

Title	Computation of consequences of piping component failures in PRA software
Author(s)	Tyrväinen, Tero
Citation	Research Report : VTT-R-03666-16 VTT, 2016
Date	2016
Rights	This report may be downloaded for personal use only.

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

By using VTT Digital Open Access Repository you are bound by the following Terms & Conditions.

I have read and I understand the following statement:

This document is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of this document is not permitted, except duplication for research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered for sale.

Report's title	
Computation of consequences of piping component failures in PRA software	
Customer, contact person, address	Order reference
State Nuclear Waste Management Fund (VYR), VTT	
Project name	Project number/Short name
Analysis of Fatigue and Other cUmulative ageing to exteND lifetime	108787/FOUND
Author(s)	Pages
Tero Tyrväinen	17/-
Keywords	Report identification code
Risk-informed in-service-inspections, probabilistic risk assessment, nuclear safety, pipe	VTT-R-03666-16
Summary	
<p>In-service inspections are applied to safety important pipe components in nuclear power plants (NPPs) to ensure their reliability. In risk-informed in-service inspections (RI-ISI), these inspections are planned according to risk importance of piping components. The quantitatively computed risk of a piping component combines the probability of the failure and the consequences of the failure.</p> <p>This report focuses on estimating the consequence of piping component failures. A typical measure for the consequences is conditional core damage probability (CCDP), and it can be calculated from a probabilistic risk assessment (PRA) model of a NPP. However, computation of CCDP values of piping component failures has not been automated in PRA software, such as FinPSA, and requires typically additional processing of the model because all piping component failures are not included in the PRA model explicitly.</p> <p>This report introduces a prototype of a new RI-ISI feature which calculates CCDP values of piping component failures in PRA software FinPSA. The RI-ISI feature enables detailed modelling of the consequences of piping component failures without complicating the PRA model itself. The CCDP values of all piping component failures can be calculated automatically at once in the same RI-ISI table based on the results of the PRA model.</p>	
Confidentiality	Public
Espoo 25.10.2016	
Written by	Reviewed by
	
Tero Tyrväinen Research Scientist	Otso Cronvall Senior Scientist
	Accepted by
	
	Tarja Laitinen Head of Research Area
VTT's contact address	
P.O. Box 1000, 02044 VTT	
Distribution (customer and VTT)	
SAFIR2018 Reference Group 5 (Structural Integrity); VTT Archive (2)	
<p><i>The use of the name of VTT Technical Research Centre of Finland Ltd in advertising or publishing of a part of this report is only permissible with written authorisation from VTT Technical Research Centre of Finland Ltd.</i></p>	

Contents

1. Introduction.....	3
2. FinPSA	3
3. Computation of consequences of piping component failures.....	3
4. Conditional core damage probability computation feature	5
4.1 RI-ISI table.....	5
4.2 Event combinations table.....	7
5. Simple example case.....	8
6. Software design for the RI-ISI feature	10
6.1 Unit RIISI	11
6.2 Unit RIISWindow.....	13
6.3 Unit RIISIFailure	13
6.4 Unit RIISIDetail	14
6.5 Unit Project.....	14
7. Conditional large early release probability.....	14
8. Conclusions	16
References.....	16

1. Introduction

In-service-inspections are applied to safety important pipe components in nuclear power plants (NPPs) to ensure their reliability. In Finland, in-service-inspections are planned carefully so that the risk of nuclear accident, employees' exposure to radiation and the cost of inspections are in balance and within acceptable limits. This approach is called risk-informed in-service-inspection (RI-ISI) [1].

Probabilistic risk assessment (PRA) is used to calculate the quantitative risk of nuclear accident and to analyse the importance of different systems and components [2]. PRA's main purpose is to support risk-informed decision making. PRA also supports RI-ISI analyses by quantifying the consequences of pipe failures.

The previous research project report [3] studied the connection between PRA and RI-ISI analyses. In the report, it was identified that it would be beneficial to develop automatic piping failure consequence calculator in a PRA software. Hence, this report introduces a prototype of a new RI-ISI feature which calculates the conditional core damage probabilities of piping component failures in PRA software FinPSA [4]. FinPSA software is briefly presented in Section 2. Section 3 discusses how the consequences of piping component failures can be calculated and points out the limitations of the current approaches. Section 4 presents the new RI-ISI feature, and Section 5 demonstrates the new feature with a simplified example model. In Section 6, the software design of the new RI-ISI feature is presented. Section 7 discusses how conditional large early release probabilities can be calculated, and Section 8 concludes the study.

2. FinPSA

FinPSA is a software tool for full-scope PRA [4]. FinPSA supports PRA levels 1 and 2. Level 1 PRA concerns accident sequences leading to core damage and calculation of the core damage frequency. Level 2 PRA concerns the progression of severe accidents after core damage and calculation of frequencies and amounts of releases.

In level 1, FinPSA uses event trees and fault trees [2, 5]. An event tree represents how an accident can evolve from an initiating event via failures of safety systems to a consequence, e.g. core damage. A fault tree represents which events can cause the analysed system to fail. Fault trees are linked to branching points in event trees. From fault trees, minimal cut sets are solved. Minimal cut sets are minimal combinations of events that can cause the top event, e.g. core damage. Probabilistic assessment is performed based on minimal cut sets and reliability data of components.

Level 2 part of FinPSA is based on dynamic containment event trees (CET) and CETL programming language [4]. The CETL language is used to define functions to calculate conditional probabilities of event tree branches, timings of the accident progression and amounts of releases. The CET models are solved by Monte Carlo simulations.

