Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P4859-cd

Multi-Unit Analyses across Application Domains

Ola Bäckström
RiskSpectrum AB, Stockholm, Sweden. E-mail: ola.backstrom@riskspectrum.com

Pavel Krcal
RiskSpectrum AB, Dresden, Germany. E-mail: pavel.krcal@riskspectrum.com

Reliable systems such as nuclear power plants, oil tanks, computation clusters, wind farms, or satellites fail rarely. One could argue that studying accidents of an individual unit provides sufficient insight for risk-informed decision making. This approach might hide potential risks that arise from the fact that there are multiple units with a similar risk profile that share dependencies. A nuclear power plant might consist of several units located at the same site, an oil terminal contains almost identical tanks, and a wind farm comprises multiple wind turbines. Studying the effects of dependencies between units might increase risk understanding and provide new perspectives for system design and operation. We investigate feasibility of a multi-unit analysis for applications where models for individual units exist. The analysis methods might differ on the basis of the size and complexity of the models. Large nuclear power plant fault trees might require different algorithms than significantly smaller models of oil tanks. Our exploration is based on the multiunit sequence method developed for the nuclear industry. We also include comparisons with analyses that use Monte Carlo simulations.

Keywords: Multi-unit risk, Dependencies, Multi-unit sequences, Analysis algorithms.

1. Introduction

Risk-informed decision making greatly benefits from mathematical models describing reliability or availability characteristics of a system in a formal manner enabling automated analysis by computer codes. Some types of technical utilities consist of a series of similar systems – *units*. For reliable systems, an analysis of a single unit brings a lot of understanding of utility vulnerabilities and strengths. Consequences of a single unit failure might be severe enough to invest in preventing it happen.

Decision makers might be still interested in understanding risks connected to the whole site or the whole connected utility. Failures of multiple units with a similar risk profile cannot be fully derived from single unit models when they share dependencies or can suffer from an external interference that affects several units at the same time. A nuclear power plant or an oil terminal consist of several units located on the same site where it is practical to share some support systems, a computer cluster contains many identical computers interconnected in the same building, satellites might form a constellation, and wind farms contain multiple wind turbines exposed to the same weather conditions. Studying the effects of dependencies between units or their vulnerabilities to external shocks provide additional value for risk management and wider perspectives for decision makers.

Risk models from different industries and applications differ in the mathematical formalisms used, scope of the analysis, their size, resolution and complexity. One of the widely used methods utilizes fault trees and event trees. We investigate feasibility of a multi-unit analysis for applications where models for individual units exist. Methods for analysis of dependencies between units might differ based on the size and complexity of the models.

We present the Multi-Unit Sequence Method Holmberg et al. (2019); Bäckström et al. (2024) developed for the nuclear industry where multi-unit analyses attract significant attention IAEA (2019, 2023); EPRI (2021); Zhou et al. (2021). Different application domains serve as examples on which we discuss applicability of the Multi-

Unit Sequence Method together with the types of restrictions and approximations that models from these domains. We also include comparisons with analyses that use more dynamic models than fault trees and evaluate them by Monte Carlo simulations.

2. Types of Models

Different types of industrial systems utilize mathematical models for reliability and availability that vary in the complexity and behavior aspects captured by their logical structure. Some systems require detailed modeling of physical phenomena, others rely on combinatorial decomposition of potential failure scenarios, while some focus on the evolution of the system in time and its reactions on unexpected events. Let us illustrate this on a couple of examples.

2.1. *Oil tanks in an oil terminal*

Oil tanks represent a relatively simple type of system. Failure behavior arises from leaks of stored fuel that can be caused, for example, by pipe ruptures. Ignition sources and wind speed and direction determine afterwards, whether a leak leads to a fire or an explosion. The analysis of the complete terminal with multiple tanks might rely on the frequencies of local incidents Alzbutas et al. (2021) or model local incidents by simple fault trees Fuentes-Bargues et al. (2017).

