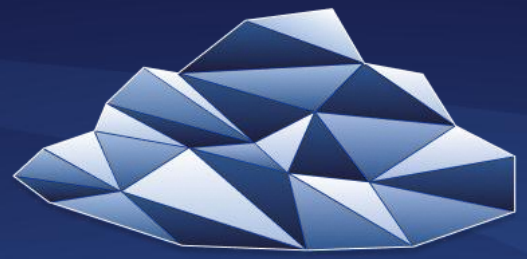


fractus

Reliability Block Diagrams with AltaRica 3.0: A step-by-step introduction

Clarifiez vos systèmes

Sécurisez vos décisions, Accélérez vos transformations

4ieme Webinaire de l'Association AltaRica – 3 Octobre 2025

Agenda

1. Context & Objectives [5 min]
2. Reliability Block Diagrams (RBDs) Basics [5 min]
3. Introduction to AltaRica 3.0 [5 min]
4. Demo - Modeling in AltaRica 3.0 Workshop [10 min]
5. Results & Analysis [5 min]
6. Perspectives & Best Practices [5 min]
7. Ressources
8. Q&A [20 min]

1. Context & Objectives

- Increasing need for **system reliability assessment** in critical industries
  - Increasing demand
  - Increasing complexity

- **Model-Based Safety Assessment (MBSA):** structured and automated approach
  - Become a standard approach to lead RAMS analysis
  - Formal approach to lead quantitative analysis

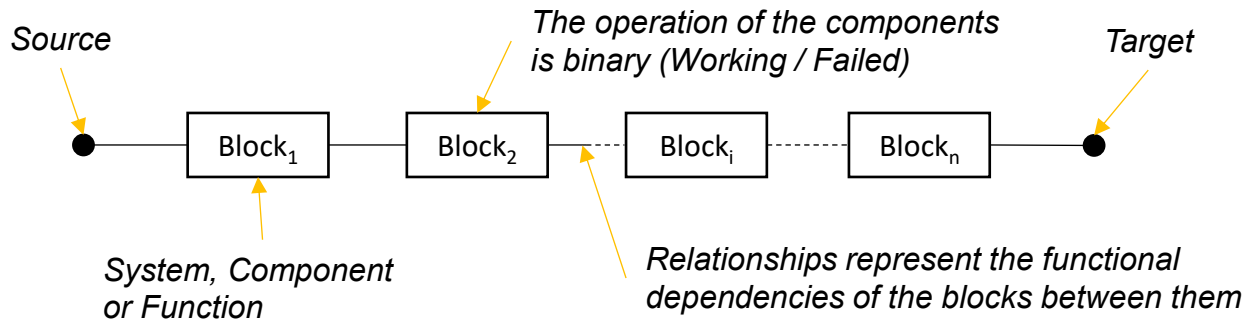
- AltaRica 3.0: dedicated language for **failure modeling & propagation**
  - The third version of the AltaRica language

- Goal of this tutorial:
  - Show how to model a simple RBD in AltaRica 3.0
  - Derive reliability results automatically

(Diagram: System → RBD model → AltaRica model → Reliability analysis tools)

2. Reliability Block Diagrams (RBDs) Basics Intuitive view

A reliability block diagram describes thus a network of connected blocks. A flow circulates in the network starting from a source node, propagating through blocks and ending in a, usually unique, target node.



- Main assumptions of RBDs
  - Independence of block failures
  - Non-repairable system
  - Non reconfigurable
  - No complex logic (priorities, time sequences).

- The system described by the network is working if there is at least one operating path from one of the source nodes to the target node.

2. Reliability Block Diagrams (RBDs) Basics Intuitive view

- Goal :
  - Assess system reliability from reliability of its components
  - List of success paths: All combinations that lead to the success of the mission/function.