Verification and validation (V&V) procedures have been established for FinPSA [6, 7]. For each new version, a set of V&V runs/tests is performed. This set contains the most important validation runs (e.g. generation of minimal cut sets for a set of models) and tests for new and modified properties.

3. Computation of consequences of piping component failures

For RI-ISI analyses, the failure probabilities of piping components and the consequences of piping failures need to be estimated [1]. Typical consequence measures are the conditional

core damage probability/frequency (CCDP/CCDF) and the conditional large early release probability/frequency (CLERP/CLERF). CCDP/CCDF is usually more convenient from the computational point of view and it can always be calculated from the PRA model of the NPP.

A failure of a piping component can, for example, cause disturbance in the plant's usage or failure of a system. All piping components are not separated in PRA. Typically, one initiating event or basic event is used to represent the failures of piping components and other events with similar consequences. Failures of safety system pipes not causing initiating events do usually not appear in PRA at all [8]. Their consequences are often analysed using other "surrogate" basic events, e.g. pump and valve failures, which are included in the PRA model. The consequences of initiating events and basic events in PRA may represent the consequences of piping failures only roughly, and therefore, more accurate consequence analyses may be needed in RI-ISI analyses.

The integration of PRA and RI-ISI analyses would improve if the PRA model contained the pipe failures that are included in RI-ISI. However, a straightforward extension could complicate the PRA model and calculations too much. PRA models include initiating events, such as loss of coolant accident (LOCA), which can be caused by several different pipe failures. Only a single frequency (possibly with uncertainty distribution) is needed for an initiating event in PRA analyses. PRA calculations would not benefit from dividing initiating events to smaller parts. In addition, pipe failures that can cause safety systems to fail are often excluded from PRA because their contribution is very small.

In typical PRA software, CCDPs/CCDFs have to be calculated separately for each component by setting the failure frequency/probability to 1 or derived from other risk importance measures [9]. The first step taken in this work was the implementation of new CCDP/CCDF importance measure in PRA software FinPSA [4]. CCDP/CCDF importance measure works in the same way as any other risk importance measure in FinPSA: CCDP/CCDF is calculated for each basic event and initiating event in the model, and they are listed from the most important to the least important. The CCDP of an initiating event is calculated as the total frequency of the minimal cut sets including the initiating event divided by the initiating event frequency. The CCDF of a basic event is calculated following the same principle.

Limitations with the CCDP/CCDF importance measure are that there is not always a perfect correspondence between piping component failures and initiating/basic events of the PRA model, and that the CCDP of an initiating event and the CCDF of a basic event are not comparable. In addition, the CCDP/CCDF list that FinPSA provides is not the list of CCDPs/CCDFs of piping component failures. In the next section, a new FinPSA feature is introduced to overcome these limitations.

To compare CCDPs to CCDFs, one approach is to transform the CCDFs into CCDPs in the following way [10]:

$$CCDP = 1 - e^{-CCDF \times t}, \quad (1)$$

where t is the time the piping failure is in effect.

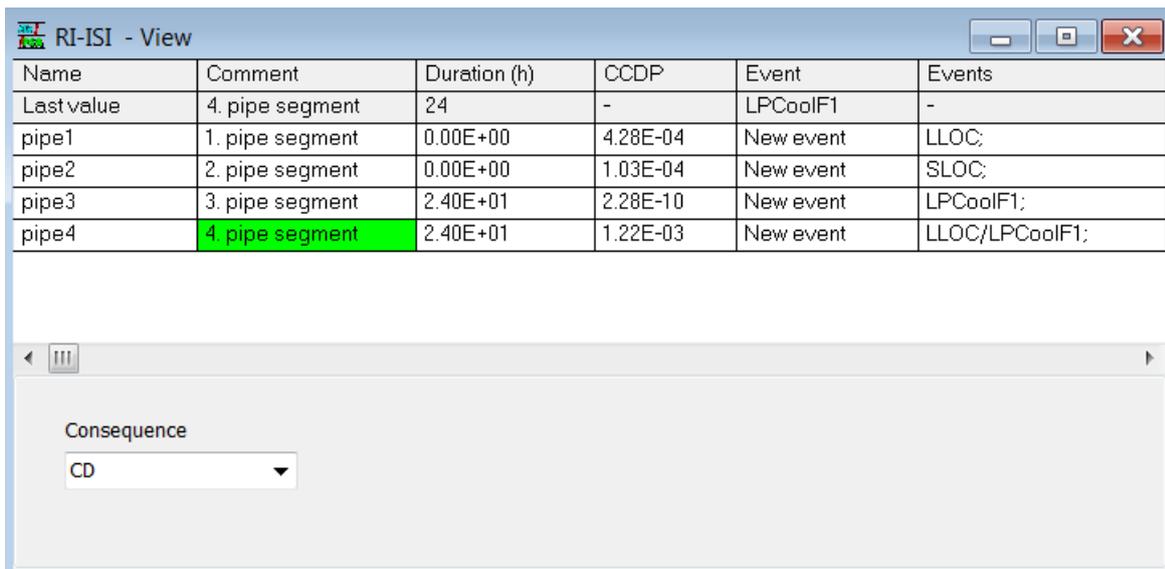
The computation of the conditional large early release probability/frequency (CLERP/CLERF) is in most cases less straightforward than the computation of CCDP/CCDF. The computation of CLERP/CLERF depends on how the level 2 PRA is implemented and integrated to level 1 PRA. The methods used in the level 2 PRA vary a lot, and the PRA levels 1 and 2 are not always integrated.