The breakdown of local incidents is important in analyzing dependencies between failures of different tanks within the same terminal. When the frequencies of incidents are obtained from a physics models or from operating experience, it behaves as a black-box for the multi-unit analysis. We might derive physics models for failures of multiple tanks or we might collect operating experience also for multiple failures.

Fault tree models could be used to also consider dependencies or the impact of external factors. The method based on Common Cause Failure (CCF) modeling described in Section applies also here. A single integrated fault tree model would probably require an acceptable analysis time with modern fault tree solvers.

2.2. *Nuclear power plants*

Probabilistic models for nuclear power plants consist of event trees and fault trees. The combination of event tree sequences and fault trees for major accidents of individual units such as core damage or large early release of radioactivity yield very large combinatorial models. The underlying fault tree can consist of tens of thousands of gates and basic events. On the other hand, the dependencies between individual units on the same site are kept relatively low compared to the single unit models.

Creating an integrated model for a multi-unit scenario IAEA (2023) might challenge the capabilities of the current event tree/fault tree solvers. For many nuclear Probabilistic Safety Assessment (PSA) studies, an integrated model will be out of the scope of modern solvers Zhou et al. (2021). Approximate approaches such as the Multi-Unit Sequence Method described in Section 3.1 must be applied, so that the important dependencies will be included in the calculation and the results help building insight into the importance of these dependencies.

2.3. *Nuclear fuel reprocessing plants*

Another example from the nuclear domain deals with a fuel reprocessing plant Yamamoto et al. (2024). A utility of this type contains a larger number of similar units that share support systems such as cooling. By this, the multi-unit aspect of the overall risk becomes much stronger than for nuclear power plants. Although the reprocessing units themselves are not as complex as nuclear power plant units, analysis of the integrated model becomes also computationally infeasible without employing any approximations. The Multi-Unit Sequence Method (Section 3.1) can be applied also here.

2.4. *Wind farms*

The renewable energy sector has grown considerably over the past several decades. Wind farms with many identical wind turbines became a common part of the scenery both inland and offshore. One of the factors that influences the failure rate of the individual wind turbines is the wind velocity. This typically affects all wind turbines located

within the same wind farm. It also varies on hourly basis. only this dependency calls for a more dynamic model in the sense that the evolution of the system and its environment over time plays a crucial role in the failure behavior. Fault trees represent a static picture of a system.

A dynamic model that captures a state of the system and its evolution in time requires different analysis methods than an event tree/fault tree model. Typically, Monte Carlo simulations are used. There are multiple modeling formalisms for such systems, including Stochastic Petri Nets, Dynamic Fault Trees, Markov Processes, or Knowledge Bases written in the Figaro modeling language. A multi-unit analysis in this context can benefit from the concepts developed for static models. Dependencies can be modeled as Common Cause Failures, including the effect of external events. The dynamic model must include the possibility to fail components in several units by a single, multi-unit cause.

A multi-unit analysis benefits from a structured approach where adding or removing units (wind turbines, in this case) does not require larger changes in the model. In the ideal case, this should require only selecting the type of the added unit and specifying its relations to the rest of the plant. We investigate some properties of a model written in the Figaro language in Section 4.1.

3. Event Tree/Fault Tree Models

This section introduces multi-unit analysis algorithms for models based on fault trees and possibly also event trees. These formalisms are used for probabilistic safety assessment in many domains, for example nuclear, aerospace, automotive, energy, and transportation. Fault trees present a static view of failure scenarios. Even large models with a detailed resolution can be handled by current analysis tools. The results give insight into overall risk profile and also ranking of systems and components according to different measures of their contribution to the risk.

Event trees enable a structured way of modeling scenarios leading from an accident initiating event through a sequence of successful or failed application of mitigation barriers. These scenarios might lead to an undesired consequence or to a safe end state of the accident sequence. Event trees refer to fault trees as models for the mitigation barrier failures and possibly also for the initiating event. In its turn, a set of sequences in an event tree can be translated into a single fault tree where a failure of the top gate represents reaching the end of at least one of the sequences.

