INS International Nuclear Security
Reducing Risk of Nuclear Terrorism

Methodology and Tool for the Physical Security Analysis of Micro and Advanced Reactors

August 2023

National Nuclear Security Administration

Intentionally blank

METHODOLOGY AND TOOL FOR THE PHYSICAL SECURITY ANALYSIS OF MICRO AND ADVANCED REACTORS

Idaho National Laboratory

Robby Christian, Christopher P. Chwasz, Michael A. Ziccarelli, Sai Zhang, Steven R. Prescott, Vaibhav Yadav, and Shawn W. St Germain

Argonne National Laboratory

Dave Grabaskas and Matthew D. Bucknor

Date: **August 2023**

Prepared for the

U.S. Department of Energy

Office of Nuclear Energy

Under DOE Idaho Operations Office

Contract DE-AC07-05ID14517

Intentionally blank

CONTENTS

LIST OF FIGURES v

LIST OF TABLES vii

ABBREVIATIONS, ACRONYMS, AND INITIALISMS ix

ABSTRACT xi

ACKNOWLEDGEMENTS xiii

Methodology and Tool for the Physical Security Analysis of Micro and Advanced Reactors 1

1. INTRODUCTION 1

1.1. Dynamic Physical Security Analysis in LWRS 1
1.2. Technology Gap to Address A/SMR Needs 2
2. METHODOLOGY AND TOOL DEVELOPMENT 5

2.1. Methodology 5
2.2. Tool Development 7
2.2.1. EMRALD 7
2.2.2. EMRALD Templates 8
2.3. Generic SFR Model 9
2.4. Generic HTGR Model 13
3. CASE STUDY 16

4. RESULTS AND DISCUSSIONS 22

5. SUMMARY AND FUTURE WORK 26

6. REFERENCES 28

APPENDIX A: How to Use EMRALD Templates A-1

APPENDIX B: SRT Code Parameters B-1

Intentionally blank

LIST OF FIGURES

Figure 1. MASS-DEF Framework. 2
Figure 2. Frequency-consequence curve. [5]....4
Figure 3. The A/C plot. 7
Figure 4. Pre-built template diagrams for A/SMR physical security. 8
Figure 5. Diagram for medium-level active delay system (ADS). 9
Figure 6. Generic SFR facility layout....10
Figure 7. Generic HTGR facility layout. 14
Figure 8. Example attack path on the reference SFR facility. 16
Figure 9. Attack scenario and response flowchart. 18
Figure 10. Main EMRALD diagram. 19
Figure 11. EMRALD subdiagram for the attack scenario. 20
Figure 12. StepFence_Done event. 20
Figure 13. EMRALD subdiagram modeling preventive safety actions. 21
Figure 14. EMRALD subdiagram modeling safety analysis. 21
Figure 15. Initial distribution of attack time. 22
Figure 16. Time histogram after security modifications. 23
Figure 17. Reduction in attack achievability and consequence. 24
Figure 18. Sankey diagram showing the branching of attack scenario. 25

Figure A-1. Right-click a diagram and select "Make Template." A-1
Figure A-2. "Create Diagram Template" window. A-2
Figure A-3. Prompt for group name when creating a new group. A-2
Figure A-4. Steps to create a sub-group. A-3
Figure A-5. Group view. A-3
Figure A-6. Tree view (collapsed). A-3
Figure A-7. Tree view (fully expanded). A-3
Figure A-8. The current path is displayed when in a group. A-4
Figure A-9. Create a new diagram. A-5
Figure A-10. Select template – "Tree View." A-5
Figure A-11. Select template – "Group View." A-5
Figure A-12. Select template – "List View." A-5

Figure A-13. “This Template Cannot Be Selected” message. A-6
Figure A-14. A valid template is selected. A-6
Figure A-15. "Import Diagram" window with conflict and required item. A-7
Figure A-16. Sidebar comparison without and with filters A-8

LIST OF TABLES

Table 1. Safety actions in response to adversary attacks in the generic SFR facility 12

Table 2. Safety actions in response to adversary attacks in the generic HTGR facility. 15

Table 3. Simulated outcomes of sabotage attack. 23

Intentionally blank

ABBREVIATIONS, ACRONYMS, AND INITIALISMS

A/SMR	Advanced/Small Modular Reactor
ANL	Argonne National Laboratory
CAD	Computer-Aided Design
CD	Core Damage
CNPRI	National Committee for Ionizing Radiation Protection
EMRALD	Event Modeling Risk Assessment using Linked Diagrams
F-C	Frequency-Consequence
FoF	Force-on-force
HTGR	High-temperature Gas-cooled Reactor
HVAC	Heating, Ventilation, and Air Conditioning
INL	Idaho National Laboratory
INSTAR	International Nuclear Security for Advanced Reactors
LWR	Light-Water Reactor
LWRS	Light Water Reactor Sustainability
MASS-DEF	Modeling and Analysis for Safety Security using Dynamic EMRALD Framework
NRC	Nuclear Regulatory Commission
PRA	Probabilistic Risk Assessment
SFR	Sodium-cooled Fast Reactor
SME	Subject Matter Expert
SQA	Software Quality Assurance
SRT	Simplified Radionuclide Transport
TRISO	TRi-structural ISOtropic particle fuel

Intentionally blank

ABSTRACT

This work proposes a dynamic evaluation methodology to relax the conservatism in physical security evaluation, by leveraging an ongoing work in the Light Water Reactor Sustainability pathway. This methodology is implemented in a dynamic risk assessment tool named Event Modeling Risk Assessment using Linked Diagrams (EMRALD).

The work extends EMRALD's capability to support a sandbox feature where analysts can easily create attack scenarios and modify advanced/small modular reactor (A/SMR) security and safety features using templates. This approach saves time and cost since the analysis does not require creating detailed computer-aided design models, as is commonly required in commercial force-on-force software tools. EMRALD is completely free to use at https://emraldapp.inl.gov. We have developed basic templates including physical barriers, intrusion sensors, physical areas, and safety actions, that can be downloaded from EMRALD's GitHub site: https://github.com/idaholab/EMRALD. **These templates use generic data commonly used for training purposes, which do not reflect any actual operating nuclear reactor**. Users may adjust the data in the templates with their own dataset and/or create new templates in EMRALD.

The proposed methodology combines security and safety by assessing sabotage effects up to the radiological consequence to the public instead of merely the core damage state. This practice follows the industry standard for advanced non-light-water reactors currently proposed for endorsement by the Nuclear Regulatory Commission. The combination of security and safety is expressed in an achievability-consequence chart. EMRALD can be used to generate data for this chart. A hypothetical case study using a representative sodium-cooled fast reactor (SFR) facility is presented in this report to demonstrate this methodology. **This case study does not contain any actual nuclear plant information**.

This work will benefit A/SMR vendors and utilities to implement security by design during the reactor design iteration phase, such that they do not have to perform upgrades and retrofits to the reactor after it is installed to improve its physical protection system. The tool may also be used to analyze domestic or foreign reactor designs to support the International Nuclear Security for Advanced Reactors (INSTAR) bilateral missions. Future works are planned to implement the methodology on a reference SFR reactor and a reference high-temperature gas-cooled reactor to obtain insights and lessons-learned for the A/SMR community.

Intentionally blank

ACKNOWLEDGEMENTS

We thank Argonne National Laboratory for their contributions on technical data and information regarding the reference sodium-cooled fast reactor facility. Note this work also uses the Simplified Radionuclide Transport created at Argonne National Laboratory and authored by Dave Grabaskas. We acknowledge the feedback from Douglas Osborne, Sondra Spence, Todd Noel (Sandia National Laboratories), and Anthony Qualantone (X-Energy) during the discussions throughout this work.

Intentionally blank

Methodology and Tool for the Physical Security Analysis of Micro and Advanced Reactors

1. INTRODUCTION

The coming generation of advanced and micro reactors advents new methodologies and tools to align with the anticipated increases in safety, deployment methodologies, business cases, and the performance-based and risk-informed regulation. The U.S. nuclear industry, advanced reactor vendors, and the U.S. Nuclear Regulatory Commission (NRC) have developed strategies to credit the increased safety of advanced reactors fulfilling the security goal of preventing radiological sabotage. The NRC has issued a draft rulemaking NRC-2017-0277, providing alternatives to the prescriptive physical security rules if a licensee can "perform a site-specific analysis to evaluate the potential offsite radiological consequences and demonstrate how the performance requirements set forth in § 73.55(b)(3) are met when selected alternatives are used." This performance-based regulatory framework allows the crediting of site systems, structures, and components (SSCs), as well as the physical security program in the prevention of "significant release of radionuclides from any source," and paves the way for implementing security by design (SeBD) and safety and security by design (SSeBD) within advanced reactor designs.

The concept of SeBD is to incorporate physical security concepts early into the design of a facility in order to prevent costly changes to the facility design in further stages of development or retrofits after the facility is built to meet physical security goals. The concept of SSeBD is to incorporate design changes for the safety system performance of a facility so that the facility is more resistant to an adversary attack, thus reducing vulnerabilities or providing additional SSC capabilities to prevent significant offsite releases from any radionuclide source. SeBD and SSeBD both have the goal of providing greater efficiencies in implementing security and providing more resilience to an adversary attack.

Implementing SseBD within a performance-based physical security regulatory framework requires new methods and tools to analyze and demonstrate facility and security program capabilities. This work seeks to develop a dynamic risk analysis tool that can model physical security events, plant SSC performance, and plant response up to offsite release to demonstrate site SseBD concepts for implementing an effective protective strategy for an advanced nuclear power plant.

1.1. DYNAMIC PHYSICAL SECURITY ANALYSIS IN LWRS

This work leverages ongoing research on the physical security of current power reactors in the Light Water Reactor Sustainability (LWRS) pathway. This section provides a brief description of that work.

The physical security program within the LWRS pathway aims to optimize the security posture of nuclear power plants with regards to protection performance and costs. This optimization is done by exploring research in modeling and simulation, the application of advanced sensors, and the deployment of

advanced weapons. Modeling and simulation are used to evaluate the margin inherent in many security postures and identify ways to maintain overall security effectiveness while lowering costs. Two areas identified for evaluation are taking credit for diverse and flexible mitigation capability (FLEX) equipment and actions taken by operators to minimize the possibility of reactor damage during an attack scenario. [1] While FLEX equipment was installed to support a plant's response to natural hazards, such as flooding or earthquakes, this equipment could also be used to provide reactor cooling in response to equipment damage caused by an attack on the plant. Likewise, there are certain actions plant operators will take when an attack occurs to minimize the likelihood and magnitude of an offsite release. It will take modeling and simulating the reactor core and systems to evaluate the effect these operator actions may have on increasing the coping time and minimizing the release from the reactor. This more inclusive process for physical security analysis is named Modeling and Analysis for Safety Security using Dynamic EMRALD (Event Modeling Risk Assessment using Linked Diagrams) Framework (MASS-DEF).

The MASS-DEF framework is illustrated in **Figure 1**. [1] A specific site has a list of attack scenarios and protection strategies to be analyzed. These are input data to design the force-on-force (FoF) model using available FoF tools such as SCRIBE3D, AVERT, Simajin, etc. Meanwhile, preventive safety actions are modeled in EMRALD. EMRALD then communicates with the FoF software, either in a unidirectional or bidirectional manner. A unidirectional coupling involves EMRALD adjusting and/or perturbing the input file of FoF, instructing the FoF tool to execute simulations, and extracting parameters of interest from the FoF outputs as initial conditions to the preventive safety actions to take. Meanwhile, a bidirectional coupling allows the software tools to communicate with each other and perform interactive simulations. For example, EMRALD may ask the FoF tool to simulate an attack and notify EMRALD when an intrusion is detected and confirmed, upon which EMRALD simulates a sequence of operator actions to shut down the reactor and fill the steam generator with cold water and notifies the FoF when it is done. This action provides a cooling margin if the reactor is sabotaged, which then extends the time window available to remove reactor residual heat and bring it to a safe stable state.

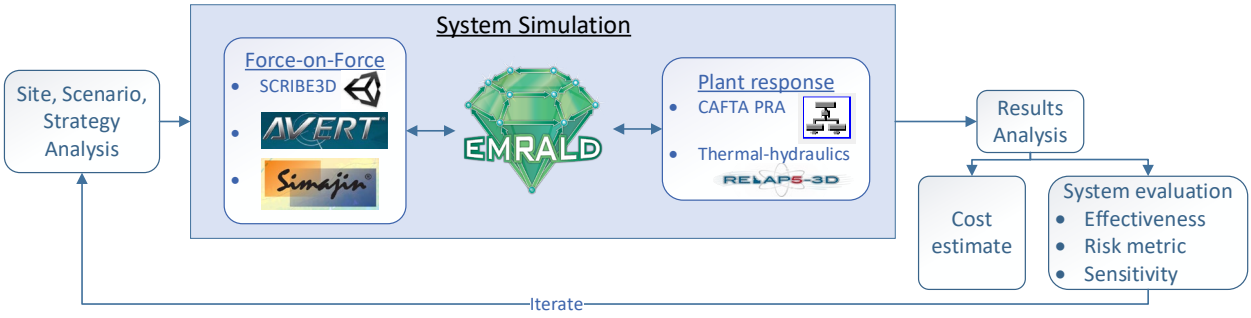


Figure 1. MASS-DEF Framework.

1.2. TECHNOLOGY GAP TO ADDRESS A/SMR NEEDS

Physical protection measures in light-water reactors (LWRs) are retrofits to existing nuclear power plants. Therefore, the facility layout, boundaries, and physical features are already in place that they can be modeled digitally in any FoF simulation tool. Similarly, the plant is already operational so that its process responses are known and can be modeled in safety analysis tools. An analysis of security-induced perturbations to the plant can therefore be conducted by integrating the FoF tool, operator actions, and plant responses altogether. However, this luxury is not available when dealing with advanced reactors

since none of them are deployed yet. There are no existing publicly available facility layouts to model in FoF. It is also costly to build three-dimensional (3D) computer-aided design (CAD) FoF models from conceptual designs, run security simulations, and iteratively modify the FoF models. It is desirable to have a tool that allows the physical security analysis with a simplistic model that is easy to build and modify. This capability is especially preferable when analysts do not have a concrete blueprint of the facility to analyze and instead only have a rough sketch or a conceptual design.

There is a need for a methodology and/or tool with the capability to analyze the sufficiency of A/SMR physical protection systems. This method or tool should allow for analysis flexibility without relying on rigid CAD models. Traditionally, this is done manually by using tabletop exercises. However, this exercise has its limitations. It is tedious for dynamic scenarios, time-consuming, and resource-intensive. It is preferable to utilize a digital tool capable of simulating dynamic scenarios well, documenting the results, and allowing easy and rapid modifications to the protection system. EMRALD as a dynamic risk assessment tool has been used for this purpose in the MASS-DEF framework. However, it lacks the capability to rapidly modify physical security design of A/SMRs. For that reason, this project aims to add such a capability to EMRALD.

