Proc. of the European Safety and Reliability Conference (ESREL 2026)
Edited by José C. Matos, Paulo B. Lourenço, Michael Beer, Daniel Oliveira,
Edoardo Patelli, Hélder S. Sousa, Rui A. Silva and Monica Santamaria
 Published by Research Publishing, Singapore.
doi: 10.3850/ESREL2026061419_esrel26-p26328-cd

MBSA Strategies for Handling Dormant and Multiple Dormant Failures

Tony GHUELDRE

IRT Saint Exupéry, France. E-mail: tony.ghueldre@irt-saintexupery.com

Christophe FRAZZA

SATODEV, France E-mail: christophe.frazza@satodev.fr

Wilkinson JOAS

IRT Saint Exupéry/Safran Aircraft Engines, France. E-mail: wilkinson.joas2@safrangroup.com

Julien VIDALIE

IRT Saint Exupéry/Airbus Protect, France E-mail: julien.vidalie@airbus.com

Maintenance allows keeping systems safe, as failed equipment are usually repaired quickly after their detection. However, an important threat to system safety is the presence of dormant failures. These failures, undetected during system operation or not annunciated when they occur, cannot be addressed immediately and are only verified during scheduled maintenance, thus resulting in a longer exposure time compared to standard failures.
Currently, modellers use construction patterns from standards of their work domain within Fault Tree Analysis (FTA) tools to address these failures, and some FTA tools offer automated generation of such patterns. Model-Based Safety Analysis (MBSA) should provide a unified approach to handle them.
This paper proposes several modelling and simulation approaches to represent dormant failures in all RAMS domains in which they are applicable.
- The first approach uses a dynamic modelling pattern that expressively captures the behaviour of a dormant failure. The component is modelled in a form where it undergoes regular maintenance. Simulation is performed using Monte Carlo methods.
- The second approach adapts ARP4761A-proposed modelling patterns for fault trees, enabling dynamic MBSA calculations that consider event sequencing.
- The third approach models components traditionally, with a failure event, and applies probabilistic laws to compute cut-set probabilities while accounting for dormancy. Some tools already implement such laws; we analyse their strengths and limitations and propose a general formulation, including average and maximum exposure times.
Especially, we tailor these approaches to be able to not only tackle single dormant failures, but also double dormant failures, as the latter are more difficult to compute and require a more complex representation.
We apply these methods to toy examples using Cecilia Workshop or SimfiaNeo, and analyse the pros and cons of each approach.

Keywords: MBSA, FTA, Dormant failures, Latent failures, Safety, RAMS.

1. Introduction

Safety-critical systems rely heavily on maintenance strategies to maintain acceptable reliability levels over time. While active failures are typically detected and addressed immediately, a significant threat arises from dormant (or latent) failures. These failures induce an exposed state to the system, for a duration significantly longer than the mission time, governed by maintenance intervals rather than mission duration.
In parallel, with the increasing complexity of modern architectures, safety engineers are shifting towards Model-Based Safety Analysis (MBSA). MBSA offers superior expressiveness by modelling system behaviour and architecture rather than just failure logic. And yet, a unified approach to handling dormant failures within MBSA tools is still a subject of research.
Currently, MBSA engineers often struggle to reconcile the dynamic nature of system simulations with the static probabilistic formulas required by certification standards for dormancy. This paper addresses this gap by proposing and comparing three distinct modelling and simulation strategies to represent dormant failures in an MBSA environment:

1. A stochastic approach using Monte Carlo simulation.
2. A structural approach adapting ARP4761A [9] Fault Tree Analysis (FTA) patterns into MBSA architectures.
3. An analytical approach applying probabilistic post-processing laws to standard failure models.

2. State of the art

2.1 Dormant failure

A dormant failure is a failure that is not evident at the time it occurs and remains undetected until it is revealed by a subsequent failure, a specific operational condition, or a maintenance action.

For example, consider a system with a cold redundancy, where a primary unit is active and a cold standby unit is intended to replace the primary unit if it fails. If the standby unit fails while it is inactive, the system will continue to operate normally using the primary unit, and the failure of the standby unit may remain undetected.

