Tutoriel A3: A Reasoned Introduction to Model-Based Risk and Safety Assessments

SAINT-MALO
11 au 13 octobre 2016

MAÎTRISER LES RISQUES DANS UN MONDE EN MOUVEMENT

Michel BATTEUX
(IRT SystemX)

Tatiana PROSVIRNOVA
(IRT Saint-Exupéry)

Antoine RAUZY
(NTNU)

IMdR
Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

Note to the Reader

This tutorial about Model-Based Risk and Safety Assessment is strongly inspired by authors' work on the modeling language AltaRica (and more precisely AltaRica 3.0).

Other authors may have a different vision of the subject.

We believe in a scientific approach of the questions debated here. For us, each and every assertion must be supported by strong mathematical arguments as well as sufficiently many practical experiments on sufficiently large case studies.

In our domain, reaching this high standard requires not only mathematical and algorithmic knowledge and rigorous experimental protocols, but also a huge effort of software development.

Michel Batteux

Tatiana Prosvirnova

Antoine Rauzy

Agenda

- What is Model-Based Risk & Safety Assessment?
- Behaviors + Structures = Models
- Behavior Modeling Frameworks
- Model Structuring Frameworks
- Model Synchronization
- Frequently Asked Questions
- Some References

WHAT IS MODEL-BASED SAFETY ASSESSMENT?

Preliminary Remarks

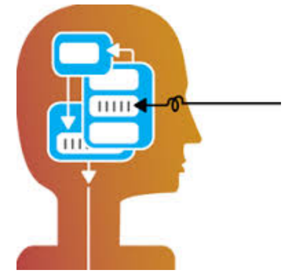
- Fault Trees, Block Diagrams, Event Trees and the like are models.
- Models are actually at the core of Risk and Safety Assessments since the very beginning of the discipline.

- **Model-Based Safety Assessment** (MBSA) differs thus from **Model-Based Systems Engineering** (MBSE) which is defined in contrast to text-based systems specifications.

What is a Model?

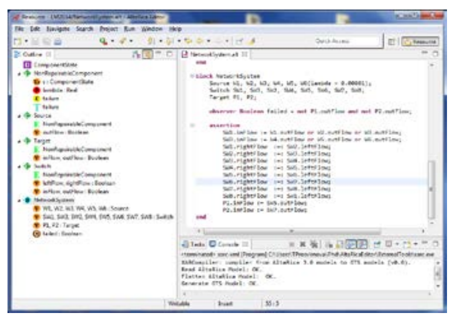
Star

Mathematical Model

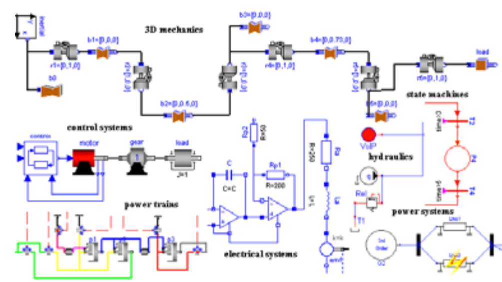


Cognitive Model

Mockup



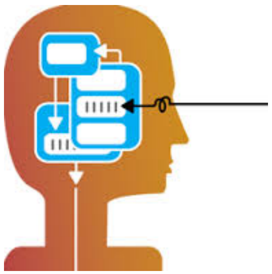
Code



Graphical Representation

All these “things” are models in some way

Models in (Safety and Reliability) Engineering

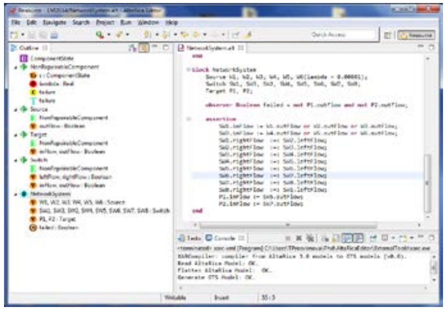


Cognitive Model

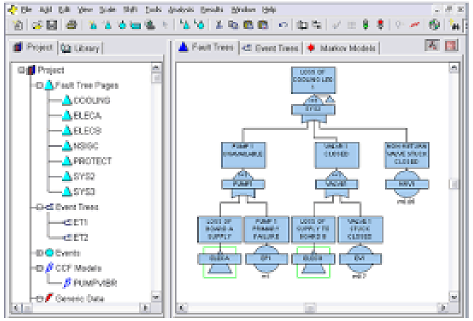
Mathematical Model

mind & paper models

computerized models



Code



Graphical Representation

Computerized Models

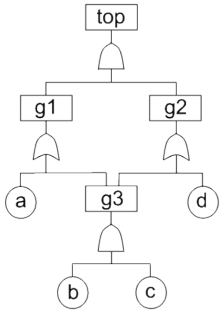
Computerized models (including graphical ones):

- are sequences of symbols that obey a given **syntax** (grammar);
- have a **formal semantics** (they are interpreted in a given mathematical framework);
- are designed primarily to perform **calculations** of risk related performance **indicators**.

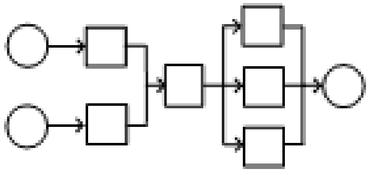
“Classical” Modeling Formalisms

Boolean formalisms | *Transitions Systems*

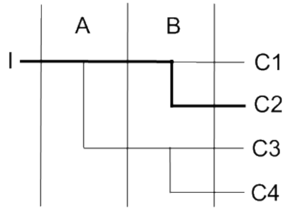
Fault Trees



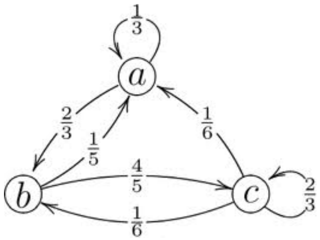
Blocks Diagrams



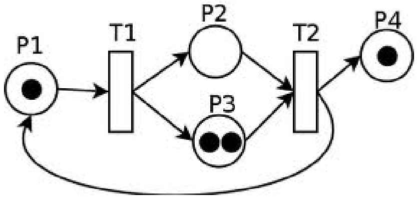
Event Trees



Markov Chains



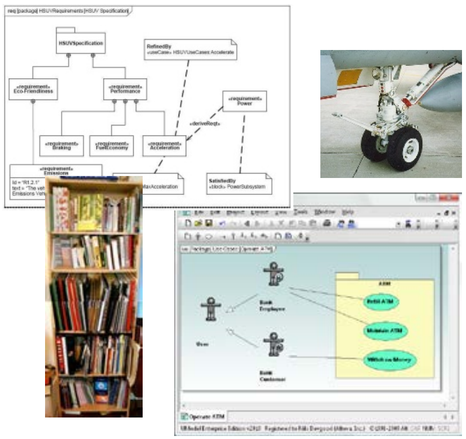
Stochastic Petri Nets



Note: for some applications, Bayesian networks are worth to consider

Issues with “Classical” Models

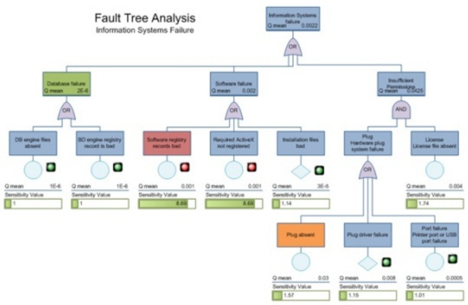
Systems Specifications



Modeling

Requirements, Certification process

Models



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

Virtual Experiments

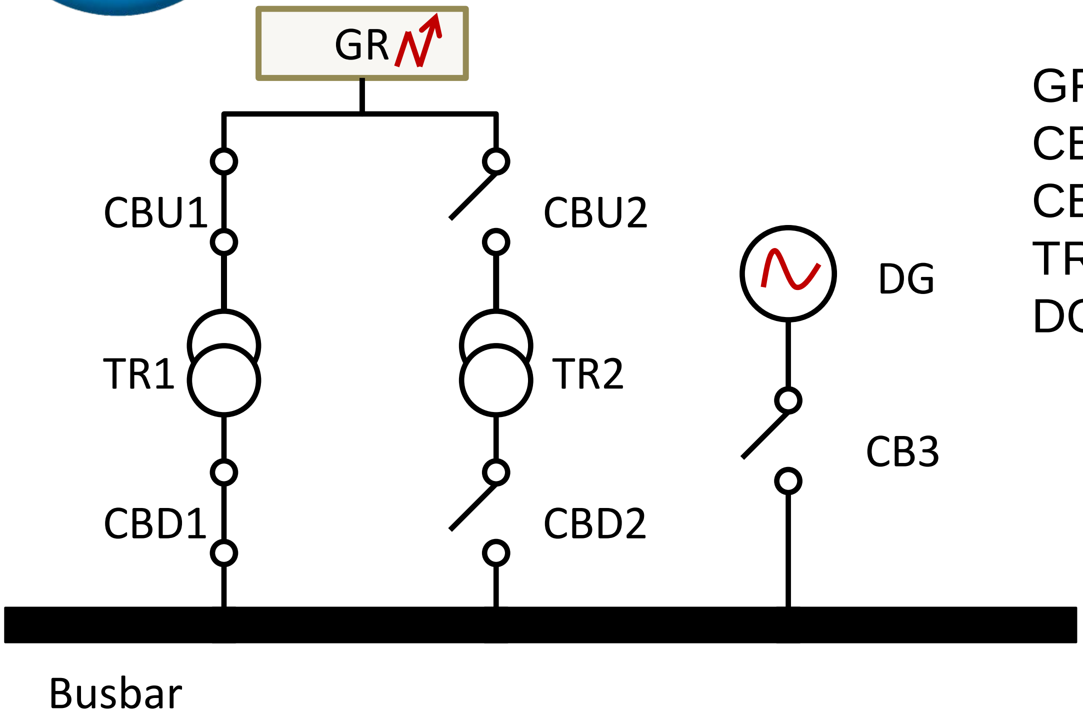
- Failure Scenarii
- Failure Probabilities

Classical modeling formalisms lack of expressive power and/or are very close to mathematical equations (lack of structure).

→ **Distance** between **systems specifications** and **models**;

→ Models are **hard to design** and even **harder** to **share with stakeholders** and to **maintain** throughout the **life-cycle** of systems.

Power Supply System(*)