Furthermore, the Physical Security LWRS work is aimed at integrating security and safety actions up to the point of core damage event. However, the framework for A/SMR extends beyond core damage, particularly for advanced non-LWR plants. The industry standard for these plants is the ASME/ANS RA-S-1.4-2021 [2] which is proposed for endorsement by the NRC through the trial regulatory guide (RG) 1.247 titled *Acceptability of Probabilistic Risk Assessment Results for Advanced Non-Light Water Reactor Risk-Informed Activities.* [3] In this framework, the safety analysis does not stop at core damage as the conventional probabilistic risk assessment (PRA) method. Rather, it performs a thorough analysis to measure the radiological consequence as its risk metric, which can be measured using the total effective dose equivalent for workers and the public. [4] For that reason, many risk analysts and researchers have expressed a desire for a risk analysis tool that can map events into the frequency-consequence (F-C) curve as shown in **Figure 2**. [5]

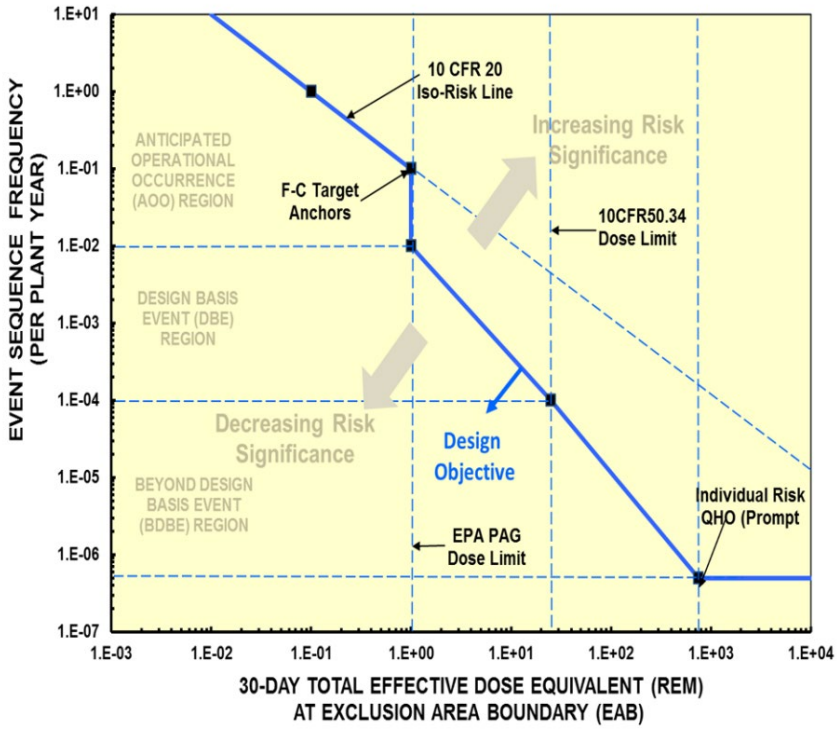


Figure 2. Frequency-consequence curve. [5]

2. METHODOLOGY AND TOOL DEVELOPMENT

The proposed methodology is based on a dynamic risk assessment approach. It is based on the consideration that there are uncertainties in the execution of an attack plan that may affect the next steps of attack or intervention actions. [6] In that sense, the dynamic approach relaxes the conservative assumption utilized in static methodologies that adversaries almost always successfully execute their attack plans. The dynamic approach then introduces more realism to the physical protection program. It has been successfully used in a previous preliminary study to model dynamic uncertainties in attacks to a hypothetical advanced reactor. [7]

2.1. METHODOLOGY

The conventional physical security analysis in LWRs focuses on the capabilities of the physical protection elements to prevent adversaries from reaching their sabotage targets and leaving the facility with their theft targets. In that regard, the success criteria for the adversaries are conservatively assumed as an adversary reaching the target or an adversary leaving the site perimeter with a theft target. However, the objective of a physical protection program is to prevent an unreasonable risk to the public health and safety. Therefore, the objective is tied to the consequence of the activity. There is a time window from when adversaries reach the target to when radioactive material is released to the environment creating an unreasonable risk to public health and safety. Within this time window, there are preventive and mitigative safety actions that can be taken to prevent or reduce the radiological consequences. For those reasons, this work proposes a methodology that integrates security with safety to measure the likelihood of an unreasonable radiological risk to the public caused by sabotage attacks.

Current physical protection requirements focus on preventing radiological sabotage through the prevention of core damage or spent fuel sabotage. The drafting of alternative regulatory requirements opens the pathway to the use of new methodologies and tools to demonstrate compliance with the protection against radiological sabotage though a demonstration of security and safety elements to keep offsite releases from anticipated security events below dose reference values. The current NRC rulemakings proposed guidance references for those values found within the regulations for the safe operation of nuclear power plants from 10 CFR 50.34(a)(1)(ii)(D)(1) and (2) and 52.79(a)(1)(vi)(A) and (B) are as follows:

- (1) An individual located at any point on the boundary of the exclusion area for any 2 hour period following the onset of the postulated fission product release, would not receive a radiation dose in excess of 25 rem total effective dose equivalent (TEDE).
- (2) An individual located at any point on the outer boundary of the low population zone, who is exposed to the radioactive cloud resulting from the postulated fission product release (during the entire period of its passage) would not receive a radiation dose in excess of 25 rem total effective dose equivalent (TEDE).

Reactor applicants and licensees must demonstrate protection against radiological sabotage from a group of adversaries with the characteristics and capabilities detailed in the NRC-defined design basis threat (DBT). The DBT details the abilities of an adversary force to include their sabotage knowledge and capabilities.

The plant must defend against the adversary force for a defined time period, after which offsite response forces (local, state, and federal law enforcement and emergency response forces) would be anticipated to arrive and secure the facility. This duration was established by the NRC for U.S. light-water commercial reactor facilities (reasonable assurance of protection time [RAPT]), and it is anticipated that NRC will establish a framework to determine a security bounding time (SBT) for reactor sites that do not have the full security program as currently required under 10 CFR 73.55.

The proposed methodology combines the effect of safety and security commonly known as 2S, in an achievability-consequence plot as shown in **Figure 3**. **Figure 3** demonstrates the categorization of SSC combinations for protective system consideration and shows how through the implementation of SeBD and SSeBD can remove SSC combinations from consideration within the protective strategy by extending the timeframe for failures that lead to offsite dose consequences beyond the reference values, hardening targets within the facility, or reducing the dose attributed to the loss of a target below the dose reference values through preventive or mitigative SSCs and or actions.

Those SSC combinations that lead to an offsite release above either dose reference value and are within the capability of the adversary to defeat, deny, or compromise must be accounted for in the protective strategy. These SSCs fall within the upper-right blue region labeled “Achievable Targets with consequences exceeding dose reference values.” The SSCs that fall within this region are of the highest importance to protect to prevent radiological sabotage.

Those SSC combinations that lead to a release but would not exceed the offsite dose reference values fall within the upper left region labeled “Achievable Targets with consequences below dose reference values.” The dose reference values are shown above and are indicated within **Figure 3** as dashed green lines. The position of these values may not be representative of actual events, where either 2-hour TEDE exposures or total TEDE exposures may be higher or lower than the other. SSC combinations may move into this space from the blue region through the reduction in offsite dose consequences by prevention or mitigation by implementation of SSeBD. SSeBD capitalizes on the performance of plant safety systems in response to a security event to prevent or mitigate consequences. Additions or changes to the plant safety systems or operating procedures can increase their resilience to adversary attack by increasing task times (increasing the complexity of a task), increasing the defense in depth associated with a system, and reducing the source term associated with a target.

Those SSC combinations that lead to a release but are not within the capability of the adversary to defeat, deny, or compromise fall within the lower region labeled “Non-achievable Targets.” SSC combinations may move into this space from the blue region through the implementation of SSeBD by extending the timeline associated with a release, such that non-reversible offsite release would not occur before an established SBT, or through the implementation of SeBD. SeBD changes to a plant design can relocate or reinforce SSCs against adversary attack such that they exceed the capabilities of the adversary or time for the attack are beyond the established SBT.

The methodology demonstrated in this report is meant to assist reactor vendors in the implementation of SeBD and SSeBD in the design of their reactor facilities and support the licensing of a performance-based security program.

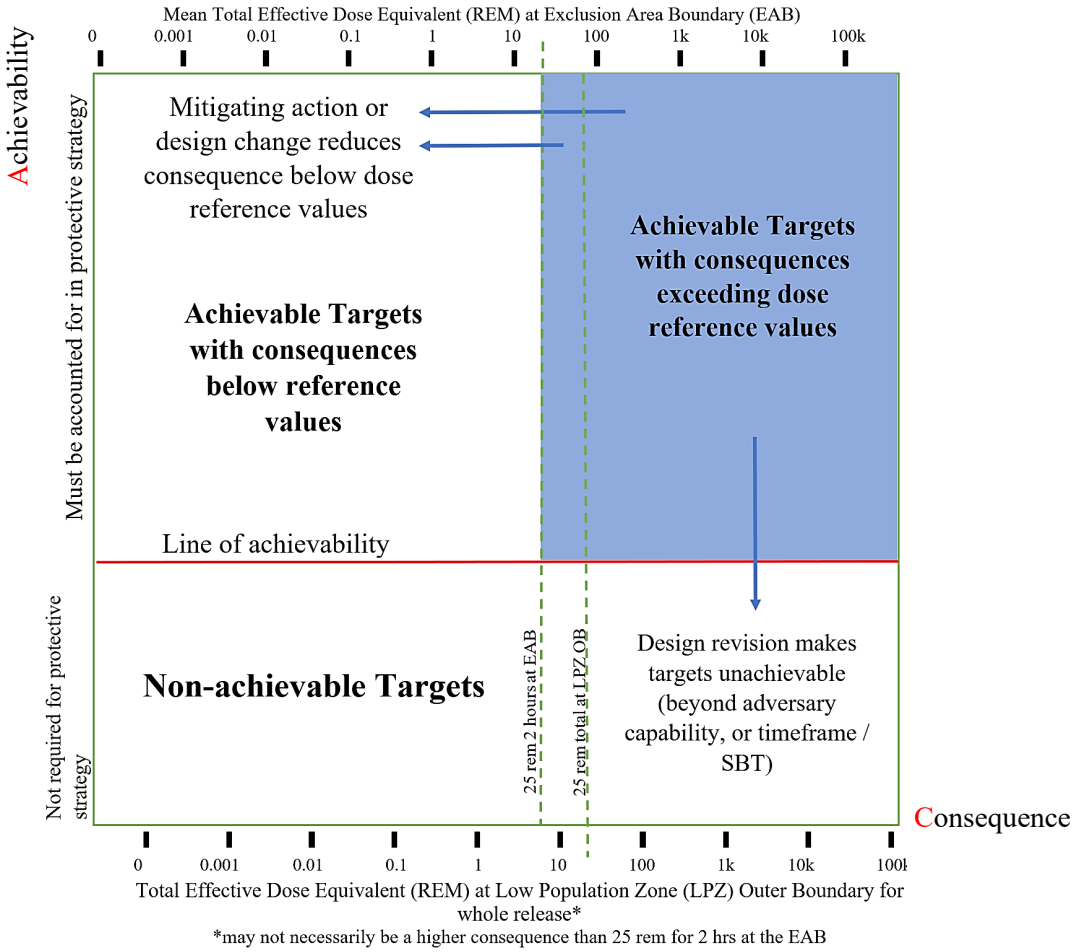


Figure 3. The A/C plot.

2.2. TOOL DEVELOPMENT

2.2.1. EMRALD

Idaho National Laboratory (INL) developed EMRALD [8], a dynamic PRA tool. This tool has two main components: (1) the model development module, which is hosted online at https://emraldapp.inl.gov and (2) the model solver module that can be downloaded from the EMRALD site. EMRALD plays a key role in the MASS-DEF framework since it can couple the FoF tool and the MAAP thermal hydraulic safety analysis model with dynamic operator actions modeled in EMRALD. Depending on the outcome of FoF simulations, EMRALD can simulate preventive safety actions and their uncertainties, feed the outcome of those actions to MAAP, and fetch MAAP's results.

EMRALD also contains traditional PRA elements such as fault trees, failure rates, and failure probabilities. Therefore, EMRALD can estimate random component failures, and even human failures, during the preventive safety actions. This capability allows the simulation to be more realistic.

This work extends EMRALD capabilities to support a "sandbox" feature that allows analysts to easily model attack scenarios, drag-and-drop pre-build templates of security and safety features, run analysis, and redesign the facility in an iterative manner. To allow this, we built a template-creation feature in

EMRALD. This recent update enables one to create and manage diagram templates and the ability to filter actions, states, and events for a specified diagram. Diagram templates make it easy to add similar diagrams to multiple models. The next subsection discusses the templates developed for A/SMR physical security analysis.

2.2.2. EMRALD Templates

When a template is created or imported, it is saved to the user's local web storage, which allows EMRALD to access the template across different browser tabs and windows. These templates can be saved to a file, shared with other users, and reused in other models. We have built generic templates that are publicly accessible at EMRALD's GitHub site: https://github.com/idaholab/EMRALD. The EMRALD app is also publicly accessible at https://emraldapp.inl.gov. The template is located at: https://github.com/idaholab/EMRALD/tree/main/Emrald_Site/Templates. Some of the templates already developed are listed in **Figure 4**. They include templates for physical barriers such as doors, fences, and padlocks, templates for intrusion sensors, such as infrared sensor, taut wire sensor and seismic cable, templates for physical areas, and templates for switching reactor operational modes and the mobilization of armed responders. The templates use generic data from various open sources, such as the hypothetical Lone Pine nuclear facility used in physical security trainings and workshops [9], [10], and generic facility data in PathTrace software that are used for training and demonstration purposes. [11] **These data do not represent any actual nuclear plant.**

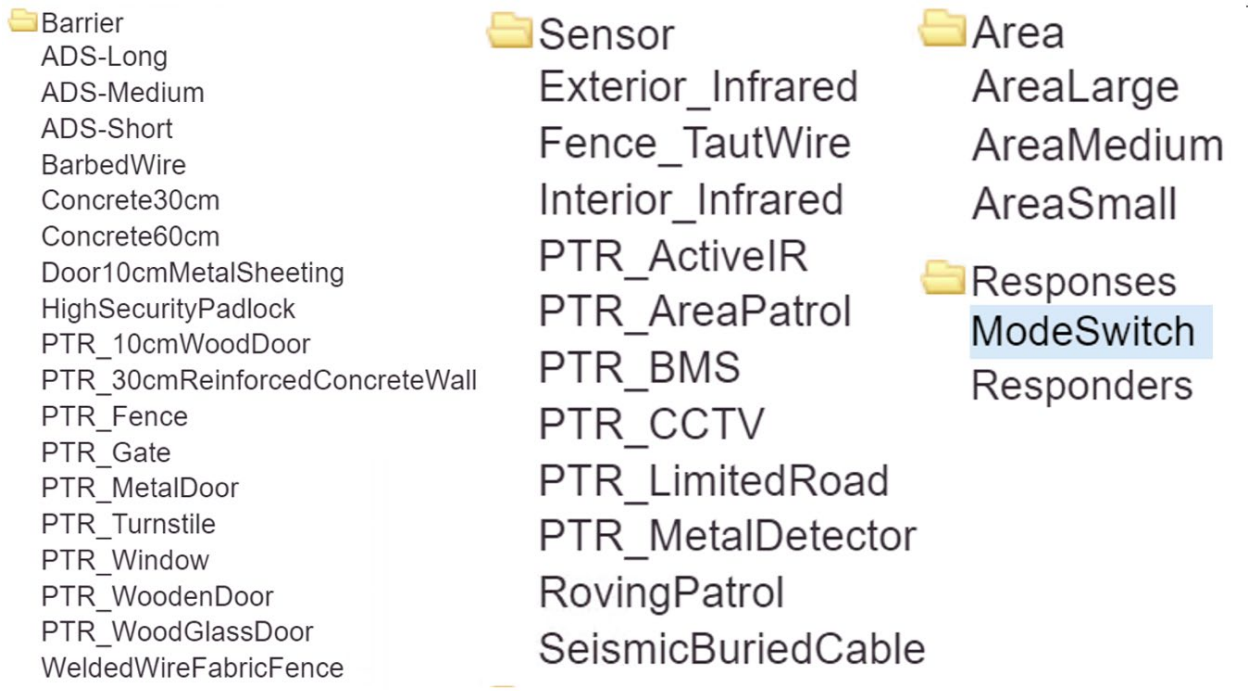


Figure 4. Pre-built template diagrams for A/SMR physical security.

Each template is modeled in its own EMRALD diagram. An example is shown in **Figure 5**. It is a barrier that activates upon detection (active delay system). It starts from the ADS-Medium-Operational state and samples the intrusion detection with a certain probability in the ADS_Medium_Activation action. If it senses a detection, it transitions to the ADS_Medium_Active state with a certain time distribution of delay. Otherwise, it transitions to the ADS_Medium_Inactive state with a faster delay distribution. These time variables depend on the breaching tools adversaries use. They are then evaluated in the ADS_Medium_Consequence state. After the delay time has elapsed, EMRALD transitions to the ADS_Medium_Breached implying that the active barrier has been defeated. Appendix A provides a user manual on creating and editing EMRALD templates.