- Here a simple example:

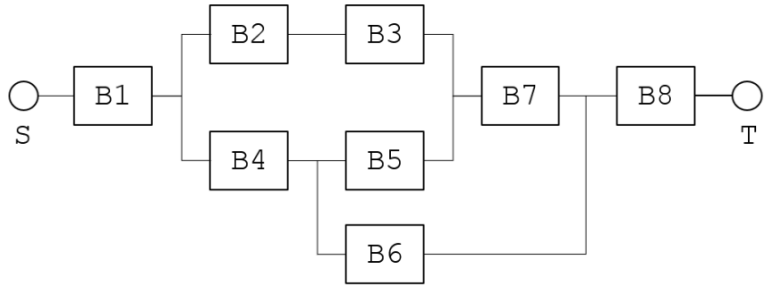


Figure 1: A reliability block diagram

- Possible S-T paths are
  - S -> B1 -> B2 -> B3 -> B7 -> B8 -> T,
  - S -> B1 -> B4 -> B5 -> B7 -> B8 -> T
  - S -> B1 -> B4 -> B6 -> B8 -> T

2. Reliability Block Diagrams (RBDs) Basics Formal view

Block states

- Each block has two states: **WORKING** or **FAILED**
- System success depends on connectivity from input to output

Failure modeling

- Failures are modeled as **independent events**
- Typical assumption: **exponential distribution** (constant failure rate)

Flow logic

- System behavior expressed as **Boolean equations**
- Example: B8.in = B6.out OR B7.out

Assumptions & limitations

- No **repairs** (time-to-failure only, not availability)
- No **dynamic reconfiguration** during mission
- No **feedback loops** (purely combinatorial structure)

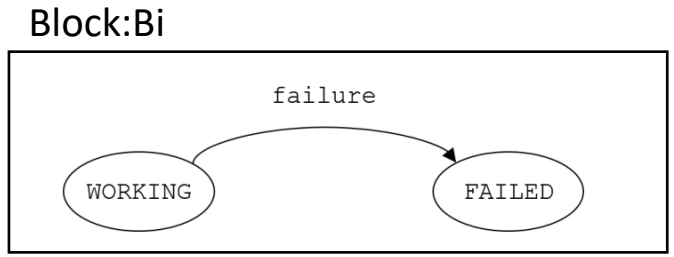


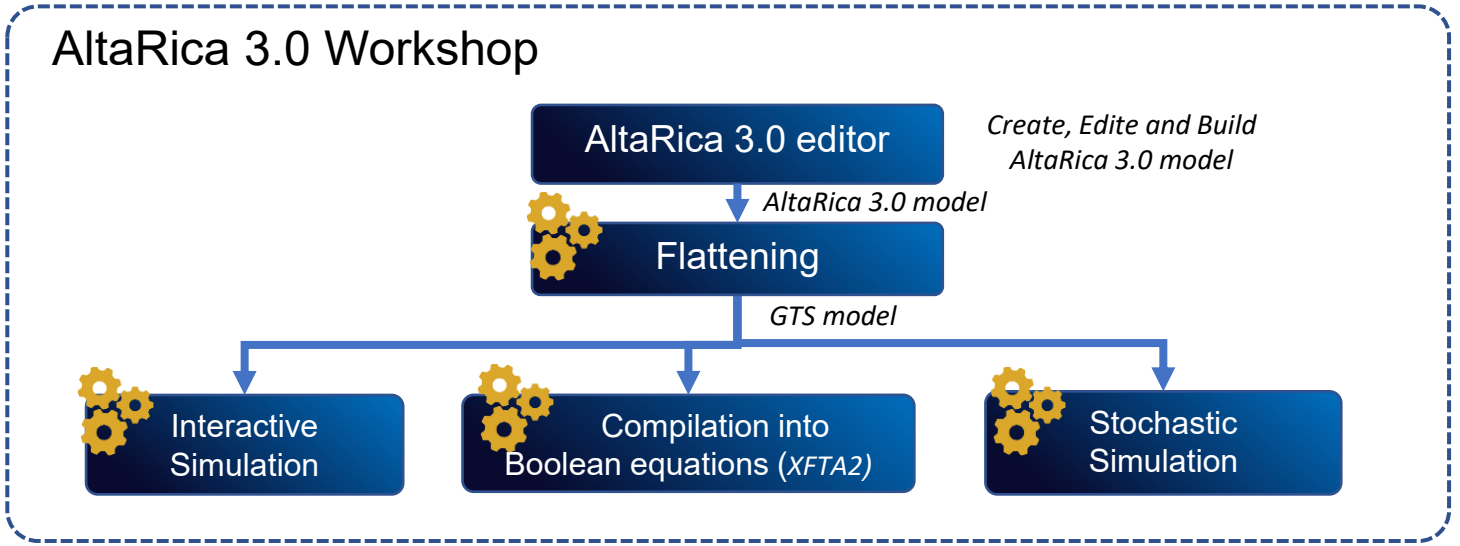
Figure 2: The finite state automaton representing a basic block

