by the DPRA system.

3.3 Neutrino

Neutrino [12] is a dynamic modeling, simulation and visualization environment. It can be used to setup a variety of scenarios involving computational fluid dynamics and its associated effect on solid rigid objects. The Neutrino's core set of solvers are based on the Smoothed-Particle-Hydrodynamics (SPH) [13] approach. However, the platform is flexible to accommodate grid based solvers such as FDS as plugins to its internal grid based system. The flexibility of Neutrino simulation environment allows interchange of grid based and particle based data thereby allowing a flexible coupling architecture whereby the correct solver can be used for the appropriate purpose.

Neutrino also provides an easy to use 3D modeling environment with methods to dynamically couple with EMRALD. It allows for quick setup to the underlying FDS model vs. current methods, which makes up for any additional time needed for coupling the DPRA model. FDS simulations are not inherently dynamically interactive. To overcome this Neutrino will be able to take the dynamic parameters from the EMRALD simulations and prepare the FDS simulation then revert or rerun the simulation where needed in order to capture required behavior. The underlying mechanism of how this would be implemented are outlined in the following sections.

3.4 EMRALD

EMRALD [14] is a software tool developed at INL for researching the capabilities of DPRA with other physics-based simulation tools. DPRA is ideal for fire analysis because of the multi stepped operator procedures, timing aspects and fire simulations currently used for analysis. EMRALD will be able to provide more accurate modeling and couple with the fire simulations to eliminate many conservatisms used in traditional PRA modeling.

In order to promote the effective use of DPRA by the general community, EMRALD focuses on the following key aspects:

1. Simplify the modeling process by providing a structure that corresponds to traditional PRA modeling methods.

2. Provide a user interface that makes it easy for the user to model and visualize complex interactions.

3. Allow the user to couple with other analysis applications such as physics based simulations. This includes one-way communication for most applications and two-way loose coupling for customizable or applications with an open Application Programing Interface (API).

4. When calculating results provide the sequence and timing of events that lead to the specified outcomes.

Traditional aspects of components with basic events, fault trees, and event trees are all captured in a dynamic framework of state diagrams, displayed in a user-friendly modeling manner. Each component is represented by a compact state diagram with basic events driving the current state of that component. A logic tree using those components, similar to a traditional PRA fault tree, can be evaluated dynamically during the simulations. Finally, event trees are captured in a plant response diagram, with events (including those from the dynamic logic evaluation) driving an end state result. This approach allows the user to implement dynamic methods with only needing to learn the dynamic state aspects of the model. See the EMRALD website for more detail.

After running the EMRALD model, the user is able to not only obtain probabilistic results, but also able to see the dynamic benefits such as timing and event sequences for specified simulation results. Additionally an open standard for communication is used which allows for coupling to other simulation-based or physics-based analysis, allowing the user to include complex phenomena simulation capabilities such as flood or fire analysis directly in the PRA model.

3.5 SAPHIRE

Nuclear facilities have existing PRA models, and these models can be used in combination with DPRA. Any scenario relevant dynamic aspects of a system, such as operator actions, can be removed and added to the DPRA model. These models then can be used directly in the dynamic simulation by adjusting the inputs into the model, solving, and evaluating the change in probability.

SAPHIRE is a probabilistic risk assessment tool was developed for and is used by the NRC and other entities. [5] There are models for each of the nuclear power plants, and a generic model used for research. With direct access to the software and code it is ideal for use in evaluating this fire analysis process.

4. DYNAMIC FIRE ANALYSIS

4.1 Approach & Benefits

The dynamic fire analysis will preliminarily look at three areas where DPRA would be the most cost efficient and effective in more accurately modeling fire risk. First, are the effects on the fire brigade response times and impacts to the Non-Suppression Probability of fires. This would allow for better credit for a procedural plant response that can be used to mitigate fires. The hopes are that fire responses can be more accurately represented in the hopes of reducing the numbers of fires that propagate to the level of damaging fire equipment. This would reduce the likelihood of sever fire within the plant and hopefully reducing the overall risk to the plant in general from fire events. The benefits to the plant would be better assessment of risk and being able to more accurately represent the plant response to accident scenarios without having to create modifications or other changes that would require costly implementation. This first area would also allow for easy evaluation of the methods and be a stepping stone to more advanced analysis. This can be further developed into more advanced procedure dominated scenarios like control room abandonment, or lead to the evaluation of procedure changes to see if optimizations could be made while maintaining specified safety level.

