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!
Main mission  Drone monitors flock and prevents bear attack
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
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 erroneous 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:

- 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
Possible hazards:

- Hurt the flock
- Collision with vehicle (road)
Possible hazards:

- Hurt the flock
- Collision with vehicle (road)

Possible adversary conditions:

- Wind or Rain $\Rightarrow$ drone can’t fly
- Poor GNSS signal $\Rightarrow$ 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 ($\Lambda$))**
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 $R$:

$$R(t) = e^{-\int_0^t \Lambda(u) du}$$
Assume items used during constant failure rate phase

\[
\Lambda(t) = \lambda; \quad R(t) = e^{-\lambda t}; \quad MTTF = \frac{1}{\lambda}
\]
Definition (Rare failure assumptions)
When $\lambda t \sim 0$ (usually $\lambda t < .1$) use Taylor expansion for computations:

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

Definition (Independence & pessimism assumption)
If two components $C_1$ and $C_2$ have independent failures with failure rate $\lambda_1$ and $\lambda_2$

\[
\begin{align*}
    p(\text{both fail}) & = p(C_1 \text{ fails})p(C_2 \text{ fails}) = \lambda_1 \lambda_2 t^2 \\
    p(\text{one fails}) & = p(C_1 \text{ fails}) + p(C_2 \text{ fails}) - p(\text{both fail}) \\
                        & = p(C_1 \text{ fails}) + p(C_2 \text{ fails})
\end{align*}
\]
The question is:

What happens if drone shepherd fails?
The question is:

What happens if drone shepherd fails?
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.
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

<table>
<thead>
<tr>
<th>severity class</th>
<th>effects description</th>
<th>acceptable frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>catastrophic</td>
<td>prevent continuous safe flight and landing: aircraft loss and loss of crew and passengers</td>
<td>$&lt; 10^{-9}$ per flight hour and no single failure leads to the FC</td>
</tr>
<tr>
<td>hazardous</td>
<td>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</td>
<td>$&lt; 10^{-7}$ per flight hour</td>
</tr>
</tbody>
</table>
## Risk acceptability for civil aircraft

<table>
<thead>
<tr>
<th>severity class</th>
<th>effects description</th>
<th>acceptable frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>major</td>
<td>significant reduction in safety margin or functional capabilities or significant increase in crew workload or discomfort to occupants possibly including injuries</td>
<td>$&lt; 10^{-5}$ per flight hour</td>
</tr>
<tr>
<td>minor</td>
<td>no significant reduction in aircraft safety.</td>
<td>$&lt; 10^{-3}$ per flight hour</td>
</tr>
<tr>
<td>no safety effect</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Risk acceptability for civil aircraft

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

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
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)

Safety Process

- Aircraft Level Functional Hazard Analysis (FHA)
- System Level FHA
- Preliminary System Safety Assessment (PSSA)
- System Safety Assessment (SSA)
- Aircraft Safety Synthesis
- Common Cause Analysis (CCA)

Design Process

- Aircraft Level Requirements
- Allocation of aircraft functions to systems
- Development of system architecture
- Allocation of requirements to hardware and software
- System implementation
- Physical system

Functional Interactions
- Failure conditions, safety objectives
- Architectural requirements
- Item requirements, safety objectives, analyses required
- Implementation

Results
- Failure conditions and effects
- Separation requirements
- Separation verification

Certification
When should we identify and classify Failure Conditions?
Safety Process (FHA)

Safety Process

Aircraft Level Functional Hazard Analysis (FHA)

System Level FHA

Failure conditions, safety objectives

Preliminary System Safety Assessment (PSSA)

System Safety Assessment (SSA)

System functions

Results

Implementation

Allocation of requirements to hardware and software

System implementation

Physical system

Aircraft functions

Allocation of aircraft functions to systems

Development of system architecture

Aircraft Level Requirements

Common Cause Analysis (CCA)

Failure conditions and effects

Functional Interactions

Separation requirements

Item requirements, safety objectives, analyses required

System architecture

Architectural requirements

Certification

Separation verification
Figure 3: Functional breakdown (cf SysML lesson)

