Towards Rebalancing Safety Design, Assessment and Assurance

Emmanuel Ledinot, Jean-Paul Blanquart, Jean Gassino, Rémy Astier, Philippe Baufreton, Jean-Louis Boulanger, Jean Louis Camus, Cyrille Comar, Philippe Quéré, Bertrand Ricque

Towards Rebalancing
Safety Design, Assessment and Assurance

Emmanuel Ledinot[(1)] Jean-Paul Blanquart[(2)], Jean Gassino[(3)],
Rémy Astier[(4)], Philippe Baufreton[(5)], Jean-Louis Boulanger[(6)],
Jean Louis Camus[(7)], Cyrille Comar[(8)], Philippe Quéré[(9)], Bertrand Ricque[(5)]

(1, 2, 3, 4, 5, 6, 7, 8, 9) Working Group "Safety Standards" – Embedded France
(1): Contact author, THALES, emmanuel.ledinot@thalesgroup.com
(2): Airbus Defence and Space; (3): Institut de Radioprotection et de Sûreté Nucléaire;
(4): Rolls-Royce Civil Nuclear; (5) Safran; (6) CERTIFER;
(7): ANSYS-Esterel Technologies; (8): AdaCore; (9): Renault

Abstract

Cyber-physical systems have evolved faster than development technologies, which in turn have evolved faster than safety standards, despite periodic revisions. By 2020, a significant cumulative gap exists between development assurance and its perceived effectiveness on safety of the highly complex systems developed nowadays. This paper explores how this gap could be at least partly closed. First, we review new techniques that are emerging from hybrid system research and that might influence verification of system safety in the future, then we discuss some problems in industrial practice of safety assessment and in safety standards. These problems are widely acknowledged in all industrial domains, especially when facing certification of AI-enabled autonomous vehicles (cars, drones, trains, underwater unmanned vehicles etc.). Finally, we propose some orientations to evolve the development assurance standards so that they may facilitate accommodation of these new techniques without *adding* new assurance requirements to the legacy ones. We advocate a new balance for future assurance that would introduce new structural and behavioural analyses while reducing some aspects of dysfunctional analysis.

Keywords: hybrid systems, CPS engineering, controllability, fault-tolerance, formal verification of systems, safety assurance.

1. Introduction

Embedded France is an initiative launched by major French industrial companies involved in the development of critical embedded systems in a wide spectrum of application domains. Its objective is to improve its members' capabilities to meet the major challenges of the development of embedded systems, in particular software-intensive safety critical embedded systems. It elaborates propositions, recommendations, roadmaps etc. based on collaborative work and discussions in dedicated thematic Working Groups.

One of these Working Groups is dedicated to safety standards[1] and gathers industrial safety experts in as many domains as automotive, aviation, defence, industrial processes, nuclear, railway and space. Some representatives of technology providers are also members of the group, the objectives of which are to identify the main similarities and dissimilarities between safety standards, with in perspective a potential cross-domain harmonization, when possible and relevant. This paper continues a series of publications from the Working Group, through which its members disseminate and encourage feedback about their work [WG-2010] [WG-2012a, WG-2012b, WG-2012c], [WG-2014] [WG-2016a; WG-2016b] [WG 2018a, WG-2018b].

This 2020 edition addresses a growing cross-domain concern. Development assurance as practiced today becomes less and less tractable and adequate when applied to the new types of systems that are being engineered today: smart cities, smart grids, smart cars, smart eHealth, industry 4.0, or autonomous vehicles (drones, cars, trains, tramways, trucks, etc.). All raise two issues that challenge the very principles of development assurance based on fault prevention through process-conformity to standards. First, most often, these systems integrate many legacy components or systems. Development process minute planning and monitoring becomes ineffective on large parts of the product tree and of the lifecycle. Second, their behavioural complexity is becoming daunting, up to challenging the very notion of verification coverage, a cornerstone of development assurance.

Note on terminology: The word "safety" is used in this paper following its definition, consistent with most safety standards' ones, as "freedom from

[1] The Working Group "Safety Standards" was created in 2009 and initially attached to the "Club des Grandes Entreprises de l'Embarqué (CG2E)".

unacceptable risks". Even though the safety assessment may be and is often supported by probabilistic measures (typically regarding random hardware failures), it must be understood that this can only be a part of the safety assessment, complemented by deterministic qualitative arguments. This is particularly true for what concerns software.