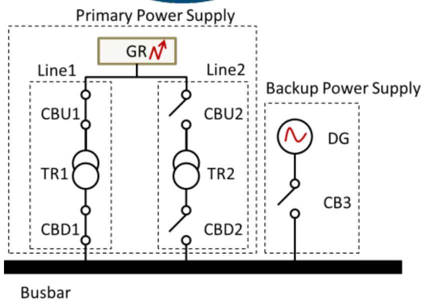


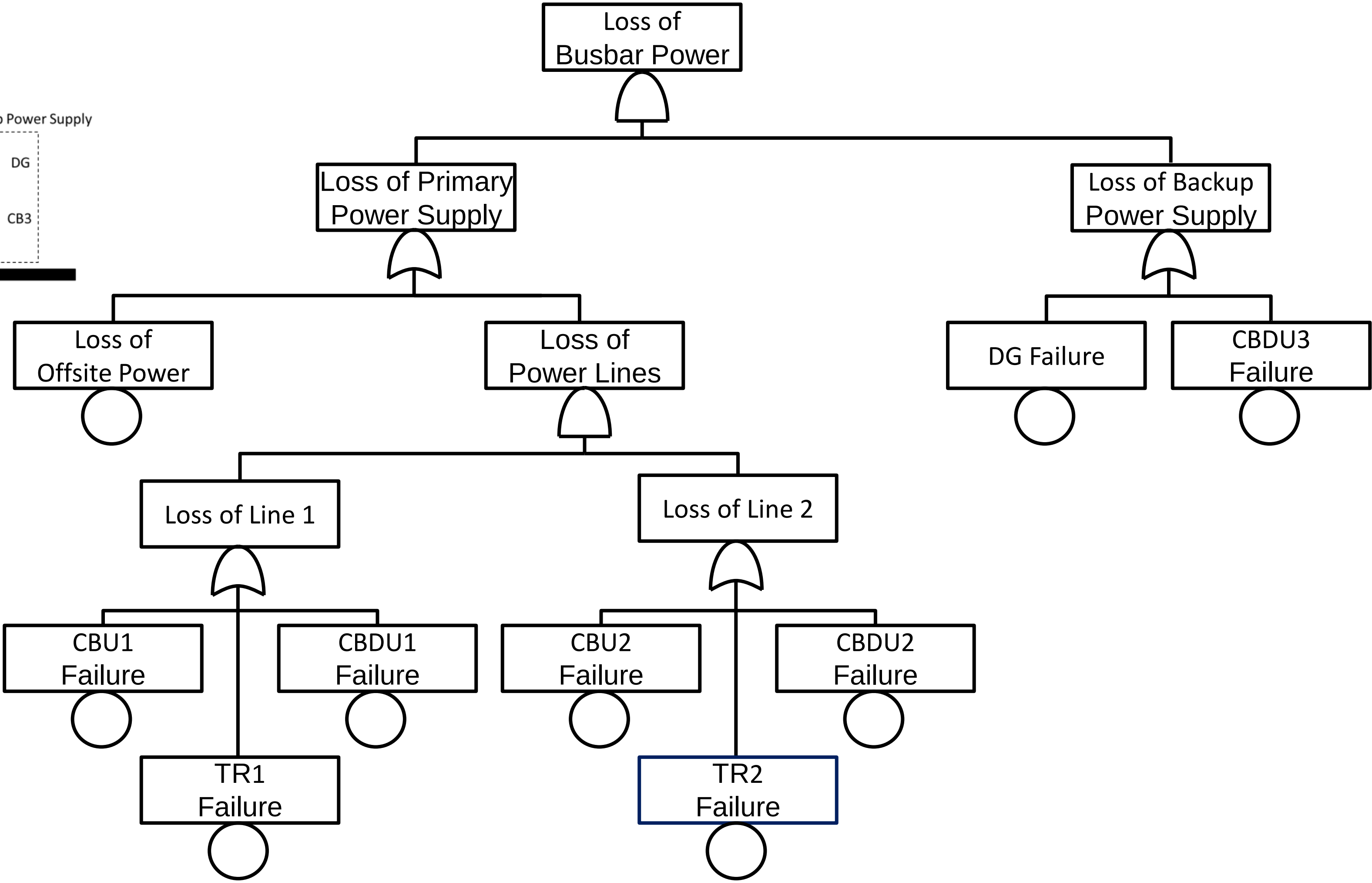
GR: Grid
CBUi: Circuit Breaker Up n°i
CBDi: Circuit Breaker Down n°i
TRi: Transformer n°i
DG: Diesel Generator

Assess the probability that the Busbar cannot be powered and find the sequences of events that lead to this situation

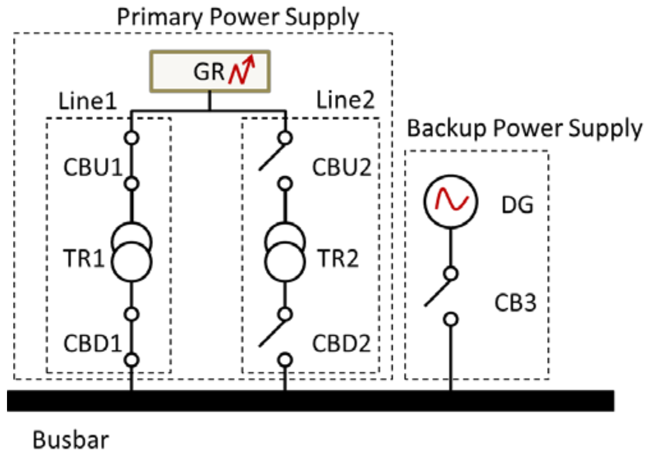
(*) Borrowed from Bouissou, M., Bon, J.L., A new formalism that combines advantages of fault-trees and Markov models: Boolean logic-driven markov processes. Reliability Engineering and System Safety 82 (2003) 149-163

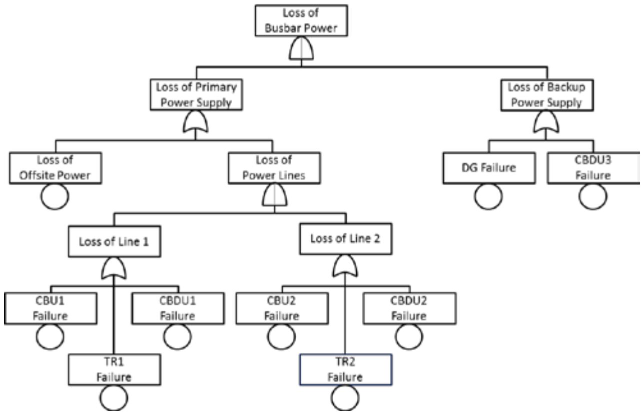
Fault Tree


Primary Power Supply
GR
Line1
Line2
Backup Power Supply
CBU1
CBU2
DG
TR1
TR2
CB3
CBD1
CBD2
Busbar


Loss of Busbar Power
Loss of Primary Power Supply
Loss of Backup Power Supply
Loss of Offsite Power
Loss of Power Lines
DG Failure
CBDU3 Failure
Loss of Line 1
Loss of Line 2
CBU1 Failure
CBDU1 Failure
CBU2 Failure
CBDU2 Failure
TR1 Failure
TR2 Failure

Issues

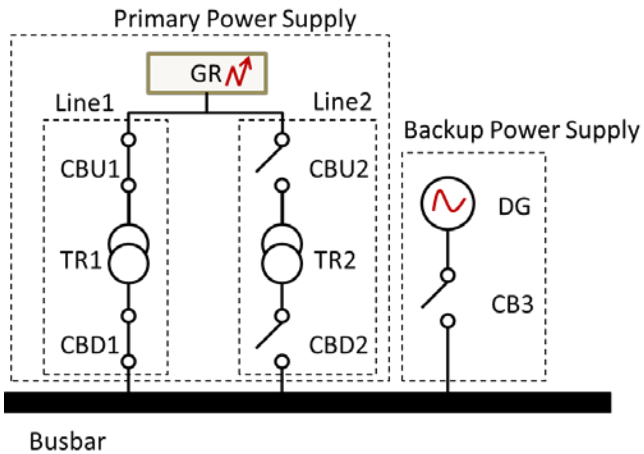




Mathematical issues (well known and accepted):

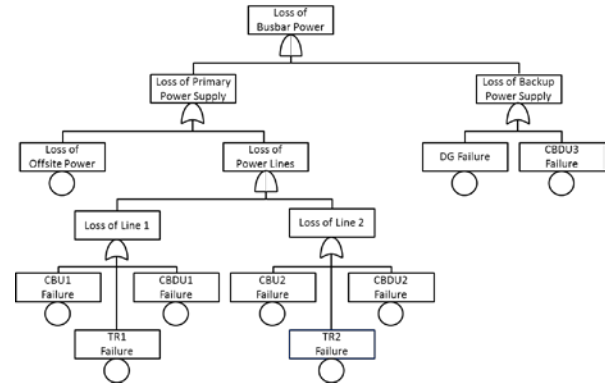
- Warm/Cold redundancies cannot be represented with Fault Trees
- Orders of events cannot be taken into account
- Common cause failures must be represented separately
- ...but the Markov chain for such system cannot be designed by hand (at least $2^9 = 512$ states)

Issues



difficult

nearly impossible



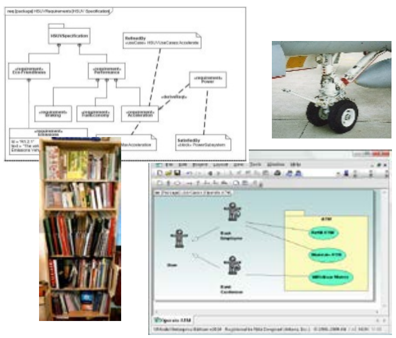
Modeling issues:

- Model does not reflect the architecture of the system (no way back)
- Model hard to check for correctness and completeness
- No possible “visual” simulation
- One model per safety goal

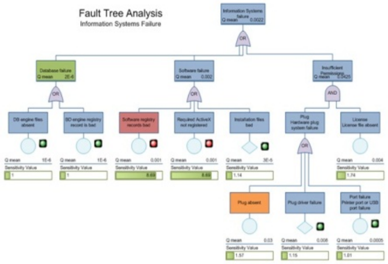
The Promise of MBSA

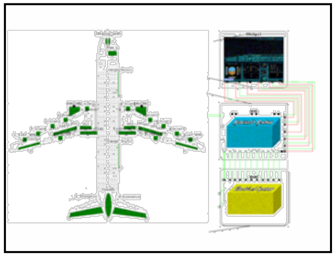
Modeling systems at **higher level** so to reduce the distance between systems specifications and models (without increasing the complexity of calculations).

Systems Specifications



Models





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

Complexity of Calculations

- **Calculations** of risk and safety related indicators are **extremely resource consuming**.
- This is not a problem of technology, it has been **mathematically proven** that they are **computationally intractable**.

→ **Models** result always of a **tradeoff** between the accuracy of the description and the ability to perform calculations.

BEHAVIORS + STRUCTURES = MODELS

Central Thesis

Behaviors + Structures = Models

Mathematic framework

- Ordinary Differential Equations
- Mealy Machines
- Probabilistic Boolean Algebras
- Petri Nets
- Bayesian Networks
- Guarded Transitions Systems
- …

Structuring paradigm

- Block Diagrams
- Object-Oriented
- Prototype-Oriented

Modelica

Lustre

Fault Trees

Reliability Block Diagrams

Special Case: Architecture Languages

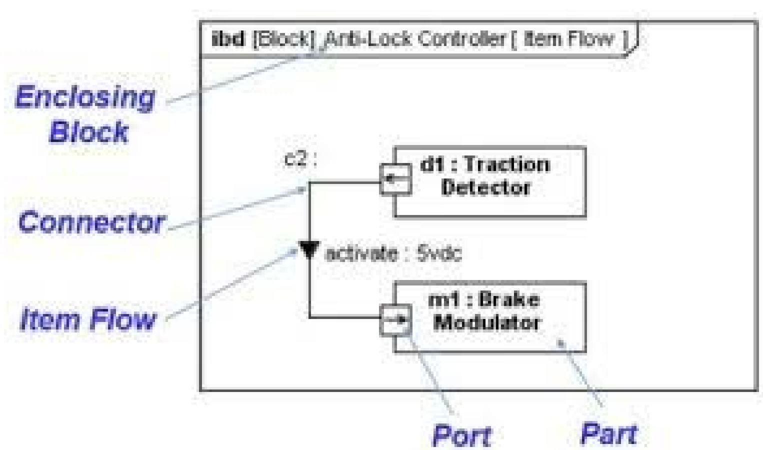
Mathematic framework

- …
- Empty
- …

Structuring paradigm

- (extended) Block Diagrams
- …

SysML structural diagrams (BDD, IBD)



Questions

- What are the good mathematical frameworks for risk and safety assessment?
- What are the good structuring paradigms for these mathematical frameworks?

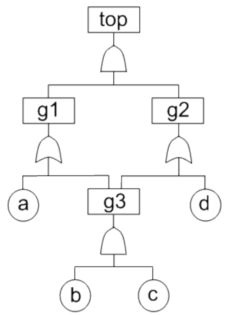
Recall: no universal panacea…

BEHAVIOR MODELING FRAMEWORKS

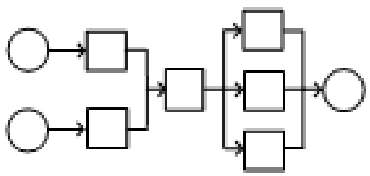
"Classical" Modeling Formalisms

Boolean formalisms | *Transitions Systems*

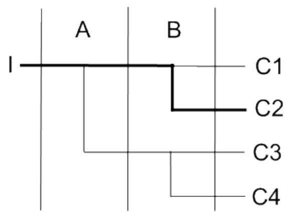
Fault Trees



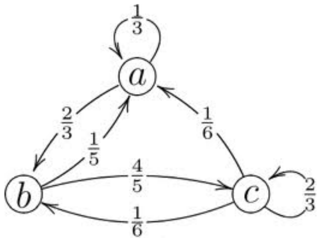
Blocks Diagrams



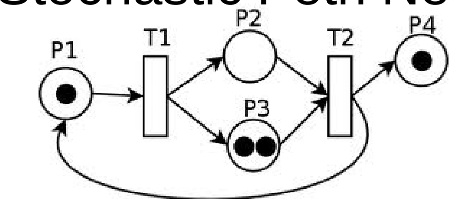
Event Trees



Markov Chains



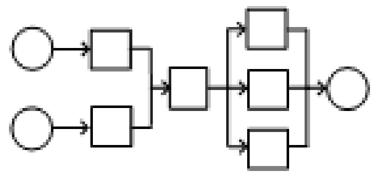
Stochastic Petri Nets



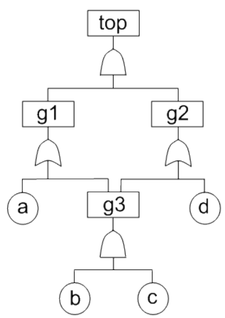
Common Characteristics:
- **Event-Based**
- **Probabilistic**

Boolean Formalisms

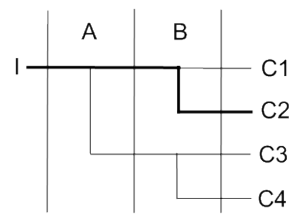
Blocks Diagrams



Fault Trees



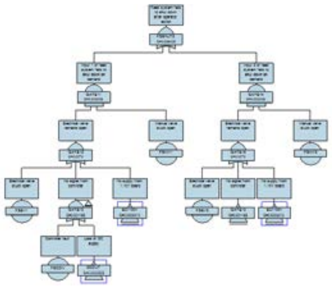
Event Trees



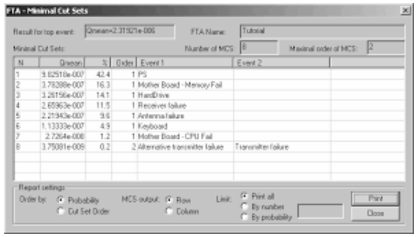
Boolean models are automatically transformed into equivalent Fault Trees before assessment.

