Simulation-based PRA for spent fuel pool

Tyrväinen, Tero; Immonen, Essi

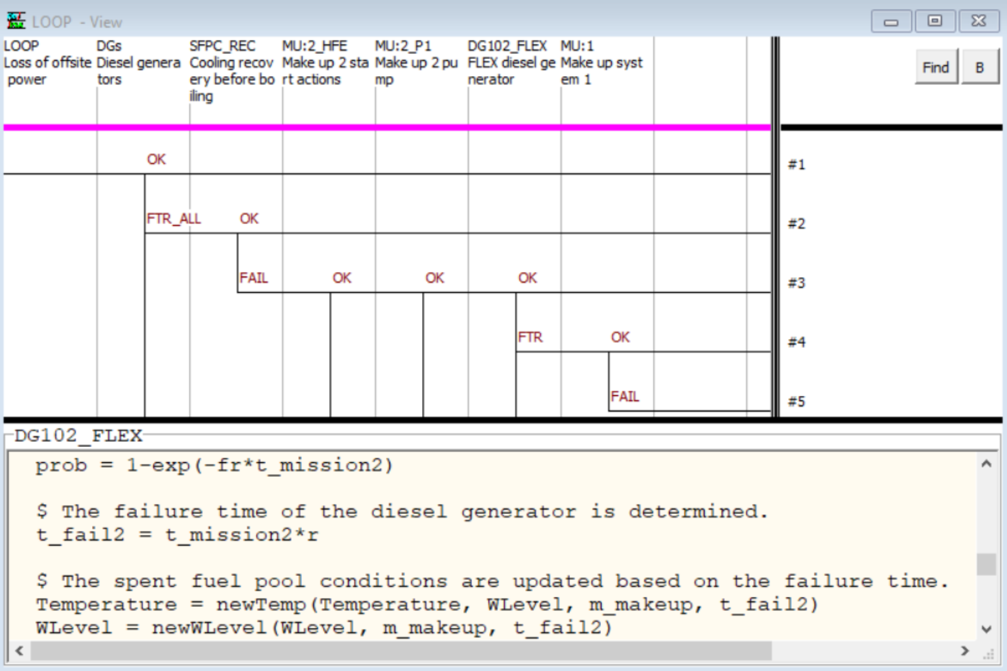
Published: 14/12/2022

Please cite the original version:
Tyrväinen, T., & Immonen, E. (2022). *Simulation-based PRA for spent fuel pool: Final report*. VTT Technical Research Centre of Finland. VTT Research Report No. VTT-R-00990-22

VTT
http://www.vtt.fi
P.O. box 1000FI-02044 VTT
Finland

RESEARCH REPORT

VTT-R-00990-22


LOOP - View
LOOP Loss of offsite power
DGs Diesel generators
SFPC_REC Cooling recovery before boiling
MU:2_HFE Make up 2 start actions
MU:2_P1 Make up 2 pump
DG102_FLEX FLEX diesel generator
MU:1 Make up system 1
Find
B
OK
FTR_ALL
OK
FAIL
OK
OK
OK
FTR
OK
FAIL
#1
#2
#3
#4
#5
DG102_FLEX

```
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)
```

Simulation-based PRA for spent fuel pool: Final report

Authors: Tero Tyrväinen, Essi Immonen

Confidentiality: VTT Public

VTT

beyond the obvious

Report's title Simulation-based PRA for spent fuel pool: Final report	
Customer, contact person, address VYR	**Order reference** SAFIR 3/2022
Project name New developments and applications of PRA	**Project number/Short name** 131764/NAPRA
Author(s) Tero Tyrväinen, Essi Immonen	**Pages** 31/30
Keywords Probabilistic risk assessment, spent fuel pool, simulation	**Report identification code** VTT-R-00990-22

Summary

This report presents an approach to simulation-based PRA of a spent fuel pool. Simulation-based event tree models are developed to analyse loss of offsite power (LOOP) and transient scenarios of a fictive spent fuel pool. In the simulation-based event tree, event 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 models are used to quantify minimal cut sets of a static PRA model more realistically. The results of dynamic and static analyses are compared. The dynamic analysis decreases the frequencies of many 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 following manual actions. The results also indicate that crediting repairs can greatly decrease the frequencies.

As an alternative to Monte Carlo simulation, discretization of time distributions is investigated as a method to select the simulation cases in order to reduce the computation time. One sequence from both LOOP and transient models is analysed using the discretization. In the transient case, the approach produces only slightly conservative result with a small number of simulation cycles. On the other hand, in the LOOP case, the result is very conservative even with a very dense discretization, because the tails of the offsite power recovery time and diesel generator repair time distributions dominate the result. The usefulness of the time discretization approach seems to be case-specific.

Confidentiality	VTT Public

Espoo 14.12.2022

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:	14 December 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. to 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 (Tyrväinen et al., 2021; Tyrväinen & Immonen, 2022). 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 with a deterministic model, 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 so far, it has been 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.

The development of the simulation-based event tree models for loss of offsite power and transient scenarios of a spent fuel pool is continued in this report. The loss of offsite power model is improved, compared to (Tyrväinen & Immonen, 2022), regarding modelling of late (but not too late) offsite power recovery. The transient model is improved regarding the modelling of spent fuel pool cooling system repair before boiling. Comparison between static and dynamic analyses is performed in detail.

The models have been solved so far by simple Monte Carlo simulation, which is not the most efficient technique. In dynamic event tree analyses, one popular approach to limit the computation time is to discretize the distributions of timing variables, such as operator response time (Karanki & Dang, 2016). Therefore, in this report, possible benefits of discretization of time distributions are studied. To enable that, selected accident sequences are implemented in Matlab.

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. The deterministic model of the spent fuel pool is briefly described in Section 4. The static PRA model that is the basis for the simulation-based analysis is presented in Section 5. Loss of offsite power scenario is analysed in Section 6, and transient scenario is analysed in Section 7. Discretization of time distributions is investigated in Section 8. Software tool development possibilities are discussed in Section 9. Section 10 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 practice, 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.

Simulation-based event trees include 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 with scope limited to 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 are readily available (while others can be programmed). A set of built-in functions is available, including some probability distribution operations.

To account for uncertainties related to parameter 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 framework 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 water 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).

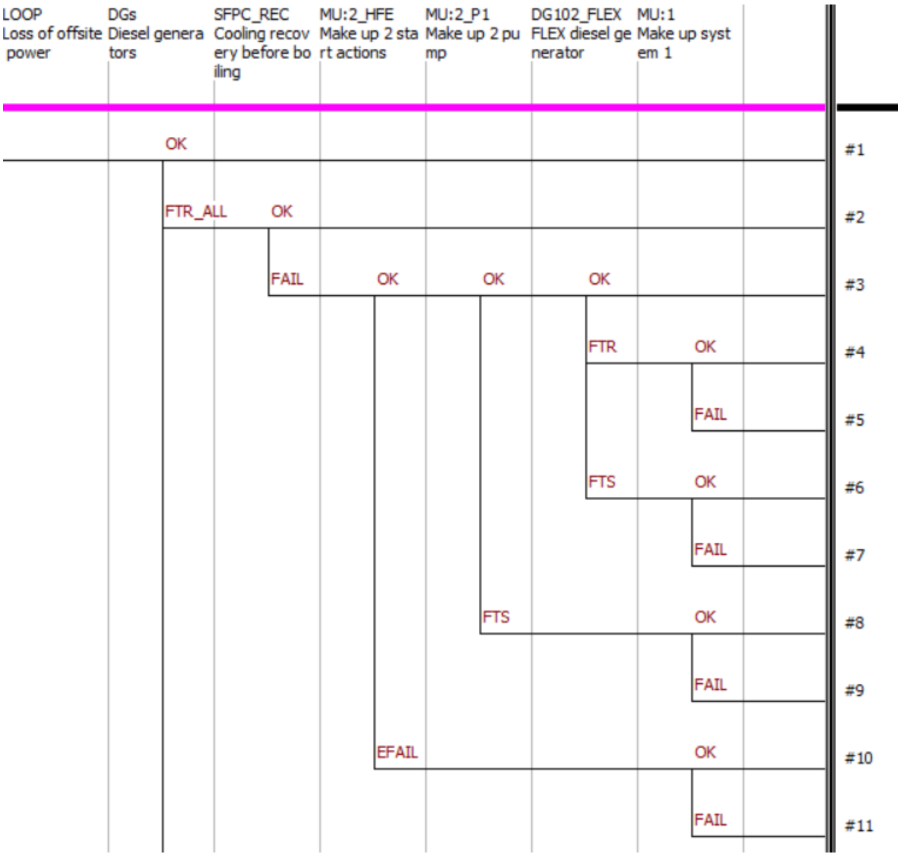


Figure 1: Upper part of the simulation-based event tree for loss of offsite power.

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 9.

The computation scripts related to the event tree are presented in Appendix C. 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, which is presented in Appendix E. The models for spent fuel pool water level and temperature are discussed in the next section.

The failure to run probability of the diesel generator that 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)

  $ Does the recovery take longer than reaching the normal water level.
  if t_mission2 < t_rec-t_start2 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-t_start2,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 recovery. 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 considering 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. Deterministic model of the spent fuel pool

The simulation-based model includes a deterministic model of the spent fuel pool water inventory. In the simulation-based event trees, the water level and temperature are updated at certain time points by the deterministic model. Such time points include the start times of safety systems and failure times of safety systems. Other timings that are calculated are the time when the pool water starts to boil, the time when the top of the fuel is uncovered and the time it takes to reach normal pool water level when a make-up system is in use. There is also a function that calculates the mission time for a make-up system based on when the pool reaches such condition that it cannot start boiling again before a given recovery time of the spent fuel pool cooling system. The scripts of the deterministic model are presented in Appendix E.

The deterministic modelling approach was chosen based on a survey of literature on thermal-hydraulic models of spent fuel pools. The zero-dimensional model of Ramadan et al. (2018) was considered simple but realistic enough to be implemented in simulation-based event tree scripts. The zero-dimensional model divides the spent fuel pool into a water zone and a humid air zone above the pool and studies the heat and mass transfer between the zones. Compared to other methods like computational fluid dynamics, the approach is significantly less computationally demanding. Some simplifications were made compared to the reference model (Ramadan et al., 2018), 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 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. 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 time-dependent behaviour of the temperature and water level is presented in Figures 2 and 3 for the case in which a make-up system is started when water level decreases to the top of the fuel assemblies, after 6.3 days.

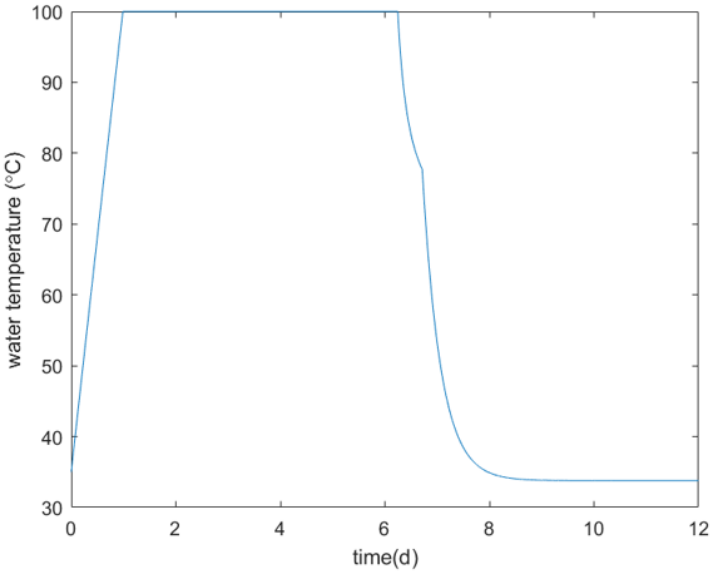


Figure 2: Water temperature as a function of time (make-up system is started at t = 6.3 d).