In the sequel we adopt the *system theoretic* approach to safety as exemplified by STAMP/STPA [Lev12]. Safety design and assessment is addressed as a problem of *controllability under perturbations*. The perturbation domains are numerous: adverse environmental conditions, physical breakdowns of components (random failures) followed by cascading effects, activation of residual development faults (systematic failures), errors in operating procedures (human factor), cyber-attack exploits, not to forget the well-known "unknown unknowns".

The paper's rationale is the following: first we review some promising perspectives on system engineering with potential positive impact on future design and assessment of safety. Then we review some safety assessment activities whose benefit-to-cost ratio is more and more questionable on the class of systems we have mentioned. Finally we sketch some orientations for evolving the future safety assurance standards, in order to introduce the new approaches while downsizing the assurance activities deemed of low-added value.

2. Evolution of CPS engineering

In this section we review some research fields that have slowly got a significant level of maturity (e.g., formal verification of hybrid systems) or that are evolving fast because of the AI Machine Learning race. All these fields are deemed with potential positive impact on safety design and verification, and in fine with potential influence on evolution of safety assurance standards. From a development process perspective, we mainly address conceptual design, model-based design, implementation and verification of cyber-physical systems (CPSs). We define CPSs as being hybrid systems [Pla18], either closed (static structures e.g., systems embedded into vehicles) or open (dynamic structures e.g., smart cities, smart grids, swarms of autonomous vehicles, etc.).

2.1. Structural analysis of hybrid systems

Structural analysis has emerged at MIT in the 1950s to unify modelling and analysis of engineering systems that can be represented as standardised networks of power lines and generalised impedances [Payn61]. It is mainly known as bond-graphs and used to study causality in multi-physics energy circuits [Kar90]. The Modelica® language has been designed as an extension of bond graphs. Structural analysis identifies the structure of the variable-to-variable dependencies in the CPS equation-based model. It determines its physical and informational *influence network*[2].

Computed by the numerical solvers of multi-physics multi-system modelling tools like Dymola/Modelica® or Simscape/Simulink®, the dependency graph of a model has many by-product applications, including separation analysis, inversion analysis, simulation parallelization, or synthesis of failure detectors and diagnostic logics [Fri17]. For software, structural analysis (control flow graphs, data-flow graphs, forward and backward influence cones, etc.) is performed by source code or binary code abstract interpreters.

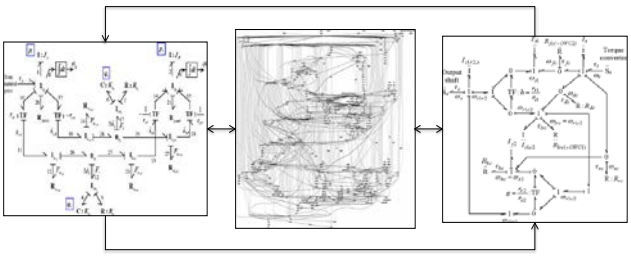

Figure 1: Notional figure illustrating the interaction structure of a piece of software controlling two physical systems represented as bond graphs

Structural analysis scales-up on large system and software models: no or limited curse of dimensionality, no combinatorial explosion. It is a *static* graph theoretic notion. It provides support to *cascading effect analysis* and substantiation of *separation* between functions (system specification), between resource components (system implementation), or between *containment regions* (FDIR[3] design).

Separation arguments are used to prove independence between redundancies which in turn provides means to demonstrate the "no k-point(s) catastrophic failure" objectives. We may encounter k=2 for space, but for most domains k=1 ("no single-point catastrophic failure").

Current safety assessment practice does not resort to structural analysis on the MBSE[4] models. Identification of the physical and informational dependency network for failure propagation is done by independent safety engineers, using FMEA

[2] Save field physics defined by Partial Differential Equations (e.g., heat transfers, EM waves, mechanical shock waves, etc.). They can only be represented in a very limited approximate way.
[3] Fault Detection Isolation and Recovery
[4] Model-Based System Engineering

(single event *forward* failure propagation), FTA (top-down, *backward* dysfunctional modelling) or MBSA[5] (bottom-up, *forward* dysfunctional modelling) [Rau17]. Forward means "from cause to effect", also named *direct* mode in dynamic system simulation and analysis. Backward means "from effect to cause", also named *inverse* mode[6].