The failure will only be discovered through another failure, or during a periodic test of the standby component. As a result, the system may unknowingly operate without effective redundancy for a given period. In this case, the exposure time to the loss of redundancy is not the operating time of the standby component, but rather the interval between two tests of that component.

ARP4761A [9] specifies the formulas for computing the probability of Failure Conditions (aeronautic equivalent of feared events) represented in a FTA, especially in paragraphs G.11.1.5.2, G.11.1.5.3 (see §4 for these formulas) and G.11.1.5.4. It also defines patterns to model dormant failures and double dormant failures. Some fault tree tools also include dormant probability laws to model dormant failures as standard events rather than having to use the equivalent and more complex pattern.

2.2 Model-Based Safety Analysis

Over the past decades, the complexity of systems has increased as they evolved to integrate more and more functionalities. This growing complexity leads to numerous studies during the system design phase. These studies aim to design the system and examine characteristics specific to engineering domains to demonstrate compliance with requirements, such as Safety ones. To cope with complexity and meet increasingly stringent safety requirements, it is now moving toward so-called "model-based" approaches.

MBSA seeks to create a representative model of the system's physical or functional architecture, enriched with dysfunctional data.

MBSA relies on formal safety-oriented models that explicitly describe the system architecture, including components and their interactions. Failures are modelled at component level, and the overall system behaviour is derived from failure propagation across components. By staying close to the system architecture, MBSA is well suited to handle the complexity of modern systems. Two main MBSA strategies exist [1]:

- extending system development models with safety artifacts, as in approaches using SysML for FMEA (Failure Modes and Effects Analysis) or fault tree generation [2, 3, 4], or tools like TTool for integrating safety and security properties;
- relying on dedicated safety models, which offer higher expressiveness by moving from Boolean to state/event-based formalisms. Dedicated MBSA languages include SAML [5], Figaro [6], and AltaRica [7], the latter based on guarded transition systems (GTS) [8].

In this paper, we restrict MBSA to models using the second strategy, and especially formal AltaRica language. These models automatically generate various safety results, such as failure scenarios leading to feared events (minimal cut sets/sequences) and their associated probabilities.

3. Use cases

For second-order scenarios involving two dormant failures, only the first may remain dormant. The second failure must occur during the mission to have any impact (to reach the "FE" Feared Event)—if it occurs earlier, its effect will be realized before the mission begins.

More generally, in a N-order scenario with N dormant failures, the first N – 1 are dormant, but the final failure must be non-dormant (i.e. occurs during the mission).

Let us consider the following RAMS example, with two failure events, "A" and "B," that together lead to a FE. Both A and B are dormant failures with:

- Dormancy time of "A" = t_A = 500 hours
- Dormancy time of "B" = t_B = 1000 hours

- Mission time = t_f = 2 hours
- λ_A is the Failure Rate (FR) of A
- λ_B is the Failure Rate of B.

If both A and B occur before the mission starts, the FE will happen pre-mission. Therefore, there are three distinct ways to reach the FE during the mission:

(i) The first scenario: A occurs before the mission (dormant), then B occurs during the mission (non-dormant).

(ii) The second scenario: B occurs before the mission (dormant), then A occurs during the mission (non-dormant).

(iii) The last scenario: both A and B occur during the mission (non-dormant)
This is least likely, since mission duration is typically much shorter than either dormancy time.

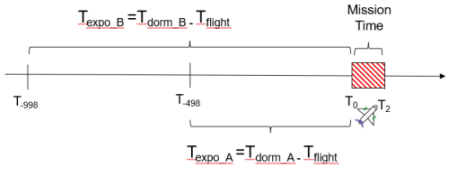

Fig. 1. Illustration in an aeronautical use case

It should be noted that the use cases were intentionally designed to be as generic as possible so that they could be replicated across different RAMS application domains. They were deliberately kept simple in order to focus on comparing the classical FTA method with the modelling capabilities of the three MBSA approaches. Consequently, the model parameters were minimized to highlight methodological differences rather than application-specific complexities.
As illustrated previously, managing dormant failures demands meticulous attention to achieve valid, usable results. The following chapters present three approaches for modelling dormant-failure behaviour within an MBSA model.

