VTT Technical Research Centre of Finland

Simulation-based probabilistic risk assessment for spent fuel pool

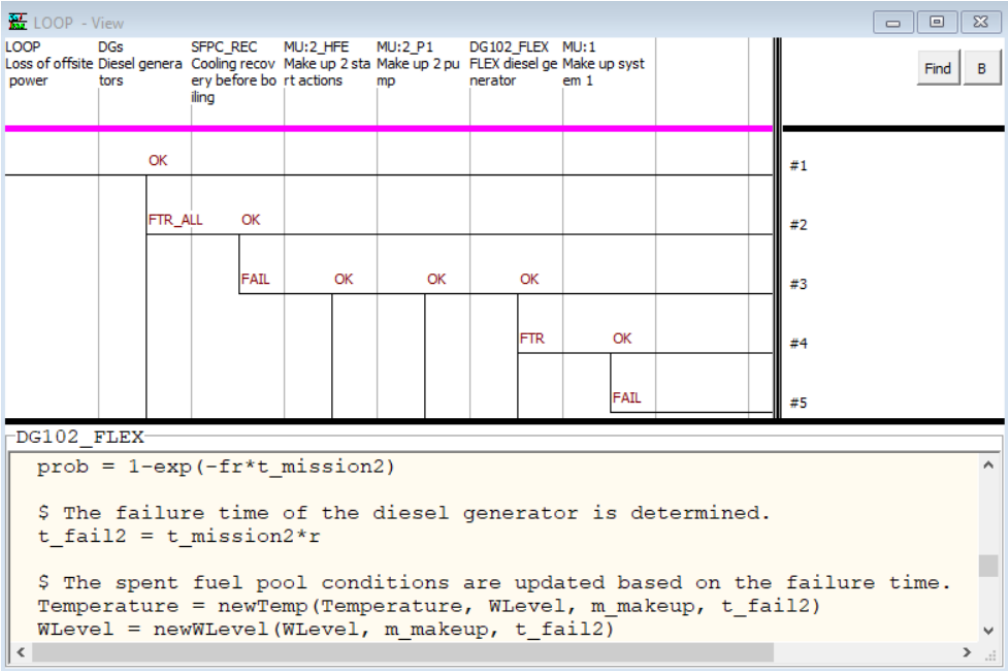
Tyrväinen, Tero; Immonen, Essi

Published: 22/02/2022

Document Version
Publisher's final version

RESEARCH REPORT

VTT-R-00016-22



Simulation-based probabilistic risk assessment for spent fuel pool

Authors: Tero Tyrväinen, Essi Immonen

Confidentiality: VTT Public

beyond the obvious

Report's title Simulation-based probabilistic risk assessment for spent fuel pool	
Customer, contact person, address VYR	**Order reference** SAFIR 3/2021
Project name New developments and applications of PRA	**Project number/Short name** 128648/NAPRA
Author(s) Tero Tyrväinen, Essi Immonen	**Pages** 27/20
Keywords Probabilistic risk assessment, spent fuel pool, simulation	**Report identification code** VTT-R-00016-22

Summary

This report presents an approach for simulation-based probabilistic risk assessment (PRA) of a spent fuel pool, and analyses a loss of offsite power scenario using the approach. In the simulation-based event tree, accident timings, such as failure times of components and durations of manual actions, are simulated to analyse time-dependencies. The time windows for probabilistic analysis, namely mission times for safety functions and available times for manual actions, are calculated based on spent fuel pool conditions affected by the timings of previous events. The model combines deterministic and probabilistic analysis; the spent fuel pool conditions are calculated by a simplified, but sufficiently realistic deterministic model.

In this report, the simulation-based model is used to quantify minimal cut sets of a static PRA model more realistically. The results of dynamic and static analyses of a loss of offsite power scenario are compared. The dynamic analysis decreases the frequencies of minimal cut sets significantly. The decrease is particularly related to more realistic definition of mission times and crediting the operation of the cooling/make-up systems before they fail, which gives more time for the subsequent manual actions. The results also indicate that crediting repairs can greatly decrease the frequencies, though those can also be modelled in static manner to some extent.

There are some challenges related to application of the approach for full-scope spent fuel pool PRA. The simulation-based event tree becomes easily very complex when there are many failure combinations to analyse, and there is no good tool support to integrate the minimal cut sets of static PRA and the simulation results. One possibility would be to develop simulation-based event trees as independent PRA model so that there would be no need for static PRA model. However, the minimal cut set information would be lost, and the identification of all relevant failure combinations could be a challenge. Another potential solution would be to develop a simulation module for automatic quantification of minimal cut sets. This would provide more flexibility than a simulation-based event tree and would give wider possibilities to perform advanced minimal cut set quantifications.

Confidentiality	VTT Public

Espoo 22.2.2022

Written by	**Reviewed by**
Tero Tyrväinen Research Scientist	Ilkka Karanta Senior Scientist

VTT's contact address
VTT Technical Research Centre of Finland Ltd, P.O. Box 1000, FI-02044 VTT, FINLAND

Distribution (customer and VTT)

SAFIR2022 RG2 members, VTT archive

Approval

VTT TECHNICAL RESEARCH CENTRE OF FINLAND LTD

Date:	22 February 2022
Signature:	DocuSigned by: Nadezhda Gotcheva E21E683840FD424...
Name:	Nadezhda Gotcheva
Title:	Research Team Leader

Contents

1. Introduction

Current probabilistic risk assessment (PRA) models for nuclear power plants and their spent fuel pools are static. It means that time-dependencies and timings of events are mostly not modelled. In reality, for example the failure time of a safety function can have an impact on how long a back-up safety function needs to function in order to reach a safe state, which means that the failure probability of the back-up safety function can depend on when the primary safety function fails. Instead of modelling that, static PRA models are simplified so that each safety function has a presumably conservative mission time, typically 24 hours in reactor models. When defining the success criteria of a back-up safety function, the primary safety function is usually assumed to fail at start. Such conservative assumptions cause overestimation of the risk.

The mission time of a safety function should be the time it takes to reach a safe state as stated in IAEA's guidelines (IAEA, 2010). However, mission times are usually not estimated based on that (Tyrväinen et al., 2020). Typically, the same mission time is used in all accident sequences, e.g. 24 hours in level 1 PRA of a reactor. The failure probability of a component depends almost linearly on its mission time. Therefore, mission times can be important parameters in PRA as identified in (Tyrväinen et al., 2020), and more accurate risk estimates could be achieved by more realistic modelling.

Another important time window in PRA is available time to perform a manual action, e.g. start a system or repair a component. There are also time-dependencies related to available times. For example, available time to start a back-up safety function may depend on how long the primary safety function operated before it failed. Available times are also usually determined based on conservative assumptions, e.g. that the primary safety function fails at start, instead of modelling the time-dependencies.

A large number of dynamic PRA methods have been developed and studied in scientific literature (Aldemir, 2013). For example, dynamic event trees (Karanki & Dang, 2016) are a method class with capability to represent time-dependencies related to available times and mission times. However, dynamic methods have not been much applied in practical PRA, because they are complex and computationally very demanding.

This report continues research on a simulation-based spent fuel pool PRA approach that was started in (Tyrväinen et al., 2021). In this approach, time-dependencies related to available times and mission times are explicitly analysed by simulations. Failure times and timings of manual actions are drawn from distributions, time-dependent conditions of the spent fuel pool are calculated based on those timings, and the time windows are calculated based on the spent fuel pool conditions, i.e. how long it takes to reach a safe state or fuel damage given specific conditions. This method could be applied in several ways, but in (Tyrväinen et al., 2021), it was used to quantify minimal cut sets (MCSs) of a static PRA model more realistically, and the same approach is followed in this report as it allows detailed comparison with the results of static PRA. The goal is to develop a method that is not too complex or heavy, and maintains the traceability of the results.

In the previous study (Tyrväinen et al., 2021), the deterministic physical model of the spent fuel pool was a very simplified test model. In this study, a more realistic physical model is developed to increase the realism of the analysis in terms of spent fuel pool's thermal behaviour.

The development of the simulation-based event tree model for loss of offsite power scenario of a spent fuel pool is continued in this report. For example, offsite power recovery is added to the model as it was not included in (Tyrväinen et al., 2021). While the simulations started previously from the start of boiling, the new analysis covers also simulation of events before boiling. Comparison between static and dynamic analyses is also performed more in-depth.

Potential improvements to the spent fuel pool PRA approach and software tools are also identified and discussed in this report, including some more general considerations related to modelling of time-dependencies in PRA.

Section 2 presents the main features of simulation-based event trees, and Section 3 describes the simulation-based spent fuel pool modelling approach selected in this study. Physical modelling of spent fuel pools is discussed and the deterministic model developed in this study is described in Section 4. Section 5 presents static PRA analyses and simulation-based PRA analyses for a loss of offsite power scenario. The definition of a safe state in PRA analyses is discussed in Section 6. Software tool development possibilities are discussed in Section 7. Section 8 concludes the study.

2. Simulation-based event trees of FinPSA

PRA software FinPSA (VTT, 2014) includes a module for simulation-based event trees (Tyrväinen et al., 2016; Tyrväinen & Karanta, 2019). The module has been developed for level 2 PRA (containment event trees), but it is, in practise, a general-purpose probabilistic risk analysis tool. The module combines event trees with computation scripts written using FinPSA's own programming language, containment event tree language (CETL). In the script files, the user defines functions that calculate probabilities of event tree branches and possibly other variable values, such as magnitudes of consequences or timings of events. The script files enable use of various modelling approaches, because contents of the scripts are not limited in any way, except that they must conform the CETL syntax.

The model includes a separate script file for each event tree section, for an initial section, and for a common section, which is common to all event trees in the project if there are multiple event trees. A function name is assigned to each event tree branch, and the function has to be defined in the script file of the corresponding event tree section. The function returns the probability of the event tree branch. It is also possible to write other functions that are called e.g. by branch functions. The model can include both global variables and local variables limited for a specific event tree section. Values of global variables can be chronologically updated when moving forward in an event tree sequence and can be utilised in the computation of event tree branch probabilities. For example, a time variable or a physical parameter, such as temperature, can be updated this way according to the events that occur during the sequence. Types of variables are ordinary data types, such as 'real', 'integer', 'Boolean' and 'string'. Probability distributions of a few different types can also be specified. A set of built-in functions is available, including some probability distribution operations.

To account for uncertainties related to variable values, it is possible to specify probability distributions for parameters and perform Monte Carlo simulations. At each simulation cycle, a value is sampled from each specified distribution, and based on that, numerical conditional probabilities are calculated for all event tree branches, and values are calculated for all variables at each end point of the event tree. After the simulations, statistical analyses are performed to calculate frequency/probability and variable value distributions for each end point among other statistical results and correlation analyses. It is also possible just to calculate point values of the event tree based on the mean values of distributions. Event tree sequences can also be grouped by a binner routine, and combined results can be calculated for the specified consequence categories.

The simulation-based event trees of FinPSA provide only the frame for modelling. The tool can be used in many ways, and it is up to the user to select or develop the actual modelling approach for the application.

3. Simulation-based approach for spent fuel pool

The modelling approach selected in (Tyrväinen et al., 2021) for spent fuel pool analysis integrates deterministic spent fuel pool behaviour and probabilistic analysis. The spent fuel pool water level and temperature are calculated in the simulations at every time point of interest, e.g. when a make-up system is started or fails. The time windows for probabilistic analysis are dynamically calculated based on the current spent fuel pool conditions. For example, the mission time of a make-up system is calculated based

on how long it takes to reach the safe state, i.e. the water level is normal and the spent fuel pool cooling system is back in operation. Similarly, the time available to start a make-up system is calculated based on how long it takes until the water level has decreased to the fuel level.

In the simulations, durations of manual actions are drawn from specified probability distributions to determine e.g. when a make-up system is started or when a diesel generator is repaired. Failure times of components are also drawn from uniform distributions covering the mission times of the components. Assuming that the failure times are exponentially distributed and the failure rates are small, the uniform distribution gives a good approximation.

Even with the abovementioned specifications, the model could be constructed in several different ways and with different scopes. Here, we follow the approach that was selected in (Tyrväinen et al., 2021), i.e. the simulation-based event trees are used to quantify the most important minimal cut sets of a static PRA model developed in (Tyrväinen et al., 2021). The simulation-based event trees are constructed so that each of the top minimal cut sets of the static model corresponds to a sequence of the simulation-based event tree. Therefore, most of the branches of the simulation-based event tree correspond to basic events of the static model. The order of events in the tree follows the accident chronology. For example, the event tree for loss of offsite power is presented in Figure 1. There are also some branches that do not originate from basic events and represent events that were not credited in the static model. The construction of the event tree can be somewhat case specific, but the basic principles are that relevant basic events (with dynamic behaviour) from minimal cut sets need to be included as branches and the accident chronology needs to be followed. It is possible to model some basic events (from different minimal cut sets) using the same branch if the simulation model is the same for those basic events (typically, this involves some simplifications).

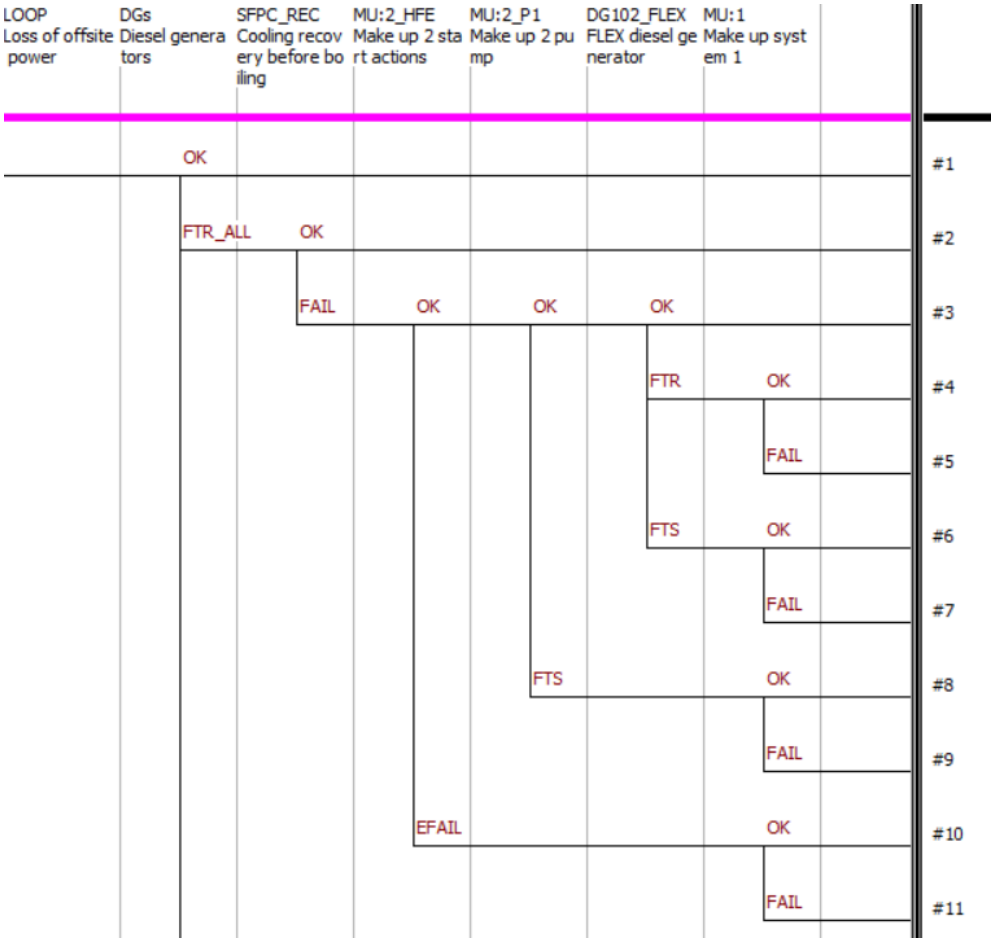


Figure 1: Upper part of the simulation-based event tree for loss of offsite power.

