INL REPORT

RAVEN: Development of the Adaptive Dynamic Event Tree Approach

Andrea Alfonsi, Cristian Rabiti, Diego Mandelli, Joshua Cogliati, Robert Kinoshita

Idaho National Laboratory

RAVEN: Development of the Adaptive Dynamic Event Tree Approach

Andrea Alfonsi
Idaho National Laboratory
P.O. Box 1625
Idaho Falls, ID 83415-3870
andrea.alfonsi@inl.gov

Cristian Rabiti
Idaho National Laboratory
P.O. Box 1625
Idaho Falls, ID 83415-3870
cristian.rabiti@inl.gov

Diego Mandelli
Idaho National Laboratory
P.O. Box 1625
Idaho Falls, ID 83415-3850
diego.mandelli@inl.gov

Joshua Cogliati
Idaho National Laboratory
P.O. Box 1625
Idaho Falls, ID 83415-3870
joshua.cogliati@inl.gov

Robert Kinoshita
Idaho National Laboratory
P.O. Box 1625
Idaho Falls, ID 83415-2210
robert.kinoshita@inl.gov

Acknowledgment

Contents

Figures

1 Introduction

RAVEN (Risk Analysis and Virtual control ENviroment), under the support of the Nuclear Energy Advanced Modeling and Simulation (NEAMS) program [1], is increasing its capabilities to perform probabilistic analysis of stochastic dynamic systems. This supports the goal of providing the tools needed by the Risk Informed Safety Margin Characterization (RISMC) path-lead [8] under the Department of Energy (DOE) Light Water Reactor Sustainability program [2]. In particular, the development of RAVEN in conjunction with the thermal-hydraulic code RELAP-7 [5], will allow the deployment of advanced methodologies for nuclear power plant (NPP) safety analysis at the industrial level. The investigation of accident scenarios in a probabilistic environment for a complex system (i.e. NPPs) is not a minor task. The complexity of such systems, and a large quantity of stochastic parameters, lead to demanding computational requirements (several CPU/hour). Moreover, high consequence scenarios are usually located in low probability regions of the input space, making even more computational demands of the risk assessment process.

This extreme need for computational power leads to the necessity to investigate methodologies for the most efficient use of available computational resources, either by increasing effectiveness of the global exploration of input space, or by focusing on regions of interest (e.g. failure/success boundaries, etc.). The milestone reported in September 2013 [4] described the capability of RAVEN to perform exploration of the uncertain domain (probabilistic space) through the support of the well-known Dynamic Event Tree (DET) approach. This report will show that the Dynamic Event Tree approach can be considered intrinsically adaptive around the failure prone input zone, if one or more of the uncertain parameters is/are responsible for the transition of interest (e.g. failure or success). Leveraging on this feature is a natural choice to extend the classical DET approach to the Adaptive Dynamic Event Tree (ADET). This extension of the DET methodology to ADET and its implementation in the RAVEN code is the subject of this report. In order to show the effectiveness of this methodology, a Station Black Out (SBO) scenario for a Pressurized Water Reactor has been employed. The ADET approach will be used to focus the exploration of the input space toward the computation of the failure probability of the system (i.e. clad failure). This report is organized in four additional sections. Section 2 recalls the concept of the DET methodology. Section 3 reports how the newer developed algorithm is employed. Section 4 is focused on the analysis performed on the PWR SBO, and, section 5 draws the conclusions.

2 The Dynamic Event Tree Methodology

The Dynamic Event Tree approach (DET) is a well-known methodology, developed based on the Static (or Classical) Event Tree (ET) technique. In the ET approach, starting from an initiating event (i.e. accident event), all possible final outcomes of the accident scenario are obtained by assembling of status combinations of the system (e.g., responsiveness of components/subsystems) that might impact the final outcome. This leads to the classical tree structure. Each branch end (accident scenario outcome) could be characterized therefore by the status of the components/subsystem that leads to such outcome. The probability of such outcome is the product of the probability of the components/subsystems to be in the corresponding status. The two main disadvantages of this approach are that timing/sequencing of events and system dynamic responses are not explicitly accounted for in the analysis.

In order to overcome these limitations a dynamic approach is necessary. The DET technique is, in fact, capable of simulating the probabilistic system evolution in a consistent way with respect to temporal evolution of the accident scenarios [3] [7]. This means the DET enables a feedback mechanism between the accident scenarios defined by probability of events over the accident timeline, and the dynamic response of the system. In the DET approach, a single initiating event (root) leads to multiple final outcomes. The branching occurs at user specified times/conditions and/or when an action is required by the operator and/or by the control system. The branching process creates a deterministic sequence of events based on the time of their occurrence (see fig. 1). The criteria for branching are generally geared toward providing complete coverage of all-possible scenarios. The number of branches (dynamic sequences) can become very large very quickly. For this reason independent branches are simulated in parallel.

This approach leads to a more realistic analysis of the considered system (e.g., NPP) than the classical static Event Tree. General Dynamic Probabilistic Risk Assessment methodologies, among which DET, are designed to consider the timing of single events explicitly, which can become extremely important especially when non linearity of the system response might alter the scenario evolution.

The main idea of this methodology is to let a system code (i.e., RELAP-7) to determine the pathway of an accident scenario within a probabilistic environment rather than statically combining the effects of all possible component status toward the final outcome of the accident scenario.

