Implementation of Conditional Quantification in RiskSpectrum

Pavel Krcal[a], Pengbo Wang[a], and Ola Bäckström[a]
[a] Lloyd's Register, Stockholm, Sweden,
{pavel.krcal, pengbo.wang, ola.backstrom}@lr.org

Abstract: Basic events in Probabilistic Safety Assessment (PSA) models are typically quantified independently of the accident sequence and of other failures that lead to a system unavailability. This simplifies quantification of undesirable consequences and in most situations, this approximation does not distort safety indicators. However, there are emerging needs for dependency handling between basic events such as (1) dependencies between operator actions, (2) correlations between events in PSA, e.g., incurred by seismic events, and (3) common cause failure modeling. In these situations, improved handling of dependencies could yield more realistic analysis results and by this increase applicability of safety indicators. Conditional quantification of basic events presents a flexible, simple, and transparent tool to model these dependencies. At the same time, it poses theoretical and algorithmic challenges to analysis tools. We describe the implementation of the first release of this feature in RiskSpectrum PSA (version 1.5.0, released in 2021). This implementation keeps the application to the human failure event dependencies in focus. The paper highlights the mathematical and algorithmic choices and presents the applied solution.

1. INTRODUCTION

Basic events in Probabilistic Safety Assessment (PSA) models are typically quantified independently of the accident sequence and of other failures that lead to a system unavailability. This simplifies quantification of undesirable consequences and in most situations, this approximation does not distort safety indicators. However, there are emerging needs for dependency handling between basic events such as (1) dependencies between operator actions, (2) correlations between events in PSA, e.g., incurred by seismic events, and (3) common cause failure modeling. In these situations, improved handling of dependencies could yield more realistic analysis results and by this increase applicability of safety indicators.

Conditional quantification of basic events presents a flexible, simple, and transparent tool to model these dependencies. At the same time, it poses theoretical and algorithmic challenges to analysis tools. We describe the implementation of the first release of this feature in RiskSpectrum PSA (version 1.5.0, released in 2021) [1], focusing on the choices taken and solutions applied. The aim of this feature is to enable users to specify conditional probabilities of basic events when needed and appropriate, focusing mainly on Human Reliability Analysis (HRA) applications (see [2] for an overview). The solution treats conditional quantification of basic events correctly throughout the whole analysis span, starting with the generation of minimal cut sets (MCS), quantification of the generated MCS list (including the MCS BDD algorithm [3]), merging and post-processing of MCS lists, as well as importance, sensitivity, time-dependency and uncertainty analyses.

Dependency treatment for operator actions removes undeserved bonus when accounting for several human failures within one scenario. Subsequent failures might depend on the fact that the operator has already failed with a previous action. Conditional quantification will then conservatively increase the human error probability, typically according to one of the pre-defined formulas specified in the applied HRA method. The implemented algorithm allows users to specify conditional probabilities as a part of the model and then run a single analysis that efficiently generates all cut sets and correctly applies the cutoff so that no cut set with the value above the cutoff after the conditional quantification is lost. Basic events that obtain a new value from conditional probability are treated as separate events. This resolves possible under-approximations due to the success treatment of conservatively estimated dependent basic events, especially in the MCS BDD quantification of the MCS list. Experiments on industrial-sized models show that our method, compared to the standardly used HRA event replacement in post-processing [4], can efficiently generate minimal cut sets which would be otherwise missing or discarded.

2. PRELIMINARIES

Conditional probability is the probability of one event occurring given that one or more other events have already occurred. For example, P(A|B) is the conditional probability of event A under the condition that B has occurred. The event A is called a dependent event, and the event B is called a condition event. A condition can contain one or more basic events. A dependent event has a reliability model with associated parameters which determines its independent value. This value is used in cutsets where no condition for this dependent event is satisfied. If a cutset contains a dependent event and at the same time a condition is satisfied in this cutset then a conditional probability is used for this dependent event.

Knowing the conditional probability is sufficient for the first order quantification of minimal cut sets. Higher order quantification or quantification by a Boolean Decision Diagram (BDD) requires answering how do we estimate the probability of A when B has not occurred. In other words, what is the conditional probability P(A|-B), where -B denotes success of B. [5] presents detailed argumentation for different options. Here we summarize the main results.