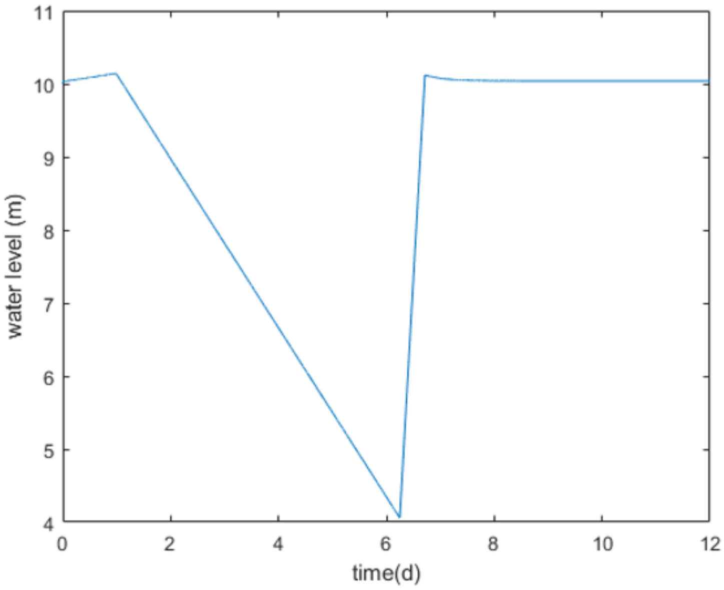


Figure 3: Water level as a function of time (make-up system is started at t = 6.3 d).

5. Static PRA model

The basis for the simulation-based analysis is a static PRA model developed in the PROSAFE project (Tyrväinen et al., 2020; 2021). Event trees for spent fuel pool transient and loss of offsite power (LOOP) are included in this study.

The event tree for transient is presented in Figure 4. One train of the spent fuel pool cooling system is in operation during normal operation. If it fails, there are three other trains in the system that can be used. 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. If the spent fuel pool cooling system and the make-up systems fail, the result is a fuel damage.

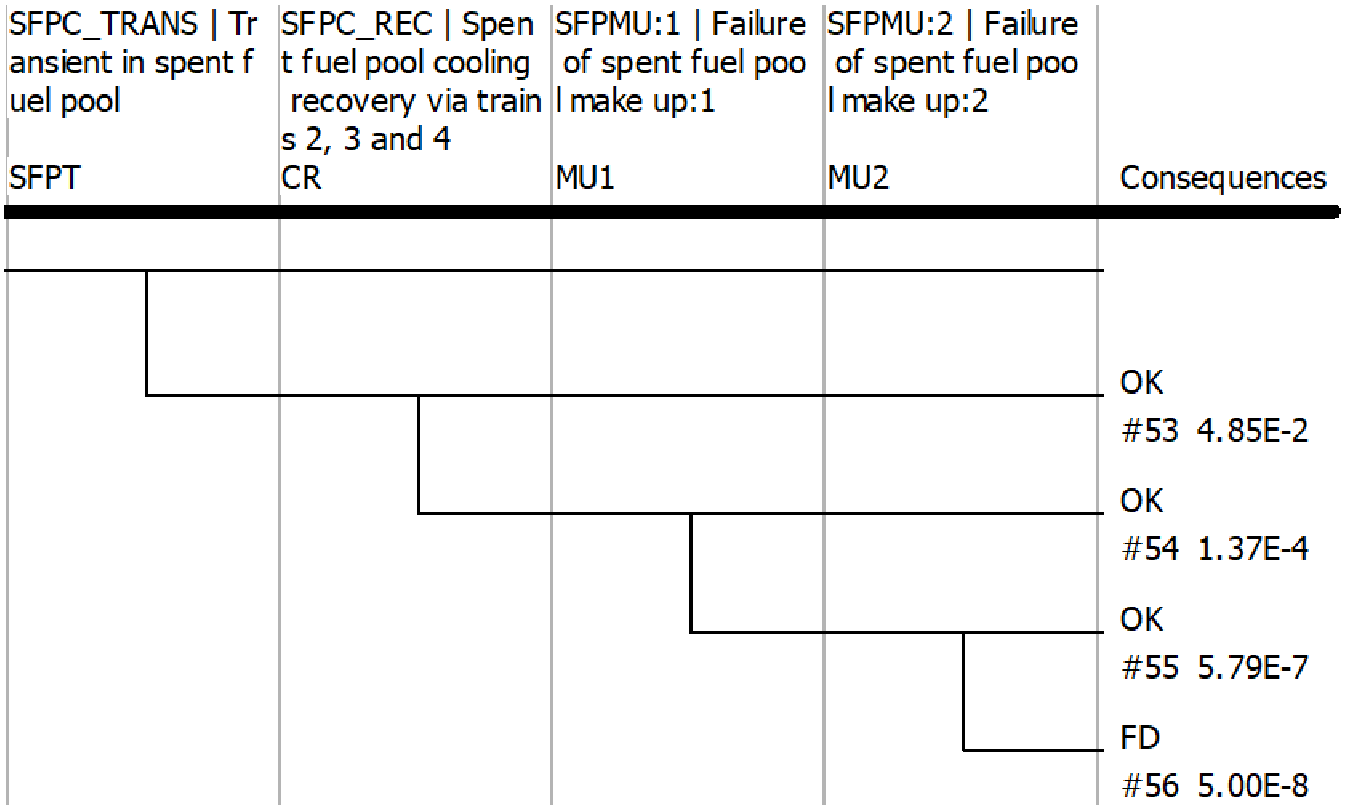

SFPC_TRANS | Transient in spent fuel pool
SFPT
SFPC_REC | Spent fuel pool cooling recovery via trains 2, 3 and 4
CR
SFPMU:1 | Failure of spent fuel pool make up:1
MU1
SFPMU:2 | Failure of spent fuel pool make up:2
MU2
Consequences
OK
#53 4.85E-2
OK
#54 1.37E-4
OK
#55 5.79E-7
FD
#56 5.00E-8

Figure 4: Event tree for transient.

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 17 most important minimal cut sets for transient are presented in Table 1 and a longer list is presented in Appendix A. Failure of the spent fuel pool cooling pump is the initiating event in minimal cut set 1-16. The initiating event of minimal cut set 17 is failure of the heat exchanger. Failure to start events of pumps (common cause failures (CCFs) in some cases) are in general the most important basic events for all systems. Failure to run event of the FLEX diesel generator also appears in many minimal cut sets. Human failure events (execution and diagnosis) to switch the train of the spent fuel pool cooling system and to start make-up system 1 are other important basic events.

Table 1: Top minimal cut sets for transient.

Mc_num	Freq	Basic event names				
1	3.73E-09	SFPC_P1____I___D	CCF-SFPC-PM--A-BCD	CCF-SFPM1-PM-A-AB	SFPMU:2_P1_____A	
2	3.67E-09	SFPC_P1____I___D	ACP__DG102_FLEX2___D	CCF-SFPC-PM--A-BCD	CCF-SFPM1-PM-A-AB	
3	3.36E-09	SFPC_P1____I___D	CCF-SFPM1-PM-A-AB	SFPC_MANSTART_H	SFPMU:2_P1_____A	
4	3.31E-09	SFPC_P1____I___D	ACP__DG102_FLEX2___D	CCF-SFPM1-PM-A-AB	SFPC_MANSTART_H	
5	2.47E-09	SFPC_P1____I___D	CCF-SFPC-PM--A-BCD	SFPMU:1_P1_____A	SFPMU:1_P2_____A	SFPMU:2_P1_____A
6	2.43E-09	SFPC_P1____I___D	ACP__DG102_FLEX2___D	CCF-SFPC-PM--A-BCD	SFPMU:1_P1_____A	SFPMU:1_P2_____A
7	2.23E-09	SFPC_P1____I___D	SFPC_MANSTART_H	SFPMU:1_P1_____A	SFPMU:1_P2_____A	SFPMU:2_P1_____A
8	2.19E-09	SFPC_P1____I___D	ACP__DG102_FLEX2___D	SFPC_MANSTART_H	SFPMU:1_P1_____A	SFPMU:1_P2_____A
9	1.51E-09	SFPC_P1____I___D	CCF-SFPM1-PM-A-AB	SFPC_DIAG______H	SFPMU:2_P1_____A	
10	1.49E-09	SFPC_P1____I___D	ACP__DG102_FLEX2___D	CCF-SFPM1-PM-A-AB	SFPC_DIAG______H	
11	1.00E-09	SFPC_P1____I___D	SFPC_DIAG______H	SFPMU:1_P1_____A	SFPMU:1_P2_____A	SFPMU:2_P1_____A
12	9.85E-10	SFPC_P1____I___D	ACP__DG102_FLEX2___D	SFPC_DIAG______H	SFPMU:1_P1_____A	SFPMU:1_P2_____A
13	8.85E-10	SFPC_P1____I___D	CCF-SFPC-PM--A-BCD	SFPMU:1_MANSTART_H	SFPMU:2_P1_____A	
14	8.70E-10	SFPC_P1____I___D	ACP__DG102_FLEX2___D	CCF-SFPC-PM--A-BCD	SFPMU:1_MANSTART_H	
15	7.98E-10	SFPC_P1____I___D	SFPC_MANSTART_H	SFPMU:1_MANSTART_H	SFPMU:2_P1_____A	
16	7.84E-10	SFPC_P1____I___D	ACP__DG102_FLEX2___D	SFPC_MANSTART_H	SFPMU:1_MANSTART_H	
17	7.47E-10	SFPC_H1____I___X	CCF-SFPC-PM--A-BCD	CCF-SFPM1-PM-A-AB	SFPMU:2_P1_____A	

Meanings of basic event names:
SFPC_P1____I___D: Failure to run of the spent fuel pool cooling system pump (initiating event)
SFPC_H1____I___X: Failure of the spent fuel pool cooling system heat exchanger (initiating event)
CCF-SFPC-PM--A-BCD: CCF to start spent fuel pool cooling system pumps
CCF-SFPM1-PM-A-AB: CCF to start make-up system 1 pumps
SFPMU:1_PX_____A: Failure to start pump X of make-up system 1
SFPMU:2_P1_____A: Failure to start the pump of make-up system 2
ACP__DG102_FLEX2___D: Failure to run of FLEX diesel generator
SFPC_MANSTART_H: Human failure to execute switching of the train of the spent fuel pool cooling system
SFPC_DIAG______H: Human failure to diagnose the need to switch the train of the spent fuel pool cooling system
SFPMU:1_MANSTART_H: Human failure to execute start of make-up system 1

The event tree for spent fuel pool LOOP is presented in Figure 5. 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. 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.

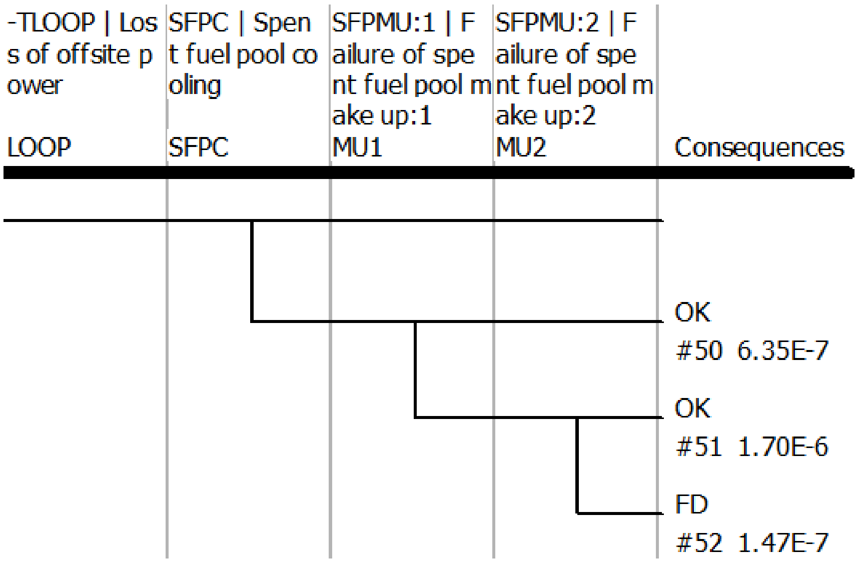

-TLOOP | Loss of offsite power
LOOP
SFPC | Spent fuel pool cooling
SFPC
SFPMU:1 | Failure of spent fuel pool make up:1
MU1
SFPMU:2 | Failure of spent fuel pool make up:2
MU2
Consequences
OK
#50 6.35E-7
OK
#51 1.70E-6
FD
#52 1.47E-7

Figure 5: Event tree for loss of offsite power.