4. Conditional core damage probability computation feature

This section presents a prototype of a new RI-ISI feature in FinPSA software [4]. It is a table that specifies piping components and their consequences, and calculates CCDPs based on PRA results. The table is a part of FinPSA's database. It is opened from FinPSA menu by choosing Database > RI-ISI. The feature contains two parts: the main part that can be used alone and an additional part that can be used for more detailed modelling of piping failure consequences.

4.1 RI-ISI table

The main part of the RI-ISI feature is a table that contains a row for each piping component in the model. A RI-ISI table with four piping components is presented in Figure 1. The name column contains the name of the piping component, which is defined by user. The comment column can contain any further information of the piping component. Duration refers to the time the piping failure is in effect and is needed only when the piping failure causes a basic event (failure of a safety system component) in the PRA model. The CCDP column contains the calculated CCDPs. The event column is used to add an initiating event or a basic event that is caused by the piping component failure. The events column contains the initiating events and basic events that the piping component failure causes.



Name	Comment	Duration (h)	CCDP	Event	Events
Last value	4. pipe segment	24	-	LPCoolF1	-
pipe1	1. pipe segment	0.00E+00	4.28E-04	New event	LLOC;
pipe2	2. pipe segment	0.00E+00	1.03E-04	New event	SLOC;
pipe3	3. pipe segment	2.40E+01	2.28E-10	New event	LPCoolF1;
pipe4	4. pipe segment	2.40E+01	1.22E-03	New event	LLOC/LPCoolF1;

Consequence
CD

Figure 1. RI-ISI table.

The data records of FinPSA contain all the basic events and initiating events that are used in the PRA model. In this approach, piping components are separate from those events and do not appear in the PRA model (e.g. in fault trees). In the RI-ISI table, it is specified which initiating events and basic events each piping component failure causes. One piping component failure can cause multiple events that are modelled in PRA. An initiating event or a basic event is assigned to the piping component by writing its name in the event column. Then, it is added to the events column, which contains the list of all events that are caused by the piping component failure. For example, in Figure 1, pipe4 failure causes initiating event LLOC and basic event LPCoolF1.

Below the table, there is an area where a consequence is selected. CCDP computation has to be performed based on the PRA results of the consequence that represents core damage, but PRA model can contain several consequences for which results are calculated (e.g. different core damage types and economic consequences). Therefore, the user has to select

which consequence represents the core damage in the model. In principle, the computation can also be performed with regard to other PRA consequences, but that is not the idea here.

The CCDP is calculated based on the minimal cut sets of core damage, and therefore, the PRA results must exist before CCDP computations. FinPSA contains the following algorithms for total probability/frequency computation [11]:

- S1-sum (simple sum)
- MCA (computation with cross-products of minimal cut sets)
- MCU (minimal cut set upper bound, minimal cut sets assumed independent)
 - Allow IE (initiating events treated as mutually exclusive)
 - Require IE (checks that each minimal cut set includes one initiating event, initiating events treated as mutually exclusive)
 - Ignore IE (initiating events are not separated from basic events)

If S1-sum is chosen, the CCDP is calculated by the following steps.

1. Set CCDP and CCDF variables to 0.
2. Get next minimal cut set.
3. Divide the frequency of the minimal cut set by probability/frequency of each event that belongs both to the minimal cut set and the events specified for the corresponding piping component.
4. If the minimal cut set contains an initiating event specified for the corresponding piping component, add the conditional probability of the minimal cut set calculated in step 3 to CCDP variable and go to step 6.
5. If the minimal cut set contains at least one basic event specified for the corresponding piping component, add the conditional frequency of the minimal cut set calculated in step 3 to CCDF variable.
6. Check if there is a next minimal cut set. If there is, go to step 2.
7. $CCDP = CCDP + (1 - e^{-CCDF \times t})$, where t is the duration.

Other algorithms work differently. Let IE denote the frequency of the initiating event in the event combination specified for the corresponding piping component. Let BE represent the probabilities of all basic events in the event combination. The total frequency/probability is calculated using three different settings and the selected computation algorithm:

- $Q(IE=0, BE=0)$
- $Q(IE=0, BE=1)$
- $Q(IE=1, BE=1)$,

where Q is the total frequency or probability. Total frequency computation algorithms can be studied from ref. [11]. $Q(IE=0, BE=0)$ is the total frequency of those minimal cut sets that do not contain any of the events specified for the corresponding piping component. Next, the following formulas are applied:

$$CCDF = Q(IE = 0, BE = 1) - Q(IE = 0, BE = 0), \quad (2)$$

$$CCDP = Q(IE = 1, BE = 1) - Q(IE = 0, BE = 1). \quad (3)$$

Finally, $CCDP = CCDF + (1 - e^{-CCDF \times t})$, where t is the duration.

The CCDP values are updated automatically when changes are made to the table.

4.2 Event combinations table

When a row is double-clicked in the RI-ISI table and the table is in view mode, another window is opened (respectively, the table is modified in edit mode). This window contains detailed analysis of the chosen piping component. An example of the window is presented in Figure 2. The upper part of the window contains data from the RI-ISI table. The lower part contains a table that specifies different failure scenarios that can occur due to the piping component failure. In the table, it is possible to specify conditional probabilities for different piping failure consequences, e.g. a small LOCA occurs with a probability of 0.5 and a large LOCA occurs with a probability of 0.3. This table will be called “event combinations table” from this point forward.

RI-ISI pipe5 - Edit							
RI-ISI Component: pipe5		Duration: 0.00E+00 (h)		CCDP: 4.24E-04			
5. pipe segment		Consequence: CD					
Index	Comment	Probability	Duration (h)	1. event	2. event	3. event	4. event
-	Large LOCA with LP cooling SS	0.2	24	LLOC	LPCoolF1	-	-
1	Small LOCA	5.00E-01	0.00E+00	SLOC			
2	Large LOCA	3.00E-01	0.00E+00	LLOC			
3	Large LOCA with LP cooling SS	2.00E-01	2.40E+01	LLOC	LPCoolF1		

Figure 2. Event combinations table.