From a simulation point of view, a DET analysis starts with a single simulation that, most likely, represents an abnormal condition of the NPP immediately after an initiating event (e.g., fire, flooding, etc.). Every time during the simulation, the system evolution leads to a probabilistic event (e.g., failure of a relief valve, recovery of a safety feature, etc.), several simulation branches are spawned. Each simulation carries along one of the possible outcomes of the probabilistic event and the associated event probability. As illustrated in fig. 1, after an initiating event, which leads to an abnormal condition of the system, the simulation follows the accident sequence and a pipe status is affected by a probability of failure (i.e., pipe failure probability as a function of the internal pressure) described by a Probabilistic Distribution Function (PDF) and a corresponding Cumulative Distribution Function (CDF). The user provides a grid of probability thresholds, with
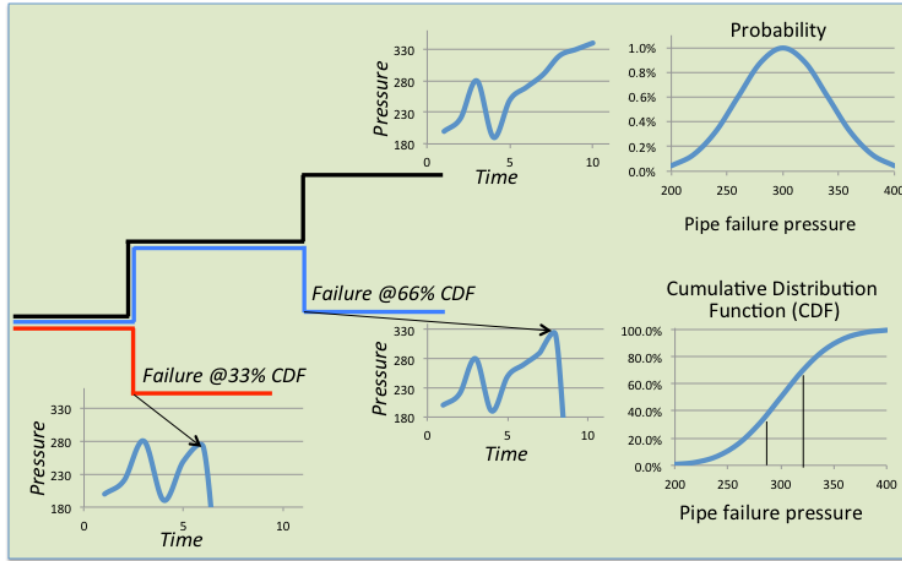
Figure 1: Dynamic Event Tree Conceptual Scheme.

respect the CDFs of the uncertain parameters (in this case on the failure pressure). Every time the simulation detects a pressure corresponding to a threshold in the probability grid, a new set of simulation branches are generated (i.e., a branch where the pipe is damaged and another where nothing happened).

In general, each sequence continues until another event occurs and a new set of branches is spawned. The simulation ends when an exit condition or a maximum mission time is reached. At the end of the DET analysis, all the branches (that represent all the possible system evolutions) are assembled in order to create a set of completed histories.

3 Adaptive Dynamic Event Tree

The main subject of this report is to explain the newer Adaptive Dynamic Event Tree (ADET) methodology developed within the RAVEN code. As already mentioned, the DET approach is extremely effective (achiving almost a perfect tradeoff) with respect to computational resources usage and the effective exploration of the uncertain domain (input space). The proposed approach is designed to further focus these features of the DET methodology toward the characterization of some properties of the input space. More specifically, the approach that will be introduced is an adaptive methodology that defines the branching strategy in order to identify more efficiently (lower computational costs) and more accurately the location of transitions in the input space (e.g., failure vs. success). The characterization of the input space is translated into an optimization-like problem for the identification of the so-called Limit Surface, and accelerated through the employment of artificial intelligence-based algorithms that help the selection of future branching points.

The concepts mentioned here in brief are explained in detail in the following sub-sections.

3.1 Concept of Limit Surface

As already mentioned, the adaptive methodology can be considered as a goal oriented sampling strategy for the research of the limit surfaces (LSs). To define the meaning of the LS, it is necessary to introduce the concept of goal function. Without making a rigorous mathematical treatment, the goal function is an object that is defined as a part of the system outcome space. In a safety context, the goal function usually represents the success or failure (transition) of the system. Therefore, if $\bar{x}_f$ is the final status of the system, $C\left(\bar{x}_f\right)$ the goal function ($C\left(\bar{x}_f\right) = 1$ system success, $C\left(\bar{x}_f\right) = -1$ system failure),), it is possible to define $\bar{x}_{f,L}$ as the transition surface in the output space with respect to the goal function. This statement can be translated in the following mathematical expression: $\bar{x}_{f,L} = \left\{\bar{x}_f \middle| \left|\overline{\nabla}C\left(\bar{x}_f\right)\right| = \infty\right\}$. If $\bar{x}\left(t,\bar{p},\bar{x}_0\right)$ represents the evolution of the system (that can be considered deterministic, once the initial condition ($\bar{x}_0$) and stochastic model parameters ($\bar{p}$) have been chosen), then is it possible to identify the set of pairs $\left(\bar{p},\bar{x}_0\right)_{LS}$, in the input space, which leads the system outcome to match $\bar{x}_{f,L}$. Such set of points $\left(\bar{p},\bar{x}_0\right)$ represents the limit surface, in the input space.The LS represent therefore the boundary between inputs leading to success or failure of the system, and more generally boundaries among transition regions.

Since the probability of a particular outcome is equivalent to the probability of being in the input space that leads to that outcome, the probability of a system outcome (e.g. success/failure) can be evaluated by computing the probability of being in the space surrounded by the LS. The determination of the LS is extremely important since its informative content. Its knowledge: allows to efficiently evaluate risk functions, informs regarding which uncertainties are the most relevant from a risk point of view (ranking of uncertain space), defines safe regions to be explored for plant optimization, identifies risk neutral design direction, etc. Unfortunately, the determination of LSs is very computationally expensive, since a brute-force approach would imply the evaluation of each point of an N-dimensional grid that discretizes the input space. The number of points in this

grid is proportional to the requested accuracy. In order to overcome these limitations, a possible approach is the employment of acceleration schemes based on Reduced Order Models (ROMs). ROMs are used to predict the location of the LS in order to drive the exploration of the input space toward its possible location. The Adaptive Dynamic Event Tree is based on this concept.
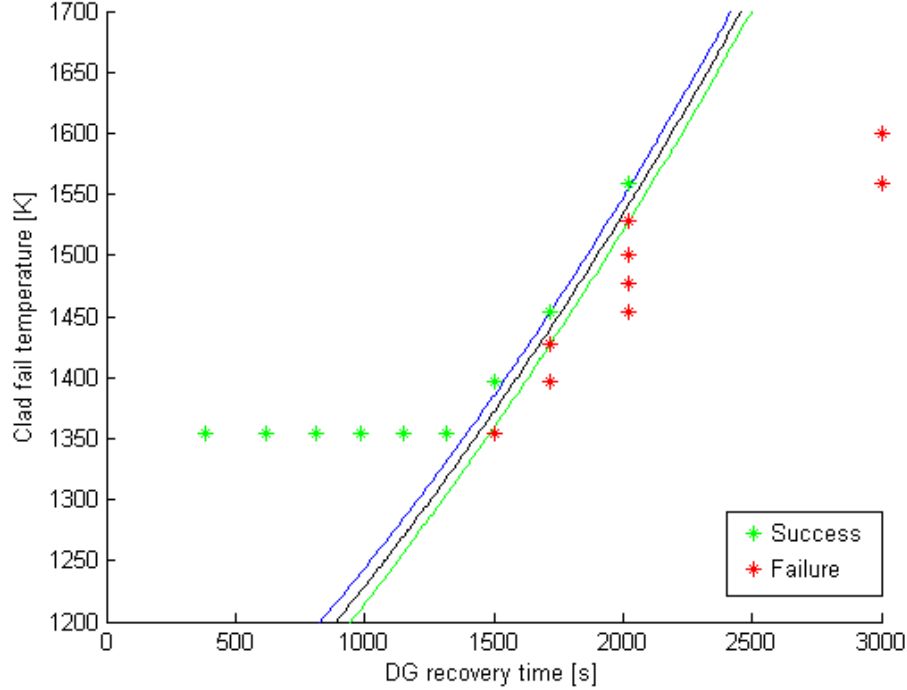