4. Stochastic approach

A stochastic simulation is defined in [10] as a simulation of a system that has variables that can change stochastically (randomly) with individual probabilities.
The purpose of the stochastic simulation is to obtain results that are not based on any formula, however the purpose here is to obtain the same results than the ARP4761A [9] formulas mentioned in §2.1 would obtain.
The two use cases for this approach are based on the three scenarios defined in §3. Use case 1 represents a specific scenario of (ii): event A is active and event B is latent. In use case 2, the three scenarios are covered by the fact that both events are dormant.
The following paragraphs define the approach, the results are given in paragraph 6.1.

4.1 Use case 1

For use case #1, two approaches were identified to obtain the expected results. For both, the MBSA model is made of 2 blocks to represent A and B with internal events based on the exponential law, and an AND2-gate block to compute the FE.

4.1.1. Standard model

For the first approach, the ARP4761A [9] formula of § G.11.1.5.2 is applied to the model:

$$P_{WC} = \lambda_A * t_f * \lambda_B * t_B \qquad (1)$$

With:

- λ_A = 1e-06 h^{-1}
- λ_B= 1e-07 h^{-1}
- t_f = 2h = t_A, A being active
- t_B= 1000h

For a stochastic simulation in a MBSA tool, it is not possible to take a mission duration (here flight duration t_f) lower than a dormancy duration. As the events' probability of occurrence is based on mission time, we need to adapt the formula $P \approx \lambda t$ using the following modified Failure Rate (FR):

$$\lambda'_A = \lambda_A * \frac{t_A}{t_f} = \lambda_A = 1e-06\ h^{-1} \qquad (2)$$

$$\lambda'_B = \lambda_B * \frac{t_B}{t_f} = \lambda_B * \frac{1000}{2} = 5e-05\ h^{-1} \qquad (3)$$

As the failure rates are very low, the number of stories needed to perform an accurate stochastic simulation is very high. To bypass this, a FR multiplier k = 100 has been used for both events. This increases the probability of occurrence, hence the number of appearances of a FE during the simulation. To translate the obtained stochastic result into the expected one, the stochastic result has to be divided by k^2.

4.1.2. Model with Dirac

The second approach introduces a Dirac-based model for event B. Its failure scenario has been represented by two events:

- The first one is based on an exponential law, and allows it to pass from its nominal state to a latent state. The failure rate for this law is the same λ'_B used previously.
- The second event is a Dirac one, taken to represent its latency. At the end of the Dirac duration, event B's state goes from latent to failed. The Dirac duration has been taken as equal to t_f/t_B = 2e-03 h, to represent one dormancy for each simulation.

The remaining of the model (failure rates, duration, multiplicative factor) is identical to the first approach.

4.2 Use case 2

For use case #2, two approaches were identified to obtain the expected results.

4.2.1. Failure rate conversion

The first approach is based on the conversion of failure rates, to bypass the limitation of the phase duration. It is based on the ARP4761A [9] formula of § G.11.1.5.3:

$$P_{Ave} = \frac{1}{2} * \lambda_A * \lambda_B * t_f * (t_A + t_B) \qquad (4)$$

With:

- λ_A = 1e-06 h^{-1}
- λ_B= 1e-07 h^{-1}
- t_f = 2h
- t_A = 500h
- t_B= 1000h

Based on this formula, events A and B failure rates were modified to maintain the same FE probability of occurrence. The new failure rates are:

$$\lambda'_A = \frac{1}{2} * \lambda_A * \lambda_B * t_A \qquad (5)$$

$$\lambda'_B = \frac{1}{2} * \lambda_A * \lambda_B * t_B \qquad (6)$$

Finally, the AND2 gate of the model was replaced by an OR2 gate. This way, the FE is calculated by:

$$P_{FC} = \lambda'_A * t_f + \lambda'_B * t_f - \lambda'_A * t_f * \lambda'_B * t_f \qquad (7)$$

(note: the last term is estimated negligible: 10 orders of magnitude below the other terms)

Replacing the new failure rates by their developed forms we obtain:

$$P_{FE} = \frac{1}{2} * \lambda_A * \lambda_B * t_A * t_f +$$

$$\frac{1}{2} * \lambda_A * \lambda_B * t_B * t_f = P_{Ave} \qquad (8)$$

As the failure rates were very low, once again a FR multiplier k has been used. The following data were put in the model:

- $\lambda'_A = 2.5e-11$ h^{-1}
- $\lambda'_B = 5e-11$ h^{-1}
- $t_f = 2h$
- k = 10.000
- The model is still based on 2 events A and B, but the AND2-gate is now an OR2 gate, so the final result has only to be divided by k.