The 15 most important minimal cut sets are presented in Table 2. In all top minimal cut sets for the fuel damage, the spent fuel pool cooling fails mainly due to failures of the gas turbine and emergency diesel generators. The most important failure modes are failure to start for the gas turbine and 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 spent fuel pool cooling pump 2 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 2: Top minimal cut sets for loss of offsite power.

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

Meanings of basic event names:
!IE-LOOP: Loss of offsite power
ACN10GT001_________A: Failure to start gas turbine
ACN10GT001_________M: Gas turbine under maintenance
ACP-DG--------D-ALL: CCF to run between all emergency diesel generators
ACP-DG--------A-ALL: CCF to start between all emergency diesel generators
ACP-DG--------D-3XX: CCF to run between three emergency diesel generators
SFPMU:2_P1_________A: Failure to start the pump of make-up system 2
ACP__DG102_FLEX2___D: Failure to run of FLEX diesel generator
ACP__DG102_FLEX2___A : Failure to start FLEX diesel generator
SFPMU:2_MANSTART___H: Human failure to start make-up system 2
SFPC_P2____________A: Failure to start spent fuel pool cooling system pump 2
ACPX0DG001_________D: Failure to run of an emergency diesel generator

6. Loss of offsite power analysis

6.1 Simulation-based event tree

The simulation-based event tree for the LOOP scenario is partly presented in Figure 1. For diesel generators, the first branch 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.

Different modelling aspects are discussed in the following sections. The scripts of the model are presented in Appendix C.

6.1.1 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 for 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.

6.1.2 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 at all. 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.

6.1.3 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 of the manual start, 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 the operation time of the system.

6.1.4 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.

6.1.5 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 3. For the executions of the start actions, uniform distributions are used, and the durations are assumed quite short, regardless of 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 3: 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 4.

Table 4: 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
Diagnosis for spent fuel pool cooling system repair (modelled only in the transient case)	Lognormal	Mean = 2h, Error factor = 18.52
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
Repair execution for a heat exchanger (modelled only in the transient case)	Exponential	Mean = 11h

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.

6.2 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 5 presents the fuel damage frequencies of the top minimal cut sets presented in Table 2. 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 5: Fuel damage frequencies (1/year) of top minimal cut sets for loss of offsite power.

MCS	Seq	Static	Refined static	Dynamic without MU repair	Dynamic with MU repair
Total		1.14E-07	2.09E-11	2.02E-11	1.92E-12
1	9	2.36E-08	4.49E-12	6.55E-12	7.21E-13
2	5	2.32E-08	4.41E-12	6.27E-13	3.91E-14
3	9	1.09E-08	2.07E-12	3.03E-12	3.33E-13
4	5	1.07E-08	2.04E-12	2.89E-13	1.80E-14
5	11	2.99E-09	5.69E-13	8.33E-13	1.45E-13
6	19	2.74E-09	2.84E-13	2.38E-13	2.84E-14
7	15	2.70E-09	2.80E-13	1.66E-14	9.69E-16
8	7	2.70E-09	5.14E-13	7.17E-13	7.91E-14
9	29	2.03E-09	3.86E-13	5.64E-13	6.22E-14
10	39	2.00E-09	3.80E-13	9.81E-13	5.52E-14
11	39	2.00E-09	3.80E-13	9.81E-13	5.52E-14
12	39	2.00E-09	3.80E-13	9.81E-13	5.52E-14
13	39	2.00E-09	3.80E-13	9.81E-13	5.52E-14
14	25	2.00E-09	3.80E-13	5.43E-14	3.37E-15
15	35	1.97E-09	3.75E-13	7.69E-14	4.19E-15

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.

The refined results calculated by static approach are much closer to the results from the simulation-based approach. In fact, the total fuel damage frequency of the refined static analysis is very close to the dynamic quantification without make-up system repair. However, detailed examination shows also significant differences:

- The frequencies of minimal cut sets with the failure to run event of the FLEX diesel generator (2, 4, 7, 14 and 15) are much smaller based on the dynamic quantification. The reasons for this are that the mission time in simulations is on average only 1.5 hours, which is much shorter than 24 hours, and that the operation of make-up system 2 gives more time to recover the power supply and start make-up system 1.
- The frequencies of many other minimal cut sets are actually larger based on the dynamic quantification. The reason for this is that all mission times depend on the recovery time of the offsite power. Therefore, the mission times are dependent, and the combined probability of the failure to run events is larger than their product, which is calculated in the static analysis. This effect is particularly strong for minimal cut sets 10-13 that include combination of a failure to run CCF of emergency diesel generators and independent failure to run event of an emergency diesel generator.
- The mission time of make-up system 1 is on average only 1.3 hours, which is much shorter than 24 hours. This decreases the frequencies of all minimal cut sets compared to the static analysis.
- In the cases of failure to run CCFs of diesel generators, crediting the operation of the spent fuel pool cooling system with a diesel generator before it fails decreases the frequencies of the corresponding minimal cut sets, because it gives more time to recover the offsite power.

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, approximately 91%. The reason is that the available time for the repair is very long on average. Modelling of another repair would decrease the result even more, because component failures occurring after repair have much larger risk contribution than actual repair failures.

The significances of dynamic effects are explored in sensitivity analyses presented in Table 6. 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 5, because much less simulation cycles have been used to reduce the computation time. The same seed number is applied 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 6: Sensitivity analyses for loss of offsite power (without make-up system repair).

Case	Total	MCS 2
1. Baseline	1.83E-11	5.44E-13
2. Make-up system 1 mission time set to 24 hours	2.74E-11 (150%)	8.04E-13 (148%)
3. Failure to run events of emergency diesel generators occur at start	8.84E-11 (484%)	1.25E-12 (230%)
4. Make-up system 2 mission time set to 24 hours	2.68E-11 (147%)	2.88E-12 (530%)
5. Failure to run event of the FLEX diesel generator occurs at start	2.05E-11 (112%)	1.74E-12 (320%)
6. The mission times of the emergency diesel generators set to 24 hours	6.54E-12 (36%)	1.33E-13 (24%)
7. All mission times set to 24 hours	1.06E-11 (58%)	1.80E-12 (331%)
8. All mission times set to 24 hours and all failures occur at start	2.09E-11 (115%)	4.25E-12 (780%)
9. All mission times set to their average values	6.65E-12 (36%)	1.12E-13 (21%)
10. Mean diagnosis times are doubled	2.57E-11 (141%)	8.82E-13 (162%)
11. Mean diagnosis times are halved	1.53E-11 (84%)	4.54E-13 (83%)
12. Execution times of manual actions are doubled (except for repair)	1.92E-11 (105%)	4.99E-13 (92%)

Sensitivity analysis cases 2-9 involve removal of some dynamic effects from the simulation model. In the baseline model, the mission times of make-up systems are so short on average that the result increases when they are set to 24 hours in cases 2 and 4, even though the related mission time dependency is removed. Cases 3 and 5 show the significance of the crediting operation of the cooling system and make-up system 2 before failure to run events occur. Particularly, the operation of the cooling system with a diesel generator has a large risk decreasing impact, because it gives more time to recover the offsite power before boiling. Cases 6, 7 and 9 show how the risk decreases when the mission time dependencies are removed. When all mission times are set to 24 hours and all failures are assumed to occur at start in case 8, the results are approximately the same as in the static analysis, because the dynamic effects have been removed from the model.

It can be concluded that it is partly a coincidence that the total results of the static and dynamic quantification are so close to each other. The reason for this is that the risk increasing effects and risk decreasing effects in the dynamic analysis cancel each other. The main risk increasing effect is the mission time dependency. The risk decreasing effects are crediting operation of the cooling system and make-up system 2 before failure, and shorter mission times of make-up systems.

Sensitivity cases 10-12 concern durations of manual actions. Changes in diagnosis times have quite significant impact on the results, but the order of magnitude of the results is not changed. Change in execution times has smaller impact. The frequency of MCS 2 actually decreases when execution times are increased, which seems counterintuitive. The reason for this is that the mission time of the FLEX diesel generator is on average shorter when it is started later, because there is shorter time until the recovery of the normal power supply. All in all, uncertainties related to durations of manual actions seem to have quite moderate impact on the results.

7. Transient analysis

7.1 Simulation-based event tree

The simulation-based event tree for transient is partly presented in Figure 6. The first section contains branches for different initiating events. In the second section, different failure modes of the standby trains of the spent fuel pool system are modelled. FTS means the CCF to start the pumps. The third section concerns repair of the spent fuel pool cooling before boiling. After that there are several branches related to different failures of the make-up systems. In the final section, make-up system repair is modelled.

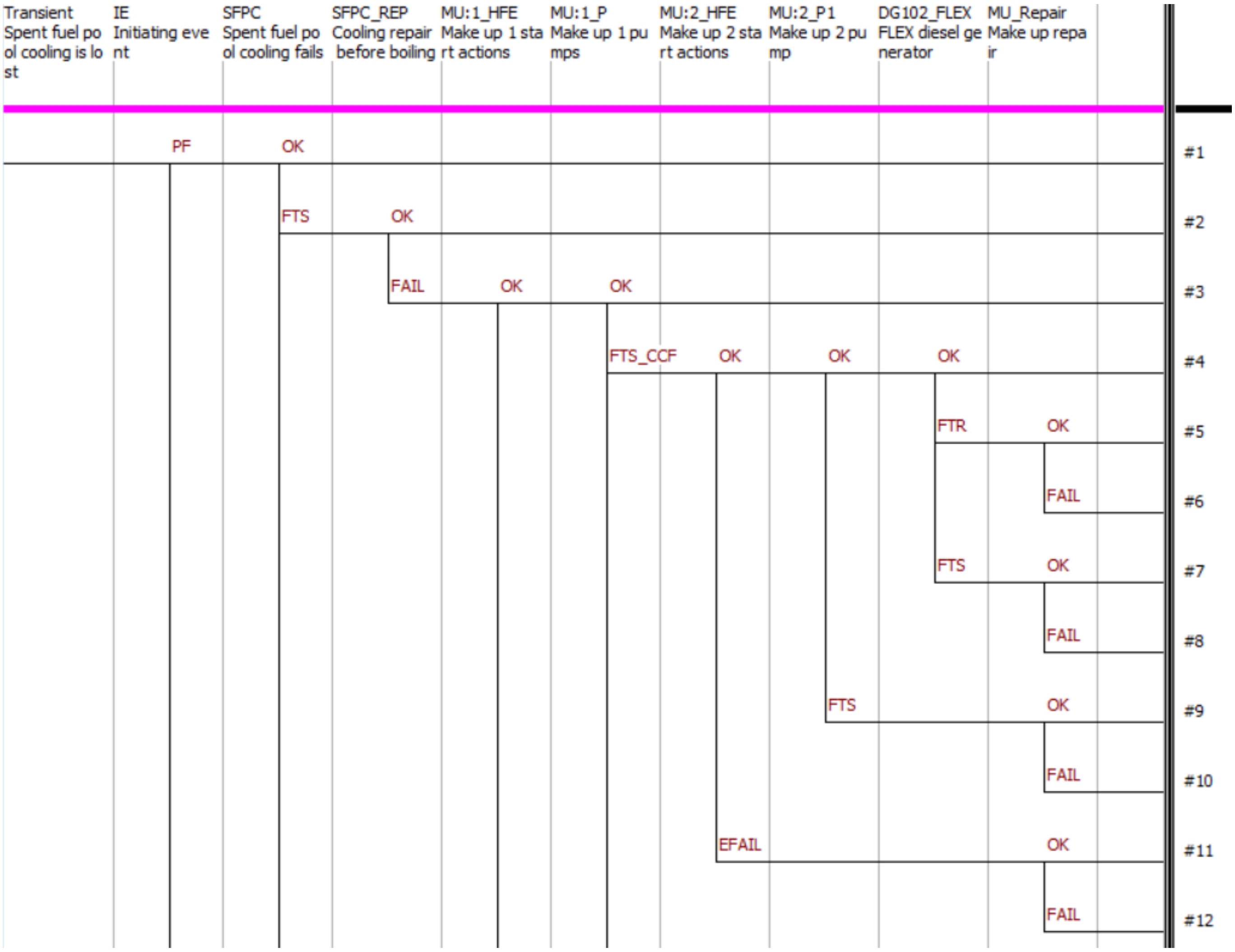


Figure 6: Upper part of the simulation-based event tree for transient.