3. Introduction to AltaRica 3.0
AltaRica 3.0 in a nutshell

Technical solution:

AltaRica 3.0 Model = Guarded Transition System (GTS) + System Structure Modeling Language (S2ML)

- A computer language dedicated to modeling and calculations in the field of **risk analysis (safety, reliability, performance) of complex systems.**
- Widely used in many sectors: **defense, aeronautics, space, oil, energy, naval, railway, etc.**



3. Introduction to AltaRica 3.0
AltaRica 3.0 in a nutshell

What is AltaRica 3.0?

- A **domain-specific language** for modeling safety and dependability of systems
- Based on the concept of **states, transitions, and flows**
- Provides a **formal yet readable syntax** close to engineers' reasoning

Key features

- **Component-based & modular**: build complex systems from simple blocks and classes
- **Multi-level abstraction**: from high-level architectures to detailed behaviors
- **Mathematically rigorous**: suitable for formal analysis and certification
- **Tool-supported**: Interactive simulation, Fault Trees, stochastic analysis

Use cases

- Safety assessment of critical systems (aerospace, nuclear, transport, defense)
- Early validation of design choices
- Bridging simple models (RBDs) with advanced analysis (Markov chains, FTA, etc.)

3. Introduction to AltaRica 3.0
Example Code: BasicBlock

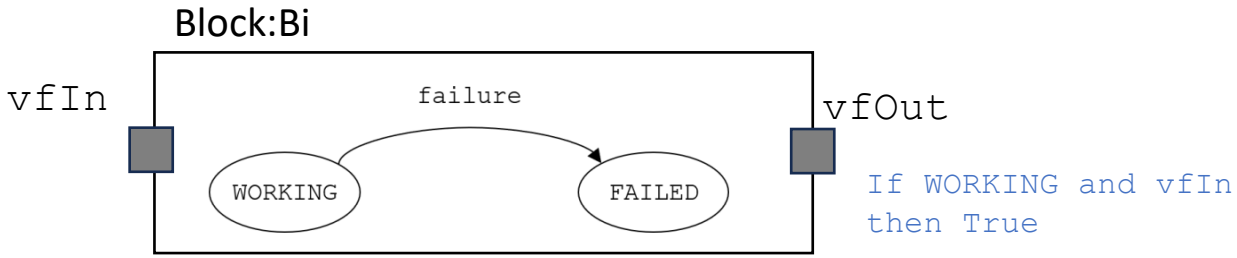


Figure 3: The basic block specification

```
class BasicBlock
  Boolean vsState (init = true);
  Boolean vfIn, vfOut (reset = false);
  parameter Real pLambda = 1.0e-3;
  event evFailure (delay = exponential(pLambda));
transition
    evFailure: vsState -> vsState := false;
assertion
    vfOut := if vsState then vfIn else false;
end
```

Declaration of the component (a 'class')