We obtain the most precise estimate if we can fully account for event success. Then we have that $P(A) = P(A|B) \cdot P(B) + P(A|{-B}) \cdot (1-P(B))$ and hence $P(A|{-B}) = [P(A) - P(A|B) \cdot P(B)]/(1-P(B))$. This implies that the success of B would decrease the probability of A. This decrease will be determined by the conditional probability P(A|B).

The main goal of the dependency analysis for HFEs is to avoid underestimating the contribution from human failures to the consequence top frequency. Dependencies are chosen to safely over-approximate the actual degree of dependency. Assume that an analyst adds a new HFE B that mostly succeeds, but the dependency analysis classifies it as strongly influencing another HFE A. The analyst considers this as safely conservative. For a sequence that contains only success of B, the probability for A would significantly decrease if we took success into account. This might be an undeserved safety bonus because it is caused by the conservative estimate of the dependency.

Another possible estimate of the conditional probability P(A|-B) disregards from event success and sets conservatively P(A|-B) to be equal to P(A). When we ignore the success in conditional quantification then adding B will not influence top value of this sequence, which is probably consistent with the analyst’s intention. We choose this estimate for the implementation of conditional probabilities, which makes it suitable for HRA applications with the typical dependency quantification methods.

Note that the calculation method ignoring dependent success applied to a minimal cut set corresponds numerically to a generally accepted method which replaces the dependent events by a new event or adds a multiplication factor to the cut set in post-processing. Whenever there is a combination of basic events in a cut set that has a dependency, this dependency will influence the cut set value. Otherwise, basic events keep their original probabilities. Conditional probabilities offer significant advantages over post-processing in two aspects. First, we specify dependencies directly by conditional probabilities instead of post-processing actions. Secondly, this method keeps the connection to original basic events, which plays an important role in, e.g., importance analysis.

Refinements of the HFE quantification methods with advanced dependency modeling, e.g., [6,7,8,9] might make it possible to include the full success treatment in the analysis. These methods refine the way in which root causes of dependencies are characterized and quantified.

Conditional quantification brings also algorithmic challenges. First, analysis of event trees and fault trees with conditional probabilities has to scale to industrial-size PSA models. One of the efficiency cornerstones is the cutoff. During the generation of minimal cut sets (MCS), cutoff needs to be effectively and correctly applied, especially for cut sets containing dependent event, since the value of dependent events depends on other events in the same cut set.

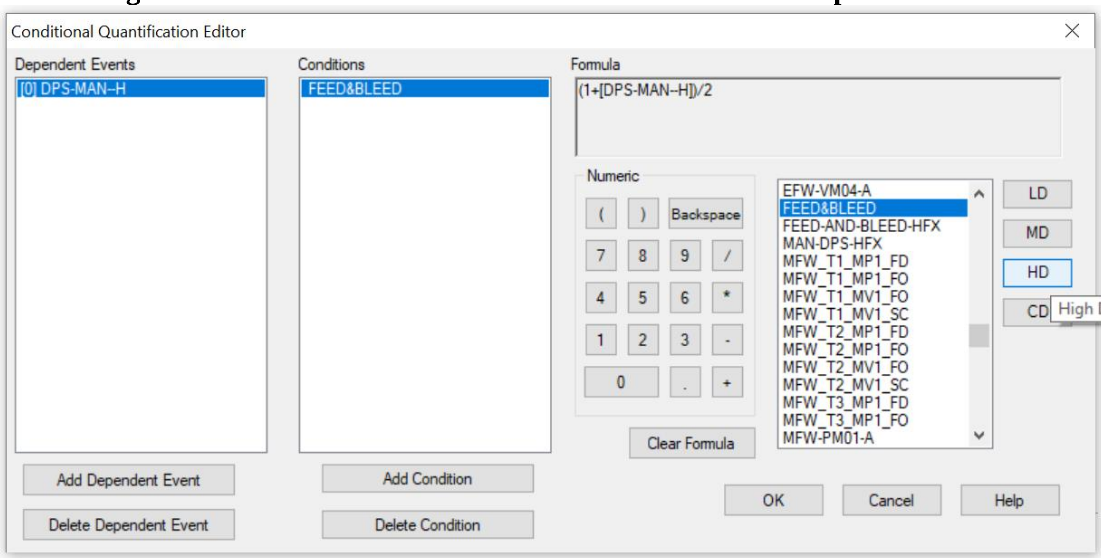
Quantification of generated minimal cut sets poses another difficulty. Quantification using the MCS BDD algorithm, and the 2nd order and 3rd order approximations shall improve over the first order estimate, and it should again scale for large minimal cut set lists. Subsequent analyses based on a generated MCS list also need to account for conditional probabilities correctly. The dependent event and its corresponding independent event should be considered together for importance, sensitivity, time-dependency, and uncertainty analysis. Merging, post-processing and processing of the results in the MCS Editor should minimize MCS lists correctly, i.e., events shall be treated in the same way irrespective of whether they are quantified conditionally or not. But they should be quantified conditionally if the conditioning events are present in the cut set.