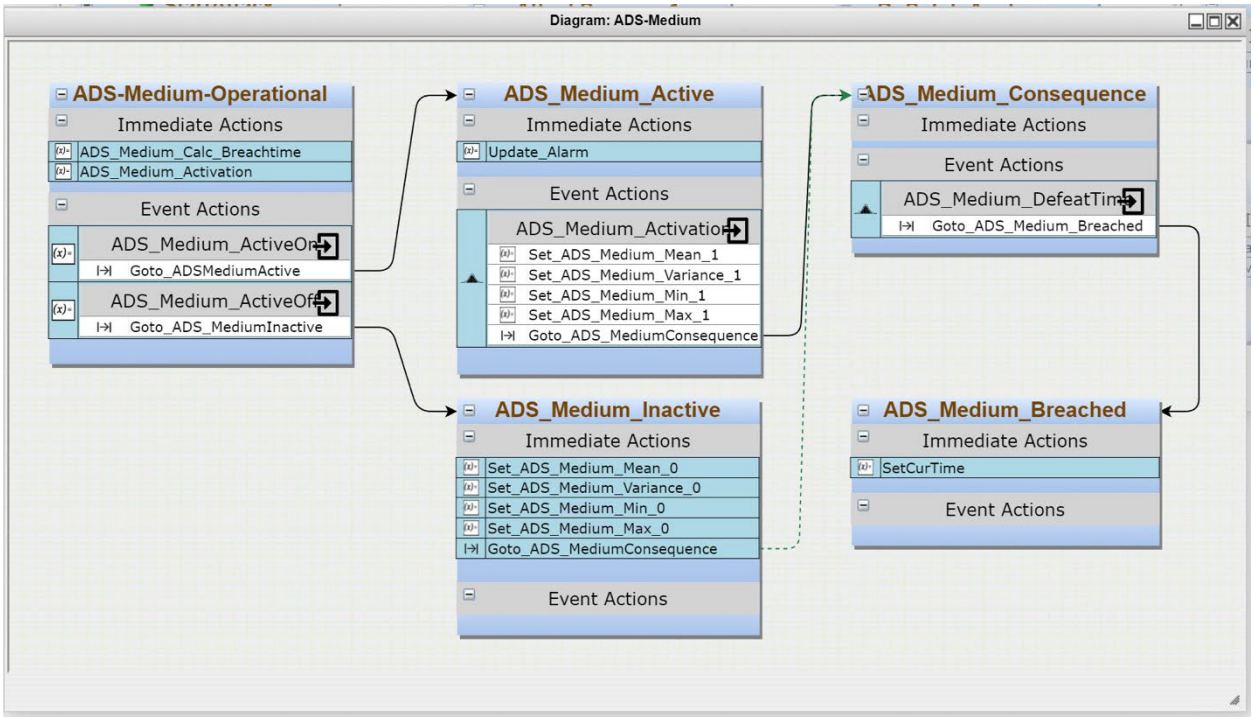


Figure 5. Diagram for medium-level active delay system (ADS).

2.3. GENERIC SFR MODEL

A generic sodium-cooled fast reactor (SFR) facility is developed in this work to be used as in case studies to demonstrate the methodology. More case studies using the generic facilities are planned in fiscal year 2024. **This generic facility does not contain proprietary or safeguards information, and it does not represent any actual operating or planned nuclear reactors**. The purpose for this generic facility is to identify possible targets in an SFR reactor in a generic manner, which then inform the various available safety actions to prevent radiological release to the environment, as well as a range of source terms released. While this generic data and information are not exact, they are useful to inform the flow process and required data to A/SMR vendors who want to follow the methodology described in this report.

The generic SFR layout is illustrated in **Figure 6**. It includes the reactor core located in the reactor hall building, and support systems located adjacent to it (i.e., the sodium loops to extract heat from the core, the sodium purification system to filter out impurities in the sodium coolant, and the cover gas

purification system to maintain the inertness of argon gas in the core and prevent chemical reactions with sodium). The steam generator and turbine function similar to any other power-generating station. A notable difference here is the presence of passive heat removal towers. Many A/SMR designs utilize passive cooling systems. This system is designed to remove decay heat from the core if the active system is unavailable. Typically, it has redundant passive trains for a higher reliability. The facility is assumed to have an onsite fuel cycle facility to wash off sodium from and to decontaminate spent fuels assemblies and to store them temporarily.

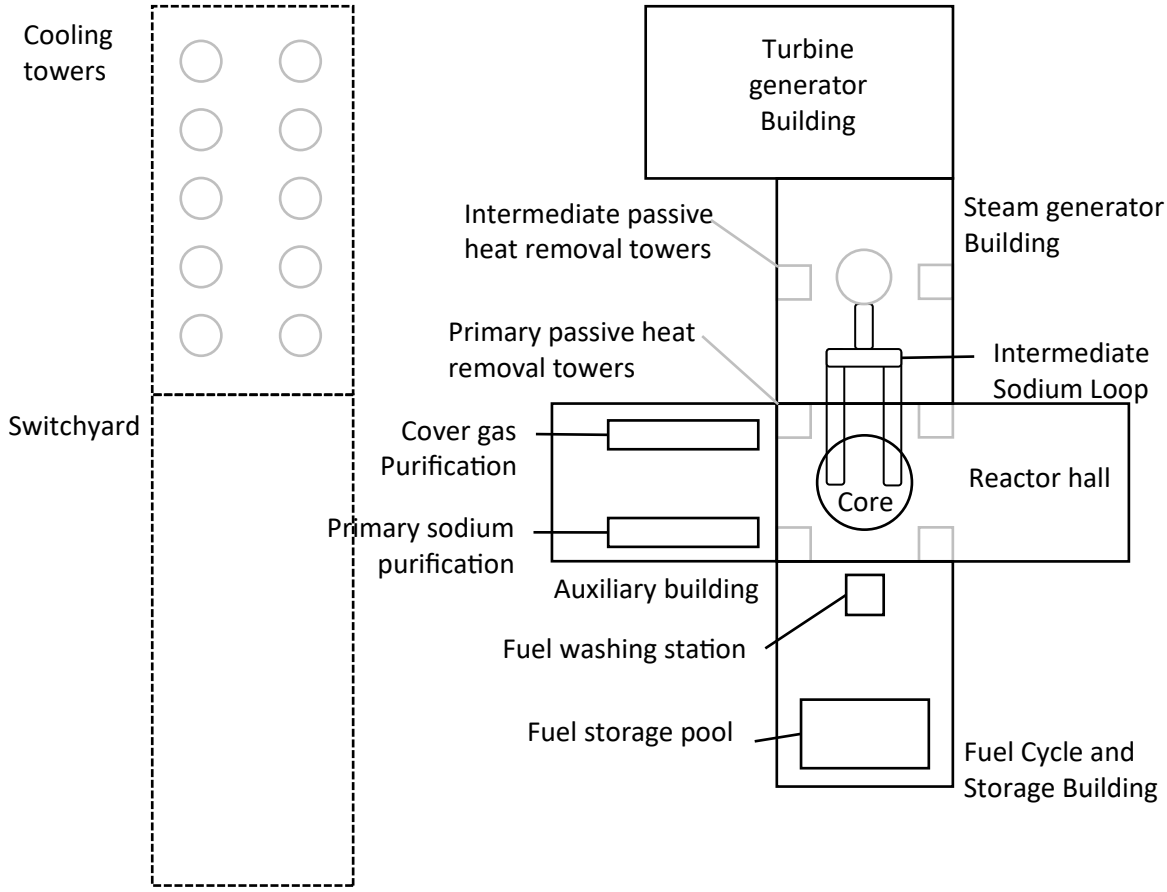


Figure 6. Generic SFR facility layout.

Several radionuclide sources are identified for the reference SFR facility, as listed in **Table 1.** The table also identifies various safety actions that can be taken to prevent the release of these radionuclide sources during various stages of an attack. These preventive actions are quite generic and may be applicable to a wide range of A/SMR designs. They are open to further refinements based on reactor-specific capabilities. An example here includes shifting the reactor mode to a hot standby, closing isolation valves in the sodium purification and cover gas purification systems, ceasing any fuel washing and fuel movement activities when an intrusion is detected. If the detection is confirmed, the operator can shut down the reactor and transition to passive heat removal systems. Therefore, heat removal does not depend on the active balance-of-plant, which is a possible adversary target. If the adversaries

complete their attack, operator can activate an offsite control room and start the emergency response protocol to protect onsite workers and evacuate nearby populations.

Table 1. Safety actions in response to adversary attacks in the generic SFR facility

Stages of Intrusion	Sources of Radioactivity					
	Core	**Primary Sodium Purification System**	**Cover Gas Purification System**	**Fuel Washing Station**	**Fuel Storage Facility**	**Intermediate Sodium Loop**
Detection[1]	Power runback (or hot standby)	Closure of isolation valves Shutdown of system	Closure of isolation valves Shutdown of system	Cessation of any fuel washing activities	Cessation of any fuel movement activities	(see core)
Confirmation[2]	Reactor shutdown Containment isolation Transition to passive heat removal system (shutdown of BOP)	Cell HVAC isolation Cold trap cooling switched to emergency power	Cell HVAC isolation Decay bed cooling switched to emergency power	Washing station HVAC isolation	Facility HVAC isolation Spent fuel pool cooling power switched to emergency power (or passive cooling)	Shutdown of intermediate sodium pumps
Sabotage[3]	Remote control room activation Emergency response implementation[4]	Emergency response implementation[4]	Emergency response implementation[4]	Activate emergency mercury flood system Emergency response implementation[4]	Emergency response implementation[4]	Activate intermediate sodium loop drain Emergency response implementation[4]

[1] *Assumption that detection has certain level of confidence (i.e., multi-sensors).*

[2] *Confirmation of site intrusion.*

[3] *If sabotage is related to the designated radionuclide source.*

[4] *As described in the plant's emergency plan. Includes protective actions and other emergency response measures for workers and public.*

2.4. GENERIC HTGR MODEL

A generic pebble-bed high-temperature gas-cooled reactor (HTGR) facility is discussed in this section. Similar to the SFR facility, **this HTGR facility does not represent any actual operating or planned nuclear reactor**.

The reference HTGR facility layout is shown in **Figure 7** as adapted from a previous open publication. [12] It consists of three reactor units. Given the intrinsic safety feature of TRi-structural ISOtropic (TRISO) particle fuels, each reactor unit is assumed to use multiple barriers to retain radionuclide release, including fuel particle kernel, fuel particle coatings, fuel-element graphite matrix, helium pressure boundary, and reactor building using a vented confinement system. These barriers jointly form a functional containment system. The three reactor units are assumed to have a shared interim dry-cask spent fuel storage building and dedicated intermediate spent fuel storage buildings. Each reactor has its own reactor building and turbine building. The reactor cores are located at the basement level of each reactor building. Each reactor has its dedicated fuel handling system to circulate fuel pebbles through the reactor core and helium purification system to purify the helium gas, both located within reactor building. The layout also includes three cooling towers dedicated to each reactor unit and a shared switchyard.

Several sources of radioactivity are identified for the reference HTGR facility, including reactor core, fuel handling system, spent fuel storage, and helium purification system. In an HTGR, radioactivity can come from two types of sources, depending on if a source contains fuel pebbles or not. Sources containing fuel pebbles include reactor core, fuel handling system, and spent fuel storage system, are all visualized in **Figure 7** and listed in **Table 2**. Sources not containing fuel pebbles include radioactive waste systems and their upstream systems generating radioactive wastes. For this study, radioactive waste systems are not considered. Helium purification system is kept in the study as a representative upstream system generating gaseous and liquid radioactive wastes. **Table 2** identifies safety actions that can be taken to prevent releases from each identified radioactivity source during various stages of an attack.

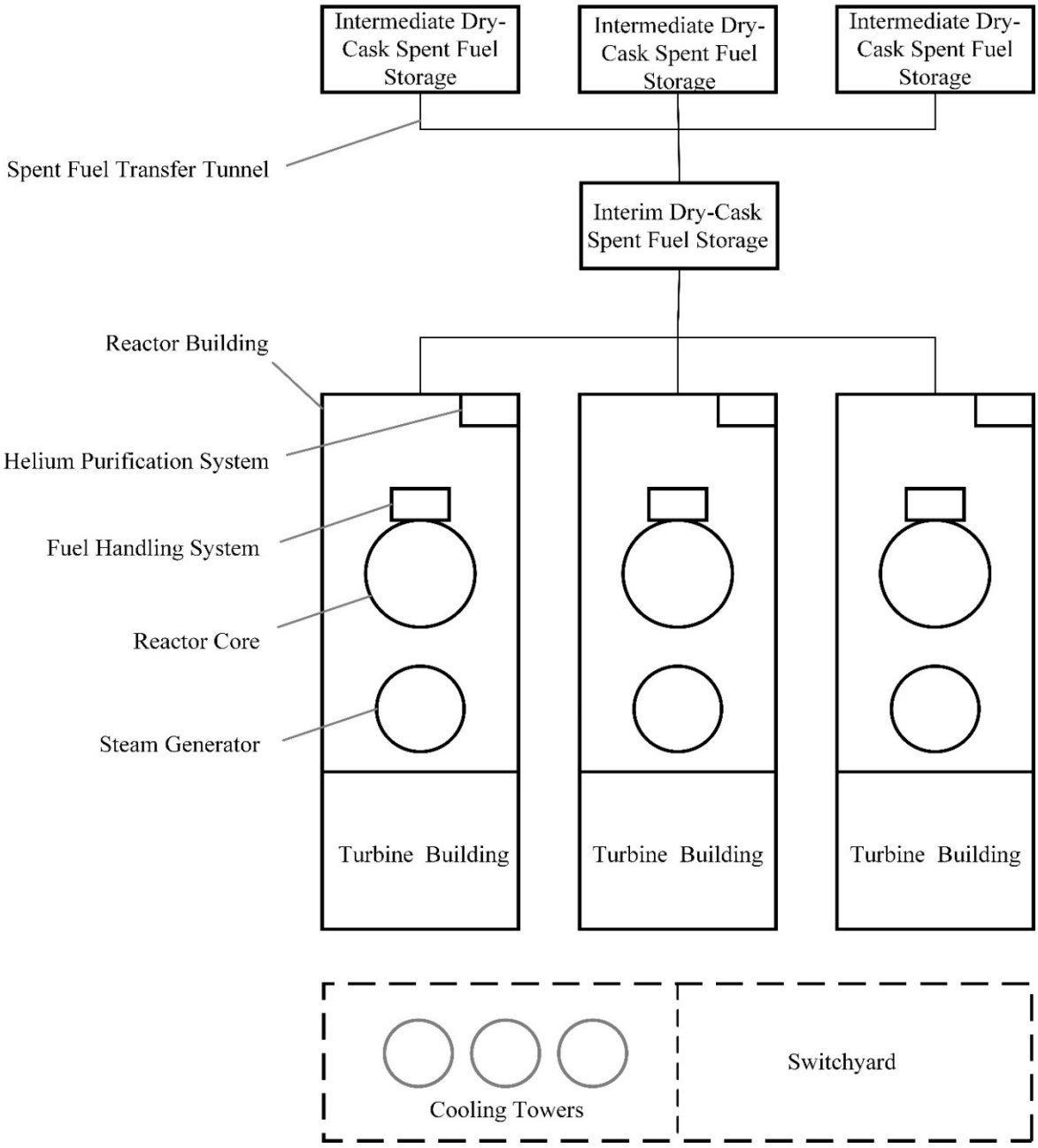


Figure 7. Generic HTGR facility layout.

Table 2. Safety actions in response to adversary attacks in the generic HTGR facility.