Figure 2: Dynamic Event Tree Limit Surface

3.2   Reduced Order Models

As mentioned in the previous section, the Adaptive Dynamic Event Tree uses ROMs in order to accelerate the search of transition boundaries represented by the LS. Simplistically, a ROM is a mathematical model of fast solution trained to predict a response of interest of a physical system. The training process is performed by sampling the response of a physical model with respect to variations of parameters that are subject to probabilistic behavior. The results (outcomes of the physical model) of those samplings are fed into the algorithm representing the ROM that tunes itself to replicate those results. More specifically, in RAVEN case the reduced order model is constructed to emulate a numerical representation of a physical system. In fact, as already mentioned, the physical system model would be too expensive to be used to explore the whole input space, exploration that would be needed for the limit surface search. Under certain assumptions, the ROM ability to predict the output space of a system improves, increasing the number of training samples (better representation of the whole domain). This is not always true, since some of the ROMs used might be subject to over-fitting issues. In RAVEN, several different Reduced Order

11

Model types are available, such as Support Vector Machine, Neighbors based, multi-class models, Quadratic Discriminants, etc. All those supervised learning algorithms have been imported via an Application Programming Interface (API) within the scikit-learn [6] library.

The acceleration schemes in the adaptive approach use a class of supervised learning algorithms usually referred to as classifiers. In essence, classifiers are ROMs specialized to represent a binary response of a system (e.g., failure/success, on/off, etc.), like the goal function of interest in our case.
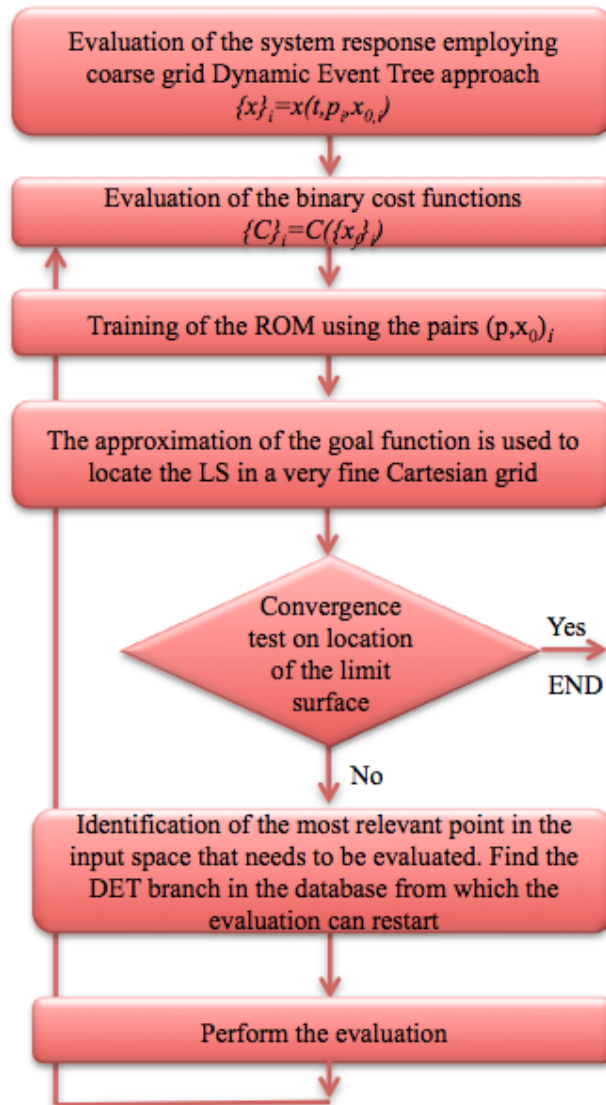

Figure 3: Adaptive Dynamic Event Tree Scheme

3.3 Adaptive DET

The main idea of the application of the previously explained adaptive sampling approach to the DET comes from the observation that the DET is intrinsically adaptive.

In order to explain this statement, as an example, fig. 2 shows a LS generated by the DET sampling methodology currently available in RAVEN. In this case, a goal function based on the clad max temperature has been used:

$$\{C\}_i = \begin{cases} 1 & T_{C,i} \geq T_{C,F} \\ -1 & T_{C,i} < T_{C,F} \end{cases} \tag{1}$$

where $T_{C,i}$ is the clad temperature and $T_{C,F}$ is the failure temperature, sampled by triangular distribtuion. As it can be seen, the DET method, when one or more uncertain parameters directly influence the outcome of the goal function, tends to search for the LS with accuracy equal to the user defined grid discretization in the input space. For this reason, it appears natural to use the DET approach to perform a goal-function oriented pre-sampling of the input space. The proposed approach can be described through the following steps (see Figs. 3 4):

1. A limited number of points in the input space $(\bar{p}, \bar{x}_0)_i$ are selected via a DET approach

2. The system code is then used to compute all the possible evolution of the system starting from the above selected input points:

$$\bar{x}_i = \bar{x}(t, \bar{p}_i, \bar{x}_{0,i}) \tag{2}$$

3. The goal function $C(\bar{x}_f)$ (in this case of type Boolean) is evaluated at the final phase space coordinate of the system (outcome):

$$\{C\}_i = \left\{\bar{x}(t = t_{end}, \bar{p}_i, \bar{x}_{0,i})\right\} \tag{3}$$

4. The set of pairs $(\bar{p}, \bar{x}_0)_i \rightarrow \{C\}_i$ are used to train a ROM

5. The ROM is used to predict the value of the goal function on a regular Cartesian grid in the domain space. The grid meshing is dictated by the user-requested accuracy in the determination of the LS location:

$$\{c\}_j \cong ROM\left(\{\bar{x}_0\}_j\right) \quad j = 1, ..., M \tag{4}$$

where $j = 1, ..., M$ points on the grid

6. The regions identified by a change in value of $\{c\}_j$ (between -1 and 1) are therefore the ROM prediction of the limit surface location:

$$(\bar{p}, \bar{x}_0)_{LS} \tag{5}$$

7. The position of the LS is compared with the one at the previous iteration; if no changes are consecutively detected for a user-defined number of times than the iterations stop; otherwise a new point, in the input space, needs to be selected to increase the ROM training set.