The idea is that the results of the simulations can be used to update the frequencies of the minimal cut sets of the static model to calculate the fuel damage frequency more realistically. A benefit of this approach is that the minimal cut set information is preserved, which is important for the traceability of the results. The approach is also convenient for the comparison of dynamic and static analyses. Possibilities to develop the approach further are discussed later in Section 7.

The computation scripts related to the event tree are presented in Appendix B. Here, we present a few illustrative examples. For example, function OK in the MU:2_HFE section is defined in the following way:

```
function nil OK
  $ Time available to start make-up system 2.
  t_avail = t_uncover(WLevel)

  t_avail2 = t_avail $ Collect to results

  $ The execution time of the make-up system 2 start is drawn from uniform distribution.
  t_exe = 2*r2+1

  $ Is there time to make the execution?
  if t_exe < t_avail then
  begin
    $ The diagnosis time of the make-up system 2 start is drawn from lognormal distribution.
    r = r*cumul(MU2D,t_avail-t_exe)
    t_diag = icumul(MU2D,r)

    $ The start time of make-up system 2.
    t_start2 = t_diag+t_exe

    $ The spent fuel pool water level is updated.
    WLevel = newWLevel(WLevel, 0, t_start2)
  end
return nil
```

This function determines the start time of make-up system 2, and updates the spent fuel pool water level and temperature based on how long the manual actions to start the system last. It is a nil function, which means that the probability of the corresponding event tree branch is calculated as the complement of the probability of the other branch. Functions t_uncover and newWLevel, as well as other functions related to spent fuel pool conditions are defined in the common section. The models for spent fuel pool water level and temperature are discussed in the next section.

The failure to run probability of a diesel generator, which serves make-up system 2, is calculated by the following function:

```
function real FTR
  fr = FR_DG   $ Failure rate

  $ The mission time is tentatively calculated as the time to reach the normal water level.
  t_mission2 = t_restore(WLevel)

  $ Time when the spent fuel pool cooling can theoretically be recovered.
  t_rec = t_repair-t_start2

  $ Does the recovery take longer than reaching the normal water level.
  if t_mission2 < t_rec then
  begin
    $ Given the recovery time of the spent fuel pool cooling,
    $ the earliest allowed failure time is calculated.
    t_earliest = EarliestTime(Temperature,t_rec,WLevel)

    $ If the earliest allowed failure time based on the recovery of the spent fuel pool cooling
    $ is larger than the time to reach the normal water level, the mission time is
    $ determined based on that.
    if t_mission2 < t_earliest then t_mission2 = t_earliest
  end

  $ The diesel generator failure probability is calculated.
  prob = 1-exp(-fr*t_mission2)

  $ The failure time of the diesel generator is determined.
  t_fail2 = t_mission2*r
```

```
  $ The spent fuel pool conditions are updated based on the failure time.
  Temperature = newTemp(Temperature, WLevel, m_makeup, t_fail2)
  WLevel = newWLevel(WLevel, m_makeup, t_fail2)
  if WLevel > InitWLevel then WLevel = InitWLevel

  $ The total time the make-up 2 system was used.
  t_mu2 = t_start2+t_fail2

  $ Mean time to repair for repair modelling of this diesel generator.
  mttr1 = MTTR_DG_FTR
return prob
```

The function determines the mission time for the diesel generator based on the time to reach the normal water level and recovery time of the power supply for the spent fuel pool cooling system. The diesel generator is allowed to fail some time before the recovery of the spent fuel pool cooling as long as the boiling does not start again before the repair. The earliest allowed failure time is calculated using the EarliestTime function, which is defined in the common section. A failure time is also drawn for the diesel generator on each simulation cycle, and the water level and temperature are updated taking into account how long make-up system 2 operated. These water level and temperature conditions affect later in the analysis the available time to start make-up system 1.

On each simulation cycle, a conditional probability for each sequence of the event tree is calculated given specific human action, failure and repair timings. Then, average probabilities are calculated for the sequences over the simulation cycles. These average probabilities are not conditional to specific timings, but reflect complete probability distributions of different timing variables. The accuracy of these probabilities depends on the number of simulation cycles, which should be sufficiently large.

4. Spent fuel pool physics

Modelling the thermal behaviour of spent fuel pools is useful in order to understand the behaviour of the pool in accident conditions. Studying the heat and mass transfer provides information on the pool water temperature and the level of the water as a function of time, which helps to determine the consequences of the accident, such as uncovering of the fuel.

4.1 Modelling approaches to spent fuel pool behaviour

Spent fuel pool accidents are typically loss of cooling or loss of water inventory accidents (Adorni et al., 2015). As a consequence to the loss of cooling or loss of water inventory, the fuel heats and degrades, which includes phenomena such as oxidation and hydrogen generation, burning of zirconium cladding, radioactive releases and criticality. The spent fuel accidents are typically modelled with different codes for each phenomenon: thermal-hydraulics behaviour, criticality, fuel behaviour and degradation and fission product release (Adorni et al., 2015).

The thermal-hydraulic behaviour of spent fuel interim storage pools is typically modelled using one of two different approaches. The approaches are divided to system codes, such as RELAP, TRACE, MELCOR or APROS and to computational fluid dynamics (CFD) analyses (Ramadan et al., 2018). System codes have been used for example by Tynys (2017) to study different accident scenarios of a spent fuel pool design and by Kaliatka et al. (2010) to study the spent fuel pool processes in case of loss of heat removal. Carlos et al. (2014) use TRACE, a best estimate code, to analyse Maine Yankee spent fuel pool safety in both steady-state and transient conditions and compared the results to measurement data. Besides pool water level and temperature, they also studied the oxidation and peak temperature of the cladding and hydrogen generation. System codes solve mass, momentum and energy conservation equations in one-dimensional form. They are typically used in accident scenario analyses to recognize safety issues in the design. The challenge of system codes is the modelling of multidimensional phenomena, which cannot be sufficiently approximated.

CFD codes, the second approach, are more typically used to the pond design, modelling the fluid flow and heat transfer scenarios in three dimensions. For example, Hung et al. (2013) use a three-dimensional CFD modelling approach to study the cooling ability of a spent fuel pool in removing decay heat. The advantage of the CFD codes is the level of detail that the 2D and 3D computations provide. On the other hand, CFD codes have challenges when modelling larger sizes of cooling ponds due to heavy computations and complexity of the phenomena like evaporation (Ramadan et al., 2018).

Computationally lighter modelling approach is presented by Ramadan et al. (2018). The authors have developed a zero-dimensional model based on the well-mixed approach that has lower computational time compared to for example CFD codes. The model however gives reasonable results that are accurate enough when performing quicker analyses. The zero-dimensional model divides the spent fuel cooling system to the water and humid air zones, and the environment. Both the water and air zones are considered to have uniformly distributed temperature, so there is no spatial dependency, but only time dependency. The physical conditions of the water and air zones are updated at each time step and the temperature and mass of water and air are calculated. Yanagi and Murase (2013) provide another example of a simple one-region model that predicts water level and temperature during loss of all AC power supplies. The one-region model is compared with a three-dimensional CFD code. The results were quite similar, which supported the validity of the results of the computationally much lighter one-region code.

4.2 Spent fuel pool behaviour model

In the simulation-based FinPSA model, the deterministic calculations are integrated to the PRA-model by updating the spent fuel conditions in the scripts. The water level and temperature are updated in certain parts of the event tree. Other necessary information are the moment when the pool water starts to boil, the moment when the top of the fuel is uncovered and the time it takes to reach normal pool water level when the make-up system is in use. These five functions were implemented in the FinPSA CETL programming language with more accurate assumptions than previously (Tyrväinen et al., 2021). A model describing the spent fuel pool temperature and water level was also implemented using Matlab, as it provides some advantages compared with FinPSA, such as better visualisation and debugging. The results of the FinPSA and Matlab models were compared to ensure that the FinPSA implementation is correct.

The modelling approach was chosen based on a survey of literature the results of which are described in section 4.1. The zero-dimensional model of Ramadan et al. (2018) was seen simple but realistic enough to be implemented in simulation-based FinPSA scripts. Some simplifications were made from the original approach, such as assuming the spent fuel building pressure and air temperature constant, since they did not have a significant effect on the results. The modelling approach is based on solving the spent fuel pool water mass and energy balances. The ordinary differential equations are solved numerically using Euler's approach.

The mass balance of the water depends on the amount of make-up water, the loss from evaporation of water and the outflow of water. No leakage is assumed in the modelled scenarios. The equation of the mass balance can be written as

$$m_p^{n+1} = m_p^n + (\dot{m}_m - \dot{m}_o - \dot{m}_{ev})^n \Delta t,$$

where the terms are the change due to make-up water $\dot{m}_m$, outflow of water $\dot{m}_o$ and evaporation $\dot{m}_{ev}$, Δt is the time step and n the number of iterations. The energy balance equation is written as

$$T_p^{n+1} = T_p^n + \left(\dot{Q}_d + \dot{m}_m C_w (T_m - T_p) - \dot{m}_o C_w T_p - \dot{m}_{ev} C_w T_p - \dot{Q}_s\right)^n \frac{\Delta t}{m_p C_w},$$

where the temperature change depends on the decay heat of the fuel elements $\dot{Q}_d$, the cooling effect of the make-up water $\dot{m}_m$, the outflow of water $\dot{m}_o$, evaporation $\dot{m}_{ev}$ and the total heat transfer $\dot{Q}_s$ at the air-water interface due to the three heat transfer modes: radiation, convection and evaporation. The heat loss to the concrete structures of the pool is excluded in this approach, as it accounts for a very small part of the total heat loss (Ramadan et al., 2018).

The amount of outflow is assumed to be zero while the water level is below the design value. When the design value is reached with the make-up water, water is removed from the pool to keep the water level constant. Then, the water outflow is the difference of the make-up water and the water loss due to evaporation,

$$\dot{m}_o = \dot{m}_m - \dot{m}_{ev}.$$

The water temperature and level are plotted in Figure 2 and **Error! Reference source not found.**, where the make-up system is started at the moment the fuel level is reached, which is approximately at 6.3 days. When the normal water level is reached, the water outflow is started, which accelerates the decrease of the pool temperature.

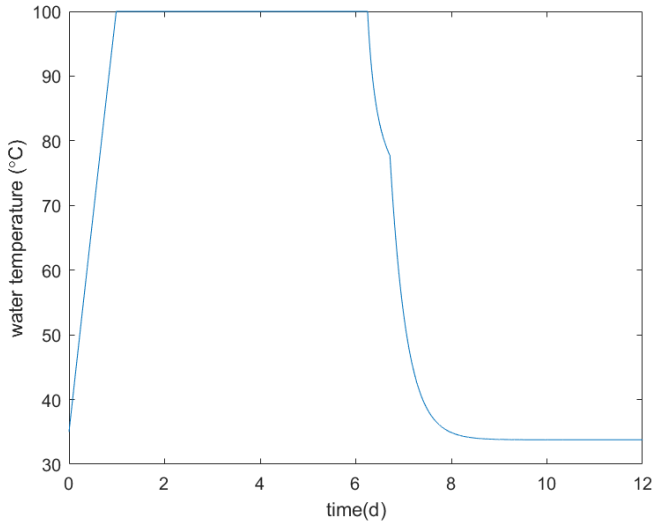


Figure 3: Water temperature when make-up system is started at t = 6.3 d.

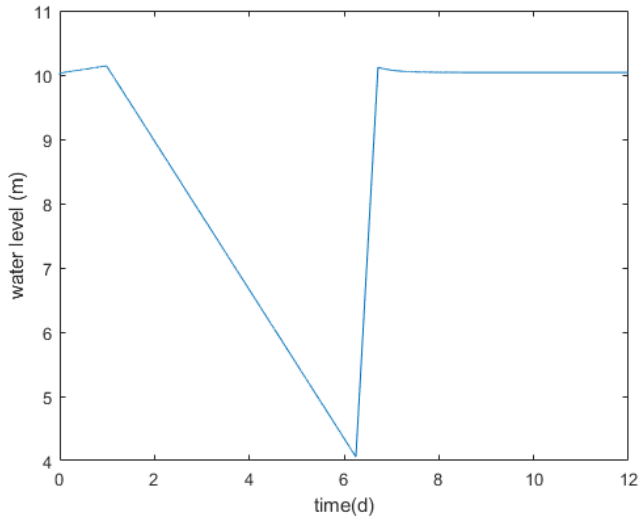


Figure 2: Water level when make-up system is started at t = 6.3 d.

The approach by Ramadan et al. (2018) was developed for large-scale cooling ponds, while in this work, the modelled spent fuel pool was significantly smaller in dimensions and in the amount of the stored spent fuel. The spent fuel decay heat was set to 4 MW, and the pool surface area to 140 m^2. Pool water depth in normal conditions was set to 10 m and the top of the fuel assemblies to 4 m measured from the bottom of the pool. These input parameters resulted in timeframes of boiling and evaporation of the pool water, that were similar to the time windows used in the previous work (Tyrväinen et al., 2021) of this task, and thus made the comparison of the results more reasonable. With the chosen parameters, it takes 23.7 hours from the normal temperature to boiling, and 127 hours from the normal water level to the top of the fuel level after the boiling starts.

The equations of the physical model are presented in more detail in the FinPSA scripts in Appendix B. Some uncertainties are still related to the model. The significance of heat losses to the pool structures, the modelling of the evaporation phenomena and the effect of the spent fuel pool room vapour behaviour are potential topics for further development.

4.3 FinPSA scripts

In FinPSA, the physical behaviour of the spent fuel pool is updated in the event tree scripts by calling functions, that

- update the temperature of the water based on a given time delay, with or without the operation of a make-up system or the spent fuel pool cooling system,
- update the level of the water based on a given time delay, with or without the operation of a make-up system,
- return the time it takes for the water to boil starting from a certain initial temperature, without the operation of the spent fuel pool cooling system or a make-up system,
- return the time it takes for the top of the fuel to uncover starting from a certain initial water level, without the operation of a make-up system and
- return the time it takes to reach the designed water level using a make-up system, starting from a certain initial water level.

For example, the water level is updated as follows:

```
$ Function that returns water level after specified time.
$ IWL = initial water level
$ t = time delay
$ MW = amount of make-up water, set to 0 if the make-up system is not used
function real newWLevel(real IWL, MW, t)
real Mass, WL, m_ev, Lc, hm
   Mass = ((IWL-FuelLevel)*As + Ar*FuelLevel)*rho    $ mass of water above fuel + between fuel racks

   Lc = As/(2*(x+y))                $ characteristic length
   hm = Sh*Dab/Lc                   $ mass transfer coef
   m_ev = As*hm*(rhovs-rhovinf)     $ evaporation rate

   if (MW == 0) then m_ev = Q_d/hfg   $ if make-up system is not used, the water is boiling. Evaporation
rate is changed.
   $update water level
   Mass = Mass +(MW - m_ev)*t*3600

   WL = (Mass/rho - Ar*FuelLevel)/As + FuelLevel  $ water level

return WL
```

The rest of the functions are presented in the Appendix B.

5. Event tree models

In this section, the frequency of fuel damage in a loss of offsite power (LOOP) scenario is analysed. The basis for the analysis is a static event tree model developed in the PROSAFE project (Tyrväinen et al., 2020; 2021). The static model is presented in subsection 5.1. The simulation-based event tree model is respectively presented in subsection 5.2, and the results are presented in subsection 5.3.