While in the table presented in Figure 1 it was possible to define only one combination of events that occur due to the piping failure, in the table presented in Figure 2 it is possible to define multiple combinations and assign conditional probabilities to them. The table of Figure 2 contains nine event columns where the names of the initiating events and basic events are placed (only four are seen in the figure). There is also a column for other events if the number of events is larger than nine (Figure 3). In this column, the names of other events are listed and separated by ‘/’. A new event can be added (in addition to using the nine numbered event column) using the event column, which is the last column in the table (by default) and works similarly as the event column of the RI-ISI table. It is also possible to define separate duration for each event combination. Duration longer than 0 in the table overrides the duration that appears in the upper part of the window. If the duration in the table is 0, the duration that appears in the upper part of the window is used in the computation.

5. event	6. event	7. event	8. event	9. event	Other events	Event
-	-	-	-	-	-	-
adsg	dsdg	adfse	qa	fff	g/jty/goo	New event
						New event

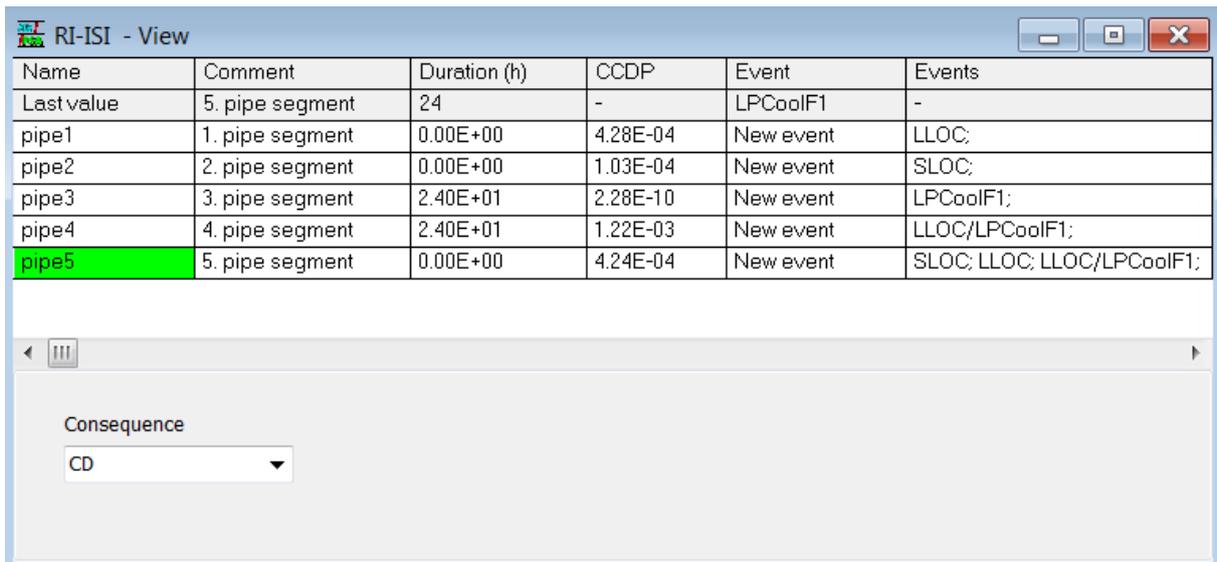
Figure 3. The last columns in the event combinations table.

The CCDP is calculated as a weighted sum of the CCDPs of the failure combinations (rows of the table):

$$CCDP = \sum_{i=1}^n P_i \times CCDP_i, \quad (4)$$

where P_i is the conditional probability of the failure combination and n is the number of the failure combinations. CCDPs of the failure combinations are calculated as described in Section 4.1.

The modifications that are made in this window affect the main RI-ISI table. For example, Figure 4 shows how the row of “pipe5” looks after the detailed piping failure consequence modelling. In the events column, different event combinations are separated by semicolon. Conditional probabilities or combination specific durations are not shown in this table, but they belong to the “pipe5” data and affect the CCDP computation.



Name	Comment	Duration (h)	CCDP	Event	Events
Last value	5. pipe segment	24	-	LPCoolF1	-
pipe1	1. pipe segment	0.00E+00	4.28E-04	New event	LLOC;
pipe2	2. pipe segment	0.00E+00	1.03E-04	New event	SLOC;
pipe3	3. pipe segment	2.40E+01	2.28E-10	New event	LPCoolF1;
pipe4	4. pipe segment	2.40E+01	1.22E-03	New event	LLOC/LPCoolF1;
pipe5	5. pipe segment	0.00E+00	4.24E-04	New event	SLOC; LLOC; LLOC/LPCoolF1;

Consequence
CD

Figure 4. RI-ISI table after detailed piping failure consequence modelling.

5. Simple example case

A simple model with two LOCA scenarios is used to demonstrate the CCDP computation. The PRA model contains two initiating events, large LOCA and small LOCA, and corresponding event trees. The event trees are presented in Figures 5 and 6.

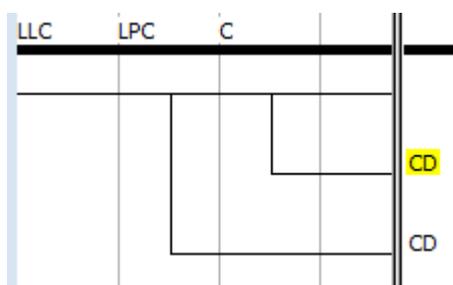


Figure 5. Event tree for large LOCA.

