CentraleSupélec

Probabilistic Risk Assessment Software: A Computer Scientist Perspective

Antoine Rauzy
Chair Blériot-Fabre* - Ecole Centrale de Paris
&
Ecole Polytechnique
FRANCE

*sponsored by SAFRAN

CentraleSupélec

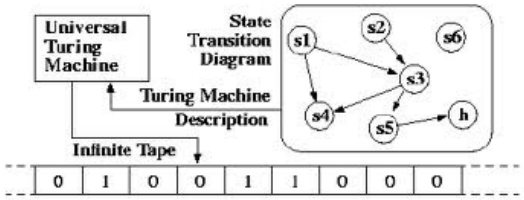
2012: Alan Turing Year

Alan Turing scientific contributions

Enigma

Cryptography:

Turing Machines

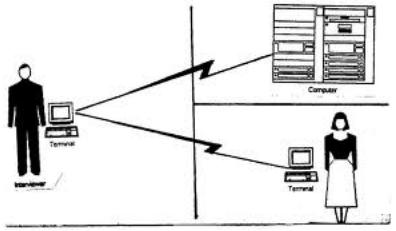


Complexity Theory:
Classification of Problems according to the amount of computation resources required to solve them.

The famous ($1 million) question:

P = NP ?

Turing Intelligence Test



Artificial Intelligence:
Two “devices” that respond the same way should be considered as equivalent

CentraleSupélec

Context

We are evolving in a changing economical/social/environmental context

- "Zero-risk", "open" society
- Increasing complexity of systems (systems of systems), ubiquity of software
- Evolution of the market: from products to capabilities

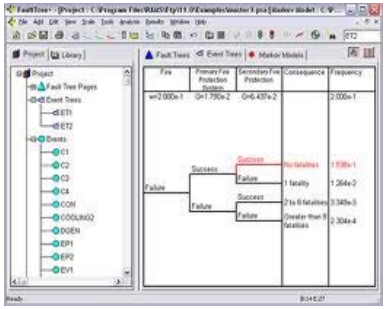
Engineering responses and **challenges**:

- Integration of Engineering Disciplines
- Model Based Design
- Use of **Probabilistic Risk Assessment** for
  - Safety Assessment (Certification…)
  - **Operational Decision** (Risk Informed Policy)
  - **Contractual Commitment** (System Availability)

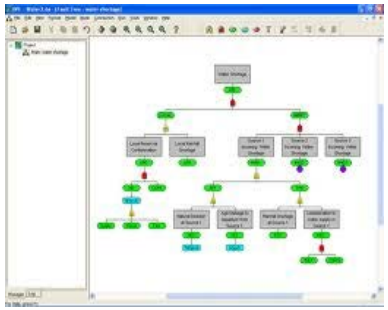
EPR Flamanville

PSA/PRA: Current Situation

Event Trees



Fault Trees



Rather well mastered PSA technology:

- Mature software
- Large models (typically 2000 Basic Events in Nuclear PSA models)
- Data bases of experience feedback

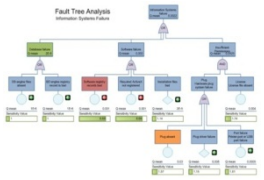
Issues:

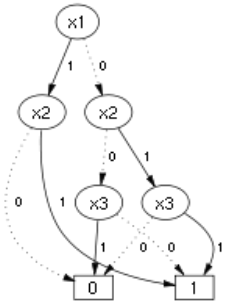
- Distance between systems and models of these systems
  - Even small changes in system specifications may require a complete review of PSA models
  - Difficult integration with other system engineering disciplines
- Limitations of modeling formalisms
  - No dynamicity, strong assumptions on statistical independence of Basic Events,…
  - Lack of reuse, difficulty of knowledge capitalization
  - Systemic effects (human factors, emergence) are hard if not impossible to take into account
- Complexity of assessments

Assessment Algorithms

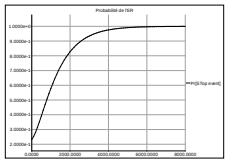
Because of repeated events, it is not possible to calculate reliability indicators straight from Fault Trees. An intermediate form has to be calculated.