5.1 Static PRA model

The event tree for spent fuel pool LOOP is presented in Figure 4. The spent fuel pool cooling system is normally powered by offsite power. When the LOOP occurs, the cooling system can be powered by a gas turbine or emergency diesel generators. The spent fuel pool cooling system can be used only when the spent fuel pool water level is normal, because it is not possible to circulate the water when the water level is lower. During boiling, the accident scenario can be managed by two make-up systems that can pump water to the pool. Make-up system 1 uses the same power supply system as the spent fuel pool cooling system, but make-up system 2 is powered by a FLEX diesel generator. If the spent fuel pool cooling system and the make-up systems fail, the result is a fuel damage.

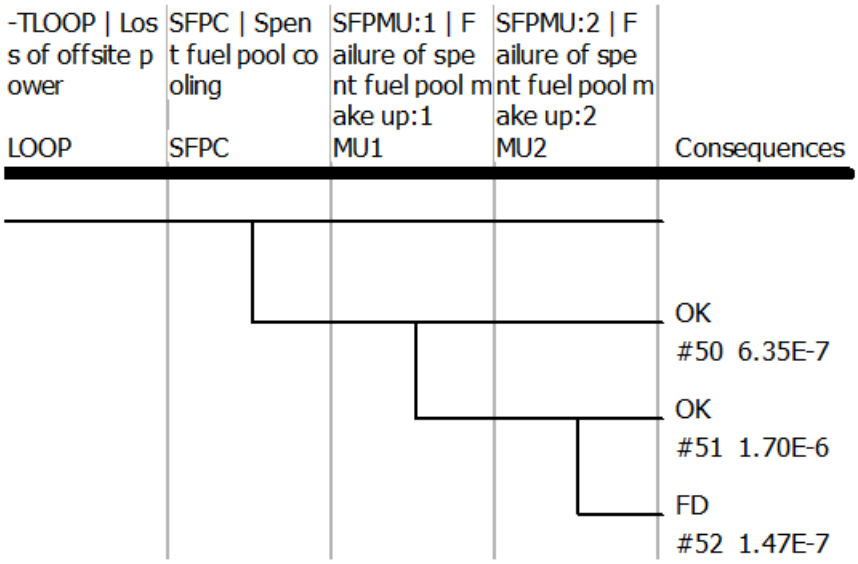


Figure 4: Event tree for loss of offsite power.

The spent fuel pool cooling system consist of four trains. It is enough to use one train at a time. Each train has its own pump and heat exchanger. There is also one emergency diesel generator for each train. The fault tree model of the system is very simplified and does not include any valves. Switching of the spent fuel pool cooling train has to be done manually. The failure of the switching is modelled with two basic events, diagnosis failure and execution failure, which are included in the fault tree of the system.

Make-up system 1 consists of two trains. It is enough to use one train at a time. Each train has its own pump. The power is supplied from the same power supply trains 3 and 4 that the spent fuel pool cooling system trains 3 and 4 use. The system has to be started manually, and the failure of the start action is modelled using diagnosis failure and execution failure basic events.

Make-up system 2 has only one train including a pump. The pump is powered only by a FLEX diesel generator. The system has to be started manually, and the failure of the start action is modelled using diagnosis failure and execution failure basic events.

The 15 most important minimal cut sets are presented in Table 1 and a longer list is presented in Appendix A. In all top minimal cut sets for the fuel damage, the spent fuel pool cooling fails mainly due to failures of the gas turbine (ACN10GT001) and emergency diesel generators (ACP-DG). The most important failure modes are failure to start for the gas turbine and common cause failure (CCF) to run for the diesel generators, but also other failure modes are present in the minimal cut sets. There are also some important minimal cut sets with a failure to start a single spent fuel pool cooling pump (SFPC-P) combined with gas turbine and diesel generator failures. Make-up system 1 is not present in the top minimal cut sets, because it fails when the power supply to the spent fuel pool cooling system fails. For make-up system 2, several different failure modes appear in the top minimal cut sets.

Table 1: Top minimal cut sets.

Mc_num	Freq	Basic event names				
1	2.36E-08	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-ALL	SFPMU:2_P1_____A	
2	2.32E-08	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-ALL	ACP__DG102_FLEX2___D	
3	1.09E-08	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-ALL	SFPMU:2_P1_____A	
4	1.07E-08	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-ALL	ACP__DG102_FLEX2___D	
5	2.99E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-ALL	SFPMU:2_MANSTART___H	
6	2.74E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------A-ALL	SFPMU:2_P1_____A	
7	2.70E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------A-ALL	ACP__DG102_FLEX2___D	
8	2.70E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-ALL	ACP__DG102_FLEX2___A	
9	2.03E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AC	SFPC_P2_____A	SFPMU:2_P1_____A
10	2.00E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AD	ACP10DG001_____D	SFPMU:2_P1_____A
11	2.00E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AA	ACP40DG001_____D	SFPMU:2_P1_____A
12	2.00E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AB	ACP30DG001_____D	SFPMU:2_P1_____A
13	2.00E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AC	ACP20DG001_____D	SFPMU:2_P1_____A
14	2.00E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AC	ACP__DG102_FLEX2___D	SFPC_P2_____A
15	1.97E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AB	ACP30DG001_____D	ACP__DG102_FLEX2___D

5.2 Simulation-based event tree

The simulation-based event tree for the LOOP scenario is partly presented in Figure 1. For diesel generators, FTR_ALL means failure to run CCF of all diesel generators. There are also three other “failure mode” branches for the diesel generators (failure to start CCF of all diesel generators; failure to run CCF of three diesel generators in combination with a pump failure to start; failure to run CCF of three diesel generators in combination with an independent failure to run of a diesel generator) outside the figure. For those branches, the following tree structure is identical to the FTR_ALL case. After the diesel generators, the possibility to recover the spent fuel pool cooling system before boiling is modelled. After that there are several branches related to different failures of make-up system 2. In the final section, make-up system 1 is modelled.

The event tree has been constructed so that for each top minimal cut set, there is a sequence corresponding to it. One sequence typically covers multiple minimal cut sets as the dynamic analysis is identical for some minimal cut sets. This way the frequencies of the minimal cut sets can easily be updated based on the simulation results. The event tree has been built to cover 32 most important minimal cut sets,

though it also covers many other similar minimal cut sets. The event tree could well be extended to cover more minimal cut sets, but this coverage is considered sufficient for the research purposes.

Compared with the previous report (Tyrväinen et al., 2021), where this scenario was already modelled, the main additions are the modelling of offsite power recovery and emergency diesel generator failures, and more realistic physical model for the spent fuel pool. Different modelling aspects are discussed in the following sections.

5.2.1 Recovery of the spent fuel pool cooling

The recovery time of the offsite power is assumed lognormally distributed. If the offsite power is recovered before boiling of the spent fuel pool, a safe state is assumed. The time to boiling can depend on whether the spent fuel pool cooling system can be operated with a diesel generator and how long. On the other hand, during boiling, recovered offsite power can power make-up system 1. Then, a safe state is assumed if make-up system 1 is able to bring the water level back to normal.

Repair of an emergency diesel generator is also modelled as in (Tyrväinen et al., 2021). A safe state is assumed if the repair is performed before boiling and the corresponding pump starts. If the pump fails to start, repair of another emergency diesel generator is assumed (conservatively starting from the boiling). If a diesel generator of train 3 or 4 is repaired, it can power make-up system 1 during boiling. It is assumed that the repaired diesel generator is always either of those. If make-up system 1 is able to bring the water level back to normal, a safe state is assumed.

5.2.2 Common cause failure of diesel generators

In the modelling of failure to run CCFs of diesel generators, it is taken into account that the spent fuel pool cooling system can be operated some time before the diesel generators fail. This can buy time for the offsite power recovery. First, the mission time of a diesel generator is determined based on the offsite power recovery time. A diesel generator needs to operate long enough so that the pool will not start to boil before the offsite power recovery, i.e. it is allowed to fail some time before the recovery. The CCF probability (probability of the branch) is calculated based on the mission time. This dynamic mission time modelling is one aspect that brings realism compared to the static model. Second, the failure time is drawn for the emergency diesel generators from a uniform distribution covering the mission time in the diesel generator CCF branches of the event tree. It is conservatively assumed that the diesel generator of the active train starts to operate immediately and operates until the drawn failure time, and the diesel generators of the other trains do not operate any time. The time it takes to switch the spent fuel pool cooling train is modelled. The switching is assumed to occur after the failure of the active train, but it is conservatively assumed that the other diesel generators fail at start. The repair process of a diesel generator is assumed to start after the switching actions.

A special case is a scenario, where one of the diesel generators fails independently. The mission time for this diesel generator is determined based on the spent fuel pool conditions after the switching action (and at that point, there is also shorter time until the offsite power recovery), the failure probability is calculated based on the mission time, the failure time is drawn based on the mission time, and the spent fuel pool conditions are updated based on the operating time.

5.2.3 Make-up system 2

Actions to start make-up system 2 are assumed to start when the boiling starts. First, the conditions of the spent fuel pool are updated (water level decreases) based on how long the start actions take. Second, different possible failures of the system are modelled in separate branches. If the failure is failure to start or execution failure, the on-demand probability is directly assigned to the branch.

To model the failure to run of the FLEX diesel generator, the mission time is first determined based on the spent fuel pool conditions, the recovery time of the offsite power and the repair time of the diesel generator (either recovery or repair is needed). The mission time is at least the time to reach the normal water level, but if the recovery of the offsite power and the repair of the diesel generator last longer, the system needs to operate long enough so that the pool will not start to boil anymore before the recovery or repair. The failure probability is calculated based on the mission time, and a failure time is drawn based on the mission time. Finally, the spent fuel pool conditions are updated based on how long the system operated.

5.2.4 Make-up system 1

Make-up system 1 is modelled in the final section of the event tree. A simplifying assumption is taken that make-up system 1 can be used only after make-up system 2 has failed, because it becomes available only after the recovery of the offsite power or repair of a diesel generator. All the failure modes of the system are modelled in the same event tree branch. Because make-up system 1 is not credited in the minimal cut sets of the static model (even though it can be used after the recovery of the offsite power or the repair of a diesel generator), all the failure modes need to be taken into account in the simulation-based quantification of a single minimal cut sets. It is therefore most convenient to model all in the same branch.

The actions to start the system are assumed to start when make-up system 2 fails. The start of the system requires also either the offsite power recovery or a repair of a diesel generator. If the start actions or the recovery and repair last longer than the time for the water level to reach the fuel level, fuel damage is assumed. If the recovery or repair comes before boiling starts again (if make-up system 2 has operated some time), a safe state is assumed. Otherwise, different failure modes of make-up system 1 are modelled. The failure modes are diagnosis failure, failure to execute the human action, failure to start and failure to run. The mission time of the system is the time to reach the normal water level.

Repair of the system is also modelled after failure if applicable. The repair time distribution depends on the failed component. Failure to repair the system before reaching the fuel level is assumed to lead to fuel damage. Failure to start and failure to run after the repair are also assumed to lead to fuel damage.

5.2.5 Manual actions

The simulations require probability distributions for the durations of human actions, which are information not generally available. For diagnosis actions, a lognormal distribution is assumed with a mean of two hours. The error factor for a diagnosis action is estimated based on the human reliability analysis (HRA) results of the PROSAFE project (Tyrväinen et al., 2021) so that the probability to exceed the available time used in HRA is the human error probability estimated in HRA. The probability distributions are presented in

Table 2. For the executions of the start actions, uniform distributions are used and the durations are assumed quite short, regardless if the actions are successful or not. The duration distribution for switching the spent fuel pool cooling system train covers all three switching actions as there are three standby trains (switching actions for different trains are not modelled separately). It is also assumed that the diagnosis to switch the train starts at the beginning of a shift. A delay that is uniformly distributed between 0 and 8 hours is therefore used, because the length of a shift is 8 hours.

Table 2: Probability distributions for the durations of human actions.

Action	Distribution	Parameters
Spent fuel pool cooling system train switching diagnosis	Lognormal	Mean = 2h, Error factor = 3.04
Spent fuel pool cooling system train switching execution	Uniform	Min = 0.5h, Max = 1.5h
Make-up system 1 start diagnosis	Lognormal	Mean = 2h, Error factor = 7.02
Make-up system 1 start execution	Uniform	Min = 0.5h, Max = 1.5h
Make-up system 2 start diagnosis	Lognormal	Mean = 2h, Error factor = 8.29
Make-up system 2 start execution	Uniform	Min = 1h, Max = 3h

Repair actions are similarly divided into diagnosis and execution parts. For repair diagnosis, similar approach is used as for other diagnosis actions, i.e. the mean values are assumptions and the error factors are based on HRA results. For repair execution, an exponential distribution is used with mean time to repair (MTTR) parameter values from (Tyrväinen et al., 2021). The distributions are presented in Table 3.

Table 3: Probability distributions for the durations of repair actions.

Action	Distribution	Parameters
Diagnosis for main diesel generator repair	Lognormal	Mean = 3h, Error factor = 20.36
Diagnosis for make-up system 1 repair	Lognormal	Mean = 2h, Error factor = 10.09
Repair execution for a pump that failed to start	Exponential	Mean = 12h
Repair execution for a pump that failed to run	Exponential	Mean = 24h
Repair execution for a diesel generator that failed to start	Exponential	Mean = 6h
Repair execution for a diesel generator that failed to run	Exponential	Mean = 10h

Offsite power recovery time is assumed lognormally distributed with a mean of 5 hours and an error factor of 10. Lognormal distributions have previously been fitted to offsite power recovery time data by Johnson & Ma (2019), but these parameters are just assumed for this study. To facilitate the comparison with the static PRA model, it is assumed that the LOOP frequency in the static model is the frequency of LOOP events that last at least 4 hours, as short LOOP events are easy to manage. This means that only the part of the lognormal distribution that exceeds 4 hours is used in the simulations.

5.3 Results

The model was simulated 100000 times both with and without make-up system repair. One set of simulations (100000 cycles) lasted about an hour when the time step of the physical model was set to 200 seconds. The simulation results (probabilities of the sequences) were then imported to an Excel tool, which

updated the frequencies of the minimal cut sets by replacing the probabilities of relevant basic events by the probabilities obtained from simulations.

Table 4 presents the fuel damage frequencies of the top minimal cut sets presented in Table 1. The total frequencies have been calculated based on the 32 minimal cut sets presented in Appendix A, even though only 15 minimal cut sets are covered in this table. Complete results are presented in Appendix A. The Seq column shows the number of the corresponding sequence in the simulation-based event tree for each MCS. The “static results” are obtained directly from the static PRA model, and the “dynamic results” have been calculated using the simulation-based event tree. The simulation-based event tree has been analysed with and without make-up (MU) system repair. The “refined static results” have been calculated by making some static refinements to the results obtained from the static PRA model. This will be explained later.

Table 4: Fuel damage frequencies (1/year) of top minimal cut sets.

