ASME 2021 International Mechanical Engineering Congress and Exposition
Track 14: Reliability and Safety in Industrial Automation Systems

Towards a New Generation of Probabilistic Safety Assessment Models and Tools.

Prof. Antoine B. Rauzy

Department of Mechanical and Industrial Engineering
Norwegian University of Science and Technology
Trondheim, Norway

One Observation, Two Questions

The observation:

Software and control mechanisms become ubiquitous in nowadays technical systems.

The two questions:

1. Are current modeling technologies for probabilistic risk/safety analysis, e.g. fault trees, still suitable to assess risks in new generations of systems?
2. Can we use the new capacities provided by information technologies to improve the probabilistic risk/safety analysis process?

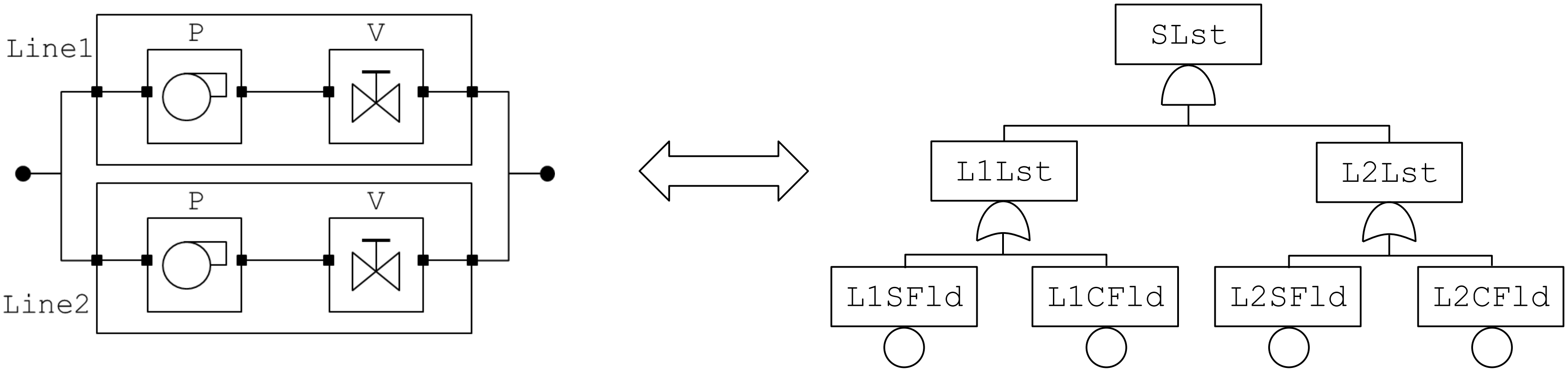
Agenda

- (R)evolution in Reliability Engineering
- The S2ML+X Family of Languages
- The Dialectic of Expressive Power and Computational Complexity
- Model Synchronization
- Wrap-Up

Issues with Current Probabilistic Safety Analyses

System specifications

Reliability models



- Combinatorial models (fault trees, reliability block diagrams, event trees) lack of expressive power to represent faithfully reconfigurations, control mechanisms, time dependencies…;
- States/events models (Markov chains, stochastic Petri nets) lack of structure;
- All are very distant from system specifications, making model hard to author, to share with stakeholders and to maintain through the life-cycle of systems.

(R)evolution in Reliability Engineering

Today:

Mechanical systems

Recording of failures

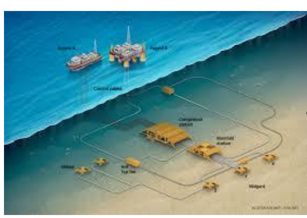
Local reliability databases

λ = 1.23e-6

Parametric distributions

Ad-hoc models, e.g. fault trees

Tomorrow:

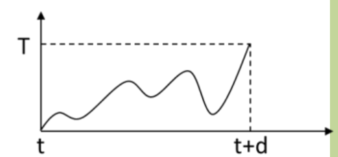


Cyber-physical systems

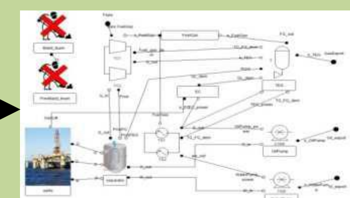
Health monitoring

Sensors

Distributed health condition databases



Learned distributions



Behavioral models

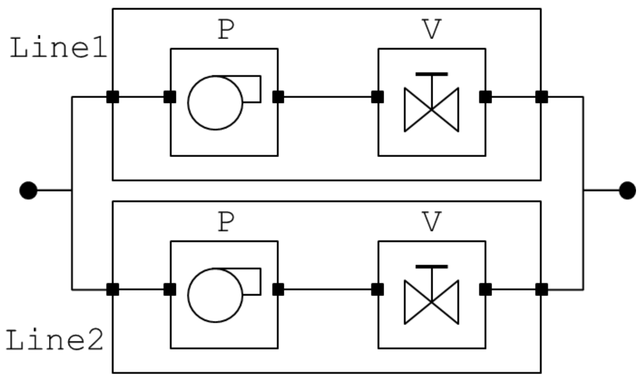
Agenda

- (R)evolution in Reliability Engineering
- The S2ML+X Family of Languages
- The Computational Complexity Challenge
- Model Synchronization
- Wrap-Up

Characteristics of Behavioral Models

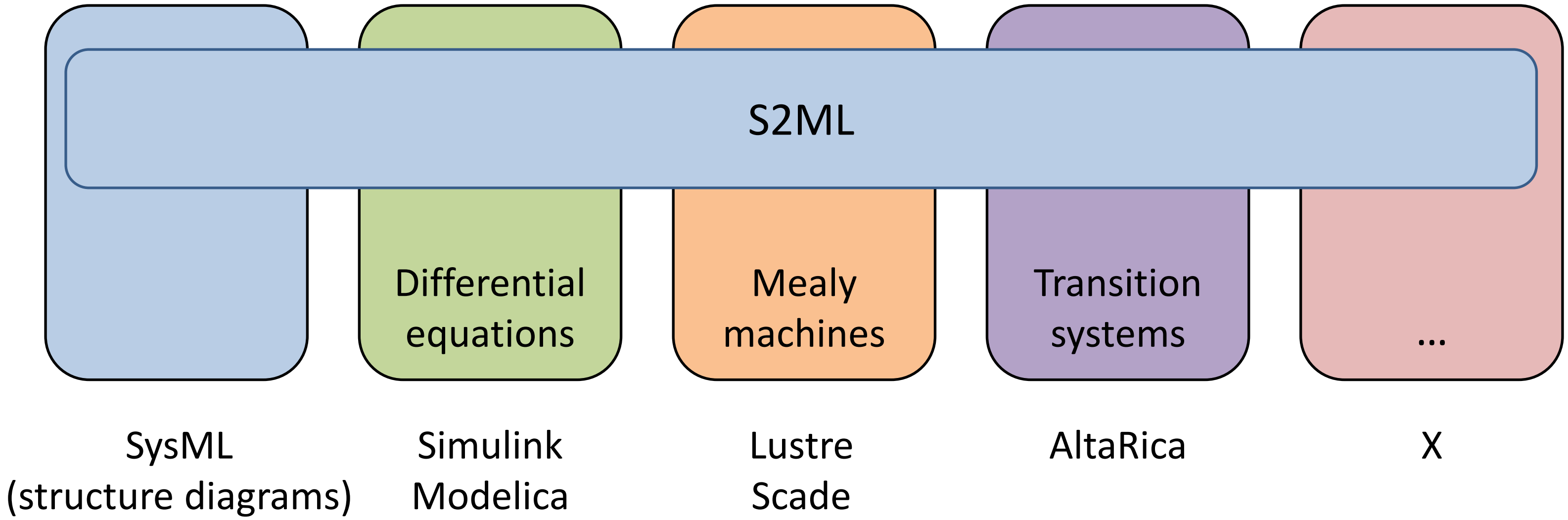
Behavior + Architecture = Model

- Any modeling language is the combination of a **mathematical framework** to describe the behavior and a **structuring paradigm** to organize the model.
- The choice of the **suitable mathematical framework** depends on which aspect of the system we want to study
- **Structuring paradigms** are to a very large extent **independent** of the chosen mathematical framework.



The S2ML+X Promise

S2ML (System Structure Modeling Language): a coherent and versatile set of **structuring constructs** for any behavioral modeling language.



- The **structure of models** reflects the **structure of the system**, even though to a **limited extent.**
- **Structuring** helps to design, to debug, to share, to maintain and to align heterogeneous models.