We assume that there are event/fault tree models for individual units. Moreover, we have identified undesired consequences relevant also for multi-unit scenarios. This is by itself in many cases a non-trivial task with a lot of research defining methodologies for specific applications IAEA (2019, 2023); EPRI (2021). In the nuclear domain, such a consequence can be a damage of the reactor core (Core Damage, CD). This gives us a set of event tree sequences that reach this consequence and the equivalent fault tree that represents the same failure logic, so called *master fault tree*.

An essential step in a multi-unit analysis defines dependencies between the units. There can be different sources and types of dependencies:

(i) Initiating events (if present in the model) might start an accident sequence in multiple units.
(ii) Failures modeled by basic events in several units might be correlated. The occurrence of this failure in one unit might mean that the actual probability of the corresponding events in other units is higher than the independent mean value.
(iii) Failures might model unavailability of equipment which is shared by several units. A cooling system might cool two units. A diesel generator might produce auxiliary power for all units. If it is unavailable, it is unavailable for all involved units.
(iv) Mobile equipment can be used only at one unit at a time. A single unit case might expect that this equipment is allocated to this unit. A multi-unit case must calculate with the fact that it will be allocated only to one of the units.
(v) Human reliability analysis (HRA) might reveal dependencies between operator actions in multiple units or dependencies arising from operating procedures or safety culture.

The type of dependencies that can be best captured by modifications in fault trees is Item ii, where Item iii is a special case of Item ii. For other types, a modification of basic event probability for relevant scenarios is sufficient. The dependency treatment for Item ii follows the established Common Cause Failure (CCF) treatment with parametric models. Let us, for simplicity of presentation, adopt the Beta parametric model in this paper. This means that either all failures occur independently or all dependent components fail because of a multi-unit reason. The original failure probability is partitioned among independent events and the multi-unit one.

A complete treatment of the dependencies requires an integrated model of all units. Such a model can be built by the following steps (see, e.g., IAEA (2023)):

- Initiating events must be unified across all units, irrelevant initiators screened out, relevant ones possibly re-calibrated to account for the fact that not all initiating event occurrences lead to a multi-unit accident.
- Events without dependencies must be local to each unit model. We can imagine renaming them with a unique and previously not used prefix UNITXX_, where XX is the number of the unit. Such renaming is not necessary if names in each model are unique. We assume this in the rest of the paper.
- We perform standard CCF replacement of basic events with dependencies by gates and newly created events (so called *Multi-unit events*) as depicted in Figure 1 and Figure 2 for the case of two units. Independent events get the prefix SU_ and we assume that their names are unique (do not occur in other units). Multi-unit events are shared among the units and get a special prefix MU_.
- We create product of all sequences leading to the consequence (so called *Master Event Tree* approach). This is logically equivalent to taking an AND of the master fault tree for individual units (so called *Single Top Fault Tree* approach). If we have a Binary Decision Diagram (BDD) representation of the master fault trees for all units then it is equivalent to the AND operation on these BDDs. Note that the shared events are exactly the initiating events and the multi-unit events.

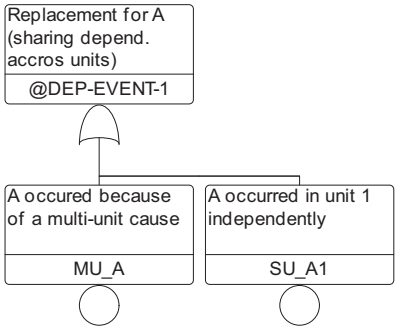

Fig. 1. A fault tree replacing the event $A1$ from Unit 1 which has dependencies with another event in Unit 2.