3. MODELLING DEPENDENCIES BY CONDITIONAL PROBABILITIES

Dependencies between human failure events are typically identified and quantified in a dedicated software, such as RiskSpectrum HRA. Once this work has been done, an analyst updates the PSA model to also reflect these dependencies. Conditional probabilities make it possible to specify dependencies without the need to add new events and without utilizing rules in post-processing actions. Users enter the minimal necessary information on top of the existing PSA model.

There is a dedicated editor that allows to specify conditional probabilities in the form $P(A|B_1 \ldots B_n) = f(A)$, where f is a function modifying the input value by simple arithmetic operations. This function can be specified by the user or selected from a list of pre-defined dependency levels. Figure 1 depicts this editor.

Figure 1: The editor for Conditional Probabilities in RiskSpectrum PSA

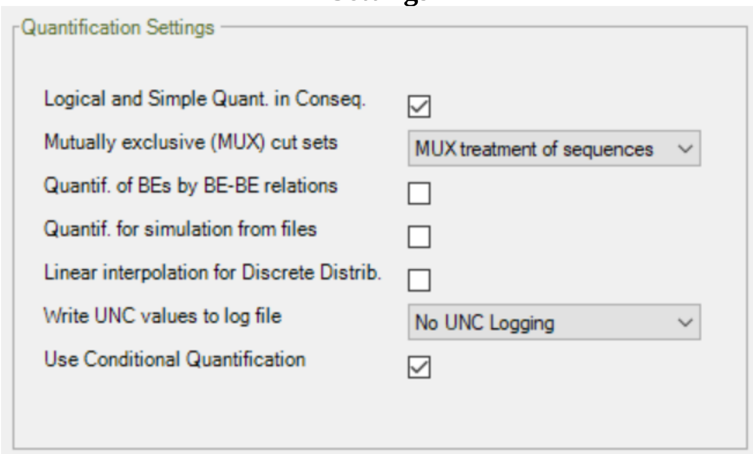


Conditional probability can be defined as a constant value, or as a function of the dependent event probability. E.g., the probability that A occurs given that B has occurred can be 0.3; or the probability that A occurs given that B has occurred can be $(1+6 \cdot P(A))/7$. Each conditional probability must be greater or equal to the original (independent) basic event probability.

