Definition and verification of functional safety concepts for the definition of safe logical architectures

Pierre Mauborgne, Éric Bonjour, Christophe Perrard, Samuel Deniaud,
Éric Levrat, Jean-Pierre Micaëlli, Dominique Loise

OUTLINE

- Context
- Industrial objective and research question
- Safe Logical Architecture Process Definition
- Application to a case study

Context

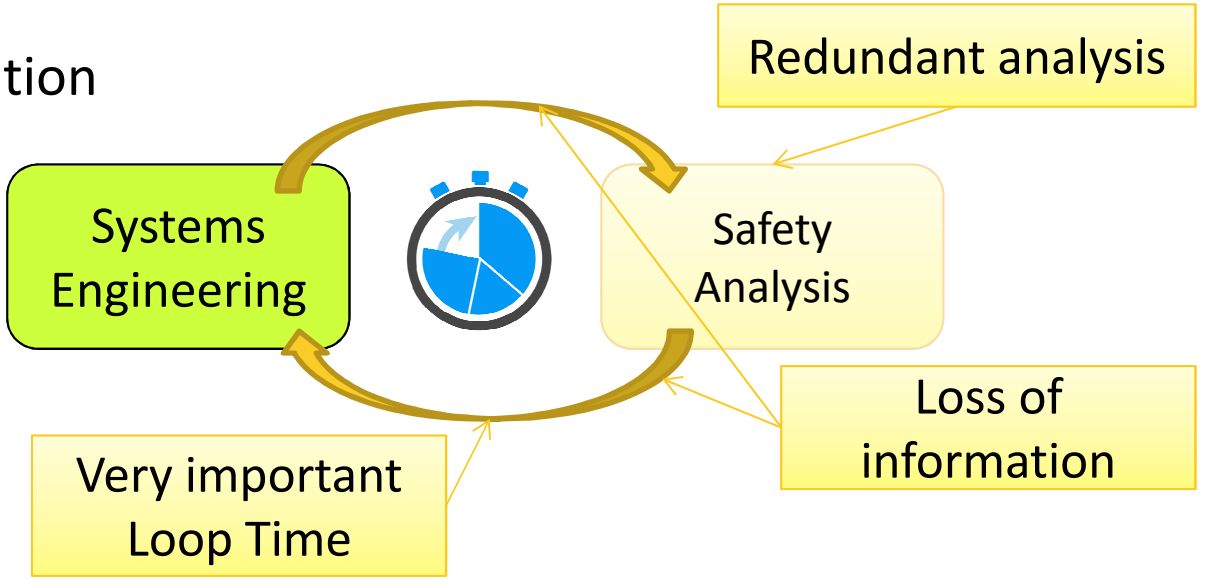
- Cars are more and more complex
  - Increasing number of Advanced Driver Assistance Systems (ADAS) and of electronic parts
  - More severe requirements :
    - Reduction of polluting emissions
    - Safety goals…

  ➔ Model-Based Systems Engineering has been introduced in car design offices. "Model-Based Systems Engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation, beginning in the conceptual design phase and continuing throughout development and later life cycle phases." *INCOSE MBSE, 2007*

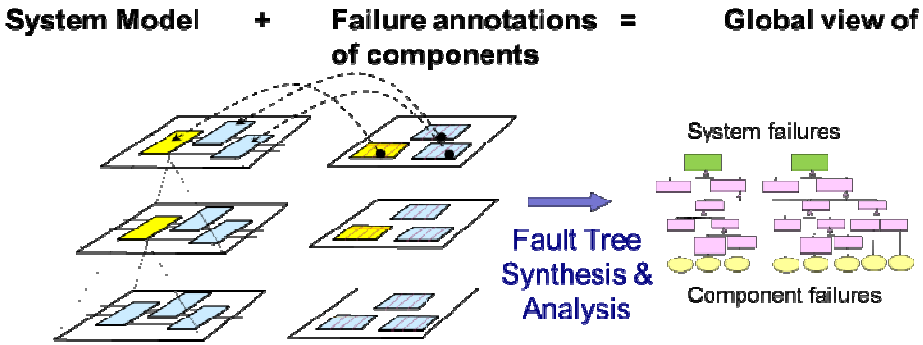
- Car designers must apply a safety standard (ISO 26262) that will guide them for the design of a safe system
  - Improved efficiency of safety analysis
  - Better traceability of safety requirements