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

- Maintain in flight
- Maintain trajectory in evolution zone
- Monitor drone state
- Abort Flight

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

Table 1: FHA example

<table>
<thead>
<tr>
<th>Function</th>
<th>Failure</th>
<th>Context</th>
<th>Consequences</th>
<th>Criticality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain trajectory loss</td>
<td></td>
<td>cannot abort flight</td>
<td></td>
<td></td>
</tr>
<tr>
<td>in evolution zone</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### FHA by the example

<table>
<thead>
<tr>
<th>Function</th>
<th>Failure</th>
<th>Context</th>
<th>Consequences</th>
<th>Criticality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain trajectory in evolution zone</td>
<td>loss</td>
<td>cannot abort flight</td>
<td>Crash outside evolution zone, possible collision with vehicles</td>
<td>Catastrophic</td>
</tr>
</tbody>
</table>

**Table 1:** FHA example
### FHA by the example

<table>
<thead>
<tr>
<th>Function</th>
<th>Failure</th>
<th>Context</th>
<th>Consequences</th>
<th>Criticality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain trajectory in evolution zone</td>
<td>loss</td>
<td>cannot abort flight</td>
<td>Crash outside evolution zone, possible collision with vehicles</td>
<td>Catastrophic</td>
</tr>
<tr>
<td>Maintain in flight</td>
<td>loss</td>
<td>can maintain in evolution zone</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1:** FHA example
## FHA by the example

<table>
<thead>
<tr>
<th>Function</th>
<th>Failure</th>
<th>Context</th>
<th>Consequences</th>
<th>Criticality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain trajectory in evolution zone</td>
<td>loss</td>
<td>cannot abort flight</td>
<td>Crash outside evolution zone, possible collision with vehicles</td>
<td>Catastrophic</td>
</tr>
<tr>
<td>Maintain in flight</td>
<td>loss</td>
<td>can maintain in evolution zone</td>
<td>Crash in evolution zone, possible hurt flock</td>
<td>Hazardous</td>
</tr>
</tbody>
</table>

**Table 1: FHA example**
## FHA by the example

<table>
<thead>
<tr>
<th>Function</th>
<th>Failure</th>
<th>Context</th>
<th>Consequences</th>
<th>Criticality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain trajectory in evolution zone</td>
<td>loss</td>
<td>cannot abort flight</td>
<td>Crash outside evolution zone, possible collision with vehicles</td>
<td><strong>Catastrophic</strong></td>
</tr>
<tr>
<td>Maintain in flight</td>
<td>loss</td>
<td>can maintain in evolution zone</td>
<td>Crash in evolution zone, possible hurt flock</td>
<td><strong>Hazardous</strong></td>
</tr>
<tr>
<td>Abort Flight</td>
<td>loss</td>
<td>can maintain in evolution zone</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1:** FHA example

- Drone behaves properly $\Rightarrow$ No safety effect
# FHA by the example

<table>
<thead>
<tr>
<th>Function</th>
<th>Failure</th>
<th>Context</th>
<th>Consequences</th>
<th>Criticality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain trajectory in evo-</td>
<td>loss</td>
<td>cannot abort flight</td>
<td>Crash outside evolution zone, possible collision with</td>
<td><strong>Catastrophic</strong></td>
</tr>
<tr>
<td>lution zone</td>
<td></td>
<td></td>
<td>vehicles</td>
<td></td>
</tr>
<tr>
<td>Maintain in flight</td>
<td>loss</td>
<td>can maintain in evolution zone</td>
<td>Crash in evolution zone, possible hurt flock</td>
<td><strong>Hazardous</strong></td>
</tr>
<tr>
<td>Abort Flight</td>
<td>loss</td>
<td>can maintain in evolution zone</td>
<td>Drone behaves properly ⇒ No safety effect</td>
<td><strong>NSE</strong></td>
</tr>
</tbody>
</table>

**Table 1: FHA example**
**Failure condition** Combination of functional failures that have an effect on system’s safety
## FHA by the example: Failure conditions