4. NEW WAY OF ANALYSIS (HRA)

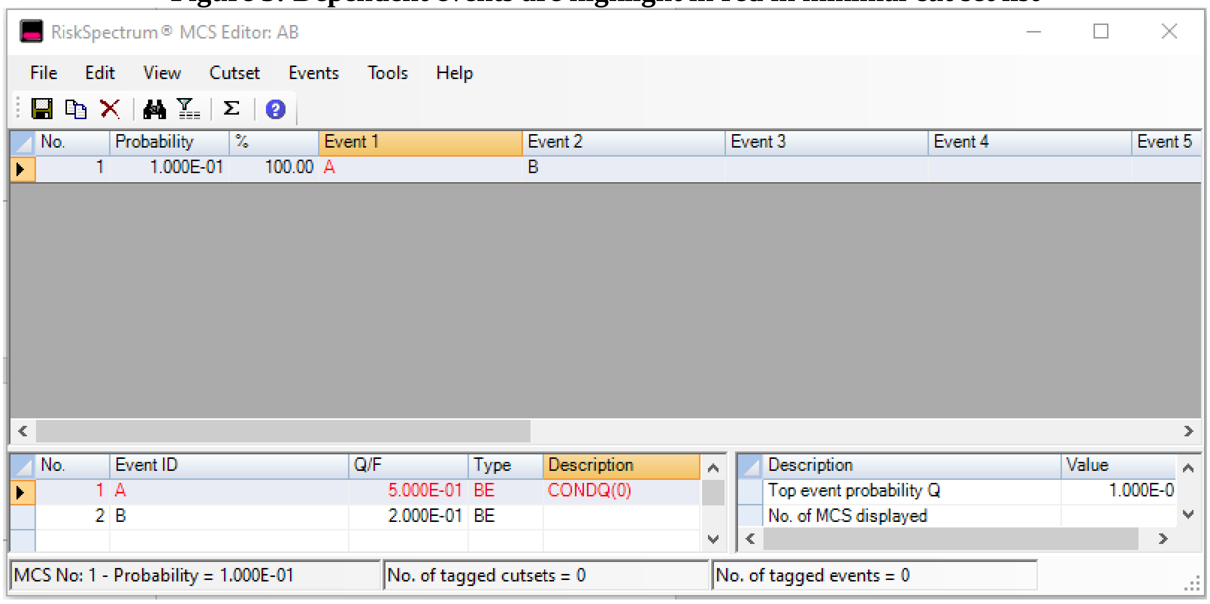
Analysis takes dependencies into account automatically. There is no need to adjust cutoff levels, to modify event probabilities or to run post-processing. All settings and parameters can stay the same as in an analysis without conditional probabilities. This means not only a simplified analysis procedure, but also a reduced chance to make a mistake in some of the analysis steps. Users must select in the Quantification Settings of the Analysis Settings whether they want to use conditional probabilities defined in the model. If this option is selected, then the application of conditional quantification is automatic.

Figure 2: It is possible to enable or disable conditional quantification of events in Analysis Settings



In the minimal cut set list displayed in the Results tab of RiskSpectrum PSA and MCS Editor, dependent events are highlighted in red, and their related conditional probability can be traced back very easily. As shown in Figure 3, event A is a dependent event, with P(A|B)=0.5; the generated minimal cutset list contains one cutset {AB}, in which A is marked in red, its mean value takes the defined conditional probability 0.5. Conditional probability events get a prefix to their Descriptions: CONDQ(x), where x is the number of the conditional probability used for their quantification. In this case, the conditional probability number 0 has been used.

Figure 3: Dependent events are highlight in red in minimal cut set list



5. HIGH LEVEL DESIGN

MCS generation requires a conservative quantification when applying cutoff to (partially) generated cut sets. By this, no cut set gets discarded when it should not. Events that may be quantified as dependent events need special treatment during the cutoff procedure, as their probability may be different in different cut sets. When a cut set is fully generated, it is quantified exactly with the conditional probabilities. If it falls below the cutoff, then it is discarded before it is stored in the result MCS list. This means that we do not have to worry about unnecessary cutoff level increases due to generating too many cut sets that should be discarded.

Conditional quantification in generated cut sets is taken care of by the following construct. For each conditional probability defined, RSAT creates internally an additional event with the same ID as the original basic event, we call it "shadow events". If the condition is satisfied in a cut set, the original basic event is replaced by the conditional event, and its mean value is calculated according to the conditional probability. For example, if P(A|B) is defined, a shadow event A' is defined, A' have the same ID as basic event A; for a cut set {A, B, C}, A is replaced by A' and the cut set becomes {A', B, C} with P(A') = P(A|B). If several conditions are satisfied as the same time in a cut set, the original basic event is replaced with the shadow event having the highest probability. This corresponds to the conditional probability quantification which disregards event success.