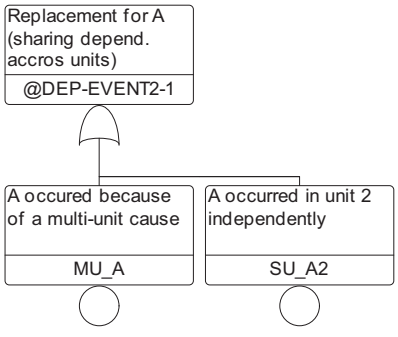

Fig. 2. A fault tree replacing the event $A2$ from Unit 2 which has dependencies with another event in Unit 1.

3.1. *Multi-unit sequence method*

We could now solve the integrated model by standard Boolean methods. Some of the scenarios will include independent failures and some multi-unit failures, affecting several units. For smaller unit models, this could be computationally feasible. For large-scale models such as those from the nuclear industry, this will be in most cases computationally infeasible. Moreover, the exact quantifi-

cation of these scenarios requires treating multi-unit cause and independent failures as mutually exclusive.

The Multi-Unit Sequence Method Holmberg et al. (2019); Bäckström et al. (2024) brings three advantages:

- It separates multi-unit dependencies from single unit models. This makes modeling more structured and makes it possible to obtain risk measures related to multi-unit events.
- It enables approximative treatments of the multi-unit dependencies and quantification on the single-unit level.
- It takes care of the mutually exclusive treatment of multi-unit and independent failures for dependent events.

This method treats occurrences of multi-unit events separately and quantifies sinlge-unit models dependent on which multi-unit events have occurred. Let us visualize multi-unit dependencies in a multi-unit event tree as in Figure 3. Failure in this event tree means that events in both units occurred simultaneously by a multi-unit cause. Success means that these events are independent.

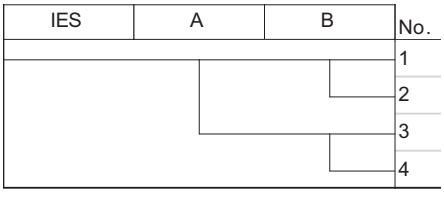

Fig. 3. An example multi-unit event tree with initiators aggregated to a single one and two function events representing dependencies between basic events A and B in both units.

Failure sequences are easy to quantify. Each failure sequence sets the shared multi-unit event to *True*, e.g., MU_A $\leftarrow$ *True*. By this, independent failures are pruned away. The probability of this branch is determined by the dependency factor β as $\beta \cdot P(A)$.

Success sequences need greater care. We know that the multi-unit event has not occurred (i.e., we can set it to *False* as by MU_A $\leftarrow$ *False*). This leaves us still with several combinations:

- SU_A1, SU_A2
- SU_A1, $\neg$ SU_A2
- $\neg$ SU_A1, SU_A2
- $\neg$ SU_A1, $\neg$ SU_A2

For each of the combinations, we have a different assignment of *True, False* to basic events SU_A1 and SU_A2. For each of them we also know how to quantify the probability of this combination based on the original event probabilities, the β-factor and the rules for the complement.

We have now a representation of the multi-unit case. For each of the branching points in the multi-unit event tree, we have a number of combinations to be taken care of. In case of two units and dependencies between pairs of basic events, we have five combinations. For each of them, we must set the truth values of the multi-unit event and the single-unit events and re-evaluate the integrated model. For a multi-unit event tree with three pairs of dependent events, this gives 125 evaluations of the integrated model. For eight pairs of dependent events, we have 390625 evaluations.

The combinatorial explosion can be handled only for small models with a very short analysis time. For large-scale models, we need some approximations.

We can approach the calculation complexity from two directions. We can limit the number of model evaluations needed. There can be a cutoff mechanism that stops exploring branches in the multi-unit event tree when their probability falls below a certain threshold Bäckström et al. (2024). Another option is to group basic event dependencies and treat them as a single, composed, function event in the multi-unit event tree Yamamoto et al. (2024). Yet another approximation does not explore all possible valuations of independent failures and successes in the multi-unit event tree. Instead, it resolves all valuations probabilistically, leaving their state unchanged, but adjusting their probability. We explore this option and the degree of approximation in Section 3.3.

