Introduction to System Dependability

Kevin Delmas (kevin.delmas@onera.fr)

November 15, 2019

General overview

Specification of functional, logical and physical architectures with SysML

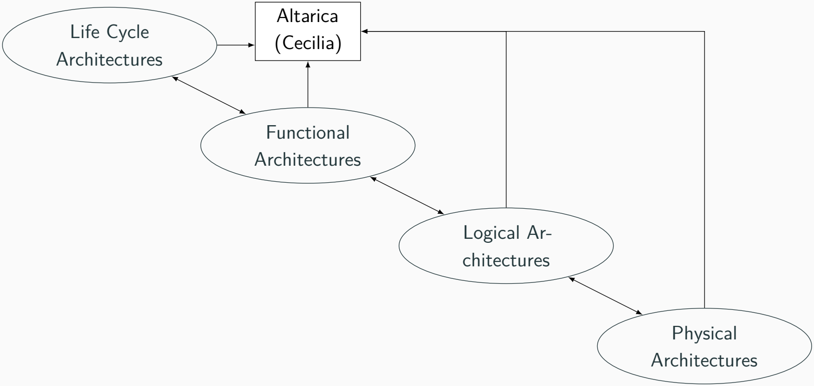


Figure 1: Dysfunctional analysis in development process

Goal of this lesson

Check if an autonomous system can be used safely to perform a mission in a given context

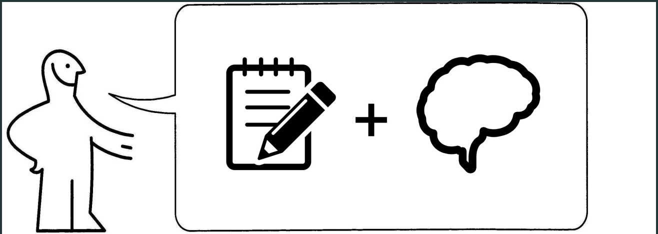
Some definitions are mandatory to understand labs (what a surprise)

= slides preparing computer lab

= reminder (should be)

Be careful !

⚠ Interactive course ahead



Numerous exercises during class

Be **active** !

Preliminary concepts

Introduction to System Dependability

What is a system?

What is a system ?

Definition (System)
A system is a set of interacting items, forming an integrated whole

Example (System)
examples of various complexity: air traffic control, aircraft + pilot, flight-control system, computers, sensors, actuators,...

Use the drone shepherd as example to illustrate safety assessment.

= slides preparing computer lab

Be careful !

Case study: Drone shepherd

Drone shepherd: Mission

Main mission Drone monitors flock and prevents bear attack

Drone shepherd: Mission

Main mission Drone monitors flock and prevents bear attack

Drone main features are

- monitor autonomously the flock (no operator interventions),
- prevent bear attack,
- send data to ground station.

Flock monitored by the drone

Ground station receives data from drone and transmits requests from operator

Operator initiates/aborts drone mission

Drone shepherd: Mission

Main mission Drone monitors flock and prevents bear attack

Drone main features are

- monitor autonomously the flock (no operator interventions),
- prevent bear attack,
- send data to ground station.

Drone shepherd: Context

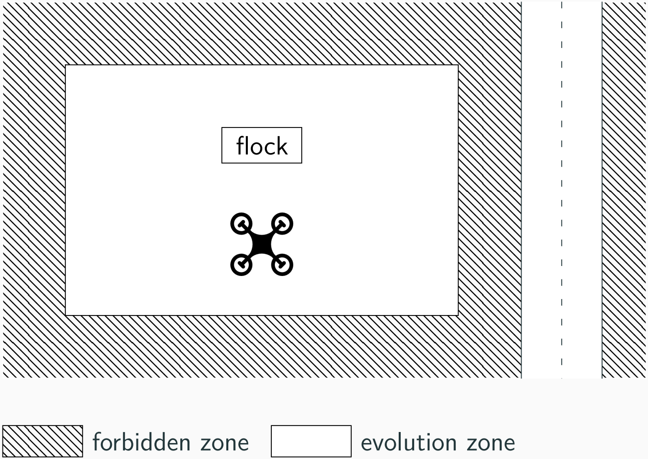


Figure 2: Overview of the system

Introduction to System **Dependability**

What is dependability?

What is dependability?

Definition (Dependability [ALRL04])
The ability of the system to deliver service that can justifiably be trusted.

Some vocabulary about dependability:

failure occurrence of the deviation of the delivered service from expected service

failure rate probability of failure per unit of time of items in operation

failure mode characterization of the way a system/item fails

Drone shepherd

Nominal function Monitor drone state

Failure UAV unables to provide a reliable state estimation

Failure modes

- the UAV does not provide any state estimation
- the UAV provides an erroenous estimation of its state

More vocabulary

System/items behaviors depend on

- control/observation interface
- internal states (not always distinguishable)
  - nominal functioning modes
  - error states part of the total state of a system/item that may lead to its subsequent failure
- fault = hypothesized or adjudged cause of an error state

Fault propagation paths:

$$\text{fault} \Rightarrow \text{error} \Rightarrow \text{failure}$$

Drone shepherd

Failure mode the UAV provides an erroneous estimation of its state

Error state memory storing the state estimation is corrupted

Fault

- Primary (intrinsic) cause: memory chip failure
- Secondary cause (extrinsic): corruption due to cosmic rays

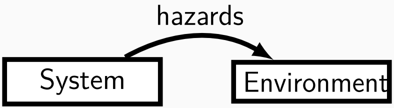
Observability Detectable if ECC or bit parity is available for state estimation data

Failure can lead to harmful events

so-called hazards

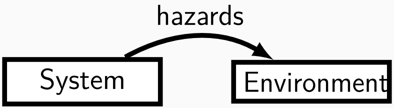
What are the hazards here ?

Possible hazards



Possible hazards :

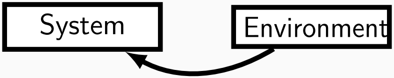
Possible hazards



Possible hazards :

- Hurt the flock
- Collision with vehicle (road)

Possible hazards



Possible hazards :

- Hurt the flock
- Collision with vehicle (road)

Possible adversary conditions:

- Wind or Rain ⇒ drone can't fly
- Poor GNSS signal ⇒ drone can't locate itself

Concretely, how to evaluate dependability?

Math corner: Dependability measures

Definition (Reliability(R))
Ability of a system S to ensure continuity of correct service:

$$R(t) = p(S \text{ non faulty over } [0, t])$$

Definition (Availability(A))
Ability of a system S to deliver a correct service at a given time:

$$A(t) = p(S \text{ non faulty at } t)$$

Definition (Maintainability(M))
Ability of a system S to undergo modifications and repair

$$M(t) = 1 - p(S \text{ non repaired over } [0, t])$$

Math corner: Dependability measures

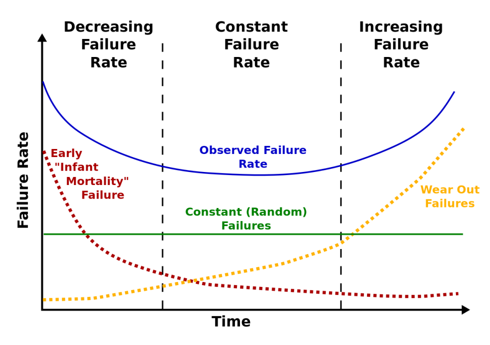
Definition (Failure Rate (Λ))
Probability of a system S to fail at $t+dt$ knowing it has not failed over $[0,t]$:

$$\Lambda(t) = \lim_{dt \to 0} \frac{p(S \text{ fails during } [t, t+dt])}{dt} \frac{1}{R(t)}$$