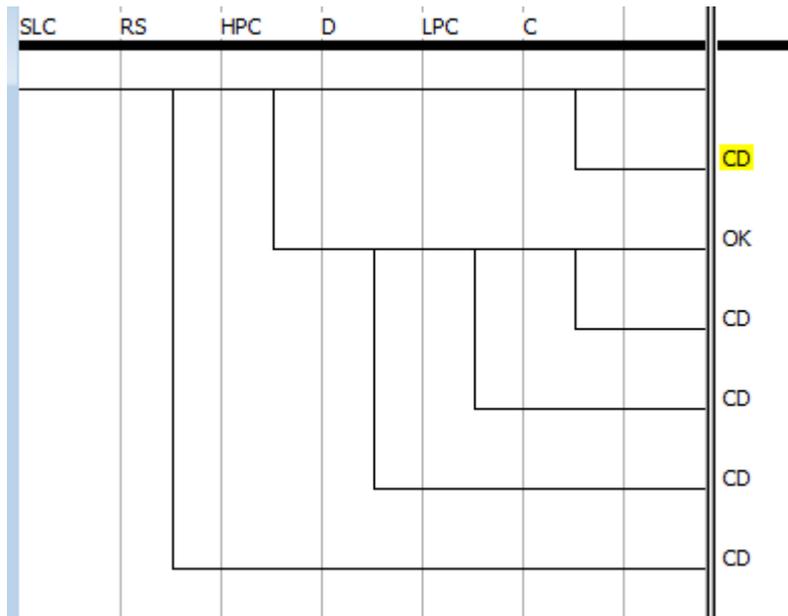


Figure 6. Event tree for small LOCA.

The event trees could correspond to a pressurized water reactor NPP, but they are very simplified and do not represent any actual NPP. In the case of large LOCA (LLC), core damage (CD) occurs if low pressure cooling (LPC) or recirculation cooling (C) does not work. In the case of small LOCA (SLC), core damage (CD) occurs if reactor scram (RS) does not work. If reactor scram works, core damage can be avoided if high pressure cooling (HPC) and recirculation cooling (C) work, or if depressurisation system (D), low pressure cooling (LPC) and recirculation cooling work. For HPC and LPC, two out of four subsystems are enough to provide adequate cooling. Failure probabilities for the safety systems are presented in Table 1. The frequency for large LOCA is $1E-4$ and the frequency for small LOCA is $1E-3$.

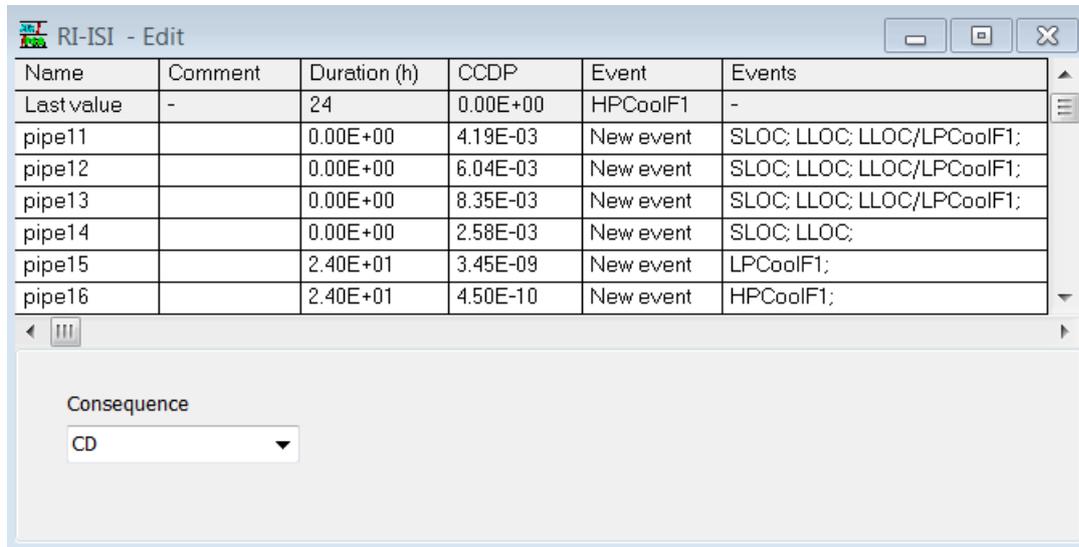
Table 1. Failure probabilities of safety systems.

System	Probability of failure
Low pressure cooling – 1 subsystem	$5E-2$
Recirculation cooling system	$1E-4$
Reactor scram system	$1E-7$
High pressure cooling – 1 subsystem	$5E-2$
Depressurisation system	$1E-2$

Six different types of piping failures are considered possible in the model. They are represented by piping components pipe11-16. Piping failures can cause either a small LOCA (SLOC), a large LOCA (LLOC), combination of large LOCA and failure of one subsystem of low pressure cooling (LLOC/LPCoolF1), failure of one subsystem of low pressure cooling (LPCoolF1) or failure of one subsystem of high pressure cooling (HPCoolF1). It is assumed that cooling functions are out of function for 24 hours if they fail.

Figure 7 and Table 2 present CCDP results for the example model. The middle part of Table 2 shows the conditional probabilities of failure combinations. The same information is

included in the FinPSA model in the same way as in the example of Figure 2. The last column of Table 2 shows the CCDPs of the piping components and the last row shows the CCDPs of failure combinations.