Second, this method will allow for more specific timing requirements for the specific HRR scenarios and allowing for more precision when either identifying the time to damage specific components or more precision in the timing to detect and suppress fires with either automatic suppression or manual suppression or a combination of both. The DPRA allows for feedback loops where parameters may change within the model (was a vent opened, did a detector malfunction, etc.) that can change the HRR profile and damage times. It also allows for the user to incorporate more complex operator actions as the modeling is less conservative and the impacts of successes and failures of chains of actions can be incorporated directly into the dynamic model using the feedback loops.

Third, we will evaluate the ability to credit component failure times and the delta Core Damage Frequency (CDF) against the conservative fire scenario. For a sampling of simulations, the time until damage will be measured and logged. These failures can then be clustered and input as flags into the PRA model in order to show the plant state and changes in CDF over time vs immediate failure. This will allow us to determine the significance of failure timing, to determine if there is enough benefit to peruse acceptable methods. The specific benefits to the Fire PRAs will likely be from various reasons and each scenario will be specific but the dynamic approach will allow an easy way to incorporate all of the modeling improvements to the scenarios and assess the benefits of each.

4.2 Requirements

The following are the key model components that need to be developed to perform the analysis:

4.2.1 Models

- EMRALD – State model containing the initiating events for the fire scenarios, plant status and response to events, and operator actions.
- 3D Fire Model – 3D model of area and components for the fire simulation.
- Data – Material properties, heat release curves, etc. as inputs to the models.
- PRA model – A generic plant PRA model to evaluate the effects of the failed components and determine the status of the plant.

The following are available from previous work and will use them for initial demonstration of the methods:

- A 3D model of a switch gear room.
- A generic Pressurized Water Reactor (PWR) PRA model.

4.2.2 Fire Scenarios

In order to effectively and most accurately demonstrate the benefits of these tests, scenarios of high significance need to be determined and used. Each fire scenario will have individual limitations incorporated into the modeling and the trick will be identifying just which DPRA methods can improve them and just how to model the changes. Incorporating the DPRA methods into these scenarios and quantifying the delta Core Damage Frequency (CDF) will allow for an accurate and quantitative assessment to see just how much benefit DPRAs can provide to the Fire PRA. FDS has directives which can specify a ramp or curve for heat release rate into the system with time. The information to FDS can be communicated from the EMRALD hazard curves.

4.2.3 Fire Suppression Systems

FDS contains models which represent deactivation conditions of active fire suppression and monitoring devices, such as smoke detectors and heat sensors (see Figure 8). These models can be utilized to trigger events such as the activation of an active fire protection system, such as sprinklers. These models can be utilized without modification to the existing FDS code, and can be integrated into any DPRA system which utilizes FDS. Location activation conditions for simulated detectors and active suppression devices are set a priori in the user-defined input file. Other items like sprinklers may be activated through triggers as a function of the value of a given field variable in the region local to the device or detector as shown in Figure 8. The *sprinkler* object serves as a Lagrangian particle emitter which emits reactive particles (i.e., H_20) into the flow field to facilitate physics-based modeling of active fire suppression systems.

```
&DEVC ID='HD_66', PROP_ID='Acme Heat', XYZ=2.3,4.6,3.4 /
&PROP ID='Acme Heat', QUANTITY='LINK TEMPERATURE', RTI=132.,
    ACTIVATION_TEMPERATURE=74. /
```

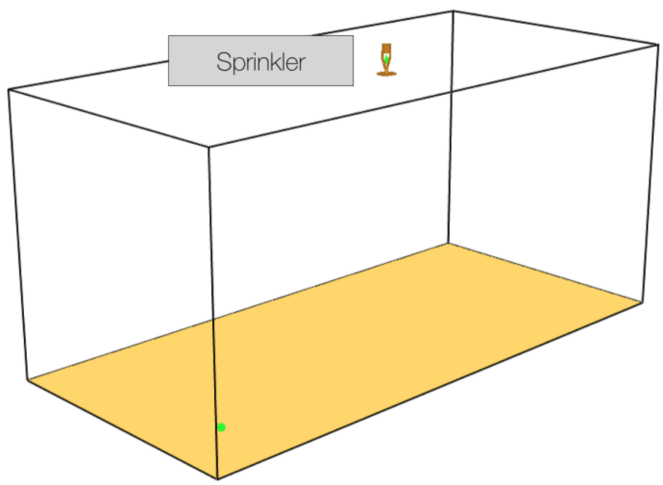

Figure 8: A *sprinkler* device as visualized using the *Smokeview* post-processing tool.