Relation with R:

$$R(t) = e^{-\int_0^t \Lambda(u) du}$$

Math corner: Bath curve failure rate



Assume items used during constant failure rate phase

$$\Lambda(t) = \lambda; \quad R(t) = e^{-\lambda t}; \quad MTTF = \frac{1}{\lambda}$$

Math corner: Computation approximation

Definition (Rare failure assumptions)
When $\lambda t \sim 0$ (usually $\lambda t < .1$) use Taylor expansion for computations:

$$1 - R(t) = 1 - e^{-\lambda t} \underset{0}{\sim} \lambda t$$

Definition (Independence & pessimism assumption)
If two components C_1 and C_2 have independent failures with failure rate λ_1 and λ_2

$$\begin{aligned}
p(\text{both fail}) &\underset{\text{independent}}{=} p(C_1 \text{ fails})p(C_2 \text{ fails}) = \lambda_1 \lambda_2 t^2 \\
p(\text{one fails}) &= p(C_1 \text{ fails}) + p(C_2 \text{ fails}) - p(\text{both fail}) \\
&\underset{\text{pessimism}}{=} p(C_1 \text{ fails}) + p(C_2 \text{ fails})
\end{aligned}$$

Risk acceptability

The question is:

What happens if ?

Risk acceptability

The question is:

What happens if drone shepherd fails?

Risk acceptability

The question is:

What happens if drone shepherd fails?

- Trajectory of the drone is not controlled
- Possible collision with vehicle
- Depending on the obstacle and aircraft speed, injury or death of passengers.

Risk acceptability

New question:

Knowing the severity of the failure, what is an acceptable frequency of such failure?

Another general definition of dependability:

"ability to avoid service failures that are frequent and more severe than acceptable"

What does service failure, severe, frequent, acceptable mean?
⇒ Regulatory texts : let us consider civil aircraft

Classification of failures

Risk acceptability for civil aircraft

When considering safety of civil aircraft:

Failure Condition (FC) kind of service failures that:

- has an effect on the aircraft and its occupants, both direct and consequential,
- caused by one or more failures, considering relevant adverse operational or environmental conditions.

Severity Failure Condition is classified in accordance to the severity of its effects as defined .

Risk acceptability for civil aircraft

severity class	effects description	acceptable frequency
catastrophic	prevent continuous safe flight and landing: aircraft loss and loss of crew and passengers	$< 10^{-9}$ per flight hour and no single failure leads to the FC
hazardous	large reduction in safety margins or functional capabilities or physical distress or high crew workload or serious or fatal injuries to a relatively small number of passengers	$< 10^{-7}$ per flight hour

Risk acceptability for civil aircraft

severity class	effects description	acceptable frequency
major	significant reduction in safety margin or functional capabilities or significant increase in crew workload or discomfort to occupants possibly including injuries	$< 10^{-5}$ per flight hour
minor	no significant reduction in aircraft safety.	$< 10^{-3}$ per flight hour
no safety effect		

Risk acceptability for civil aircraft

Example (Severity & objectives)
"Total loss of drone shepherd " is classified , so

Risk acceptability for civil aircraft

Example (Severity & objectives)
"Total loss of drone shepherd " is classified Catastrophic, so

- the probability rate of this failure condition shall be less than 10^{-9} /FH and
- No single event shall lead to this failure condition

Warnings:

- The regulation is not the same for military aircraft
- The regulation for civil UAV is still in discussion
- A generic agreed classification is an open question for a lot of domains

How to apply these concepts to build a complex dependable system?

Dependability process: focus on aeronautic process

Process based approach

Main steps:

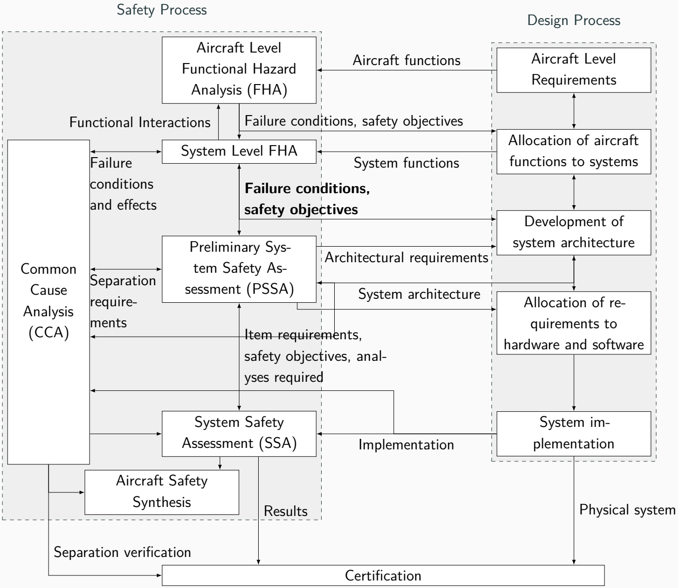
- Identify dependability requirements
- Specify a system architecture to ensure these properties
- Assess whether the proposed specification fulfills the dependability requirement
- If OK, refine the system design and iterate

Guidelines tuned according to the system kind:

- ISO 26262 [ISO10] for automotive systems
- ECSS Q-ST 40 for space systems
- ARP 4754A [SAE10], ARP 4761 [SAE96] for aeronautic systems

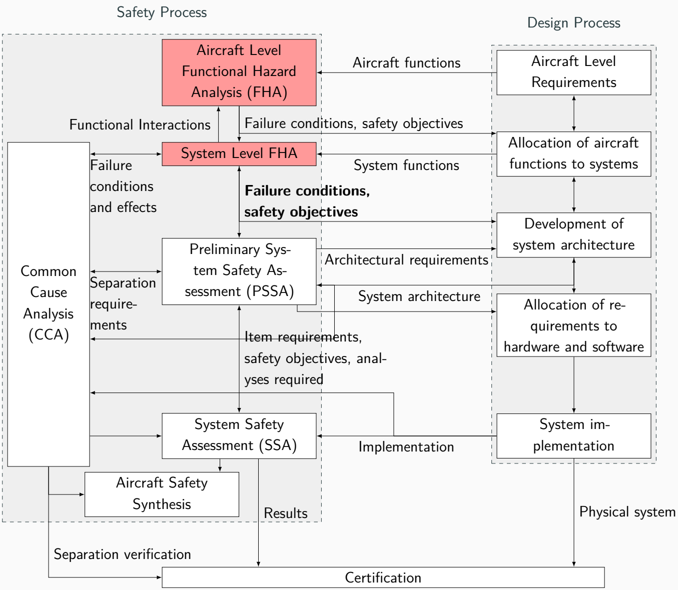
When should we perform safety activities?

Safety Process (Complete)



When should we identify and classify Failure Conditions?