**Failure condition** Combination of functional failures that have an effect on system’s safety

<table>
<thead>
<tr>
<th>Function</th>
<th>Failure</th>
<th>Context</th>
<th>Consequences</th>
<th>Criticality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain trajectory in evolution zone</td>
<td>loss</td>
<td>cannot abort flight</td>
<td>Crash outside evolution zone, possible collision with vehicles</td>
<td><strong>Catastrophic</strong></td>
</tr>
</tbody>
</table>


FHA by the example: Failure conditions

**Failure condition**  Combination of functional failures that have an effect on system’s safety

<table>
<thead>
<tr>
<th>Function</th>
<th>Failure</th>
<th>Context</th>
<th>Consequences</th>
<th>Criticality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain trajectory in evolution zone</td>
<td>loss</td>
<td>cannot abort flight</td>
<td>Crash outside evolution zone, possible collision with vehicles</td>
<td><strong>Catastrophic</strong></td>
</tr>
</tbody>
</table>

**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 on a system’s safety

<table>
<thead>
<tr>
<th>Function</th>
<th>Failure</th>
<th>Context</th>
<th>Consequences</th>
<th>Criticality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain in flight</td>
<td>loss</td>
<td>can maintain in evolution zone</td>
<td>Crash in evolution zone, possible hurt flock</td>
<td>Hazardous</td>
</tr>
</tbody>
</table>

**CAT_SOL**  cannot maintain trajectory in evolution zone and cannot abort flight
**Failure condition** Combination of functional failures that have an effect on the system’s safety

<table>
<thead>
<tr>
<th>Function</th>
<th>Failure</th>
<th>Context</th>
<th>Consequences</th>
<th>Criticality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain in flight</td>
<td>loss</td>
<td>can maintain in evolution zone</td>
<td>Crash in evolution zone, possible hurt flock</td>
<td>Hazardous</td>
</tr>
</tbody>
</table>

**CAT_SOL** cannot maintain trajectory in evolution zone and cannot abort flight

**HAZ_SOL** cannot maintain in flight
Failure condition Combination of functional failures that have an effect on the system’s safety

<table>
<thead>
<tr>
<th>Function</th>
<th>Failure</th>
<th>Context</th>
<th>Consequences</th>
<th>Criticality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abort Flight</td>
<td>loss</td>
<td>can maintain in evolution zone</td>
<td>Drone behaves properly ⇒ No safety effect</td>
<td>NSE</td>
</tr>
</tbody>
</table>

**CAT_SOL** cannot maintain trajectory in evolution zone and cannot abort flight

**HAZ_SOL** cannot maintain in flight
Safety objectives bounds over indicators commensurate with failure condition criticality

Example (Safety objectives)
What are the safety objectives for CAT_SOL?
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 ≤ 10^{-7}
When should we check dependability requirements?
Figure 5: Failure conditions and functional architectures
Acquire Data each data acquired by independent function, failure modes are:

- erroneous: send inconsistent data,
- lost: stop sending data.
Monitor drone state each data is checked by independent and perfect alarms (neglected failures in the Lab but not in real life !!!).
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".

\[ \text{Mission} < \text{Hovering} < \text{Landing} < \text{Flight Termination} \]

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

Intermediate Event 1

$p_e$ $PE_1$

Intermediate Event 2

$	ext{and}$ $PE_3$ $PE_2$

Intermediate Event 3

$	ext{or}$ $PE_3$ $PE_2$

$e$ primary event

$e$ Intermediate event

$e$ copy of primary event
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?
Fault tree $\Leftrightarrow$ formula $\varphi$ 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 $\varphi$ 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
**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:

\[
\text{order}(FC) = \min_{c \in \text{MCS}_FC} (|c|)
\]

**Example (Order)**
For \(\text{MCS}_{FC} = \{\{a, b\}, \{c\}\}\) we have:

\[
\text{order}(FC) = \min_{c \in \text{MCS}_{FC}} (|c|) \\
= \min(|\{a, b\}|, |\{c\}|) \\
= 1
\]
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)