Key areas of this work will be the addition of active or operator action fire suppression. While FDS does have the capability to model many standard active fire suppression systems (sprinklers, for example), many FDS modeling methods must be specified a priori in the simulation input file, and therefore are not able to capture complex/operator-driven active or adjustable suppression method. This is where the EMRALD model and Neutrino come into effect. The EMRALD code will sample conditions and pass parameters to Neutrino which will dynamically adjust or set up the FDS model at runtime, both before and potentially during a given simulation. The sampling will predetermine simulation parameters such as heat release rates and active response times. This will allow for many variations of the simulation runs, producing probabilistic results for suppression, component failures, and timing.

This implements a deeper integration between the FDS active fire suppression system and detection models with a PRA model through a message based communication protocol. The communication model to be implemented as part of linking FDS to Neutrino or directly to any DPRA system is discussed in detail in 4.2.5.

4.2.4 Quantities of Measurement

Any field variable calculated by FDS may be utilized by the DPRA system. Common field variables of interest include temperature, the concentration of oxygen or combustion products, soot, velocity, etc. These quantities are available in several formats. Derived quantities, such as validated methods for quantifying visibility in smoke-obscured environments, are also supported.

FDS can output specific quantities at user-defined gage locations, defined a priori in the FDS input file. FDS will also output volumetric properties on the complete simulation grid, which may be utilized for subsequent post-processing. At present, in memory access to field variables in real time is not facilitated (one would need to watch a directory, at present, and parse output files as they are written, likely not practical, particularly when considering large simulation domains which would need to be written and processed at sufficiently small output intervals). This is not needed for this initial work. However, the implementation of a socket or event-based communication model as described subsequently, may allow real-time in memory access by external codes for more advanced analysis in the future.

For this initial work direct two-way coupling will not be needed and final result parameters can be used to determine statistical outcomes. If a set of fire simulation results indicate that a predetermined action is not possible, such as too much smoke or lack of oxygen exists for an assigned preset manual suppression attempt, that simulation will be rerun with that action excluded.

4.2.5 Communication

The EMRALD DPRA system will maintain the overall control system for the fire scenarios. Two main FDS features will allow for successful communication from the main control system. First, FDS allows a simulation to be stopped and restarted at a specific point by means of maintaining appropriate files which store the state of the simulation as it proceeds. When a signal from EMRALD is received to terminate, pause the Fire simulation, Neutrino will handle the message and manage the files and processing this functionality in FDS. Second, FDS has a limited set of events are available and can be dynamically triggered as a function of a field variable, this is described in more detail in section 4.2.6. This will allow EMRALD messages to trigger things like fire suppression at an arbitrary time.

Parameters for the initial FDS simulation setup will be sampled by EMRALD prior to execution. In most cases a direct run of the FDS simulation with the initial parameters will be all that is needed. However in some scenarios, events after the FDS simulation has started, may trigger the need to adjust the simulation setup. Since FDS's initial state is determined a-priori any changes to the scene would need to involve a customized approach.

Two methods can be used to obtain dynamically adjusted results from a FDS simulation scene. First, when a message to Neutrino indicates a change to the FDS simulation, Neutrino can reconstruct the FDS model/setup and rerun the FDS simulation. This would give the same results as if there were a dynamic coupling to the FDS simulation. However, this would have high computational cost for any adjusted simulations. The second method would be to develop an in-memory/message-based communication model. This model will require some minimal modification of the existing FDS source code (FDS is an open source code, so access is not an issue). These modifications will also be submitted back to the developers at NIST for inclusion in the standard FDS distribution if they so desire.

Although it may be possible to utilize FDS outputs in pseudo-real-time via a file-based approach, this is likely impractical when considering the scale of simulation required to model realistic fire scenarios of interest to analysts. The problem becomes particularly onerous when one considers that in order to access many of the variables of interest, without a priori knowledge of where quantities will be need to be sampled (for example by utilizing a gauge for probe), the entire simulation domain and field variables that would need to be written to disk, and interpreted at each time interval. This would be impractical due to the significant storage requirements, input/output constraints, and speed limitations.

To quickly evaluate the benefits of this dynamic Fire PRA methodology, the first option will be used. An evaluation of the increase of efficiency by using the second method (i.e., how many times it restarted) would also be made to determine if a modification effort would be worth the benefit.