Fault Tree





Reliability Indicators



Most efficient algorithms proposed so-far:

	Minimal Cutsets	Sum of Disjoint Products
Top-Down	(RSAT – RiskSpectrum) (XFTA)	
Bottom-up	ZBDD (FTREX – Cafta)	BDD (Aralia – Riskman)
	• approximation • efficient	• exact results • much more costly

For large PSA models, only **unwarranted approximations** can be calculated. They consists in keeping only **failure scenarios** whose **probability is over a predefined threshold**.

Probabilistic Assessment Uncertainty Principle


Turing Valiant
and others…

Theorem:

The calculation of a **warranted approximation** of the **top event probability** is **provably intractable** (under assumptions that are strongly believed to hold)!

Practical Consequence 1:

The failure scenarios we are considering are those with few (less than 10) basic components failed. We are looking at **small deviations from regular operations**, probably **not at accidents** where a large number of systems are failed, e.g. Fukushima.

Practical Consequence 2:

Refining a model (decomposing failure scenarios into finer ones) **may decrease the accuracy** of the assessment. At the limit, if scenarios are too much decomposed, they will all fall under the threshold. As a consequence, one will conclude that there is no risk.

PSA uncertainty principle:

We cannot have both a detailed model and accurate/complete calculations.

Probabilistic Assessment Uncertainty Principle

Turing

Valiant

and others…

Theorem:

The calculation of a **warranted approximation** of the **top event probability** is **provably intractable** (under assumptions that are strongly believed to hold)!

Practical Consequence 1:

The failure scenarios we are considering are those with few (less than 10) basic components failed. We are looking at **small deviations from regular operations**, probably **not at accidents** where a large number of systems are failed, e.g. Fukushima.

Practical Consequence 2:

Refining a model (decomposing failure scenarios into finer ones) **may decrease the accuracy** of the assessment. At the limit, if scenarios are too much decomposed, they will all fall under the threshold. As a consequence, one will conclude that there is no risk.

PSA uncertainty principle:

We cannot have both a detailed model and accurate/complete calculations.

Asking More to Models

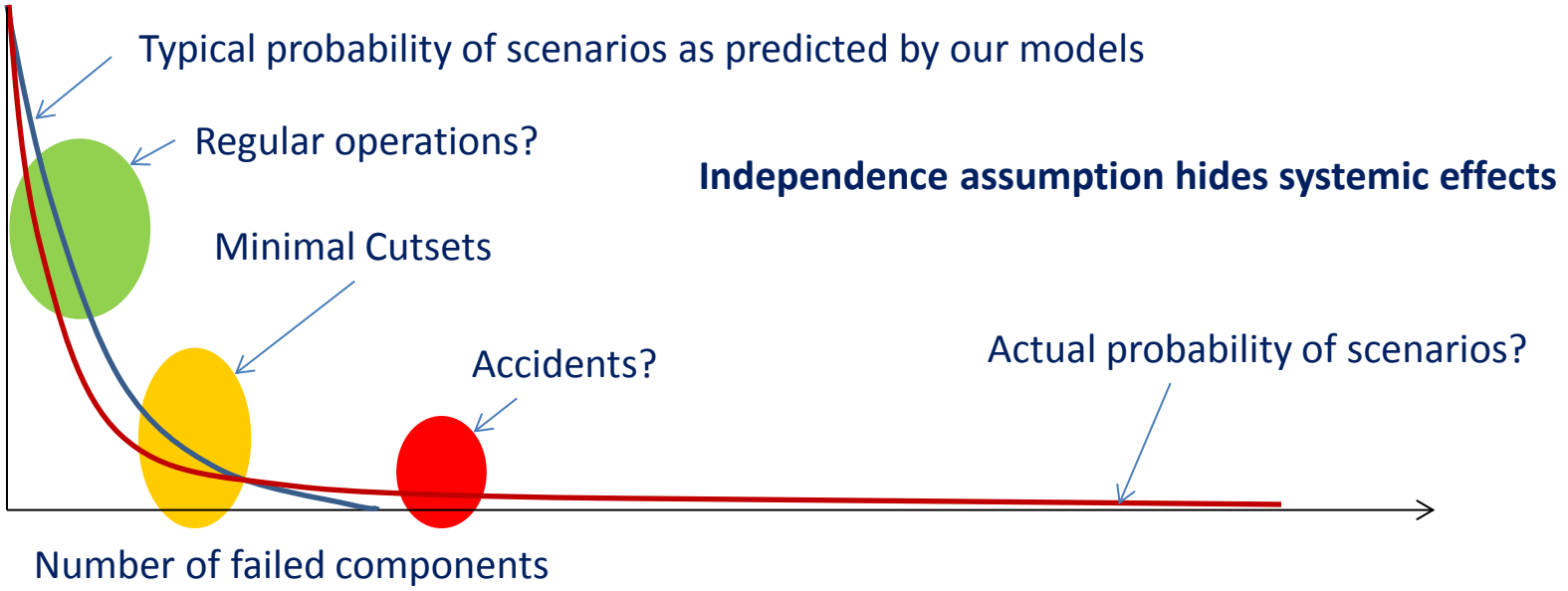
Challenge/research direction:

Design mathematical concepts, algorithms and tools to explore failure scenarios with more than a few components failed.

PSA models are made of two parts:

- A description of the “logic” of failure.
- Probability distributions for Basic Events.

We could probably keep the logic and use differently probability distributions.



Independence assumption hides systemic effects

Probabilistic Assessment Uncertainty Principle

Turing

Valiant

and others…

Theorem:

The calculation of a **warranted approximation** of the **top event probability** is **provably intractable** (under assumptions that are strongly believed to hold)!

Practical Consequence 1:

The failure scenarios we are considering are those with few (less than 10) basic components failed. We are looking at **small deviations from regular operations**, probably **not at accidents** where a large number of systems are failed, e.g. Fukushima.

Practical Consequence 2:

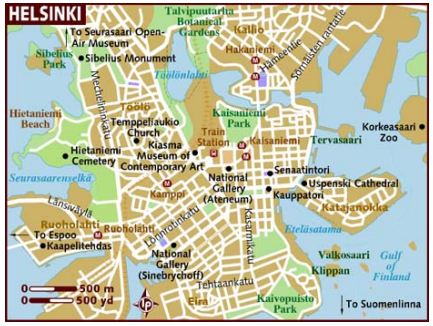
Refining a model (decomposing failure scenarios into finer ones) **may decrease the accuracy** of the assessment. At the limit, if scenarios are too much decomposed, they will all fall under the threshold. As a consequence, one will conclude that there is no risk.

PSA uncertainty principle:

We cannot have both a detailed model and accurate/complete calculations.

The Right Level of Abstraction

A model is designed to capture/study one aspect of the system under study/design.
It should be at the **right level of abstraction**.
E.g. a map abstracts away many irrelevant details and is useful because it does so.



Achieving the right level of abstraction is easy to say, but difficult to achieve as illustrated by Nuclear PSA models.

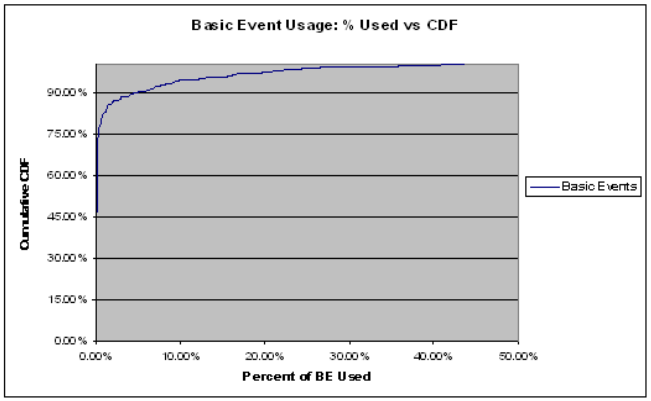
An illustrative and representative example (US).
What has been designed:

- ~2 500 Basic Events PSA model

What has been calculated:

- ~100 000 Minimal Cutsets
- 95% of the Core Damage Frequency with less than 5% of the Basic Events, 100% with 25%

In a word, 75% of the model is “useless”!