<table>
<thead>
<tr>
<th>Criticality</th>
<th>Qualitative Requirement</th>
<th>Quantitative Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catastrophic</td>
<td>order ≥ 2</td>
<td>p ≤ 10^{-9} / flight hour</td>
</tr>
<tr>
<td>Hazardous</td>
<td>order ≥ 1</td>
<td>p ≤ 10^{-7} / flight hour</td>
</tr>
<tr>
<td>Major</td>
<td>order ≥ 1</td>
<td>p ≤ 10^{-5} / flight hour</td>
</tr>
<tr>
<td>Minor</td>
<td>order ≥ 1</td>
<td>p ≤ 10^{-3} / flight hour</td>
</tr>
</tbody>
</table>

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
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
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?
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?

\[
\text{order}(FC) = \min_{c \in MCS_{FC}} (|c|) = 1 \Rightarrow OK
\]
\[
p(FC) = p(a)p(b) + p(c) \approx 10^{-4} > 10^{-5} \Rightarrow KO
\]
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
General overview

Specification of functional, logical and physical architectures with SysML

Figure 8: Dysfunctional analysis in development process
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.
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.**
SAE. Aerospace Recommended Practices 4761 - guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment, 1996.

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

$\Rightarrow C \rightarrow \{d, c\}$

$\Rightarrow$ 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

 Req 1: distg, disty and distb need independence

 Req 2: three functions out of EDPy, EMPg EDPg and distb must be independent

Cut 1

Cut 2

Order requirement (3)

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
**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

<table>
<thead>
<tr>
<th>Description</th>
<th>Objective</th>
<th>Ref</th>
<th>Applicability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Software high-level requirements comply with system requirements.</td>
<td></td>
<td>6.3.1a</td>
<td>I I R R</td>
</tr>
<tr>
<td>2 High-level requirements are accurate and consistent.</td>
<td></td>
<td>6.3.1b</td>
<td>I I R R</td>
</tr>
<tr>
<td>3 High-level requirements are compatible with target computer.</td>
<td></td>
<td>6.3.1c</td>
<td>R R</td>
</tr>
</tbody>
</table>

- High DAL level $\Rightarrow$ great number of assurance activities
  $\Rightarrow$ costly
- $\Rightarrow$ minimize the DAL of software and hardware
Based on the severities of the FCs that function fault contributes to.

<table>
<thead>
<tr>
<th>Sev(FC)</th>
<th>DAL(FC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT</td>
<td>A</td>
</tr>
<tr>
<td>HAZ</td>
<td>B</td>
</tr>
<tr>
<td>MAJ</td>
<td>C</td>
</tr>
<tr>
<td>MIN</td>
<td>D</td>
</tr>
<tr>
<td>NSE</td>
<td>E</td>
</tr>
</tbody>
</table>

Table 3: Link between severity and DAL
What does "the severities of the FCs that function fault $f$ contributes to" mean?

⇒ 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 $f_1$?
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 $DAL(f_1) = \text{worst}(DAL(fc_1), DAL(fc_2)) = DAL(HAZ) = B$

Question: What is the basic DAL of $f_2$?
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 $DAL(f_1) = \text{worst}(DAL(fc_1), DAL(fc_2)) = DAL(\text{HAZ}) = B$

Question: What is the basic DAL of $f_2$?

Answer: $f_2$ contained only in $MCS_1$ so $DAL(f_2) = \text{worst}(DAL(fc_1)) = DAL(\text{HAZ}) = B$
Designers can downgrade the basic DAL \textit{basic} of a function using independence, the allocation must fulfill the following rules:

**Rule 1** \textit{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 \( DAL(f_i) = DAL(f_j) = basic - 1 \)
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