2.2. Formal verification of hybrid systems

At present, hybrid system models are mainly used for functional simulation, for uncertainty propagation (robustness analysis), for performance and dimensioning optimisation, and for control engineering possibly including code generation. New analyses, inspired by formal methods at software level (abstract interpretation), are emerging for hybrid systems.

Accessibility analysis and uncertainty propagation. Model-checking is now possible for some classes of hybrid system models [Fre11], [Bel17], [Gou17]. Roughly speaking numerical integration of ODEs[7] and DAEs[8] has been extended to intervals or to more sophisticated representations of behaviour sets (zonotopes, support functions etc.). A (conservative) envelope of the infinitely many simulation scenarii is computed as flow pipes and accessibility spaces. Exhaustive[9] verification has stepped in the catalog of system engineering tools with potential impact on future safety assurance. Scalability is already industrially meaningful, though still moderate (e.g., a few hundreds of state variables in the affine case).

Finite-state abstraction. Predicate abstraction on software has been extended to hybrid systems. Given a set of state predicates it is possible to compute the finite state automaton that faithfully summarises the infinitely many behaviours of the continuous-state model. The abstract states of the "summary" group the states of the concrete model that give the same value to the state predicates (equivalence classes w.r.t. the observation criterion). Function losses, resource losses, and tilts of FDIR detectors can be defined as state predicates. Our conjecture is that this abstract interpretation machinery may be an option to solve the consistency problem between MBSA and MBSE models. Proving soundness of the MBSA abstractions w.r.t. the MBSE models, an unsolved issue with fault-trees and hand-made MBSA models, may become tractable, at least at small and medium scale.

Timed hybrid system abstraction[10] is also possible [Slo13], [Mov13], [Bou15]. Time-aware finite state abstractions would make a significant difference to reconcile fidelity (w.r.t. physics and software[11]) and computational efficiency in combinatorial dysfunctional analysis: the enumeration of the uncontrollable configurations whose occurrence probabilities have to be summed together.

Geometrical analysis of behavioural spaces. Accessibility analysis of hybrid systems compute over-approximated behaviour envelopes. It does not handle the true shape of the reachable state space as a geometric object. Thanks to recent progress in n-dimensional computational geometry [Cha17], [Boi18], there are perspectives to mesh the boundary of the reachable state space, and then to compute its geometrical and topological properties like volume, 2D or 3D projections, intersection with "stay-out" regions etc. In this context, the safety properties would no longer be handled only as (temporal) logical formulas or observers (intensional definitions), they would also become *shapes* in the behavioural space (extensional definitions), giving access to measurement of *behavioural coverage* at system level. It may also enable visual reviews of the safety state invariants in addition to formal verification of their logical formulation.

2.3. Other emerging techniques

We just list some of them; all are deemed to have positive potential impact on future safety assessment and possibly on future assurance standards.

Contract-based design [Bog14] [Ben18], *invariant-based design* [Bac09], and compositional verification [Slo12] [Wil16] have been extended from software to hybrid systems.

Model-based virtual sensing and *dynamics learning* in operation supported by embedded massive data recorders and cloud-based Big Data analytics open

[5] Model-Based Safety Assessment

[6] Note that ARP 4761A, like some other standards and many papers on dysfunctional modelling, use *deductive-inductive,* in place of forward-backward or direct-inverse, which is confusing and inappropriate. Deductive and inductive both qualify logical *inference modes* (respectively "from the general to the particular" and "from the particular to the general"). Causality analysis i.e., cause-effect *ordering* of physical or informational events is *not* a matter of logical inference.

[7] Ordinary Differential Equations

[8] Differential Algebraic Equations

[9] Warning: "exhaustive verification" does not imply exhaustive identification of the properties to be verified, exhaustiveness applies to behavioural space exploration.

[10] Adding "stay-in"(min, max) duration intervals to the states of the automata

[11] More precisely the system model of control specification allocated to software

interesting perspectives to augment *observability* of systems without adding sensors that impede reliability and costs. As a consequence it will improve *detection coverage* of FDIR detectors (also named safety monitors, safety nets, safety bags etc.), a critical aspect of functional safety design.

Search-based testing, powered by AI techniques (evolutionary optimisation) or cyber-security techniques (fuzzers) open interesting perspectives to explore more extensively the behavioural space of a system model or of a software component.