Basic event and its shadow events share the same ID and are considered as a same event during the generation of minimal cut sets, during the minimization of the cut set list in MCS analysis cases, and in importance analyses. This provides the transparency of the analysis because users see only the basic event which they have defined. This basic event is quantified differently depending on the context.

6. MORE DETAILS ON IMPLEMENTATION

This section presents additional details on the actual implementation of the conditional quantification, highlighting the choices made and explaining consequences of these choices.

6.1. MCS Generation

Generating minimal cut sets with a cutoff has to satisfy two criteria:

- All minimal cut sets above the cutoff are generated [*Correctness*]
- Calculation time is acceptable for the purpose [*Efficiency*]

Conditional quantification might increase the value of a basic event if conditioning events are added to the cut set during the generation process. This can move a discarded cut set above the cutoff. Therefore, keeping the original event mean values might violate the correctness condition. A safely correct solution might be to set all events with conditional probabilities to 1.0, which hampers the efficiency of the cutoff procedure and possibly also exceeds the limit on the maximal number of minimal cut sets. This could in its turn lead to unnecessary increases of the cutoff and a decreased precision of the result. Recalculating the conditional probability each time a new basic event is added to a cut set would also increase the calculation time, especially for large-scale models.

The solution chosen in our implementation is a compromise between the choices above. We apply a fixed mean value for all basic events which still guarantees the correctness. Before a minimal cut set is stored in the result MCS list, it is quantified exactly, also with respect to conditional probabilities. This ensures that the algorithm does not need to increase the cutoff level unnecessarily.

A necessary condition for this is that basic events related to conditional probabilities cannot be a part of a module. This includes dependent basic events, i.e., events that have a conditional probability, and condition basic events, i.e., events which are a part of a condition in a conditional probability.

6.2. MCS Quantification

The first order quantification (rare event approximation, min cut upper bound) quantifies minimal cut sets in isolation including the conditional probabilities. This does not introduce any overheads or difficulties for these algorithms. The top value is clearly a conservative estimate.

For the second order and the third order approximation, we need to consider dependencies between minimal cut sets. The main goal here is to ensure that the second order approximation gives a result which is below the actual top value and the third order approximation returns a conservative estimate. To this end, different dependent events are considered as different events for union of cutsets, even if they share the same ID or come from the same original event. The MCS BDD quantification algorithm also considers dependent events and the original event as different events.

Currently, we do not quantify cut sets in success modules with conditional probabilities. There are pros and cons for both options. A reason for quantifying SMs also conditionally is that the success and failure branch will sum up to 1.0. A reason against is that conditional probabilities are conservatively estimated and here we would use them to calculate success which could underestimate the top value. Also, one can get different results depending on how event trees are built. Consider the event trees in Figure 4a and Figure 4b.

Figure 4a: An event tree where the first three sequences encode success of (A AND B)

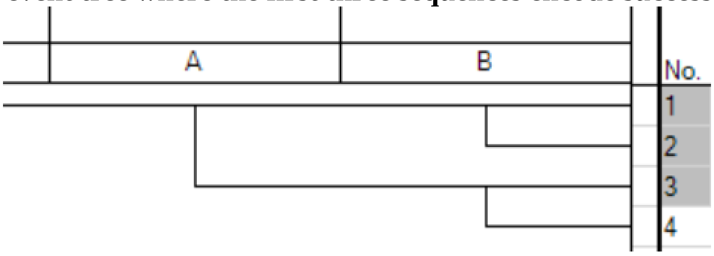
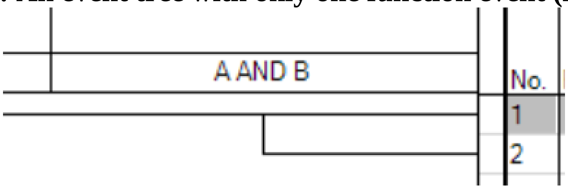


Figure 4b: An event tree with only one function event (A AND B)



The sum of the sequences 1-3 in the event tree from Figure 4a should be the same as the first sequence in the event tree from Figure 4b. This is not the case if P(A | B) = 0.9 > P(A) and if we quantify cut sets in SMs conditionally. There are three success modules in the first ET and none of them contains a cut set with both A and B. Conditional quantification does not apply here. The SM from the second event tree contains one cut set {A, B}, where A would be quantified conditionally. This would give us a different result from the first ET, but the sum of sequences in the second ET would give us the initiating event. This is not the case in the first ET, irrespective of how we quantify success modules.