Stages of Intrusion	Sources of Radioactivity			
	Core	**Helium Purification System**	**Fuel Storage Facilities[6]**	**Fuel Handling System**
Detection[1]	Power runback (or hot standby)	Closure of isolation valves Shutdown of system	Cessation of any fuel movement activities	Closure of isolation valves Cessation of any pebble handling activities
Confirmation[2]	Reactor shutdown Reactor building isolation Transition to passive heat removal system[4] (shutdown of BOP)	System HVAC isolation System cooling switched to emergency power (or passive cooing)	Facility HVAC isolation	System cooling switched to emergency power (or passive cooing)
Sabotage[3]	Remote control room activation Emergency response implementation[5]	Emergency response implementation[5]	Emergency response implementation[5]	Emergency response implementation[5]

[1] *Assumption that detection has certain level of confidence (i.e., multi-sensors).*

[2] *Confirmation of site intrusion.*

[3] *If sabotage is related to the designated radionuclide source.*

[4] *Assumption that HTGR uses active heat removal systems during normal operation such as using motor-driven helium circulators.*

[5] *As described in the plant's emergency plan. Includes protective actions and other emergency response measures for workers and public.*

[6] *Assumption that HTGR spent fuel pebbles are stored in closed dry casks using passive cooling. No cooling power switch is needed under intrusion.*

3. CASE STUDY

A case study is developed to illustrate the use of the proposed methodology. Consider the reference SFR facility layout in **Figure 8** as a preliminary layout of an SFR reactor, without any security features (i.e., fences, sensors, locks, and video cameras) in place yet. Suppose adversaries plan to sabotage a target set consisting of two systems, identified as Target 1 and Target 2 in the facility layout, where these targets are hypothetical for illustration purposes only. EMRALD can then be used to first model the attack scenario and drag-drop security and safety features from the templates into the model, such as the sensors and barriers shown in **Figure 8**. The analyst can then compare the likelihood and consequence of the sabotage attack before and after the facility is improved.

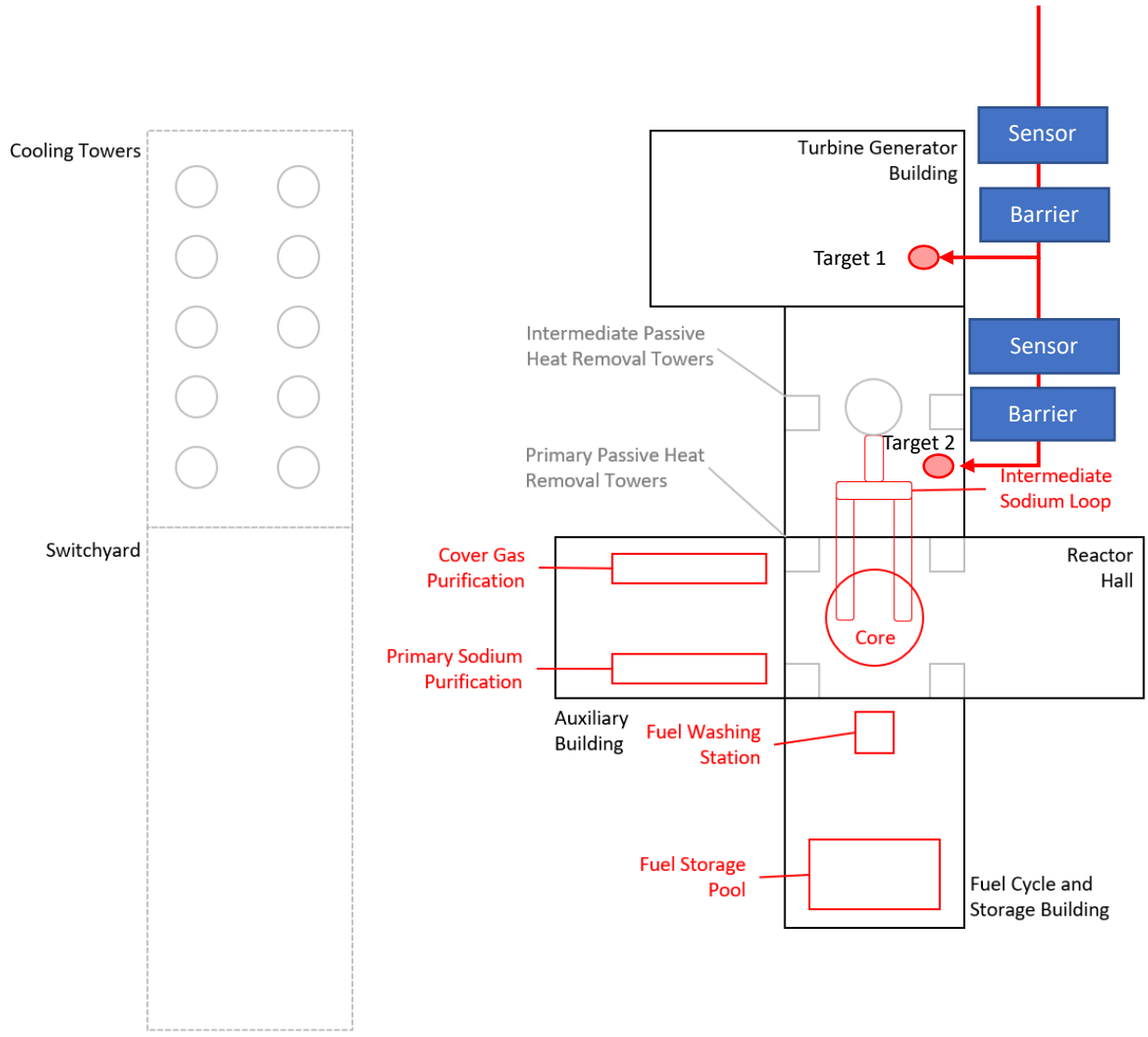


Figure 8. Example attack path on the reference SFR facility.

The flowchart of attack and response plans is shown in **Figure 9**. As the adversaries carry out their attack, intrusion detection sensors may detect the intrusion. Upon detection, the reactor switches to a hot

standby mode following the safety protocols outlined in **Table 1**. After confirming that the alarm detects an actual intrusion (not a false alarm), the reactor is shut down and security responders are mobilized to intercept the intrusion. The reactor then activates post-shutdown actions such as transitioning to a passive residual decay heat removal system and/or activating containment isolation (e.g., deploying foam) if required to reduce the radionuclide release fraction from the core. If adversaries manage to sabotage the target set, the emergency response protocol is implemented to evacuate nearby population before the reactor is damaged releasing radionuclides to the environment. Note that A/SMRs are expected to have a longer time margin until such an adverse outcome happens compared to LWRs.

Due to the timing competition between the sabotage attack and preventive safety responses, there may be different outcomes on the sabotage as identified in **Table 1**. The attack may be completed undetected such that the reactor was operating normally, or the attack may be completed when the reactor is at the hot-standby mode, or when the reactor is shut down, and/or when the emergency response procedure is completed. The state of the reactor when it is sabotaged affects the source term dispersion rate from the argon cover gas region and from the containment. Meanwhile, the state of emergency response affects the atmospheric dispersion parameter and breathing rate of the nearby population. Therefore, the dynamics of the attack scenario may result in different levels of radiological consequences. This is the phenomena the user needs to capture in EMRALD.

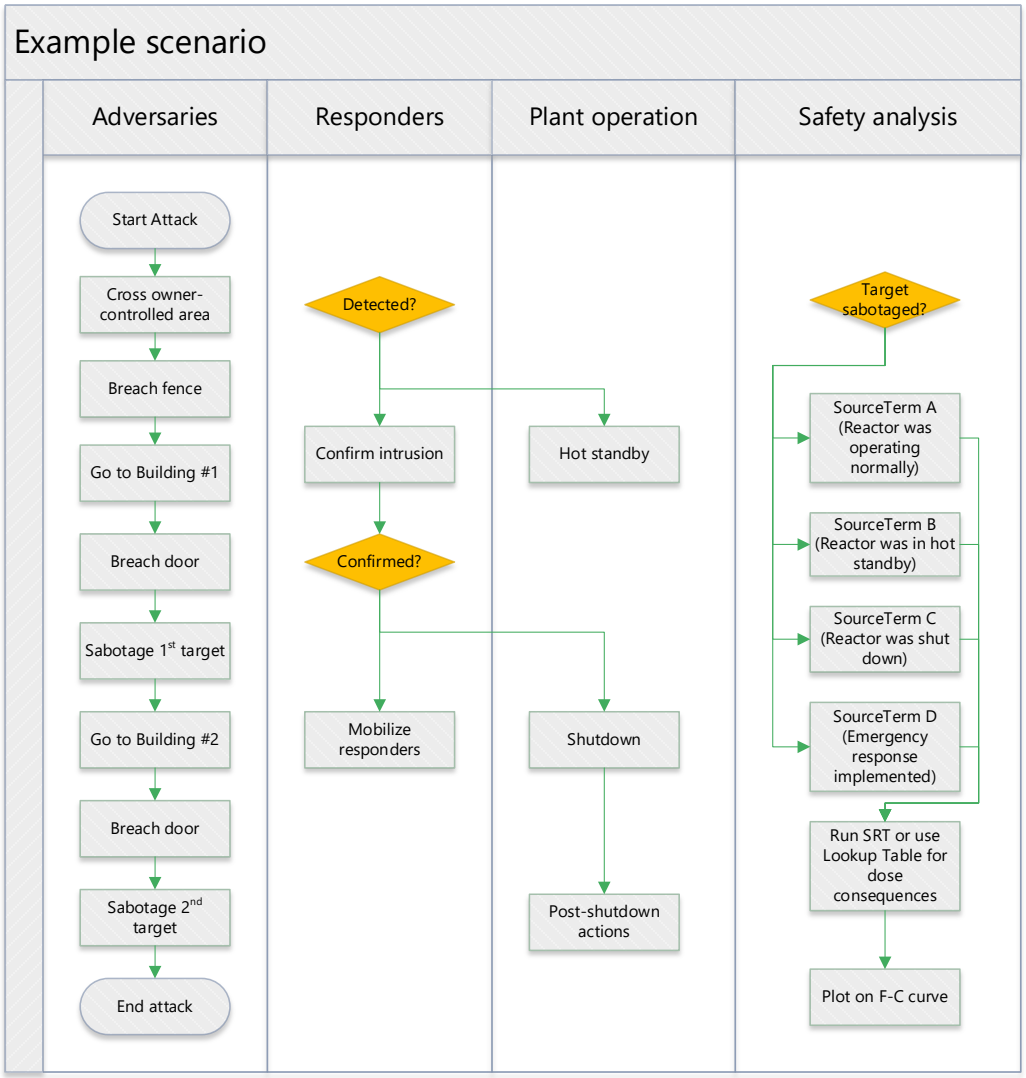


Figure 9. Attack scenario and response flowchart.

An EMRALD model of the attack and response scenarios is developed, starting from the main diagram shown in **Figure 10**. The simulation starts from the StartAttack state and executes two immediate actions modeled in other subdiagrams (i.e., the reactor operational state and the intrusion alarm monitoring system). There are three attack scenarios modeled in **Figure 10**, and the analyst can evaluate different scenarios by dragging the Goto_AttackScenarios arrow to any of the scenarios. The analyst can also evaluate all scenarios by linking the arrow to all three scenarios and assigning probability values to each arrow branch. This approach reflects the likelihood for each scenario when an attack happens, which may be inferred from the attractiveness of each target set and/or expert evaluation from subject-matter experts (SMEs).

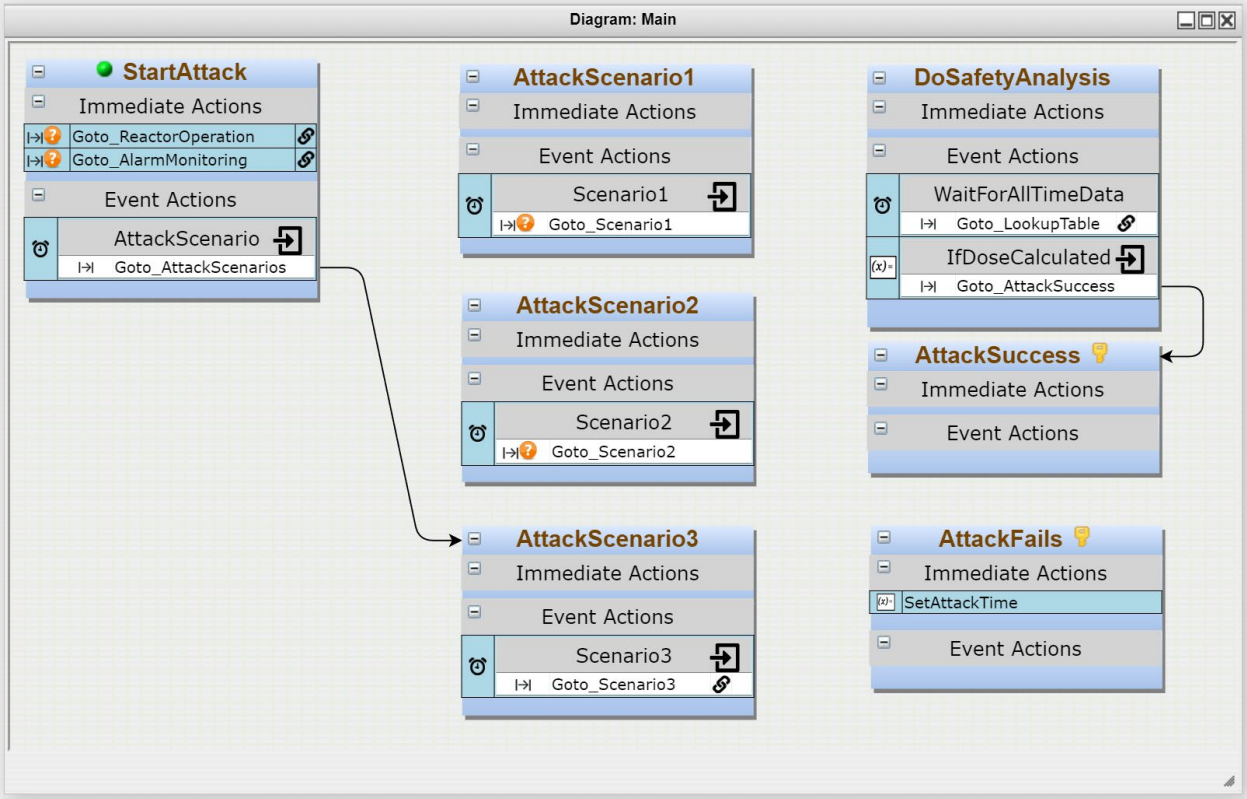


Figure 10. Main EMRALD diagram.

The attack scenario as shown in **Figure 9** is modeled in **Figure 11**. The left-hand-side of the figure shows the EMRALD templates user can attach to the attack scenario. For example, there are two transition actions at the fence state: Fence_Sensor and Fence_Barrier. The Fence_Sensor action can transition to any of the sensor templates, and the Fence_Barrier action can transition to any of the barrier templates. The analyst then drags and drops the state that represents the barrier being breached, into the StepFence_Done event as shown in **Figure 12**. Therefore, the event is active when the Fence_Barrier is breached after some delay time with a statistical time distribution. This case study considers two situations: when the facility is operated without sensors, barriers, and preventive safety actions versus when the facility is equipped with various security and safety features.