On the probabilistic side there are methods to address *quantification on high fidelity models* [Mor16], as opposed to the ubiquitous and debatable use of fault-tree abstractions. These methods use accelerated Monte Carlo estimations on simulation-based experiments. Model abstraction and reduced-order models are additional means to speed-up predicate evaluation when hundreds of thousands of simulation runs are needed. The MBSA-MBSE relationship could be revisited along these lines, to perform safety assessment on the system engineering models, ensuring de facto *consistency between safety design and safety assessment*.

3. Revisiting dysfunctional analysis

We would like to point to some aspects of dysfunctional analysis that are common to the standards of all domains, and that we think may constitute a rare opportunity for economically significant downsizing of development assurance costs. This downsizing is envisioned and advocated as an enabler for introducing new assurance goals without increasing the overall cost of safety assurance.

In other words we are looking for ways of *substituting* low added-value assurance goals by new high value ones instead of *adding* assurance goals, up to saturation of applicants. The “snowball” or “Swiss cheese” policies are no longer sustainable to introduce innovation in development processes and in the associated assurance processes. This section points to a candidate area for downsizing: excess in enumerative dysfunctional analysis.

Here is the difficulty common to all domains: safety design needs to define the ‘hazardous behaviours’ and the ‘accidents’ at top level. Then it has to identify what kind of causes could lead to such harmful deviations from normality. This is an analysis from the global effects to their local causes when done by FTA, and from the local causes to their global effects when done by FMEA.

In the inverse mode case i.e., from global effects to local causes, from the outer to the inner, guessing the behaviour of the interacting components from the result of that interaction may be unsolvable. One faces some sort of “inverse dynamics” problem, though different from classical *dynamic inversion* as used in robotics with acausal[12] models: given a specified trajectory of the outputs (e.g., the kinematics of the robot’s arm), an acausal solver enables computation of the inputs (the commands to the electrical engines) that once submitted to the model will generate the specified outputs. An acausal model enables computing the causes (inputs) from the specified effects (outputs) *if* the inverse problem is *well-posed*. Most often it is not. Intractability is even worse to invert a *dysfunctional* behaviour because we do not have the laws of the dysfunctional, or we have only the part that has been anticipated.

However, all safety standards recommend extensive activities supported by FTA and FMEA, the aim of which is to *exhaustively enumerate* the possible causes of the anticipated hazards and accidents.

3.1. Dubious inverse dysfunctional abstract interpretation of hybrid systems

FTA is an *inversion-dependent* analysis method (derivation of causes from effects). It plays a key role in *all* safety standards. In the aeronautic domain for instance it is the only method explicitly mentioned by ARP 4754A to elaborate the pivotal notion of Functional Failure Sets (FFS) that enables derivation of nothing less than DAL assignment and detection of single point failure in architectures [ED79A/ARP4754A], [ED135/ARP4761].

Even with the future methods listed in the previous section, all of which analyse hybrid system dynamics in *direct* mode, there is no hope to compute what is *implicitly* targeted by safety standards through recommendation of Fault-Tree Analysis: *completeness* of identification of hazard causes.

This paper would like to underline a subtle drift over time that has had significant impact on what we perceive as effort waste and barrier to future certification reformation. Yes FTA makes sense w.r.t. the completeness goal in the purely *structural case* of the 70s. The Boolean transcription of the cascading effects (function or resource losses) through a static serial-parallel dependency network is rigorous. Completeness of cause combinations may be ensured. At that time there was nearly no software in systems, no issue of inverse abstract interpretation of dysfunctional behaviours. But smoothly and progressively, generalising “by analogy” FTA to software, to entire software-intensive systems, an ever increasing amount of

[12] i.e., relational instead of functional equation-based models (Modelica®, Simscape™, etc.)

human-brained *inverse* qualitative-physics reasoning and *inverse* abstract interpretation of system specifications have been silently added in FTA and backed by Authorities. Validity of this behavioural extension of FTA is debatable regarding both soundness and completeness. Soundness can be tested on single cause failures on the final product, or on a "digital twin" high fidelity model. But double and triple cause failures can't be tested as soon as failure modes account for hundreds or thousands which is common place with large systems. As to completeness i.e., exhaustiveness of cause identification, it is *never* testable. Even reducing the ambition to the *anticipated* failure initiators, testing completeness may be intractable because of infeasibility of dynamic inversion i.e., computing all the failure paths that lead to the upper level deviations of interest.