Safety Process (FHA)



Functional breakdown

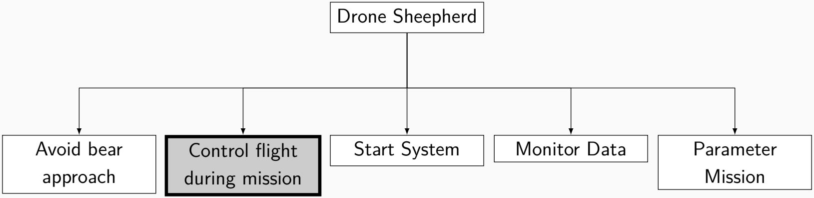


Figure 3: Functional breakdown (cf SysML lesson)

Risks : trouble in flight during mission ⇒ refine decomposition on Control flight during mission

Functional breakdown

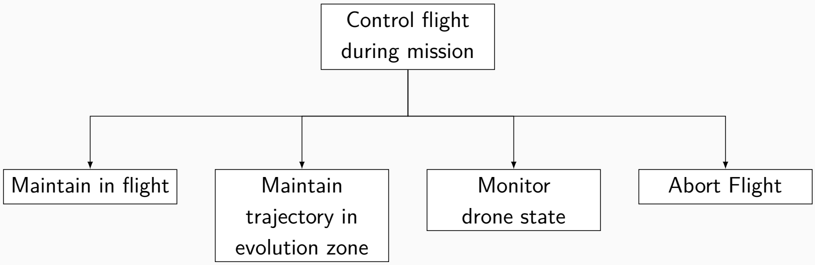


Figure 4: Functional breakdown

FHA : assess the consequences and the criticality of the loss or misapplication of each function in a given context

FHA by the example

Function	Failure	Context	Consequences	Criticality
Maintain trajectory in evolution zone	loss	cannot abort flight		

Table 1: FHA example

FHA by the example

Function	Failure	Context	Consequences	Criticality
Maintain trajectory in evolution zone	loss	cannot abort flight	Crash outside evolution zone, possible collision with vehicules	**Catastrophic**

Table 1: FHA example

FHA by the example

Function	Failure	Context	Consequences	Criticality
Maintain trajectory in evolution zone	loss	cannot abort flight	Crash outside evolution zone, possible collision with vehicules	**Catastrophic**
Maintain in flight	loss	can maintain in evolution zone		

Table 1: FHA example

FHA by the example

Function	Failure	Context	Consequences	Criticality
Maintain trajectory in evolution zone	loss	cannot abort flight	Crash outside evolution zone, possible collision with vehicules	**Catastrophic**
Maintain in flight	loss	can maintain in evolution zone	Crash in evolution zone, possible hurt flock	**Hazardous**

Table 1: FHA example

FHA by the example

Function	Failure	Context	Consequences	Criticality
Maintain trajectory in evolution zone	loss	cannot abort flight	Crash outside evolution zone, possible collision with vehicules	**Catastrophic**
Maintain in flight	loss	can maintain in evolution zone	Crash in evolution zone, possible hurt flock	**Hazardous**
Abort Flight	loss	can maintain in evolution zone		

Table 1: FHA example

FHA by the example

Function	Failure	Context	Consequences	Criticality
Maintain trajectory in evolution zone	loss	cannot abort flight	Crash outside evolution zone, possible collision with vehicules	**Catastrophic**
Maintain in flight	loss	can maintain in evolution zone	Crash in evolution zone, possible hurt flock	**Hazardous**
Abort Flight	loss	can maintain in evolution zone	Drone behaves properly $\Rightarrow$ No safety effect	**NSE**

Table 1: FHA example

FHA by the example: Failure conditions

Failure condition Combination of functional failures that have an effect an system's safety

FHA by the example: Failure conditions

Failure condition Combination of functional failures that have an effect an system's safety

Function	Failure	Context	Consequences	Criticality
Maintain trajectory in evolution zone	loss	cannot abort flight	Crash outside evolution zone, possible collision with vehicules	**Catastrophic**

FHA by the example: Failure conditions

Failure condition Combination of functional failures that have an effect an system's safety

Function	Failure	Context	Consequences	Criticality
Maintain trajectory in evolution zone	loss	cannot abort flight	Crash outside evolution zone, possible collision with vehicules	**Catastrophic**

CAT_SOL cannot maintain trajectory in evolution zone and cannot abort flight

FHA by the example: Failure conditions

Failure condition Combination of functional failures that have an effect an system's safety

Function	Failure	Context	Consequences	Criticality
Maintain in flight	loss	can maintain in evolution zone	Crash in evolution zone, possible hurt flock	**Hazardous**

CAT_SOL cannot maintain trajectory in evolution zone and cannot abort flight

FHA by the example: Failure conditions

Failure condition Combination of functional failures that have an effect an system's safety

Function	Failure	Context	Consequences	Criticality
Maintain in flight	loss	can maintain in evolution zone	Crash in evolution zone, possible hurt flock	**Hazardous**

CAT_SOL cannot maintain trajectory in evolution zone and cannot abort flight

HAZ_SOL cannot maintain in flight

FHA by the example: Failure conditions

Failure condition Combination of functional failures that have an effect an system's safety

Function	Failure	Context	Consequences	Criticality
Abort Flight	loss	can maintain in evolution zone	Drone behaves properly ⇒ No safety effect	**NSE**

CAT_SOL cannot maintain trajectory in evolution zone and cannot abort flight

HAZ_SOL cannot maintain in flight

FHA by the example: Safety objectives

Safety objectives bounds over indicators commensurate with failure condition criticality

Example (Safety objectives)
What are the safety objectives for CAT_SOL ?

FHA by the example: Safety objectives

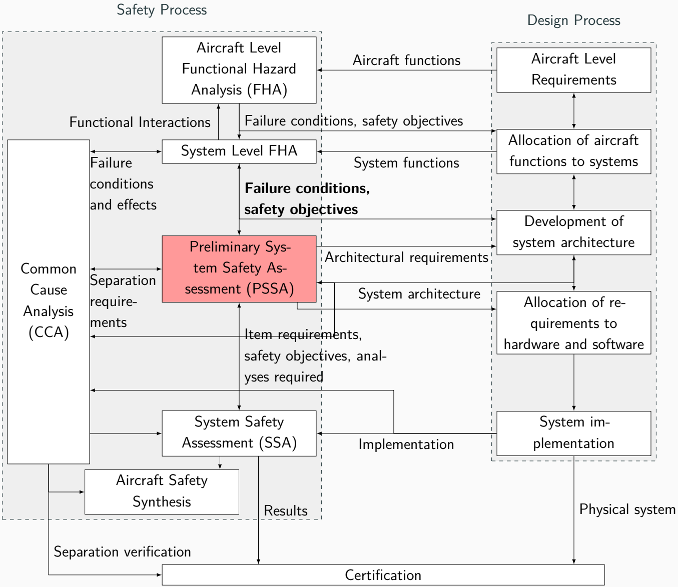
Safety objectives bounds over indicators commensurate with failure condition criticality

Example (Safety objectives)
What are the safety objectives for CAT_SOL ?

minimal number of failures ≥ 2 and probability $\leq 10^{-7}$

When should we check dependability requirements?