Models as Scripts

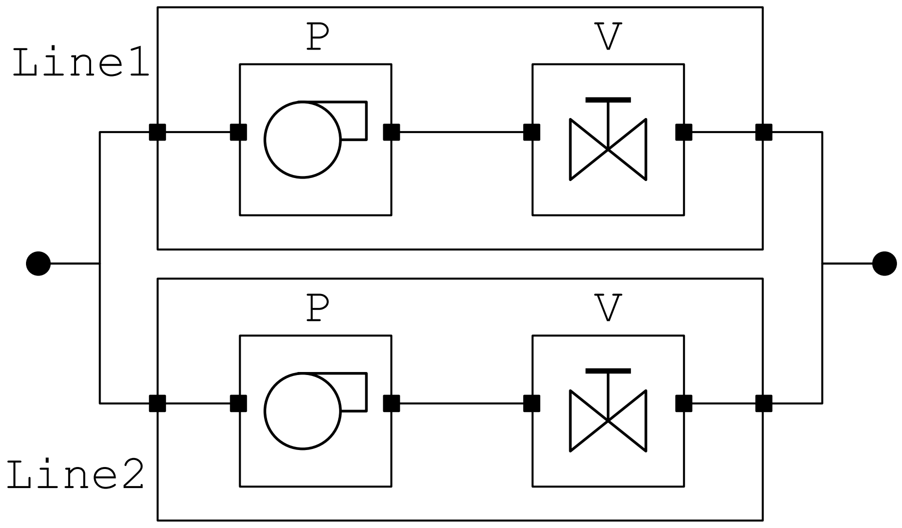
The **model "as designed"** is a script to build the **model "as assessed"**.

```
domain WF {WORKING, FAILED} WORKING<FAILED;

operator Series arg1 arg2 =
   return if state1==WORKING and state2==WORKING then WORKING else FAILED;

class Component
   WF state(init = WORKING);
   WF in, out(reset = WORKING)
   probability state FAILED = (exponentialDistribution lambda (missionTime));
   parameter Real lambda = 1.0e-3;
   assertion
      out := Series(in, state);
end
```

Complex models can be built using **libraries** of **reusable modeling components** and **modeling patterns**.

S2ML + Stochastic Boolean Equations

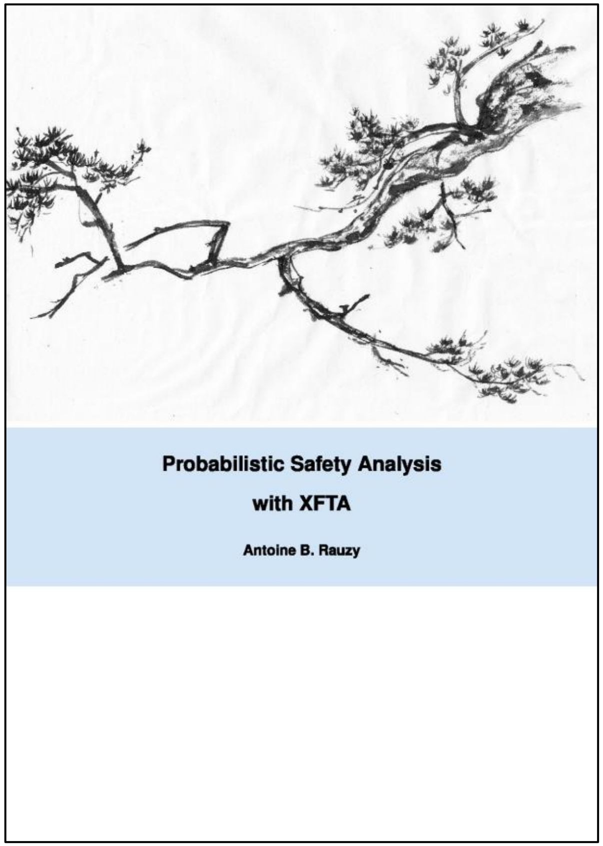
Enhancing classical **reliability models** (fault trees, reliability block diagrams) with the **expressive power of object-orientation** at **no algorithmic cost**




```
class Pump
    extends RepairableUnit
    …
end

block System
    block Line1
        Pump P;
        …
    end
    clones Line1 as Line2;
    …
end
```

```
Line1.in := in;
Line1.P.in := Line1.in;
Line1.P.out := Line1.P.in and not Line1.P.failed;
…
```