8. The next selected point to be added to the training set is the one located on the limit surface that is the farther from all the other already selected points

9. A hierarchical searching process is performed on the DET branches already evaluated and the starting point (branch) for the new evaluation is determined
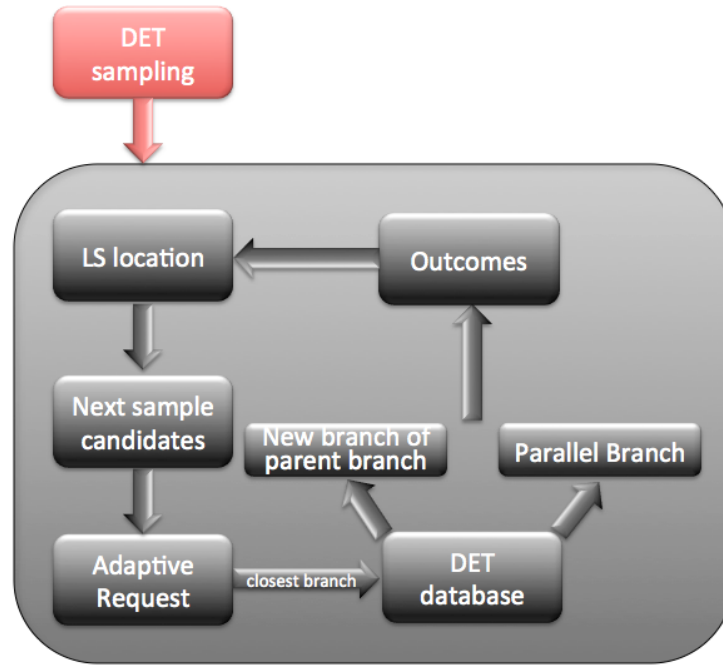
10. The process restart from point 2.

Figure 4: Dynamic Event Tree Flow Chart

In order to understand how the algorithm works, it is necessary to explain in details the first and ninth steps of the previous algorithm flow. The ADET methodology starts with a standard DET that performs the initial exploration of the input space, exploiting its capability to investigate the uncertain domain in a reasonably short time. As already mentioned and shown in fig. 2, when a goal function is directly influenced by one (or more) uncertain parameter(s), the DET tends to increase the sampling density in proximity of the transition boundaries (LS). This means that the acceleration ROM gets trained with an initial dataset closed to the LS that reasonably helps the convergence of the method.

After the initial DET, the actual adaptive scheme starts to be employed following the calculation flow previously reported. Every time a new point, in the input space, is requested, a search in the DET database is performed. This search is needed to identify the closest branch with respect to the next point needs to be evaluated. Once the closest branch has been identified, the system code

is run using that branch as starting point (i.e. the calculation restarts from an advanced point in time). This new evaluation is then added to the DET database and can be used for sub-sequential requests. This approach speeds the adaptive search up and employs an effective exploration of the input space in proximity of the transition boundaries.

In essence this methodology combines the adaptive search algorithm already present in RAVEN with a DET approach, allowing the re-usage of branches of simulations already performed rather than restarting the simulation always from the begin.

4   Dynamic PRA analysis on a simplified PWR model

In order to show the capabilities introduced with the new Adaptive DET module, a simplified PWR Dynamic PRA analysis is here presented. Figure 5 shows the scheme of the PWR model. The reactor vessel model consists of the down-comers, the lower plenum, the reactor core model, and the upper plenum. core channels (flow channels with heat structure attached to each of them) are used to describe the reactor core. The core model consists of three parallel core channels and one bypass flow channel. There are two primary loops, i.e., loop A and loop B. Each loop consists of hot leg, heat exchanger and secondary side pipes, cold leg, and primary pump. A pressurizer is attached to the loop A piping system to control system pressure. A time dependent volume (pressure boundary conditions) component is used to represent the pressurizer. Since the RELAP-7 code two-phase flow capability has not been used for this test, single-phase counter-current heat exchanger models are implemented to mimic the function of steam generators to transfer heat from the primary to the secondary.

The DPRA analysis of this simplified model here presented, it has been performed by controlling unconventional parameters to speed up the simulation and overcome some modeling challenges. In the following paragraph, the PRA station black out sequence of events is reported.
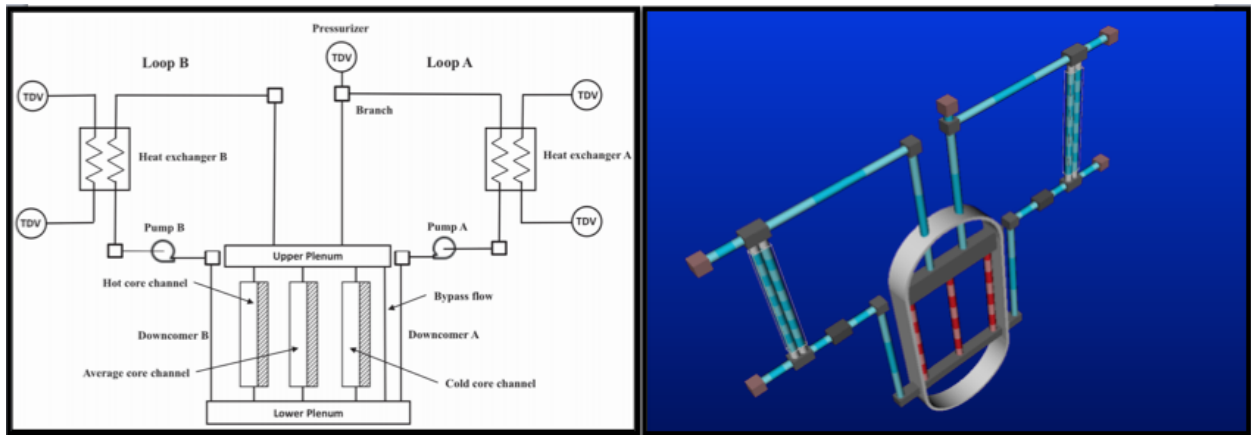