Safety Process (PSSA)



PSSA: Example

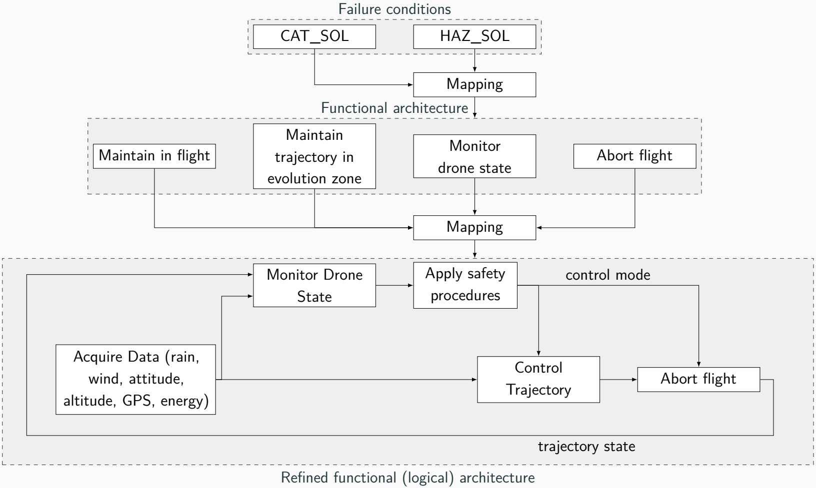
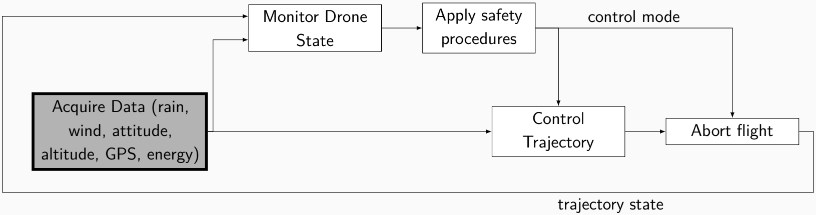


Figure 5: Failure conditions and functional architectures

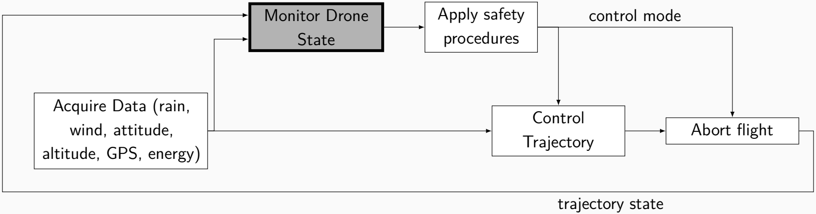
Functional architecture



Acquire Data each data acquired by independent function, failure modes are:

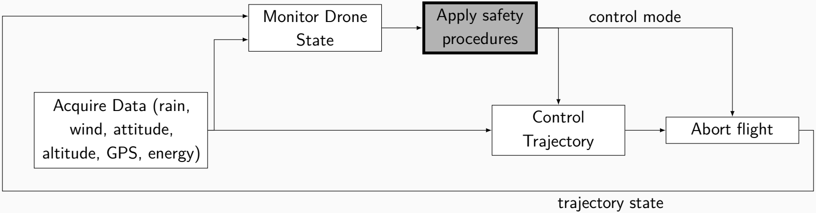
- erroneous: send inconsistent data,
- lost: stop sending data.

Functional architecture



Monitor drone state each data is checked by independent and perfect alarms (neglected failures in the Lab but not in real life !!!).

Functional architecture



Apply safety procedures according to alarms, select control mode.

Apply safety procedures

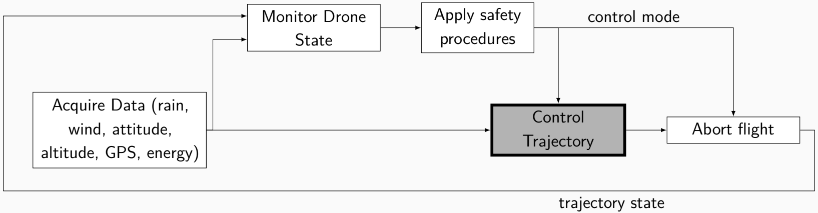
Control mode is selected according to following rules:

1. attitude **or** trajectory not OK ⇒ flight termination,
2. rain **or** wind **or** altitude **or** energy not OK ⇒ landing,
3. loss of GNSS or localization ⇒ hovering,
4. **No regression rule**: Once degraded mode selected, next modes cannot be "less degraded".

$$Mission < Hovering < Landing < Flight\ \ Termination$$

5. **Pessimism Rule**: several modes can be selected ⇒ most degraded mode must be chosen

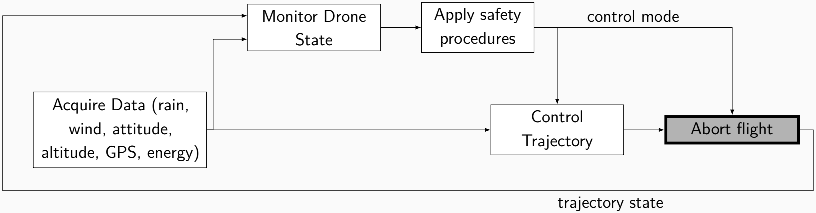
Functional architecture



Control Trajectory navigation and pilot functions computing actuators commands from flight parameters and control mode. Each function can be:

- erroneous: compute incorrect commands,
- lost: stop computing any command.

Functional architecture



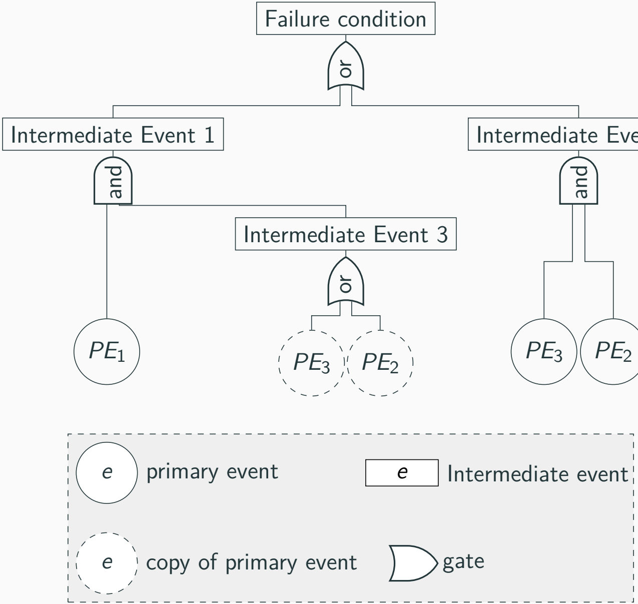
Abort Flight function cutting motors power supply if flight termination mode selected, failure modes are :

- failed_permanent: untimely triggering of flight termination,
- failed_lost: ignore flight termination request.

How to check dependability requirements?

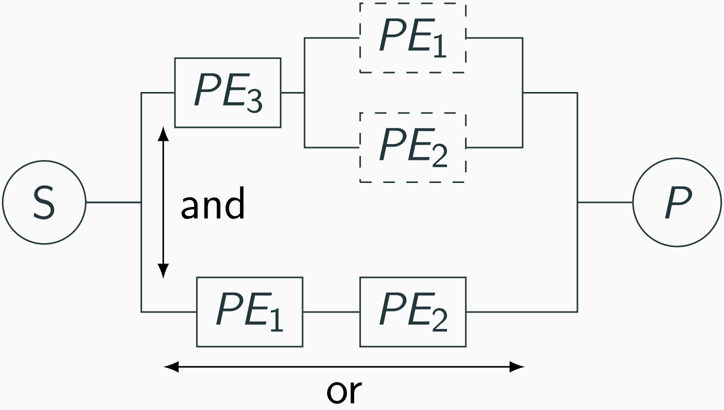
⇒ **several complementary methods**

Failure propagation: The Fault Tree



Failure propagation: Reliability Block Diagram

Alternative notation for fault trees (analogy with serial-parallel electrical circuits)



How do we use these representations?