Name	Comment	Duration (h)	CCDP	Event	Events
Last value	-	24	0.00E+00	HPCoolF1	-
pipe11		0.00E+00	4.19E-03	New event	SLOC; LLOC; LLOC/LPCoolF1;
pipe12		0.00E+00	6.04E-03	New event	SLOC; LLOC; LLOC/LPCoolF1;
pipe13		0.00E+00	8.35E-03	New event	SLOC; LLOC; LLOC/LPCoolF1;
pipe14		0.00E+00	2.58E-03	New event	SLOC; LLOC;
pipe15		2.40E+01	3.45E-09	New event	LPCoolF1;
pipe16		2.40E+01	4.50E-10	New event	HPCoolF1;

Consequence
CD

Figure 7. RI-ISI table for the example model.

Table 2. CCDP computation for the example model.

Failure combination	SLOC	LLOC	LLOC/LPCoolF1	LPCoolF1	HPCoolF1	CCDP
pipe11	0.5	0.3	0.2	-	-	4.19E-3
pipe12	0.3	0.4	0.3	-	-	6.04E-3
pipe13	0.3	0.2	0.5	-	-	8.35E-3
pipe14	0.3	0.7	-	-	-	2.58E-3
pipe15	-	-	-	1.0	-	3.45E-9
pipe16	-	-	-	-	1.0	4.50E-10
CCDP	1.48E-4	3.62E-3	1.52E-2	3.45E-9	4.50E-10	

Piping component pipe13 has the highest CCDP (8.35E-3) because it causes both large LOCA and failure of one subsystem of low pressure cooling with the highest probability (0.5). Those piping failures that cause only failures of safety cooling functions, but not initiating events have insignificant CCDPs compared to piping failures that cause initiating events.

6. Software design for the RI-ISI feature

Understanding this section requires some knowledge on the design of FinPSA [12, 13]. FinPSA is programmed in Delphi language [14]. FinPSA contains its own database system. Tables in the FinPSA database are implemented using DataTable unit, and the elements/rows of the tables are descendants of TNode class. TablForm and TablFrame units provide user interface of the tables. The RI-ISI table is another table in the database of

- FRData: TRIISIData;
- FConsequence: string;

FConsequence is the chosen consequence. FRData is a packed record containing the data of a row in the table. FRData contains the following fields:

- FEvents : TDataTable;
- FComponentCode : integer;
- FDuration : single;
- FCCDP : single;

FEvents is a data table that contains the combinations of failures and events that the piping failure can cause (which are represented by TRIISIFailure objects). FComponentCode is an identifier for the RI-ISI component and is not shown for the user. FDuration is the time the piping failure is in effect. FCCDP is the calculated CCDP value.

TRIISI has the following main functions/procedures:

procedure AddEvent(const Value: NameStr);

Adds an event with name Value to a TRIISIFailure object of FEvents. Calls procedure AddEvent of TRIISIFailure to do this. Creates new TRIISIFailure object if there is none in FEvents. Calls setCCDP function in the end.

function GetDataString(F: TDataId): String;

Gives AnsiString for the field F. E.g. in the case of the events column, the function gets the event names from FEvents, combines them in one string and gives it for the field.

procedure SetDataString(F: TDataId; S: String);

Puts data AnsiString for field F. E.g. in the case of the event column, the procedure calls AddEvent function if S is not 'New event'.

procedure Load(F: TByteFile);

Loads TRIISI object. Loads also separately TRIISIFailure objects in FEvents.

procedure Save(F: TByteFile);

Saves TRIISI object. Saves also separately TRIISIFailure objects in FEvents.

function setCCDP(con: String): single;

Calculates CCDP of the RI-ISI component. See algorithms in Section 4.1 and formula in Section 4.2. Before computation, the function loads the minimal cut sets of the consequence (TComputedCutsets object). To calculate CCDP, the function calls function ComputeCCDP which is defined in unit CutLoad for TComputedCutsets object.

procedure CheckProbabilities;

Checks if the sum of the conditional probabilities of the event combinations is over 1.

There are also some standard functions for setting and getting data fields and managing the fields of the table.

In addition, RIISI unit contains function CreateRIISI which creates TRIISI object, and TRIISIFiler object which is a descendant of TReaderFiler and loads TRIISI object when FinPSA is started.

6.2 Unit RIISWindow

Unit RIISWindow defines TRIISForm form that defines the graphical user interface of the RI-ISI table. TRIISForm is a descendant of TTableForm. When the user selects a consequence, procedure CalculateCCDP goes through each piping component in the table and calls SetCCDP function of the corresponding TRIISI objects. Otherwise, the functionality is inherited from TTableForm.

6.3 Unit RIISIFailure

Unit TRIISIFailure has very similar structure as unit RIISI. The unit defines TRIISIFailure object which is a descendant of TNode and defines a combination of events/failures that can be caused by the piping failure. TRIISIFailure represents a row in the event combinations table.

TRIISIFailure contains field FRFData, which is a packed record containing information of the event combination. FRFData contains the following fields:

- FEventCodes : array[0..MaxEventsInCombination-1] of integer;
- FEvents : array[0..MaxEventsInCombination-1] of NameStr;
- FDuration : single;
- FProbability : single;
- FNumberOfEvents : integer;
- FIndex : integer;

FEventCodes contains integer codes of initiating and basic events which are used in many parts of FinPSA to handle data efficiently. FEvents contains the names of initiating and basic events. FDuration is the time the piping failure is in effect. FProbability is the conditional probability of the event combination. FNumberOfEvents is the number of the initiating and basic events. FIndex is the index of the event combination seen in the table.