Figure 5: PWR model scheme

4.1   Station Black Out (SBO) analysis

The simulation of a SBO initiating event required the introduction, in the control logic, of several components (see Fig. 6):

- Set of 3 diesel generators (DGs) and associated emergency buses

- Primary power grid line 138 KV (connected to the NSST switchyard)

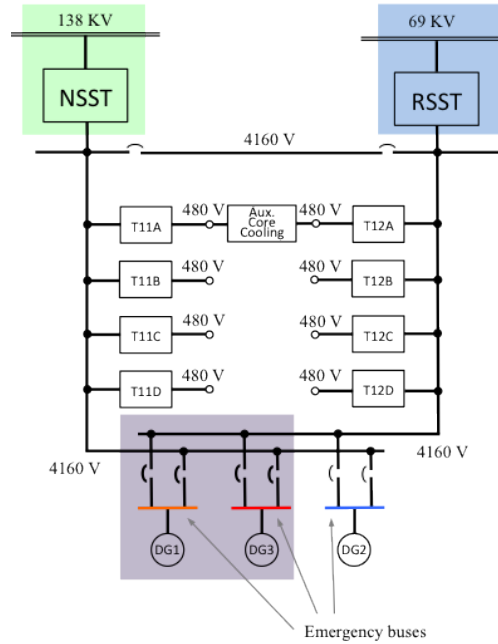- Auxiliary power grid line 69 KV (connected to the RSST switchyard)

Figure 6: Scheme of the electrical system of the PWR model

- Electrical buses: 4160 V (step down voltage from the power grid and voltage of the electric converter connected to the DGs) and 480 V for actual reactor components (e.g., reactor cooling system)

The scenario is as follows:

- An external event (e.g. earthquake, etc.) causes a Loss Of Off-site Power (LOOP) due to damage of the 138 kV line and RSST switchyard; the reactor successfully scrams and, thus, thermal energy generated in the core follows the characteristic exponential decay curve

- The set of DGs fails to start up and, hence, conditions of SBO are reached (4160 V and 480 V buses are not energized); all auxiliary cooling systems are subsequently off-line, and therefore the removal of decay heat from the reactor core is not ensured

- Without the ability to cool the reactor core, its temperature starts to rise

- In order to recover AC electric power on the 4160 V and 480 V buses, two recovery teams are assembled with the following strategy:

  - Recovery Team 1 focuses on the recovery of the DGs: due to internal damage at the DG building, two DGs (i.e., DG1 and DG3) need to be repaired (see Fig. 7 (a))

  - Recovery Team 2 focuses on the recovery of the RSST switchyard; 69KV line is energized but the RSST switchyard needs to be recovered (see Fig. 7 (b))

- Meanwhile the utility company is working on the restoration of the primary 138 KV line (see Fig. 7 (c))
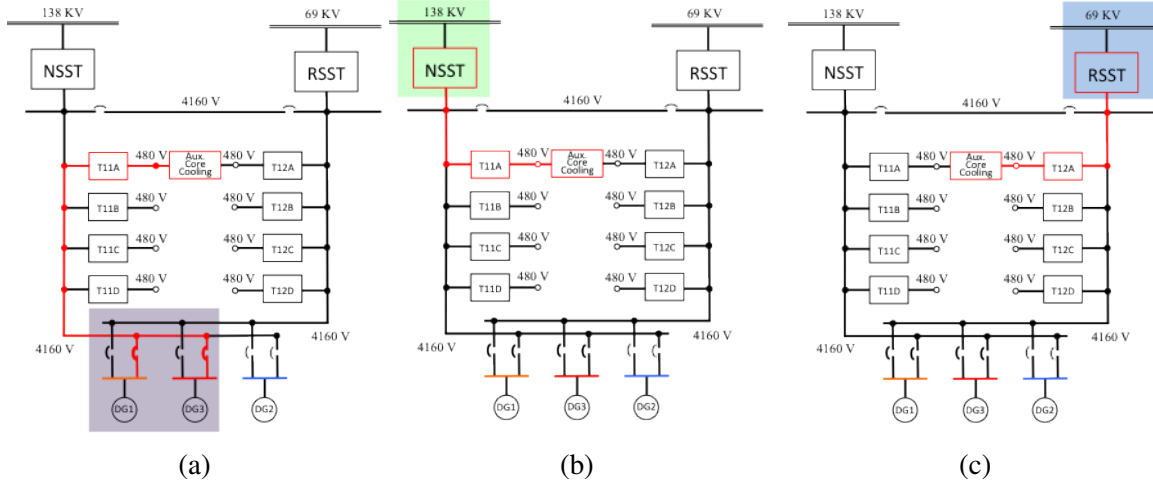
17

Figure 7: AC power recovery paths through: DGs (a), RSST (b) and 138KV line (c). Red lines indicate electrical path to power Auxiliary cooling system

- When the 4160 V buses are energized (through the recovery of the DGs, RSST or 138KV line), the auxiliary cooling system is able to cool the reactor core and, thus, core temperature decreases

Given the uncertainties associated to the recovery timing of DGs, RSST and 138KV line, a probabilistic model has been employed to represent these events. The corresponding probability distribution functions are as follows:

- DGs: a dead time of 100s is required by Team 1 to reach the DGs building and DGs' repair time TDG has a normal distribution having mu = 800 and sigma = 200. This distribution is also truncated such that $0 < TDG1 < 2500$

- RSST: a dead time of 400s is needed to assess the damage at the RSST switch-yard and to organize its recovery. Recovery time for RSST, TRSST , is normally distributed with mu = 1400 and sigma= 400

- 138KV line: the recovery of the main AC line T138 is normally distributed with mu = 2000 and sigma = 500

In addition, the clad failure temperature $T_{C,fail}$, our success/failure criterion, is probabilistic distributed with a triangular distribution with the following characteristics:

- mode(apex): 1477.59 K, 10CFR regulatory limit

- lower bound: 1255.37 K, PRA success criterion

- upper bound: 1699.82 K, Urbanic-Heidrick transition temperature [9]
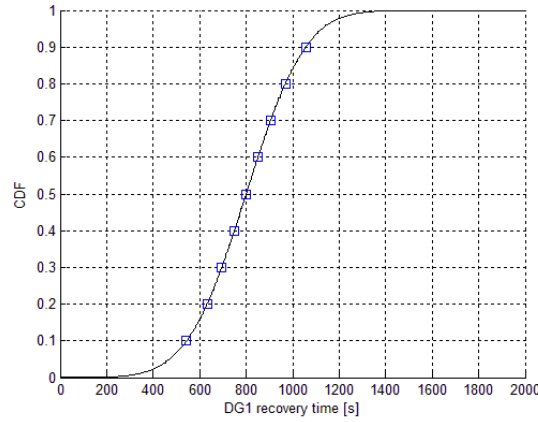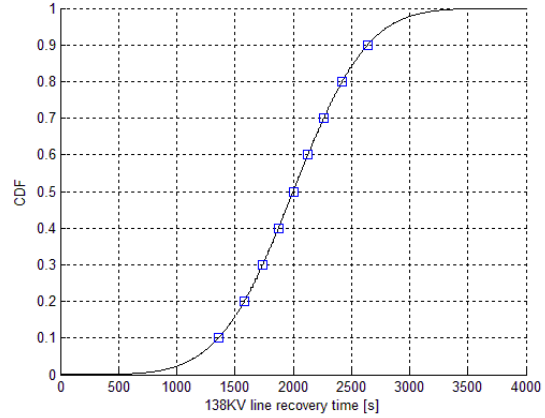
(a)

Figure 8: Cumulative Distribution Functions and DET bins for DGs
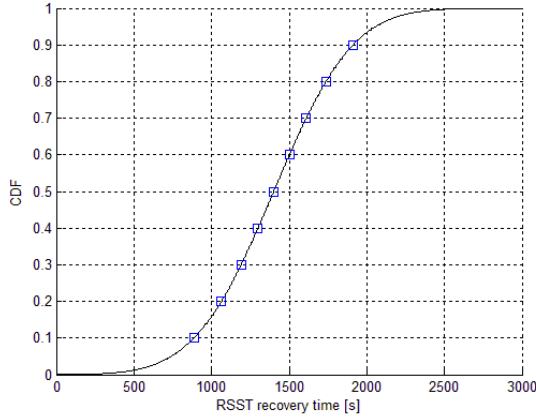
(a)   (b)

Figure 9: Cumulative Distribution Functions and DET bins: (a) 138kV line, (b) RSST

The PRA analysis for this accident scenario has been carried over using the ADET algorithm. The parameters controlling the convergence of the iterative process has been chosen such as the maximum error in terms of the determination of the probability of failure of the system would be below 1.0E-4.

As already mentioned, the ADET methodology performs a pre-screening of the input space through a standard DET approach, before starting the real adaptive scheme. In this specific case, pre-screening DET calculation has been run using equally spaced branching probability thresholds for all uncertain parameters considered in this scenario.

The probability threshold values are reported below:

- DGs' recovery time distribution (Fig. 8 (a) - blue points):

    – Probability Thresholds: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
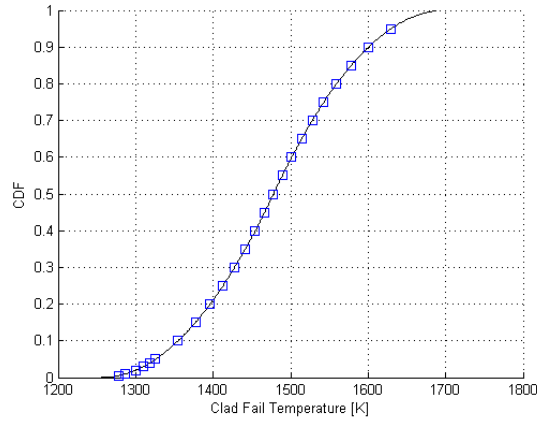
(a)

Figure 10: Cumulative Distribution Function and DET bin for Clad Failure Temperature Distribution

– Variable Values (s) : 543.69 , 631.68, 695.12, 749.33, 800.00, 850.67, 904.88, 968.32, 1056.31

- RSST recovery time distribution (Fig. 9 (a) - blue points):

  – Probability Thresholds: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
  – Variable Values (s) : 887.38, 1063.35, 1190.24, 1298.66, 1400.00, 1501.34, 1609.76, 1736.65, 1912.62

- 138 kV line recovery time distribution (Fig. 9 (b) - blue points):

  – Probability Thresholds: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
  – Variable Values (s) : 1359.22, 1579.19, 1737.80, 1873.33, 2000.00, 2126.67, 2262.20, 2420.81, 2640.78

- Clad Failure Temperature distribution (Fig. 10 - blue points):

  – Probability Thresholds: 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.10, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95
  – Variable Values (K) : 1304.57, 1307.51, 1313.27, 1318.89, 1324.38, 1329.37, 1354.78, 1376.74, 1395.86, 1412.68, 1427.70, 1441.33, 1453.96, 1465.94, 1477.60, 1489.26, 1501.24, 1513.87, 1527.50, 1542.51, 1559.34, 1578.45, 1600.40, 1625.79

In the following section, the simulations results are shown and discussed.

4.2 Results

In order to show the newer capability added in RAVEN code, a standard DET [3, 4] analysis analysis and an Adaptive tree (ADET) have been compared.

Figure 11 shows histograms of the sampled clad failure temperatures for DET and ADET. It can be noticed how the Adaptive DET (right figure) focuses on the higher temperatures, reasonably closer to failure regions. The standard DET (left figure) presents a spike in the histogram for the lowest temperatures. This is due to the intrinsic initial mechanics of the methodology in the exploration of the input space (in this case, no branches on clad failure temperature until the first transition is reached).

As already mentioned, the cooling of the system is ensured when:

- Diesel Generators are repaired, or,

- the 138 kV line is restored, or,

- the RSST switch-yard is repaired.