Failure propagation

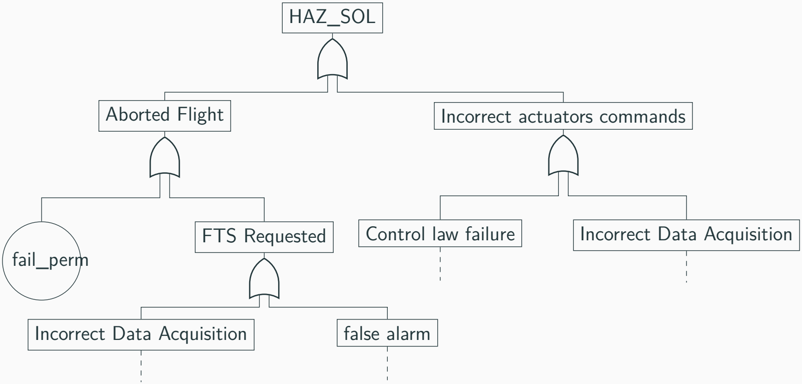


Figure 6: Incomplete fault tree of HAZ_SOL

How to perform safety assessment out of fault trees?

Minimal Cutsets

Fault tree ⇔ formula φ describing the failure combinations leading to a failure condition

Definition (Minimal cutsets (MCS))
A cutset $C = \{f_1, \cdots, f_n\} \in MCS$ iff :

- if all $f \in C$ occurs then φ is true;
- it does not exist another cutset C' satisfying the previous properties and such that $C' \subset C$

Minimal Cutsets

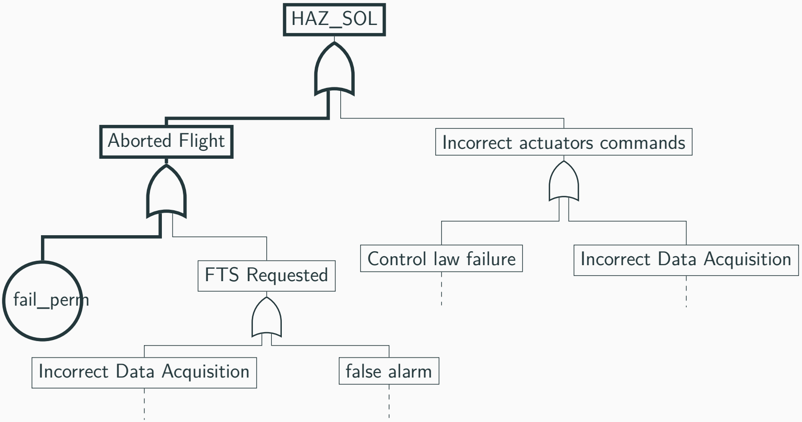


Figure 7: Incomplete fault tree of HAZ_SOL

MCS Order

Definition (Order)
Order of an FC is the minimal number of failures leading to FC. Formally, let MCS be the minimal cutsets for FC, then the order is the minimal cardinality of MCS:

$$order(FC) = \min_{c \in MCS_{FC}} (|c|)$$

Example (Order)
For $MCS_{FC} = \{\{a, b\}, \{c\}\}$ we have:

$$\begin{aligned} order(FC) &= \min_{c \in MCS_{FC}} (|c|) \\ &= min(|\{a, b\}|, |\{c\}|) \\ &= 1 \end{aligned}$$

Approximate probability computation

Let MCS be the minimal cutsets for FC, and $p(event)$ probability of failure for primary events:

$$p(FC) = \sum_{cut \in MCS} \prod_{event \in cut} p(event)$$

Example (Approximate computation)
Let MCS=$\{\{a, b\}, \{c\}\}$ be the minimal cutsets for FC :

$$p_{approx}(FC) = p(a)p(b) + p(c)$$

Safety objectives (Reminder)

criticality	qualitative requirement	quantitative requirement
Catastrophic	order ≥ 2	$p \leq 10^{-9}/\textit{flight hour}$
Hazardous	order ≥ 1	$p \leq 10^{-7}/\textit{flight hour}$
Major	order ≥ 1	$p \leq 10^{-5}/\textit{flight hour}$
Minor	order ≥ 1	$p \leq 10^{-3}/\textit{flight hour}$

Table 2: Acceptability matrix

Definition (Order)
The order is the minimal cardinality of MCS

Example (Order)
The order of $MCS = \{\{a, b\}, \{c\}\}$ is 1

Requirements verification

⚠ We assume that primary events are independent

1. Determine the failure conditions and their criticality (from FHA)
2. Build the fault trees for each failure condition
3. Compute the minimal cutsets
4. Qualitative verification : Compute the order and compare it to the required bound
5. Quantitative verification : Compute the probability and compare it to the required bound

Requirements verification

Example (Verification)
Let $MCS_{FC} = \{\{a,b\},\{c\}\}$ with $p(a) = p(b) = p(c) = 10^{-4}$.
Is it acceptable if FC criticality is Hazardous ?

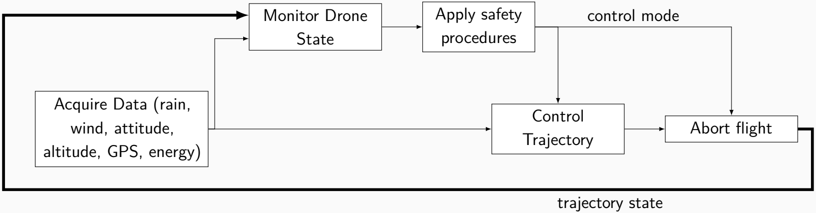
Requirements verification

Example (Verification)
Let $MCS_{FC} = \{\{a,b\},\{c\}\}$ with $p(a) = p(b) = p(c) = 10^{-4}$.
Is it acceptable if FC criticality is Hazardous ?

$$\begin{aligned} order(FC) &= \min_{c \in MCS_{FC}} (|c|) = 1 \Rightarrow OK \\ p(FC) &= p(a)p(b) + p(c) \simeq 10^{-4} > 10^{-5} \Rightarrow KO \end{aligned}$$

Wait we didn't completely built the fault tree, how to deal with the reconfiguration ?

Limitation of fault trees



With fault trees enroll reconfiguration steps yourself
⇒ time-consuming, tedious and error-prone

With altarica encode directly reconfiguration and let tool analyze system for you

What's Altarica ?

Next lesson ! Now a recap

Specification of functional, logical and physical architectures with SysML

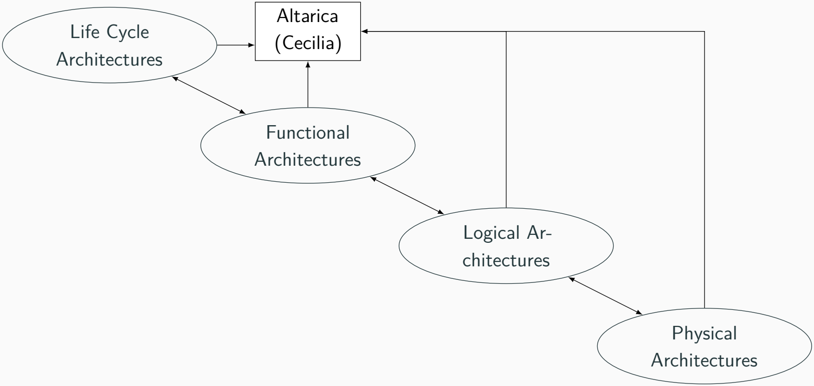


Figure 8: Dysfunctional analysis in development process

Today's lesson in 30"

Perform safety assessment is:

1. Define system mission and operational context
2. Identify the risks
3. Determine for each high level function the criticality of its failure and deduce failure conditions
4. Build fault tree (or other representations) for each failure condition
5. Compute MCS and probability and compare it to the safety objectives.