MCS	Seq	Static	Dynamic without MU repair	Dynamic with MU repair	Refined static
Total		1.14E-07	8.66E-11	4.43E-12	3.39E-10
1	9	2.36E-08	3.02E-11	1.62E-12	7.25E-11
2	5	2.32E-08	2.68E-12	1.15E-13	7.13E-11
3	9	1.09E-08	1.40E-11	7.50E-13	3.35E-11
4	5	1.07E-08	1.23E-12	5.33E-14	3.29E-11
5	11	2.99E-09	3.85E-12	2.59E-13	9.19E-12
6	19	2.74E-09	2.13E-12	1.02E-13	4.60E-12
7	15	2.70E-09	6.50E-14	2.77E-15	4.53E-12
8	7	2.70E-09	3.32E-12	1.78E-13	8.30E-12
9	29	2.03E-09	3.17E-12	1.62E-13	6.24E-12
10	39	2.00E-09	3.32E-12	1.41E-13	6.15E-12
11	39	2.00E-09	3.32E-12	1.41E-13	6.15E-12
12	39	2.00E-09	3.32E-12	1.41E-13	6.15E-12
13	39	2.00E-09	3.32E-12	1.41E-13	6.15E-12
14	25	2.00E-09	1.03E-13	4.67E-15	6.15E-12
15	35	1.97E-09	9.76E-14	4.03E-15	6.06E-12

The simulation-based approach gives much smaller results than the static PRA model. The main reason to that is that offsite power recovery and diesel generator repair have not been modelled in the static PRA model. It is likely that offsite power recovery or diesel generator repair is complete already before boiling and even more likely that the recovery or repair is complete before fuel damage. If the power supply is

recovered before boiling, a safe state is assumed. If the power supply is recovered during boiling, make-up system 1 can still be used, which is not credited in the static PRA model. The static PRA model is therefore very conservative.

To make the results from the static PRA model more comparable to the dynamic analysis, they have been refined to credit the offsite power recovery and diesel generator repair. The frequency of each minimal cut set has been multiplied by the probability that the offsite power recovery before boiling fails, the probability that diesel generator repair before boiling fails, and the probability that power supply is not recovered before fuel damage or make-up system 1 fails. These probabilities have been calculated in static manner without crediting operation of the spent fuel pool cooling system or make-up system 2, and using mission time of 24 hours for make-up system 1. This approach corresponds to some extent to the enhanced fault/event tree approach described in (Tyrväinen et al., 2021).

The refined results calculated by static approach are much closer to the results from the simulation-based approach, but they are still significantly larger. The reason for the difference is the dynamic computation of the time windows in the simulation-based event tree. Concerning all minimal cut sets, a significant factor causing difference is the mission time of make-up system 1, which is on average only 0.9 hours in the simulations, whereas it is 24 hours in the static PRA model (see sensitivity case 2 in Table 5). Operation of the spent fuel pool cooling system with a diesel generator some time before it fails lowers the frequencies of all minimal cut sets with failure to run events of diesel generators (see sensitivity case 3), because it gives more time to recover the offsite power. In addition, the refined static analysis is performed with an assumption that make-up system 1 is operated with a repaired diesel generator not offsite power. In the simulations, offsite power may be recovered first, which increases the reliability of make-up system 1, and decreases minimal cut set frequencies significantly (see sensitivity case 4).

The frequencies of minimal cut sets with failure to run of the FLEX diesel generator (2, 4, 7, 14 and 15) are significantly lowered by the dynamic analysis, because the average mission time from simulations is only 1.5 hours, whereas 24 hours is used in the static analysis (see sensitivity case 5). In addition, operating make-up system 2 some time before failure decreases the frequencies of those minimal cut sets significantly (see sensitivity case 6), because it increases the water level and decreases the temperature giving more time for recovery and repair actions. When these effects are removed from the model, e.g. MCS 2 has almost the same frequency as MCS 1 as in the static analysis.

For failure to run events of the main diesel generators, the mission times are actually slightly longer than 24 hours on average, which increases the frequencies of the corresponding minimal cut sets compared with the results from static analysis. For failure to run CCF of all diesel generators, the average mission time is 24.7 hours. This means that the mission time of 24 hours in the static PRA model should be increased. The mission time should be selected based on the offsite power recovery time distribution.

One hidden impact that is not so easily seen from results is that all mission times are dependent as they all depend on the offsite power recovery time and some depend on diesel generator repair times. This relatively increases the frequencies of the minimal cut sets (10-13 and 15) including a failure to run CCF of three diesel generators and a failure to run event of a single diesel generator. When the offsite power recovery takes a long time, both events have large probability. The combined probability of those events is 155% larger in the simulations than in the static analysis with a mission time of 24 hours. Therefore, one should be careful when assigning independent mission times for events like these in static PRA.

When all the above mentioned dynamic effects are removed from the simulation model, the results are quite close to the results of the refined static analysis as they should be.

When a make-up system repair after the failure of make-up system 1 is added to the analysis, the fuel damage frequency decreases significantly, almost 96%. The reason is that the available time for the repair is very long. It seems obvious that modelling of another repair would decrease the result even more.

Table 5 presents a set of sensitivity analyses. All cases have been simulated without make-up system repair and with 10000 simulation cycles. The baseline result differs from the result presented in Table 4,

because much less simulation cycles have been used to reduce the computation time. The same seed number is used for all sensitivity cases so that poor accuracy is not an issue for relative results. The total frequency (1/year) and the frequency of MCS 2 (1/year) are presented for each case. The relative result compared with the baseline is presented in parentheses.

Table 5: Sensitivity analyses (without make-up system repair).

Case	**Total**	**MCS 2**
1. Baseline	8.64E-11	2.72E-12
2. Make-up system 1 mission time set to 24 hours	1.52E-10 (176%)	4.78E-12 (176%)
3. Failure to run events of diesel generators occur at start	1.82E-10 (211%)	3.89E-12 (143%)
4. Make-up system 1 always operated with a repaired diesel generator	1.10E-10 (128%)	3.60E-12 (132%)
5. Make-up system 2 mission time set to 24 hours	1.24E-10 (144%)	1.49E-11 (548%)
6. Failure to run event of the FLEX diesel generators occurs at start	9.47E-11 (110%)	7.51E-12 (276%)
7. Diesel generator repair not possible	7.70E-9 (8910%)	9.81E-10 (36100%)
8. Mean time for offsite power recovery is 10h (5h in the baseline)	2.14E-10 (247%)	7.18E-12 (264%)
9. Error factor of the offsite power recovery time is 20 (10 in the baseline)	3.41E-10 (395%)	1.01E-11 (371%)
10. Offsite power recovery time exponentially distributed with mean 5h (lognormally in the baseline)	1.47E-12 (1.7%)	1.47E-14 (0.54%)
11. Offsite power recovery time exponentially distributed with mean 10h	2.52E-11 (29%)	4.99E-13 (18%)
12. Offsite power recovery time exponentially distributed with mean 20h	1.70E-10 (197%)	4.78E-12 (176%)

Cases 2-6 show the impacts of removing some dynamic aspects from the model. In all cases, the risk is increased significantly by the reduction of dynamicity as already discussed earlier.

Case 7 highlights the importance of diesel generator repair. Without possibility to repair a diesel generator, the risk is increased by two orders of magnitude. The impact on MCS 2 is even larger, because without diesel generator repair, the mission time of the FLEX diesel generator is increased on average and the dependency between different mission times affected by the offsite power recovery time is stronger.

The results are relatively sensitive to the probability distribution of the offsite power recovery time based on the sensitivity cases 8-12. The tail of the distribution is particularly important, because the risk significant cases are those where the offsite power recovery lasts a long time. An exponential distribution produces significantly smaller frequencies than lognormal distribution unless a much longer mean time is used.

Similar sensitivity analyses have been performed for the model with a make-up system repair included. The results are presented in Appendix A in Table 8. The sensitivities are mostly similar to the results without a make-up system repair, but there are some differences. When diesel generator repair is removed from the model in case 7, the make-up system repair also reduces the risk very little, because the scenario where the offsite power is recovered so late that fuel damage occurs before make-up system 1 can be started dominates the result. In that case, it could make sense to repair make-up system 2 to buy more time, but that has been left out of the model, because it would not be important in the base model. The sensitivity is also greater in cases 3 and 9, because the offsite power recovery takes a longer time (relatively compared to boiling time in case 3), and therefore, the scenario where the power supply to make-up system 1 is not recovered in time starts to dominate over scenarios where a make-up system repair is possible.

6. Safe state consideration

In general, a safe state in PRA can be defined as a state, where the risk is negligible compared to overall PRA results as discussed in (Tyrväinen et al., 2021). Following this definition, it is case-specific what a safe state really is. A specific state can be considered safe in some accident sequence, while it is not safe, i.e. a significant risk is related to it, in another accident sequence. In some cases, it can be difficult to assess accurately what state is safe and what is not. Therefore, simplifications and expert judgments are needed, and suitable conservative assumptions can also be helpful if there is significant uncertainty.

In this spent fuel pool study, different safe states have been considered. Operation of the spent fuel pool cooling system with offsite power has been considered a safe state in any case. Also, operation of the spent fuel pool cooling system with a diesel generator has been considered a safe state in some cases, but not all. If a diesel generator is successfully started after loss of offsite power, it clearly cannot be considered a safe state, because the risk of failure to run and consequent fuel damage is significant (it even appears in the first MCS). On the other hand, if diesel generators first fail, one is repaired, and the operation of the spent fuel pool cooling system is then started again, the scenario with another failure and consequent fuel damage does not necessarily have significant frequency. At least, it is far from the most important scenario. In this study, operation of the spent fuel pool cooling system after diesel generator repair has been assumed as a safe state, but it is uncertain if the related risk is negligible. It could be studied by modelling another failure and the consecutive accident progression. On the other hand, it can be confidently said that the underestimation of the risk is only small if significant at all. Hence, it is not necessarily worthwhile to study such matter as the model includes other conservative assumptions.

A further issue in the diesel generator related safe state consideration is if successful start of the spent fuel pool cooling system is required for the safe state, or if it is enough that the diesel generator is repaired. The risk related to failure to start the spent fuel pool cooling system after diesel generator repair and consequent fuel damage is surely small, but its significance is also an open question. In the model, the failure to start has been modelled when the repair is complete before boiling, but not if the spent fuel pool has been recovered from boiling conditions. The modelling decisions have been based on expert judgments on what is significant and what is not, also considering the required modelling efforts to extend the scenarios. These issues are not only relevant for dynamic analysis, but also for static PRA. They could be explored further, since they are at least theoretically interesting. It is surely not practical to perform such analyses in every PRA study, but experience from one study could be helpful in making other similar modelling decisions.

7. Software tool development possibilities

7.1 Improvements to the spent fuel pool PRA

The simulation-based event tree approach for spent fuel pool was developed to enable modelling of time-dependencies in spent fuel pool accident scenarios. For example, the mission time of a make-up system depends on the spent fuel pool conditions at the start time of the make-up system and the recovery time of the spent fuel pool cooling, and the repair probability of a component depends on the spent fuel pool conditions at the failure time, etc. It is not possible to capture this type of dependencies in fault trees. On the other hand, the simulation-based event tree approach, as applied in this report, becomes impractical when the number of failure combinations is large, because the model grows too large. It would be important to resolve this problem so that the method would be more useful for practical PRA. Three possible solutions are identified here:

1. The simulation-based event tree would be developed as an independent PRA model, and would not be used only for the quantification of minimal cut sets. Failures with same impacts would be merged, system level failure modes would be used in the event tree, and the computation of the failure rates and probabilities for the system level failure modes would be performed in background.
2. Fault trees would be integrated to the simulation-based event trees.
3. The simulation model would be separated from the event tree and the simulation-based quantification of the minimal cut sets would be automated.

The first option requires least method and tool development, but it would also mean loss of minimal cut set results. Since minimal cut sets are important qualitative results from PRA, we will focus on options 2 and 3.

7.1.1 Integration of fault trees to the simulation-based event trees

The main benefit in the integration of fault trees to the simulation-based event trees would be that there would be no need to develop a separate event tree branch for each basic event of the static PRA model, there would be no need to have a separate static PRA model, and the final results could be calculated directly using the simulation-based event trees instead of a separate tool, such as spreadsheet. Static fault trees and dynamic simulation model would be integrated into one model. Failure probabilities, failure rates and MTTR parameters used in the simulations would come directly from the fault trees linked to the sections of the simulation-based event tree.

In the simplest case, a fault tree could be used just to calculate the probability of an event tree branch or a probability parameter value to be used in script-based calculations. It would be fairly straightforward to implement. While it could be useful in some cases, more advanced functionality would be needed. Challenges include how to take into account dependencies between fault trees, how to handle different basic event combinations in simulations, and how to get the relevant information, e.g. MTTR values, from basic events.

To take into account dependencies between fault trees, a simple solution is to solve the minimal cut sets of the combined fault tree (fault trees connected by an AND gate). Therefore, minimal cut sets should be solved for each event tree sequence, just like in normal level 1 PRA. If this approach is selected, then there is a question how the minimal cut sets and the simulations are connected. One possibility would be to perform the simulations for each minimal cut set utilizing the information related to the basic events in the minimal cut sets. The analysis would include two phases: first minimal cut sets would be solved for the event tree sequences, and second the minimal cut sets would be simulated using the event tree scripts. This could work, but the number of required simulations would be a concern. Possibly shared simulations could be performed for the minimal cut sets with same simulation parameters (e.g. failure rates and MTTR values) to reduce the computational burden, like in the spent fuel pool analyses performed so far.

If minimal cut sets are solved first for event tree sequences, and simulations are performed after that, the correspondence of basic events and event tree sections is kind of lost, i.e. a basic event in a minimal cut set is not directly linked to any event tree section. Therefore, there needs to be a way to identify which basic event defines which simulation parameter. Basic events could e.g. have some sort of attributes that would be used in the script files. For example, a specific MTTR parameter in the computation scripts would come from a basic event with a specific attribute.

In this context, attributes and parameters may be different from regular level 1 PRA data. For example, the MTTR of a repair after failure during an accident scenario is not necessarily the same as the MTTR used in the computation of unavailability related to repair time in normal level 1 PRA. The basic event data used in the simulations could therefore be separate from normal PRA data. CCFs would also need to have such parameters, or those should be possible to derive from the corresponding single failure basic events. Attributes could also be used to control which script parts are executed for a specific minimal cut set. For example, failure to run and failure to start events need to be handled differently in the scripts, so they could simply have different attributes, which would be more convenient than having separate event tree branches for those.

7.1.2 Simulation module for minimal cut set quantification

The simulation model used in the minimal cut set quantification could also be separate from the event tree. It could anyway work with quite similar principles as the integrated simulation-based event tree model discussed in the previous section (attributes to control which script parts are executed, etc.). Since the idea is to simulate minimal cut sets solved for the end points of an event tree, there is no particular need to integrate the simulation model section by section to the same event tree (so that the event tree needs to be gone through twice). Instead, a separate script-based model for minimal cut set quantification could have significant benefits:

- It would be more flexible than a simulation model tied to an event tree. It would also enable simpler calculations with short scripts only related to e.g. one or two basic events in a minimal cut set instead of all. It could be used just to apply customized computation formulas instead of extensive simulation modelling, if that would be useful for the application in question. It would offer more flexibility for simulations, e.g. to model actions and events that can occur in different orders as identified in the previous report (Tyrväinen et al., 2021).
- It could be implemented as a separate software module that would read the minimal cut sets and relevant input data. The module would not necessarily need to be part of an existing software, but could be a separate one and could support different minimal cut set formats. However, the source codes of FinPSA level 2 would provide a good basis for the development of the module.

The implementation of the simulation module would not be very complex compared with simulation-based event trees. Instead of having a predefined event tree sequence for each minimal cut set, the basic events of a minimal cut set would determine which functions are executed in the simulation model. For this, each basic event should have some sort of attribute. Possible basic event attributes could be e.g. initiating event, specific failure mode of a specific system/train, specific human failure event, etc. The attributes would be defined by the user case specifically. In addition, there could be an attribute defining static basic events that would not participate in the simulations and would be handled in the minimal cut set frequency computation in the normal way.