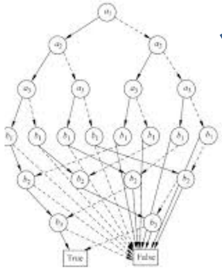
Assessment Algorithms

Model (Fault Tree)

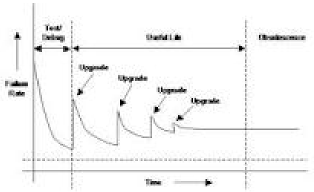


Minimal Cutsets, Prime Implicants





Binary Decision Diagrams



Indicators

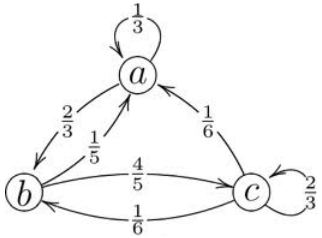
- Unavailability
- Importance Factors
- Safety Integrity Level
- …

Pros & Cons

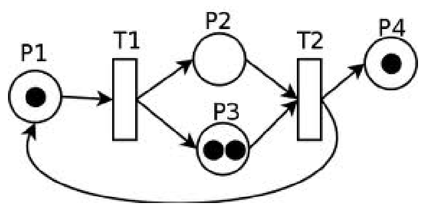
- Pros
  - Well mastered
  - “Easy” to understand
  - Efficient assessment algorithms (see articles by A. Rauzy)
  - Many available software
  - …
- Cons
  - Lack of expressive power
  - Very distant from systems specifications
  - One model per safety goal
  - …
- Possible extension
  - Finite domain algebra, e.g. {low, medium, high}

Transitions Systems

Markov Chains



Stochastic Petri Nets



Modeling

- Much more expressive power than Boolean formalisms
- Lack of structure (Markov chains, Petri nets)

Assessment

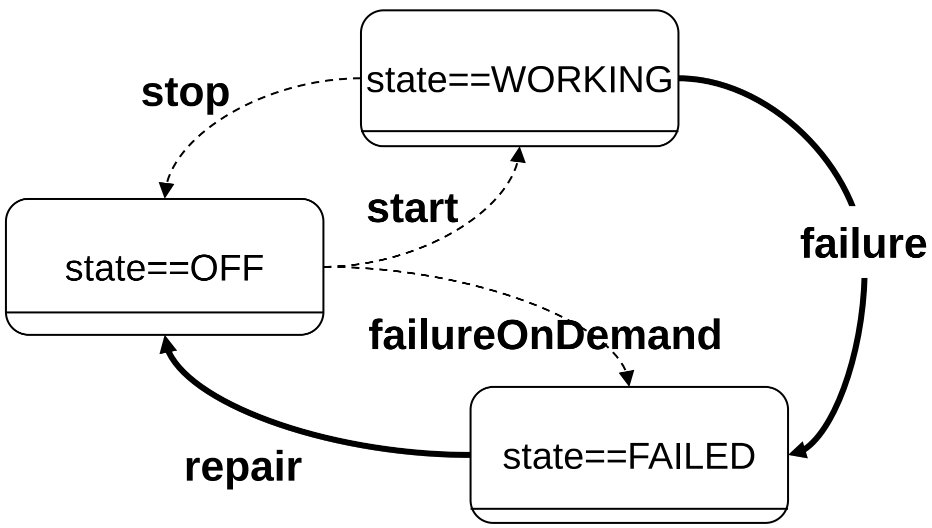
- Compilation into fault trees (not always possible)
- Compilation into Markov chains (not always possible)
- Sequence generation
- Monte-Carlo Simulation
- Model-checking
- …

Generic mathematical framework

- **Guarded Transitions Systems**

Guarded Transitions Systems

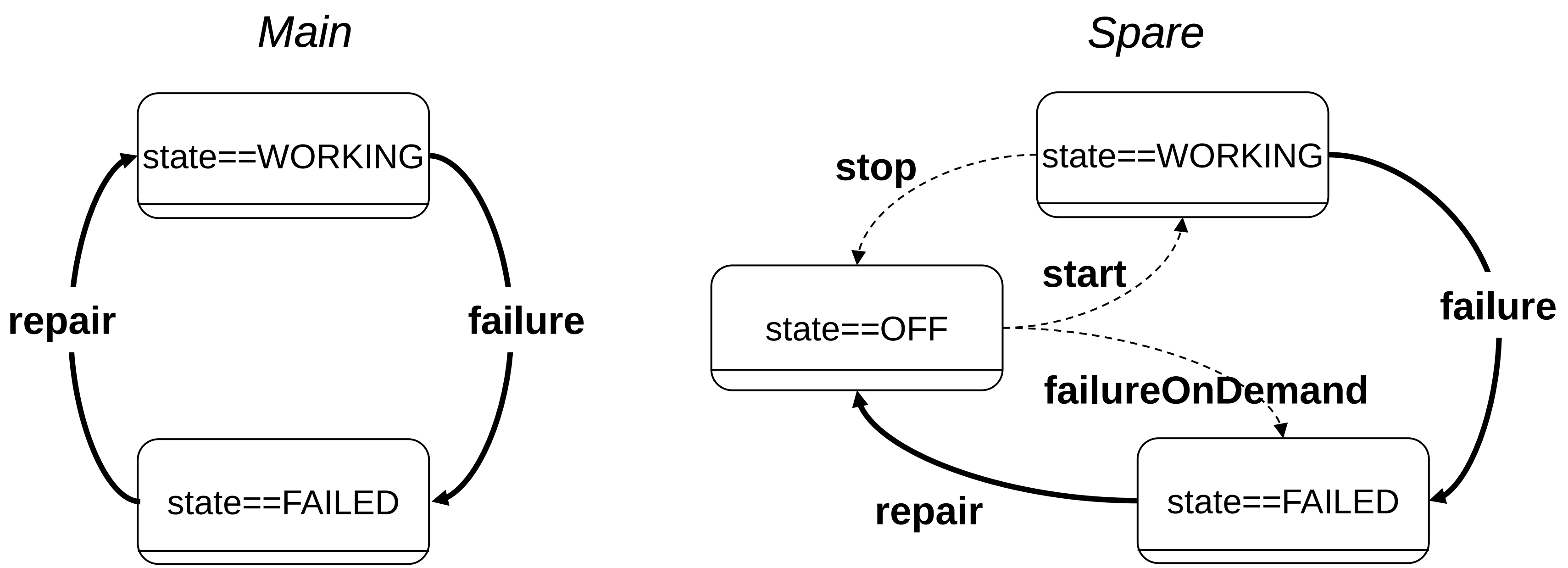
Spare Component



- The state of the system is represented by means of **(state) variables**.
- Variables take their value into domains (Boolean, sets of symbolic constants, integers…)
- Variables change of value when and only when an **event** occur, i.e. when the **transition** it labels is fired.
- A transition is fireable only when its **guard** (pre-condition) is satisfied.
- Events are associated with (stochastic) **delays** and/or with **probabilities**

Composition

The **synchronized composition** of two (or more) GTS is a GTS

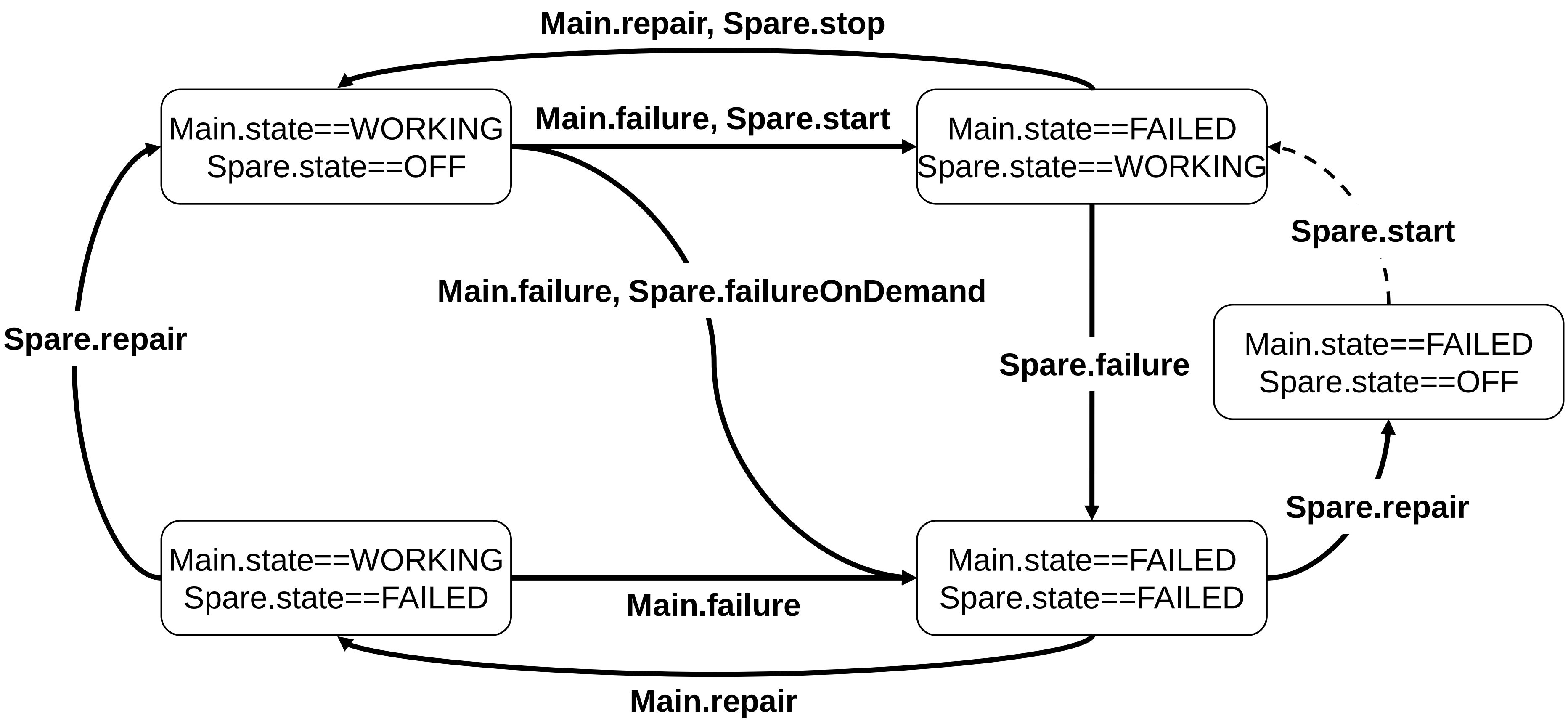


Synchronizations

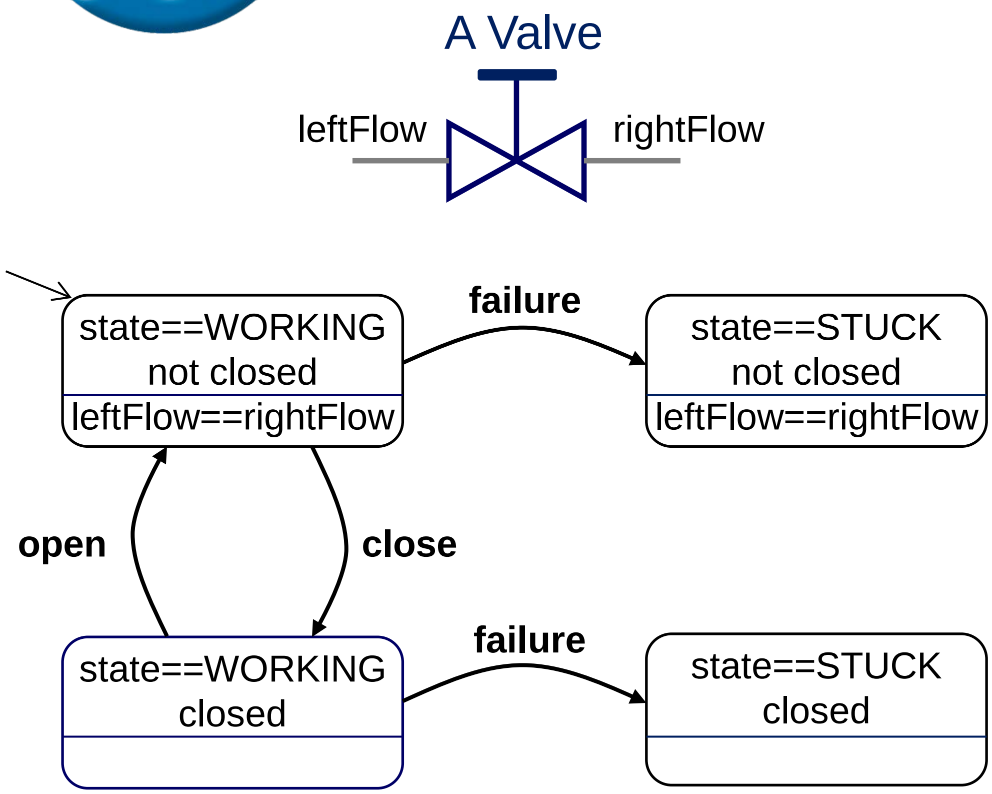
- Main.failure & Spare.start
- Main.failure & Spare.failureOnDemand
- Main.Repair & Spare.stop