<table>
<thead>
<tr>
<th>basic DAL cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_3$</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>${f_1, f_2, f_4}$</td>
<td>$\geq B$</td>
<td>$\geq D$</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>${f_1, f_2, f_4}$</td>
<td>$\geq B$</td>
<td>$\geq D$</td>
</tr>
<tr>
<td></td>
<td>${f_3}$</td>
<td>-</td>
<td>$\geq B$</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>${f_1, f_2, f_4}$</td>
<td>$\geq B$</td>
<td>$\geq D$</td>
<td>-</td>
<td>$\geq D$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>${f_3}$</td>
<td>-</td>
<td>-</td>
<td>$\geq B$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>${f_1, f_3}$</td>
<td>$\geq C$</td>
<td>-</td>
<td>$\geq E$</td>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_3$</td>
</tr>
<tr>
<td>B</td>
<td>${f_1, f_2, f_4}$</td>
<td>$\geq B$</td>
<td>$\geq D$</td>
</tr>
<tr>
<td></td>
<td>${f_3}$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>${f_1, f_3}$</td>
<td>$\geq C$</td>
<td>-</td>
</tr>
<tr>
<td>Result</td>
<td>$\geq B$</td>
<td>$\geq D$</td>
<td>$\geq B$</td>
</tr>
<tr>
<td>Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Is it the cheapest option?

⇒ Let’s try again!
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

<table>
<thead>
<tr>
<th>basic DAL cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_1$</td>
<td>$f_2$</td>
</tr>
</tbody>
</table>
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 $E = 0$

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>DAL 1</th>
<th>DAL 2</th>
<th>DAL 3</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>${f_1, f_2, f_4}$</td>
<td>$\geq C$</td>
<td>$\geq C$</td>
<td>-</td>
<td>$\geq D$</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_3$</td>
</tr>
<tr>
<td>B</td>
<td>${f_1, f_2, f_4}$</td>
<td>$\geq C$</td>
<td>$\geq C$</td>
</tr>
<tr>
<td></td>
<td>${f_3}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>${f_1, f_2, f_4}$</td>
<td>$\geq C$</td>
<td>$\geq C$</td>
</tr>
<tr>
<td></td>
<td>${f_3}$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>C</td>
<td>${f_1, f_3}$</td>
<td>$\geq E$</td>
<td>$-$</td>
</tr>
</tbody>
</table>
Suppose $f_1, f_2, f_3$ and $f_4$ are independent and cost:\nDAL A = 20, DAL B = 15, DAL C = 5, DAL D = 4, DAL E = 0

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$f_1$</td>
<td>$f_2$</td>
</tr>
<tr>
<td>B</td>
<td>{f_1, f_2, f_4}</td>
<td>$\geq$ C</td>
<td>$\geq$ C</td>
</tr>
<tr>
<td></td>
<td>{f_3}</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>{f_1, f_3}</td>
<td>$\geq$ E</td>
<td>-</td>
</tr>
<tr>
<td>Result</td>
<td></td>
<td>$\geq$ C</td>
<td>$\geq$ C</td>
</tr>
<tr>
<td>Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Whoopsie, $f_1$ and $f_3$ are not independent

⇒ Any impact on last allocation?
**DAL Allocation: Degradation rules**

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

<table>
<thead>
<tr>
<th>basic DAL cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_3$</td>
</tr>
</tbody>
</table>
$f_1, f_3$ not independent $\Rightarrow$ replace them by a new function failure $f_{1,3}$.

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_3$</td>
<td>$f_4$</td>
</tr>
<tr>
<td>B</td>
<td>${f_{1,3}, f_2, f_4}$</td>
<td>$\geq C$</td>
<td>$\geq C$</td>
</tr>
</tbody>
</table>
\[ f_1, f_3 \text{ not independent} \Rightarrow \text{replace them by a new function failure } f_{1,3}. \]

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>{f_{1,3}, f_2, f_4}</td>
<td>\begin{align*} f_1 &amp; \geq C \ f_2 &amp; \geq C \ f_3 &amp; - \ f_4 &amp; \geq D \end{align*}</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>{f_{1,3}}</td>
<td>\begin{align*} f_1 &amp; - \ f_2 &amp; - \ f_3 &amp; \geq B \ f_4 &amp; - \end{align*}</td>
<td>-</td>
</tr>
</tbody>
</table>
$f_1, f_3$ not independent $\Rightarrow$ replace them by a new function failure $f_{1,3}$.