	Clad Temp Failure	DGS Recovery Time	SecPG Recovery Time	PrimPG Recovery Time	Clad Damaged
Clad Temp Failure	1.00E+00	-5.99E-18	-9.66E-19	-1.31E-17	-2.78E-01
DGS Recovery Time	-5.99E-18	1.00E+00	3.56E-17	3.56E-18	3.20E-01
SecPG Recovery Time	-9.66E-19	3.56E-17	1.00E+00	-2.37E-18	1.60E-02
PrimPG Recovery Time	-1.31E-17	3.56E-18	-2.37E-18	1.00E+00	2.38E-01
Clad Damaged	-2.78E-01	3.20E-01	1.60E-02	2.38E-01	1.00E+00

Table 1: Pearson Correlation Matrix.

Such a situation determines that the timing of these events can be pratically collapsed into a global recovery time of the cooling system. This helps us in the 2-Dimensional visualization of the results (ACS recovery time vs. clad failure temperature). Figure 12 shows the sampled points (translated in collapsed time) for the DET (left picture) and ADET (right figure) methodologies. It can be seen how both methodologies march toward higher recovery times, looking for the first transition that determines the first branch for the Clad Failure temperature. Obviously, the ADET method starts its adaptive strategy in proximity to the clad failure transitions, which is the goal function transition surface.

Figure 13 shows the limit surface obtained by the ADET. As expected, the LS testifies that the higher is the clad failure temperature, the more time we can wait before recovering the auxiliary cooling systems. The LS for the DET is not shown here, as it is extremely similar to the one just seen. Table 1 shows the correlation coefficients' matrix. As is shown in the last column, the failure of the clad is inversely proportional to the clad failure temperature. This means, as expected, increasing its threshold reduces the probability of failure of the system. The correlation coefficients

of the other parameters (timing of recovery of the cooling system) are instead all positive; meaning the probability of failure increases if those parameters increase. Since the transient, subject of this analysis, is quite fast, obviously the DGs recovery time is the most important (mean = 800 seconds) and the recovery of the 138 kV power line is of lesser importance (mean = 2,000 seconds). As for the LS, the DET correlation matrix is not show here, since it does not add any different information.

For both approaches, the probability of failure has been computed. The ADET and DET methods determined a probability of failure of 0.277 ± 0.229 and 0.227 ± 0.402, respectively. Since the Adaptive methodology is much more focused on the failure regions, its sigma is consequently lower.
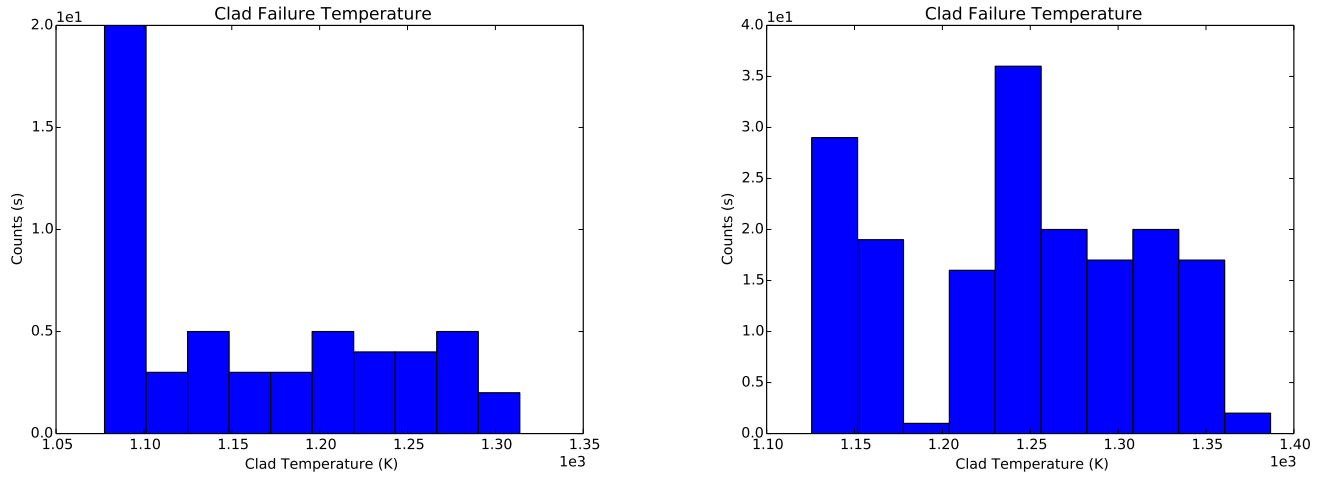

Figure 11: Histogram of Clad Failure Temperature for DET (Left) and ADET (Right)
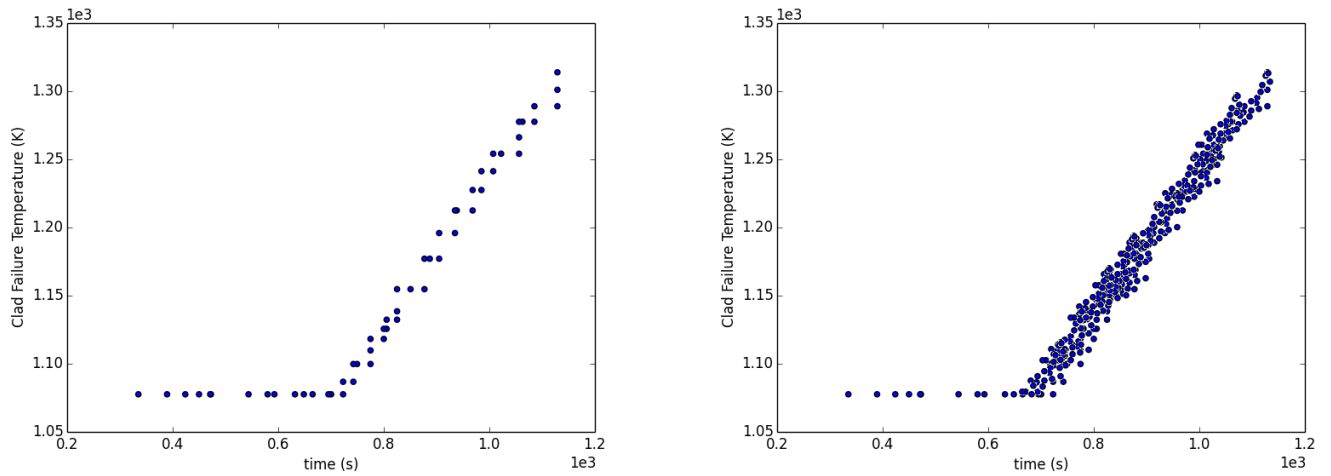