The model has been constructed so that there is one to one correspondence between minimal cut sets and event tree sequences, i.e. each basic event and initiating event has its own branch in the tree. The simulation-based event tree calculates directly the frequencies of the minimal cut sets. The event tree has been built to cover 32 most important minimal cut sets, though it also covers many other minimal cut sets that consist of the basic events that appear in the tree. The event tree could be extended to cover more minimal cut sets, but this coverage is considered sufficient for the research purposes. The size of the tree could however become a problem at some point, because the current tree contains already 170 sequences.

Minimal cut sets that include diagnosis failures of both make-up systems are excluded from the simulation-based analysis, because it is difficult to find suitable assumptions for dynamic modelling of those. The difficult issues would be to select the start time of the second diagnosis and to model the correlation. In the static analysis (Tyrväinen et al., 2021), a simple conditional probability was applied to the second diagnosis to take into account the correlation. In this study, the minimal cut sets are included in the analysis, but they are quantified only in the static manner.

Different modelling aspects are discussed in the following sections. The modelling of manual actions is described in Section 6.1.5. The scripts of the model are presented in Appendix D.

7.1.1 Spent fuel pool cooling system

All failure probabilities related to the spent fuel pool cooling system are determined in a static manner, because the basic events are of on demand type. However, dynamicity plays a role in the modelling of repair of the system. Repair is conservatively assumed to start after complete failure of the system, i.e. after all trains have failed. The time delay related to the switching of the cooling system train is modelled in the same way as in the LOOP case, and the repair process starts after that delay. If the diagnosis of the train switching fails, the repair process is assumed to start from the beginning of boiling. The repair time depends on which components have failed. The component with the shortest MTTR is selected for repair.

A safe state is assumed if the repair is complete before boiling and the pump of the repaired train starts successfully. If the pump fails to start, its repair is conservatively modelled to start from the beginning of boiling. If a make-up system is able to bring the water level back to normal and the repair is complete, a safe state is assumed.

7.1.2 Make-up systems

Actions to start make-up system 1 are assumed to begin when the boiling starts. Make-up system 2 is assumed to be used only if make-up system 1 fails, and the actions to start it are assumed to begin from the time of make-up system 1 failure. The manual actions and failures of the make-up systems are modelled according to the principles described for make-up system 2 in Section 6.1.3.

Repair of a make-up system is modelled in the last event tree section. The repaired system is make-up system 1, unless the manual action to start it failed. The repair process is assumed to begin from the failure time of make-up system 2. The repair time distribution depends on the failed component. If the repair is not complete before the fuel uncovery, a fuel damage is assumed. In addition to the failure to repair, possible failures of the system after repair are modelled. The mission time of the system is determined based on the time to reach the normal water level and the repair time of the spent fuel pool cooling system as described in Section 6.1.3. All failure modes are modelled in the same branch to maintain one to one correspondence between sequences and minimal cut sets, because the repair is not credited in minimal cut sets.

If a make-up system fails after repair, another make-up system repair is modelled. The modelling is performed inside the same branch function. The repaired system is make-up system 2, unless the manual action to start it failed. 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.

7.2 Results

The model was simulated 50000 times both with one make-up system repair and two make-up system repairs. One set of simulations (50000 cycles) lasted 1-1.5 hours when the time step of the physical model was set to 200 seconds.

Table 7 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 B, even though only 17 minimal cut sets are covered in this table. Complete results are presented in Appendix B. The “static results” are obtained directly from the static PRA model, and the “dynamic results” have been calculated using the simulation-based event tree. Again, the static results have been refined to make the results of static and dynamic analyses more comparable. The simulation-based event tree has been analysed with one and two make-up (MU) system repairs. Results without make-up system repair have been derived from the other results (by summing the frequencies of certain sequences) and are also presented in the table.

The fuel damage frequency obtained directly from the static PRA model is of the same order of magnitude with the fuel damage frequency calculated from the simulation model without crediting make-up system repair. The main difference comes from the modelling of the spent fuel pool cooling system repair, which is not included in the static PRA model. Therefore, the static results have been refined by multiplying the minimal cut set frequencies with the repair failure probabilities. The failure probability of the spent fuel pool cooling system repair covers repair diagnosis failure, repair execution failure and failure to start of the pump after repair. The MTTR has been defined MCS specifically based on the failed component. The impacts of make-up system repairs have also been calculated in static manner. The available time for repair has been calculated in each case based on the mean values of earlier time variables in the sequence. Failures that occur after repair have also been included in the calculations.

After the refinements, the fuel damage frequency from static analysis is quite close to the result of the dynamic analysis without make-up system repair. For many minimal cut sets, the difference in frequency is very small. Differences are however still large for some minimal cut sets with failure to run event of the FLEX diesel generator, because the mission time is on average much shorter than 24 hours. The mission time depends on the repair time of the spent fuel pool cooling system, and therefore, also on how the spent fuel pool cooling system failed. For minimal cut sets with a pump failure as an initiating event combined with a manual failure to switch the spent fuel pool cooling system train, the mission time is around 18 hours, and the difference is not so large for those minimal cut sets. Some other minimal cut sets have mission times smaller than 10 hours, and therefore, larger difference in frequency.

Modelling of the repair of the spent fuel pool cooling system has different effect on different minimal cut sets due to different MTTR. The order of the minimal cut sets changes, and a minimal cut set with human failure to switch the spent fuel pool cooling system train becomes the most important minimal cut set. The effect is the same both for refined static analysis and dynamic analysis.

For minimal cut sets that do not include a failure to run event (about half of the MCSs), the impact of dynamic modelling is practically negligible when make-up system repair is not modelled, and static calculation gives a sufficient frequency estimate. Therefore, failure to run events are the only reasons to perform dynamic modelling.

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

MCS	Original static	Refined static			Dynamic		
		Without MU repair	With one MU repair	With two MU repairs	Without MU repair	With one MU repair	With two MU repairs
Total	3.95E-08	2.00E-08	1.13E-09	3.21E-10	1.62E-08	9.19E-10	2.77E-10
1	3.73E-09	1.25E-09	4.97E-11	4.41E-12	1.23E-09	4.88E-11	2.31E-12
2	3.67E-09	1.23E-09	4.89E-11	4.33E-12	4.12E-10	1.64E-11	6.97E-13
3	3.36E-09	1.91E-09	7.60E-11	6.75E-12	1.86E-09	7.44E-11	3.51E-12
4	3.31E-09	1.88E-09	7.49E-11	6.64E-12	1.32E-09	5.26E-11	2.22E-12
5	2.47E-09	8.26E-10	3.29E-11	2.92E-12	8.13E-10	3.24E-11	1.53E-12
6	2.43E-09	8.13E-10	3.24E-11	2.87E-12	2.74E-10	1.08E-11	4.62E-13
7	2.23E-09	1.27E-09	5.05E-11	4.48E-12	1.24E-09	4.93E-11	2.32E-12
8	2.19E-09	1.24E-09	4.96E-11	4.39E-12	8.79E-10	3.49E-11	1.47E-12
9	1.51E-09	1.51E-09	6.01E-11	5.34E-12	1.51E-09	6.00E-11	2.82E-12
10	1.49E-09	1.49E-09	5.93E-11	5.26E-12	1.17E-09	4.65E-11	1.96E-12
11	1.00E-09	1.00E-09	3.98E-11	3.54E-12	1.00E-09	3.98E-11	1.87E-12
12	9.85E-10	9.85E-10	3.92E-11	3.48E-12	7.77E-10	3.08E-11	1.30E-12
13	8.85E-10	2.96E-10	2.48E-11	2.14E-12	2.91E-10	1.55E-11	9.24E-13
14	8.70E-10	2.91E-10	2.44E-11	2.09E-12	9.77E-11	5.74E-12	4.51E-13
15	7.98E-10	4.53E-10	3.81E-11	3.28E-12	4.43E-10	2.85E-11	1.91E-12
16	7.84E-10	4.46E-10	3.74E-11	3.20E-12	3.15E-10	2.07E-11	1.47E-12
17	7.47E-10	2.28E-10	9.08E-12	8.06E-13	2.25E-10	8.94E-12	4.23E-13

When a repair of a make-up system is added to the analysis, the fuel damage frequency decreases 94% due to long available time for repair. For many minimal cut sets, the decrease is larger than that. The minimal cut sets with diagnosis failures for both make-up systems become the most important minimal cut sets (see the results for MCS 29 and 32 in Appendix B), because the make-up systems cannot be repaired in those scenarios. Of other minimal cut sets, the repair has the smallest effect when the manual action to start make-up system 1 fails and make-up system 2 is therefore repaired instead. The reason for this is that make-up system 2 is more unreliable after repair, because it needs the FLEX diesel generator. The failure probabilities associated with the repairs actually come mainly from component failures that occur after repair instead of failures to repair, because the available time for repairs is so long.

When a second repair of a make-up system is added to the analysis, the fuel damage frequency decreases 70% further. The decrease is however not as large as with the first repair. The reason is that the minimal cut sets with diagnosis failures for both make-up systems start to dominate the results (see the results for MCS 29 and 32 in Appendix B). They cover 88% of the fuel damage frequency. This indicates that modelling a third make-up system repair would not decrease the total result significantly, even though it could still decrease the frequencies of many other minimal cut sets.

With one repair of a make-up system, the difference between the results of dynamic and static analysis is not much larger than without a make-up system repair. In some minimal cut sets, the mission time modelling of the failure to run of the FLEX diesel generator after repair causes some difference. With two repairs of make-up systems, the FLEX diesel generator modelling causes larger differences because it applies to most minimal cut sets. However, in the total results, the difference is small, because the diagnosis failures of make-up systems dominate the results.

Table 8 presents results from sensitivity analyses. All cases have been simulated with 10000 simulation cycles. The baseline result differs from the result presented in Table 7, because much less simulation cycles have been used to reduce the computation time. The same seed number is applied 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, without make-up system repairs and with one make-up system repair. The relative result compared with the baseline is presented in parentheses.

Table 8: Sensitivity analyses for transient.

Case	**No MU repair**		**One MU repair**	
	Total	**MCS 2**	**Total**	**MCS 2**
1. Baseline	1.61E-8	4.11E-10	9.15E-10	1.63E-11
2. Make-up system 2 mission time set to 24 hours	1.94E-8 (120%)	1.16E-9 (281%)	1.04E-9 (113%)	4.59E-11 (281%)
3. Failure to run event of the FLEX diesel generator occurs at start	1.61E-8 (100%)	4.11E-10 (100%)	9.40E-10 (103%)	1.64E-11 (101%)
4. All mission times set to 24 hours	1.94E-8 (120%)	1.16E-9 (281%)	1.09E-9 (119%)	4.60E-11 (282%)
5. Mean diagnosis times are doubled	1.81E-8 (112%)	5.62E-10 (137%)	1.02E-9 (112%)	2.26E-11 (139%)
6. Mean diagnosis times are halved	1.52E-8 (94%)	3.57E-10 (87%)	8.74E-10 (96%)	1.41E-11 (87%)
7. Execution times of manual actions are doubled (except for repair)	1.60E-8 (99%)	3.68E-13 (89%)	9.08E-10 (99%)	1.46E-11 (90%)

In the transient case, there are fewer dynamic effects than in the LOOP case, and the sensitivities are also, in general, smaller. Failure of the FLEX diesel generator is the only failure to run event in the minimal cut sets, and failure to run of the repaired make-up system is also modelled when the repair is included. For the frequency of MCS 2, the mission time has a significant impact, but not very large for the total frequency. The operation time of make-up system 2 has very small significance when a make-up system repair is included in the analysis (case 3), and no significance when the repair is not included (because it literally has no impact in the model in that case). The dependency between the mission times of the FLEX diesel generator and the repaired make-up system does not seem significant based on the results. Setting the mission times to 24 hours increases the results even though the dependency is removed.