4.2.2. ARP-based model

The second method is based on the ARP4761A [9] model from Figure G24, which is used to represent how two latent failures' combinations lead to the FE. It consists of the gathering, under an OR3-gate, of three AND2-gate branches, each one representing a use case of §3. The MBSA model has been adapted to match the results obtained with the ARP4761A [9] methodology:

- Events A and B are now more complex blocks. Each block's state takes the values {OK, Active, Dormant}, to determine if they are respectively working, failed active or failed dormant. Each block has 2 different events, "Active" and "Dormant", each one leading to the corresponding failed state. The output of each block is their state's value.
- The dormant failure rate has been adapted to the mission duration of 2 hours: $\lambda'_A = \lambda_A * \frac{t_A}{t_f}$, same for event B.
- A FR multiplier k = 100 has been used for both events.
- A final block performs the 3 AND2-gates and the final OR3 gate of ARP4761A [9] Figure G24 model. If one of the block's states is failed dormant and the other block's state is failed active, or if the two block's states are failed active, the FE is reached.

With this approach, it is possible to differentiate the worst case probability and the average probability, by applying the average probability formula events ($P = \lambda T/2$) for each dormant failure.

5. ARP4761A approach

5.1 Generalities

In the aeronautical context, the quantitative objective is related to the probabilistic risk for an average flight ; risk being defined in ARP4761A [9] as "The potential of an occurrence to cause harm defined by its probability and the severity of its consequence(s)". *Mission time* corresponds to the duration of an average flight.

During the flight, except in rare cases, components are not repairable thus their failure probability is given by an exponential law with a constant failure rate. As the maintenance is performed on ground, the repair time of a component can be considered as null.

If the time reference is the life of the aircraft, components can be considered as periodically tested. The *Inspection periodicity* corresponds to the time between two maintenance tasks.

As a result, between two maintenance tasks, these failures can be considered as *latent* (or dormant, or hidden), as long as the combination with other failures does not lead to a failure condition, i.e., a detectable event. In this case, we can talk about a *significant latent failure*.

5.2 Calculation principle

5.2.1. Concept of maximum and average risk

Let's consider a system made up of two components, C_1 and C_2, with their respective failure rates λ_1 and λ_2 and their inspection time T_1 and T_2, with $T_2 = 4\ T_1$.

Pr_1 and Pr_2 are the probability of failure of the components. Pr_S is the instantaneous probability of failure of the system. Its maximum is reached at t = T_2. Pr_S avg is the instantaneous average probability of failure of the system.

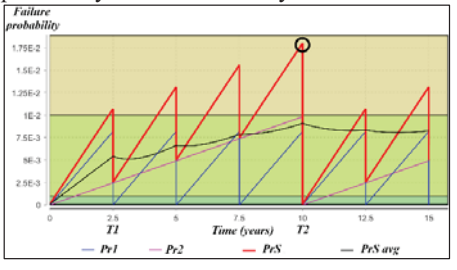

Fig. 2: Maximum and average risk concept

The maximum risk approach may be very detrimental in case of latent failures, that's why ARP4761A [9] authorises, in specific conditions, to use the average risk approach.