Bibliography i

Algirdas Avizienis, J-C Laprie, Brian Randell, and Carl Landwehr.
Basic concepts and taxonomy of dependable and secure computing.
IEEE transactions on dependable and secure computing, 1(1):11–33, 2004.

ISO.
ISO-26262 -Road vehicles – Functional safety, 2010.

Bibliography ii

SAE.
Aerospace Recommended Practices 4761 - guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment, 1996.

SAE.
Aerospace Recommended Practices 4754a - Development of Civil Aircraft and Systems, 2010.

Thank you

Deal with dependencies

Requirements verification

⚠ We assume that primary events are independent

1. Determine the failure conditions and their criticality (from FHA)
2. Build the fault trees for each failure condition
3. Compute the minimal cutsets
4. Qualitative verification : Compute the order and compare it to the required bound
5. Quantitative verification : Compute the probability and compare it to the required bound

What if some primary events are not independent (tire burst, engine burst,...)?

Deal with dependencies

What could cause the simultaneous failure of several components?

- Adversary conditions: overheat, electromagnetic perturbations, ...
- Destruction of a whole zone: engine burst, in-flight fire,...
- But also: implementation common mode (functions depending on the same equipments), specification errors, systematic development errors,...

What are the consequences?

- Possible violation of safety objective
  $\Rightarrow$ Identify and analyze common mode during the Common Cause Analysis (CCA)

Deal with dependencies

Example (Dependencies impact)
Minimal cut $C = \{a, b, c\}$ for a catastrophic FC, if a and b are not independent (triggered by d):

- ⇒ $C \rightarrow \{d, c\}$
- ⇒ Order goes from 3 to 2
- ⚠ System does not fulfil requirements

Deal with dependencies

Event in MCS shall be independent to avoid that their implementation introduces a common mode reducing the size of the MCS under the order requirement.

⇓

Define the segregation requirements to ensure independence

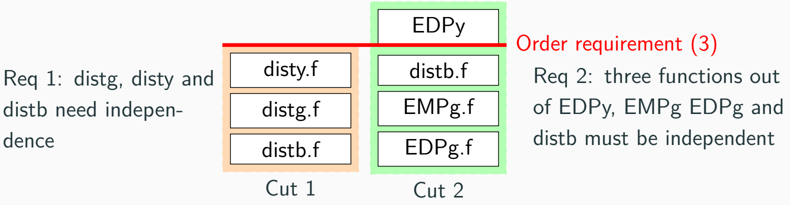


Figure 9: Independence requirements for Total hydraulic system

Limitation of fault trees

What could cause the simultaneous failure of several components?

- Adversary conditions: overheat, electromagnetic perturbations, . . .
- Destruction of a whole zone: engine burst, in-flight fire,. . .
- But also: implementation common mode (functions depending on the same equipments), specification errors, systematic development errors,. . .

Limitation of fault trees

What could cause the simultaneous failure of several components?

- Adversary conditions: overheat, electromagnetic perturbations, ... ⇒ Random faults
- Destruction of a whole zone: engine burst, in-flight fire,... ⇒ Random faults
- But also: implementation common mode (functions depending on the same equipments), specification errors, systematic development errors,... ⇒ Systematic faults

Acceptability cannot be based on probability assessment !
⇒ ensure a level of confidence in development correctness

Design Assurance Level

Limitation of fault trees

DAL Development Assurance Level (ARP4754) is the level (from E to A) of rigor of development assurance tasks performed on functions and items (software, hardware) whose fault result

Warning:

- DAL can be associated with
  - Functions: FDAL
  - Items: IDAL
- For each DAL level, assurance activities are listed in:
  - ARP4754 for FDAL
  - DO178 (SW) and DO254 (HW) for IDAL

Assurance Activities Examples

	Objective		Applicability			
	Description	Ref	A	B	C	D
1	Software high-level requirements comply with system requirements.	6.3.1a	I	I	R	R
2	High-level requirements are accurate and consistent.	6.3.1b	I	I	R	R
3	High-level requirements are compatible with target computer.	6.3.1c	R	R		

- High DAL level ⇒ great number of assurance activities
  ⇒ costly
  ⇒ minimize the DAL of software and hardware

DAL Allocation: Basic Allocation

Based on the severities of the FCs that function fault contributes to.

Sev(FC)	DAL(FC)
CAT	A
HAZ	B
MAJ	C
MIN	D
NSE	E

Table 3: Link between severity and DAL

What does "the severities of the FCs that function fault f contributes to" mean?

$\Rightarrow$ the severities of the FCs whose MCS contains f

DAL Allocation: Basic Allocation

Context
- Let fc_1 (resp fc_2) be a failure condition of severity HAZ (resp. MAJ)
- Let $MCS_1 = \{\{f_1, f_2, f_4\}, \{f_3\}\}$ and $MCS_2 = \{\{f_1, f_3\}\}$

Question What is the basic DAL of f_1?

DAL Allocation: Basic Allocation

Context
- Let fc_1 (resp fc_2) be a failure condition of severity HAZ (resp. MAJ)
- Let $MCS_1 = \{\{f_1, f_2, f_4\}, \{f_3\}\}$ and $MCS_2 = \{\{f_1, f_3\}\}$

Question What is the basic DAL of f_1?

Answer f_1 contained in MCS_1 and MCS_2 so $DAL(f_1) = worst(DAL(fc_1), DAL(fc_2)) = DAL(HAZ) = B$

Question What is the basic DAL of f_2?

DAL Allocation: Basic Allocation

Context
- Let fc_1 (resp fc_2) be a failure condition of severity HAZ (resp. MAJ)
- Let $MCS_1 = \{\{f_1, f_2, f_4\}, \{f_3\}\}$ and $MCS_2 = \{\{f_1, f_3\}\}$

Question What is the basic DAL of f_1?

Answer f_1 contained in MCS_1 and MCS_2 so $DAL(f_1) = worst(DAL(fc_1), DAL(fc_2)) = DAL(HAZ) = B$

Question What is the basic DAL of f_2?

Answer f_2 contained only in MCS_1 so $DAL(f_2) = worst(DAL(fc_1)) = DAL(HAZ) = B$

DAL Allocation: Degradation rules

Designer can downgrade the basic DAL *basic* of a function using independence, the allocation must fulfill the following rules:

Rule 1 *basic* can be degraded at most by two levels

Rule 2 For all cuts $\{f_1, \cdots, f_n\} \in MCS_{fc}$ where $f_1, \cdots, f_n$ are independent, either:

- Option 1: it exists f_i such that $DAL(f_i) = basic$
- Option 2: it exists f_i, f_j such that $DAL(f_i) = DAL(f_j) = basic - 1$

DAL Allocation: Degradation rules

Suppose f_1, f_2, f_3 and f_4 are independent and cost : DAL A $=$ 20, DAL B $=$ 15, DAL C $=$ 5, DAL D $=$ 4, DAL E $=$ 0

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	

DAL Allocation: Degradation rules

Suppose f_1, f_2, f_3 and f_4 are independent and cost : DAL A $=$ 20, DAL B $=$ 15, DAL C $=$ 5, DAL D $=$ 4, DAL E $=$ 0

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	
B	$\{f_1, f_2, f_4\}$	$\geq$ B	$\geq$ D	-	$\geq$ D	1