With regard to the durations of manual actions, the conclusions are similar to the LOOP case. Changes in diagnosis times have somewhat significant impact on the results, but the order of magnitude of the results is not changed. Change in execution times has smaller impact. The results actually decrease when execution times are increased, which seems counterintuitive. The reason for this is that the mission time of the FLEX diesel generator is on average shorter when it is started later, because there is shorter time until the repair of the spent fuel pool cooling. All in all, uncertainties related to durations of manual actions seem to have quite moderate impact on the results.

8. Discretization of time distributions

In dynamic event tree analyses, one popular approach to limit the computation time is to discretize the distributions of timing variables, such as operator response time (Karanki & Dang, 2016). It means that each time distribution in the model is divided into discrete intervals, e.g. based on 5th, 50th, 95th and 99.9th percentiles. From each interval, the worst value is selected for simulation to ensure that the analysis is conservative. Each combination of selected values of the timing variables is simulated, and the probabilities of the computation cases are calculated based on the intervals. In other words, while the computation cases are randomly drawn in Monte Carlo simulation, the computation cases come from the discrete time intervals in this approach. The goal of the approach is to calculate a sufficiently conservative result with a smaller number of simulations than what would be needed for Monte Carlo simulation.

The issue of simulating timings of events is common to the simulation approach presented in this report and dynamic event trees. Therefore, in this section, discretization of time distributions is tested with selected sequences from the simulation-based spent fuel pool PRA model.

8.1 Loss of offsite power

Minimal cut set 2 of the LOOP scenario is selected for this analysis, because it includes the failure to run event of the FLEX diesel generator, i.e. there is interesting dynamicity to model. Sequence 5 of the simulation-based event tree of the LOOP scenario (Figure 1) represents this minimal cut set and is thus implemented in Matlab with discretization of time distributions.

The first challenge is that the sequence involves 12 timing variables. The number of computation cases can easily become very large when time intervals of 12 distributions are combined. However, some variables are just summed in the model, and only the sum matters. This is the case for manual actions, for which the diagnosis time and execution time are summed. Therefore, some variables can be combined for discretization. When each manual action is represented only with one timing variable, seven variables remain. The distributions of timings of manual actions are determined by Monte Carlo simulation before the actual analyses.

The percentiles for the discrete intervals are selected before the analysis. However, most of the intervals are determined dynamically based on the percentiles during the simulations. For example, for the manual actions, only that part of the distribution which does not exceed the available time needs to be discretized, while the available time depends on the timings of other events in the sequence. For the failure to run events, the discretization needs to focus on the mission time, which also depends on the timings of other events in the sequence.

The last intervals of the offsite power recovery time distribution and diesel generator repair time distribution are special cases, because the worst value in the intervals is infinite. In those cases, infinite is replaced by a very high value. When the offsite power recovery time has this extreme value, the probability of failure to run CCF of emergency diesel generators is set to 1, because the mission time is infinitely long. If also diesel generator repair time has the extreme value, fuel damage is assumed after the boiling starts, because the mission time to bring the spent fuel pool to a safe state would be infinite. It needs to be ensured in the discretization that these cases have sufficiently small probability so that they do not affect the results.

The analysis is performed so that each combination of selected intervals is simulated with the timings representing the worst values of the intervals, the fuel damage frequency is calculated for each combination, the probability of each combination is calculated based on the lengths of the intervals, and finally, the weighted average of the fuel damage frequencies is calculated with the probabilities of the combinations as the weights.

The first set of discrete intervals was the following. The values are cumulative probabilities of the distributions. The number of simulations was 14400.

Offsite power recovery time: [0, 0.05], (0.05, 0.5], (0.5, 0.95], (0.95, 0.999], (0.999, 1]
Emergency diesel generator failure time: [0, 0.5), [0.5, 0.95), [0.95, 1]
Time for switching the spent fuel pool cooling system train: [0, 0.05], (0.05, 0.5], (0.5, 0.95], (0.95, 1]
Diesel generator repair time: [0, 0.05], (0.05, 0.5], (0.5, 0.95], (0.95, 0.999], (0.999, 1]
Make-up system 2 start time: [0, 0.5], (0.5, 0.95], (0.95, 0.999], (0.999, 1]
FLEX diesel generator failure time: [0, 0.5), [0.5, 0.95), [0.95, 1]
Make-up system 1 start time: [0, 0.5], (0.5, 0.95], (0.95, 0.999], (0.999, 1]

This discretization however gave an extremely conservative result, 2.8E-10/year, whereas the result from Monte Carlo simulations was 6.3E-13/year. Particularly, the tail of the offsite power recovery time was not properly discretized as the fuel damage frequency correlates significantly with the offsite power recovery time through the mission times. Significant improvements were needed for the discretization to get credible results.

Importance measure computation for the intervals was implemented in Matlab scripts to facilitate improvements to the discretization. For each interval, the contribution to the fuel damage frequency was calculated as well as the conditional fuel damage frequency given the timing related to the interval. With that information, the discretization could be improved effectively, and the result was gradually improved. However, to get even a moderately good result, the discretization needs to be very dense. The following discretization gave 2.9E-12/year with 907200 simulation cases:

Offsite power recovery time: [0, 0.3], (0.3, 0.5], (0.5, 0.6], (0.6, 0.7], (0.7, 0.8], (0.8, 0.85], (0.85, 0.9], (0.9, 0.95], (0.95, 0.97], (0.97, 0.975], (0.975, 0.98], (0.98, 0.983], (0.983, 0.985], (0.985, 0.988], (0.988, 0.99], (0.99, 0.992], (0.992, 0.993], (0.993, 0.994], (0.994, 0.995], (0.995, 0.996], (0.996, 0.997], (0.997, 0.9975], (0.9975, 0.998], (0.998, 0.9985], (0.9985, 0.999], (0.999, 0.9993], (0.9993, 0.9995], (0.9995, 0.9997], (0.9997, 0.9998], (0.9998, 0.9999], (0.9999, 0.99995], (0.99995, 0.99999], (0.99999, 0.999999], (0.999999, 0.9999999], (0.9999999, 1]
Emergency diesel generator failure time: [0, 0.2), [0.2, 0.4), [0.4, 0.5), [0.5, 0.6), [0.6, 0.8), [0.8, 1]
Time for switching the spent fuel pool cooling system train: [0, 0.25], (0.25, 0.5], (0.5, 0.95], (0.95, 1]
Diesel generator repair time: [0, 0.5], (0.5, 0.7], (0.7, 0.8], (0.8, 0.9], (0.9, 0.95], (0.95, 0.97], (0.97, 0.99], (0.99, 0.995], (0.995, 0.996], (0.996, 0.997], (0.997, 0.998], (0.998, 0.999], (0.999, 0.9999], (0.9999, 0.999999], (0.999999, 1]
Make-up system 2 start time: [0, 0.95], (0.95, 0.999], (0.999, 0.9999], (0.9999, 1]
FLEX diesel generator failure time: [0, 0.1), [0.1, 0.2), [0.2, 0.3), [0.3, 0.5), [0.5, 0.8), [0.8, 1]
Make-up system 1 start time: [0, 0.95], (0.95, 0.999], (0.999, 1]

Even with this very dense discretization, the result is very conservative. High risk contributions come from the tails of the offsite power recovery time distribution and the diesel generator repair time distribution. Clearly, the discretization approach is not useful in this case. However, some things can be learned from this analysis.

To converge, Monte Carlo simulation also requires a large number of simulation cycles so that the tails can be sufficiently covered. The 100000 simulations that were performed using FinPSA are not really enough for the fuel damage frequency to converge (while the accuracy is surely much better than with the discretization approach). Due to computation time, it may actually not be practical to aim for full convergence in this case, but a sufficient balance between the accuracy and computation time needs to be found. It would require more investigation to determine what would be a sufficient number of simulations. It could also be worthwhile to investigate other sampling techniques, such as importance sampling (Blanchet & Lam, 2012).

One limitation of the static PRA model used in this study was that the mission times were set to 24 hours without modelling the dependency to the offsite power recovery time. In principle, more accurate static modelling could be performed by discretizing the offsite power recovery time distribution and the mission time distribution. Different mission times could be modelled with separate basic events. However, the previous analysis shows that this approach would not really be practical in this case as very dense discretization would be needed to produce sufficient results.

On the other hand, a relevant question is also how realistic the distribution assumptions actually are. Do the tails of lognormal and exponential distributions really represent recovery and repair times in extreme cases, or is the problem created only because of poor distribution assumptions? It is problematic that the tails of the distributions play such an important role in the analysis, while there is no sufficient recovery or repair data to estimate the distributions that well.

8.2 Transient

Since the discretization did not work well in the LOOP case, a sequence from the transient model is analysed to obtain wider view on the use of the discretization approach. Minimal cut set 2 is selected. It corresponds to sequence 6 in the simulation-based event tree. This sequence also includes the failure to run of the FLEX diesel generator.

When one make-up system repair is included in the analysis, there are six variables to discretize after combining those variables that can be combined. However, after trying some discretization options, it was noticed that the repair time of make-up system 1 does not matter at all (because failure to start the pump after repair dominates over failure to run event). It can therefore be assumed that the repair occurs just before fuel uncovery. Only five variables remained.

The discretization was iterated with help of the importance measures mentioned in the previous section. The following discretization was found quite good:

Time for switching the spent fuel pool cooling system train: [0, 0.5], (0.5, 0.95], (0.95, 1]
Repair time of a spent fuel pool cooling system pump: [0, 0.7], (0.7, 0.8], (0.8, 0.85], (0.85, 0.9], (0.9, 0.93], (0.93, 0.95], (0.95, 0.97], (0.97, 0.99], (0.99, 0.997], (0.997, 0.999], (0.999, 0.9999], (0.9999, 0.999999999], (0.999999999, 1]
Make-up system 1 start time: [0, 0.95], (0.95, 0.999], (0.999, 1]
Make-up system 2 start time: [0, 0.95], (0.95, 0.999], (0.999, 1]
FLEX diesel generator failure time: [0, 0.3), [0.3, 1]

The result was 1.95E-11/year, while Monte Carlo gave 1.64E-11/year. This result is only slightly conservative, and it was obtained with only 702 simulations, whereas Monte Carlo was performed with 50000 simulations. Of course, it would still be possible to make denser discretization to improve the estimate if desired.

8.3 Conclusions on discretization of time distributions

Very different results were obtained in the LOOP and transient cases. The usefulness of discretization of time distributions clearly depends on the case. The transient analysis shows that it can be a useful method to reduce computation time. However, the LOOP analysis shows that it does not always work well, when the tails of some distributions dominate the results and there are many variables. It seems that it is useful to perform Monte Carlo analysis for comparison so that the conservativeness of the analysis can be evaluated.

It has to be noted that development of a good discretization may require some iterations, which means that the model has to be executed several times before near optimal solution can be found. Computation of importance measures for the discrete time intervals was found a good way to identify ways to improve the discretization. Even if the model has to be executed several times, discretization can save time in the long run as simulation models usually have to be executed many times during their development and after updates anyway.

9. Software tool development possibilities

9.1 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 always 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 have been identified:

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. Minimal cut sets are however important qualitative results from PRA, and it would be desirable to preserve those. Of the other two options, option 3 is considered better, because it would offer more flexibility for simulations, e.g. to model actions and events that can occur in different orders as identified in (Tyrväinen et al., 2021).

A separate software module that would read the minimal cut sets and relevant input data could be developed for the simulations. 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.

Such module 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.

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 in practice replace the event tree. 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. When a specific basic event attribute appeared in the minimal cut set, a specific function would be 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.

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.

9.2 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 practice, 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. However, 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. There are methods to perform approximate uncertainty analysis without heavy two-stage Monte Carlo (Hofer et al., 2002; Karanki et al., 2017), and there are more efficient sampling techniques than Monte Carlo (Rahman et al., 2018).