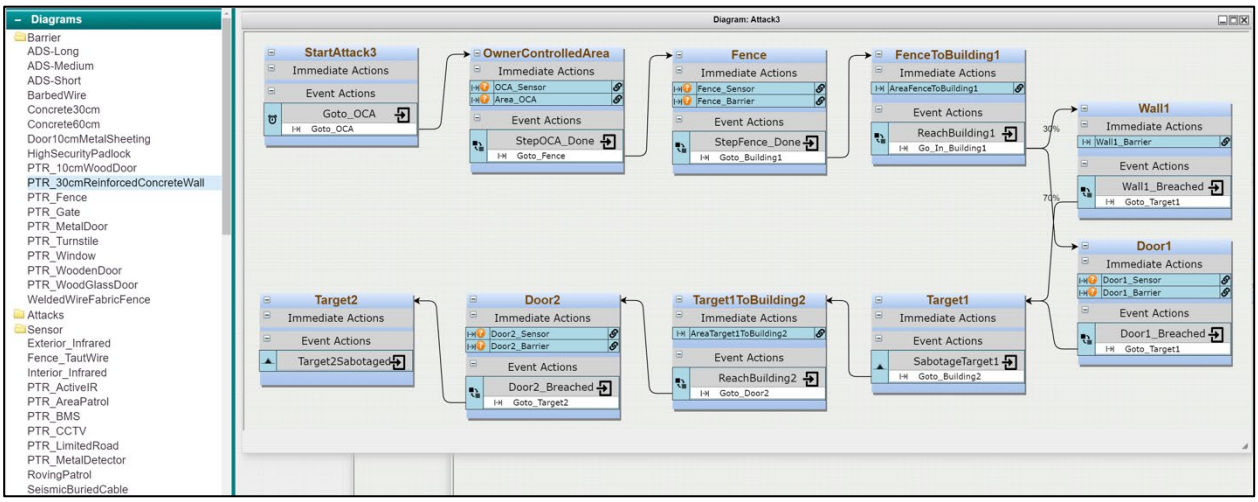


Figure 11. EMRALD subdiagram for the attack scenario.

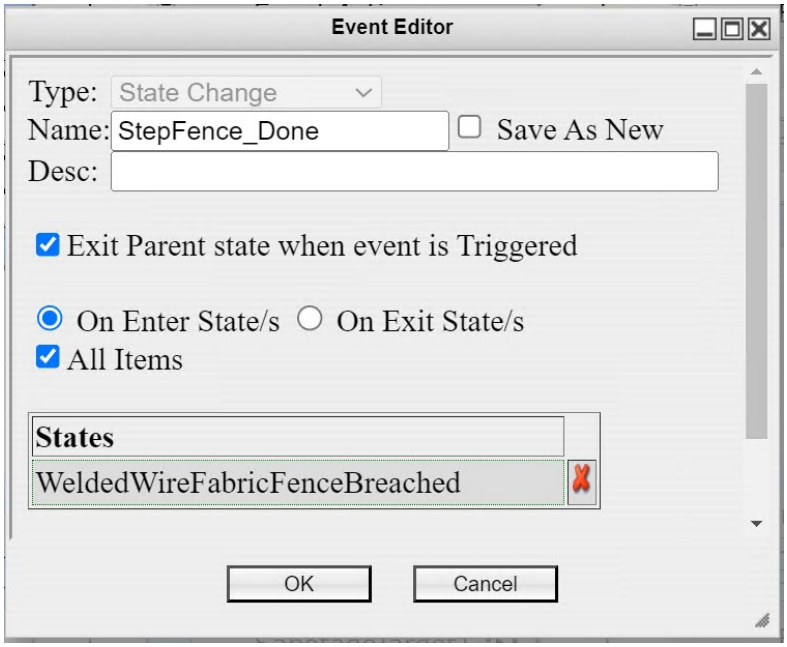


Figure 12. StepFence_Done event.

The preventive safety actions are modeled in **Figure 13**. It starts from the normal PowerOperation state. The AlarmTriggered event is active when the intrusion alarm is triggered, which transitions the reactor state to the HotStandby state. There are time distributions to accomplish the hot standby and confirm the intrusion, modeled in HotStandbyAchieved and AlarmConfirmation events, respectively. Once these two events are achieved, the IfAlarmConfirmationAndReactorState event becomes active and switches the reactor state to the Shutdown state. Similarly, there are time distributions to shut down the reactor and isolate the containment. After the two events are achieved, the IfCoreShutdown&ContIsolation event activates the EmergencyResponse state.

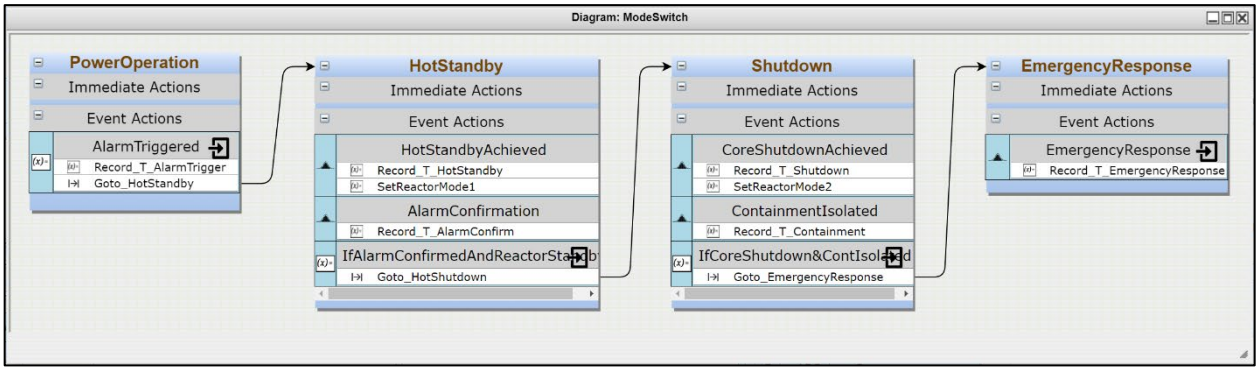


Figure 13. EMRALD subdiagram modeling preventive safety actions.

After the attack and response simulation is complete, EMRALD records the necessary data such as attack timing, response timing, reactor state, emergency response state, etc., to estimate the extent of radiological release to the environment, if any. This model is shown in **Figure 14**. EMRALD does this by comparing the data with a look-up table in LookupTable state or by running a safety analysis code for that particular scenario (in this case, running the Simplified Radionuclide Transport (SRT) code [13][14] in the RunSRT state and reading the results in the ReadSRT state). The SRT is a fast-running radiological release assessment code developed mostly for advanced reactors by Argonne National Laboratory, although EMRALD also supports other safety analysis codes such as MELCOR, MAAP, RELAP5, etc. The radiological release parameters in the SRT input file for this case study, and the RunSRT and ReadDose actions, are detailed in Appendix B: SRT Code Parameters. The LookupTable state is useful if we have run and saved various safety scenarios previously, such that those results can be reused without having to rerun safety analysis codes for each random iteration, which can be computationally expensive.

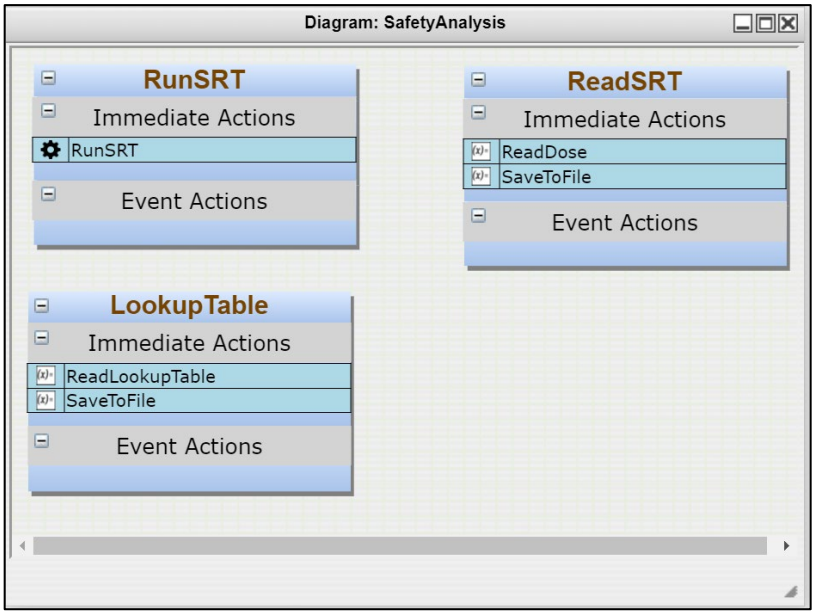


Figure 14. EMRALD subdiagram modeling safety analysis.

4. RESULTS AND DISCUSSIONS

Results for the initial model when the reference plant was not equipped with any security and preventive safety features are shown in **Figure 15**. Because there is nothing preventing adversaries from sabotaging the plant, they managed to execute the attack successfully all the time, in less than 15 minutes.

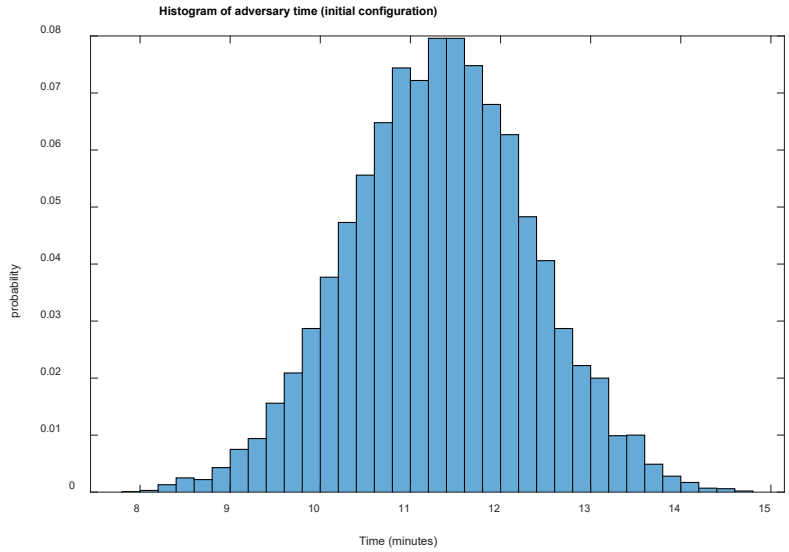


Figure 15. Initial distribution of attack time.

Results for the updated model when the reference plant was equipped with security and safety features are shown in **Figure 16**. The figure reveals more complex information, including the time distribution for adversaries, intrusion detection time, responders' arrival time, time when reactor is put on hot standby, on shutdown, and the emergency evacuation time. It shows that most of the time, the responders arrived and intercepted adversaries before they complete their sabotage (note: for this case study, it is assumed that a timely interception is sufficient to stop the attack). Only a small percentage of the cases shows that adversaries complete the sabotage before responders' arrival. The responder's mobilization time was arbitrarily assumed to be between 5 to 11 hours, which allows the facility to utilize offsite responders instead of onsite.

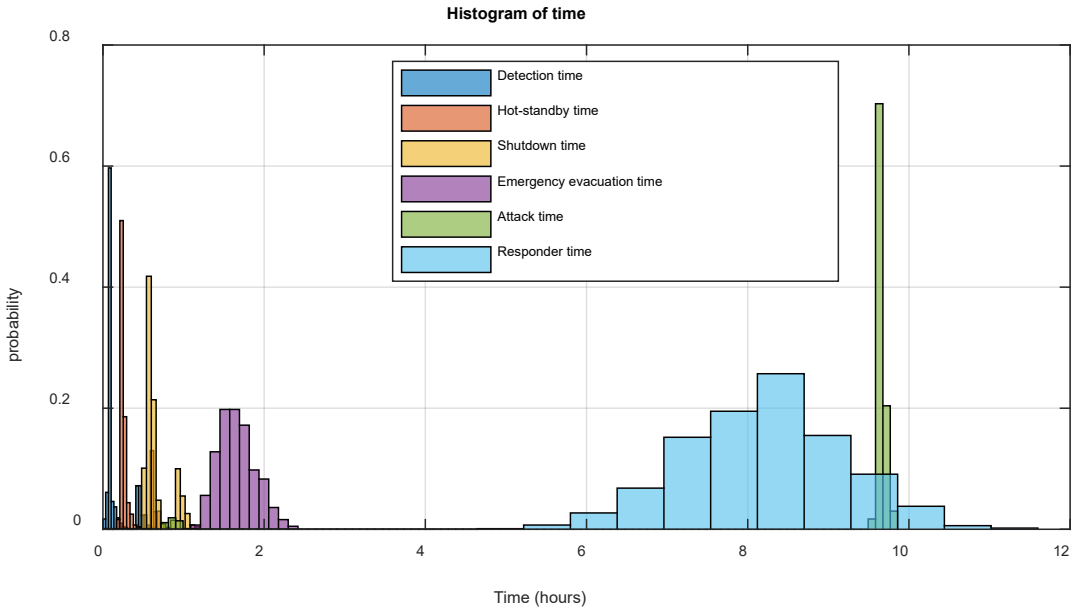


Figure 16. Time histogram after security modifications.

Following the classifications of attack outcome listed in **Table 1**, this case study also lists the attack outcomes in **Table 3**. This table is not automatically generated by EMRALD, but it was a result of post-processing the simulation data saved by EMRALD. The table shows the statistical variations of attack outcomes and the corresponding radiological consequences.

Table 3. Simulated outcomes of sabotage attack.

Attack Outcome		Reactor State	Probability*	Radiological Consequence (Rem)*
Attack fails due to timely interruption		Reactor full power	88.3%	≈ 0
Attack successful	Sabotage completed after preventive actions	Reactor is shut down, and nearby population is evacuated	7.2%	2E-2
	Sabotage completed before preventive actions	Reactor is shut down	3.6%	2.07
	Sabotage completed before detection was confirmed	Reactor hot standby	0.9%	7.1

	Sabotage goes undetected	Reactor full power	≈ 0%	17.83

** **Disclaimer**: The data used to generate these results are arbitrary; therefore, the **results do not reflect any actual nuclear plant**.*

The resulting probability of executing the attack successfully and the consequence from this attack scenario are plotted in the A/C chart as shown in **Figure 17**. The initial model had an achievability of 1 since there is nothing stopping the adversaries, and the calculated consequence was 17.83 rem. The modified model had an achievability of 0.12 and a dose consequence of 1.2 rem. The A/C chart shows that security features reduce the attack's achievability while the safety actions reduce the radiological consequence. The resultant vector shows the 2S risk reduction. Note that the final numbers are not important because it is merely a hypothetical case study. However, it illustrates how physical security can be assessed for an early A/SMR design by combining security elements and safety actions.

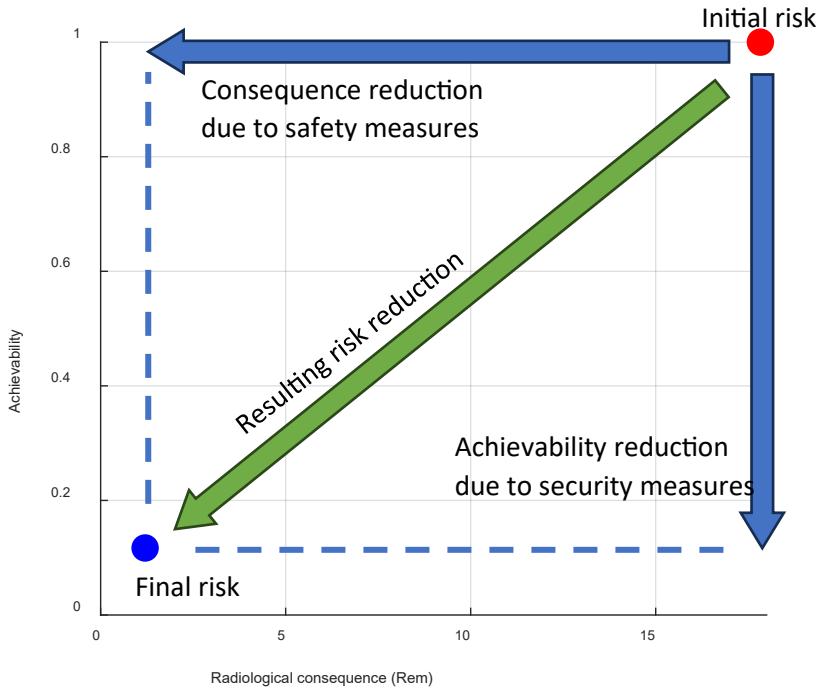


Figure 17. Reduction in attack achievability and consequence.

EMRALD provides a visual flow of the simulated scenarios in a Sankey diagram format as shown in **Figure 18**. This diagram allows analysts to see where the scenario branches and the number of scenarios in each branch. It is particularly useful for dynamic scenarios. The figure below shows that adversaries are interrupted most of the time while they are attempting to breach Door2, with a smaller percentage at when they are working to sabotage Target2. Such insights are useful for the analyst to track the scenario's progressions and identify the vulnerable areas to improve on.