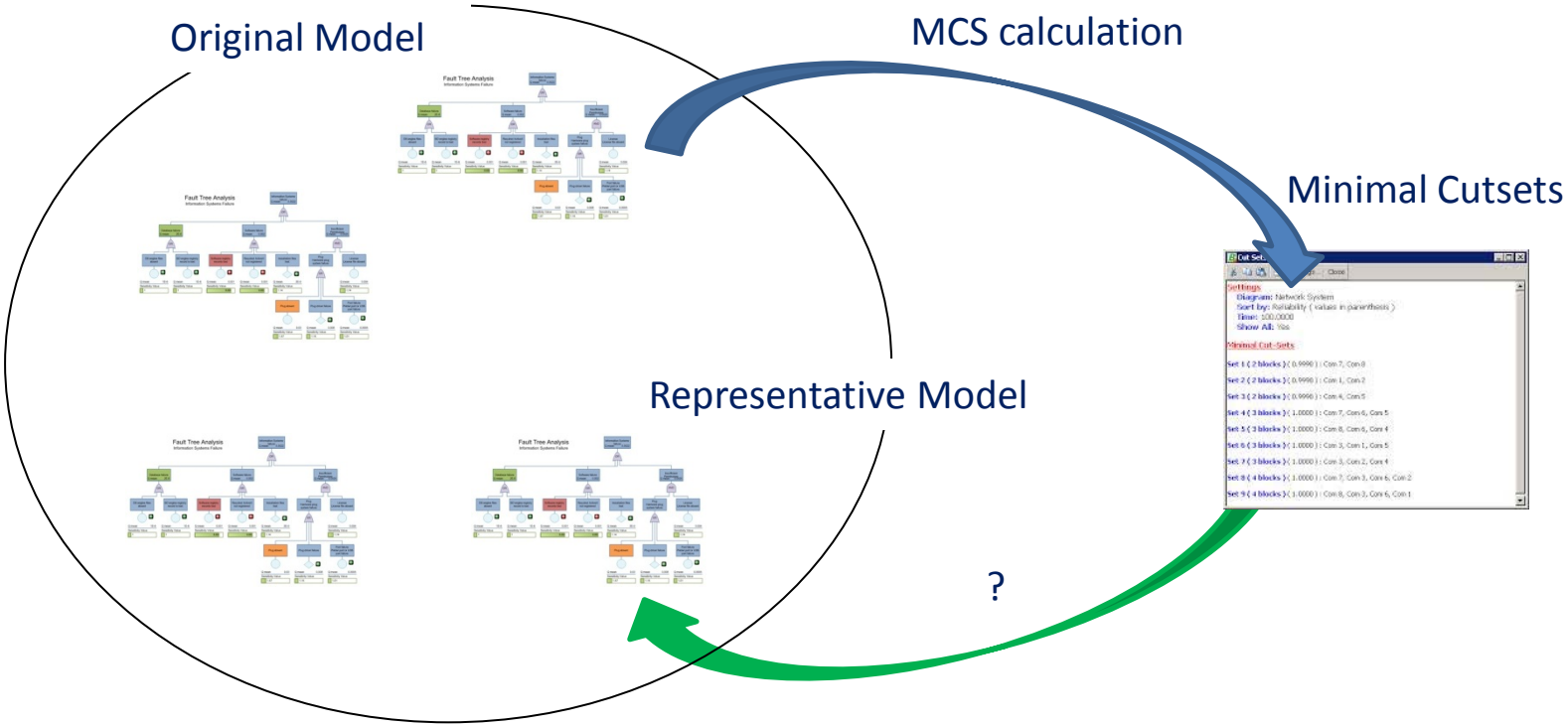
Categories of Models

Challenge/research direction:

Many possibly very different models are **undistinguishable** by **observation means**, i.e. results of virtual experiments (typically, calculation of failure scenarios). They are **equivalent** in the **Turing test** sense.