EMRALD is using a general communication protocol running over the Extensible Messaging and Presence Protocol (XMPP) framework. Although it was designed originally for flood simulations, the recent modifications to generalize the protocol means little to no modifications should be needed to successfully run the fire simulations.

4.2.6 Simulation Events

FDS allows for events to be dynamically triggered as a function of a field variable, and has support for triggering the activation of active fire suppression systems, Such as sprinklers etc., utilizing this methodology. Measurement points can also be added to the FDS simulation to return data back to the DPRA model for component failures, such as cable tray temperature above the fire. This will then be combined with component failure or success data for probabilistic instead of deterministic evaluation of components. For example, currently a FDS fire detection element will be triggered if the temperature/smoke reaches a set number. When coupled with the DPRA the detection can be a probability associated with the temperature/smoke level.

These basic FDS triggers may be extended to accommodate a custom set of triggers and events. An example of this functionality is implemented as part of Evac, and is utilized to trigger the movement in response of user agents within an evacuation simulation. As stated in the FDS/Evac guide [15] the pre-movement time of the evacuating agents is decided by the user input by giving distribution for the detection and reaction times. In addition to the detection time given by the user, the local smoke concentration can be used to trigger the detection of a fire and can also be connected to the control logic of FDS. This is facilitated through and interface with the smoke and heat detector input lines called PROP/DEVC, and through extension of the corresponding Fortran modules. As an example, a smoke detector is defined in an FDS case input file with an entry similar to:

```
   &DEVC ID='SD_29', PROP_ID='Acme Smoke Detector', XYZ=2.3,4.6,3.4 /
   &PROP  ID='Acme  Smoke  Detector',  QUANTITY='CHAMBER  OBSCURATION',
LENGTH=1.8,
         ACTIVATION_OBSCURATION=3.24 /
```

More complex models, such as the four parameter Cleary model may be specified as follows:

```
   &PROP ID='Acme Smoke Detector I2', QUANTITY='CHAMBER OBSCURATION',
         ALPHA_E=1.8, BETA_E=-1.1, ALPHA_C=1.0, BETA_C=-0.8,
         ACTIVATION_OBSCURATION=3.24 /
```

This functionality may be extended to the existing code base to include models for operator mitigation actions and event triggering based on reaction time. For example, the simulation of an operator to evaluate a scene, acquire an extinguisher and suppress the fire, instead of simply exiting the premises as is the current case for Evac. These events may also include stochastic or other non-deterministic components as part of the underlying model.

Initial work will just use sampled times for operator responses and assign the numbers to the FDS model, follow-on work could focus on areas such as this to further increase the accuracy of the model.

The process of triggering an event via the extension of the heat/smoke detector models in FDS achieved by first implementing a new PROP and DEVC types, both of which may be included in the user-defined input file the code defining the functionality of a heat/smoke detector may be utilized as a template and modified to accommodate an appropriate model for human observation in response. The implementation of a socket based event based communication model is a natural extension to FDS and Evac, will facilitate coupling with external codes such as DPRA analysis tools in a natural, and lightweight matter.

4.2.7 Early Terminations and Resuming Calculations

DPRA analysis needs simulations to be paused and continued sometimes due to some change in the parameters. To do this, there is a file with an extension stop. FDS checks for the existence of this file at each time step, and if it finds it, gracefully shuts down the calculation after first creating a final Plot3D file and a file called CHID.restart (or CHID_nn.restart). To restart a job, the file(s) CHID.restart would be present working directory, and the phrase RESTART=.TRUE. will be presented as a MISC directive to FDS. The existence of a restart file with the same CHID as the original job tells FDS to continue saving the new data in the same files as the old.

It is also possible to use the new control function feature as specified in the FDS User Guide to stop a calculation or dump a restart file when the computation reaches some measurable condition such as a first sprinkler activation.

However, between job stops and restarts, major changes cannot be made in the calculation like adding or removing vents and obstructions. The changes are limited to those parameters that do not instantly alter the existing flow field. In other words adding new obstructions in the middle of a fire plume would create instabilities and might crash the simulation. Any change to the scene and characteristics should be gradual.