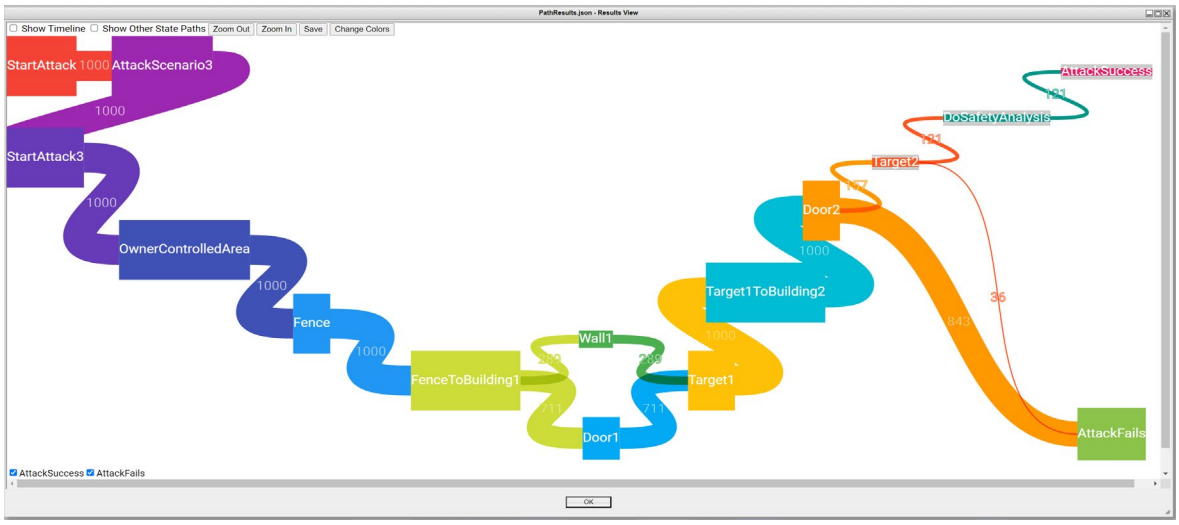


Figure 18. Sankey diagram showing the branching of attack scenario.

5. SUMMARY AND FUTURE WORK

This work proposes a dynamic evaluation methodology to relax the conservatism in physical security evaluation, by leveraging an ongoing work in the Light Water Reactor Sustainability pathway utilizing EMRALD tool. The work extends EMRALD's capability to support a sandbox feature where analysts can easily create attack scenarios and modify advanced/small modular reactor (A/SMR) security and safety features using templates. EMRALD is completely free to use at https://emraldapp.inl.gov. We have developed basic templates including physical barriers, intrusion sensors, physical areas, and safety actions, that can be downloaded from EMRALD's GitHub site: https://github.com/idaholab/EMRALD . These templates use generic data that users can adjust with their own dataset.

The proposed methodology combines security and safety by assessing sabotage effects up to the radiological consequence to the public instead of merely the core damage state. This practice follows the ASME/ANS standard for advanced non-light-water reactors currently proposed for endorsement by the NRC. The combination of security and safety (2S) is expressed in an achievability-consequence chart. A hypothetical case study using a generic sodium-cooled fast reactor (SFR) facility is presented in this report to demonstrate this methodology.

This work will benefit A/SMR vendors and utilities to implement security by design during the reactor design iteration phase, such that they do not have to perform upgrades and retrofits to the reactor after it is installed to improve its physical protection system. The tool may also be used to analyze domestic or foreign reactor designs to support the International Nuclear Security for Advanced Reactors (INSTAR) bilateral missions.

This work was proposed for two years, with the first year is focused on the development of methodology and tool, and the second year is focused on running the case studies using generic SFR and HTGR facilities to explore the capabilities of the tool and to derive lessons learned from implementing the methodology on various scenarios. Since these various scenarios will extend beyond reactor core, we plan to improve the capabilities of radiological assessment code SRT to support sabotage consequences on non-core sources.

The models within SRT currently focus on potential radionuclide release events associated with fuel within the core of the reactor. However, given the utilization of passive and inherent safety systems and features, radionuclide release from the core may be very unlikely (from both a safety and security perspective). Given this, there is an increased focused on potential radionuclide release for non-core sources of radioactivity, such as purification systems, fuel movement, and fuel storage. To address these scenarios, the capabilities within SRT are to be expanded to permit the release of radionuclides at any location within the modeled reactor facility. Therefore, radionuclide release for non-core sources could be assessed utilizing the same models in SRT developed for the reactor and core. In addition, potential events with releases from multiple sources simultaneously (including core and non-core) could also be simulated. This future work to further enable use for non-core sources can prove to be very useful for SSEBD and to developers and possibly end users.

Given the utilization of SRT within design and licensing activities for reactor facilities, there is a thorough software quality assurance (SQA) program in place for the code, which includes verification and validation testing and documentation. As part of the expansion of code capabilities, the existing SQA

suite will also be updated and expanded, including new verification tests and revised code documentation.

6. REFERENCES

1. Christian, R., et al. 2023. "Technical and Regulatory Aspects of Integrating Safety and Security at Nuclear Power Plants." ANS-PSA 2023, Knoxville, Tennessee. https://www.ans.org/meetings/npic13psa2023/sessions/attachment/paper-7927/.
2. ASME/ANS. 2021. "Probabilistic Risk Assessment Standard for Advanced Non-LWR Nuclear Power Plants." American Society of Mechanical Engineers/ American Nuclear Society, ASME/ANS RA-S-1.4-2021. https://www.asme.org/codes-standards/find-codes-standards/ra-s-1-4-probabilistic-risk-assessment-standard-advanced-non-light-water-reactor-nuclear-power-plants/2021/drm-enabled-pdf.
3. U.S. NRC. 2022. "Regulatory Guide 1:247: Acceptability of Probabilistic Risk Assessment Results for Non-Light-Water Reactor Risk-Informed Activities." United States Nuclear Regulatory Commission. https://www.nrc.gov/docs/ML2123/ML21235A008.pdf.
4. Grabaskas, D., et.al. 2022. "Development of the Versatile Test Reactor Probabilistic Risk Assessment." Nuclear Science and Engineering, Vol. 196, pp. 278—288. https://doi.org/10.1080/00295639.2021.2014741.
5. NEI. 2019. "Risk-Informed Performance-Based Technology Inclusive Guidance for Non-Light Water Reactor Licensing Basis Development." NEI 18-04 Rev. 1, Nuclear Energy Institute. https://www.nrc.gov/docs/ML1924/ML19241A472.pdf.
6. Christian, R., et al. 2022. "A Dynamic Risk Framework for the Physical Security of Nuclear Power Plants." Nuclear Science and Engineering. https://doi.org/10.1080/00295639.2022.2112899.
7. INL. 2023. "Physical Security Timeline Analysis in Support of Advanced Reactor Demonstration and Deployment." Idaho National Laboratory, INL/RPT-23-71219. https://www.osti.gov/biblio/1959000.
8. INL. n.d. "EMRALD." Idaho National Laboratory, Idaho Falls, ID. Accessed Jul. 28, 2021. https://emrald.inl.gov/SitePages/Overview.aspx.
9. SNL. 2019. "Modeling for Existing Nuclear Power Plant Security Regime." Sandia National Laboratory, SAND2019-12014. https://www.osti.gov/biblio/1570399.
10. IAEA. 2019. "Nuclear Power Plant Case Study for the Nuclear Security Assessment Methodologies for Regulated Facilities." International Atomic Energy Agency, IAEA-TECDOC-1868. https://inis.iaea.org/collection/NCLCollectionStore/_Public/51/003/51003773.pdf.
11. SNL. n.d. "PathTrace: Adversary Pathway Analysis and Exploration." Sandia National Laboratory, Accessed July 28, 2023. https://insetools.sandia.gov/pathtrace.
12. SNL. 2021. "U.S. Domestic Pebble Bed Reactor: Security-by-Design." Sandia National Laboratory, SAND2021-13122 R. https://www.osti.gov/biblio/1832296.
13. Grabaskas, D. 2022. "Development of the Simplified Radionuclide Transport (SRT) Code Version 2.0 for Versatile Test Reactor (VTR) Mechanistic Source Term Calculations." 2022 International Conference on Fast Reactors and Related Fuel Cycles, Beijing, China. April 25- 28, 2022. https://conferences.iaea.org/event/218/papers/18975/files/8152-259.pdf.
14. ANL. 2022. "Simplified Radionuclide Transport (SRT) Code: User's Manual." Argonne National Laboratory, ANL-SRT-4, Rev 2.0.2, 2022.

Intentionally Blank

APPENDIX A: HOW TO USE EMRALD TEMPLATES

To create a template based on an existing diagram, the user can right-click on the diagram in the sidebar and choose “Make Template” (this is shown in **Figure A-1**).

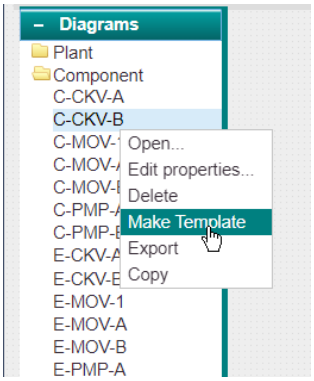


Figure A-1. Right-click a diagram and select "Make Template.”

In the window that appears (shown in **Figure A-2**), the user is presented with several options to control how the template will be saved. In the top portion of the window, a default name will be given to the template, which will be the diagram’s name with “_Template” appended as a suffix; the user can edit this as desired. The user can also add an optional description for the template.

The template can also be assigned to a “group,” which functions similarly to folders on a computer; the user can create as many groups as necessary, and sub-groups can be created within a group. To create a group, the user can simply click the “Create a new Group” button and enter the name of the group in the following prompt (shown in **Figure A-3**). To create a sub-group, the user can navigate to the group which will be the parent group and click the “Create a new Sub Group” button; this process is shown in **Figure A-4**. To make navigating groups easier, there are two views; the default view is the group view, as shown in **Figure A-5**. In group view, the user can navigate to subgroups simply by clicking a group and navigate back up to parent groups by clicking the “Go Up” button. The user can switch to tree view by clicking the “Tree View” button; an example of a tree view fully collapsed is shown in **Figure A-6**, and a fully expanded example is shown in **Figure A-7**. Simply clicking a group name within the tree will navigate the user to that group; clicking the folder next to the name will expand or collapse that group. When navigating within groups, the current path will be shown in a white bar just above the “Create a new Group” button (shown in **Figure A-8**); this is the group path that will be assigned to the template (if the white path bar is not visible, then the template will not be assigned to any group and will appear in the root group path.

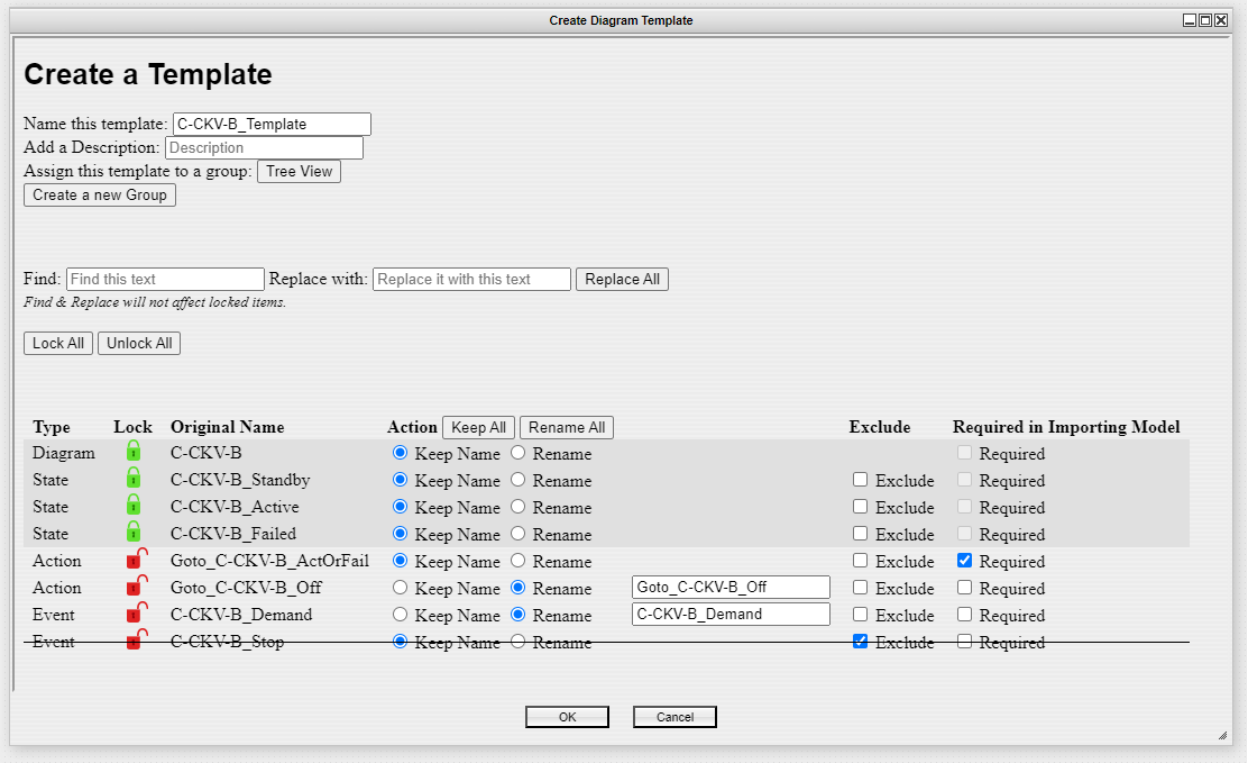


Figure A-2. "Create Diagram Template" window.

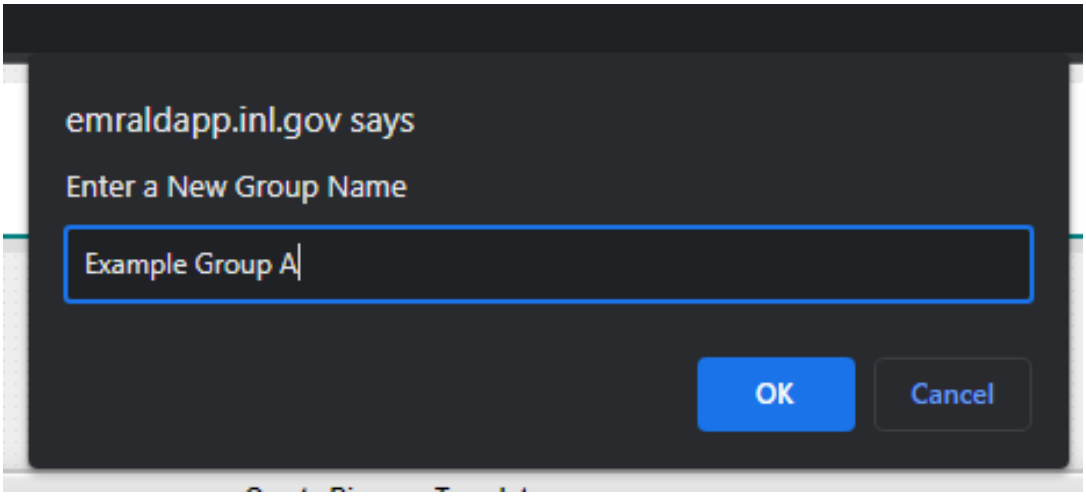
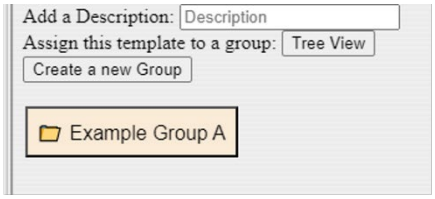
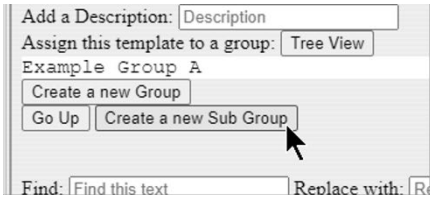


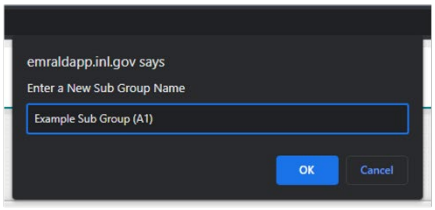
Figure A-3. Prompt for group name when creating a new group.