Composition

The **synchronized composition** of two (or more) GTS is a GTS

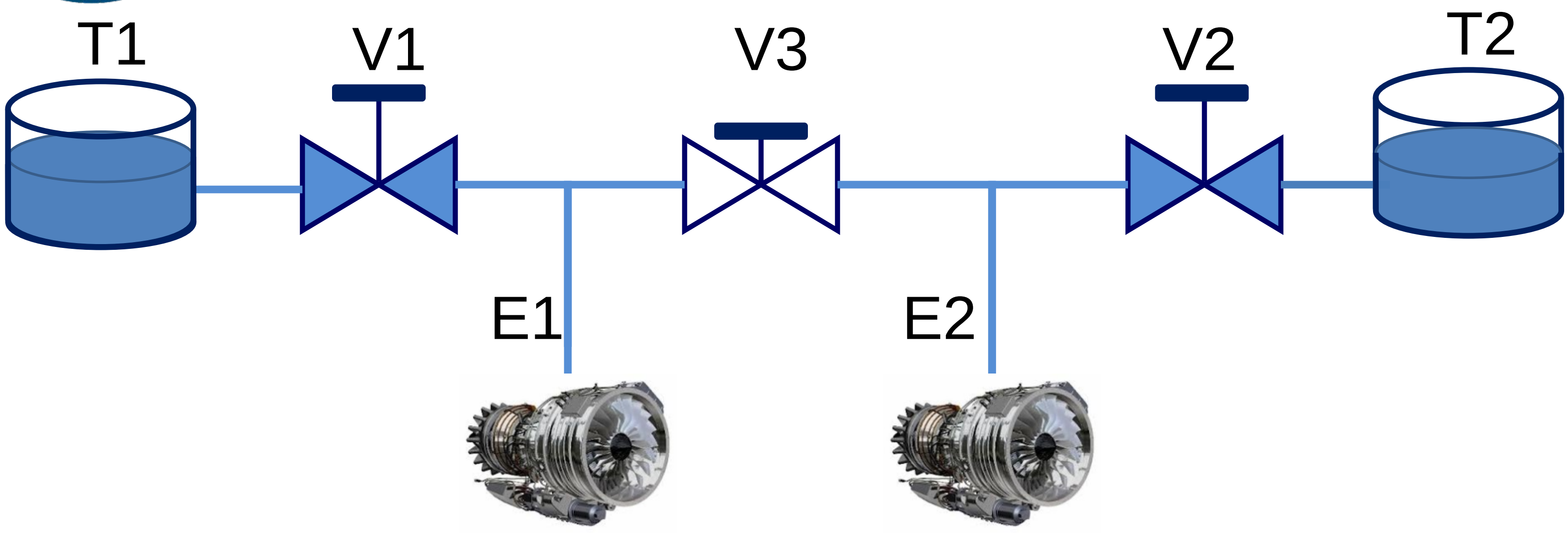


Flow Variables



- Flows of information/matters/energy circulating in the system are represented by means of **(flow) variables**.
- Flow variables take their value into domains (Boolean, sets of symbolic constants, integers…)
- Flow variables depend functionally on state variables: their value is entirely determined by the values of state variable

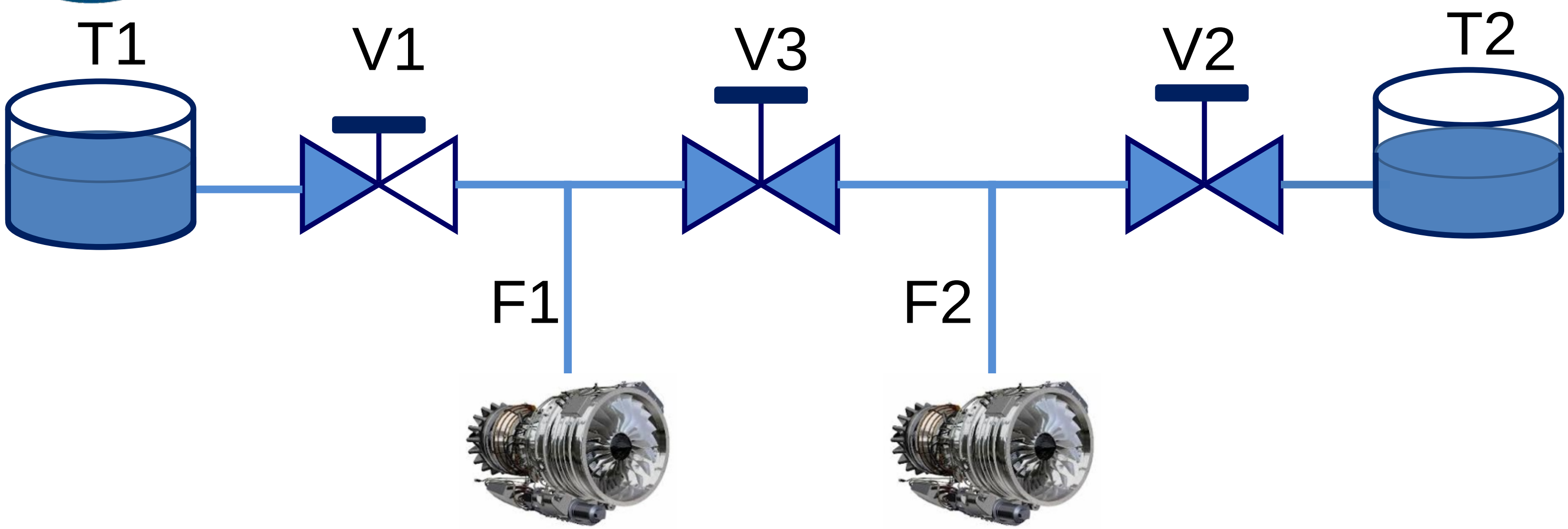
Flow Propagation (1)



The engine E1 is fueled through T1, and V1:

- not T1.isEmpty $\Rightarrow$ T1.outFlow
- T1.outFlow $\Rightarrow$ V1.leftFlow
- V1.leftFlow and not V1.closed $\Rightarrow$ V1.rightFlow
- V1.rightFlow $\Rightarrow$ E1.inFlow

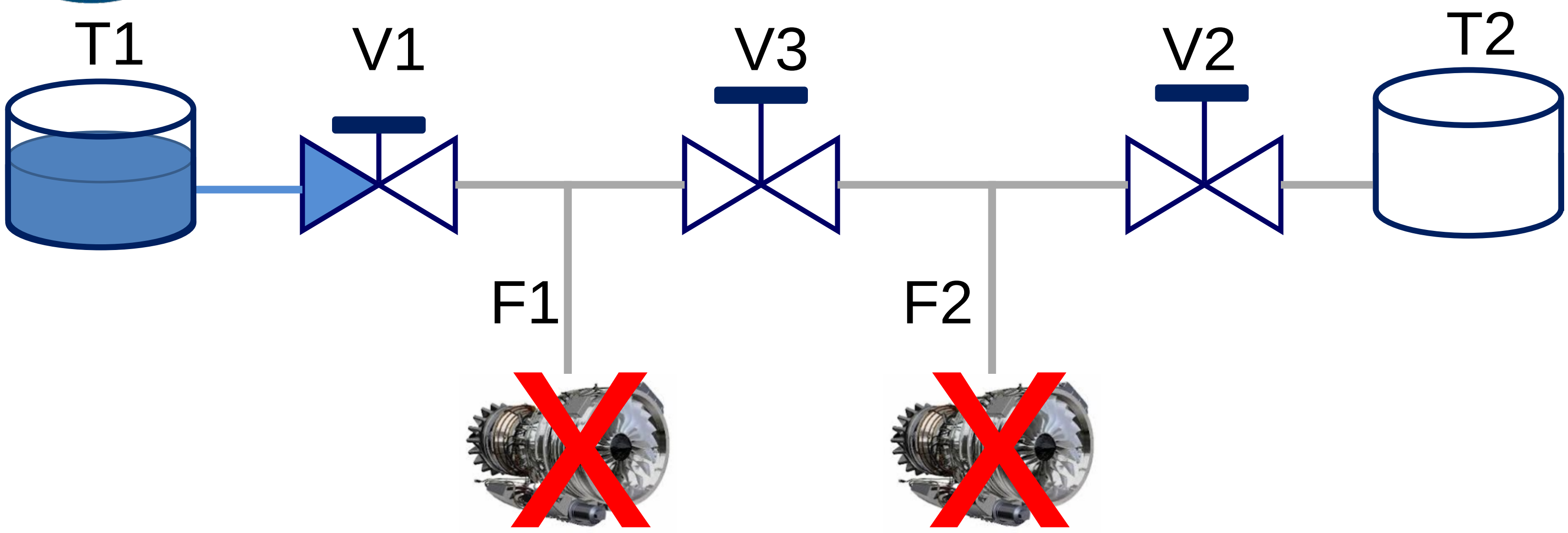
Flow Propagation (2)



Now, the engine E1 is fueled through T2, V2 and V3:

- not T2.isEmpty ⇒ T2.outFlow
- T2.outFlow ⇒ V2.rightFlow
- V2.rightFlow and not V2.closed ⇒ V2.leftFlow
- V2.leftFlow ⇒ V3.rightFlow
- V3.rightFlow and not V3.closed ⇒ V3.leftFlow
- V3.leftFlow ⇒ E1.inFlow

Flow Propagation (3)

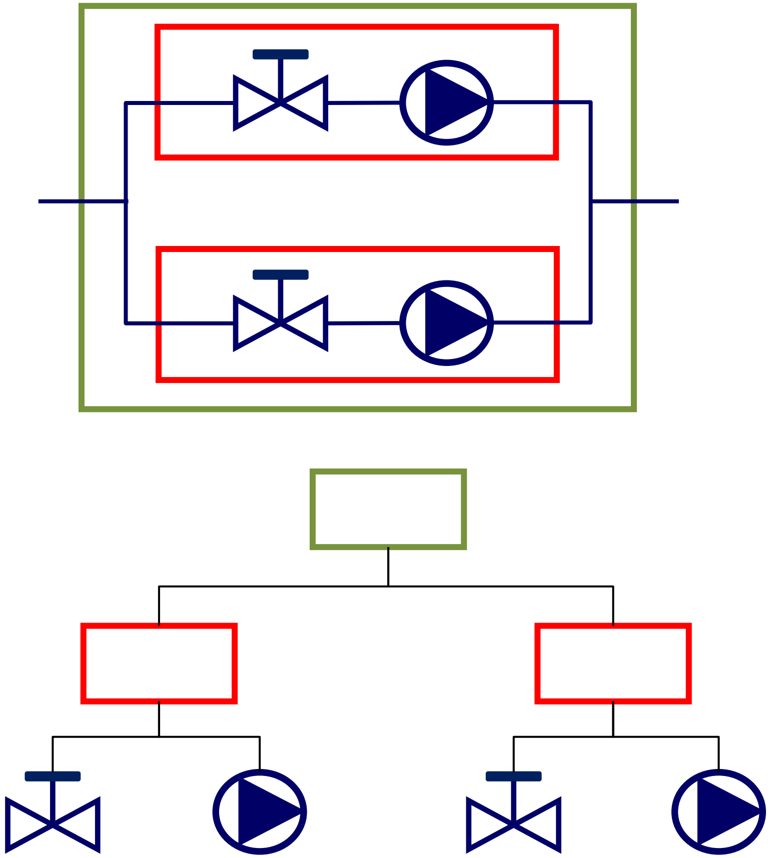


Now the engine E1 is not fueled

- not T2.isEmpty ⇒ T1.outFlow
- T1.outFlow ⇒ V1.leftFlow

The other flow variables are reset to their default values (false).