4.2.8 Visualization & Post Processing

As stated, the benefit of simulation is to know the progress of event over time. Given a set of simulation results, post processing can determine clustering of common scenario events. Additionally, the fire simulation setups can be saved so that users can rerun selected scenarios and view the events in the 3D simulation. Both of these tools can help determine where to focus mitigation efforts for significant reduction in risk. Currently the visualization and post processing capabilities for fire has been in the works and being added to the Neutrino platform. FDS produces a few output files, one set of outputs is a sequence of data in which parameters like temperature, HRR, pressure, density, and velocity can be output. The format of these files is written as an industry format Plot3D data. Neutrino parses these files and uses a custom nVIDIA Computer Unified Device Architecture (CUDA) based rendering engine based on GVariant DataBase File (GVDB) [16] to create value added visualizations of the fire. The images in Figure 9 indicate a visualization of a sample fire setup in which the components are visualized changing color as their critical failure temperature is reached. nVIDIA along with Centroid LAB have jointly partnered to produce a visualization platform to visualize in 3D, results from an FDS simulations in real time.

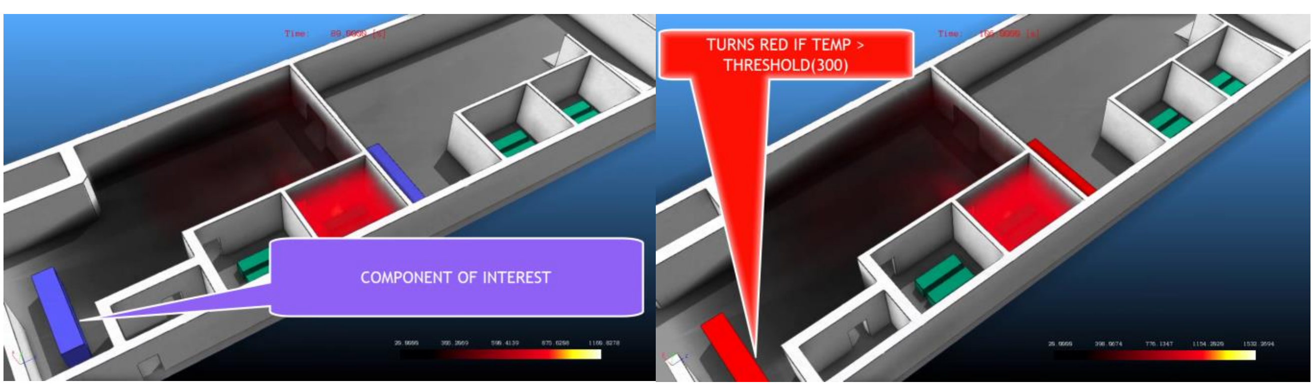

Figure 9: Example of visualization capabilities in Neutrino.

.

It is anticipated that this platform will be extended to perform more advanced visualizations and post processing of data and simultaneously link it with the event progression in EMRALD so that the user can visualize the 3D simulation as it proceeds along with the sequence of actions which lead to the critical outcome.

4.3 Setup in FDS & Neutrino

4.3.1 Test Setup – Fire in a Switch Gear Room and Suppression

Initial steps of the following scenario of a fire which ignites in a switch gear room of a nuclear power plant, were tested with FDS and Neutrino. The FDS fire simulation emulates a short circuit, causing a fire, until there is an alarm and the operator notices the fire and gets an extinguisher to put out the fire. Slow burn electrical cabinet fires, as in this scenario, have a higher frequency and can progress to become significant. Several key parameters in the scenario are identified which can be evaluated effectively using DPRA analysis, namely operator action, dynamic suppression, heat release time, temperature etc. A full Neutrino and FDS model coupled with a DPRA model of this example will evaluate its feasibility and setup costs.

4.3.2 Model Preparation for FDS Simulation

Since all calculations in FDS have to be performed in rectilinear volumes called meshes, one such has to be setup and passed on to the simulation. The resolution of the model relates to the accuracy of the simulation so that is generally adjusted empirically to the desired accuracy needed. Some key dimensionless numbers are determined, in the case of the control room fire this was 128 cells along the width 256 cells along the length 64 cells in the vertical direction. The simulation domain itself is 6 m x 24 m x 2.5 m as indicated in the Figure 10.

There are several components, rooms and obstructions in the control room. A simplified version of the control room is modeled in 3D and exported as Computer-Aided Design CAD geometry in the form of stl files. FDS however needs a rectilinear volumetric decomposition of the surface geometry. This process involves advanced computational geometric algorithms and these algorithms have been implemented in the Neutrino environment. This simplifies the process of generating a decomposition of a complex CAD model into bounding boxes of various obstacles participating in the simulation. The various OBST directives are output and placed in the FDS input file. Figure 10 illustrates such a decomposition. The rectangular bounding boxes of the various volumes participating in the simulation are indicated by black bordered line.