3.2. Delusive completeness of dysfunctional inventories

There is value in considering a sufficient number of different dysfunctional events and scenarii, and preferably the worst ones:

1. to challenge the design of the "stay-in regions" (named 'safety constraints' in [Lev12]),
2. to challenge the design of the FDIR detectors,
3. to verify the FDIR containment regions,
4. and last but not least to challenge the robustness of the safety controls.

But once safety control design is completed (named 'green engineering' in table 1 below), what is the added-value of costly endless decompositions of hazardous events (FTAs), or endless enumerations of single cause cascading effects (FMEAs)? The higher the complexity of the system, the more dubious the value of these inventories. They are exhausting, not exhaustive.

3.3. Fault tolerance at the heart of a new methodological balance?

Beware. We do *not* intend to discredit FMEA and FTA, nor MBSA (grouped under the name of 'red engineering' in table 1). We only question the silent shift from the structural to the behavioural when putative inverse dysfunctional dynamics is at work. We suggest that overconfidence in preservation of completeness of failure cause identification during this shift has led to over-expenditure not on par with value for safety.

The unquestionable value of the various dysfunctional analyses is their contribution to fail-safe design as a controllability problem, and more specifically to the specification of the "stay-in" regions and to identification of the perturbation classes. Fault tolerance mechanisms extend the controllability domain. They fundamentally depend on defining the frontier between the normal and the abnormal, which in turn implies some principled exploration of the abnormal.

What matters for safety control is not enumeration of the failure paths end to end[13]. It is:

1. valid definition of the "stay-in" regions,
2. correct control laws (functional safety) to satisfy these invariants under perturbations that stay within the controllability domain defined by the FDIR detectors,
3. correctness of the decision logics that define the controllability conditions (tolerated perturbation domains, FDIR detectors),
4. correctness of the *stability preserving* FDIR recovery mechanisms that isolate the failed containment regions and activate the new independent ones.

Safety assessment and safety assurance standards do not formulate explicit goals on 3) and 4), especially on analysis of the risks of false positives and false negatives in 3), a critical issue with first order impact both on the deterministic and the probabilistic sides of safety.

4. Proposals for evolving safety assurance standards

Three trends suggest there may be some relevance in revisiting the development assurance rationales.

4.1. Process assurance saturation

In all industrial domains there is some concern on how costs and effectiveness of assurance will evolve with the new levels of complexity reached by the engineered systems. The benefit-to-cost ratio of minute description of all development activities as fault prevention means, or that of systematic unit testing of any tiny bit of software as verification

[13] From the initiators to the top level accidents or incidents, back (FTA) and forth (FMEA and MBSA).

means, are being challenged for complex software intensive systems in various domains (aeronautics, manufacturing and batch industries noticeably).

4.2. Probabilistic dominance

Two reasons seem to underlie a trend towards extension of probabilistic quantification to all aspects of safety assessment:

1. *Complexity growth*. There is a line of thought according to which beyond a complexity threshold (e.g., hundreds of millions of code lines distributed over thousands of computing nodes), predictability is no longer tractable and shouldn't be kept as an assurance goal,

2. *Machine Learning AI*. Statistical estimation of programs is going to introduce randomness in that stronghold of determinism named 'safety critical software'. The AI trend in autonomy helps suggesting that probabilities could be the unifying concept for all aspects of safety assurance,

But this mindset, ethically disputable on one side, is also silent on the intractability of the probabilistic calculations that such software-inclusive probabilistic approach would need. Even though machine learning research is boosting progress on high dimensional statistical analysis, there are remaining intractable problems to compute rare event probabilities on high dimensional high fidelity CPS models [Mor16].

4.3. Leveraging formal methods

What has been achieved for software is being generalised for hybrid systems. Scalability of these methods will be an issue as it has been the case for software, but this is probably only a matter of time for such an issue to be solved.

4.4. Orientations for revisiting effort balance

The overall trend we would favour for the future of safety assurance would be to restrain any growth of process-based assurance, of probabilistic calculations and of dysfunctional inventories, in favour of more advanced structural and behavioural analyses, associated to "*design for verification*" policies that would restrain *uncontrolled growth of complexity*.