DAL Allocation: Degradation rules

Suppose f_1, f_2, f_3 and f_4 are independent and cost : DAL A $=$ 20, DAL B $=$ 15, DAL C $=$ 5, DAL D $=$ 4, DAL E $=$ 0

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	
B	$\{f_1, f_2, f_4\}$	$\geq$ B	$\geq$ D	-	$\geq$ D	1
	$\{f_3\}$	-	-	$\geq$ B	-	-

DAL Allocation: Degradation rules

Suppose f_1, f_2, f_3 and f_4 are independent and cost : DAL A $=$ 20, DAL B $=$ 15, DAL C $=$ 5, DAL D $=$ 4, DAL E $=$ 0

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	
B	$\{f_1, f_2, f_4\}$	$\geq$ B	$\geq$ D	-	$\geq$ D	1
	$\{f_3\}$	-	-	$\geq$ B	-	-
C	$\{f_1, f_3\}$	$\geq$ C	-	$\geq$ E	-	1

DAL Allocation: Degradation rules

Suppose f_1, f_2, f_3 and f_4 are independent and cost : DAL A = 20, DAL B = 15, DAL C = 5, DAL D = 4, DAL E = 0

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	
B	$\{f_1, f_2, f_4\}$	$\geq$ B	$\geq$ D	-	$\geq$ D	1
	$\{f_3\}$	-	-	$\geq$ B	-	-
C	$\{f_1, f_3\}$	$\geq$ C	-	$\geq$ E	-	1
Result		$\geq$ B	$\geq$ D	$\geq$ B	$\geq$ D	
Cost		38				

Is it the cheapest option?

$\Rightarrow$ **Let's try again!**

DAL Allocation: Degradation rules

Suppose f_1, f_2, f_3 and f_4 are independent and cost : DAL A = 20, DAL B = 15, DAL C = 5, DAL D = 4, DAL E = 0

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	

DAL Allocation: Degradation rules

Suppose f_1, f_2, f_3 and f_4 are independent and cost : DAL A = 20, DAL B = 15, DAL C = 5, DAL D = 4, DAL E = 0

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	
B	$\{f_1, f_2, f_4\}$	$\geq$ C	$\geq$ C	-	$\geq$ D	2

DAL Allocation: Degradation rules

Suppose f_1, f_2, f_3 and f_4 are independent and cost : DAL A $=$ 20, DAL B $=$ 15, DAL C $=$ 5, DAL D $=$ 4, DAL E $=$ 0

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	
B	$\{f_1, f_2, f_4\}$	$\geq$ C	$\geq$ C	-	$\geq$ D	2
	$\{f_3\}$	-	-	$\geq$ B	-	-

DAL Allocation: Degradation rules

Suppose f_1, f_2, f_3 and f_4 are independent and cost : DAL A $=$ 20, DAL B $=$ 15, DAL C $=$ 5, DAL D $=$ 4, DAL E $=$ 0

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	
B	$\{f_1, f_2, f_4\}$	$\geq$ C	$\geq$ C	-	$\geq$ D	2
	$\{f_3\}$	-	-	$\geq$ B	-	-
C	$\{f_1, f_3\}$	$\geq$ E	-	$\geq$ C	-	1

DAL Allocation: Degradation rules

Suppose f_1, f_2, f_3 and f_4 are independent and cost : DAL A $=$ 20, DAL B $=$ 15, DAL C $=$ 5, DAL D $=$ 4, DAL E $=$ 0

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	
B	$\{f_1, f_2, f_4\}$	$\geq$ C	$\geq$ C	-	$\geq$ D	2
	$\{f_3\}$	-	-	$\geq$ B	-	-
C	$\{f_1, f_3\}$	$\geq$ E	-	$\geq$ C	-	1
Result		$\geq$ C	$\geq$ C	$\geq$ B	$\geq$ D	
Cost		29				

Whoopsie, f_1 and f_3 are not independent

$\Rightarrow$ Any impact on last allocation?

DAL Allocation: Degradation rules

f_1, f_3 not independent $\Rightarrow$ replace them by a new function failure $f_{1,3}$.

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	

DAL Allocation: Degradation rules

f_1, f_3 not independent $\Rightarrow$ replace them by a new function failure $f_{1,3}$.

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	
B	$\{f_{1,3}, f_2, f_4\}$	$\geq$ C	$\geq$ C	-	$\geq$ D	2

DAL Allocation: Degradation rules

f_1, f_3 not independent $\Rightarrow$ replace them by a new function failure $f_{1,3}$.

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	
B	$\{f_{1,3}, f_2, f_4\}$	$\geq$ C	$\geq$ C	-	$\geq$ D	2
	$\{f_{1,3}\}$	-	-	$\geq$ B	-	-

DAL Allocation: Degradation rules

f_1, f_3 not independent $\Rightarrow$ replace them by a new function failure $f_{1,3}$.

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	
B	$\{f_{1,3}, f_2, f_4\}$	$\geq$ C	$\geq$ C	-	$\geq$ D	2
	$\{f_{1,3}\}$	-	-	$\geq$ B	-	-
C	$\{f_{1,3}\}$	$\geq$ C	-	$\geq$ C	-	-

DAL Allocation: Degradation rules

f_1, f_3 not independent $\Rightarrow$ replace them by a new function failure $f_{1,3}$.

basic DAL	cuts	DAL				Option
		f_1	f_2	f_3	f_4	
B	$\{f_{1,3}, f_2, f_4\}$	$\geq$ C	$\geq$ C	-	$\geq$ D	2
	$\{f_{1,3}\}$	-	-	$\geq$ B	-	-
C	$\{f_{1,3}\}$	$\geq$ C	-	$\geq$ C	-	-
Result		$\geq$ C	$\geq$ C	$\geq$ B	$\geq$ D	
Cost		29				

Your turn! Allocate the DAL of green system.

DAL Allocation: Exercise

Assume FC is Major, all independent except *EMP* and *eng*1, and DAL cost for *EDP* and *elec* is twice the initial cost.

basic DAL	cuts	DAL						Option
		dist	*rsv*	*EMP*	*EDP*	*eng*1	*elec*	
?	$\{dist\}$	$\geq$?	-	-	-	-	-	?
	$\{rsv\}$	-	$\geq$?	-	-	-	-	?
	$\{EMP, EDP\}$	-	-	$\geq$?	$\geq$?	-	-	?
	$\{EMP, eng1\}$	-	-	$\geq$?	-	$\geq$?	-	?
	$\{elec, EDP\}$	-	-	-	$\geq$?	-	$\geq$?	?
	$\{elec, eng1\}$	-	-	-	-	$\geq$?	$\geq$?	?
Result		$\geq$?	$\geq$?	$\geq$?	$\geq$?	$\geq$?	$\geq$?	
Cost				?				

DAL Allocation: Exercise

Assume FC is Major, all independent except *EMP* and *eng*1, and DAL cost for *EDP* and *elec* is twice the initial cost.

basic DAL	cuts	DAL						Option
		dist	*rsv*	*EMP*	*EDP*	*eng*1	*elec*	
C	$\{dist\}$	≥ C	-	-	-	-	-	-
	$\{rsv\}$	-	≥ C	-	-	-	-	-
	$\{f_{EMP,eng1}, EDP\}$	-	-	≥ C	≥ E	-	-	1
	$\{f_{EMP,eng1}\}$	-	-	≥ C	-	≥ C	-	-
	$\{elec, EDP\}$	-	-	-	≥ D	-	≥ D	2
	$\{elec, f_{EMP,eng1}\}$	-	-	-	-	≥ C	≥ E	1
Result		≥ C	≥ C	≥ C	≥ D	≥ C	≥ D	
Cost		36						

What about IDAL?