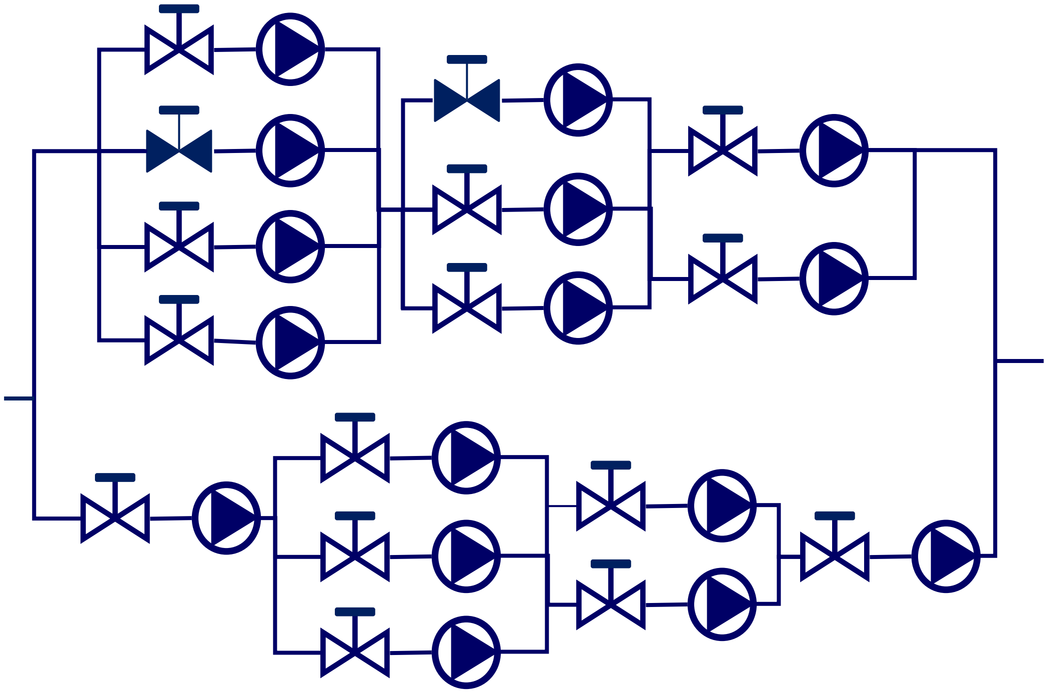
Hierarchical



GTS make it possible:

- To design models of systems by **composing** models of subsystems into **hierarchies**.

Implicit Representation of the State Space

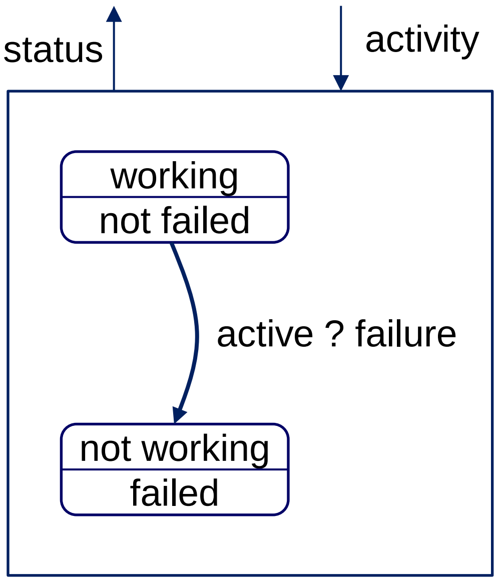


GTS make it possible:

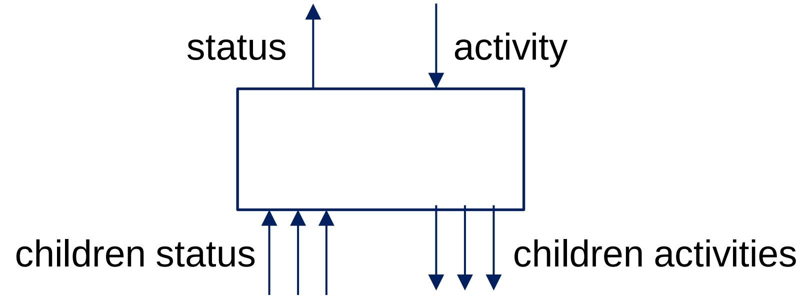
- To represent in an **implicit way** actual states and transitions of the system (**reachability graph**).
- To avoid (to some extent) the combinatorial explosion of the size of the model and to allow approximate calculations based on most probable scenarios/states.

GTS versus (Dynamic) Fault Trees

Basic Event



Gates

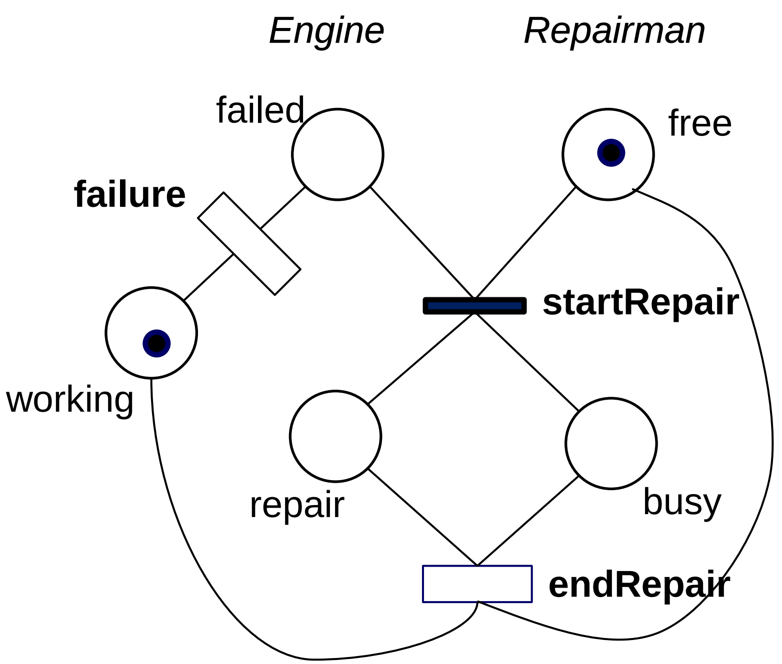


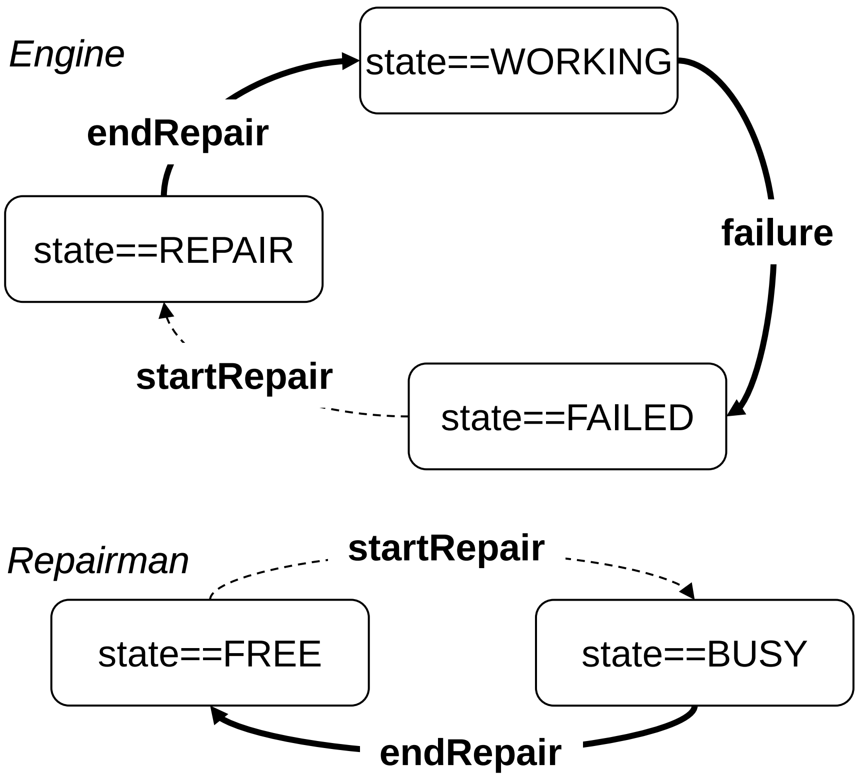
Idea: Basic Events and Gates

- calculate their status (working or failed) bottom-up;
- are activated top-down (in regular Fault Trees, basic events and gates are always active).

GTS generalize (at no cost) Dynamic Fault Trees

GTS versus Petri Nets





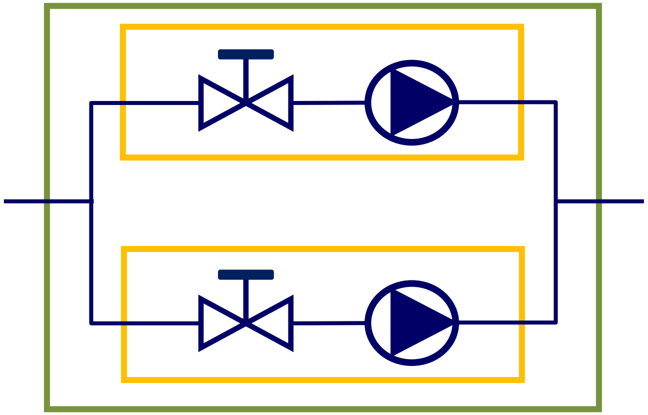
GTS generalize (at no cost) Stochastic Petri Nets (and various extensions of).

Wrap-Up

- Two main mathematical frameworks for risk & safety assessments:
  - Probabilistic Boolean algebra (fault trees)
  - Transitions systems
- Both have advantages and drawbacks
- Guarded Transitions Systems are the most generic framework of the second category

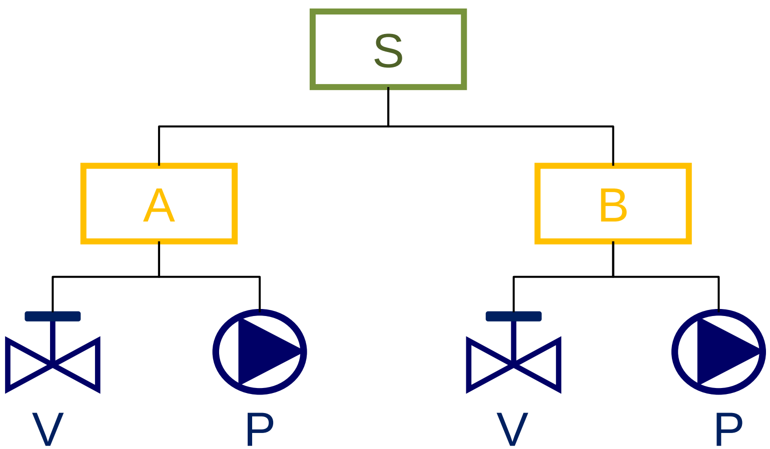
MODEL STRUCTURING FRAMEWORKS

Composition



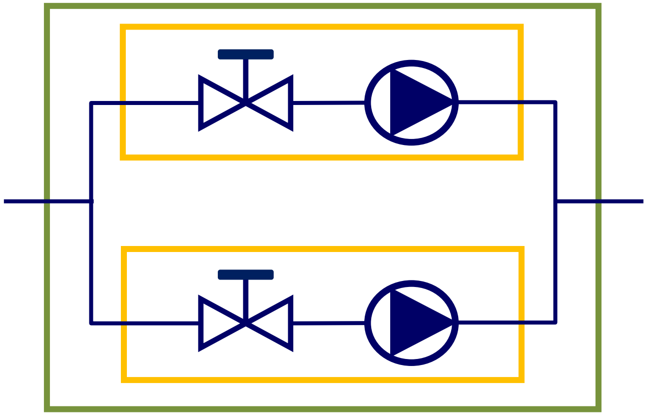
One cannot expect **models of complex systems** to be simple. To capture interesting aspects they have to be complex too, and therefore they **must be structured**.

The simplest structuring relation is the **composition**: a system composes a component means that the component "**is part of**" the system.
Many modeling formalisms implement composition.



