Integration of Commonalities in the Paradigm of Model-Based Safety Assessment in Aerospace

POLITECNICO MILANO 1863

Isabella Lanzani, Luca Uliano, Riccardo Scattolini

LEONARDO

The two Keywords

COMMONALITIES

Development of aircraft architectures is a structured procedure, ruled by many standards and certifications.

Among others, **Common Causes of Failure** must be addressed
- to avoid Single Points of Failures
- to validate probabilistic results from the Fault Tree Analysis

MBSA

Model-Based Safety Assessment techniques aid the safety process

The **Safety Model** contains safety-relevant information about the architecture to be analyzed and can be used for :

→ Automatic generation of Fault Tree Analysis (FTA) and Minimal Cut Sets (MCS)
→ Interactive simulation capabilities
→ Process easily updates of new information from the system development team

Scope

GAP

MBSA does not treat in an industry-relevant way the analysis of common causes.

AIM

Extraction of safety-relevant information from the MBSA model via **semantic rules** and **naming patterns**.
Post-process automatic MCS to address **Independence Assertions** (claims that attest reasonably to an independence requirement).

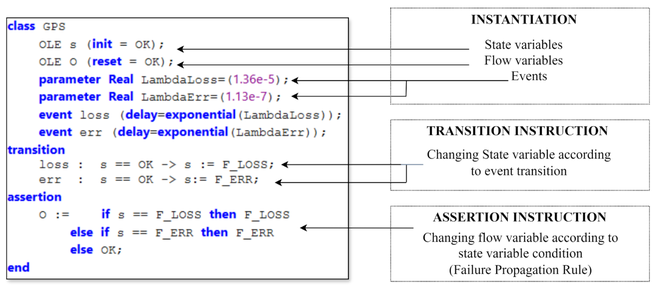
Proposed Approach

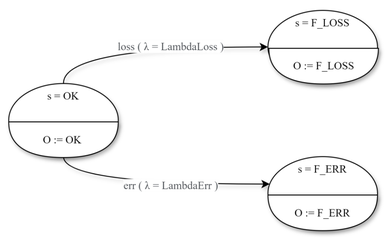
We wish to propose an enrichment procedure to cover this gap, focusing on a realistic architecture description for a Flight Control System.

1. **MBSA Modelling**
2. **Shared Resources Allocation**
3. **Common Mode Analysis Synthesis**

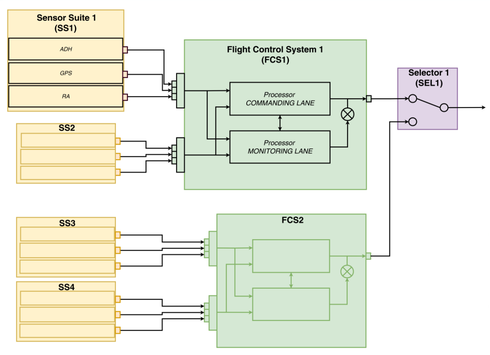
1. AltaRica Safety model

AltaRica language allows for a formal definition of the architecture where **transition instruction** rules the State Machine Automaton and **assertion instruction** rules the Failure Propagation Rules.





The safety model in AltaRica was written following **modelling rules**,
For identification of **:**
1. **functional redundancies** : components performing the same functionality
2. **different modes of failure** : components fails as *Loss* or *Erroneous*
3. **Physical connections** : ports connecting different components



EXPERIMENT

Architecture composed by :

→ 3 Sensor Suites (SS) (GPS, AHRS, RADALT)
→ 2 Flight Control System (FCS) (COM/MON lanes)
→ 1 Selector (SEL)

(only 1 of the 2 architectures compared in the full paper)

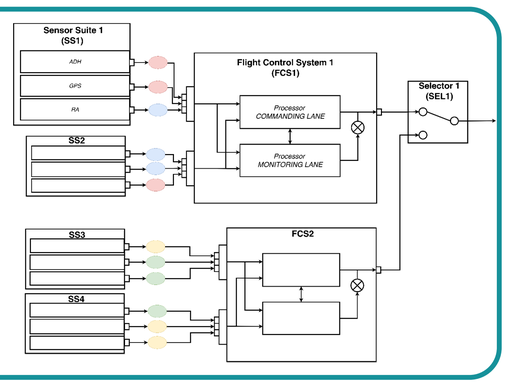
We automatically generate **Fault Tree Analysis (FTA)** and **Minimal Cut Set (MCS)** from this model used as support material for Preliminary System Safety Assessment and Common Mode Analysis.

Critical Pairs were extracted from the MCS list.
Those are all pairs of failure modes inside a MCS (or equivalently, are under the same "AND" gate of FTA) of most severe failure conditions.

2. DalCulator Allocation

EXPERIMENT

We design the need for MIMO A/D converters to transmit data from SS to FCS.
We produced an optimal allocation, avoiding the risk that a failure of the shared A/D converted would trigger the failure event.



DalCulator Software was used for computing the **Allocation of Shared Resources**. Receiving MCS as input and a set of hierarchical constraints, the tool gives the optimal allocation avoiding Single Points of Failure.

3. Synthesis of Results

We answer three **Independent Assertions**:
For each Critical Pair :
1 → Is the pair not functionally redundant?
2 → Does the pair fail with similar modes of failure?
For the proposed architecture :
3 → Which is the number of resources?

EXPERIMENT

Indicator	Description	Result
General information	N° Critical pairs	657
	N° Failure Events	49
Independence Assertions	Assertion 1	574
	Assertion 2	651
	Assertion 3	4

Conclusions & Future Works

In the authors' opinion, the most important feature here addressed is the possibility to automatically and freely **address criticalities** and **foreseen unsafe aspects** related to **dependencies** between components, from the very first parts of the preliminary design development. Moreover, the flexibility of using naming conventions directly from the safety model **enhances the usefulness of MBSA** techniques, maximizing the **extraction of relevant information**.

Future work aims to add metrics and assertions to cover the analysis as much as possible, including probabilistic consideration. We are curious to carry out similar analyses also in the verification phase.
The flexibility of DalCulator in different sharing resource scenarios will be investigated, as well as its other functionalities for DAL and budget allocations.

Acknowledgments

We'd like to thank the ONERA RIME Team: Kevin Delmas, Pierre Bieber, Christel Seguin, Tatiana Prosvirnova, Sergio Pizziol. Thank you for giving us support to DalCulator and for many helpful comments.
We would like to thank Enrico Zio for his endless interest in this journey.

Contacts and Additional Material

Isabella Lanzani
PhD student

FIND ME

DEIB, Politecnico di Milano,
Via Giuseppe Ponzio, 34,
Milano, Italy

+39 3707159191

isabella.lanzani@polimi.it

Download

AltaRica

ONERA DalCulator