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_3$</td>
</tr>
<tr>
<td>B</td>
<td>${f_{1,3}, f_2, f_4}$</td>
<td>$\geq C$</td>
<td>$\geq C$</td>
</tr>
<tr>
<td></td>
<td>${f_{1,3}}$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>${f_{1,3}}$</td>
<td>$\geq C$</td>
<td>-</td>
</tr>
</tbody>
</table>
### DAL Allocation: Degradation rules

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

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_1, f_2, f_4$</td>
<td>$f_1$</td>
<td>$f_2$</td>
</tr>
<tr>
<td>B</td>
<td>${f_{1,3}}$</td>
<td>$\geq C$</td>
<td>$\geq C$</td>
</tr>
<tr>
<td></td>
<td>${f_{1,3}}$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>${f_{1,3}}$</td>
<td>$\geq C$</td>
<td>-</td>
</tr>
<tr>
<td>Result</td>
<td></td>
<td>$\geq C$</td>
<td>$\geq C$</td>
</tr>
<tr>
<td>Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cost: 29
Your turn! Allocate the DAL of green system.
**DAL Allocation: Exercise**

Assume FC is Major, all independent except *EMP* and *eng1*, and DAL cost for *EDP* and *elec* is twice the initial cost.

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><em>dist</em></td>
<td><em>rsv</em></td>
</tr>
<tr>
<td>?</td>
<td></td>
<td>≥ ?</td>
<td>-</td>
</tr>
<tr>
<td>{dist}</td>
<td></td>
<td>-</td>
<td>≥ ?</td>
</tr>
<tr>
<td>{rsv}</td>
<td></td>
<td>-</td>
<td>≥ ?</td>
</tr>
<tr>
<td>{EMP, EDP}</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>{EMP, eng1}</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>{elec, EDP}</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>{elec, eng1}</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

|--------|-----|-----|-----|-----|-----|-----|

<table>
<thead>
<tr>
<th>Cost</th>
<th>?</th>
</tr>
</thead>
</table>
### DAL Allocation: Exercise

Assume FC is Major, all independent except $EMP$ and $eng1$, and DAL cost for $EDP$ and $elec$ is twice the initial cost.

<table>
<thead>
<tr>
<th>basic DAL</th>
<th>cuts</th>
</tr>
</thead>
<tbody>
<tr>
<td>{dist}</td>
<td></td>
</tr>
<tr>
<td>{rsv}</td>
<td></td>
</tr>
<tr>
<td>$C$</td>
<td></td>
</tr>
<tr>
<td>${f_{EMP,eng1}, EDP}$</td>
<td>$\geq C$</td>
</tr>
<tr>
<td>${f_{EMP,eng1}}$</td>
<td>-</td>
</tr>
<tr>
<td>${elec, EDP}$</td>
<td>-</td>
</tr>
<tr>
<td>${elec, f_{EMP,eng1}}$</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>dist</td>
<td>rsv</td>
</tr>
<tr>
<td>$\geq C$</td>
<td>-</td>
</tr>
<tr>
<td>$C$</td>
<td>$\geq C$</td>
</tr>
</tbody>
</table>

**Result**

| Cost | 36 |

**Cost**
What about IDAL?
IDAL 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?
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 $i^{f_i}_j$ and all items are independent.

What is the IDAL of $i^{f_i}_j$ without no double downgrade rule?
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 $i^{f_i}_i$ and all items are independent.
- What is the IDAL of $i^{f_i}_i$ 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 $\Rightarrow$
  $$IDAL(i^{f_1}_1) = C, IDAL(i^{f_1}_2) = C, \cdots$$

Functions contributing to highly critical FC (Cat) implemented by low development assurance level items (Major).
Now a Recap
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*
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 ⇒ 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!
**Trend**  Huge trend to automate complex tasks performed by operators (professional or not)

**Breakdown**  New technologies involving complex sensor fusion or image processing
**Trend**  Huge trend to automate complex tasks performed 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)

ANIITI PhD proposal: We are seeking for answers, perhaps from you!
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, ...)

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?
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?
Join us to work on:

- **pilot/UAV interactions**:

- **assessment of on-ground collision probability**
Take the number of vehicles in the field.

Multiply it by the probable rate of failure.

Then multiply the result by the average out of court settlement.