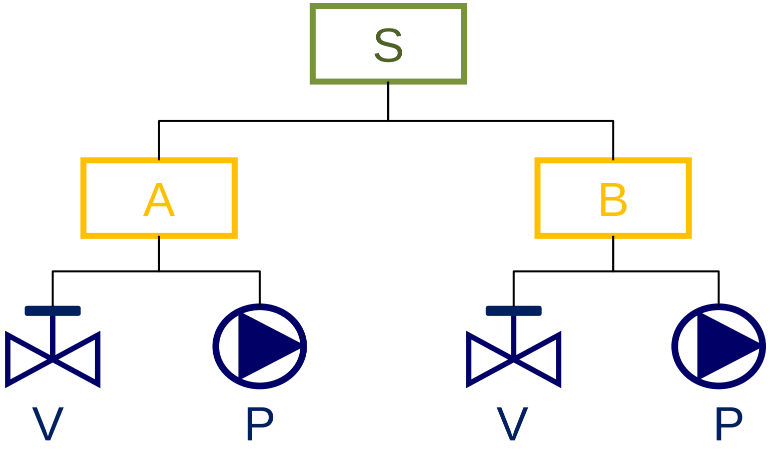
Note: S.A.V is different from S.B.V. although both components are "named" V.

Prototypes

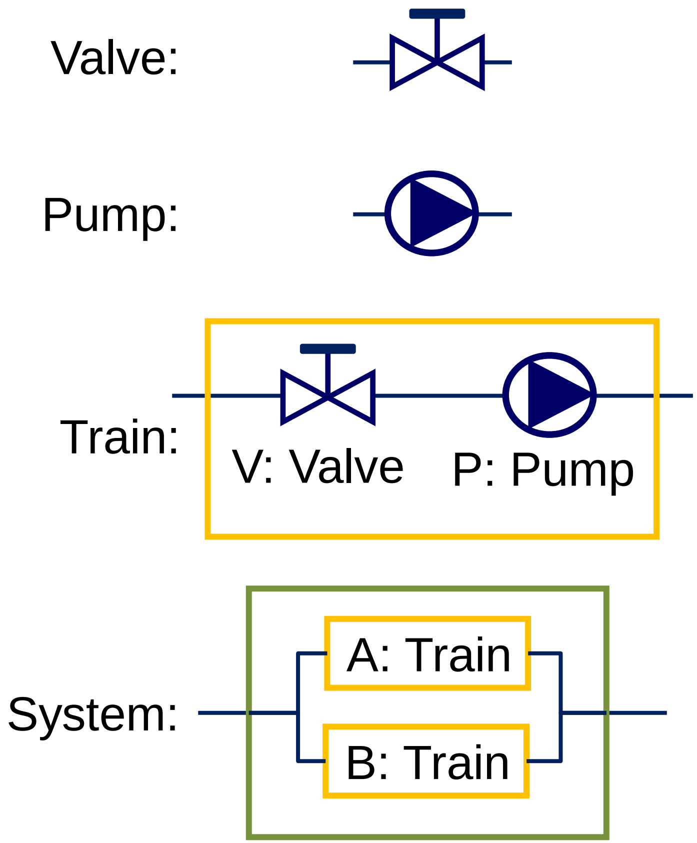


In a hierarchical decomposition, each block (S, S.A, S.A.V…) is supposed to be unique. A block with a unique occurrence is called a **prototype**.

In general, at system level, many blocks are unique.



Classes



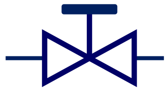
However, it is often the case that components (or even subsystems) are similar (e.g. S.A.V and S.B.V, S.A and S.B). Having only prototypes is not very suitable for **knowledge capitalization and reuse**.

Classes are **on-the-shelf**, **reusable** modeling components. Classes can be **instance**d in a model, e.g. V is an instance of the class Valve in the class Train. An instance of a class is called an **object**.

Several modeling formalisms implement classes, but extremely few both prototypes and classes.

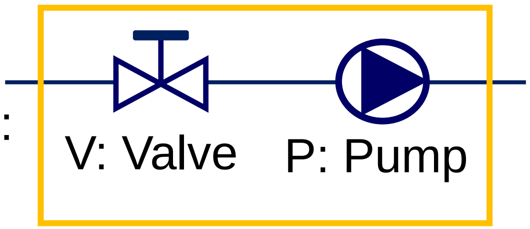
The Box-in-Box-in-Box Issue

Valve:

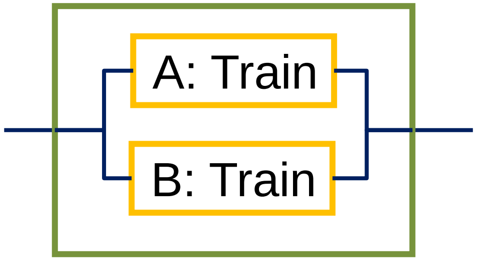


Pump:

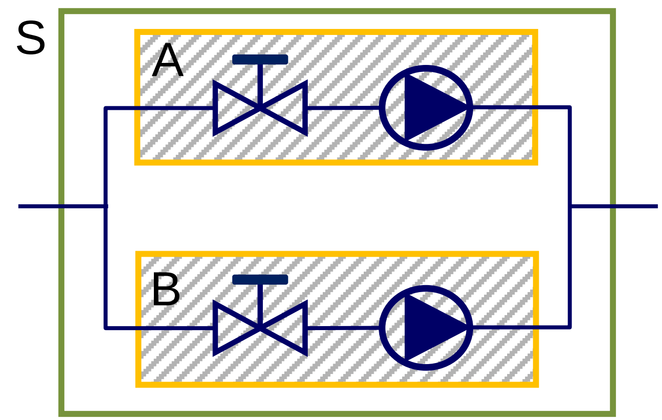
Train:



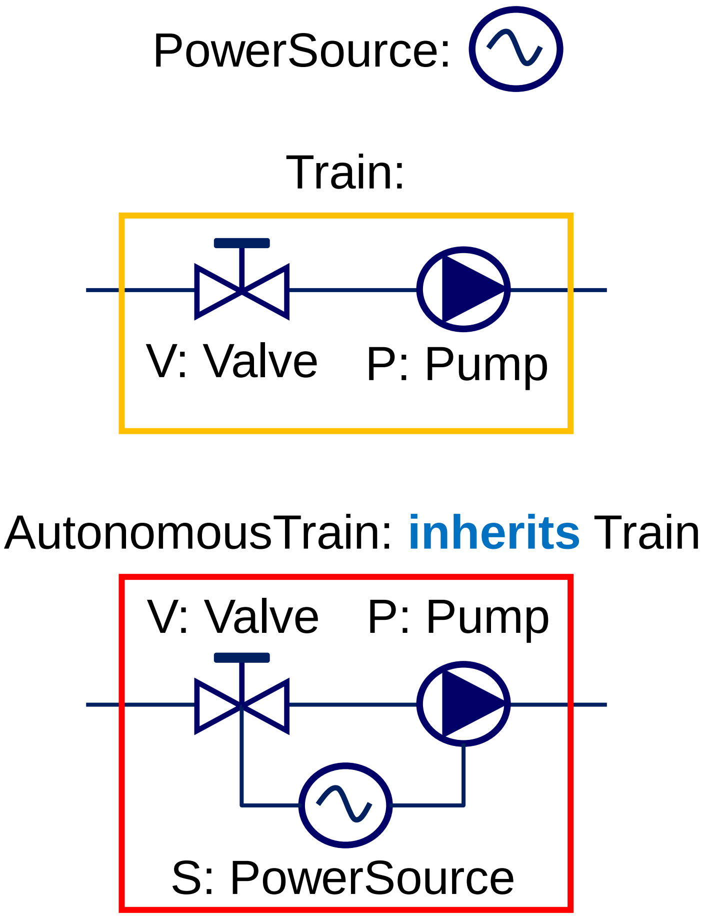
System:



It is not possible to modify a class through its instance, because it would impact not only that particular instance, but all (possibly unknown and even not yet created) instances of the class.



Inheritance

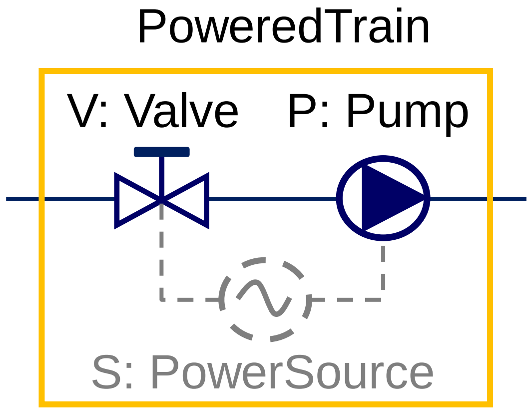


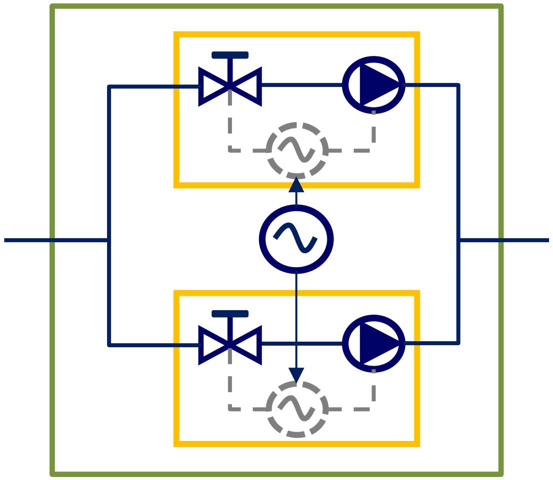
In some cases, we want to modify or extend the characteristics of a modeling component/class without changing its nature. In these cases, composition is not really suitable because we would like to be able to substitute the modified/extended component for any occurrence of the original one.

Inheritance makes it possible.
Inheritance is a “**is-a**” relation between modeling components, e.g. an AutonomousTrain is a Train.

Very few modeling formalisms implement inheritance.

Aggregation





In some cases, we want to capture that a subsystem needs some component, but that this component is not part of the subsystem and may be shared by several subsystems.

Aggregation makes it possible.
Aggregation is a “**uses**” relation between modeling components, e.g. a PoweredTrain aggregates/uses a PowerSource.

Very few modeling formalisms implement aggregation.

Wrap-Up

- Model structuring mechanisms are (almost) independent of behavioral constructs. They originate from mechanisms to structure programs

- **Prototypes**, **classes**, **composition** (**is-part-of** relation), **inheritance** (**is-a** relation) and **aggregation** (**uses** relation) are the fundamental concepts of model structuring.
  - Prototypes + composition: **hierarchical modeling** paradigm.
  - Classes + composition: **structured modeling** paradigm.
  - Classes + composition + inheritance: **object-oriented** paradigm
  - Prototypes + Classes + composition + inheritance + aggregation: **prototype-oriented** paradigm

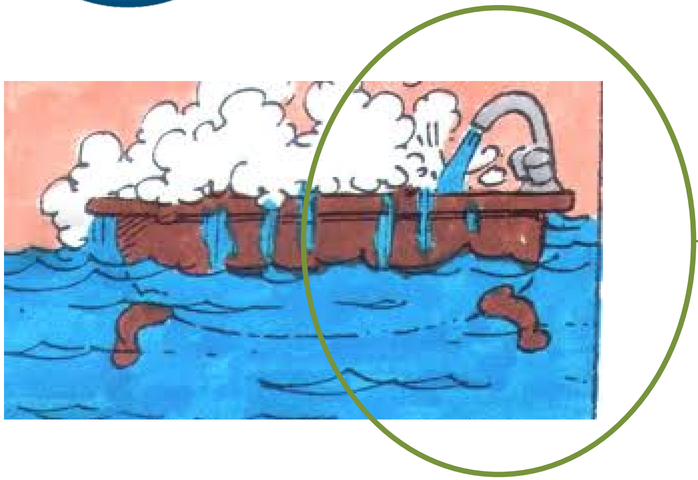
MODEL SYNCHRONIZATION

A Double Challenge

- Systems designed by industry are more and more **complex**.
- To face this complexity, the different engineering disciplines (mechanics, thermic, electric and electronic, software, safety…) virtualized their contents to a large extent, i.e. they are designing **models**. Each system comes with dozens of models.
- There is a here **double challenge**:
  - **Integrating** the different **engineering disciplines**
  - **Integrating** the **models** they produce
- As a consequence, we need to design tools and methods to support this integration.