Industrial objective

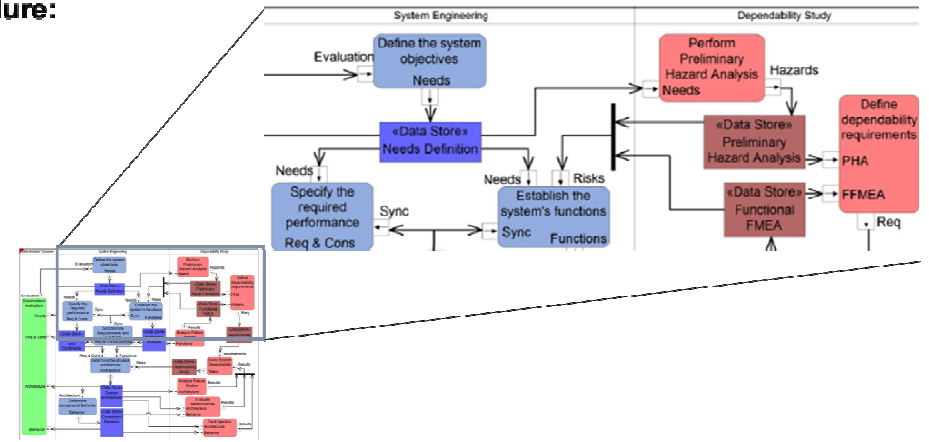
- Usual situation



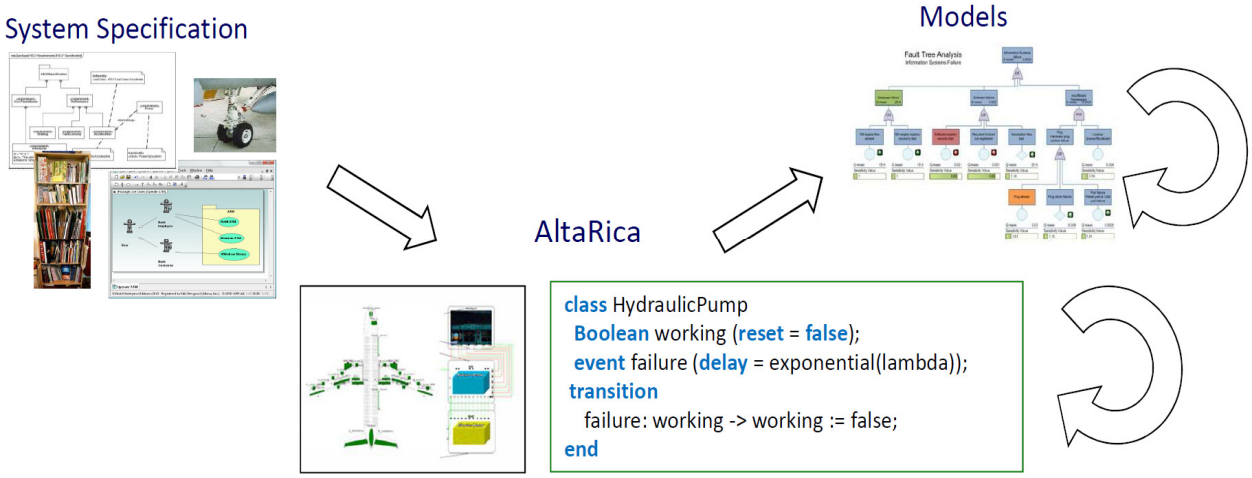
Research question



Extract useful informations from SE Models for **safety analysis** (Papadopoulos, 2013) (Lanusse, 2013)



Link SE and Safety processes to improve process exchanges (Cressent, 2012)