Equivalent models form a **category**.

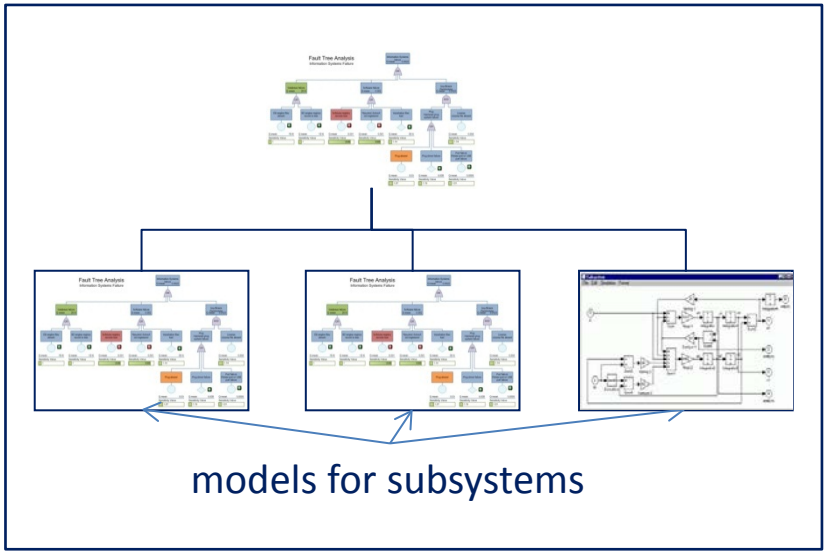
Design mathematical concepts, algorithms and tools to determine the **most representative** (simplest?) model of a category.



Abstraction

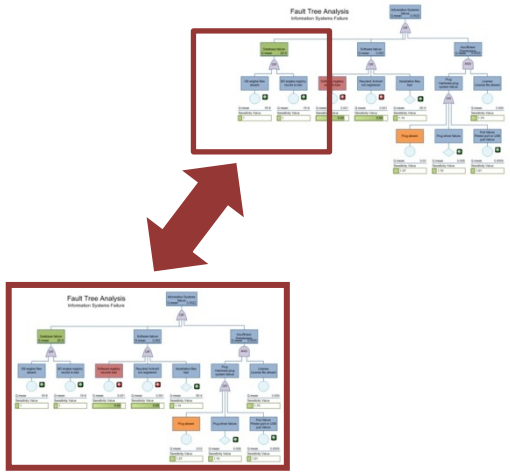
Complex systems need to be described by **multi-scale models**

- The composition of models of subsystems is often too big to be handled
- Models of subsystems are often heterogeneous… and designed by suppliers



Challenge/research direction:

Design mathematical concepts, algorithms and tools to **abstract** the model of subsystems into the model of the system and vice-versa



CentraleSupélec

Open-PSA Initiative

Statement of Purpose:

“We hope to provide an open and transparent public forum to disseminate information, independently review new ideas, and spread the word. We want to emphasize an openness which will lead to methods and software with higher quality, lead to better understanding of PSA models, encourage peer review, and allow the transportability of models and methods.”

from www.open-psa.org

Standard Representation Formats

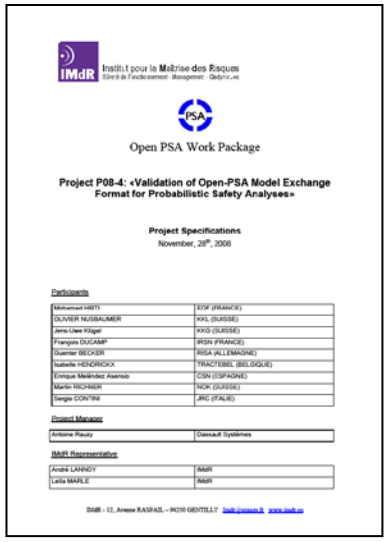
Two major trends:

- Models are more and more used as a contractual basis
- A high quality assurance is demanded on models

As a consequence, models must be:

- Peer-reviewed
- **Tool independent**

The **Open-PSA Standard Representation Format** for Fault Trees and Event Trees



Open PSA Work Package

Project P08-4: «Validation of Open-PSA Model Exchange Format for Probabilistic Safety Analyses»

Project Specifications

November, 28th, 2008

```
<define-fault-tree name="FT1" >
  <define-gate name="top" >
    <or>
      <gate name="G" />
      <basic-event name="C" />
    </or>
  </define-gate>
  <define-gate name="G" >
    <and>
      <basic-event name="A" />
      <basic-event name="B" />
    </and>
  </define-gate>
</define-fault-tree>
```

Challenge/research direction:

Define **standard representation formats**, with all the necessary constructs, with a clear and sound semantics

Version 3 of the OpenPSA standard under redaction

- Simplifications
- Block Diagrams
- Multi-phase Markov Chains with Rewards

CentraleSupélec

The XFTA Project

Provide the community with a Fault Tree solver

- Open-Source (C++)
  - Free of use, even in commercial packages
- Supporting the Open-PSA Model Exchange Format
- Implementing State-of-the-Art algorithms
  - Works for coherent and non-coherent models
- Calculating (reliability) indicators of interest
  - Minimal Cutsets
  - Top Event Probability, Importance Factors
  - Sensitivity Analyses, Time Dependent Analyses, Safety Integrity Levels

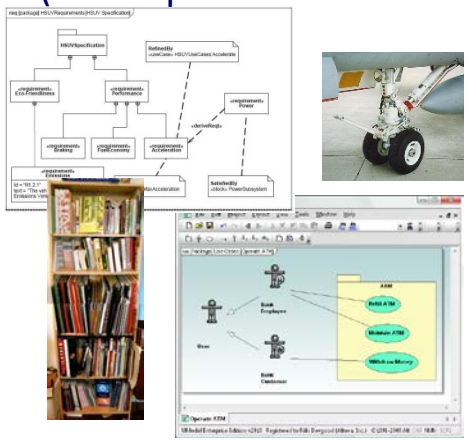
Experiments (on my laptop)

	#gates	#BE	cutoff	#MCS	running time
PSA 1	2096	1055	1.0e-12	146831	30s53
PSA 2	2722	1430	1.0e-12	36717	15s46

Version 1.1 available on my web page at Ecole Polytechnique (http://www.lix.polytechnique.fr/~rauzy)

Probabilistic Risk Assessment

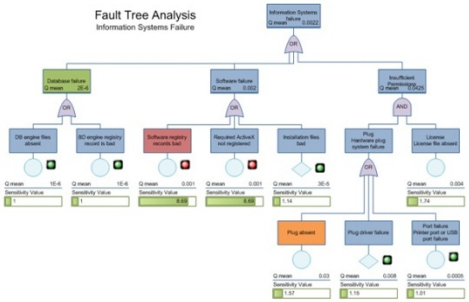
System Specifications
(and experience feedback)



Modeling

Improvements
Certification

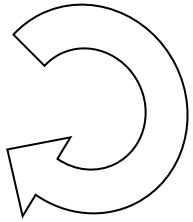
Models



Fault Trees, Event Trees,
Markov Chains, Stochastic
Petri Nets…

Virtual Experiments

- Failure Scenarii
- Reliability Indicators

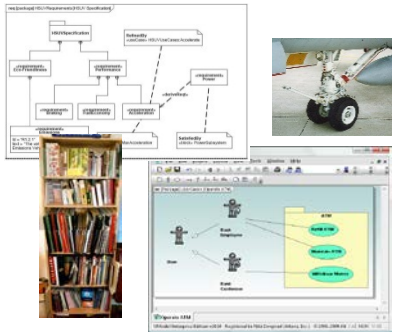


Issues:

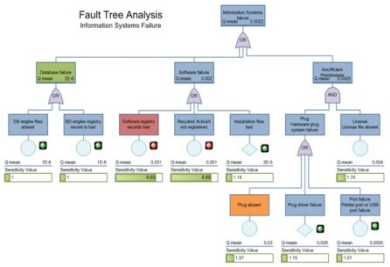
- **Completeness** of specifications with respect to safety concerns
- **Distance** between system specifications and safety models
- **Size** of the models
- **Complexity** of virtual experiments

The AltaRica Language

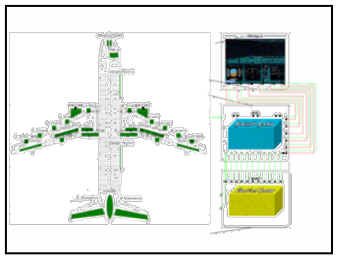
System Specification



Models



AltaRica



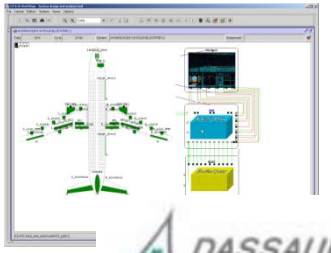
```
class component
  state Boolean working (init = true);
  event failure (delay = exponential(lambda));
 transition
    failure: working -> working := false;
end
```

Features of the language

- Formal
- Event-Based
- Textual & graphical
- Multiple assessment tools

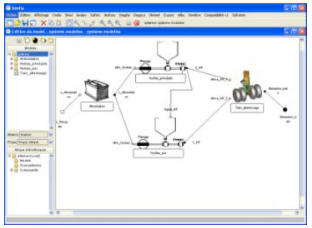
More than 10 years of industrial experience

Cecilia OCAS



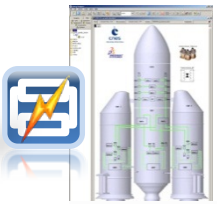
DASSAULT AVIATION

SimfiaV2



CASSIDIAN
AN EADS COMPANY

Safety Designer



DASSAULT SYSTEMES

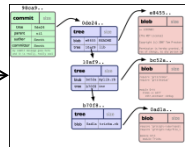
CentraleSupélec

The AltaRica 3.0 Project

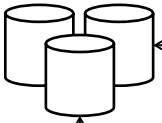
Reliability Data

GUI for modeling
GUI for simulation

Version & Configuration Management System



Libraries patterns

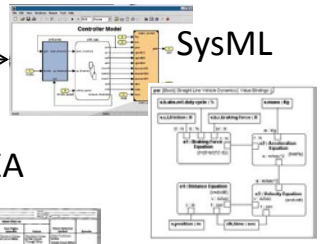


AltaRica

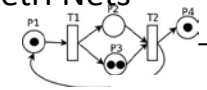
```
class Pump
...
end
```

AADL

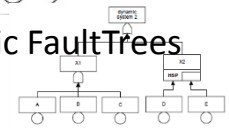
SysML



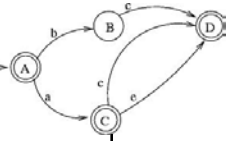
Petri Nets



Dynamic FaultTrees



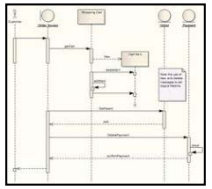
Guarded Transition Systems



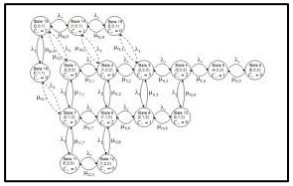
FMEA

compilation to Fault Trees

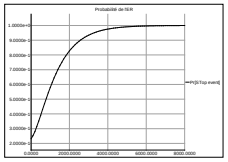
generation of sequences



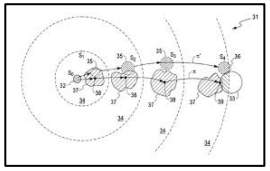
compilation to Markov Chains



stochastic simulation



model checking



reliability allocation

$$C_i(R_i) = e^{(1-f)\cdot\frac{R_i - R_{\min,i}}{R_{\max,i} - R_i}},$$