The emerging science (and engineering) of complex systems is a science (and engineering) of models

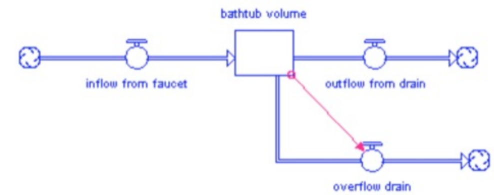
The diversity of models is irreducible



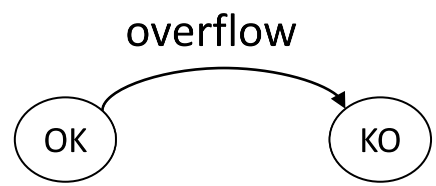
Fluid mechanics

$$\frac{\partial \vec{v}}{\partial t} + \left(\vec{v} \cdot \vec{\nabla}\right)\vec{v} = -\frac{1}{\rho}\vec{\nabla} p + \nu \nabla^2 \vec{v} + \vec{f}$$

Multiphysics simulation



Safety analyses



Insurance

Tableau 1 Effectifs et taux de mortalité standardisés par type d'accident de la vie courante, selon l'âge, France métropolitaine, 2008 (taux pour 100 000 personnes) / Table 1 Number and standardized home and leisure injuries-related death rates by type and age, Metropolitan France, 2008 (rate per 100,000 population)

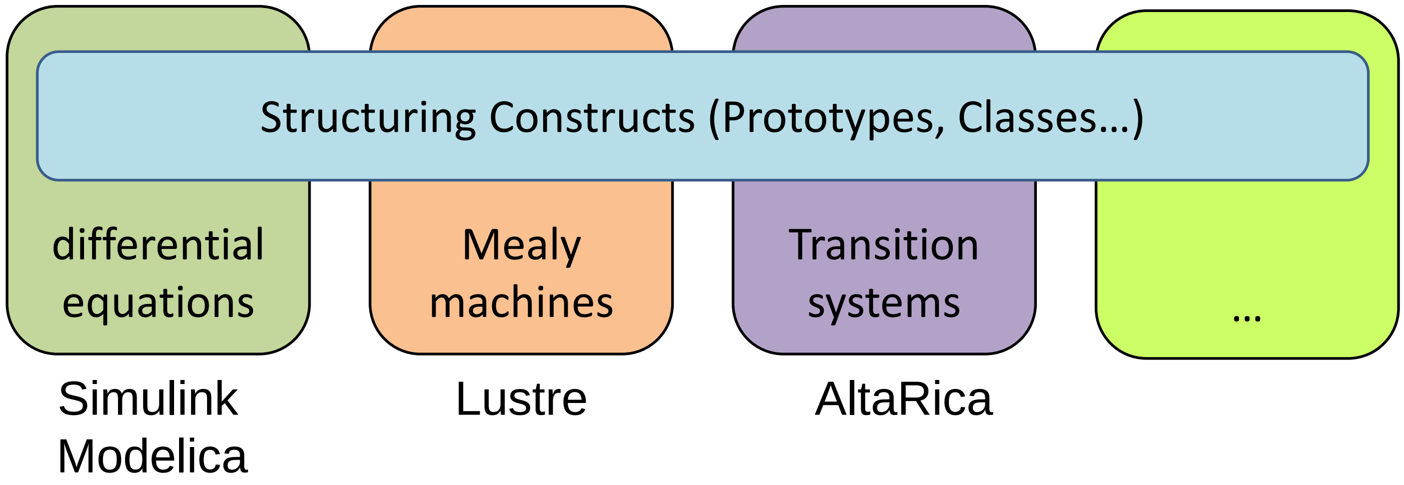
Âge	Chutes		Suffocations		Noyades		Intoxications		Feu		Autres, précisés		Autres, non précisés		Total	
	N	Taux	N	Taux	N	Taux	N	Taux	N	Taux	N	Taux	N	Taux	N	Taux
< 1 an	4	0,51	22	2,79	6	0,76	5	0,64	1	0,13	1	0,13	2	0,25	41	5,21
1-4 ans	19	0,63	18	0,59	31	1,02	2	0,07	24	0,79	4	0,13	13	0,43	111	3,66
5-14 ans	15	0,20	12	0,16	33	0,44	4	0,05	14	0,18	8	0,11	16	0,21	102	1,35
15-24 ans	48	0,61	16	0,20	61	0,78	54	0,69	8	0,10	30	0,38	126	1,60	343	4,36
25-44 ans	229	1,38	123	0,74	174	1,05	350	2,11	86	0,52	109	0,66	300	1,81	1 371	8,26
45-64 ans	741	4,61	433	2,70	352	2,19	347	2,16	143	0,89	214	1,33	660	4,11	2 890	18,0
65-74 ans	660	13,3	323	6,5	152	3,07	126	2,55	70	1,42	91	1,84	423	8,55	1 845	37,3
75-84 ans	2 513	63,2	825	20,7	151	3,80	255	6,41	71	1,79	99	2,49	981	24,7	4 895	123
85 ans et plus	5 183	358	1 227	84,8	68	4,70	233	16,1	59	4,08	58	4,01	1 277	88,3	8 105	560
Total	9 412	11,3	2 999	3,77	1 028	1,50	1 376	1,97	476	0,68	614	0,88	3 798	4,96	19 703	25,1

The **level of abstraction** of a model depends on what is to be observed, i.e. on the **virtual experiments** to be performed on that model.

There cannot be no such a thing as unique model or even a master model of a complex system

Commonalities between models stand in their structuring

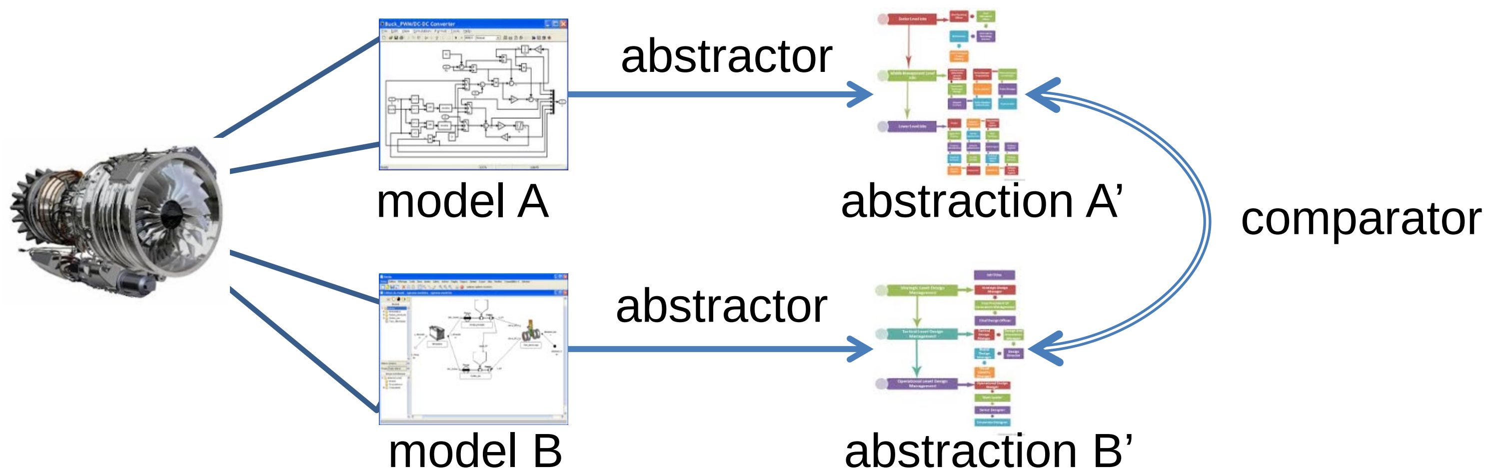
- Any **modeling language** is the **composition** of a **mathematical framework** and a set of **constructs to structure** models.



- The **structure of models** reflects the **structure of the system**, but only to a **limited extent**

Synchronization = Abstraction + Comparison

- The **design/production/operation/decommissioning** of a system involves the design of dozens if not hundred of models. These models are designed by **different teams** in **different languages** at **different levels of abstraction**, for **different purposes.** They have **different maturities**.
- The question is how to **synchronize** these models, i.e. to ensure that they are speaking about the same system.
- **Abstraction** is a key tool for model synchronization.



- The suitable abstractors/comparators depend on the project, phase of the project…

FREQUENTLY ASKED QUESTIONS

What are the tools/languages supporting the MBSA approach?

- AltaRica
  - SimFia (EADS Apsys)
  - Safety Designer (Dassault Systemes)
  - Cecilia-OCAS (Dassault Aviaton, not distributed)
  - OpenAltaRica tools (IRT SystemX & AltaRica Association)
  - ARC/AltaRica Studio (University of Bordeaux)
- Figaro (EdF)
- SAML (University of Magdeburg)
- HIP-HOPS (to some extent) (University of Hull)
- SOFIA (to some extent) (CEA-LIST)
- Petro (specific to Oil & Gas) (SATODEV)

How mature is the MBSA approach?

	helpful	harmful
internal origin	• Theoretical framework • High Level Models are much easier to design, to debug, to master, to maintain, to share, to reuse… • Generalization of “classical” formalisms such as Block Diagrams, Markov chains, Generalized Stochastic Petri Nets • Richness of assessment algorithms	• Trend to design too big and unique models • Difficulty to handle systems whose architecture changes during the mission • Initial cost to train analysts
external origin	• Significant audience in France • Certification process accepted by FAA and EASA (Dassault F7X), mentioned in last version of ARP4761 • Graphical simulation • Used beyond safety analyses (performance analysis)	• Development costs • Redundant developments

References

DASSAULT AVIATION

apsys SIMPLIFYING COMPLEXITY

AIRBUS AN EADS COMPANY

Snecma SAFRAN Group

SNCF

THALES

DGA

PSA PEUGEOT CITROËN

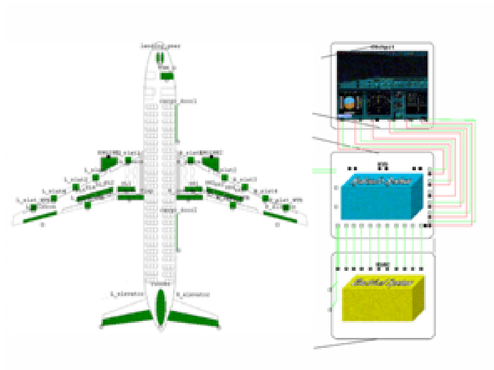
ALSTOM

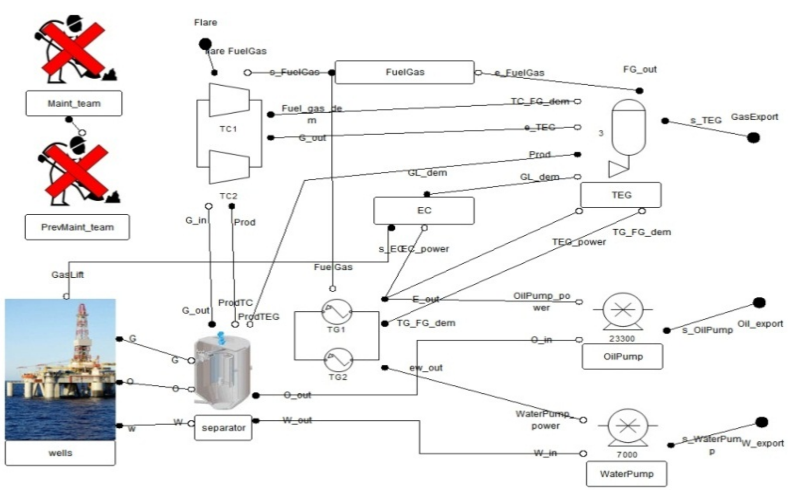
SAGEM

RENAULT

cnes

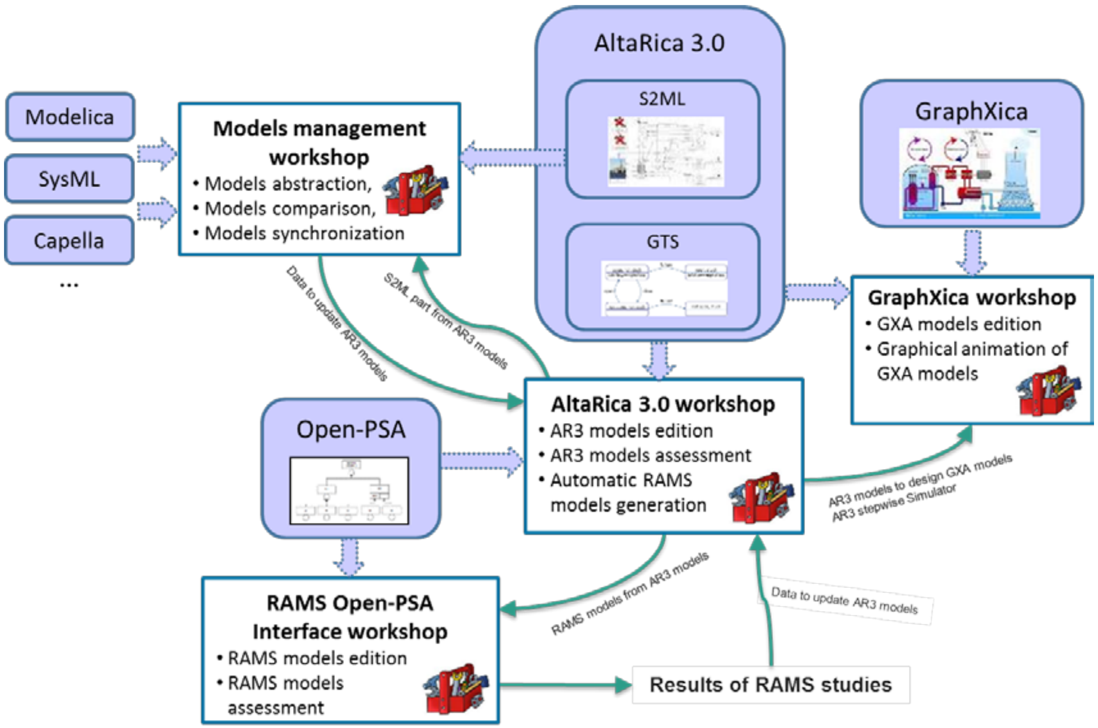
DCNS





Is the AltaRica project active?