6.3. Importance Analysis

A conditional event can have different mean values under different conditions. Importance measures of this event should consider all of those different conditions, including the event as an independent event and as dependent events under each condition.

If a basic event has a conditional probability defined, importance of the basic event includes the original basic event and all related conditional events (shadow events). For each given conditional dependency between event B and A, A|B, the event A is replaced with A’ in any cutset containing both B and A. Concretely, for a basic event A:

- FV(A) = Q(cutset contains A or any A’) / Q(all),
- RIF(A) = Q(A = 1, and all A’ = 1) / Q(all),
- RDF(A) = Q(all) / Q(A=0, and all A’=0).

For example, let us consider events A, B and M, with B|A. The minimal cut set list is {AM, MB, AB}. B is replaced by B' in the cutset that contains both A and B. Then the cut set list becomes {AM, MB, AB'}. We note Q(all) = Q({AM,MB,AB}), then importance factors for basic events A and B are:

- FV(A) = Q(has A) / Q(all) = Q(AM+AB') / Q(all),
- FV(B) = Q(has B or B') / Q(all) = Q(MB+AB') / Q(all),
- RIF(A) = Q(A=1) / Q(all),
- RIF(B) = Q(B=1, B'=1) / Q(all),
- RDF(A) = Q(all) / Q(A=0),
- RDF(B) = Q(all) / Q(B=0, B'=0).

If the original event is included in an "attribute/ component/ system/ event group", then all related shadow events are added into the same "attribute/ component/ system/ event group".

6.4. Sensitivity analysis

Similar to the importance analysis, if a basic event has conditional probability defined, the sensitivity of the basic event includes the sensitivity of the original basic event and all related dependent events. When a conditional probability is defined as a constant value, it is not multiplied/divided by sensitivity factor. If it is defined as a function of the original basic event, the conditional probability is recalculated with the nominal value of the basic event multiplied/divided by sensitivity factor.

6.5. Merging, post-processing, and MCS Editor

For merging, post-processing and minimization in MCS Editor, shadow events should behave as their corresponding original events. When we have obtained the final MCS list then we apply conditional quantification by replacing dependent events by shadow events whenever these apply.

This way ensures that the resulting MCS lists are minimal, and that conditional quantification is applied correctly with respect to the final form of minimal cut sets.

6.6. Restrictions for conditional probability

There are several limitations on how conditional probabilities can be used in the current implementation in RiskSpectrum PSA. The reason for these restrictions is a simplification of the solution, both theoretically and on the efficiency level. These restrictions should not limit usability of conditional probabilities, especially for modelling of dependencies between HFEs. The limitations are:

- Frequency events cannot be defined as dependent events.
- Dependent events cannot be included in MUX Sets.
- Events having definition by other basic events cannot be defined as a dependent event.
- Events used in a CCF group cannot be used for conditional probability, neither as a dependent event nor in a condition.
- I&AB method does not support conditional probabilities.
- It is not possible to specify negated events as a part of a condition.
- Conditional probability is not applied in success modules.

7. EXPERIMENTAL RESULTS

Here we report experiences with the conditional quantification applied to HFE dependencies on an industrial sized PSA model. This model has over 200 post-processing actions related to HRA dependency modeling. The analysis uses nominal mean values for HFEs and decreases the overall cutoff value to try to prevent losing minimal cut sets. Analysis times for different cutoff levels are summarized in Table 1. The following cases are analyzed:

- Case 1: cutoff 1E-09, no conditional quantification. Model is run as is, and then post-processing is applied.
- Case 2: cutoff 1E-12, no conditional quantification. Model is run as is, and then post-processing is applied. The lower cutoff is applied as a countermeasure for the increases in HFE probability due to dependencies.
- Case 3: cutoff 1E-09 with conditional quantification.
- Note: Case 1 and 2 cannot be run with setting all HFEs to probability 1. It is not possible to calculate results.