State variables

Flow variables

Parameters

Events

Internal elements

Internal
to the component

External
Relations with other components

Behavior

3. Introduction to AltaRica 3.0
Building a Full RBD

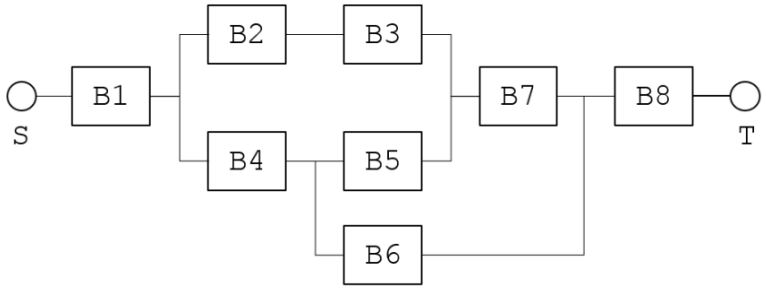


Figure 1: A reliability block diagram

- Instantiate several blocks (B1 … B8).
- Connect them with equations (B8.in := B6.out or B7.out).
- Add an *observer* to monitor system failure:
  - `observer Boolean failed = not T;`

```
block Plant
    Boolean S (reset = false);
    Boolean T (reset = false);

    BasicBlock B1(pLambda = 1.0e-6);
    BasicBlock B2(pLambda = 1.0e-4);
    BasicBlock B3(pLambda = 1.0e-4);
    BasicBlock B4(pLambda = 1.0e-4);
    BasicBlock B5(pLambda = 1.0e-4);
    BasicBlock B6(pLambda = 1.0e-5);
    BasicBlock B7(pLambda = 1.0e-6);
    BasicBlock B8(pLambda = 1.0e-6);

    observer Boolean failed = not T;

    assertion
        T := B8.vfOut;
        B8.vfIn := B6.vfOut or B7.vfOut;
        B7.vfIn := B3.vfOut or B5.vfOut;
        B6.vfIn := B4.vfOut;
        B5.vfIn := B4.vfOut;
        B4.vfIn := B1.vfOut;
        B3.vfIn := B2.vfOut;
        B2.vfIn := B1.vfOut;
        B1.vfIn := S;
        S := true;
end
```

Declaration of the component (a 'block')

Flow variables

Class instantiation
and parameter overloading

Obervers
indicator to be assessed

Internal elements

External
Relations with other components

Behavior

4. Step-by-step Modeling in AltaRica Workshop Live Demo (Steps)

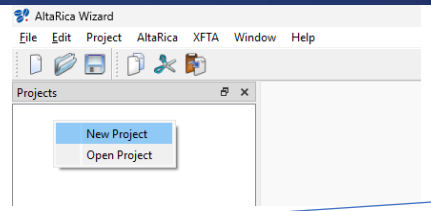
1. Create a **new project** in **AltaRica 3.0 Workshop**
   1. Right click in « Projects view », then « new Project »
   2. Define the Project Name et directory then Click on « OK »

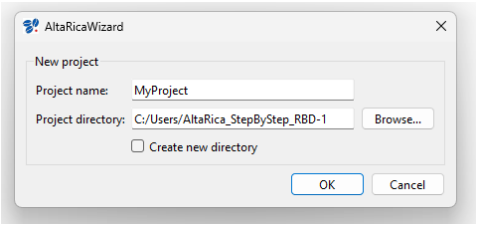
2. Create new **AltaRica model**
   1. Right click in our project in « Projects view », then « Add new file to « MyProject »
   2. Define name the name of your AltaRica model and finish by « .alt »

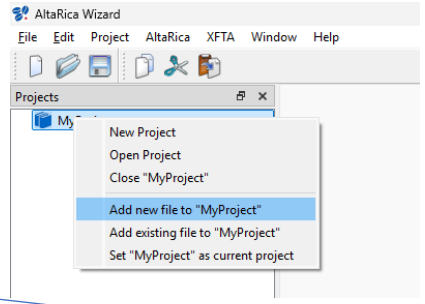
3. Create the **BasicBlock class** in .
   1. In editor view, complete the your AltaRica model to define a BasicBlock class.

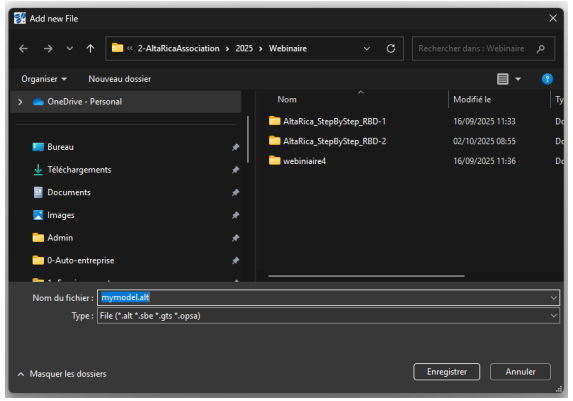
4. **Instantiate** and **connect blocks**.
   1. In editor view, complete the your AltaRica model to define a RBD block.

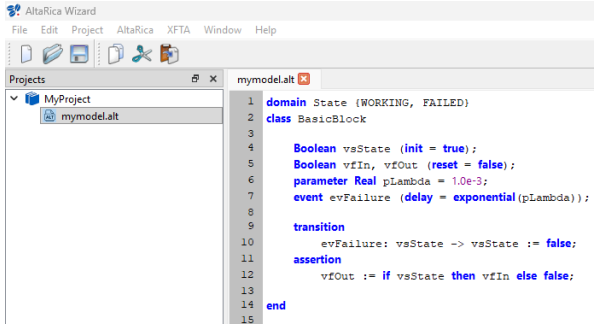
5. Add an **observer**.