5.2.2. Calculation of average risk for two latent failures

Mathematically, the global average risk can be calculated using the integral of Pr_S between 0 and T_2. In the case of the previous example, the resulting equation is as follows:

$$Q_{avg} = \frac{1}{T_2}\left[\begin{array}{l}\int_0^{T_1}\left((1-exp^{-\lambda_1.t}).(1-exp^{-\lambda_2.(0.T_1+t)})\right).dt \\ +\int_0^{T_1}\left((1-exp^{-\lambda_1.t}).(1-exp^{-\lambda_2.(1.T_1+t)})\right).dt \\ +\int_0^{T_1}\left((1-exp^{-\lambda_1.t}).(1-exp^{-\lambda_2.(2.T_1+t)})\right).dt \\ +\int_0^{T_1}\left((1-exp^{-\lambda_1.t}).(1-exp^{-\lambda_2.(3.T_1+t)})\right).dt\end{array}\right] \quad (9)$$

With the assumptions $\lambda_1.T_1 << 1$ and $\lambda_2.T_2 << 1$, this equation can be simplified as follows:

$$Q_{avg} = \frac{1}{T_2}\left[\begin{array}{l}\int_0^{T_1}\left(\lambda_1.t.\lambda_2.(0.T_1+t)\right).dt \\ +\int_0^{T_1}\left(\lambda_1.t.\lambda_2.(1.T_1+t)\right).dt \\ +\int_0^{T_1}\left(\lambda_1.t.\lambda_2.(2.T_1+t)\right).dt \\ +\int_0^{T_1}\left(\lambda_1.t.\lambda_2.(3.T_1+t)\right).dt\end{array}\right] \quad (10)$$

It can be generalized for n, where $T_2 = n\ T_1$:

$$Q_{avg} = \frac{1}{T_2}\sum_{p=1}^{n}\left[\int_0^{T_1}\left(\lambda_1.t.\lambda_2.((p-1).T_1+t)\right).dt\right] \quad (11)$$

Using the sum of n consecutive terms of an arithmetic progression:

$$Q_{avg} = \frac{\lambda_1.\lambda_2}{2.T_1}\left[\int_0^{T_1} t^2.dt + \int_0^{T_1}(t^2.dt + (T_2 - T_1).t.dt)\right] \quad (12)$$

Thus finally:

$$Q_{avg} = \frac{\lambda_1.\lambda_2.T_1.(3.T_2 + T_1)}{12} \quad (13)$$

5.3 Application in MBSA tools

5.3.1. Second approach

This approach consists of explicitly adapting the construction patterns proposed by ARP4761A [9] within the MBSA environment. This method forces the model structure to strictly represent the disjoint scenarios leading to the Failure Condition (FC) in a double dormant failure context.
The ARP4761A [9] standard acknowledges that for a double dormant failure to lead to a loss of function during a flight, specific sequencing is required. Since the system is checked before the flight, at least one of the failures must occur during the flight (active failure), or else the loss of function would have been detected on the ground (pre-dispatch).
To represent this in MBSA, we apply a "splitting" strategy. Each dormant failure involved in the FC is modelled not as a single failure event, but is split into two distinct, mutually exclusive failure events:

1- Dormant Failure: The component fails during the inspection interval prior to the flight (T_{dorm} - T_{flight}).
2- Mission Failure: The component fails during the flight mission (T_{flight}).

5.3.2. Third approach

This approach consists of applying probability laws representing dormancy in a MBSA model consisting of 2 events, A and B. Those events will be dormant or active, depending on the use case defined in §3.

6. Results with application in tools

6.1 Stochastic results

The stochastic calculation of the MBSA models from §4 has been performed using SimfiaNeo. The following results were all obtained using 100 million stories:

Table 1. Results obtained with the Stochastic method

Use Case	Method	Expected result	Obtained result
2	Standard	2.00E-10	1.89E-10
2	With Dirac	2.00E-10	2.12E-10
3	FR conversion (*)	1.50E-10	1.75E-10
3	ARP-based, Worst Case	3.00E-10	2.82E-10
3	ARP-based, Average	1.50E-10	1.71E-10

(*) Complementary note: to target the $P_{average}$ probability using the FTA formula for latent events ($P = \lambda T/2$) only resulted in a FE result too low: 5.5E-11 with 100M stories. This approach shall not be applied, the results obtained being inconsistent.