TRIISIFailure has the following main functions/procedures:

procedure SetDataString(F: TDataId; S: String);

Puts data AnsiString for field F. E.g. in the case of an event field, the procedure calls AddEvent procedure.

procedure AddEvent(const Value: NameStr; num: integer);

Adds an event with name Value to FEvents to a place that is defined by num. Organizes the table so that there are no spaces, if needed.

function GetEventCode(Ind: integer): integer;

Gets the integer code of an event in FEvents determined by Ind and updates FEventCodes.

There are also some standard functions for setting and getting data fields, managing the fields of the table, and saving and loading objects.

The unit contains also function CreateRIISIFailure which creates TRIISIFailure object.

6.4 Unit RIISIDetail

Unit RIISIDetail defines TRIISIDetailFrame form that defines the graphical user interface of event combinations table/window. TRIISIDetailFrame is a descendant of TTableForm.

The form has two fields in addition to the labels seen in the upper part of the window (see Section 4.2 and Figure 2):

- FName: NameStr;
- Table: TDataTable;

FName is the name of the RI-ISI component. Table is the data table of TRIISIFailure objects.

The form has the following procedures:

procedure SetCCDP;

Calls SetCCDP function of TRIISI and sets the calculated CCDP value to the label in the upper part of the window.

procedure PerformAction (TheTag: integer);

This procedure is mostly inherited from TTableForm. However, if insert is pressed, a new TRIISIFailure object is created. Then, the new TRIISIFailure object is added to Table.

The form has also procedures to create itself and to set FName. Otherwise, the functionality is inherited from TTableForm.

6.5 Unit Project

Unit Project defines a FinPSA project as TProject object. Its design is presented in ref. [12]. Here, only a few RI-ISI related procedures are described.

TProject has a procedure ShowDatabaseTable which shows the database table with given ID. In this procedure, RI-ISI table is handled mostly in the same way as other tables. However, TRIISIForm's own create function is called instead of that of TTableForm. It is also defined that if an element is selected in the RI-ISI table (double-clicked or enter pressed on it), ShowRIISIDetails function from unit MainWindow is called.

ShowRIISIDetails function from unit MainWindow calls ShowRIISIDetail function of TProject with the selected element index. ShowRIISIDetail creates a new TRIISIDetailFrame. Data from the selected TRIISI object and its TRIISIFailure objects are set to the TRIISIDetailFrame. Save action is disabled for TRIISIDetailFrame. Hence, all the data is saved via TRIISISWindow.

7. Conditional large early release probability

This section presents how conditional large early release probabilities (CLERPs) could be calculated in FinPSA. The actual software implementation has however not been programmed yet. The section serves as a plan and basis for the next year's work.

In FinPSA, level 1 results are passed to level 2 as plant damage states (PDS). Each PDS has its own containment event tree (CET) in level 2. Using level 1 event trees and interface trees in combination, minimal cut sets are generated for a PDS, and the frequency of the PDS is calculated from the minimal cut sets.

The first step in the computation of the CLERP of a piping failure is to calculate the conditional probability/frequency for each PDS. Conditional plant damage state probability/frequency (CPDSP/CPDSF) can be calculated from the minimal cut sets in the same way as CCDP/CCDF.

Level 2 part of the calculation is more complicated because it contains a complex non-binary simulation model which does not contain basic events like level 1. Level 2 accident sequences are categorised into release categories. However, typically there is not a release category for large early release. Instead, there are multiple release categories that correspond to large early release. The large early release has to therefore be defined as a union of particular release categories. A new function needs to be developed to FinPSA for this.

The simplest computation case is that the analysed piping failure does not affect level 2 at all (except via PDSs), which is also the most common case. The existing level 2 results include mean frequencies of release categories separately for each CET. These frequencies are scaled by the CPDSP/CPDSF values divided by the original PDS frequencies. The resulting mean conditional probabilities/frequencies of the release categories belonging to the large early release from each CET are summed, and the result is the CLERP/CLERF. The CLERP of event E is computed as:

$$CLERP_E = \sum_{t=1}^L \sum_{R \in LER} f_{R(t)} \times \frac{P(PDS_t|E=1)}{f_{PDS_t}}, \quad (5)$$

where L is the number of CETs/PDSs, R is release category, LER is the set of release categories belonging to the large early release, $f_{R(t)}$ is the mean frequency of release category R in t :th CET, $P(PDS_t|E=1)$ is the conditional probability of t :th PDS and f_{PDS_t} is the frequency of t :th PDS. Equation (5) can be generalised for multiple events. The equation for CLERF is similar, except that conditional frequency of t :th PDS is included instead of conditional probability. CLERF values can be transformed into CLERP values in the same way as CCDF values are transformed into CCDP values (see equation (1) in Section 3).

If the analysed piping failure affects level 2, the computation is more complicated. A standard way of modelling piping failures in level 2 is needed. Piping failures or events caused by them should therefore be modelled using Boolean variables so that they can be taken into account in this method. User also needs to define these variables as collected variables so that their values appear in simulation results. Using this approach, the name of a variable can be used in the RI-ISI table just like the name of a basic event.