There could be a so-called main function that would call the functions related to the basic events. This main function would replace the event tree in practise. The functions related to the basic events would correspond to the branch functions of the simulation-based event tree. These basic event functions would be executed according to IF-ELSE clauses based on the basic event attributes that appear in the minimal cut set. For example, when a specific basic event attribute appears in the minimal cut set, a specific function is executed. In the same way, parameters in the simulation model (MTTR, failure rate, probability)

would be read from the basic events with help of the attributes, i.e. a specific parameter comes from a basic event with a specific attribute.

Based on these preliminary considerations, the development of a separate simulation model for minimal cut set quantification is seen as a better option than integrated simulation-based event tree and fault trees. A separate simulation model would be more flexible, and it would be beneficial to have a separate module to do this advanced quantification.

7.1.3 Uncertainty analysis

Proper uncertainty analysis in this simulation-based approach would require separation of epistemic and aleatory uncertainties as discussed in (Tyrväinen & Karanta, 2019). The reason for such separation is that the aim would be to estimate the epistemic uncertainty related to fuel damage frequency, whereas the fuel damage frequency itself represents the aleatory uncertainty with regard to the occurrence of the fuel damage. In practise, the simulation model should include epistemic and aleatory variables, which should be treated separately. For example, a distribution defined for a repair time variable would represent aleatory uncertainty, and the distributions of the parameters of the repair time distribution would represent epistemic uncertainty. The most straightforward way to perform the uncertainty analysis would be to have two separate sampling loops in the Monte Carlo simulation, the outer loop for the epistemic uncertainties and the inner loop for the aleatory uncertainties.

FinPSA does not currently include capability to perform Monte Carlo in two separate sampling loops for uncertainty analysis. It would be useful to develop such capability, not only for this application, but also for any probabilistic simulation that includes aleatory variables. If the new simulation module described in the previous section was developed, possibility for two-stage Monte Carlo could also be developed for it. Performing Monte Carlo in two separate loops requires a very large number of simulations. Ways to perform the analysis more efficiently could therefore also be studied. For example, discretisation of time distributions could be a way to reduce the number of simulation runs. That approach is often used in dynamic event trees (Karanki & Dang, 2016). There are also ways to perform approximate uncertainty analysis without heavy two-stage Monte Carlo (Hofer et al., 2002; Karanki et al., 2017). Furthermore, there are more efficient sampling techniques than Monte Carlo (Rahman et al., 2018).

7.2 Other development possibilities related to time windows

These development ideas concern PRA in general, not the spent fuel pool PRA approach on which this report has focused.

7.2.1 Different mission times in different sequences

Sometimes, there is need to model different mission times for the same safety function in different sequences as identified e.g. in (Tyrväinen et al., 2020). It is, of course, possible to create separate basic events and fault trees for different mission times, but it is not very efficient. It would be handier if the mission time could change automatically e.g. based on the sequence, i.e. the mission time could be a sequence specific property. It could mean that a basic event would not have a fixed mission time, but it would come from the sequence definitions, when that option is selected.

Having the same basic event with different mission times in different sequences would not be a problem as long as quantification would be performed for one sequence at a time. Therefore, it would be important to preserve the sequence information when combining minimal cut sets from different sequences and quantifying the combined minimal cut set list. It would require modification to the computation algorithms, but it should not be difficult to implement.

It would be better to have only one basic event for a specific failure event instead of having multiple basic events with different mission times, because the importance analysis would be more straightforward. However, the algorithms used in the importance analysis should then take into account that the same

basic event can have different mission times and different probability in different minimal cut sets, but it would not be difficult to program.

In theory, minimization of cut sets could become a problem if the same basic event had different mission times. There could be a minimal cut set with a shorter mission time, and a minimal cut set with a longer mission time and additional event. It would be incorrect to minimize the minimal cut set with a longer mission time, but it is not known if such cases would occur in real models. It would be easiest not to minimize at all in that case. For example, AIMS-PSA software (Han et al., 2016) does not minimize cut sets when it combines minimal cut sets from different sequences, which is justifiable, because accident sequences are mutually exclusive.

7.2.2 Automatic generation of fault trees

Automatic generation of fault trees is another way that could facilitate e.g. use of different mission times. In this context, it could mean that different versions of the fault tree of a safety function would be automatically created based on one master fault tree. One could specify parameters, such as mission time, for the fault tree generation, e.g. so that the basic events in the new version of the fault tree would have the selected mission time.

One possible application could be modelling of dynamic success criteria as speculated in (Tyrväinen et al., 2020). Basic events related to different time windows could be created automatically and placed in the fault trees, and the top fault tree gate would be selected based on the success criterion in the corresponding time interval.

A drawback in this approach would be that there would be multiple basic events representing the same failure event, which would be inconvenient for importance analysis. A functionality to handle those basic events as one in importance analysis would then be useful.

7.2.3 Convolution

As noticed in this study, basic events can be dependent through shared or dependent mission time. Loss of offsite power is a typical example, because the mission times of diesel generators depend on offsite power recovery time. One option for modelling is to divide offsite power recovery times into categories that are modelled separately with separate basic events corresponding to the mission times defined by the categories. This is however not very handy. Another solution would be to use convolution as discussed in (U.S. NRC, 2017). It could mean that a probability distribution would be assigned to the mission time parameter, which would be common to all relevant basic events. The quantification could be performed e.g. by Monte Carlo simulation based on the distribution or quantifying discrete time intervals separately. This could be a useful feature in a PRA software.

7.3 Other methods

7.3.1 Markov models

Markov models are handy in modelling dynamic failure-repair processes. I&AB module in RiskSpectrum (Tyrväinen et al., 2021; Bouissou, 2018) enables use of Markov approach with large PRA models. It is a post-processing tool for minimal cut set quantification. This approach is very efficient and good when the assumptions related to Markov analysis are sufficient. However, the Markov approach restricts one to use constant repair and failure rates, and it is difficult to take time-dependent plant conditions into account. Simulation-based approaches, on the other hand, are much more flexible with regard to assumptions and modelling of dynamic plant conditions.

7.3.2 Dynamic event trees

In dynamic event tree approach (Karanki & Dang, 2016; Karanki et al., 2017), a simulator generates accident sequences automatically together with a plant simulator. During a simulation run, the tool identifies stochastic branching points based on the plant conditions (e.g. when a safety function is activated and can fail), and later, the tool generates simulation runs with the events related to those branching points. For each generated accident sequence, the simulator determines whether e.g. core damage would occur or not. With a realistic simulator, dynamic event tree analysis is highly realistic, but also computationally very demanding. It does not seem suitable option for practical PRA at this point, but might be such in the future, with more powerful computers and algorithms. The simulation-based approach discussed in this report has some similar qualities as dynamic event trees, though accident sequences are not generated automatically, but are defined beforehand except for timings. In this sense, the development of the simulation-based approach could be a way towards dynamic event trees in long-term.

8. Conclusions

This report has presented an approach for simulation-based PRA of a spent fuel pool. A simulation-based event tree model has been developed to analyse loss of offsite power accident of a spent fuel pool. In the simulation-based event tree, accident timings, such as failure times of components and durations of manual actions, are simulated to analyse time-dependencies. The time windows for probabilistic analysis, namely mission times for safety functions and available times for manual actions, are calculated based on spent fuel pool conditions affected by the timings of previous events. The model combines deterministic and probabilistic analysis; the spent fuel pool conditions are calculated by a simplified, but sufficiently realistic deterministic model.

Compared to the previous part of the study (Tyrväinen et al., 2021), more realistic models for the behaviour of the temperature and water level of the spent fuel pool were implemented in this report. The modelling approach was chosen based on a survey of literature. The zero-dimensional model of Ramadan et al. (2018) was seen simple but realistic enough to be implemented to the simulation-based FinPSA scripts. Some simplifications were made from the original approach, such as assuming the spent fuel building pressure and air temperature constant, since it did not have a significant effect on the results. The modelling approach is based on solving the spent fuel pool water mass and energy balances. The differential equations are solved numerically using Euler's approach. For real application, the model should still be validated against a more realistic computer code. This study anyway demonstrated how a deterministic physical model can be integrated in FinPSA scripts and used as a part of a dynamic PRA model.

In this report, the simulation-based model was used to quantify minimal cut sets of a static PRA model more realistically, while there would also be other possibilities to apply the method. The results of dynamic and static analyses of a loss of offsite power scenario were compared. The dynamic analysis decreased the frequencies of minimal cut sets significantly. The decrease was particularly related to more realistic definition of mission times and crediting the operation of the cooling/make-up systems before they fail, which gives more time for the following manual actions. The results also indicated that crediting repairs can greatly decrease the frequencies, though those can also be modelled in static manner to some extent.

Even though the frequencies were significantly decreased by dynamic analysis, it was also noticed that dependencies between mission times can have a risk increasing effect. For example, when the recovery of the offsite power takes a long time, all dependent mission times are long and the basic events have large probabilities. Therefore, one should be careful when assigning independent mission times that are really dependent in static PRA.

Even though dynamic analysis is more realistic, static analysis may also provide satisfactory results depending on the desired level of accuracy. In this case study, the results of the original static PRA model

needed to be refined to reduce the excessive conservatism. This was done by adding extra terms to the minimal cut sets concerning offsite power recovery, repair of a diesel generator and make-up system 1. This produced conservative results that may be considered quite acceptable. On the other hand, the development of the dynamic model helped in understanding what kind of refinements were needed for the static analysis. Dynamic analysis, in general, can provide insights not obtained from static analyses.

Some needs to develop the simulation-based model further were identified during this work. Modelling of the case where offsite power recovery comes soon after diesel generator repair when operating make-up system 1 is quite conservative in the current model, because the offsite power recovery is not credited if the diesel generator fails again. In addition, the transient scenario modelled in (Tyrväinen et al., 2021) could be updated by proper modelling of the mission times of the standby trains of the spent fuel pool cooling system, which were not included in the simulations in (Tyrväinen et al., 2021). The mission times could possibly depend on the repair time of the active spent fuel pool cooling train.

There are some challenges related to application of the approach for full-scope spent fuel pool PRA. The simulation-based event tree becomes easily very complex when there are many failure combinations to analyse, and there is no good tool support to integrate the minimal cut sets of static PRA and the simulation results. Potential solutions were discussed in the report. One possibility would be to develop simulation-based event trees as independent PRA model so that there would be no need for static PRA model. However, the minimal cut set information would be lost, and the identification of all relevant failure combinations could be a challenge. Another potential solution would be to develop a simulation module for automatic quantification of minimal cut sets. There is no particular need to tie the simulations to an event tree. The execution of the simulation scripts could be controlled by attributes related to the basic events in the minimal cut sets. This would provide more flexibility than a simulation-based event tree and would give wider possibilities to perform advanced minimal cut set quantifications.

References

Adorni, M, Esmaili, H, Grant, W, Hollands, T, Hozer, Z, Jaeckel, B, Munoz, M, Nakajima, T, Rocchi, F, Strucic, M, Tregoures, N, Vokac, P, Ahn, KI, Bourgue, L, Dickson, R, Douxchamps, PA, Herranz, LE, Jernkvist, LO, Amri, A, Kissane, MP. (2015). Status report on spent fuel pools under loss-of-cooling and loss-of-coolant accident conditions - final report. NEA/CSNI/R(2015)2. OECD Nuclear Energy Agency. http://inis.iaea.org/search/search.aspx?orig_q=RN:46066604

Aldemir, T. (2013). A survey of dynamic methodologies for probabilistic safety assessment of nuclear power plants, Annals of Nuclear Energy, 52, 113-124.

Bouissou, M. (2018). Extensions of the I&AB method for the reliability assessment of the spent fuel pool of EPR, European Safety and Reliability Conference (ESREL 2018), Trondheim, Norway, 17-21 June, 2018.

Carlos, S, Sanchez-Saez, F, Martorell, S. (2014). Use of TRACE best estimate code to analyze spent fuel storage pools safety, Progress in Nuclear Energy, 77, 224–238. https://doi.org/10.1016/j.pnucene.2014.07.008

Han, SH, Lim, H-G, Jang, S-C, Yang, J-E. (2016). AIMS-PSA: A software for integrated PSA, 13th international conference on probabilistic safety assessment and management (PSAM13), Seoul, Korea, 2-7 October, 2016.

Hofer, E, Kloos, M, Krzykacz-Hausmann, B, Peschke, J, Woltereck, M. (2002). An approximate epistemic uncertainty analysis approach in the presence of epistemic and aleatory uncertainties, Reliability Engineering and System Safety, 77, 229-238.

Hung, T-C, Dhir, VK, Pei, B-S, Chen, Y-S, Tsai, FP. (2013). The development of a three-dimensional transient CFD model for predicting cooling ability of spent fuel pools, Applied Thermal Engineering, 50 (1), 496–504. https://doi.org/10.1016/j.applthermaleng.2012.06.042

International Atomic Energy Agency. (2010). Development and application of level 1 probabilistic safety assessment for nuclear power plants, specific safety guide series No. SSG-3, Vienna.

Johnson, N, Ma, Z. (2019). Analysis of loss-of-offsite-power events 1987-2018, INL/EXT-19-54699. Idaho National Laboratory, Idaho.

Kaliatka, A, Ognerubov, V, Vileiniskis, V. (2010). Analysis of the processes in spent fuel pools of Ignalina NPP in case of loss of heat removal, Nuclear engineering and design, 240 (5), 1073–1082. https://doi.org/10.1016/j.nucengdes.2009.12.026

Karanki, DR, Dang, VN. (2016). Quantification of dynamic event trees - A comparison with event trees for MLOCA scenario, Reliability Engineering and System Safety, 147, 19-31.

Karanki, DR, Rahman, S, Dang, VN, Zerkak, O. (2017). Epistemic and aleatory uncertainties in integrated deterministic and probabilistic safety assessment: Tradeoff between accuracy and accident simulations, Reliability Engineering and System Safety, 162, 91-102.

Rahman, S, Karanki, DR, Epiney, A, Wicaksono, D, Zerkak, O, Dang, VN. (2018). Deterministic sampling for propagating epistemic and aleatory uncertainty in dynamic event tree analysis, Reliability Engineering and System Safety, 175, 62-78.

Ramadan, A, Hasan, R, & Penlington, R. (2018). Zero-dimensional transient model of large-scale cooling ponds using well-mixed approach, Annals of nuclear energy, 114, 342–353. https://doi.org/10.1016/j.anucene.2017.12.043

Tynys, H. (2017). Safety assessment of interim spent nuclear fuel storage [Master's thesis]. Lappeenranta University of Technology, Lappeenranta, Finland.

Tyrväinen, T, Karanta, I. (2019). Dynamic containment event tree modelling techniques and uncertainty analysis, VTT-R-06892-18, VTT Technical Research Centre of Finland Ltd, Espoo, Finland.

Tyrväinen, T, Karanta, I, Kling, T, He, X, Olofsson, F, Bäckström, O, Massaiu, S, Sparre, E, Eriksson, C, Cederhorn, E, Authen, S. (2020). Prolonged available time and safe states, NKS-432, Nordic nuclear safety research, Roskilde.

Tyrväinen, T, Karanta, I, Kling, T, He, X, Olofsson, F, O, Massaiu, S, Sparre, E, Eriksson, C, Cederhorn, E, Authen, S. (2021). Prolonged available time and safe states, NKS-444, Nordic nuclear safety research, Roskilde.

Tyrväinen, T, Silvonen, T, Mätäsniemi, T. (2016). Computing source terms with dynamic containment event trees, 13th international conference on probabilistic safety assessment and management (PSAM13), Seoul, Korea, 2-7 October, 2016.