Figure 12: Sampled Parameters (Collapsing Recovery times) for DET (Left) and ADET (Right)
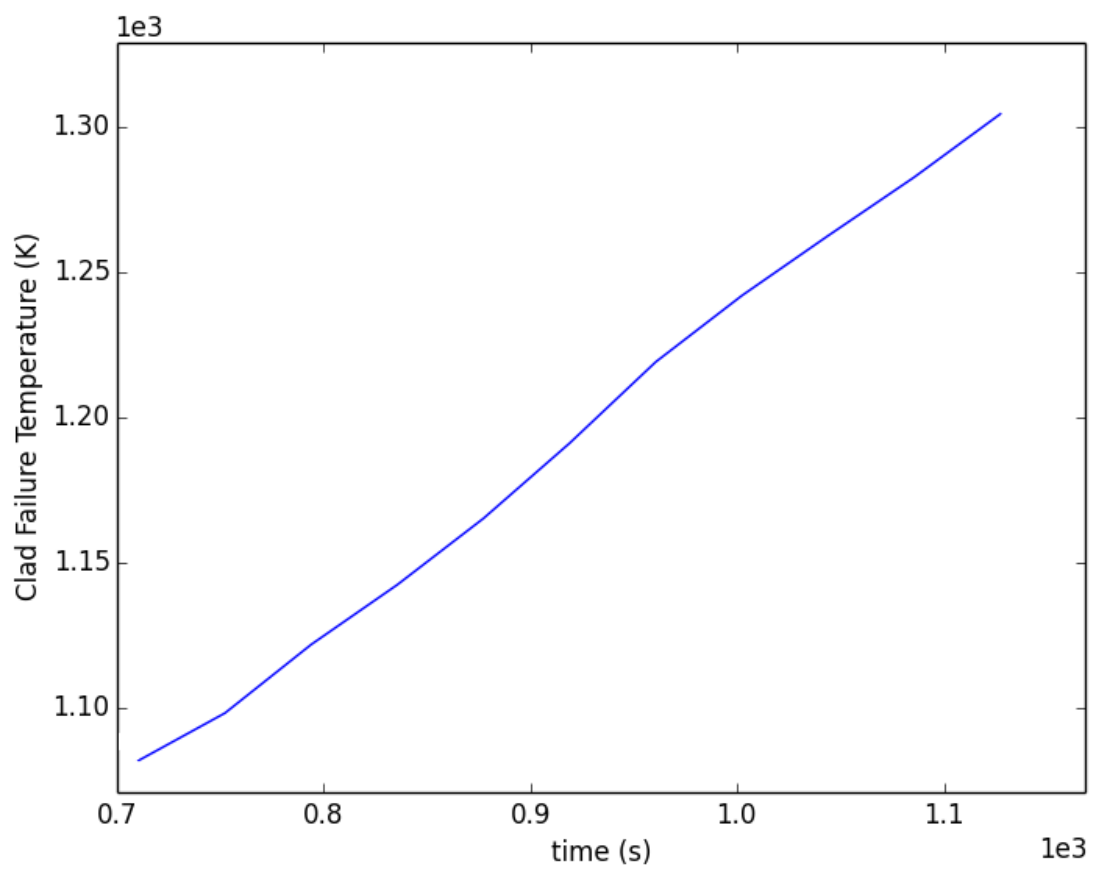
Figure 13: Limit Surface DET/ADET.

5 Conclusions

The milestone of developing the Adaptive Dynamic Event Tree (DET) methodology has been fully achieved. The Adaptive Dynamic Event Tree (ADET) module implemented in RAVEN is fully functional and helps RAVEN to become the next generation PRA simulation tool. The developed approach exploits the intrinsic characteristics of the DET methodology, and incorporates these characteristics into the adaptive LS searach concept. The introduction of adaptivity in the determination of the simulation braches allows to improve the trade off between computational time and accuracy in the determination of the limit surface location in the input space. This report shows how the methodology works and how it can be used in a realistic PRA analysis. The coupling of RAVEN and RELAP-7 has been tested in conjunction with this new feature for a demo of a PWR SBO core damage probabilistic analysis. The results have been compared with standard DET approach and show agreement within the prescribed tolerances.

In the future, a more stringent investigation of the theoretical basis will be performed in order to complete the method development.

References

[1] Neams: The nuclear energy advanced modeling and simulation program. Technical Report ANL/NE-13/5.

[2] Light water reactor sustainability program integrated program plan. Technical Report INL-EXT-11-23452, April 2013.

[3] A. Alfonsi, C. Rabiti, D. Mandelli, J. Cogliati, B. Kinoshita, and A. Naviglio. Dynamic event tree analysis through raven. In *Proceedings of ANS PSA 2013 International Topical Meeting on Probabilistic Safety Assessment and Analysis, Columbia (South Carolina)*, 2013.

[4] A. Alfonsi, C. Rabiti, D. Mandelli, J. Cogliati, and R. Kinoshita. Raven: Dynamic event tree approach level iii milestone. Technical Report INL/EXT-13-30332, INL, September 2013.

[5] David Anders, Ray Berry, Derek Gaston, Richard Martineau, John Peterson, Hongbin Zhang, Haihua Zhao, and Ling Zou. Relap-7 level 2 milestone report: Demonstration of a steady state single phase pwr simulation with relap-7. Technical Report INL/EXT-12-25924, Idaho National Laboratory (INL), 2012.

[6] Pedregosa et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, pages 2825–2830, 2011.

[7] A. Hakobyan, T. Aldemir, R. Denning, S. Dunagan, D. Kunsman, B. Rutt, and U. Catalyurek. Dynamic generation of accident progression event trees. *Nuclear Engineering and Design*, 238(12):3457 – 3467, 2008.

[8] D. Mandelli and C. Smith. Integrating safety assessment methods using the risk informed safety margins characterization (rismc) approach. In *Proceeding of American Nuclear Society (ANS), San Diego (CA)*, volume 107, pages 883–885, 2012.

[9] V. F. Urbanic and T. R. Heidrick. High-temperature oxidation of zircaloy-2 and zircaloy-4 in steam. *Nuclear Materials*, 251(75), 1978.