Here are the orientations we suggest. They are derived from the reviewed innovation perspectives and from the critics formulated about the excesses in enumerative dysfunctional analysis:

a) For software-intensive systems it is suspected that interaction failures (i.e. system functional specification errors) might become on par with random failures. In this context at least, safety should primarily be addressed as a control issue [Lev12]. This has been the case for the new IEC 63187 standard (defence systems) that is undergoing development along lines promoted by some members of our group,

b) Fault tolerance mechanisms should be a primary focus of safety assurance, especially FDIR detectors (safety monitors) and their associated *detection coverage*. Fault tolerance design is pivotal at the frontier between *impossibility engineering*, and *rareness engineering* (see table 1).

c) Independence between safety design and safety assessment does not necessarily imply that their respective models should be distinct. Risks of inconsistency between the two sorts of models should be considered, as well as predicate abstraction of hybrid system design models when tractable,

d) Inverse top-down dysfunctional abstract interpretation of complex CPS design models or paper specifications is a delusive error-prone activity. Fault tree analysis applied to software- intensive systems should be limited to *structural* analysis.

Table 1 is a tentative classification of the safety engineering activities. 'Impossibility engineering' groups all the activities devoted to deterministic controllability. Because control prevents the state from leaving the (green) safe "stay-in region" and from entering the red hazardous "stay-out region", control makes an accident *impossible* as long as the perturbations remain in the FDIR-tolerance domain. FDIR passivates some of the perturbations; it extends controllability as far as possible. Safety control is implemented in the product by the green activities. The green-impossibility part of table 1 should be the primary focus of assurance because it has first order impact on safety of the product.

Red engineering is "adversarial"[14]. It explores the perturbation domains and checks the tolerated/non-tolerated frontier i.e., the 'still-controllable"/"no-longer-controllable" frontier.

[14] In the game theoretic or IA sense: it aims at challenging the design, at finding counter-examples that falsifies some expected properties.

For brevity reasons we do not mention the 'amber' regions (safety margins, robustness zones) that may lie (geometrically) between the green and red regions.

The green probabilistic analyses are rare. They may become more common with the advent of machine learning AI in safety critical functions. They address randomness introduced *by design* in *normal conditions* i.e., a type of randomness that is not related to perturbations.

PAC is the 'Probably Approximately Correct' learning model [Shai14] to address learning assurance. Green quantification is needed when the *normal condition* estimation-related random failures may contribute to the dreaded events whose occurrence probabilities are upper-bounded by regulatory thresholds.

Red quantification should address entry in the red "stay-out" regions. When "all is lost" sufficient rareness becomes the only remaining engineering goal. This mindset is opposed to a rationale where rareness calculation is used to keep some control loss possibilities in the design *before* any attempt to design some prevention mechanisms.

Trustable probabilistic calculation on highly-integrated software-intensive systems is hard. Good reliability parameter estimation (MTBF) is hard as well, especially on new hardware and physical components. Any claim of *hypothesised* sufficient rareness to limit the 'impossibility-design' is at risk. Probabilistic dominance and economic competition combine to favour rareness engineering as the default option, supplemented with impossibility engineering only when required by regulation.

STPA lies on both sides of the green-red boundary. It lies on the green side because defining the 'safety constraints', in particular the safety invariants ("stay-in regions"), participates to design of control i.e., to green engineering. On the other hand, the part of STPA that identifies the accident scenarios and the Unsafe Control Actions (UCAs) participate to perturbation analysis (red) and functional safety requirements (green).

Table 1 distinguishes FTA applied in the structural case (loss dependencies) and in the behavioural case. In the latter, only *sufficient* combinations of causes are identified. Completeness is at risk.

Table 1: Tentative classification of the various components of safety engineering

	IMPOSSIBILITY Engineering		**RARENESS Engineering**
	Structural Analysis	**Behavioural Analysis**	**Probabilistic Analysis**
Green Engineering **Controllability**	Influence Networks Containment Regions Replication Policies	**'Stay-in' Regions** Safety Controls Safety Monitors SOTIF[15] STPA	Estimation-related Functional Failures (Signal processing, AI Machine Learning).
Perturbation Analysis **Red Engineering**	FMEAs FTAs Common Cause Analysis	STPA FMEAs FTAs (*inverse mode*) MBSA **'Stay-out' Regions**	Availability computation Reliability computation Quantification of *all* the 'controllability escapes'

5. Testing future assurance concepts

Admittedly this paper is conceptual and speculative. Some concrete experiments are needed to challenge and test these proposals. This is the purpose of the public domain multi-system use case originating from research in aeronautic certification [Led17] and from the Embedded France Working Group that co-signs this paper.