United States Nuclear Regulatory Commission. (2017). Risk assessment of operational events, Handbook, Volume 1 - Internal events, Revision 2.02, Washington DC.

VTT Technical Research Centre of Finland Ltd. (2014), FinPSA - Tool for promoting safety and reliability, https://www.simulationstore.com/finpsa (link accessed 21.9.2021).

Yanagi, C, Murase, M. (2013). One-region model predicting water temperature and level in a spent fuel pit during loss of all AC power supplies, Journal of Power and Energy Systems, 7 (1), 18–31. https://doi.org/10.1299/jpes.7.18

Appendix A: Detailed results

Table 6: Top minimal cut sets.

Mc_num	Freq	Basic event names				
1	2.36E-08	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-ALL	SFPMU:2_P1_____A	
2	2.32E-08	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-ALL	ACP__DG102_FLEX2___D	
3	1.09E-08	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-ALL	SFPMU:2_P1_____A	
4	1.07E-08	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-ALL	ACP__DG102_FLEX2___D	
5	2.99E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-ALL	SFPMU:2_MANSTART___H	
6	2.74E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------A-ALL	SFPMU:2_P1_____A	
7	2.70E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------A-ALL	ACP__DG102_FLEX2___D	
8	2.70E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-ALL	ACP__DG102_FLEX2___A	
9	2.03E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AC	SFPC_P2_____A	SFPMU:2_P1_____A
10	2.00E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AD	ACP10DG001_____D	SFPMU:2_P1_____A
11	2.00E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AA	ACP40DG001_____D	SFPMU:2_P1_____A
12	2.00E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AB	ACP30DG001_____D	SFPMU:2_P1_____A
13	2.00E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AC	ACP20DG001_____D	SFPMU:2_P1_____A
14	2.00E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AC	ACP__DG102_FLEX2___D	SFPC_P2_____A
15	1.97E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AB	ACP30DG001_____D	ACP__DG102_FLEX2___D
16	1.97E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AD	ACP10DG001_____D	ACP__DG102_FLEX2___D
17	1.97E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AC	ACP20DG001_____D	ACP__DG102_FLEX2___D
18	1.97E-09	!IE-LOOP	ACN10GT001_____A	ACP-DG--------D-3AA	ACP40DG001_____D	ACP__DG102_FLEX2___D
19	1.38E-09	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-ALL	SFPMU:2_MANSTART___H	
20	1.26E-09	!IE-LOOP	ACN10GT001_____M	ACP-DG--------A-ALL	SFPMU:2_P1_____A	
21	1.24E-09	!IE-LOOP	ACN10GT001_____M	ACP-DG--------A-ALL	ACP__DG102_FLEX2___D	
22	1.24E-09	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-ALL	ACP__DG102_FLEX2___A	
23	9.37E-10	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-3AC	SFPC_P2_____A	SFPMU:2_P1_____A
24	9.22E-10	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-3AA	ACP40DG001_____D	SFPMU:2_P1_____A
25	9.22E-10	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-3AC	ACP20DG001_____D	SFPMU:2_P1_____A
26	9.22E-10	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-3AD	ACP10DG001_____D	SFPMU:2_P1_____A
27	9.22E-10	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-3AC	ACP__DG102_FLEX2___D	SFPC_P2_____A
28	9.22E-10	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-3AB	ACP30DG001_____D	SFPMU:2_P1_____A
29	9.07E-10	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-3AC	ACP20DG001_____D	ACP__DG102_FLEX2___D
30	9.07E-10	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-3AA	ACP40DG001_____D	ACP__DG102_FLEX2___D
31	9.07E-10	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-3AB	ACP30DG001_____D	ACP__DG102_FLEX2___D
32	9.07E-10	!IE-LOOP	ACN10GT001_____M	ACP-DG--------D-3AD	ACP10DG001_____D	ACP__DG102_FLEX2___D

Table 7: Fuel damage frequencies (1/year) of minimal cut sets.

MCS	Seq	Static	Dynamic without MU repair	Dynamic with MU repair	Refined static
Total		1.14E-07	8.66E-11	4.43E-12	3.39E-10
1	9	2.36E-08	3.02E-11	1.62E-12	7.25E-11
2	5	2.32E-08	2.68E-12	1.15E-13	7.13E-11
3	9	1.09E-08	1.40E-11	7.50E-13	3.35E-11
4	5	1.07E-08	1.23E-12	5.33E-14	3.29E-11
5	11	2.99E-09	3.85E-12	2.59E-13	9.19E-12
6	19	2.74E-09	2.13E-12	1.02E-13	4.60E-12
7	15	2.70E-09	6.50E-14	2.77E-15	4.53E-12
8	7	2.70E-09	3.32E-12	1.78E-13	8.30E-12
9	29	2.03E-09	3.17E-12	1.62E-13	6.24E-12
10	39	2.00E-09	3.32E-12	1.41E-13	6.15E-12
11	39	2.00E-09	3.32E-12	1.41E-13	6.15E-12
12	39	2.00E-09	3.32E-12	1.41E-13	6.15E-12
13	39	2.00E-09	3.32E-12	1.41E-13	6.15E-12
14	25	2.00E-09	1.03E-13	4.67E-15	6.15E-12
15	35	1.97E-09	9.76E-14	4.03E-15	6.06E-12
16	35	1.97E-09	9.76E-14	4.03E-15	6.06E-12
17	35	1.97E-09	9.76E-14	4.03E-15	6.06E-12
18	35	1.97E-09	9.76E-14	4.03E-15	6.06E-12
19	11	1.38E-09	1.78E-12	1.20E-13	4.24E-12
20	19	1.26E-09	9.78E-13	4.69E-14	2.11E-12
21	15	1.24E-09	2.99E-14	1.27E-15	2.08E-12
22	7	1.24E-09	1.53E-12	8.19E-14	3.81E-12
23	29	9.37E-10	1.46E-12	7.47E-14	2.88E-12
24	39	9.22E-10	1.53E-12	6.50E-14	2.83E-12
25	39	9.22E-10	1.53E-12	6.50E-14	2.83E-12
26	39	9.22E-10	1.53E-12	6.50E-14	2.83E-12
27	25	9.22E-10	4.75E-14	2.15E-15	2.83E-12
28	39	9.22E-10	1.53E-12	6.50E-14	2.83E-12
29	35	9.07E-10	4.49E-14	1.86E-15	2.79E-12
30	35	9.07E-10	4.49E-14	1.86E-15	2.79E-12
31	35	9.07E-10	4.49E-14	1.86E-15	2.79E-12
32	35	9.07E-10	4.49E-14	1.86E-15	2.79E-12

Table 8: Sensitivity analyses (with make-up system repair).

Case	Total	MCS 2
1. Baseline	3.72E-12	1.12E-13
2. Make-up system 1 mission time set to 24 hours	6.47E-12 (174%)	1.99E-13 (178%)
3. Failure to run events of diesel generators occur at start	2.59E-11 (698%)	1.64E-13 (146%)
4. Make-up system 1 always operated with a repaired diesel generator	4.67E-12 (126%)	1.47E-13 (131%)
5. Make-up system 2 mission time set to 24 hours	5.27E-12 (142%)	6.14E-13 (548%)
6. Failure to run event of the FLEX diesel generators occurs at start	4.37E-12 (118%)	3.21E-13 (286%)
7. Diesel generator repair not possible	7.63E-9 (205000%)	9.73E-10 (868000%)
8. Mean time for offsite power recovery is 10h	9.30E-12 (250%)	3.02E-13 (269%)
9. Error factor of the offsite power recovery time is 20	3.30E-11 (888%)	4.33E-13 (386%)
10. Offsite power recovery time exponentially distributed with mean 5h	6.34E-14 (1.7%)	6.06E-16 (0.54%)
11. Offsite power recovery time exponentially distributed with mean 10h	1.07E-12 (29%)	2.03E-14 (18%)
12. Offsite power recovery time exponentially distributed with mean 20h	7.12E-12 (192%)	1.94E-13 (173%)

Appendix B: Scripts of the simulation-based event tree

The scripts of the simulation-based event tree are presented in the following. The common section containing global variables and functions is presented last.

Initial section

```
$ Most global variables are defined in the common section.

$ Random variables for timing determination
real rr

real t_sfpc, t_missionDGs, t_mission1, t_avail2, bt

$ Offsite power recovery time distribution
LOGNOR OPR = (5, 10)

$ Variable values that are collected to results
```

```
Collect t_mission2, t_repair, t_missionDGs, t_mission1, OPRecT, t_avail2, bt

$ Routine init is executed first
routine init
  FD = false

  WLevel = InitWLevel
  Temperature = NormalTemp

  $ Boiling time is calculated.
  boiltime = t_boil(NormalTemp, InitWLevel)
  bt = boiltime

  $ Probability that the recovery is not performed before boiling.
  BINFREQ = 1-cumul(OPR, boiltime)

  $ Offsite power recovery time from lognormal distribution.
  rr = 1-random()*BINFREQ
  OPRecT = icumul(OPR, rr)

  $ Recovery failure probability scaled.
  $ Only LOOP events over 4 hours are counted in the LOOP frequency.
  BINFREQ = BINFREQ/(1-cumul(OPR, 4))

  $ Initialization
  t_mu1 = 0
  mttr1 = 0
  mttr2 = 0
  rr2 = random()
  rr3 = random()
  MU1EFAIL = false
  MU2EFAIL = false
return


routine finish
  $ No final calculations in this model.
return

$ Routine binner is used to categorise accident sequences based on e.g. Boolean variables.
Class FD
routine binner active
(true, 'FD'),
(*,    'OK')
return
```

DGs (Diesel generators)

```
real prob, fr, t_mission, r, r2, r3, r4, r5, t_avail, t_diag,
     t_exe, t_start, t_delay, t_fail, t_shift

routine init
  r = random()     $ Random value between 0 and 1
  r2 = random()
  r3 = random()
  r4 = random()
  r5 = random()

  t_shift = 8
return


function nil OK

return nil


$ Failure to run CCF between 4 diesel generators
```

```
function real FTR_ALL
  $ Failure rate
  fr = FR_DG_ALL

  $ Mission time based on the offsite power recovery time.
  $ Failure before the recovery time is allowed
  $ as long as the boiling does not start before the recovery.
  t_mission = OPRecT-boiltime

  $ The failure probability is calculated.
  prob = 1-exp(-fr*t_mission)

  $ The failure time is determined.
  t_fail = t_mission*r

  $ Time available to switch to other redundancy is the time to boiling.
  t_avail = boiltime

  $ Delay related to switching crew shift
  t_delay = r3*t_shift

  $ The execution time is drawn from uniform distribution.
  t_exe = 1*r2+0.5

  $ The diagnosis time is drawn from lognormal distribution.
  r4 = r4*cumul(SFPCD,t_avail-t_exe-t_delay)
  t_diag = icumul(SFPCD,r4)

  $ The time delay related to the train switching actions
  t_start = t_delay+t_diag+t_exe

  $ Temperature is updated based on the delay.
  Temperature = newTemp(NormalTemp, WLevel, 0, t_start)

  $ The time when all diesel generators are failed.
  $ DGs 2-4 are conservatively assumed to fail at start.
  t_sfpc = t_start + t_fail

  t_missionDGs = t_mission  $ Collect to results

  $ Mean time to repair a diesel generator.
  sfpcmrt = MTTR_DG_FTR
return prob


$ Failure to start CCF between 4 diesel generators
function real FTS_ALL
  $ Time available to switch to other redundancy is the time to boiling.
  t_avail = boiltime

  $ Delay related to switching crew shift
  t_delay = r3*t_shift

  $ The execution time is drawn from uniform distribution.
  t_exe = 1*r2+0.5

  $ The diagnosis time is drawn from lognormal distribution.
  r4 = r4*cumul(SFPCD,t_avail-t_exe-t_delay)
  t_diag = icumul(SFPCD,r4)

  $ The time delay related to the train switching actions
  t_start = t_delay+t_diag+t_exe

  $ Temperature is updated based on the delay.
  Temperature = newTemp(NormalTemp, WLevel, 0, t_start)

  $ CCF probability
  prob = 1.68E-5
```

```
  $ The time when all diesel generators are failed.
  t_sfpc = t_start

  $ Mean time to repair a diesel generator.
  sfpcmrt = MTTR_DG_FTS
return prob


$ Failure to run CCF between 3 diesel generators (the 4th train fails to start)
function real FTR_3
  $ Failure rate
  fr = FR_DG_3

  $ Mission time based on the offsite power recovery time.
  $ Failure before the recovery time is allowed
  $ as long as the boiling does not start before the recovery.
  t_mission = OPRecT-boiltime

  $ The failure probability is calculated.
  prob = 1-exp(-fr*t_mission)

  $ The failure time is determined.
  t_fail = t_mission*r

  $ Time available to switch to other redundancy is the time to boiling.
  t_avail = boiltime

  $ Delay related to switching crew shift
  t_delay = r3*t_shift

  $ The execution time is drawn from uniform distribution.
  t_exe = 1*r2+0.5

  $ The diagnosis time is drawn from lognormal distribution.
  r4 = r4*cumul(SFPCD,t_avail-t_exe-t_delay)
  t_diag = icumul(SFPCD,r4)

  $ The time delay related to the train switching actions
  t_start = t_delay+t_diag+t_exe

  $ Temperature is updated based on the delay.
  Temperature = newTemp(NormalTemp, WLevel, 0, t_start)

  $ The time when all diesel generators are failed.
  $ Standby trains are conservatively assumed to fail at start.
  t_sfpc = t_start + t_fail

  t_missionDGs = t_mission  $ Collect to results

  $ Mean time to repair a diesel generator.
  sfpcmrt = MTTR_DG_FTR
return prob

$ Failure to run CCF between 3 diesel generators and
$ independent failure to run of one diesel generator
function real FTR_3_1
  $ Failure rate for CCF
  fr = FR_DG_3

  $ Mission time based on the offsite power recovery time.
  $ Failure before the recovery time is allowed
  $ as long as the boiling does not start before the recovery.
  t_mission = OPRecT-boiltime

  $ The failure probability is calculated.
  prob = 1-exp(-fr*t_mission)
```

```
  $ The failure time is determined.
  t_fail = t_mission*r

  $ Time available to switch to other redundancy is the time to boiling.
  t_avail = boiltime

  $ Delay related to switching crew shift
  t_delay = r3*t_shift

  $ The execution time is drawn from uniform distribution.
  t_exe = 1*r2+0.5

  $ The diagnosis time is drawn from lognormal distribution.
  r4 = r4*cumul(SFPCD,t_avail-t_exe-t_delay)
  t_diag = icumul(SFPCD,r4)

  $ The time delay related to the train switching actions
  t_start = t_delay+t_diag+t_exe

  $ Temperature is updated based on the delay.
  Temperature = newTemp(NormalTemp, WLevel, 0, t_start)

  $ Time when a standby train is started
  t_start = t_start+t_fail

  $ Failure rate for the independent failure
  fr = FR_DG

  $ Mission time based on the offsite power recovery time.
  $ Failure before the recovery time is allowed
  $ as long as the boiling does not start before the recovery.
  t_mission = EarliestTime(Temperature,OPRecT-t_start,WLevel)

  if (t_mission == 0) then
  begin
    prob = 0   $ Safe state is reached already after the start.
  end
  else
  begin
    $ The failure probability is calculated.
    prob = prob*(1-exp(-fr*t_mission))

    $ The failure time is determined.
    t_fail = t_mission*r5

    $ The spent fuel pool conditions are updated based on the failure time.
    Temperature = newTemp(Temperature, WLevel, m_sfpcs, t_fail)

    $ The time when all diesel generators are failed.
    t_sfpc = t_start + t_fail
  end

  t_missionDGs = t_mission   $ Collect to results

  $ Mean time to repair a diesel generator.
  sfpcmrt = MTTR_DG_FTR
return prob
```