10. Conclusions

This report has presented an approach for simulation-based PRA of a spent fuel pool. Simulation-based event tree models have been developed to analyse loss of offsite power and transient scenarios of a fictive spent fuel pool. In the simulation-based event tree, event 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 models were 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 were compared. The dynamic analysis decreased the frequencies of many 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. Some repairs can also well be modelled in a static manner, whereas some repairs are difficult to model without a dynamic method due to complex time-dependencies.

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 actually 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 study, the results of the original static PRA model needed to be refined to reduce conservatism, particularly in the LOOP case. This was done by adding extra terms to the minimal cut sets concerning offsite power recovery, repairs and make-up system 1. This produced conservative results that may be considered quite acceptable. On the other hand, static modelling of mission times was very conservative in some cases (and also non-conservative in some cases). Also for the static analysis, the development of the dynamic model helped in understanding what kind of refinements were needed. Dynamic analysis, in general, can provide insights not obtained from static analyses.

As an alternative to Monte Carlo simulation, discretization of time distributions was investigated as a method to select the simulation cases in order to reduce the computation time. One sequence from both LOOP and transient models was analysed using the discretization. In the transient case, the approach produced only slightly conservative result with a small number of simulation cycles. On the other hand, in the LOOP case, the result was very conservative even with a very dense discretization, because the tails of the offsite power recovery time and diesel generator repair time distributions dominated the result. The usefulness of the time discretization approach seems to be case-specific. It can be useful to perform Monte Carlo simulations for comparison so that the conservativeness of the analysis can be evaluated.

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 an independent PRA model so that there would be no need for a 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

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

Blanchet, J, Lam, H. (2012) State-dependent importance sampling for rare-event simulation: An overview and recent advances, Surveys in Operations Research and Management Science, 17, 1, 38-59.

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.

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.

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

Tyrväinen, T, Immonen, E. (2022) Simulation-based probabilistic risk assessment for spent fuel pool, VTT-R-00016-22, VTT Technical Research Centre of Finland Ltd, Espoo, 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.

VTT Technical Research Centre of Finland Ltd. (2014), FinPSA - Tool for promoting safety and reliability, https://www.simulationstore.com/finpsa (link accessed 25.11.2022).

Appendix A: Detailed results for loss of offsite power

Table 9: Top minimal cut sets for loss of offsite power.

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

Meanings of basic event names:
!IE-LOOP: Loss of offsite power
ACN10GT001_________A: Failure to start gas turbine
ACN10GT001_________M: Gas turbine under maintenance
ACP-DG--------D-ALL: CCF to run between all emergency diesel generators
ACP-DG--------A-ALL: CCF to start between all emergency diesel generators
ACP-DG--------D-3XX: CCF to run between three emergency diesel generators
SFPMU:2_P1_________A: Failure to start the pump of make-up system 2
ACP__DG102_FLEX2___D: Failure to run of FLEX diesel generator
ACP__DG102_FLEX2___A : Failure to start FLEX diesel generator
SFPMU:2_MANSTART___H: Human failure to start make-up system 2
SFPC_P2____________A: Failure to start spent fuel pool cooling system pump 2
ACPX0DG001_________D: Failure to run of an emergency diesel generator

Table 10: Fuel damage frequencies (1/year) of minimal cut sets for loss of offsite power.

MCS	Seq	Static	Dynamic without MU repair	Dynamic with MU repair	Refined static
Total		1.14E-07	2.02E-11	1.92E-12	2.09E-11
1	9	2.36E-08	6.55E-12	7.21E-13	4.49E-12
2	5	2.32E-08	6.27E-13	3.91E-14	4.41E-12
3	9	1.09E-08	3.03E-12	3.33E-13	2.07E-12
4	5	1.07E-08	2.89E-13	1.80E-14	2.04E-12
5	11	2.99E-09	8.33E-13	1.45E-13	5.69E-13
6	19	2.74E-09	2.38E-13	2.84E-14	2.84E-13
7	15	2.70E-09	1.66E-14	9.69E-16	2.80E-13
8	7	2.70E-09	7.17E-13	7.91E-14	5.14E-13
9	29	2.03E-09	5.64E-13	6.22E-14	3.86E-13
10	39	2.00E-09	9.81E-13	5.52E-14	3.80E-13
11	39	2.00E-09	9.81E-13	5.52E-14	3.80E-13
12	39	2.00E-09	9.81E-13	5.52E-14	3.80E-13
13	39	2.00E-09	9.81E-13	5.52E-14	3.80E-13
14	25	2.00E-09	5.43E-14	3.37E-15	3.80E-13
15	35	1.97E-09	7.69E-14	4.19E-15	3.75E-13
16	35	1.97E-09	7.69E-14	4.19E-15	3.75E-13
17	35	1.97E-09	7.69E-14	4.19E-15	3.75E-13
18	35	1.97E-09	7.69E-14	4.19E-15	3.75E-13
19	11	1.38E-09	3.84E-13	6.68E-14	2.63E-13
20	19	1.26E-09	1.09E-13	1.30E-14	1.31E-13
21	15	1.24E-09	7.63E-15	4.45E-16	1.29E-13
22	7	1.24E-09	3.29E-13	3.63E-14	2.36E-13
23	29	9.37E-10	2.60E-13	2.87E-14	1.78E-13
24	39	9.22E-10	4.52E-13	2.54E-14	1.75E-13
25	39	9.22E-10	4.52E-13	2.54E-14	1.75E-13
26	39	9.22E-10	4.52E-13	2.54E-14	1.75E-13
27	25	9.22E-10	2.50E-14	1.56E-15	1.75E-13
28	39	9.22E-10	4.52E-13	2.54E-14	1.75E-13
29	35	9.07E-10	3.54E-14	1.93E-15	1.73E-13
30	35	9.07E-10	3.54E-14	1.93E-15	1.73E-13
31	35	9.07E-10	3.54E-14	1.93E-15	1.73E-13
32	35	9.07E-10	3.54E-14	1.93E-15	1.73E-13

Appendix B: Detailed results for transient

Table 11: Top minimal cut sets for transient.

Mc_num	Freq	Basic event names				
1	3.73E-09	SFPC_P1____I___D	CCF-SFPC-PM--A-BCD	CCF-SFPM1-PM-A-AB	SFPMU:2_P1_____A	
2	3.67E-09	SFPC_P1____I___D	ACP__DG102_FLEX2___D	CCF-SFPC-PM--A-BCD	CCF-SFPM1-PM-A-AB	
3	3.36E-09	SFPC_P1____I___D	CCF-SFPM1-PM-A-AB	SFPC_MANSTART_H	SFPMU:2_P1_____A	
4	3.31E-09	SFPC_P1____I___D	ACP__DG102_FLEX2___D	CCF-SFPM1-PM-A-AB	SFPC_MANSTART_H	
5	2.47E-09	SFPC_P1____I___D	CCF-SFPC-PM--A-BCD	SFPMU:1_P1_____A	SFPMU:1_P2_____A	SFPMU:2_P1_____A
6	2.43E-09	SFPC_P1____I___D	ACP__DG102_FLEX2___D	CCF-SFPC-PM--A-BCD	SFPMU:1_P1_____A	SFPMU:1_P2_____A
7	2.23E-09	SFPC_P1____I___D	SFPC_MANSTART_H	SFPMU:1_P1_____A	SFPMU:1_P2_____A	SFPMU:2_P1_____A
8	2.19E-09	SFPC_P1____I___D	ACP__DG102_FLEX2___D	SFPC_MANSTART_H	SFPMU:1_P1_____A	SFPMU:1_P2_____A
9	1.51E-09	SFPC_P1____I___D	CCF-SFPM1-PM-A-AB	SFPC_DIAG______H	SFPMU:2_P1_____A	
10	1.49E-09	SFPC_P1____I___D	ACP__DG102_FLEX2___D	CCF-SFPM1-PM-A-AB	SFPC_DIAG______H	
11	1.00E-09	SFPC_P1____I___D	SFPC_DIAG______H	SFPMU:1_P1_____A	SFPMU:1_P2_____A	SFPMU:2_P1_____A
12	9.85E-10	SFPC_P1____I___D	ACP__DG102_FLEX2___D	SFPC_DIAG______H	SFPMU:1_P1_____A	SFPMU:1_P2_____A
13	8.85E-10	SFPC_P1____I___D	CCF-SFPC-PM--A-BCD	SFPMU:1_MANSTART_H	SFPMU:2_P1_____A	
14	8.70E-10	SFPC_P1____I___D	ACP__DG102_FLEX2___D	CCF-SFPC-PM--A-BCD	SFPMU:1_MANSTART_H	
15	7.98E-10	SFPC_P1____I___D	SFPC_MANSTART_H	SFPMU:1_MANSTART_H	SFPMU:2_P1_____A	
16	7.84E-10	SFPC_P1____I___D	ACP__DG102_FLEX2___D	SFPC_MANSTART_H	SFPMU:1_MANSTART_H	
17	7.47E-10	SFPC_H1____I___X	CCF-SFPC-PM--A-BCD	CCF-SFPM1-PM-A-AB	SFPMU:2_P1_____A	
18	7.35E-10	SFPC_H1____I___X	ACP__DG102_FLEX2___D	CCF-SFPC-PM--A-BCD	CCF-SFPM1-PM-A-AB	
19	6.73E-10	SFPC_H1____I___X	CCF-SFPM1-PM-A-AB	SFPC_MANSTART_H	SFPMU:2_P1_____A	
20	6.62E-10	SFPC_H1____I___X	ACP__DG102_FLEX2___D	CCF-SFPM1-PM-A-AB	SFPC_MANSTART_H	
21	4.95E-10	SFPC_H1____I___X	CCF-SFPC-PM--A-BCD	SFPMU:1_P1_____A	SFPMU:1_P2_____A	SFPMU:2_P1_____A
22	4.87E-10	SFPC_H1____I___X	ACP__DG102_FLEX2___D	CCF-SFPC-PM--A-BCD	SFPMU:1_P1_____A	SFPMU:1_P2_____A
23	4.73E-10	SFPC_P1____I___D	CCF-SFPC-PM--A-BCD	CCF-SFPM1-PM-A-AB	SFPMU:2_MANSTART_H	
24	4.46E-10	SFPC_H1____I___X	SFPC_MANSTART_H	SFPMU:1_P1_____A	SFPMU:1_P2_____A	SFPMU:2_P1_____A
25	4.39E-10	SFPC_H1____I___X	ACP__DG102_FLEX2___D	SFPC_MANSTART_H	SFPMU:1_P1_____A	SFPMU:1_P2_____A
26	4.27E-10	SFPC_P1____I___D	ACP__DG102_FLEX2___A	CCF-SFPC-PM--A-BCD	CCF-SFPM1-PM-A-AB	
27	4.26E-10	SFPC_P1____I___D	CCF-SFPM1-PM-A-AB	SFPC_MANSTART_H	SFPMU:2_MANSTART_H	
28	3.85E-10	SFPC_P1____I___D	ACP__DG102_FLEX2___A	CCF-SFPM1-PM-A-AB	SFPC_MANSTART_H	
29	3.83E-10	SFPC_P1____I___D	CCF-SFPC-PM--A-BCD	SFPMU:1_DIAG___H	SFPMU:2_DIAG___HD	
30	3.59E-10	SFPC_P1____I___D	SFPC_DIAG______H	SFPMU:1_MANSTART_H	SFPMU:2_P1_____A	
31	3.53E-10	SFPC_P1____I___D	ACP__DG102_FLEX2___D	SFPC_DIAG______H	SFPMU:1_MANSTART_H	
32	3.45E-10	SFPC_P1____I___D	SFPC_MANSTART_H	SFPMU:1_DIAG___H	SFPMU:2_DIAG___HD	

Meanings of basic event names:
SFPC_P1____I___D: Failure to run of the spent fuel pool cooling system pump (initiating event)
SFPC_H1____I___X: Failure of the spent fuel pool cooling system heat exchanger (initiating event)
CCF-SFPC-PM--A-BCD: CCF to start spent fuel pool cooling system pumps
CCF-SFPM1-PM-A-AB: CCF to start make-up system 1 pumps
SFPMU:1_PX_____A: Failure to start pump X of make-up system 1
SFPMU:2_P1_____A: Failure to start the pump of make-up system 2
ACP__DG102_FLEX2___D: Failure to run of FLEX diesel generator
ACP__DG102_FLEX2___A: Failure to start FLEX diesel generator
SFPC_MANSTART_H: Human failure to execute switching of the train of the spent fuel pool cooling system
SFPC_DIAG______H: Human failure to diagnose the need to switch the train of the spent fuel pool cooling system
SFPMU:X_MANSTART_H: Human failure to execute start of make-up system
SFPMU:1_DIAG___H: Human failure to diagnose the need to start make-up system 1
SFPMU:2_DIAG___HD: Human failure to diagnose the need to start make-up system 2

Table 12: Fuel damage frequencies (1/year) of minimal cut sets for transient.