- Yes! The OpenAltaRica project



www.openaltarica.fr www.altarica-association.org

Is there a conference dedicated on MBSA?

- Yes!

Next International Conference on Model-Based Safety Assessment, IMBSA 2017, will be collocated with SAFECOMP 2017 in Trento (Italy)

SOME REFERENCES

Some References

Algorithms for Fault Tree assessment

- Antoine Rauzy. Mathematical Foundation of Minimal Cutsets. IEEE Transactions on Reliability. IEEE Reliability Society. 50:4. pp. 389–396. december, 2001.doi:10.1109/24.983400.
- Antoine Rauzy, BDD for Reliability Studies, in *Handbook of Performability Engineering*, pages 381-396. K.B. Misra ed., Elsevier, ISBN 978-1-84800-130-5, 2008
- Antoine Rauzy, XFTA: Pour que cent arbres de défaillance fleurissent au printemps, In J.F. Barbet ed., *Actes du Congrès Lambda-Mu 18*. Octobre, 2012.
- Yves Dutuit and Antoine Rauzy. Importance Factors of Coherent Systems: a Review. Journal of Risk and Reliability. Professional Engineering Publishing. 228:3. pp. 313–323. December, 2013. doi:10.1177/1748006X13512296.

Algorithms for Markov chains assessment

- William J. Stewart. Introduction to the Numerical Solution of Markov Chains. Princeton University Press. Princeton, New Jersey, USA. ISBN 978-0691036991. 1994.
- Antoine Rauzy. An Experimental Study on Six Algorithms to Compute Transient Solutions of Large Markov Systems. Reliability Engineering and System Safety. Elsevier. 86:1. pp. 105–115. October, 2004. doi:10.1016/j.ress.2004.01.007.

Stochastic Simulation for RAMS Studies

- Enrico Zio. The Monte Carlo Simulation Method for System Reliability and Risk Analysis. Springer London. London, England. ISBN 978-1-4471-4587-5. 2013.

Some References

Dynamic Fault Trees

- Joanne Bechta Dugan, Salvatore J. Bavuso and Marks A. Boyd. Dynamic fault-tree models for fault-tolerant computer systems. IEEE Transactions on Reliability. IEEE. 41:3. pp. 363–377. September, 1992. doi:10.1109/24.159800.
- Marc Bouissou and Jean-Louis Bon. A new formalism that combines advantages of Fault-Trees and Markov models: Boolean logic-Driven Markov Processes. Reliability Engineering and System Safety. Elsevier. 82:2. pp. 149–163. 2003. doi:10.1016/S0951-8320(03)00143-1.
- Antoine Rauzy. Towards a Sound Semantics for Dynamic Fault Trees. Reliability Engineering and System Safety. Elsevier. 142. pp. 184–191. October, 2015. doi:10.1016/j.ress.2015.04.017.

“Model-Based” extensions of Stochastic Petri Nets

- Yves Dutuit, Eric Châtelet, Jean-Pierre Signoret and Philippe Thomas. Dependability modeling and evaluation by using stochastic Petri nets: application to two test cases. Reliability Engineering and System Safety. Elsevier. 55:2. pp. 117–124. 1997. doi:10.1016/S0951-8320(96)00108-1.
- Jean-Pierre Signoret, Yves Dutuit, Jean-Pierre Cacheux, Cyrille Folleau, Stéphane Collas and Philippe Thomas. Make your Petri nets understandable: Reliability block diagrams driven Petri nets. Reliability Engineering and System Safety. Elsevier. 113. pp. 61–75. 2013. doi:10.1016/j.ress.2012.12.008.

Some References

AltaRica Foundations & Specifications

- Antoine Rauzy. Guarded Transition Systems: a new States/Events Formalism for Reliability Studies. Journal of Risk and Reliability. Professional Engineering Publishing. 222:4. pp. 495–505. 2008. doi:10.1243/1748006XJRR177.
- Antoine Rauzy. AltaRica 3.0 Specifications. Working Document (on demand).

Tutorial examples of AltaRica

- Marie Boiteau, Yves Dutuit, Antoine Rauzy and Jean-Pierre Signoret. The AltaRica Data-Flow Language in Use: Assessment of Production Availability of a MultiStates System. Reliability Engineering and System Safety. Elsevier. 91:7. pp. 747–755. July, 2006. doi:10.1016/j.ress.2004.12.004.
- Frédéric Milcent, Tatiana Prosvirnova and Antoine Rauzy. Modélisation des réseaux en AltaRica 3.0. Actes du congrès Lambda-Mu 19 (actes électroniques). Institut pour la Maîtrise des Risques. ISBN 978-2-35147-037-4. Dijon, France. October, 2014.
- Hala Mortada, Tatiana Prosvirnova and Antoine Rauzy. Safety Assessment of an Electrical System. Proceedings of the 4th International Symposium on Model-Based Safety Assessment, IMBSA 2014. Springer Verlag. 8822. pp. 181–194. Munich, Germany. October, 2014.
- Abraham Cherfi, Michel Leeman and Antoine Rauzy. AltaRica 3.0 Based Models for ISO 26262 Automotive Safety Mechanisms. Proceedings of the 4th International Symposium on Model-Based Safety Assessment, IMBSA 2014. Springer Verlag. 8822. pp. 123–136. Munich, Germany. October, 2014.

Some References

Compilation of AltaRica (GTS) into Fault Trees

- Antoine Rauzy. Modes Automata and their Compilation into Fault Trees. Reliability Engineering and System Safety. Elsevier. 78:1. pp. 1–12. October, 2002.doi:10.1016/S0951-8320(02)00042-X.
- Tatiana Prosvirnova and Antoine Rauzy. Automated generation of Minimal Cutsets from AltaRica 3.0 models. International Journal of Critical Computer-Based Systems. Inderscience Publishers. 6:1. pp. 50–79. 2015.doi:10.1504/IJCCBS.2015.068852.

Compilation of AltaRica (GTS) into Markov chains

- Pierre-Antoine Brameret, Antoine Rauzy and Jean-Marc Roussel. Automated generation of partial Markov chain from high level descriptions. Reliability Engineering and System Safety. Elsevier. 139. pp. 179–187. July, 2015. doi:10.1016/j.ress.2015.02.009.

Some References

Some selected PhD Theses related to AltaRica (mostly in French)

- Gérald Point. AltaRica: Contribution à l'unification des méthodes formelles et de la sûreté de fonctionnement. LaBRI -- Université Bordeaux I. Janvier, 2000.
- Aymeric Vincent. Conception et réalisation d'un vérificateur de modèles AltaRica. Université de Bordeaux. December, 2003.
- Claire Pagetti. Extension temps réel du langage AltaRica. École Centrale de Nantes et de l'Université de Nantes. 2004.
- Christophe Kehren. Motifs formels d'architectures de systèmes pour la sûreté de fonctionnement. Ecole Nationale Supérieure de l'Aéronautique et de l'Espace (SUPAERO). Toulouse, France. 2005.
- Sophie Humbert. Déclinaison d'exigences de sécurité, du niveau système vers le niveau logiciel, assistée par des modèles formels. Université de Bordeaux I. April, 2008.
- Laurent Sagaspe. Allocation sûre dans les systèmes aéronautiques : modèlisation, vérification et génération. Université de Bordeaux. December, 2008.
- Minh Thang Khuu. Contribution à l'accélération de la simulation stochastique sur des modèles AltaRica Data Flow. Université de la Méditerranée (Aix-Marseille II). 2008.
- Romain Bernard. Analyse de Sécurité multi-systèmes. Université de Bordeaux. November, 2009.
- Tatiana Prosvirnova. AltaRica 3.0: a Model-Based Approach for Safety Analyses. École Polytechnique. Palaiseau, France. November, 2014.
- Pierre-Antoine Brameret. Assessment of Reliability Indicators From Automatically Generated Partial Markov Chains. Thèse de l'ENS Cachan, July 2015

Some References

Additional References on Model-Based Safety Assessment

- Marc Bouissou, Henri Bouhadana, Marc Bannelier and Nathalie Villatte. Knowledge modelling and reliability processing: presentation of the FIGARO language and of associated tools. Proceedings of SAFECOMP'91 -- IFAC International Conference on Safety of Computer Control Systems. Johan F. Lindeberg Ed.. Pergamon Press. ISBN 0-08-041697-7. pp. 69–75. Trondheim, Norway. 1991.
- Marc Bouissou and Jean-Christophe Houdebine. Inconsistency Detection in KB3 Models. Proceedings of European Safety and Reliability conference, ESREL'2002. ISdF. pp. 754–759. Lyon, France. March, 2002.
- Marc Bouissou, Sibylle Humbert, Sabine Muffat and Nathalie Villatte. KB3 tool: feedback on knowledge bases. Proceedings of European Safety and Reliability conference, ESREL'2002. ISdF. pp. 114–119. Lyon, France. March, 2002.
- Marco Bozzano and Adolfo Villafiorita. Design and safety assessment of critical systems. Auerbach Publications (Tailor & Francis Group). Boca Raton, FL, USA. ISBN 978-1-439-80331-8. 2011.
- Matthias Güdemann and Frank Ortmeier. A Framework for Qualitative and Quantitative Model-Based Safety Analysis. Proceedings of the IEEE 12th High Assurance System Engineering Symposium (HASE 2010). IEEE. ISBN ISBN 978-1-4244-9091-2. pp. 132–141. San Jose, CA, USA. 2010.doi:10.1109/HASE.2010.24.
- Masakazu Adachi, Yiannis Papadopoulos, Septavera Sharvia, David Parker and Tetsuya Tohdo. An approach to optimization of fault tolerant architectures using HiP-HOPS. Software Practice and Experience. Wiley Inderscience. 41. pp. 1303–1327. 2011.doi:10.1002/spe.1044.
- Papadopoulos Y., Walker M., Parker D., Rüde E., Hamann R., Uhlig A., Grätz U., Lien R. (2011) Engineering Failure Analysis & Design Optimisation with HiP-HOPS, Journal of Engineering Failure Analysis, DOI: 10.1016/j.engfailanal.2010.09.025, Elsevier Science, ISSN: 1350-6307

Some References

Paradigms to structure programs/models

- Mauricio Abadi and Luca Cardelli. *A Theory of Objects*. Springer-Verlag. Monographs in Computer Science. New York Inc. ISBN-10: 0387947752, ISBN-13: 978-0387947754, 1998
- James Noble, Antero Taivalsaari and Ivan Moore. *Prototype-Based Programming: Concepts, Languages and Applications*. Springer-Verlag Berlin and Heidelberg GmbH & Co. K. May. ISBN-10: 9814021253. ISBN-13: 978-9814021258, 1999
- Michel Batteux, Tatiana Prosvirnova and Antoine Rauzy. System Structure Modeling Language (S2ML). AltaRica Association. 2015. archive hal-01234903, version 1.

Graphical Modeling & Architecture Modeling Formalisms

- Sanford Friedenthal, Alan Moore and Rick Steiner. A Practical Guide to SysML: The Systems Modeling Language. Morgan Kaufmann. The MK/OMG Press. San Francisco, CA 94104, USA. ISBN 978-0123852069. 2011.
- Hauke A. L. Fuhrmann. On the Pragmatics of Graphical Modeling. Book on Demand. Norderstedt, Germany. ISBN 978-3844800845. 2011.