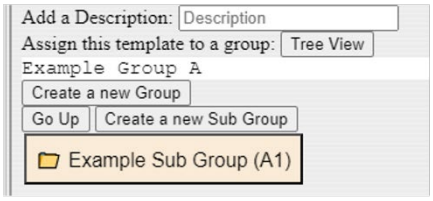
Step 1: Navigate to a group by clicking on it (in this case, click "Example Group A")



Step 2: The current path is displayed with a white background; verify the path is correct. Then, click the "Create a new Sub Group" button.



Step 3: Enter a name for the sub-group and then click "OK."



Step 4: Observe the sub-group has been created. Navigate to it by clicking it or navigate back out of the "Example Group A" group by clicking the "Go Up" button.

Figure A-4. Steps to create a sub-group.

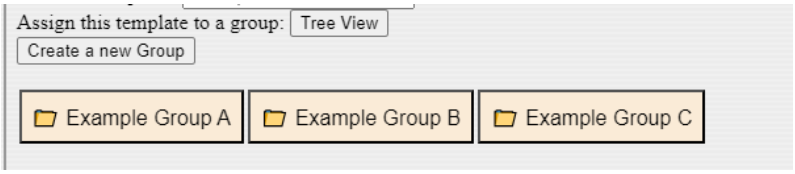


Figure A-5. Group view.

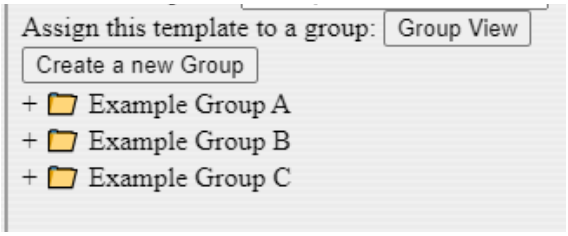


Figure A-6. Tree view (collapsed).

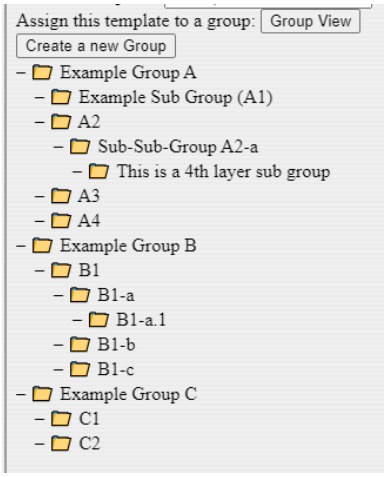


Figure A-7. Tree view (fully expanded).

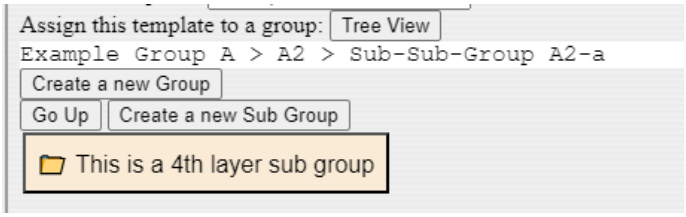


Figure A-8. The current path is displayed when in a group.

Beneath the groups is a find and replace feature. This will find the specified text within the names of "unlocked" items and replace it with some other specified text. The items are listed in the table below; unlocked items are denoted by red open-lock icons, and locked items are denoted by green closed-lock icons. Some items may be locked or unlocked by default. Clicking on the lock icon will toggle its lock state. There are also buttons available to lock or unlock all items.

In addition to the lock icons, the table of items also displays the item's type (such as "Diagram," "State," "Event," etc.), the original name as it appears in the diagram, an action to either keep the original name or rename the item for the template, an exclude option, and a required option. When the "Keep Name" action option is selected, the original name will be the name that appears in the template; when the "Rename" action option is selected, a text box will appear to the right of that option for the user to enter a new name. (Buttons have been included to easily switch between renaming or keeping the original names of all items; these buttons only affect unlocked items). The exclude option, when checked, will exclude the item from being included in the template; this is denoted by a line through the item's table row. When the "Required in Importing Model" option is checked, then the template can only be used in models that already have an item existing in the model with the same name and type as the required item.

When all template settings are to the user's liking, the user can click the "OK" button to create the template.

To create a diagram from a template, the user can right-click on the "Diagrams" header in the sidebar and choose "New Diagram" (this is shown in **Figure A-9**). In the window that appears, the user can navigate groups to find the desired template in a fashion similar to the group navigation when creating a template. In addition to the group view and tree view previously described, there is also a list view, which simply shows a list of templates with their assigned group path. Examples of these three views are shown in **Figure A-10**, **Figure A-11**, and **Figure A-12**. Some templates may appear to be grayed-out; when this happens, it means that the template requires an item to already be existing in the model. Clicking the grayed-out template will display a message to the user explaining why the template cannot be selected; an example of this message is shown in **Figure A-13**. When a valid template is selected, the templates background will turn a darker gray to indicate it is selected, and the "Type," "Name," and "Desc" fields will be disabled; the name and description can be edited after clicking "OK" (this is shown in **Figure A-14**).

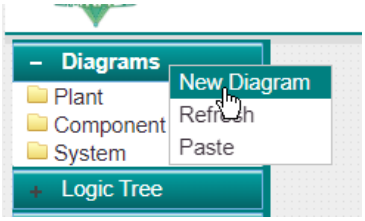


Figure A-9. Create a new diagram.

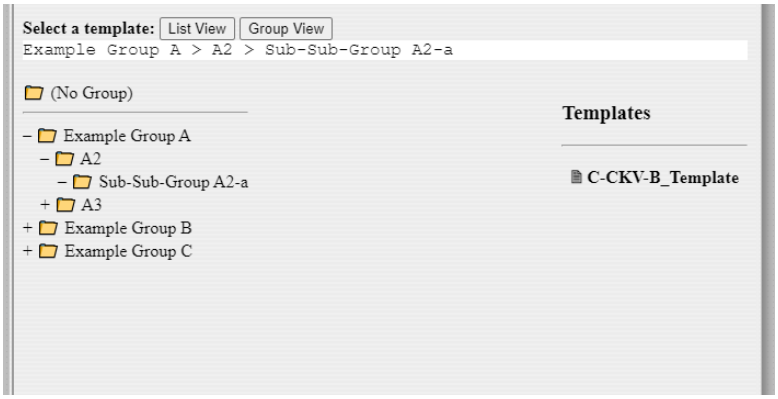


Figure A-10. Select template – "Tree View."

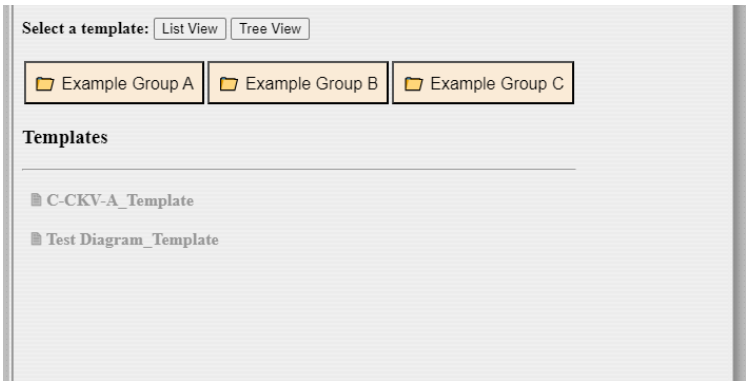


Figure A-11. Select template – "Group View."

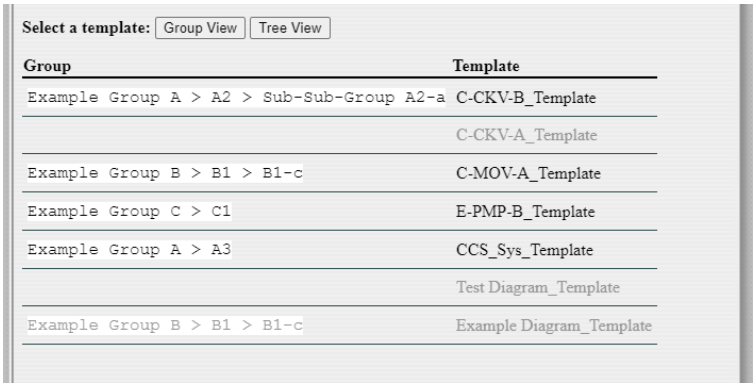


Figure A-12. Select template – "List View."

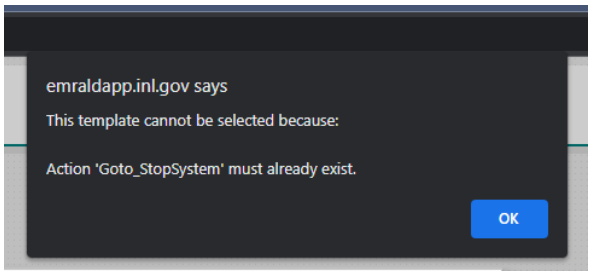


Figure A-13. "This Template Cannot Be Selected" message.

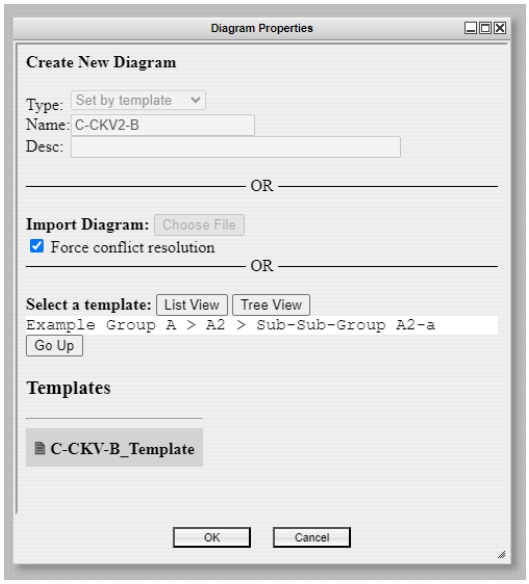


Figure A-14. A valid template is selected.

After clicking "OK" a new "Import Diagram" window will appear. From here, the user can use a find and replace feature with the ability to lock items as in the template creation process. Unlike the template creation window, though, there are two new "Action" options: "Ignore" and "Replace." The ignore option will not import that item when "OK" is pressed. The replace option will replace the existing item with the same name and type that is currently in the model with the item being imported from the template (if replace is selected and an item with the same name and type is not already present in the model, clicking "OK" will not import the template, and the "Import Diagram" window will remain visible). The rename option, along with the find and replace, and locking features work similarly as in the template creation window. There is also a column for conflicts; if an item has the same name and type as an existing item, "Conflicts!" will appear in the "Conflict" column, as shown in **Figure A-15**. Any conflicts must be resolved by selecting an appropriate action before the template can be imported. If an item is grayed out and disabled, this means the item has been flagged in the template as required. This item will not be imported from the template and must already exist in the current model; an example of a required item is shown in **Figure A-15**. When all conflicts are resolved and the appropriate actions are selected, the user can click "OK" to finish importing the template into the model.

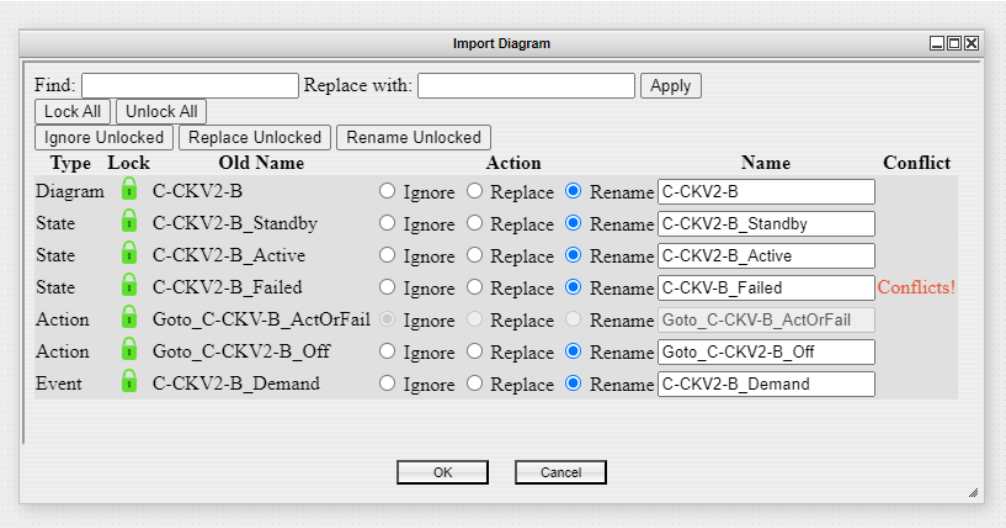


Figure A-15. "Import Diagram" window with conflict and required item.

When browsing an EMRALD model, there can often be many states, actions, and events to scroll through. To help make browsing these items easier, a filter system has been implemented. In the sidebar, a new “Filters” header has been added; expanding this will present the user with a drop-down box of the different diagram folders. When a folder has been selected, another drop-down box will appear with the diagrams contained in that folder. Selecting a diagram from this drop-down box will filter the “All Actions,” “All Events,” and “All States” lists to show only those actions, events, and states that are used in the selected diagram. For additional clarity, the “Filters” header will change from “Filters” to “Filters (Active)” when filters are active. An example comparison of what the sidebar looks like with and without filters is provided in **Figure A-16**.

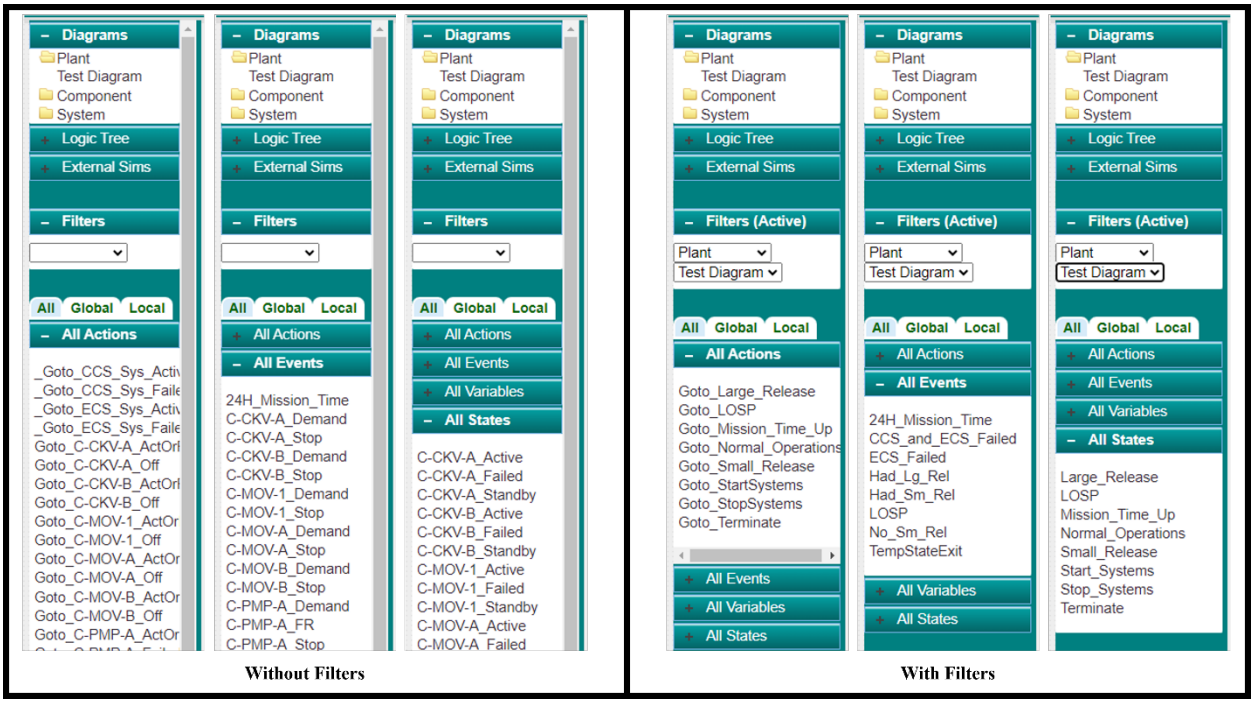


Figure A-16. Sidebar comparison without and with filters.