Existing simulation results can also be utilised when a piping failure appears in level 2 model. Simulation results need to be filtered so that only the results of those sequences where the analysed variable or variables have value 'true' are counted. This means that some simulation cycles can be completely left out and that only some sequences from a particular simulation cycle can be included while the others are left out. The frequencies of the sequences in simulation results are scaled by the CPDSP/CPDSF values divided by the original PDS frequencies. In each CET, the conditional probabilities/frequencies of the sequences leading to large early release are summed for each simulation cycle, and the average value of the sums is calculated (without including those simulation cycles where the analysed variable or variables did not have value 'true'). Finally, the average values of the conditional probabilities/frequencies are summed over all CETs, and the result is the CLERP/CLERF. Now, the CLERP of event E is computed as:

$$CLERP_E = \sum_{t=1}^L \frac{1}{N_t(E=1)} \sum_{i \in S_t(E=1)} \sum_{R \in LER} \sum_{j(i) \in R} f_{j(i)} \times \frac{P(PDS_t|E=1)}{f_{PDS_t}} \times E_{j(i)}, \quad (6)$$

where $N_t(E=1)$ is the number of simulation cycles for which variable E is 1 ('true') in at least one sequence of t :th CET, $S_t(E=1)$ is the set of simulation cycles for which variable E is 1

(‘true’) in at least one sequence of t :th CET, $f_{j(i)}$ is the frequency of sequence j in i :th simulation cycle and $E_{j(i)}$ is the value of variable E in sequence j in i :th simulation cycle. Again, equation (6) can be generalised for multiple events, and the equation for CLERP is similar, except that conditional frequency of t :th PDS is included instead of conditional probability.

If tight integration of PRA levels 1 and 2 is used so that some information from level 1 other than PDS frequencies is used in level 2 computations, it can be difficult to calculate CLERP using existing level 2 results. In some cases, it can be possible to deduce the conditional probabilities/frequencies of level 2 sequences from the existing results only, but otherwise completely new level 2 calculations with adjusted input data are needed.

8. Conclusions

This report introduces a new RI-ISI feature which calculates CCDPs of piping component failures in PRA software FinPSA. The CCDPs of all piping component failures can be calculated automatically at once in the same RI-ISI table, based on the results of the PRA model. This feature does not complicate the PRA model at all, regardless of how many piping components are included. The feature contains two levels and it supports, respectively, both simple and detailed piping failure consequence modelling. If there is one initiating or basic event for each piping component in the PRA model, only the first level of the feature can be used. It is, however, also possible to assign combinations of initiating and basic events to piping components, and even give conditional probabilities to different event combinations, if the user wants to model uncertainty in piping failure consequences.

The new RI-ISI feature is currently just a prototype and is not included in the official version of FinPSA. It can be added to the official FinPSA when users accept or request it.

CLERP is another important consequence measure for piping failures. Its computation was also discussed, but not implemented. CLERP could also be added to the RI-ISI feature. However, its computation is more challenging to implement than that of CCDP. The plan is to implement CLERP computation in FinPSA in 2017.

The CCDP calculation feature could also be applied to other events than piping failures, such as fires. Actually, it can be used calculate any conditional consequence probabilities.

Currently, the RI-ISI feature calculates only CCDPs, but RI-ISI analyses require also failure probabilities of piping components. It could be possible to add failure probabilities to the RI-ISI table of FinPSA and implement some sort of inspection interval decision rules or optimisation algorithm. However, the approach to determine inspection intervals should be decided first. On the other hand, it is also simple to export CCDP values to other applications, where inspection interval optimisation can be performed based on the results.

References

1. International Atomic Energy Agency. Risk-informed in-service inspection of piping systems of nuclear power plants: process, status, issues and development. IAEA Nuclear Energy Series, No. NP-T-3.1. Vienna, 2010. ISBN: 978-92-0-103410-6.
2. International Atomic Energy Agency. Development and application of level 1 probabilistic safety assessment for nuclear power plants. Specific safety guide, No. SSG-3. Vienna, 2010. ISBN 978-92-0-114509-3.

3. Tyrväinen T. Connection between PRA and RI-ISI analyses. VTT Technical Research Centre of Finland Ltd., Espoo, 2015. VTT-R-04536-15.
4. Mätäsniemi T, Tyrväinen T, Björkman K and Niemelä I. FinPSA knowledge transfer (FINPSA-TRANSFER). In: SAFIR2014 – The Finnish Research Programme on Nuclear Power Plant Safety 2011-2014, Final Report. VTT Technical Research Centre of Finland, Espoo, 2015.
5. Vesely WE, Goldberg FF, Roberts NH and Haasl DF. Fault tree handbook. United States Nuclear Regulatory Commission, Washington, D.C., 1981. NUREG-0492.
6. Tyrväinen T, FinPSA – Software testing – Verification and validation plan version 1.2. VTT Technical Research Centre of Finland, Espoo, 2014. VTT-R-04125-14.
7. Tyrväinen T, FinPSA – Software testing – Verification and validation results for version 1.0.0.6. VTT Technical Research Centre of Finland Ltd., Espoo, 2015. VTT-R-03715-15.
8. Gandossi L. Risk-informed in-service inspection: an overview for the non-specialist. European Commission, Joint Research Centre, Petten, 2012. ENIQ report No 46.
9. Van Der Borst M and Schoonakker H. An overview of PSA importance measures. Reliability Engineering and System Safety, 72, 2001, 241-245.
10. Hietanen O, Jänkälä K, Bergroth N, Paussu R and Nikula V. Risk-informed methodology of new ISI program for unit 1 of Loviisa NPP. 6th international conference on NDE in relation to structural integrity for nuclear and pressurized components, Budapest, 2007.
11. VTT Technical Research Centre of Finland. FinPSA manual. Espoo, 2013.
12. Niemelä I and Mätäsniemi T. FinPSA – Software design – Main unit documentation. VTT Technical Research Centre of Finland, Espoo, 2013. VTT-R-00353-13.
13. Niemelä I and Mätäsniemi T. FinPSA – Software design – Supporting unit documentation. VTT Technical Research Centre of Finland, Espoo, 2013. VTT-R-00351-13.
14. Embarcadero Technologies, Inc..Delphi. 2016.
<http://www.embarcadero.com/products/delphi> [Referred 9.3.2016].