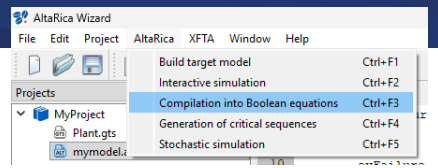


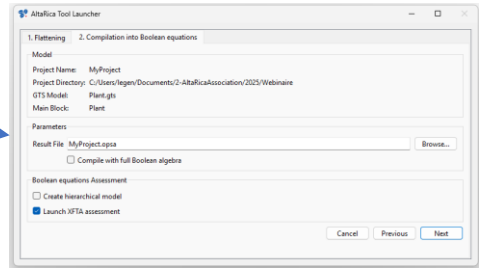


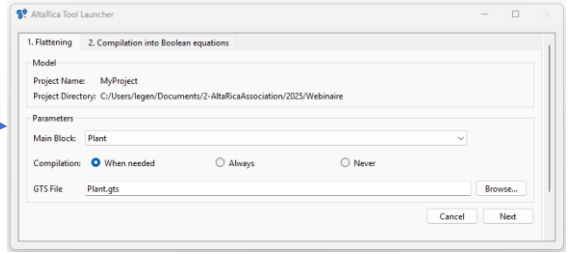


4. Step-by-step Modeling in AltaRica Workshop Example Results

1. Compile into Boolean equations (FTA).
   1. Then click on Next (Flattening)
   2. Name the FTA file ending by « .opsa »
   3. Click on Launch XFTA assessment

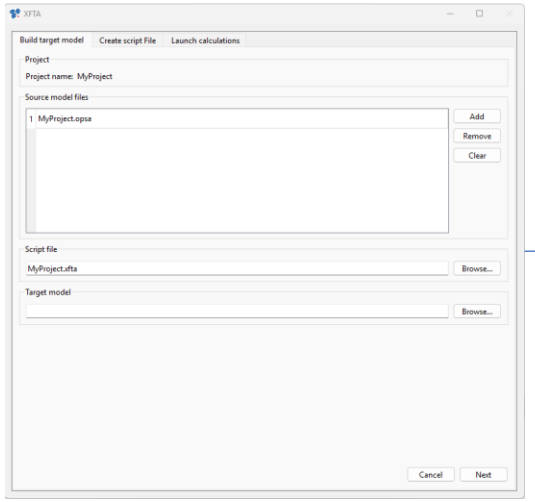


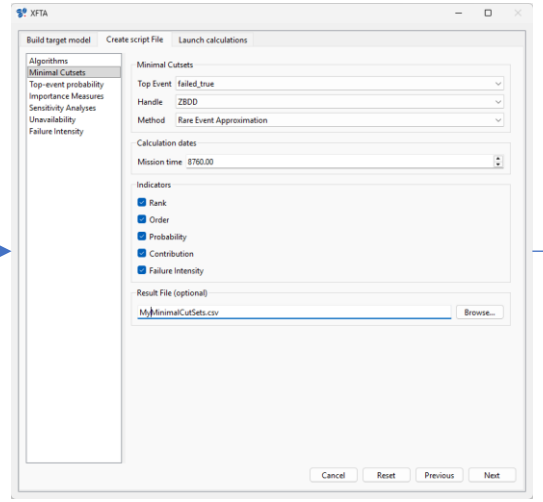


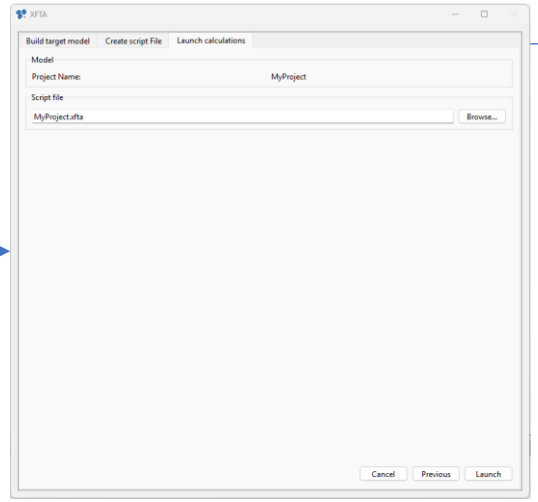


Run analysis to get Minimal Cutsets

1. In « **BuildTargetModel** » click, **next**,
2. Select the « **Minimal Cutsets** » view et select **Top Event** and **Indicators** you want, Complete **Mission Time** and **Result File(csv)**.
3. Then click **Next** then **Launch** to Generated Result File.