Intentionally blank

APPENDIX B: SRT CODE PARAMETERS

The SRT code uses several parameter files to initialize the reactor parameters. Since there are various reactor states when sabotage happens as listed in **Table 1**, the SRT parametric files need to be adjusted accordingly. Argonne National Laboratory has provided several parameter input files for the reference SFR reactor and other parameters needed for SRT to simulate the scenarios according to **Table 1**. **Note that these are postulated parameters for demonstration purposes only and do not represent any actual nuclear power plant**.

SRT Parameters for Reactor Core:

1) If sabotage occurs before "Detection":

Assumptions:

- Reactor shutdown when sabotage event occurs and no decay heat removal capabilities:
  - Fuel failures occur 20 hours after accident initiation, modeled using these SRT parameters:
    - fuel.output <- "Fuel_Core_Det.csv"
    - hot.pool.output <- "Pool_Core_Det.csv"
- No system isolation
  - Cover gas region leakrate 1.0 vol%/day, modeled using this SRT parameter:
    - leakrate["head",] <- c(1,1.0,0)
  - Containment leakrate 10 vol%/day, modeled using this SRT parameter:
    - leakrate["cont",] <- c(1,10.0,0)
- No emergency response implementation:
  - Simplistically assume that the population is closer to reactor facility (change in X/Q value and breathing rate)
  - X/Q: 5E-4 (s/m^3), modeled using this SRT parameter:
    - x.q.unc <- c(1,5E-4,0)
  - Breathing Rate: 8.3E-4 m^3/s, modeled using this SRT parameter:
    - breathing.rate.unc <- c(1,8.3e-4,0)

2) If sabotage occurs after "Detection" but before "Confirmation":

Assumptions:

- Power runback at detection stage with limited decay heat removal capabilities:
  - Fuel failures occur 30 hours after accident initiation, modeled using these SRT parameters:
    - fuel.output <- "Fuel_Core_Conf.csv"
    - hot.pool.output <- "Pool_Core_Conf.csv"
- No system isolation
  - Cover gas region leakrate 1.0 vol%/day, modeled using this SRT parameter:
    - leakrate["head",] <- c(1,1.0,0)
  - Containment leakrate 10 vol%/day, modeled using this SRT parameter:

        - leakrate[“cont”,] <- c(1,10.0,0)

- No emergency response implementation:
    - Simplistically assume that the population is closer to reactor facility (change in X/Q value and breathing rate)
    - X/Q: 5E-4 (s/m^3) , modeled using this SRT parameter:
        - x.q.unc <- c(1,5E-4,0)
    - Breathing Rate: 8.3E-4 m^3/s, modeled using this SRT parameter:
        - breathing.rate.unc <- c(1,8.3e-4,0)

3) If sabotage occurs after “Confirmation” but before mitigative actions (actions in “sabotage” row):

Assumptions:

- Reactor shutdown at confirmation stage with limited decay heat removal capabilities:
    - Fuel failures occur 35 hours after accident initiation, modeled using these SRT parameters:
        - fuel.output <- "Fuel_Core_Mit.csv"
        - hot.pool.output <- "Pool_Core_Mit.csv"
- System isolation occurs:
    - Cover gas region leakrate 0.1 vol%/day, modeled using this SRT parameter:
        - leakrate["head",] <- c(1,0.1,0)
    - Containment leakrate 1.0 vol%/day, modeled using this SRT parameter:
        - leakrate["cont",] <- c(1,1.0,0)
- No emergency response implementation:
    - Simplistically assume that the population is closer to reactor facility (change in X/Q value and breathing rate)
    - X/Q: 5E-4 (s/m^3) , modeled using this SRT parameter:
        - x.q.unc <- c(1,5E-4,0)
    - Breathing Rate: 8.3E-4 m^3/s, modeled using this SRT parameter:
        - breathing.rate.unc <- c(1,8.3e-4,0)

4) If sabotage occurs with all prevention/mitigation actions successful:

Assumptions:

- Reactor shutdown at confirmation stage with limited decay heat removal capabilities:
    - Fuel failures occur 35 hours after accident initiation, modeled using these SRT parameters:
        - fuel.output <- "Fuel_Core_Mit.csv"
        - hot.pool.output <- "Pool_Core_Mit.csv"
- System isolation occurs:
    - Cover gas region leakrate 0.1 vol%/day, modeled using this SRT parameter:
        - leakrate["head",] <- c(1,0.1,0)
    - Containment leakrate 1.0 vol%/day, modeled using this SRT parameter:

        - leakrate["cont",] <- c(1,1.0,0)

- Emergency response implementation:
    - Simplistically assume that the population is further from reactor facility (change in X/Q value and breathing rate)
    - X/Q: 1E-5 (s/m^3), modeled using this SRT parameter:
        - x.q.unc <- c(1,1E-5,0)
    - Breathing Rate: 3.5E-4 m^3/s, modeled using this SRT parameter:
        - breathing.rate.unc <- c(1,3.5e-4,0)

SRT Parameters for Fuel Storage Facility:

1. If sabotage occurs before "Detection":

Assumptions:

- Spent fuel pool port open to transfer cask:
    - Spent fuel pool vessel leakrate at 10 vol%/day, modeled using this SRT parameter:
        - leakrate["head",] <- c(1,10.0,0)
- Facility HVAC not isolated
    - Building leakrate 100 vol%/day, modeled using this SRT parameter:
        - leakrate["cont",] <- c(1,100.0,0)
- No backup power for pool cooling
    - Gross spent fuel failures occur at 60 hours, modeled using these SRT parameters:
        - channels.pins <- c(2710)
        - fuel.output <- "Fuel_FSF_Det.csv"
        - hot.pool.output <- "Pool_FSF_Det.csv"
- No emergency response implementation:
    - Simplistically assume that the population is closer to reactor facility (change in X/Q value and breathing rate)
    - X/Q: 5E-4 (s/m^3) , modeled using this SRT parameter:
        - x.q.unc <- c(1,5E-4,0)
    - Breathing Rate: 8.3E-4 m^3/s, modeled using this SRT parameter:
        - breathing.rate.unc <- c(1,8.3e-4,0)

2. If sabotage occurs after "Detection" but before "Confirmation":

Assumptions:

- Spent fuel pool port closed to transfer cask:
    - Spent fuel pool vessel leakrate at 1 vol%/day, modeled using this SRT parameter:
        - leakrate["head",] <- c(1,1.0,0)
- Facility HVAC not isolated
    - Building leakrate 100 vol%/day, modeled using this SRT parameter:

        - leakrate["cont",] <- c(1,100.0,0)
- No backup power for pool cooling
    - Gross spent fuel failures occur at 60 hours, modeled using these SRT parameters:
        - channels.pins <- c(2710)
        - fuel.output <- "Fuel_FSF_Det.csv"
        - hot.pool.output <- "Pool_FSF_Det.csv"
- No emergency response implementation:
    - Simplistically assume that the population is closer to reactor facility (change in X/Q value and breathing rate)
    - X/Q: 5E-4 (s/m^3) , modeled using this SRT parameter:
        - x.q.unc <- c(1,5E-4,0)
    - Breathing Rate: 8.3E-4 m^3/s, modeled using this SRT parameter:
        - breathing.rate.unc <- c(1,8.3e-4,0)

1) If sabotage occurs after "Confirmation" but before mitigative actions (actions in "sabotage" row):
Assumptions:

- Spent fuel pool port closed to transfer cask:
    - Spent fuel pool vessel leakrate at 1 vol%/day, modeled using this SRT parameter:
        - leakrate["head",] <- c(1,1.0,0)
- Facility HVAC isolated
    - Building leakrate 5 vol%/day, modeled using this SRT parameter:
        - leakrate["cont",] <- c(1,5.0,0)
- Backup power for pool cooling started
    - Minor fuel failures occur at 100 hours, modeled using these SRT parameters:
        - channels.pins <- c(271)
        - fuel.output <- "Fuel_FSF_Mit.csv"
        - hot.pool.output <- "Pool_FSF_Mit.csv"
- No emergency response implementation:
    - Simplistically assume that the population is closer to reactor facility (change in X/Q value and breathing rate)
    - X/Q: 5E-4 (s/m^3) , modeled using this SRT parameter:
        - x.q.unc <- c(1,5E-4,0)
    - Breathing Rate: 8.3E-4 m^3/s, modeled using this SRT parameter:
        - breathing.rate.unc <- c(1,8.3e-4,0)

2) If sabotage occurs with all prevention/mitigation actions successful:
Assumptions:

- Spent fuel pool port closed to transfer cask:

- Spent fuel pool vessel leakrate at 1 vol%/day, modeled using this SRT parameter:
  - leakrate["head",] <- c(1,1.0,0)

- Facility HVAC isolated
  - Building leakrate 5 vol%/day, modeled using this SRT parameter:
    - leakrate["cont",] <- c(1,5.0,0)

- Backup power for pool cooling
  - Minor fuel failures occur at 100 hours, modeled using these SRT parameters:
    - channels.pins <- c(271)
    - fuel.output <- "Fuel_FSF_Mit.csv"
    - hot.pool.output <- "Pool_FSF_Mit.csv"

- Emergency response implementation:
  - Simplistically assume that the population is further from reactor facility (change in X/Q value and breathing rate)
  - X/Q: 1E-5 (s/m^3) , modeled using this SRT parameter:
    - x.q.unc <- c(1,1E-5,0)
  - Breathing Rate: 3.5E-4 m^3/s, modeled using this SRT parameter:
    - breathing.rate.unc <- c(1,3.5e-4,0)

Based on these defined parameters, a RunSRT action is modeled in EMRALD using the following C# script:

For Preprocess code:

```
int counter=RunIdx;

int modifier=1;

string baseDir=@"C:\\ SRT\\";

string settingsLoc = baseDir+"SFR_Core_Base.R";

if (File.Exists(settingsLoc))

{

 string settingsStr = File.ReadAllText(settingsLoc);

 settingsStr=settingsStr.Replace("TEMPLATEFOLDER","Run"+counter);

 settingsStr=settingsStr.Replace("TEMPLATENAME","Run"+counter);


          string root = baseDir+"Run"+counter;

          // If directory does not exist, don't even try

          if (Directory.Exists(root))
```

```
        {
            Directory.Delete(root,true);
        }
 string FuelOutput="";
 string HotPoolOutput="";
 string LeakRateHead="";
 string LeakRateCont="";
 string XQ="";
 string BreathingRate="";

//undetected sabotage or sabotage while reactor is at full power
 if((Dbl_AttackTime>0)&&((Dbl_DetectionTime==0)||(Dbl_AttackTime<Dbl_T_HotStandby)))
 {
   FuelOutput="Fuel_Core_Det.csv";
            HotPoolOutput="Pool_Core_Det.csv";
            LeakRateHead="c(1,1.0,0)";
            LeakRateCont="c(1,10.0,0)";
            XQ="c(1,5E-4,0)";
            BreathingRate="c(1,8.3e-4,0)";
 }

//Sabotage occurs after “Detection” but before “Confirmation”
 else if((Dbl_AttackTime>0)&&(Dbl_AttackTime<Dbl_T_CoreShutdown))
 {
   FuelOutput="Fuel_Core_Conf.csv";
            HotPoolOutput="Pool_Core_Conf.csv";
            LeakRateHead="c(1,1.0,0)";
            LeakRateCont="c(1,10.0,0)";
            XQ="c(1,5E-4,0)";
            BreathingRate="c(1,8.3e-4,0)";
```

```
}

//Sabotage occurs after “Confirmation” but before mitigative actions (actions in “sabotage” row)
else if((Dbl_AttackTime>0)&&(Dbl_AttackTime<Dbl_T_EmergencyResponse))
{
  FuelOutput=”Fuel_Core_Mit.csv”;
        HotPoolOutput=”Pool_Core_Mit.csv”;
        LeakRateHead=”c(1,0.1,0)”;
        LeakRateCont=”c(1,1.0,0)”;
        XQ=”c(1,5E-4,0)”;
        BreathingRate=”c(1,8.3e-4,0)”;
}

//Sabotage occurs with all prevention/mitigation actions successful
else if((Dbl_AttackTime>0)&&(Dbl_AttackTime>Dbl_T_EmergencyResponse))
{
  FuelOutput="Fuel_Core_Mit.csv";
        HotPoolOutput="Pool_Core_Mit.csv";
        LeakRateHead="c(1,0.1,0)";
        LeakRateCont="c(1,1.0,0)";
        XQ="c(1,1E-5,0)";
        BreathingRate="c(1,3.5e-4,0)";
}
settingsStr=settingsStr.Replace("TEMPLATEFUELCSV",FuelOutput);
settingsStr=settingsStr.Replace("TEMPLATEPOOLCSV",HotPoolOutput);
settingsStr=settingsStr.Replace("LEAKRATEHEADTEMPLATE",LeakRateHead);
settingsStr=settingsStr.Replace("LEAKRATECONTTEMPLATE",LeakRateCont);
settingsStr=settingsStr.Replace("XQTEMPLATE",XQ);
settingsStr=settingsStr.Replace("BREATHINGRATETEMPLATE",BreathingRate);
```

```
 File.WriteAllText(baseDir+"Run"+counter+".R", settingsStr);
}

return (" SRT.R Run"+counter+".R");
```

For the Postprocess script:

```
string baseDir=@"C:\\SRT\\";
int counter=RunIdx;
List<String> retStates = new List<String>();

string resultsLoc = baseDir+"Run"+counter+@"\Summary\Summary_TEDE_CDF.txt";
if (File.Exists(resultsLoc))
{
        retStates.Add("+ReadSRT");
}

return retStates;
```

The Preprocess script in the RunSRT action does the following:

1. Obtains the EMRALD iteration number (RunIdx)
2. Reads SRT's input file for the reference SFR, named SFR_Core_Base.R
3. Replaces the TEMPLATEFOLDER and TEMPLATENAME fields in SFR_Core_Base.R with a Run number: Run1, Run2, Run3, etc.
4. Compares time variables from EMRALD simulation, i.e., intrusion detection time, attack time, responder arrival time, reactor hot standby time, reactor shutdown time, and emergency response time, to determine the applicable state of the attack outcome
5. Substitutes the parameters in SFR_Core_Base.R accordingly
6. Saves the updated parameter file in a new text file named as Run1, Run2, Run3, etc.
7. Executes SRT code using the new parameter file and wait for SRT's completion

The Postprocess script in the RunSRT action does the following:

1. Locates the SRT output file that records the radiological dose: Summary_TEDE_CDF.txt
2. If the output file is found, transitions EMRALD to the ReadSRT state

The ReadSRT state reads SRT output file using the following C# script:

```
double mydose=0.0;
string baseDir=@"C:\\SRT\\";
string resultsLoc = baseDir+"Run"+RunIdx+@"\Summary\Summary_TEDE_CDF.txt";

using(System.IO.StreamReader sr = new System.IO.StreamReader(resultsLoc))
{
  string line;
  int counter=1;
  while((line = sr.ReadLine()) != null)
  {
                    string[] split = line.Split();
        if(counter==15)
        {
                    mydose=Double.Parse(split[3]);
        }
    counter++;
  }
}

return mymean;
```

What the script does is:

1. Opening the Summary_TEDE_CDF.txt
2. Reading the 15th line of the file, where the radiological dose is printed
3. Performing string operations to extract the radiological dose value