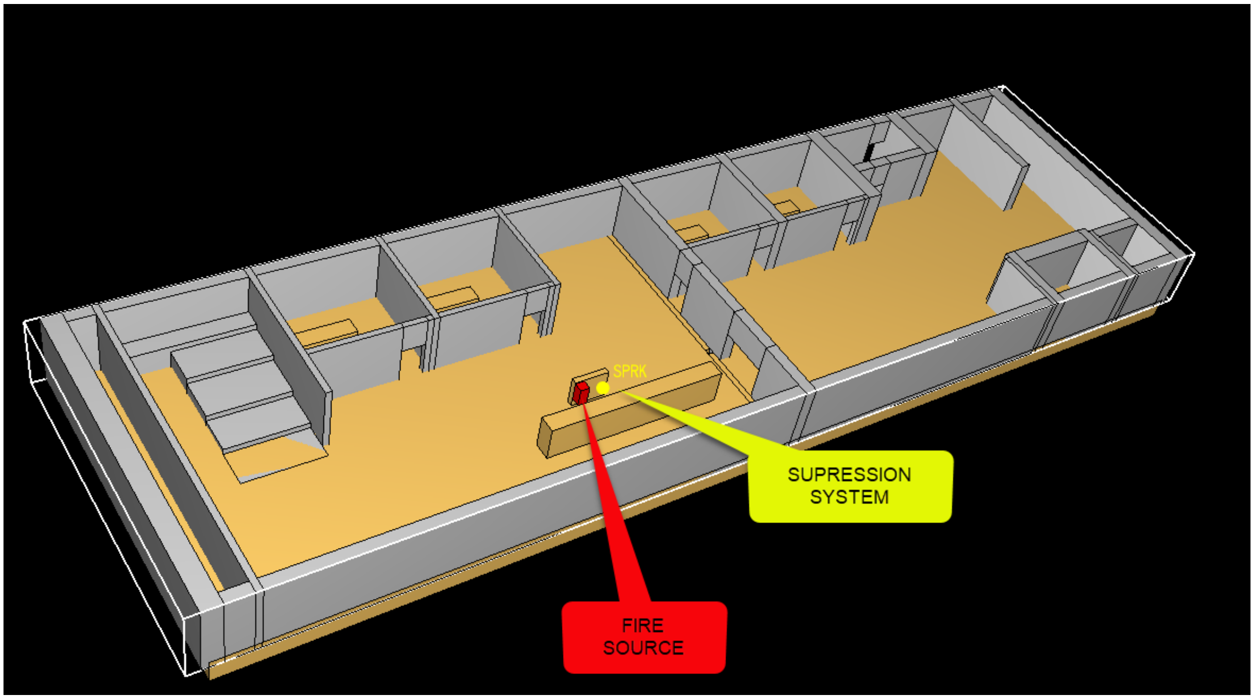

Figure 10: 3D bounding box layout of switch gear room with fire source and anticipated suppression.

4.3.3 Model and Fire Visualization in Neutrino

Once the input FDS file is setup, the simulation is run on multiple cores. In initial evaluation, the entire scenario is simulated with one computational mesh for reasons of convenience. If a higher resolution scenario needs to be simulated the computational mesh can be split up into multiple meshes and distributed among several machines.

In the sample setup a trigger event (Figure 11) is set and the heat release was specified as heat release per unit area with the values at 1000 kW at 10 seconds into the simulation. The operator reacts to the fire at 5 seconds into the simulation and tries to extinguish the fire with water. The fire finally gets fully extinguished at 12 seconds. The simulation sequence is visualized in Neutrino as the operator responds to the fire by arriving at the site to put it out (see Figure 12). Although this situation depicts a situation in which the operator reacts too quickly and the fire grows quite fast, it was chosen mainly for simulation speed and evaluate capabilities. In a real test case, these parameters will be determined by sampling realistic rates.