6.2 Second approach results

6.2.1. Description of the application

We implemented the previously defined pattern (see §5.3.1) using the SimfiaNeo tool. As illustrated in Fig. 3, the model architecture explicitly reconstructs the three logical branches defined in the ARP4761 [9] Figure G24 fault tree pattern.
The model involves two components, A and B. The structure is composed of four leaf-events representing the split failure modes. They are contained in four blocks: A_Dormant, A_Mission, B_Dormant, and B_Mission. These events are fed into three sub-systems (AND gates) representing the three scenarios defined in §3:

1- Scenario (i) (System): This corresponds to the connection of A_Dormant and B_Mission in the model.
2- Scenario (ii) (System2): This corresponds to the connection of B_Dormant and A_Mission.
3- Scenario (iii) (System1): This corresponds to the connection of A_Mission and B_Mission..

A top-level observer block, ER_ARP_FTA (equivalent to an OR gate), collects these three scenarios to compute the global probability of the Failure Condition.

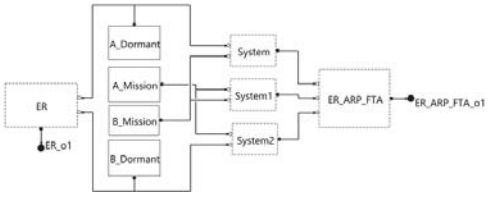

Fig. 3: MBSA implementation of the ARP4761A [9] double dormant pattern in SimfiaNeo

To ensure the calculation matches the analytical results of the ARP4761A [9], the quantification of the split events must be adjusted.
In SimfiaNeo, we used the CMT (Constant Mission Time) law or adjusted exponential laws to assign the correct exposure times to each event:

- For Mission events (A_Mission, B_Mission), the exposure time is strictly the flight duration (t_f).

- For Dormant events (A_Dormant, B_Dormant), the exposure time represents the average latency period. According to ARP4761A [9] guidelines, this is typically derived from the inspection interval (T_n) minus the flight time, or often simplified to the inspection interval if $t_f << T_n$.

By coding this combinatorial logic directly into the MBSA tool, we bypass the need for complex stochastic convergence for these specific latent sequences. The tool calculates the probability of the union of these cut-sets.

6.2.2. MBSA results with SimfiaNeo

The model built in SimfiaNeo, as presented in §5.3, offers the following results:

Table 2. Results obtained with SimfiaNeo

Use Case	Method	Expected result	Obtained result
1	Minimal Cutsets Classic probability CMT Law	1.00E-10	1.0017E-10
2	Minimal Cutsets Classic probability CMT Law	2.00E-10	2.002E-10
3	ARP-Based worst case	3.00E-10	3.0036E-10

Note that the CMT Law in SimfiaNeo should not be used for multiple dormancies but rather the ARP-based pattern.

6.3 Third approach results

6.3.1. Description of the application

In Cecilia Workshop, the complete equation (without any simplification) is used and calculated using a multiphase Markov graph.

To use Cecilia Workshop properly with latent failures, once the events parameters have been duly completed, a *Nominal computation* shall be launched. The computation type shall be *Average Risk Approach*.

In order to deal with *significant latent failures,* a *Dormant* function has been implemented.

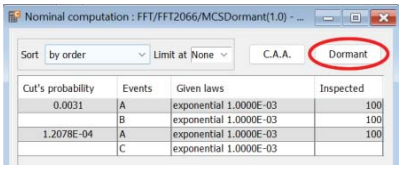

Fig. 4: *Nominal computation* results

In case of cuts including only latent failures, this function will post-process the results and divide these cuts: one event with a significant latent failure (noted with a star: *), the other events remaining with latent failures.

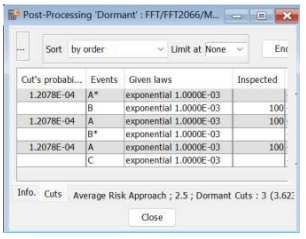

Fig. 5: *Nominal computation* results after Dormant post-processing

In our case, the global probability has been significantly reduced.

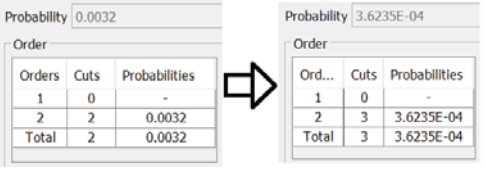

Fig. 6: *Probabilities* after Dormant post-processing

Caution: a *dormant* law exists in Cecilia Workshop, where λ is the failure rate, MTTR the mean repair time and T the inspection time.