DAL Allocation: IDAL

- IDAL is derivated from the FDAL of the functions implemented by the item
- Same rules as FDAL but cannot downgrade DAL twice (in function and item)

Why should we avoid double downgrade?

DAL Allocation: IDAL

- Let FC be a CAT and $MCS_{fc} = \{\{f_1, f_2, f_3\}\}$ where f_i are mutually independent.
- Each f_i needs at least one item $i_i^{f_i}$ and all items are independent.
- What is the IDAL of $i_i^{f_i}$ without no double downgrade rule?

DAL Allocation: IDAL

- Let FC be a CAT and $MCS_{fc} = \{\{f_1, f_2, f_3\}\}$ where f_i are mutually independent.
- Each f_i needs at least one item $i_i^{f_i}$ and all items are independent.
- What is the IDAL of $i_i^{f_i}$ without no double downgrade rule?
- Apply option 1 on FDAL $\Rightarrow$ $FDAL(f_1) = B, FDAL(f_2) = B, FDAL(f_3) = C$
- Apply option 1 on IDAL $\Rightarrow$ $IDAL(i_1^{f_1}) = C, IDAL(i_2^{f_1}) = C, \cdots$

DAL Allocation: IDAL

- Let FC be a CAT and $MCS_{fc} = \{\{f_1, f_2, f_3\}\}$ where f_i are mutually independent.
- Each f_i needs at least one item $i_i^{f_i}$ and all items are independent.
- What is the IDAL of $i_i^{f_i}$ without no double downgrade rule?
- Apply option 1 on FDAL $\Rightarrow$ $FDAL(f_1) = B, FDAL(f_2) = B, FDAL(f_3) = C$
- Apply option 1 on IDAL $\Rightarrow$ $IDAL(i_1^{f_1}) = C, IDAL(i_2^{f_1}) = C, \cdots$

Functions contributing to highly critical FC (Cat) implemented by low development assurance level items (Major)

Now a Recap

Today's lesson in 30"

Deal with dependencies

During design Trace independence assumptions during assessment ⇒ became requirements during implementation

During verification Identify the potential sources of dependencies & integrate them in safety assessment

Today's lesson in 30"

Emphasis on systematic errors:

- Currently, avoid systematic faults with design assurance level (DAL)
- DAL allocation depends on:
  - criticality of functions/items failures,
  - independence between them,
  - cost of DAL related activities.

You understand highlighted terms

$\Rightarrow$ congratulations you've got the idea

Otherwise check out the slides !

Let’s talk about the (your) future!

What are the new safety challenges?

What are the new safety challenges?

Let's have a quick (and non-exhaustive) overview!

From I to AI

Trend Huge trend to automate complex tasks preformed by operators (professional or not)

Breakdown New technologies involving complex sensor fusion or image processing

From I to AI

Trend Huge trend to automate complex tasks preformed by operators (professional or not)

Breakdown New technologies involving complex sensor fusion or image processing

What are the risks related to the massive adoption of such systems?

An Example Automotive anti-collision system

`https://youtu.be/ZMFbMV5QNzk?t=81`

Challenge 1: Trust Me I Am Autonomous

- Classical software correctness demonstrated by:
  1. validation: the specification breakdown is sound, complete and testable (ABS example)
  2. verification: the implementation is compliant to the specification (Offshore example)
- V&V achieved thanks to testing, traceability and formal verification

Challenge 1: Trust Me I Am Autonomous

- Classical software correctness demonstrated by:
  1. validation: the specification breakdown is sound, complete and testable (ABS example)
  2. verification: the implementation is compliant to the specification (Offshore example)
- V&V achieved thanks to testing, traceability and formal verification

What is the specification breakdown of an AI-based pedestrian detection system?

How to provide confidence on safety integrity for critical function based on AI?

Challenge 2: Taking into account new failures

- Safety impact of hardware failure addressed in safety critical systems (redundancy, mutual checks, lock-step)

Challenge 2: Taking into account new failures

- Safety impact of hardware failure addressed in safety critical systems (redundancy, mutual checks, lock-step)

What is the safety impact of an hardware failure executing AI-based software?

Can we detect & manage this failure?

Challenge 2: Taking into account new failures

- Safety impact of hardware failure addressed in safety critical systems (redundancy, mutual checks, lock-step)

What is the safety impact of an hardware failure executing AI-based software?
Can we detect & manage this failure?

ANITI PhD proposal: We are seeking for answers, perhaps from you!

Challenge 3: Safe integration of tomorrow aircrafts

- Various applicative domains can benefit from new aircraft concepts (VTOL, UAV, ...)
  - Infrastructure inspection (SCNF, ERDF, ...)
  - Package delivery (Amazon, CDiscount, La Poste, ...)
  - Flying taxi (Airbus' Vahana project, Boeing, Uber, ...)

Challenge 3: Safe integration of tomorrow aircrafts

- Various applicative domains can benefit from new aircraft concepts (VTOL, UAV, . . .)
  - Infrastructure inspection (SCNF, ERDF, . . .)
  - Package delivery (Amazon, CDiscount, La Poste, . . .)
  - Flying taxi (Airbus' Vahana project, Boeing, Uber, . . .)

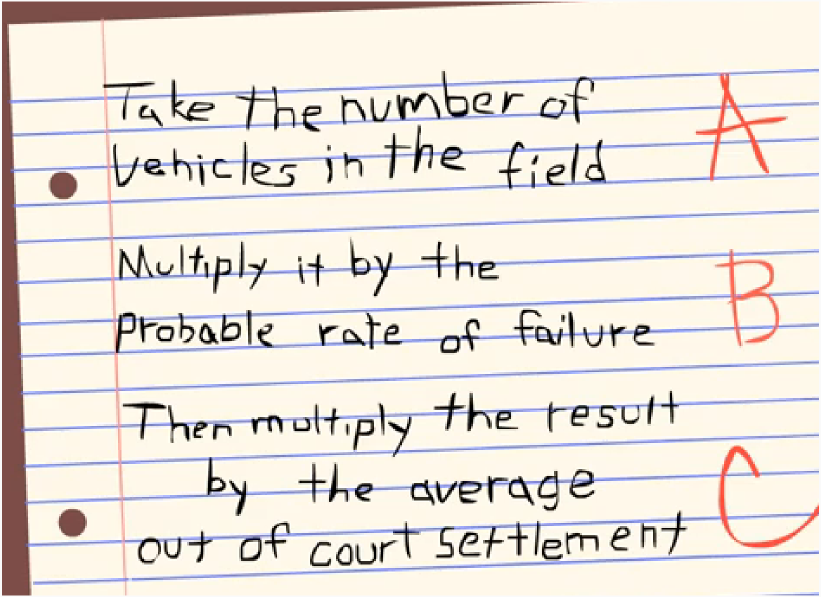
What are the new risks related to the integration of such aircraft in the flight traffic?

How to adapt safety analyses to take into account distributed procedures, autonomous avoidance systems?

ONERA Master Intership proposals

Join us to work on:

- pilot/UAV interactions :
  https://w3.onera.fr/stages/sites/w3.onera.fr.stages/files/dtis-2020-23.pdf
- assessment of on-ground collision probability
  https://w3.onera.fr/stages/sites/w3.onera.fr.stages/files/dtis-2020-31.pdf


Take the number of vehicles in the field
A
Multiply it by the probable rate of failure
B
Then multiply the result by the average out of court settlement
C