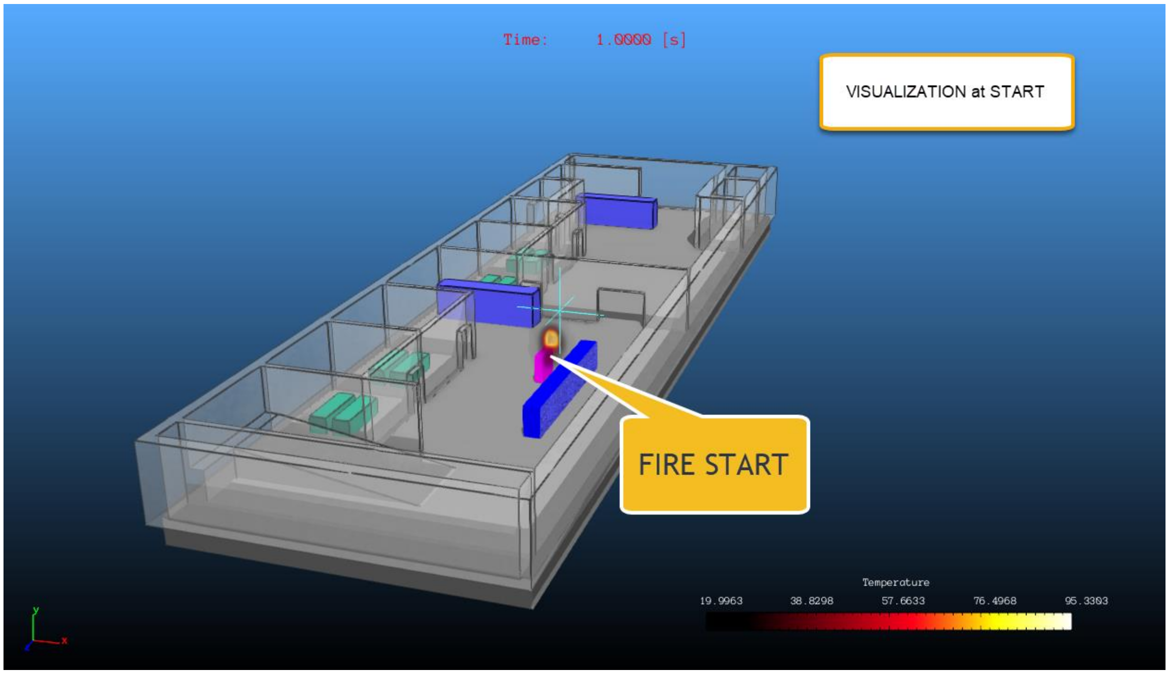

Figure 11: Fire visualization showing the trigger event - fire ignition.

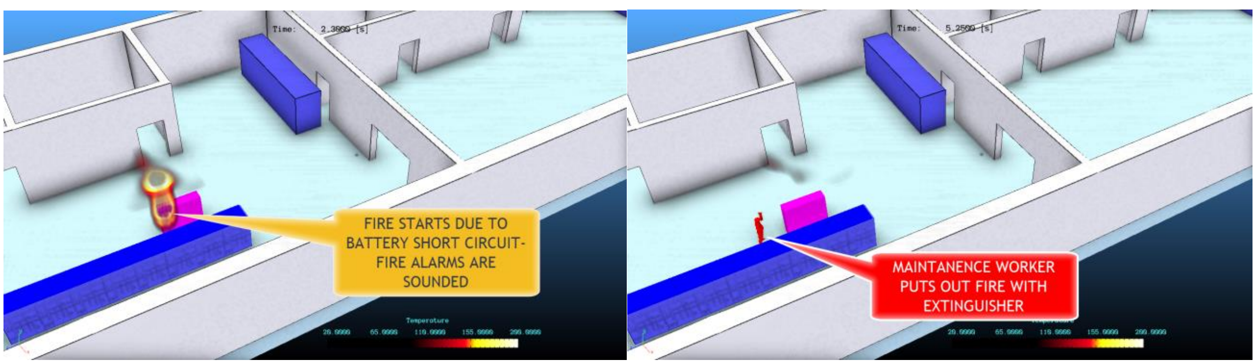

Figure 12: Visualization of fire being suppressed by an operator.

4.3.4 Probabilistic Setup

The overall simulation will be driven by an EMRALD model of operator actions and event probabilities. For this initial evaluation, the system was setup using Neutrino and FDS and the peak HRR curve was done manually. Future work will have the detection and operator response times sampled from probability curves and possibly even variations of the HRR curve, supplied by EMRALD and a range in the hazard curves. Compiling multiple simulation results will enable a probabilistic outcome for the given scenario.