3.2. *Approximative fault tree solutions*

Another approximation direction decreases the time needed for model evaluation under different multi-unit scenarios. Quantification time of a BDD or a Minimal Cut Set (MCS) list is a small fraction of the time needed to build a BDD or generate a MCS list from a master fault tree. Therefore, the first attempt for a multi-unit analysis should be to obtain an exact representation of the combinatorial logic by a BDD or by a MCS list. This will, however, not be possible in many real-life cases where the underlying fault tree model is complex. In this case, an approximate representation of the master fault tree can be used.

A MCS list generated with a cutoff or an approximate BDD, possibly also built from a MCS list generated with a cutoff represent a computationally feasible solution. Ultimately, analysts determine the trade-off between the precision and the computation time needed to generate an approximate result for a single-unit model.

It might be difficult to estimate the error caused by a cutoff-based approximation. Different branches in the multi-unit event tree might set multi-unit events to *True* and by this increase the probability of some scenarios, which in its turn would bring them above the cutoff if we recalculated the MCS list or the BDD from the original fault tree model. Increasing the precision in the cutoff error estimation presents an important topic for future research.

3.3. *Independent Failures*

A precise solution treating independent failures and a multi-unit failure as mutually exclusive explores all combinations of failure and success of independent events. This might lead to a large number of re-quantifications of the single-unit models. We can approximate this treatment by not exploring all combinations of independent failures in the multi-unit event tree. Independent failures are not set to *True* or *False*. Their failure probability is adjusted to reflect the fact that the multi-unit event did not occur. The questions to solve here are:

- How to quantify the success in the multi-unit event tree, and
- How to update the failure probabilities of the basic events with dependencies where the failures are considered independent.

A conservative treatment disregards from the success quantification in the multi-unit event tree. In this case, the probability of the independent failures can be determined by the CCF model. In case of the β-model, the value is $(1 - \beta) \cdot P(A)$, where A is the original event with a dependency. This over-approximates the combination where independent events do not occur (e.g., $\neg$ SU_A1, $\neg$ SU_A2). If the multi-unit failure has a high probability and the fractional contribution of the independent failures is low then this over-approximation can be significant.

Another option quantifies the success branch as the complement to the failure branch. In case of a β-model, the probability of success is $1 - \beta \cdot P(A)$. If we now use the β-model also for the quantification of the independent basic events then we under-estimate the combinations where at least one of the independent failures occurs. A conservative treatment uses a scaling factor for the quantification of independent failures which removes the multi-unit event tree success probability. The value of independent events is then $(1 - \beta) \cdot P(A)/(1 - \beta \cdot P(A))$.

4. Method application

The Multi-Unit Sequence Method is applicable to all models based on event trees and fault trees, irrespective of the domain or type of the system. For small models, we can work with the exact analysis in the Multi-Unit Sequence Method. The preferred option is to encode the complete logic into a BDD and to quantify this BDD. For a moderate number of dependent basic events, it is possible to treat all combinations of independent failures as well. When the numebr of dependent events grows, one can switch to the approximate quantification of independent failures as described in Section 3.3.

For large-scale models, the Multi-Unit Sequence Method requires further approximations. Single-unit models must be solved approximately.

Both options of a MCS list or a BDD can be used. In most of the cases, the cutoff in the multi-unit event tree and the approximate treatment of independent failures will help to decrease the analysis time.

4.1. *Dynamic Models – a Wind Farm*

Wind farms present an example where the failure behavior of the plant requires a more dynamic model than fault trees. Such models are typically analyzed by Monte Carlo simulations. We present a sketch of a model according to the Knowledge Base methodology Bouissou et al. (1991). The model consists of high-level components, their configurations, and their relations. A set of component types available for modeling specific systems forms a knowledge base. Deterministic and stochastic behavior of component types, including possibilities to relate to other components, is described in the modeling language called Figaro. This defines the complete set of behaviors of a specific model.