SFPC_REC (Cooling recovery before boiling)

```
real p, r, r2, t_avail, t_diag

routine init
  r = random()
  r2 = random()
return
```

```
function nil OK

return nil


$ Failure to recover the spent fuel pool cooling before boiling.
$ Both offsite power recovery and diesel generator repair take too long.
function real FAIL
  $ Time available before boiling is calculated.
  t_avail = t_boil(Temperature, WLevel)
  boiltime = t_sfpc+t_avail

  if OPRecT < boiltime then
  begin
    p = 0                  $ Offsite power recovery in time
    WLevel = InitWLevel
  end
  else
  begin
    $ Diagnosis time for diesel generator repair
    t_diag = icumul(DGRD,r2)

    $ Is the diagnosis successful before boiling?
    if t_diag < t_avail then
    begin
      $ Probability that repair execution is not performed before boiling
      p = EXP(-(t_avail-t_diag)/sfpcmrt)

      $ Repair time exceeding the boiling time is drawn from exponential distribution.
      $ Boiling time is the 0-point.
      r = 1-r*p
      t_repair = t_diag-LN(1-r)*sfpcmrt+t_sfpc-boiltime

      $ Failure to start probability is added representing the scenario where the repair
      $ is performed in time, but the cooling train does not start. In that case,
      $ another DG repair is assumed with the previously determined repair time.
      p = p + (1-p)*P_ALL_FTS
    end
    else
    begin
      $ The diagnosis is complete after the boiling has started.
      p = 1

      $ Repair time is drawn from exponential distribution.
      $ Boiling time is the 0-point.
      t_repair = t_diag-LN(1-r)*sfpcmrt+t_sfpc-boiltime
    end

    $ t_repair represents the spent fuel pool cooling recovery time.
    $ It is the minimum of the offsite power recovery time and
    $ diesel generator repair time.
    if OPRecT-boiltime < t_repair then t_repair = OPRecT-boiltime

    $ Boiling conditions are the starting point for the next analysis phase.
    Temperature = BoilingTemp
    WLevel = InitWLevel
  end
return p
```

MU:2_HFE (Make up 2 start actions)

```
real prob, r2, r, t_avail, t_diag, t_exe

routine init
  r = random()    $ Random value between 0 and 1
  r2 = random()
return
```

```
$ Make-up 2 start is performed successfully
function nil OK
  $ Time available to start make-up system 2.
  t_avail = t_uncover(WLevel)

  t_avail2 = t_avail $ Collect to results

  $ The execution time of the make-up system 2 start is drawn from uniform distribution.
  t_exe = 2*r2+1

  $ Is there time to make the execution?
  if t_exe < t_avail then
  begin
    $ The diagnosis time of the make-up system 2 start is drawn from lognormal distribution.
    r = r*cumul(MU2D,t_avail-t_exe)
    t_diag = icumul(MU2D,r)

    $ The start time of make-up system 2.
    t_start2 = t_diag+t_exe

    $ The spent fuel pool water level is updated.
    WLevel = newWLevel(WLevel, 0, t_start2)
  end
return nil


$ Execution fails
function real EFAIL
  $ Time available to start make-up system 2.
  t_avail = t_uncover(WLevel)

  $ The execution time of make-up system 2 start is drawn from uniform distribution.
  t_exe = 2*r2+1

  $ Is there time to make the execution?
  if t_exe < t_avail then
  begin
    $ The diagnosis time of make-up system 2 start is drawn from lognormal distribution.
    r = r*cumul(MU2D,t_avail-t_exe)
    t_diag = icumul(MU2D,r)

    $ Time when execution attempt is finished.
    t_start2 = t_diag+t_exe

    $ The spent fuel pool water level is updated.
    WLevel = newWLevel(WLevel, 0, t_start2)

    $ Execution failure probability
    prob = P_EXE2
  end
  else $ No time to start the system
  begin
    prob = 1

    Temperature = BoilingTemp
    WLevel = FuelLevel
    t_start2 = t_avail
  end

  t_mu2 = t_start2

  MU2EFAIL = true
return prob
```

MU:2_P1 (Make up 2 pump)

```
real prob
```

```
routine init

return


function nil OK
  $ Nil-function returns 1-prob
return nil


$ Failure to start
function real FTS
  prob = P_PUMP_FTS

  t_mu2 = t_start2

  mttr1 = MTTR_P_FTS
return prob
```

DG102_FLEX (FLEX diesel generator)

```
real prob, r, t_earliest, fr, t_rec

routine init
  r = random()     $ Random value between 0 and 1
return


function nil OK
  $ Nil-function returns 1-prob
return nil

$ Failure to run
function real FTR
  fr = FR_DG   $ Failure rate

  $ The mission time is tentatively calculated as the time to reach the normal water level.
  t_mission2 = t_restore(WLevel)

  $ Time when the spent fuel pool cooling can theoretically be recovered.
  t_rec = t_repair-t_start2

  $ Does the recovery take longer than reaching the normal water level.
  if t_mission2 < t_rec then
  begin
    $ Given the recovery time of the spent fuel pool cooling,
    $ the earliest allowed failure time is calculated.
    t_earliest = EarliestTime(Temperature,t_rec,WLevel)

    $ If the earliest allowed failure time based on the recovery of the spent fuel pool
cooling
    $ is larger than the time to reach the normal water level, the mission time is
    $ determined based on that.
    if t_mission2 < t_earliest then t_mission2 = t_earliest
  end

  $ The diesel generator failure probability is calculated.
  prob = 1-exp(-fr*t_mission2)

  $ The failure time of the diesel generator is determined.
  t_fail2 = t_mission2*r

  $ The spent fuel pool conditions are updated based on the failure time.
  Temperature = newTemp(Temperature, WLevel, m_makeup, t_fail2)
  WLevel = newWLevel(WLevel, m_makeup, t_fail2)
  if WLevel > InitWLevel then WLevel = InitWLevel
```

```
  $ The total time the make-up 2 system was used.
  t_mu2 = t_start2+t_fail2

  $ Mean time to repair for repair modelling of this diesel generator.
  mttr1 = MTTR_DG_FTR
return prob

$ Failure to start
function real FTS
  prob = P_DG_FTS

  t_mu2 = t_start2

  mttr1 = MTTR_DG_FTS
return prob
```

MU:1 (Make up system 1)

```
real prob, t_boiling, r, r1, r2, r3, t_earliest, t_start, t_mission, t_avail, p_fts, t_fail,
mttr,
     p_ftr, t_st, p_exe, fr, t_exe, r11, r12


routine init
  r = random()    $ Random value between 0 and 1
  r1 = random()
  r2 = random()
  r3 = random()
  r11 = random()
  r12 = random()
return


function nil OK
  $ Nil-function returns 1-prob
return nil

$ Failure to recover the offsite power and repair the diesel generator supplying
$ the spent fuel pool cooling system in time, or
$ bring the water level back to normal by make-up system 1.
$ This function essentially calculates the conditional probability for fuel damage
$ after the failure of make-up system 2.
function real FAIL
  $ Time available for repair/recovery.
  t_boiling = t_boil(Temperature, WLevel)
  t_avail = t_boiling + t_uncover(WLevel)

  p_fts = P_ALL_FTS          $ Failure to start probability of a make-up system 1 train
  p_exe = P_EXE1             $ Make up 1 start execution failure probability
  fr = FR_DG+FR_PUMP         $ Failure rate of a make-up system 1 train

  $ If the power supply is not recovered before fuel damage time,
  $ fuel damage is assumed.
  if (t_avail < t_repair-t_mu2) then
  begin
    prob = 1
  end
  else $ The power supply recovery and make-up 1 start come before fuel damage.
  begin
    $ If boiling is going on or starts before the power supply recovery.
    if (WLevel < InitWLevel) or (t_boiling < t_repair-t_mu2) then
    begin
      $ Duration of make-up 1 start execution
      t_exe = r12+0.5

      $ Is there time to start make-up system 1
      if t_avail > t_exe then
      begin
```

```
$ Four failure modes of make up system 1 are evaluated in the following:
$ start diagnosis failure, start execution failure, failure to start and failure to
run.
$ In each case, also repair possibility is considered.
$ The probabilities of the failure modes (including repair failures) are summed.

$ Failure mode 1: start diagnosis failure
$ ----------------------------------------

$ Probability that diagnosis is not performed in time
prob = 1-cumul(MU1D,t_avail-t_exe)

$ Make-up 1 start time is drawn.
r11 = r11*(1-prob)
t_start1 = icumul(MU1D,r11)+t_exe

$ The spent fuel pool conditions are updated depending on
$ if the system is started before or after boiling.
if t_start1 < t_boiling then
begin
  Temperature = newTemp(Temperature, WLevel, 0, t_start1)
end
else
begin
  Temperature = BoilingTemp
  WLevel = newWLevel(WLevel, 0, t_start1-t_boiling)
end

$ Failure mode 2: start execution failure
$ ----------------------------------------

$ Make-up 1 start execution fails and make-up 2 repair fails or is not possible.
if MU2EFAIL then
begin
  $ Make-up 2 start execution failed, so the system cannot be repaired.
  $ Probability that make-up 1 start execution fails.
  prob = prob+(1-prob)*p_exe
end
else
begin
  $ Probability that make-up 1 start execution fails and repair of make-up 2 fails.
  DG = true
  prob = prob+(1-prob)*p_exe*RepairFail(Temperature, WLevel, mttr1, t_start1)
end

$ Failure mode 3: failure to start
$ --------------------------------

$ Start time for make-up system 1 is determined dependending on
$ the power supply recovery time and the duration of the manual start actions.
t_st = t_repair-t_mu2
if t_st < t_start1 then t_st = t_start1

$ The spent fuel pool conditions are updated.
if t_start1 > t_boiling then
begin
  WLevel = newWLevel(WLevel, 0, t_st-t_start1)
end
else
begin
  WLevel = newWLevel(WLevel, 0, t_st-t_boiling)
end
Temperature = BoilingTemp

$ Does the diesel generator repair or offsite power recovery come earlier?
if t_repair < OPRecT-boiltime then
begin
  $ MTTR depends on whether the pump or DG fails to start.
```

```
          if r2 < CP_DG_FTS then mttr = MTTR_DG_FTS else mttr = MTTR_P_FTS
          DG = true
        end
        else  $ When the offsite power is recovered, both make-up 1 trains are available.
        begin
          $ Both make-up 1 pumps fail (CCF or independent failures)
          p_fts = 3.51E-3
          mttr = MTTR_P_FTS
          DG = false
        end

        $ Probability that make-up 1 fails to start and its repair fails.
        prob = prob+(1-prob)*p_fts*RepairFailLOOP(Temperature, WLevel, mttr, t_st)

        $ Failure mode 4: failure to run
        $ ------------------------------

        $ Mission time for make-up system 1 is the time to normal water level.
        t_mission = t_restore(WLevel)

        $ The failure time of the system is determined.
        t_fail = t_mission*r

        $ The spent fuel pool conditions are updated based on the failure time.
        Temperature = newTemp(Temperature, WLevel, m_makeup, t_fail)
        WLevel = newWLevel(WLevel, m_makeup, t_fail)
        if more(WLevel,InitWLevel) then WLevel = InitWLevel

        $ Does the diesel generator repair or offsite power recovery come earlier?
        if t_repair < OPRecT-boiltime then
        begin
          $ Diesel generator is assumed as the failed component with possibility to repair.
          mttr = MTTR_DG_FTR
          DG = true
        end
        else  $ When the offsite power is recovered, both make-up 1 trains are available.
        begin
          $ Failure rate for CCF of make-up 1 pumps
          fr = 2.67E-7
          mttr = MTTR_P_FTR
          DG = false
        end

        t_mission1 = t_mission  $ Collect to results

        $ Probability that make-up system 1 fails to run and its repair fails.
        prob = prob+(1-prob)*(1-exp(-fr*t_mission))*RepairFailLOOP(Temperature, WLevel, mttr,
t_st+t_fail)
      end
      else
      begin
        $ No time to start make-up 1
        prob = 1
      end
    end
    else
    begin
      $ Power supply recovery comes before boiling, SFPCS operation can be started and safe
state is reached.
      prob = 0
    end
  end

  FD = true
return prob
```