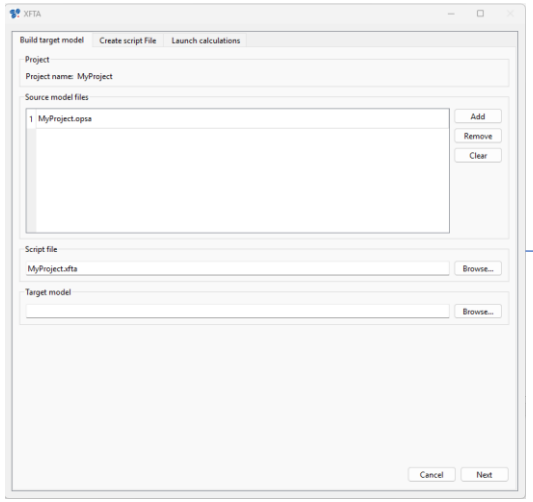
ResultFile.csv

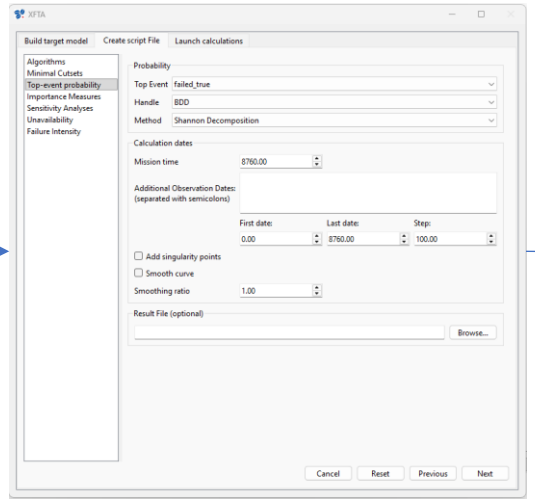
minimal-cutsets		
B1.evFailure		
B4.evFailure	B2.evFailure	
B4.evFailure	B3.evFailure	
B4.evFailure	B7.evFailure	
B6.evFailure	B2.evFailure	B5.evFailure
B6.evFailure	B3.evFailure	B5.evFailure
B6.evFailure	B7.evFailure	
B8.evFailure		

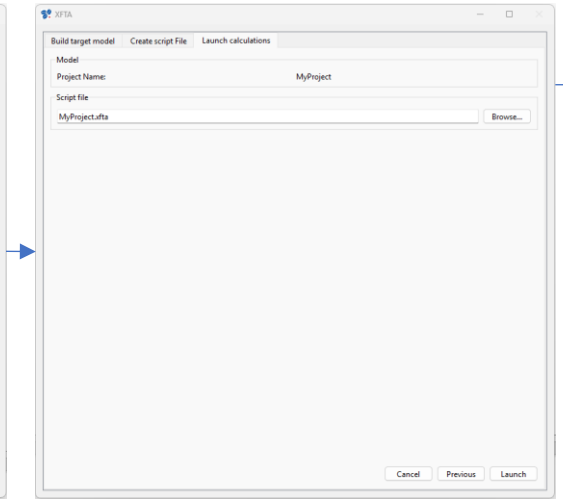
4. Step-by-step Modeling in AltaRica Workshop Example Results

Run analysis

1. In « **BuildTargetModel** » click, next,
2. Select the « **Top-event probability** » view et select **Top Event**, Complete **Mission Time**, **Last Date** and **Step** and **Result File(csv)**.
3. Then click **Next** then **Launch** to Generated Result File.