Table 1: Calculation comparison for an industrial-size model

Case	Cutoff / Conditional Quantification	Number of MCS	Top value	Calculation time (s)
1	1E-09, No	8850	Not relevant	824
2	1E-12, No	619311	8.8E-5	31093
3	1E-09, Yes	18232	6.0E-4	13969

This experiment shows that the conditional quantification produces more complete results in a significantly shorter time than the analysis with post-processing (compare case 2 and 3). The conditional quantification slows down the calculation compared to the same cutoff level (case 1 and 3) without the conditional quantification, which is not surprising (but note that the results from case 1 are irrelevant). On the other hand, even the lower cutoff level does not prevent losing cut sets due to the cutoff.

Another aspect to consider in analysis without the conditional quantification is that the post-processing does not cover multiple dependencies in a single cut set. Only one exchange is performed and afterwards, if the cut set does not satisfy the condition for further dependencies, then they are not applied. This is an incorrect application of post-processing that can be corrected, but it illustrates the advantage of the transparent dependency modeling by conditional probabilities. An analysis with setting all HFEs to 1.0 to ensure generating all combinations was not computationally feasible for the example model.

8. CONCLUSIONS AND FUTURE WORK

The new conditional quantification feature in RiskSpectrum PSA presented in this paper enables users to specify conditional probabilities of basic events when needed and appropriate, focusing mainly on Human Reliability Analysis (HRA) applications. The solution treats conditional quantification of basic events correctly throughout the whole analysis span, starting with the generation of minimal cut sets (MCS), quantification of the generated MCS list (including the MCS BDD algorithm), merging and post-processing of MCS lists, as well as importance, sensitivity, time-dependency, and uncertainty analyses. Experiments on industrial-sized models show that our method, compared to the standardly used HRA event replacement in post-processing, can efficiently generate minimal cut sets which would be otherwise missing or discarded.

Further development will focus on extending conditional quantification for other applications. Common cause failure modeling might benefit from a more flexible way to specify dependencies between events. For instance, not fully symmetrical situations might be transparently modeled by specifying conditional probabilities. Correlations between seismic events represent another possible area for conditional quantification, where this concise way of specifying dependencies might improve both modeling and result precision.

References

[1] RiskSpectrum Risk and Reliability Software. www.riskspectrum.com

[2] He, X. "*Dependencies in HRA*". Technical Report 41-001:01, Nordic PSA Group (NPSAG). (2015)

[3] W. Wang, O. Bäckström, P. Krcal. "*Quantification of MCS with BDD, Accuracy and Inclusion of Success in the Calculation – the RiskSpectrum MCS BDD Algorithm*", In Proc. PSAM'12, Honolulu, USA, (2014).

[4] Grobbelaar, J., M. Hirt, and M. Presley. "*Human Reliability Dependency Analysis and Model Integration Process*". In Proc. PSAM'12, Honolulu, Hawaii, USA. (2014)

[5] P. Krcal, X. He, P. Wang, O. Bäckström. "*Conditional Quantification in PSA*", In Proc. PSAM'15, Venice, Italy, (2020).

[6] N. J. Ekanem and A. Mosleh, "*Phoenix - a model-based human reliability analysis methodology: Quantitative analysis procedure and data base*", In Proc.PSAM'12, Honolulu, Hawaii, USA, (2014).

[7] R. Boring. "*A dynamic approach to modeling dependence between human failure events*", In Proc. ESREL'15, Zurich, Switzerland, (2015).

[8] J. Peschke and M. Roewekamp. "*A Method for Modeling Human Behavior as a Dynamic Process in the Context of External and Internal Hazards*", In Proc. PSAM'14, Los Angeles, CA, USA, (2018).

[9] G. Petkov. "*Symptom-based Approach for Dynamic HRA and Accident Management*", In Proc. ESREL'18, Trondheim, Norway, (2018).