One of the possible multi-unit dependencies is caused by an external factor – the wind speed. We can define a component type for wind speed modeling over time in the knowledge base. This component type can determine wind speed from a mathematical model, possibly with a stochastic element. Alternatively, it can utilize historical data for a certain area. All wind turbines in the model that are located in the same area should be connected to the component for wind speed in this area. The failure rate of each individual wind turbine will be influenced by the current speed of wind. This type of dependency is fully dynamic and does not have a corresponding equivalent in fault tree models.

Another type of dependency can be caused by incorrect maintenance. A failure because of maintenance of an individual wind turbine can have two causes. Either it is a failure independent of all other maintenance activities. Or, it is a failure common to several wind turbines maintained during the same occasion (e.g., by the same maintenance crew or using the same maintenance material). Maintenance scheme can be again modeled as a separate component type, for example by defining maintenance groups – sets of wind turbines that are maintained together. This component type can also contain stochastic models for underlying maintenance failures. This corresponds to CCF modeling of multi-unit dependencies in the fault tree case.

5. Conclusions

This paper discusses multi-unit scenario analyses in a number of domains that share common elements but differ the type and in the complexity of single-unit models. We show that the Multi-Unit Sequence Method developed in previous work is applicable in all situations where single-unit models consist of event and fault trees. The concepts of multi-unit dependencies and their treatment in a multi-unit event tree can be used in other modeling scenarios as well.

References

Alzbutas, R., M. Vaisnoras, I. Saruniene, R. Krikstolaitis, M. Valincius, E. Babilas, J. Augutis, S. Rimkevicius, T. Iesmantas, F. Anusauskas, and L. Mataitis (2021). Aggregated risk assessment and survey for risk reduction in oil terminals. *Sustainability 21*, 12169.

Bäckström, O., P. Krcal, F. Tanaka, and P. Wang (2024). Multi-Unit PSA Based on Multi-Unit Sequences. In *Proc. of PSAM17&ASRAM2024*.

Bouissou, M., H. Bouhadana, M. Bannelier, and N. Villatte (1991). Knowledge modelling and reliability processing: Presentation of the figaro language and associated tools. In *IFAC Symposium on Safety of Computer Control Systems 1991 (SAFECOMP'91)*, pp. 69–75.

EPRI (2021). *Framework for Assessing Multi-Unit Risk to Support Risk-Informed Decision-Making: General Framework and Application-Specific Refinements*. Palo Alto, CA: 3002020765.

Fuentes-Bargues, J. L., M. C. González-Cruz, C. González-Gaya, and M. P. Baixauli-Pérez (2017). Risk analysis of a fuel storage terminal using HAZOP and FTA. *International journal of environmental research and public health 7*, 705.

Holmberg, J.-E., S. Authén, K. Björkman,

O. Bäckström, X. He, S. Massaiu, and T. Tyrväinen (2019). *Site risk analysis for nuclear installations*. Number 419 in NKS. Roskilde.

IAEA (2019). *Technical Approach to Probabilistic Safety Assessment for Multiple Reactor Units*. Number 96 in Safety Reports Series. Vienna: International Atomic Energy Agency.

IAEA (2023). *Multi-unit Probabilistic Safety Assessment*. Number 110 in Safety Reports Series. Vienna: International Atomic Energy Agency.

Yamamoto, M., K. Takebe, T. Kodama, F. Tanaka, I. Hongo, A. Kawasaki, and M. Takahashi (2024). Development of methods for evaluating the frequency of accidents related to multiple storage tanks and multiple events at the Rokkasho Reprocessing Plant. In *Proc. of PSAM17&ASRAM2024*.

Zhou, T., M. Modarres, and E. L. Droguett (2021). Multi-unit nuclear power plant probabilistic risk assessment: A comprehensive survey. *Reliability Engineering & System Safety 213*, 107782.