XFTA 2 + XFTA Book



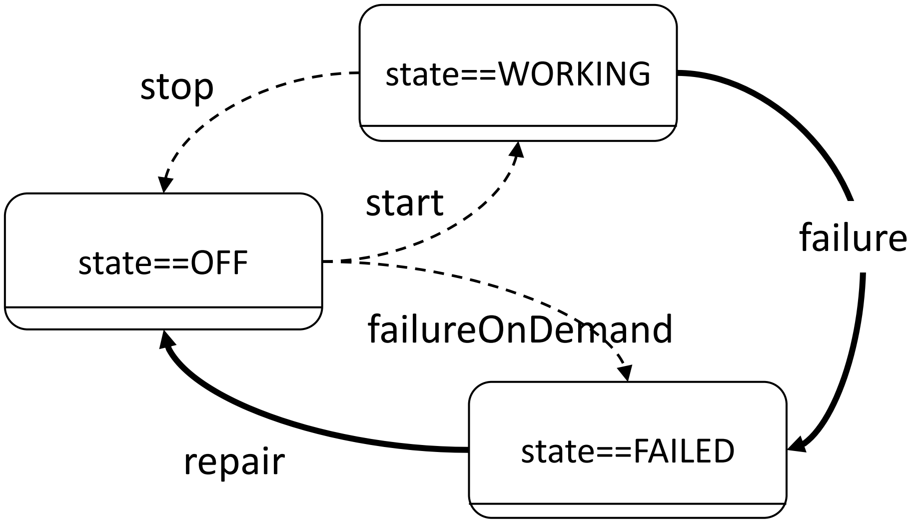
XFTA 2:

- Calculation engine for fault trees and related models.
- Input language: S2ML+SBE
- State of the art assessment algorithms: as of today most efficient calculation engine
- Calculation of all usual risk indicators:
  - Top event probability
  - Importance factors
  - Sensitivity analyses
  - Approximation of system reliability
  - Safety integrity levels
- Free of use, including for commercial purposes.

AltaRica 3.0 (S2ML + Guarded Transitions Systems)

Guarded Transitions Systems:

- Are a probabilistic Discrete Events System formalism.
- Are a compositional formalism.
- Generalize existing mathematical framework.
- Take the best advantage of existing assessment algorithms.



OpenAltaRica SYSTEM SAFETY ANALYSIS TECHNOLOGY | SystemX INSTITUT DE RECHERCHE TECHNOLOGIQUE

SAFRAN

THALES

AIRBUS

Agenda

- (R)evolution in Reliability Engineering
- The S2ML+X Family of Languages
- The Dialectic of Expressive Power and Computational Complexity
- Model Synchronization
- Wrap-Up

Classes of Modeling Languages

Combinatorial Formalisms

- Fault Trees
- Event Trees
- Reliability Block Diagrams
- Finite Degradation Structures

States Automata

- Markov chains
- Dynamic Fault Trees
- Stochastic Petri Nets
- …

Process Algebras

- Agent-based models
- Process algebras
- Python/Java/C++
- …

Expressive power →

States | States + transitions | Deformable systems

Complexity of assessments →

#P-hard but reasonable polynomial approximation | PSPACE-hard | Undecidable

Difficulty to design, to validate and to maintain models →

Best in Class Modeling Languages

Combinatorial Formalisms

Boolean models:

- Stochastic Boolean Equations
- S2ML+SBE
- XFTA

Multistate systems:

- Finite degradation structures
- S2ML+FDS
- Emmy (proof of concept)

States Automata

- Guarded Transition Systems
- S2ML+GTS = AltaRica 3.0
- AltaRica Wizard

Process Algebras

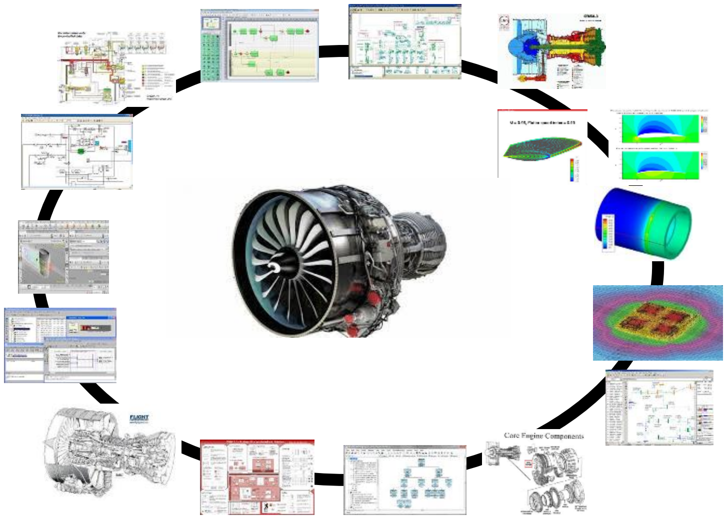
- Stochastic Process Algebras
- S2ML+SPA = Systema
- Systema Simulator (proof of concept)

Agenda

- (R)evolution in Reliability Engineering
- The S2ML+X Family of Languages
- The Dialectic of Expressive Power and Computational Complexity
- Model Synchronization
- Wrap-Up

Model Diversity

Models are designed by **different teams** in **different languages** at **different levels of abstraction**, for **different purposes**, making **different approximations**. They have also **different maturities**.