Common section

```
ranseed = 161915

real Temperature,  $ Spent fuel pool temperature
     WLevel,       $ Spent fuel pool water level

     t_mission2,   $ Mission time for FLEX diesel generator
     t_repair,     $ Repair time of the spent fuel pool cooling system
     t_start1,     $ Start time of make up 1
     t_start2,     $ Start time of make up 2
     t_fail2,      $ Failure time of FLEX diesel generator
     mttr1,        $ MTTR for first make up system repair
     mttr2,        $ MTTR for second make up system repair
     sfpcmrt,      $ MTTR for spent fuel pool cooling
     boiltime,     $ Time from SFPC failure to boiling
     uncoverytime, $ Time from boiling to fuel uncovery
     t_mu1,        $ MU1 use time (manual action + operation)
     t_mu2,        $ MU2 use time (manual action + operation)
     OPRecT,       $ Offsite power recovery time in LOOP scenario

     $ Model parameters
     Q_d = 4E+6,             $ Fuel decay heat
     hfg = 2264E+3,          $ Latent heat
     Dab = 3E-5,             $ Diffusion coefficient
     Sc = 0.616,             $ Schmidt number
     Sh = 76.2,              $ Sherwood number
     hc = 7.0404,            $ Heat transfel coefficient
     Cw = 4181,              $ Specific heat of water
     rho = 958,              $ Water density
     rhovs = 0.0828,         $ Vapour surface density
     rhovinf = 0.012,        $ Vapour ambient density
     Epsilon = 0.95,         $ Emissivity
     Sigma = 5.67E-8,        $ StefanBoltzmann constant
     NormalTemp = 35,        $ Normal temperature of the spent fuel pool
     InitWLevel = 10,        $ Normal water level of the spent fuel pool
     BoilingTemp = 100,      $ Boiling temperature
     FuelLevel = 4,          $ The height of the upper part of the fuel (m)
     CoolantTemp = 20,       $ Temperature of the coolant for all systems
     WallTemp = 30,          $ Room wall temperature
     TempHall = 30,          $ Room air temperature
     m_makeup = 20,          $ Amount of make-up water (kg/s)
     m_sfpcs = 20,           $ Amount of water circulated by the spent fuel pool cooling
system (kg/s)
     D = 200,                $ Time-step

     As = 140,               $ Pool surface area
     Ar = 110,               $ Water surface area between fuel racks
     x = 10,                 $ Pool length
     y = 14,                 $ Pool width

     $ Mean time to repair parameters
     MTTR_P_FTS = 12,      $ Pump failure to start
     MTTR_P_FTR = 24,      $ Pump failure to run
     MTTR_DG_FTS = 6,      $ Diesel generator failure to start
     MTTR_DG_FTR = 10,     $ Diesel generator failure to run

     FR_DG = 1.65E-3,      $ Failure rate of diesel generator
     FR_PUMP = 5E-6,       $ Failure rate of pump
     FR_HEX = 1E-6,        $ Failure rate of heat exchanger
     FR_DG_ALL = 6.15E-6,  $ Failure rate of CCF with 4 DGs
     FR_DG_3 = 1.54E-5,    $ Failure rate of CCF with 3 DGs
     P_DG_FTS = 4.52E-3,   $ Failure to start probability of diesel generator
     P_PUMP_FTS = 3.95E-2, $ Failure to start probability of pump
     P_ALL_FTS = 4.40E-2,  $ Failure to start probability of pump and DG
     P_EXE1 = 5E-4,        $ Execution failure probability of make up 1 start
     P_EXE2 = 5E-3,        $ Execution failure probability of make up 2 start
     CP_DG_FTS = 0.103,    $ Conditional FTS prob of DG given that the system fails to start

     rr2, rr3              $ Random variable for timing determination
```

```
boolean FD,          $ Whether fuel damage occurs or not
        MU1EFAIL,    $ Make up 1 start execution failed?
        MU2EFAIL,    $ Make up 2 start execution failed?
        LOOP,        $ Loss of offsite power scenario?
        DG           $ Whether make-up 1 power supply comes from a diesel generator

$ Distributions for diagnosis durations
LOGNOR MU1D = (2, 7.02),    $ Make-up 1
       MU2D = (2, 8.29),    $ Make-up 2
       SFPCD = (2, 3.04),   $ Spent fuel pool cooling system train switching
       DGRD = (3, 20.36),   $ Diesel generator repair
       MURD = (2, 10.09),   $ Make-up repair
       SFPCRD = (2, 18.52) $ Spent fuel pool cooling repair

$ The temperature after specified time is calculated.
$ IT = initial spent fuel pool temperature.
$ IWL = initial water level
$ t = time delay
$ MW = amount of make-up/cooling water, set to 0 if the make-up/cooling system is not used
function real newTemp (real IT, IWL, MW, t)
  real Temp, Mass, Time, m_ev, OF, Q_ev, Q_rad, Q_con, Q_s, Lc, hm, WL

  Temp = IT
  Mass = ((IWL-FuelLevel)*As + Ar*FuelLevel)*rho   $ mass of water above fuel + between fuel
racks
  Time = 0

  Lc = As/(2*(x+y))              $ characteristic length
  hm = Sh*Dab/Lc                 $ mass transfer coefficient
  m_ev = As*hm*(rhovs-rhovinf)   $ evaporation rate
  Q_ev = m_ev*hfg                $ heat loss due to evaporation

  $ Temperature change is calculated in discrete time steps
  while Time < t do
  begin
    Q_rad = As*Epsilon*Sigma*(pow((Temp+273),4)-pow((WallTemp+273),4))  $ radiation
    Q_con = hc*As*(Temp-TempHall)                                       $ convection
    Q_s = Q_ev + Q_rad + Q_con                                          $ total heat loss at
the air-water interface
    OF = MW - m_ev                                                      $ water outflow is
such, that the water mass is constant
    WL = (Mass/rho - Ar*FuelLevel)/As + FuelLevel
    if(WL < InitWLevel) or (MW == 0) then  OF = 0
    Mass = Mass + (MW - OF - m_ev)*D
    Temp = Temp + (Q_d + MW*Cw*(CoolantTemp - Temp) - OF*Cw*Temp - m_ev*Cw*Temp -
Q_s)*D/(Mass*Cw)
    Time = Time+D/3600
  end
return Temp


$ Function that returns water level after specified time.
$ IWL = initial water level
$ t = time delay
$ MW = amount of make-up water, set to 0 if the make-up system is not used
function real newWLevel(real IWL, MW, t)
real Mass, WL, m_ev, Lc, hm
   Mass = ((IWL-FuelLevel)*As + Ar*FuelLevel)*rho    $ mass of water above fuel + between fuel
racks

   Lc = As/(2*(x+y))                $ characteristic length
   hm = Sh*Dab/Lc                   $ mass transfer coef
   m_ev = As*hm*(rhovs-rhovinf)     $ evaporation rate

   if (MW == 0) then m_ev = Q_d/hfg   $ if make-up system is not used, the water is boiling.
Evaporation rate is changed.
```

```
   $update water level
   Mass = Mass +(MW - m_ev)*t*3600

   WL = (Mass/rho - Ar*FuelLevel)/As + FuelLevel  $ water level

return WL


$ function to calculate start of boiling
$ IT = initial temperature
$ IWL = initial water level
function real t_boil(real IT, IWL)
   real Mass, H, Temp, Time, m_ev, Lc, hm, Q_ev, Q_rad, Q_con, Q_s

   Time = 0
   H = IWL
   Mass = ((H-FuelLevel)*As + Ar*FuelLevel)*rho
   Temp = IT

   Lc = As/(2*(x+y))              $ characteristic length
   hm = Sh*Dab/Lc                 $ mass transfer coef
   m_ev = As*hm*(rhovs-rhovinf)   $ evaporation rate
   Q_ev = m_ev*hfg                $ heat loss due to evaporation

   while  Temp < BoilingTemp do
   begin
    Q_rad = As*epsilon*sigma*(pow((Temp+273),4)-pow((WallTemp+273),4))  $ radiation
    Q_con = hc*As*(Temp-TempHall)                                       $ convection
    Q_s = Q_ev + Q_rad + Q_con                                          $ total heat loss at
the air-water interface
    Mass = Mass - m_ev*D
    Temp = Temp +(Q_d - m_ev*Cw*Temp - Q_s)*D/(Mass*Cw)
    Time = Time + D/3600
   end

return Time

$ function to calculate the time when water level reaches fuel level, if no make-up water is
added
$ WL = initial water level
function real t_uncover(real WL)
   real Mass, H, Time, m_ev, mass_at_fuel

   Time = 0
   H = WL
   Mass = ((H-FuelLevel)*As + Ar*FuelLevel)*rho  $ mass of water above fuel + between fuel
racks
   m_ev = Q_d/hfg                                $ evaporation rate
   mass_at_fuel = Ar*FuelLevel*rho

   $ mass is updated until top of fuel is uncovered
   while Mass > mass_at_fuel do
   begin
     $ update mass
     Mass = Mass - m_ev*D

     $ time of uncover is returned
     Time = Time + D/3600
   end
return Time

$ Function to calculate how long it takes to reach normal water level with make-up system in
operation
$ WL = initial water level
function real t_restore(Real WL)
   real Mass, H, Time, m_ev, Lc, hm

   Time = 0
```

```
   H = WL
   Mass = ((H-FuelLevel)*As + Ar*FuelLevel)*rho
   Lc = As/(2*(x+y))                              $ characteristic length
   hm = Sh*Dab/Lc                                 $ mass transfer coef
   m_ev = As*hm*(rhovs-rhovinf)                   $ evaporation rate

   while H < InitWLevel do
   begin
    $ update mass
    Mass = Mass + (m_makeup - m_ev)*D

    $ calculate water level
    H = (Mass/rho - Ar*FuelLevel)/As + FuelLevel

    $ time of restored level is returned
    Time = Time + D/3600
   end
return Time

$ The earliest allowed failure time given the spent fuel pool cooling system repair time is
calculated.
$ The failure can occur before the repair if the temperature is below 100, because there is
still some
$ time before the boiling starts.
$ IT = initial spent fuel pool temperature.
$ RT = repair time of the spent fuel pool cooling system.
$ IWL = initial water level
function real EarliestTime (real IT, RT, IWL)
  real Temp, Time, WL, t_boiling

  Temp = IT
  Time = 0
  WL = IWL

  $ The earliest allowed failure time is reached when the temperature is such that boiling
could not start
  $ before the spent fuel pool cooling system repair.

  t_boiling = t_boil(Temp, WL)
  while Time + t_boiling < RT do
  begin
    Temp = newTemp(Temp, WL, 0, D/3600)
    WL = newWLevel(WL, m_makeup, D/3600)
    $ the time it takes for the water to boil is updated based on temperature and water level
change
    t_boiling = t_boil(Temp,WL)
    Time = Time + D/3600
  end
return Time


$ Failure probability of a repair is calculated.
$ Failure to start or run after the repair are also included in the probability.
function real RepairFail(real Temp, Level, mrt, t_mu1r)
  real t_b, t_ava, t_st, t_miss, t_earl, p, p2, p_fts, fr

  $ Failure mode 1: repair failure
  $ ------------------------------

  $ Time available for repair.
  t_b = t_boil(Temp, Level)
  t_ava = t_b + t_uncover(Level)

  $ Probability that diagnosis takes too long
  p = 1-cumul(MURD,t_ava)

  $ Diagnosis time is drawn given that it is performed in time
  rr3 = rr3*(1-p)
```

```
  t_st = icumul(MURD,rr3)

  $ The repair execution failure probability is calculated assuming exponential distribution
  $ for the repair time.
  p2 = EXP(-(t_ava-t_st)/mrt)

  $ The repair time is drawn from exponential distribution.
  rr2 = rr2*(1-p2)
  t_st = t_st-LN(1-rr2)*mrt

  $ Total repair failure probability
  p = p + (1-p)*p2

  $ The spent fuel pool conditions are updated depending on
  $ if the system is started before or after boiling.
  if t_st < t_b then
  begin
    Temp = newTemp(Temp, Level,0, t_st)
  end
  else
  begin
    Temp = BoilingTemp
    Level = newWLevel(Level, 0, t_st-t_b)
  end

  $ Failure mode 2: failure to start
  $ ---------------------------------

  $ Failure to start probability of the make up system.
  $ The probability depends on whether the power supply comes from the grid or a diesel
generator.
  p_fts = P_ALL_FTS $ failure to start probability of DG and pump
  if not(DG) then p_fts = P_PUMP_FTS

  p = p+(1-p)*p_fts

  $ Failure mode 3: failure to run
  $ ------------------------------

  $ Failure rate is defined.
  $ It depends on whether the power supply comes from the grid or a diesel generator.
  fr = FR_DG+FR_PUMP
  if not(DG) then fr = FR_PUMP

  $ The mission time is tentatively calculated as the time to reach the normal water level.
  t_miss = t_restore(Level)

  $ Time when the spent fuel pool cooling can theoretically be recovered.
  t_earl = t_repair-t_st-t_mu1-t_mu2-t_mu1r

  $ Does the recovery take longer than reaching the normal water level.
  if t_miss < t_earl then
  begin
    $ Given the recovery time of the spent fuel pool cooling,
    $ the earliest allowed failure time is calculated.
    t_earl = EarliestTime(Temp,t_earl,Level)

    $ If the earliest allowed failure time based on the recovery of the spent fuel pool
cooling
    $ is larger than the time to reach the normal water level, the mission time is
    $ determined based on that.
    if t_miss < t_earl then t_miss = t_earl
  end

  $ The failure to run probability is calculated.
  p = p+(1-p)*(1-exp(-fr*t_miss))
return p
```

```
$ Failure probability of a repair is calculated.
$ Failure to start or run after the repair are also included in the probability.
$ This function is for LOOP cases, where make-up 1 is repaired.
function real RepairFailLOOP(real Temp, Level, mrt, t_mu1r)
  real t_b, t_ava, t_st, t_miss, t_earl, p, p2, p_fts, fr, t_rec

  $ Failure mode 1: repair failure
  $ ------------------------------

  $ Time available for repair.
  t_b = t_boil(Temp, Level)
  t_ava = t_b + t_uncover(Level)

  $ Relative offsite power recovery time
  t_rec = OPRecT-boiltime-t_mu1-t_mu2-t_mu1r

  $ Is the offsite power recovery still coming before fuel damage?
  if Not(DG) or (t_ava < t_rec) then   $ No, too late or has already came
  begin
    $ Probability that diagnosis takes too long
    p = 1-cumul(MURD,t_ava)

    $ Diagnosis time is drawn given that it is performed in time
    rr3 = rr3*(1-p)
    t_st = icumul(MURD,rr3)

    $ The repair execution failure probability is calculated assuming exponential distribution
    $ for the repair time.
    p2 = EXP(-(t_ava-t_st)/mrt)

    $ The repair time is drawn from exponential distribution.
    rr2 = rr2*(1-p2)
    t_st = t_st-LN(1-rr2)*mrt

    $ Total repair failure probability
    p = p + (1-p)*p2
  end
  else   $ Offsite power recovery comes before fuel damage
  begin
    $ No repair is modelled, only the offsite power recovery.
    $ It is conservatively assumed that no repair is performed before the recovery.
    $ Pump failure modelling in the following is also conservative.
    p = 0
    t_st = t_rec
    DG = false
  end

  $ The spent fuel pool conditions are updated depending on
  $ if the system is started before or after boiling.
  if t_st < t_b then
  begin
    Temp = newTemp(Temp, Level, 0, t_st)
  end
  else
  begin
    Temp = BoilingTemp
    Level = newWLevel(Level, 0, t_st-t_b)
  end

  $ Failure mode 2: failure to start
  $ --------------------------------

  $ Failure to start probability of the make-up system 1.
  $ The probability depends on whether the power supply comes from the grid or a diesel
generator.
  p_fts = P_ALL_FTS $ failure to start probability of DG and pump
  if not(DG) then p_fts = P_PUMP_FTS
```

```
  p = p+(1-p)*p_fts

  $ Failure mode 3: failure to run
  $ -------------------------------

  $ Failure rate is defined.
  $ It depends on whether the power supply comes from the grid or a diesel generator.
  fr = FR_DG+FR_PUMP
  if not(DG) then fr = FR_PUMP

  $ The mission time is calculated as the time to reach the normal water level.
  t_miss = t_restore(Level)

  $ The failure to run probability is calculated.
  p = p+(1-p)*(1-exp(-fr*t_miss))
return p
```

DocuSign

Certificate Of Completion

Envelope Id: 8D0F5D7C4D6F44D0BF8E3BD50F044AE8

Status: Completed

Subject: Please DocuSign: VTT-R-00016-22 Simulation-based PRA for spent fuel pool.docx

Source Envelope:

Document Pages: 49

Signatures: 1

Certificate Pages: 1

Initials: 0

AutoNav: Enabled

EnvelopeId Stamping: Enabled

Time Zone: (UTC+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius

Envelope Originator:
Anne Räsänen
Vuorimiehentie 3, Espoo,
., . P.O Box1000,FI-02044
Anne.Rasanen@vtt.fi
IP Address: 88.148.189.30

Record Tracking

Status: Original
22 February 2022 | 15:47

Holder: Anne Räsänen
Anne.Rasanen@vtt.fi

Location: DocuSign

Signer Events	Signature	Timestamp
Nadezhda Gotcheva nadezhda.gotcheva@vtt.fi Research Team Leader Security Level: Email, Account Authentication (None), Authentication	DocuSigned by: Nadezhda Gotcheva E21E683840FD424... Signature Adoption: Pre-selected Style Using IP Address: 130.188.17.16	Sent: 22 February 2022 | 15:49 Viewed: 22 February 2022 | 16:27 Signed: 22 February 2022 | 16:30

Authentication Details
SMS Auth:
Transaction: 65FC874F21C012049192A156516AA505
Result: passed
Vendor ID: TeleSign
Type: SMSAuth
Performed: 22 February 2022 | 16:27
Phone: +358 40 1326030

Electronic Record and Signature Disclosure:
Not Offered via DocuSign

In Person Signer Events	Signature	Timestamp

Editor Delivery Events	Status	Timestamp

Agent Delivery Events	Status	Timestamp

Intermediary Delivery Events	Status	Timestamp

Certified Delivery Events	Status	Timestamp

Carbon Copy Events	Status	Timestamp

Witness Events	Signature	Timestamp

Notary Events	Signature	Timestamp

Envelope Summary Events	Status	Timestamps
Envelope Sent	Hashed/Encrypted	22 February 2022 | 15:49
Certified Delivered	Security Checked	22 February 2022 | 16:27
Signing Complete	Security Checked	22 February 2022 | 16:30
Completed	Security Checked	22 February 2022 | 16:30

Payment Events	Status	Timestamps