We intend to experiment the listed new techniques on this use case and to document the outcome in a collaborative book co-authored by industrials and academics.

[15] Safety of The Intended Function, concept and standard under development for autonomous cars

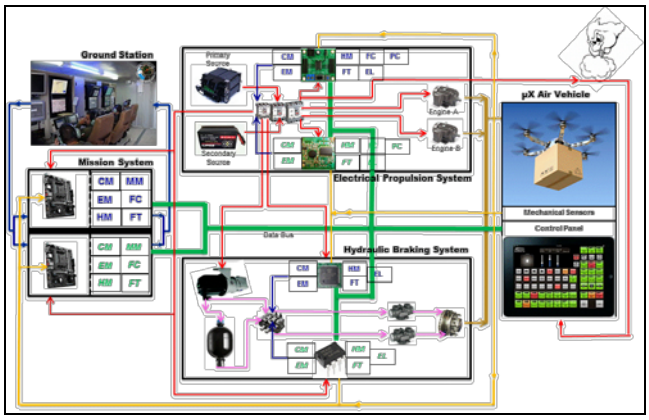

Figure 2: an open source toy multi-physics multi-system use case to test new verification techniques and candidate evolutions of development assurance standards

6. Conclusion

This paper originated form a consensus of our group: development assurance based on process conformity is jostled by the new classes of systems currently under development (autonomous vehicles, smart cities, smart grids etc.). Within development assurance, safety assurance is struggling. There is a need for foundational investigation of the present situation, for defining new orientations to overcome the widening gap between the complexity of engineered systems and the state of the art in system engineering, safety assessment, and development assurance.

We do not claim to have bridged this gap. We have reviewed promising research results along the lines this group promotes to address engineering of safety critical systems: a formal approach to system and software specification, design, verification and justification.

We have pointed towards two possible opportunities to optimise effort in safety assessment, and we have underlined the safety design aspects that to our opinion should deserve more attention in safety assurance standards.

Finally we have tried to sketch a better balance between probabilistic and deterministic analysis, between impossibility and rareness engineering.

In line with [Lev12] we advocate some primacy of 'impossibility engineering' over 'rareness engineering', and some primacy of 'green engineering' over 'red engineering', though in the end, and in both cases, they work hand-in-hand.

7. References

[Bac09] Back R. J**.** "Invariant-based programming: basic approach and teaching experiences". Formal aspects of computing 2009 21:227-244.

[Bel17] Belta C. Yordanov B. Gol E. A. "Formal Methods for Discrete-Time Dynamical Systems", Springer 2017.

[Ben18] A. Benveniste B. Caillaud & al., "Contracts for System Design" Foundations and Trends in Electronic Design Automation Now (2018).

[Bog14] S. Bogomolov, G. Freshe. & al. "Assume-Guarantee Abstraction Refinement Meets Hybrid Systems", HVC 2014, LNCS 8855 pp116-131 (2014).

[Boi18] Boissonnat J.D Chazal F. Yvinec M. "Geometric and Topological Inference". Cambridge University Press 2018.

[Bou15] Bouyer P., Markey N. & al. "Timed-automata abstraction of switched dynamical systems using control funnels" FORMATS'15 LNCS 9268 Springer 2015.

[Cha17] Chazal F. Michel B. "An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists" HAL 11 Oct.2017.

[ED79A/ARP4754A] "Guidelines for Development of Civil Aircraft and Systems", EUROCAE ED-79A and SAE Aerospace Recommended Practice ARP 4754A, 21/12/2010.

[ED135/ARP4761] "Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment", EUROCAE ED135 and SAE Aerospace Recommended Practice ARP 4761, 12/1996.

[Fre11] Freshe G. & al. "SpaceEx: Scalable Verification of Hybrid Systems" CAV 2011.

[Fri17] Frisk E. Krysander M. Jung D. "A toolbox for analysis and design of model-based diagnosis systems for large scale models". IFAC Paper On-line 2017 3287-3293.

[Gou17] Goubault E. Putot S. "Forward inner-approximated reachability of non-linear continuous systems" HSCC 2017.