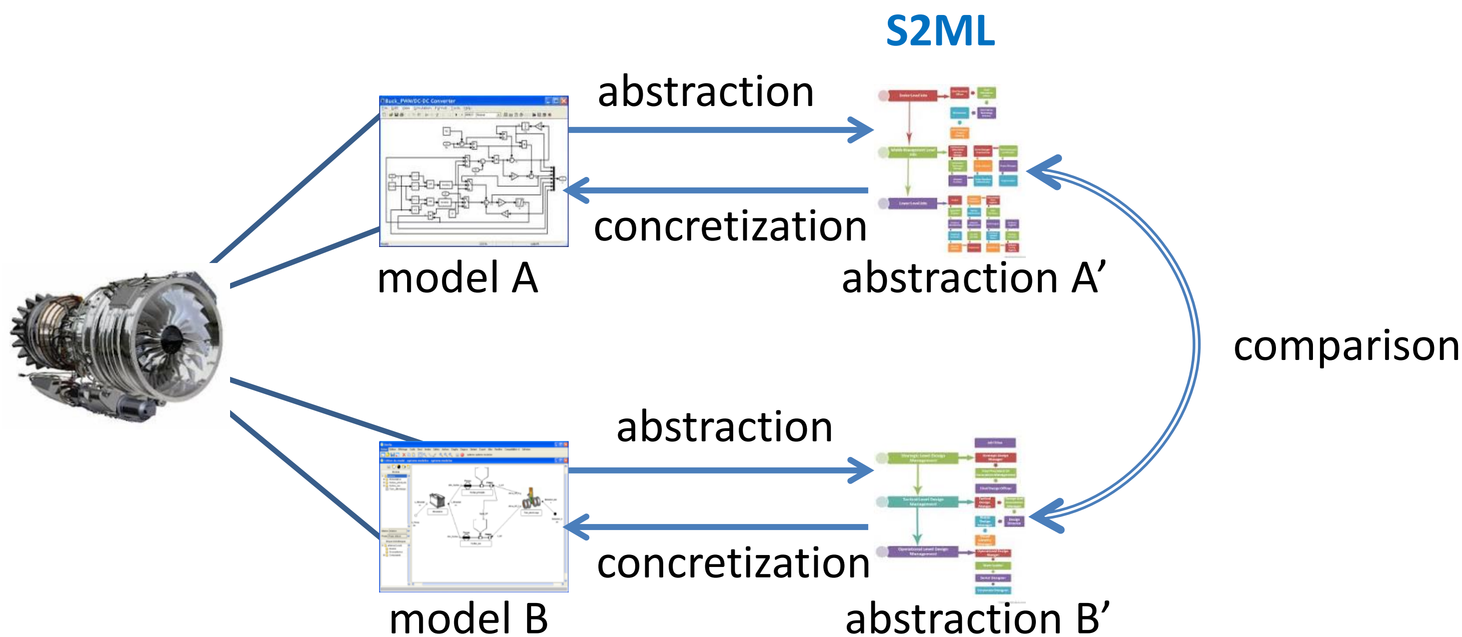


complexity → simplexity

The **diversity** of models is **irreducible.**

Model Synchronization

Abstraction + Comparison = Synchronization



How to agree on disagreements?

Agenda

- Introduction
- The S2ML+X Family of Languages
- The Dialectic of Expressive Power and Computational Complexity
- Model Synchronization
- Wrap-Up

Wrap-Up & Conclusion

- "Traditional" modeling approaches in reliability engineering are **no longer sufficient**:
  - Because the **systems** we are dealing with are **more complex**.
  - Because **new information technologies** open **new opportunities**.
  - Because **reliability models** should be **integrated** with models from other engineering disciplines.
- **Huge benefits** can be expected from a full-scale deployment of model-based systems engineering. However, this requires:
  - To set up solid **scientific foundations** for **models engineering**.
  - To **bring to maturity** some **key technologies**.
- The **biggest challenge** is to **train new generation of engineers**:
  - With skills and competences in **discrete mathematics** and **computer science**, and
  - With skills and competences in **system thinking**, and
  - With skills and competences in **specific application domains**.