MCS	Static	Refined static without MU repair	Refined static with one MU repair	Refined static with two MU repairs	Dynamic without MU repair	Dynamic with one MU repair	Dynamic with two MU repairs
Total	3.95E-08	2.00E-08	1.13E-09	3.21E-10	1.62E-08	9.19E-10	2.77E-10
1	3.73E-09	1.25E-09	4.97E-11	4.41E-12	1.23E-09	4.88E-11	2.31E-12
2	3.67E-09	1.23E-09	4.89E-11	4.33E-12	4.12E-10	1.64E-11	6.97E-13
3	3.36E-09	1.91E-09	7.60E-11	6.75E-12	1.86E-09	7.44E-11	3.51E-12
4	3.31E-09	1.88E-09	7.49E-11	6.64E-12	1.32E-09	5.26E-11	2.22E-12
5	2.47E-09	8.26E-10	3.29E-11	2.92E-12	8.13E-10	3.24E-11	1.53E-12
6	2.43E-09	8.13E-10	3.24E-11	2.87E-12	2.74E-10	1.08E-11	4.62E-13
7	2.23E-09	1.27E-09	5.05E-11	4.48E-12	1.24E-09	4.93E-11	2.32E-12
8	2.19E-09	1.24E-09	4.96E-11	4.39E-12	8.79E-10	3.49E-11	1.47E-12
9	1.51E-09	1.51E-09	6.01E-11	5.34E-12	1.51E-09	6.00E-11	2.82E-12
10	1.49E-09	1.49E-09	5.93E-11	5.26E-12	1.17E-09	4.65E-11	1.96E-12
11	1.00E-09	1.00E-09	3.98E-11	3.54E-12	1.00E-09	3.98E-11	1.87E-12
12	9.85E-10	9.85E-10	3.92E-11	3.48E-12	7.77E-10	3.08E-11	1.30E-12
13	8.85E-10	2.96E-10	2.48E-11	2.14E-12	2.91E-10	1.55E-11	9.24E-13
14	8.70E-10	2.91E-10	2.44E-11	2.09E-12	9.77E-11	5.74E-12	4.51E-13
15	7.98E-10	4.53E-10	3.81E-11	3.28E-12	4.43E-10	2.85E-11	1.91E-12
16	7.84E-10	4.46E-10	3.74E-11	3.20E-12	3.15E-10	2.07E-11	1.47E-12
17	7.47E-10	2.28E-10	9.08E-12	8.06E-13	2.25E-10	8.94E-12	4.23E-13
18	7.35E-10	2.24E-10	8.94E-12	7.92E-13	6.99E-11	2.77E-12	1.18E-13
19	6.73E-10	2.05E-10	8.18E-12	7.26E-13	2.02E-10	8.05E-12	3.81E-13
20	6.62E-10	2.02E-10	8.05E-12	7.13E-13	6.30E-11	2.50E-12	1.07E-13
21	4.95E-10	1.51E-10	6.02E-12	5.34E-13	1.49E-10	5.93E-12	2.80E-13
22	4.87E-10	1.49E-10	5.92E-12	5.25E-13	4.63E-11	1.84E-12	7.84E-14
23	4.73E-10	1.58E-10	6.30E-12	2.83E-13	1.56E-10	6.21E-12	2.93E-13
24	4.46E-10	1.36E-10	5.42E-12	4.81E-13	1.34E-10	5.34E-12	2.52E-13
25	4.39E-10	1.34E-10	5.34E-12	4.73E-13	4.18E-11	1.66E-12	7.06E-14
26	4.27E-10	1.43E-10	5.69E-12	5.04E-13	1.34E-10	5.36E-12	2.50E-13
27	4.26E-10	2.42E-10	9.64E-12	4.33E-13	2.37E-10	9.47E-12	4.46E-13
28	3.85E-10	2.19E-10	8.71E-12	7.72E-13	2.05E-10	8.18E-12	3.81E-13
29	3.83E-10	1.28E-10	1.28E-10	1.28E-10	1.28E-10	1.28E-10	1.28E-10
30	3.59E-10	3.59E-10	3.01E-11	2.60E-12	3.58E-10	2.37E-11	1.60E-12
31	3.53E-10	3.53E-10	2.96E-11	2.54E-12	2.77E-10	1.83E-11	1.25E-12
32	3.45E-10	1.15E-10	1.15E-10	1.15E-10	1.15E-10	1.15E-10	1.15E-10

Appendix C: Scripts for loss of offsite power

The scripts of the simulation-based event tree of the LOOP scenario are presented in the following section by section.

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, p_rec

$ 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.
  p_rec = 1-cumul(OPR, boiltime)

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

  $ Recovery failure probability scaled.
  $ Only LOOP events over 4 hours are counted in the LOOP frequency.
  BINFREQ = p_rec/(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, r6, 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.
  r6 = r4*cumul(SFPCD,t_avail-t_exe-t_delay)
  t_diag = icumul(SFPCD,r6)

  $ 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.
  r6 = r4*cumul(SFPCD,t_avail-t_exe-t_delay)
  t_diag = icumul(SFPCD,r6)

  $ 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.
  r6 = r4*cumul(SFPCD,t_avail-t_exe-t_delay)
  t_diag = icumul(SFPCD,r6)

  $ 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.
  r6 = r4*cumul(SFPCD,t_avail-t_exe-t_delay)
  t_diag = icumul(SFPCD,r6)

  $ 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, r3, 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.
      r3 = 1-r*p
      t_repair = t_diag-LN(1-r3)*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_rec represents the spent fuel pool cooling recovery time.
    $ It is the minimum of the offsite power recovery time and
    $ diesel generator repair time.
    t_rec = t_repair
    if OPRecT-boiltime < t_repair then t_rec = 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, r3, 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.
    r3 = r*cumul(MU2D,t_avail-t_exe)
    t_diag = icumul(MU2D,r3)

    $ 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.
    r3 = r*cumul(MU2D,t_avail-t_exe)
    t_diag = icumul(MU2D,r3)

    $ 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

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)

  $ Does the recovery take longer than reaching the normal water level.
  if t_mission2 < t_rec-t_start2 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-t_start2,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, r4, t_earliest, t_start, t_mission, t_avail, p_fts,
t_fail, mttr,
     p_ftr, t_st, p_exe, fr, t_exe, r11, r12, fr_CCF, p_fr

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)

  $ If the power supply is not recovered before fuel damage time,
  $ fuel damage is assumed.
  if (t_avail < t_rec-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_rec-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.
        r4 = r11*(1-prob)
        t_start1 = icumul(MU1D,r4)+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
        $ ---------------------------------------

        p_exe = P_EXE1             $ Make up 1 start execution failure probability

        $ 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.

        $ If offsite power recovery time comes before theoretical fuel damage time,
        $ the system is conservatively assumed to start at the offsite power recovery time,
        $ even if diesel generator repair comes earlier.
        if OPRecT-boiltime-t_mu2 < t_avail then
        begin
          t_st = OPRecT-boiltime-t_mu2
          DG = false
        end
        else   $ Make-up system 1 is started with a diesel generator.
        begin
          t_st = t_repair-t_mu2
          DG = true
        end

        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

        $ The failure probability depends on whether the offsite power has been recovered
        $ or is the system operated with a diesel generator.
        if DG then
        begin
          p_fts = P_ALL_FTS    $ One train available

          $ 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
        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 = 2.45E-3
          mttr = MTTR_P_FTS
        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

        $ The failure probability depends on whether the offsite power has been recovered
        $ or is the system operated with a diesel generator.
        if DG then
        begin
```

```
          fr = FR_DG+FR_PUMP         $ Failure rate of a make-up system 1 train

          $ Diesel generator is assumed as the failed component with possibility to repair.
          mttr = MTTR_DG_FTR

          $ 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  $ When the offsite power is recovered, both make-up 1 trains are available.
        begin
          fr_CCF = 1.37E-7     $ Failure rate for CCF of make-up 1 pumps
          fr = FR_PUMP         $ Failure rate for one pump
          mttr = MTTR_P_FTR

          $ Failure to run CCF probability
          p_fr = 1-exp(-fr_CCF*t_mission)

          $ Probability that one pump fails to start and one pump fails to run
          p_fr = p_fr + 2*0.0374*(1-exp(-fr*t_mission))

          $ Probability that make-up system 1 fails to run and its repair fails.
          prob = prob+(1-prob)*p_fr*RepairFailLOOP(Temperature, WLevel, mttr, t_st+t_fail)
        end

        t_mission1 = t_mission  $ Collect to results
      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
```

Appendix D: Scripts for transient

The scripts of the simulation-based event tree of the transient scenario are presented in the following section by section.

Initial section

```
$ Global variables are defined in the common section.

$ Variable values that are collected to results
Collect t_mission2, t_mu1, t_mu2, t_start1, t_start2, t_fail2, t_repair, t_startC

$ Routine init is executed first
routine init
  FD = false

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

  $ Time to uncovery from boiling (just for information)
  $ uncoverytime = t_uncover(WLevel)
  $ The total frequency of the initiating events
  BINFREQ = 0.0485

  $ Initialization
  mttr1 = 0
  mttr2 = 0
  rr2 = 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
```

IE (Initiating event)

```
real p

routine init

return


$ Pump failure
function nil PF
  $ Mean time to repair the spent fuel pool cooling system.
  sfpcmrt = MTTR_P_FTR
return nil


$ Heat exchanger failure
function real HF
  $ Mean time to repair the spent fuel pool cooling system.
  sfpcmrt = MTTR_HEX_FTR

  $ The heat exchanger's share of the total initiating event frequency
  p = 1/6
return p
```

SFPC (Spent fuel pool cooling fails)

```
real r2, r3, r4, r5, t_avail, t_diag,
     t_exe, t_delay, t_shift, p_fts, p_e, p_d

routine init
  r2 = random()
  r3 = random()
  r4 = random()

  t_shift = 8

  p_fts = 0.00111
  p_e = 0.001
  p_d = 0.00045
return
```

```
function nil OK

return nil


$ Three standby pumps fail to start (CCF)
function real FTS
  $ 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.
  r5 = r4*cumul(SFPCD,t_avail-t_exe-t_delay)
  t_diag = icumul(SFPCD,r5)

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

  $ Mean time to repair the spent fuel pool cooling system.
  if sfpcmrt > MTTR_P_FTS then sfpcmrt = MTTR_P_FTS
return p_fts


$ Human failure to execute the switching of the spent fuel pool cooling system train
function real EFAIL
  $ 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.
  r5 = r4*cumul(SFPCD,t_avail-t_exe-t_delay)
  t_diag = icumul(SFPCD,r5)

  $ The time delay related to the train switching actions
  t_startC = t_delay+t_diag+t_exe
return p_e


$ Human failure to diagnose the need to switch the spent fuel pool cooling system train
function real DFAIL
  $ This leads to boiling conditions.
  t_startC = boiltime
return p_d
```

SFPC_REP (Cooling repair before boiling)

```
real p, p2, r, r2, r3, r4, r5, t_diag

routine init
  r = random()
  r2 = random()
  r3 = random()
  r4 = random()
return
```