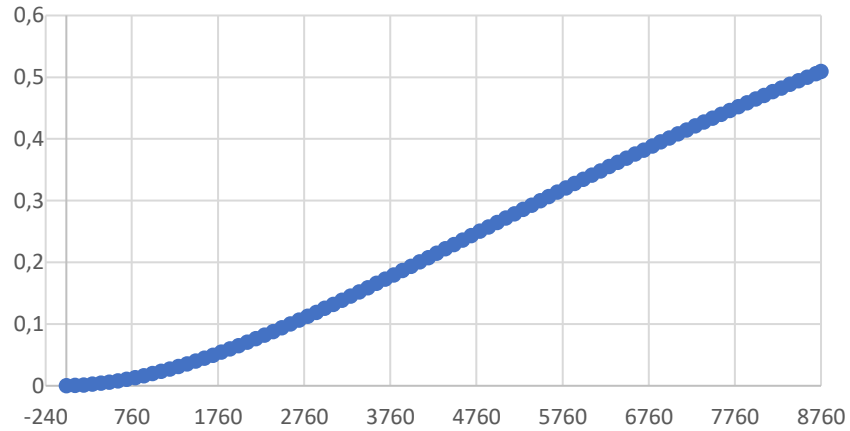






ResultFile.csv

Unreliability of the Plant



6. Perspectives & Best Practices

Benefits of AltaRica 3.0

- **Reusable & modular models**: build once, reuse across systems and projects
- **Scalable abstraction**: from simple RBDs to complex hybrid models
- **Rich toolchain**: simulation, Fault Trees (FTA), stochastic analysis, safety indicators
- **Formal consistency**: single modeling language across multiple analyses
- **Extensibility**: Fault Trees, Critical Sequences, Markov chains, Dynamic FTA…

Best Practices

- **Start simple**: begin with RBDs as an entry point before refining
- **Validate incrementally**: check assumptions and results at each modeling level
- **Keep modularity**: encapsulate subsystems for reusability and easier debugging
- **Trace requirements**: link models back to system specifications & safety goals
- **Document assumptions**: independence, failure laws, mission profile

Perspectives

- Increasing adoption in **industrial safety-critical systems** (aerospace, nuclear, automotive)
- Integration with **digital twins** for design validation and predictive maintenance
- Bridging gap between **academic models** and **industrial certification needs**

7. Ressources

📘 **Available resources** 📘

- **AltaRica software**: altarica-association.org/Products
- **Step-by-step case study**: altarica-association.org/Documentation/StepByStep
  - Includes PDF guide & AltaRica project files

Coming soon

- A **new step-by-step tutorial** will be published on the website
- Will be followed by a **dedicated webinar** to explore it in detail

contact@altarica-association.org

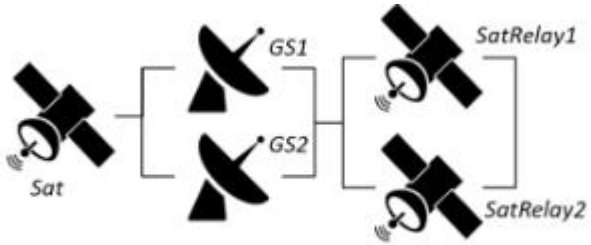


Figure 1: Reliability block diagram of the satellite communication system.

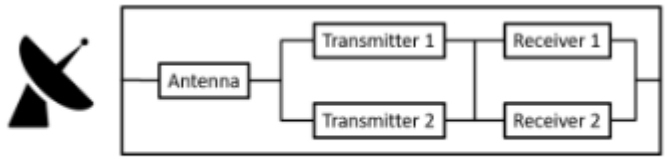


Figure 2: Reliability Block Diagram of the ground station (**GS1** or **GS2**).

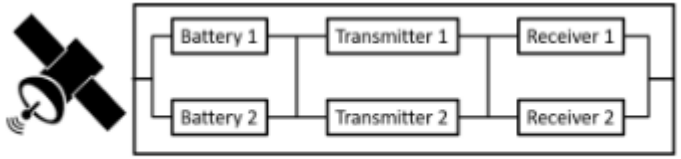


Figure 3: Reliability Block Diagram of the satellite (**Sat**, **SatRelay1** or **SatRelay2**).

8. Q&A

Thank you for your attention!

Any questions?

For more details or to share your needs, feel free to contact us:
contact@altarica-association.org

fractus

www.fractus.fr

Retrouvez nous sur :

anthony.legendre@fractus.fr | +33 (0)6 62 96 81 27