[Kar90] Karnopp, D. Margolis D. Rosenberg R. "System dynamics: a unified approach". Wiley 1990.

[Lev12] Leveson N. "Engineering a safer world. Systems Thinking Applied to Safety". MIT Press 2012.

[Led17] Ledinot E. "Experimenting with the Overarching Properties: a use case by the RESSAC Project", Certification Together Conference (CTIC), Toulouse, March 21-23 2017.

[Mor16] Morio J. Balesdent M. "Estimation of Rare Event Probabilities in Complex Aerospace Systems" Woodhead Publishing (2016).

[Mov13] Mover S. Cimatti A. Tiwari A. Tonetta S.. “Time-Aware Relational Abstractions for Hybrid Systems” EMSOFT’13 Montréal Sept.29 – Oct 4 2013.

[Pay61] Paynter H. M. “Analysis and Design of Engineering Systems” MIT Press 1961.

[Pla18] Platzer A. “Logical Foundations of Cyber-Physical Systems” Springer 2018.

[Rau17] Rauzy A. “Model-Based Safety Assessment with Altarica 3.0” ESREL Conference, Portoroz Slovenia, June 18-22, 2017.

[Shai14] Shai Shalev-Schwartz and Shai Ben-David. “Understandig Machine Learning – From Theory to Algorithms”, Cambridge University Press, 2014.

[Slo12] C. Sloth, J. Pappas, R. Wisniewski “Compositional Safety Analysis using Barrier Certificates” HSCC 12 April 17-19 2012 Beijing, China.

[Slo13] C. Sloth, R. Wisniewski, M. Ergenstedt, “Complete abstractions of dynamical systems by timed automata” Nonlinear Analysis: Hybrid Systems 7 (2013).

[Wil16] Wilkinson C. “Integration of complex digitally intensive systems” – FAA Streamlining Assurance Processes Workshop – Dallas, September 13-15, 2016.

[WG-2010] P. Baufreton, JP. Blanquart, JL. Boulanger, H. Delseny, JC. Derrien, J. Gassino, G. Ladier, E. Ledinot, M. Leeman, J. Machrouh, P. Quéré, B. Ricque, “Multi-domain comparison of safety standards”, ERTS-2010, Toulouse, France, May 19-21 2010.

[WG-2012a] JP. Blanquart, JM. Astruc, P. Baufreton, JL. Boulanger, H. Delseny, J. Gassino, G. Ladier, E. Ledinot, M. Leeman, J. Machrouh, P. Quéré, B. Ricque, “Criticality categories across safety standards in different domains”, ERTS-2012, Toulouse, France, 1-3 February 1-3 2012.

[WG-2012b] E. Ledinot, J. Gassino, JP. Blanquart(, JL. Boulanger, P. Quéré, B. Ricque “A cross-domain comparison of software development assurance”, ERTS-2012, Toulouse, France, February 1-3 2012.

[WG-2012c] J. Machrouh, JP. Blanquart, P. Baufreton, JL. Boulanger, H. Delseny, J. Gassino, G. Ladier, E. Ledinot, M. Leeman, JM. Astruc, P. Quéré, B. Ricque, “Cross domain comparison of System Assurance”, ERTS-2012, Toulouse, France, February 1-3 2012.

[WG-2014] E. Ledinot, JP. Blanquart, Ph. Baufreton, C. Comar, J. Gassino, H. Delseny, “Joint use of static and dynamic software verification techniques: a cross-domain view in safety critical system industries”, ERTS-2014, Toulouse, France, February 5-7 2014.

[WG-2016a] E. Ledinot, J. Gassino, JP. Blanquart “Perspectives on Probabilistic Assessment of Systems and Software”, ERTS-2016, Toulouse, France, January 27-29 2016.

[WG-2016b] JP. Blanquart E. Ledinot, J. Gassino, “Software Safety Assessment and Probabilities”, DSN 2016, Toulouse, France, June 28 – July 1 2016.

[WG-2018a] JP. Blanquart E. Ledinot, J. Gassino, “Software Safety: A Journey across domains and safety standards”, ERTS-2018, Toulouse, France, January 31 – February 2, 2018.

[WG-2018b] B. Ricque, J.P. Blanquart, J. Gassino, “A cross-domain comparison of systematic errors control strategies”, Lambda-Mu-2018, Reims, France, October 16-18 2018.