4.4 Expected Benefits and Results

To assess the benefits of the dynamic Fire PRA the best approach will be to look at the change in CDF for the scenarios or the fire areas in question. If you can find a significant reduction in CDF by reducing the number of conservative assumptions or allowing for the accurate credit for actions and components it will allow the plants a better idea and a more accurate portrayal of the fire risks to the plant. A crucial aspect showing that DPRA modeling of these scenarios isn't going to require a great deal of additional effort vs. current FDS modeling efforts. If significant sequences require fire simulation to reduce the CDF of the plant, then with little effort a significant gains can be made from DPRA modeling in the following areas:

- Operator action fire suppression – Reducing the overall significant fires by accurately showing manual suppression actions.
- Probabilistic fire detection – Add probabilistic capabilities to detection element in the FDS model.
- Component failure timing – Generate component failure timings for significant fires and uses these generate change sets for the PRA model for a delta CDF.
- Visualization & Mitigation methods – Easy methods to evaluate critical scenarios and then develop and test mitigation hardware or procedural options.
- Simplification of PRA model – Dynamic aspects of a model are simpler to depict in EMRALD. This allows for a more accurate model while reducing the initial modeling time and maintenance efforts.

Secondary economic benefits could also develop from utilizing the dynamic models vs a static model. These may include the following:

- Identifying the reduction in time and effort for PRA analysts to perform detailed fire modeling in Neutrino.
- Identifying the added benefit in risk reduction from incorporating the dynamic model and how much extra effort is required.
- Reducing CDF may allow the plants to lift some precautionary measures they are crediting in their fire protection plans.
- Reducing maintenance schedules or planned modifications to protect plant equipment thought to mitigate fire risk.
- Optimizing protection plans to maintain safety while minimizing labor resources.

While the main goal is to correctly model the overall risk to the plant, it is imperative that DPRA also provide additional economic benefit to the plants as they would be more likely to utilize the tools that are developed.

4.5 Limitations

The main limiting factor a dynamic Fire PRA model is the computation time. FDS is computationally expensive and in order to obtain probabilistic results, requires more simulations than current analysis. Without modifications to the FDS software (which is open source), some feedback loop scenarios may be limited or not possible. These limitations will not be fully known until specific scenarios are encountered and implemented.

REFERENCES

1. Curtis Smith, Cristian Rabiti, R.M.R.S.: Risk-Informed Safety Margins Characterization (RISMC) Pathway Technical Program Plan, INL/EXT-11-22977, September 2016
2. NUREG-1824, EPRI-1011999: Verification and Validation of Selected Fire Models for Nuclear Power Plant Applications Volume 1: Main Report. May 2007
3. NUREG-1934, EPRI-1023259: Nuclear Power Plant Fire Modeling Analysis Guidelines (NPP FIRE MAG). Final Report. November 2012
4. ASME: RA-Sb-2013 Addenda to ASME/ANS RA-S-2008 Standard for Level 1/Large Early Release Frequency Probabilistic Risk Assessment for Nuclear Power Plant Applications, 2008
5. Idaho National Lab, SAPHIRE, http://saphire.inl.gov, 2014
6. Office of Nuclear Reactor Research, Fire Dynamics Tools (FDT s) Quantitative Fire Hazard Analysis Methods for the. Readings. 1, 2004
7. Richard Peacock, Paul Reneke, NIST Special Publication 1026 CFAST – Consolidated Model of Fire Growth and Smoke Transport (Version 6) Technical Reference Guide. Nist Spec. Publ. 1026, April 2009.
8. L. Gay, “User Guide of MAGIC Software V4.1.1,” EdF HI82/04/022/B, Electricité de France, France, April 2005.
9. McGrattan, K., Hostikka, S., McDermott, R., Floyd, J., Weinschenk, C., Overholt, K.: FDS User’s Guide., 2004. doi:10.6028/NIST.SP.1019
10. Jasak, H., Jemcov, A., Kingdom, U.: OpenFOAM: A C++ Library for Complex Physics Simulations. In: International Workshop on Coupled Methods in Numerical Dynamics, IUC. pp. 1–20, 2007
11. Li, Y.Z., Huang, C., Anderson, J., Svensson, R.: Verification , validation and evaluation of FireFOAM as a tool for performance design., 2017
12. Neutrino Documentation, http://docs.neutrinodynamics.com
13. Monaghan, J.J.: An introduction to SPH. Comput. Phys. Commun. 48, pg 89–96, 1988
14. Idaho National Lab, EMRALD, http://emrald.inl.gov, 2018
15. Korhonen, T., Hostikka, S.: Fire Dynamics Simulator with Evacuation: FDS + Evac Technical Reference and User ’ s Guide, 2010
16. Guide, P.: GVDB Voxels. NVIDIA Digit. Libr, 2017, https://developer.nvidia.com/gvdb~~)~~