This law was implemented for interoperability purposes with *Isograph FaultTree+* and should not be used otherwise.

6.3.2. MBSA results with Cecilia Workshop

The MBSA model for Cecilia Workshop was a combination of two events, following the use cases definition of §3. The results obtained are the following:

Table 3. Results obtained with Cecilia Workshop

Use Case	Method	Expected result	Obtained result
1	Sum of minimal cuts	1.00E-10	9.998E-11
2	Sum of minimal cuts	2.00E-10	2.00E-10
3	Sum of minimal cuts + 'Dormant'	1.50E-10	1.50E-10
3	Average risk + 'Dormant'	3.00E-10	2.999E-10

Conclusion

Handling dormant failures in MBSA is not merely a modelling challenge but a necessity for complying with certification objectives. In this paper, we explored three strategies to tackle the complexity of single and double dormant failures.

The stochastic approach, while conceptually the most "true-to-life" regarding system dynamics, proves computationally expensive. As demonstrated, achieving convergence for rare events like double latency requires significant simulation time, or artificial acceleration techniques such as failure rate multipliers, which can introduce bias if not carefully managed. It should also be noted that, due to their probabilistic nature, results of this approach would require systematic scrutiny to be trusted.

The structural approach applied in SimfiaNeo model (ARP Pattern adaptation) offers the benefit of exact analytical traceability. By explicitly modelling the "Prior-to-flight" and "During-flight" scenarios, it bridges the gap between FTA and MBSA. However, this comes at the cost of model complexity. It forces the modeller to manually "hard-code" failure sequences into the architecture, arguably negating the primary benefit of MBSA: automatically generating failure logic from a system description.

Finally, the analytical approach applied in Cecilia Workshop (Probabilistic laws), appears to be the most pragmatic for industrial scale. By keeping the model simple and handling the dormancy mathematics (Worst case and Average Risk approaches) during the quantification phase, it preserves the readability of the MBSA model while ensuring accurate results. However, this relies heavily on the tool's specific capabilities to implement complex formulas for higher-order cut-sets.

Acknowledgement

This work was conducted within the CoSMoS (Collaborative Safety (&RAMT) Modelling Studies) from IRT Saint Exupéry and would not have been possible without its funding. Therefore, the authors would like to thank the project and its partners.

References

[1] Oleg Lisagor, Tim Kelly, and Ru Niu. Model-based safety assessment: Review of the discipline and its challenges. ICRMS'2011 - Safety First, Reliability Primary: Proceedings of 2011 9th International Conference on Reliability, Maintainability and Safety, pages 625–632, 2011.

[2] Kester Clegg, David Stamp, and John McDermid. Binding Fault Logic to System Design: A SysML Approach. (July):880–887, 2021

[3] Myron Hecht and David Baum. Use of SysML for the creation of FMEAs for Reliability, Safety, and Cybersecurity for Critical Infrastructure. INCOSE International Symposium, 29(1):145–158, 2019.

[4] Pierre De Saqui-sannes, Ludovic Apvrille, Rob Vingerhoeds, Checking SysML Models Against Safety and Security Properties ; HAL Id : hal-03423073. 2021.

[5] Matthias Güdemann and Frank Ortmeier. A framework for qualitative and quantitative formal model-based safety analysis. Proceedings of IEEE International Symposium on High Assurance Systems Engineering, pages 132–141, 2010.

[6] M. Bouissou, H. Bouhadana, M. Bannelier, and N. Villatte. Knowledge Modelling and Reliability Processing: Presentation of the Figaro Language and Associated Tools. IFAC Proceedings Volumes, 24(13):69–75, 1991.

[7] Michel Batteux, Tatiana Prosvirnova, and Antoine Rauzy. AltaRica 3.0 Language Specification. page 126, 2017.

[8] Antoine Rauzy. Guarded transition systems: A new states/events formalism for reliability studies. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 222(4):495–505, 2008.

[9] SAE International. Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment ARP4761A. Technical report, SAE International, 2023.

[10] DLOUHÝ, M.; FÁBRY, J.; KUNCOVÁ, M.. Simulace pro ekonomy. Praha : VŠE, 2005.