```
function nil OK

return nil


$ Repair of the spent fuel pool cooling system before boiling fails.
function real FAIL
  $ Diagnosis time for the repair.
  t_diag = icumul(SFPCRD,r)

  $ Probability that the repair is not performed before boiling is determined.
  if t_diag < boiltime-t_startC then
  begin
    p = EXP(-(boiltime-t_startC-t_diag)/sfpcmrt)
  end
  else
  begin
    p = 1
  end

  $ Repair time of the spent fuel pool cooling system from exponential distribution.
  $ Time point 0 is when the boiling starts.
  r5 = 1-r2*p
  t_repair = t_diag-LN(1-r5)*sfpcmrt-(boiltime-t_startC)

  $ A special case is the case where the pump fails to start after repair.
  if ((p < 1) and (r3 < P_PUMP_FTS/(P_PUMP_FTS+p))) then
  begin
    sfpcmrt = MTTR_P_FTS

    $ New repair is assumed to start when the boiling starts.
    $ New repair time is drawn.
    t_repair = -LN(1-r4)*sfpcmrt
  end

  t_rec = t_repair

  $ Probability that the pump fails to start after repair is added.
  p = p + (1-p)*P_PUMP_FTS

  $ Conditions at the beginning of boiling
  Temperature = BoilingTemp
  WLevel = InitWLevel
return p
```

MU:1_HFE (Make up 1 start actions)

```
$ Local variables
real prob, r2, r, r3, t_avail, t_diag, t_exe, ti

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


$ Make up 1 start is performed successfully
function nil OK
  $ Time available to start make up system 1.
  t_avail = t_uncover(WLevel)

  $ The execution time of make up system 1 start is drawn from uniform distribution.
  t_exe = r2+0.5

  $ The diagnosis time of make up system 1 start is drawn from lognormal distribution.
  r3 = r*cumul(MU1D,t_avail-t_exe)
  t_diag = icumul(MU1D,r3)
  $ The start time of make up system 1.
  t_start1 = t_diag+t_exe

  $ The water level is updated.
  WLevel = newWLevel(WLevel, 0, t_start1)
return nil


$ Execution fails
function real EFAIL
  $ Time available to start make up system 1.
  t_avail = t_uncover(WLevel)

  $ The execution time of make up system 1 start is drawn from uniform distribution.
  t_exe = r2+0.5

  $ The diagnosis time of make up system 1 start is drawn from lognormal distribution.
  r3 = r*cumul(MU1D,t_avail-t_exe)
  t_diag = icumul(MU1D,r3)

  $ Time when execution attempt is finished.
  t_start1 = t_diag+t_exe

  $ The water level is updated.
  WLevel = newWLevel(WLevel, 0, t_start1)

  $ Execution failure probability
  prob = P_EXE1

  t_mu1 = t_start1

  MU1EFAIL = true
return prob
```

MU:1_P (Make up 1 pumps)

```
real prob

routine init

return


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

$ Common cause failure of make up 1 pumps
function real FTS_CCF
  prob = 2.11E-3

  t_mu1 = t_start1

  mttr1 = MTTR_P_FTS
return prob

$ Make up 1 pumps fail independently
function real FTS_2
  prob = 3.74E-2*3.74E-2

  t_mu1 = t_start1

  mttr1 = MTTR_P_FTS
return prob
```

MU:2_HFE (Make up 2 start actions)

```
$ Local variables
real prob, r2, r, r3, t_avail, t_diag, t_exe, t_boiling

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_boiling = t_boil(Temperature, WLevel)
  t_avail = t_boiling + t_uncover(WLevel)

  $ 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.
    r3 = r*cumul(MU2D,t_avail-t_exe)
    t_diag = icumul(MU2D,r3)

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

    $ The spent fuel pool conditions are updated depending on
    $ if the system is started before or after boiling.
    if t_start2 < t_boiling then
    begin
      Temperature = newTemp(Temperature, WLevel, 0, t_start2)
    end
    else
    begin
      Temperature = BoilingTemp
      WLevel = newWLevel(WLevel, 0, t_start2-t_boiling)
    end
  end
return nil


$ Execution fails
function real EFAIL
  $ Time available to start make up system 2.
  t_boiling = t_boil(Temperature, WLevel)
  t_avail = t_boiling + t_uncover(WLevel)

  $ 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.
    r3 = r*cumul(MU2D,t_avail-t_exe)
    t_diag = icumul(MU2D,r3)

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

    $ The spent fuel pool conditions are updated depending on
    $ if the system start attempt is finished before or after boiling.
    if t_start2 < t_boiling then
    begin
```

```
      Temperature = newTemp(Temperature, WLevel, 0, t_start2)
    end
    else
    begin
      Temperature = BoilingTemp
      WLevel = newWLevel(WLevel, 0, t_start2-t_boiling)
    end

    t_mu2 = t_start2

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

    Temperature = BoilingTemp
    WLevel = FuelLevel
  end

  MU2EFAIL = true

  $ If start executions fail for both make up systems, fuel damage is assumed.
  if MU1EFAIL then FD = 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


function real FTS
  prob = P_PUMP_FTS

  t_mu2 = t_start2

  $ This pump is either the first or second make up component to be repaired,
  $ depending on if make up 1 start execution failed.
  if MU1EFAIL then mttr1 = MTTR_P_FTS else mttr2 = MTTR_P_FTS
return prob
```

DG102_FLEX (FLEX diesel generator)

```
real prob, r, t_earliest, fr

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 normal water level.
  t_mission2 = t_restore(WLevel)

  $ Does the recovery take longer than reaching the normal water level.
  if t_mission2 < t_rec-t_start2-t_mu1 then
  begin
    $ Given the repair time of the spent fuel pool cooling system,
    $ the earliest allowed failure time is calculated.
    $ The EarliestTime function is defined in the common section.
    t_earliest = EarliestTime(Temperature,t_rec-t_start2-t_mu1,WLevel)

    $ If the earliest allowed failure time based on the repair of the spent fuel pool cooling
    $ system 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

  $ This diesel generator is either the first or second make up component to be repaired,
  $ depending on if make up 1 start execution failed.
  if MU1EFAIL then mttr1 = MTTR_DG_FTR else mttr2 = MTTR_DG_FTR
return prob

$ Failure to start
function real FTS
  prob = P_DG_FTS

  t_mu2 = t_start2

  $ This diesel generator is either the first or second make up component to be repaired,
  $ depending on if make up 1 start execution failed.
  if MU1EFAIL then mttr1 = MTTR_DG_FTS else mttr2 = MTTR_DG_FTS
return prob
```

MU_Repair (Make up repair)

```
$ Local variables
real prob, t_boiling, r, r1, r2, r3, r4, r5, t_earliest, t_start, t_mission, t_avail, p_fts,
     t_diag, t_fail, mttr, fr, p

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


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

```
$ Failure to repair make up systems. Failures after repairs are also included.
$ Two make up system repairs are modelled. The second repair is modelled by calling
$ RepairFail function, which is defined in the common section.
$ The failure probability that this function calculates covers both repairs.
function real FAIL
  $ Three failure modes of make up system 1 (or 2) are evaluated in the following:
  $ failure to repair, failure to start and failure to run.
  $ For the latter two, another repair is considered.
  $ The probabilities of the failure modes are summed.

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

  $ Time available for repair.
  t_boiling = t_boil(Temperature, WLevel)
  t_avail = t_boiling + t_uncover(WLevel)

  $ Diagnosis failure probability is determined and a diagnosis time is drawn.
  p = 1-cumul(MURD,t_avail)
  r4 = r3*p
  t_diag = icumul(MURD,r4)

  $ The repair failure probability is calculated assuming exponential distribution
  $ for the repair time.
  prob = EXP(-(t_avail-t_diag)/mttr1)

  $ The repair time is drawn from exponential distribution.
  r5 = r1*(1-prob)
  t_start = t_diag-LN(1-r5)*mttr1

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

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

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

  $ Failure to start probability of make up 1,
  $ or make up 2 if make up 1 start execution failed.
  p_fts = P_PUMP_FTS
  if MU1EFAIL then p_fts = P_ALL_FTS

  $ MTTR for another repair after the failure to start is determined.
  $ It depends on if make up system start execution failure has occurred.
  mttr = mttr2
  if MU1EFAIL or MU2EFAIL then
  begin
    if MU1EFAIL then
    begin
      $ Make-up system 2 is repaired.
      $ 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
    begin
```

```
      $ Make-up system 1 is repaired.
      mttr = MTTR_P_FTS
      DG = false
    end
  end

  $ Probability of failure to start and failure of consecutive repair is added.
  prob = prob+(1-prob)*p_fts*RepairFail(Temperature, WLevel, mttr, t_start)

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

  $ If MU1 was repaired only pump failure is considered.
  $ If MU2 was repaired both DG and pump failures are considered.
  if MU1EFAIL then fr = FR_DG+FR_PUMP else fr = FR_PUMP

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

  if t_mission < t_rec-t_start-t_mu1-t_mu2 then
  begin
    $ Given the repair time of the spent fuel pool cooling system,
    $ the earliest allowed failure time is calculated.
    $ The EarliestTime function is defined in the common section.
    t_earliest = EarliestTime(Temperature,t_rec-t_start-t_mu1-t_mu2, WLevel)

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

  $ The failure time of the diesel generator 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

  $ MTTR for another repair after the failure to start is determined.
  $ It depends on if make up system start execution has occurred.
  mttr = mttr2
  if MU1EFAIL or MU2EFAIL then
  begin
    if MU1EFAIL then
    begin
      $ Make-up system 2 is repaired.
      $ DG failure is assumed, because it dominates the total failure rate.
      mttr = MTTR_DG_FTR
      DG = true
    end
    else
    begin
      $ Make-up system 1 is repaired.
      mttr = MTTR_P_FTR
      DG = false
    end
  end

  $ Probability of failure to run and failure of consecutive repair is added.
  prob = prob+(1-prob)*(1-exp(-fr*t_mission))*RepairFail(Temperature, WLevel, mttr,
t_start+t_fail)

  FD = true
return prob
```

Appendix E: Common scripts

The scripts of the common section of the simulation-based event trees are presented in the following.

```
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_startC,     $ Start time of the spent fuel pool cooling system
     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
     t_rec,        $ Spent fuel pool cooling recovery time (if normal water level)

     $ 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
     MTTR_HEX_FTR = 11,    $ Heat exchanger 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 power supply for repaired make-up system 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 = recovery time of the spent fuel pool cooling
$ 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, rrr, rrr2

  $ 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
  if t_ava > 0 then p = 1-cumul(MURD,t_ava) else p = 1

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

  $ 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.
  rrr2 = rr2*(1-p2)
  t_st = t_st-LN(1-rrr2)*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_rec-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, rrr, rrr2

  $ 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
  if t_ava > 0 then p = 1-cumul(MURD,t_ava) else p = 1

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

  $ 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.
  rrr2 = rr2*(1-p2)
  t_st = t_st-LN(1-rrr2)*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 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: CF669882FA324931A797B6ED4D190DAF		Status: Completed
Subject: DocuSign: VTT-R-00990-22		
Source Envelope:		
Document Pages: 62	Signatures: 1	Envelope Originator:
Certificate Pages: 1	Initials: 0	Anne Räsänen
AutoNav: Enabled		Vuorimiehentie 3, Espoo,
EnvelopeId Stamping: Enabled		., . P.O Box1000,FI-02044
Time Zone: (UTC+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius		Anne.Rasanen@vtt.fi
		IP Address: 130.188.17.16

Record Tracking

Status: Original 14 December 2022 | 15:06	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: 14 December 2022 | 15:07 Viewed: 14 December 2022 | 15:17 Signed: 14 December 2022 | 15:17

Authentication Details
SMS Auth:
Transaction: 6614431EBA440D04919289F5AA6A40BE
Result: passed
Vendor ID: TeleSign
Type: SMSAuth
Performed: 14 December 2022 | 15:16
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	14 December 2022 | 15:07
Certified Delivered	Security Checked	14 December 2022 | 15:17
Signing Complete	Security Checked	14 December 2022 | 15:17
Completed	Security Checked	14 December 2022 | 15:17

Payment Events	Status	Timestamps