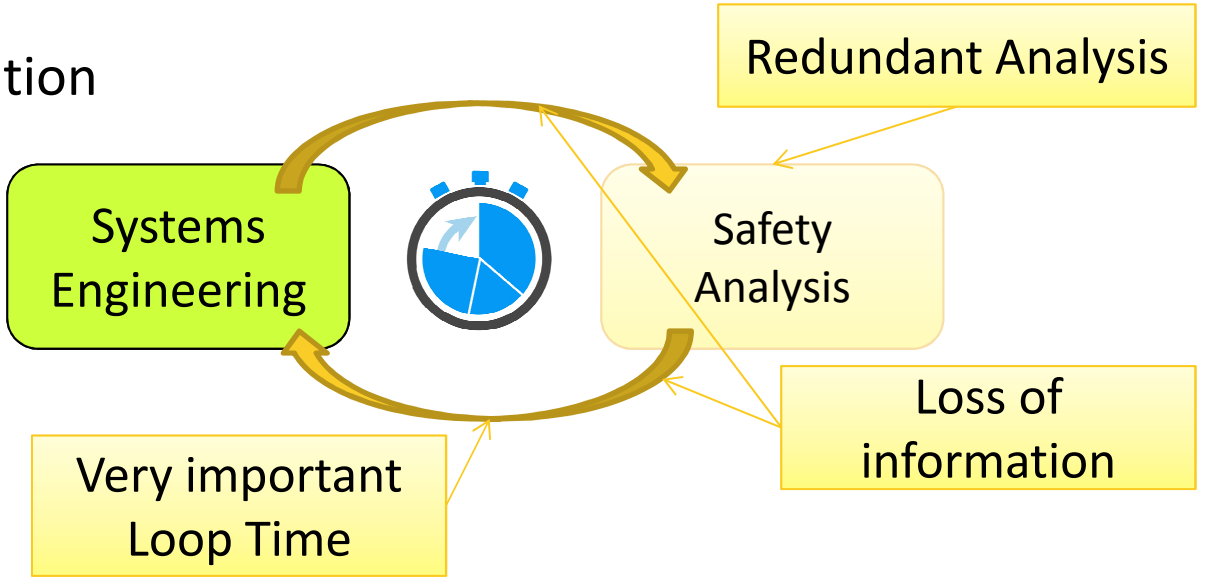
➔ *No safe SE process*

Research question: how to better integrate (Model-Based) Systems Engineering processes and safety studies?

Define **a pivot language** to articulate SE and Safety Domains (Rauzy, 2014)

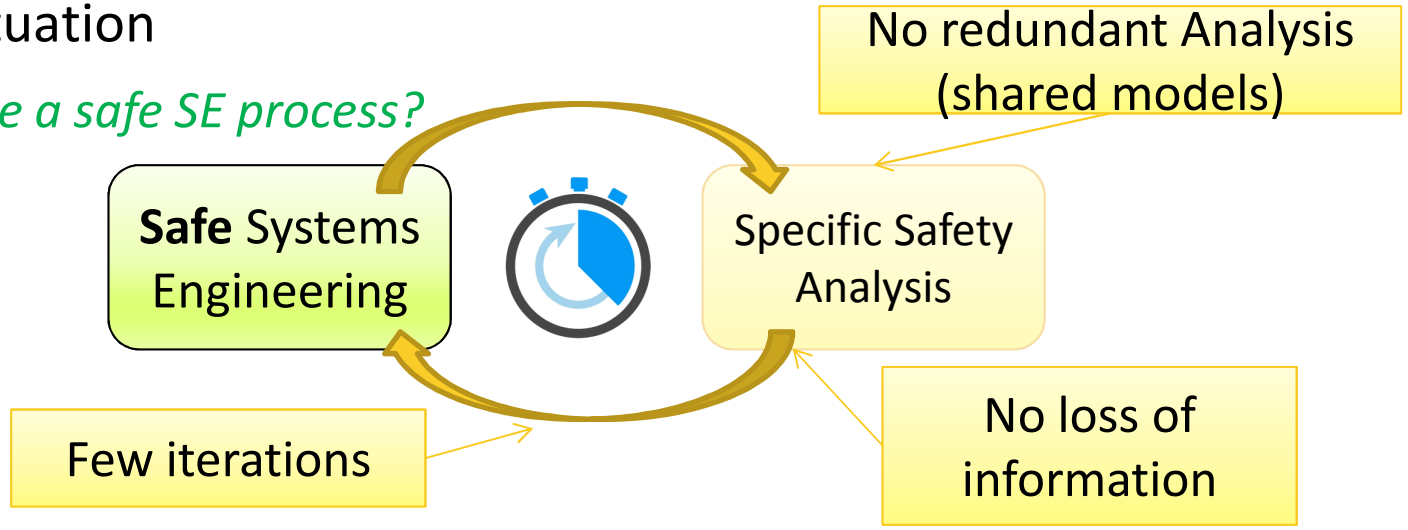
Industrial objective and research question

- Usual situation



- Target situation

➔ *How to define a safe SE process?*



A safe systems requirement engineering process

Hazard: potential source of harm
Harm: physical injury or damage to the health of people

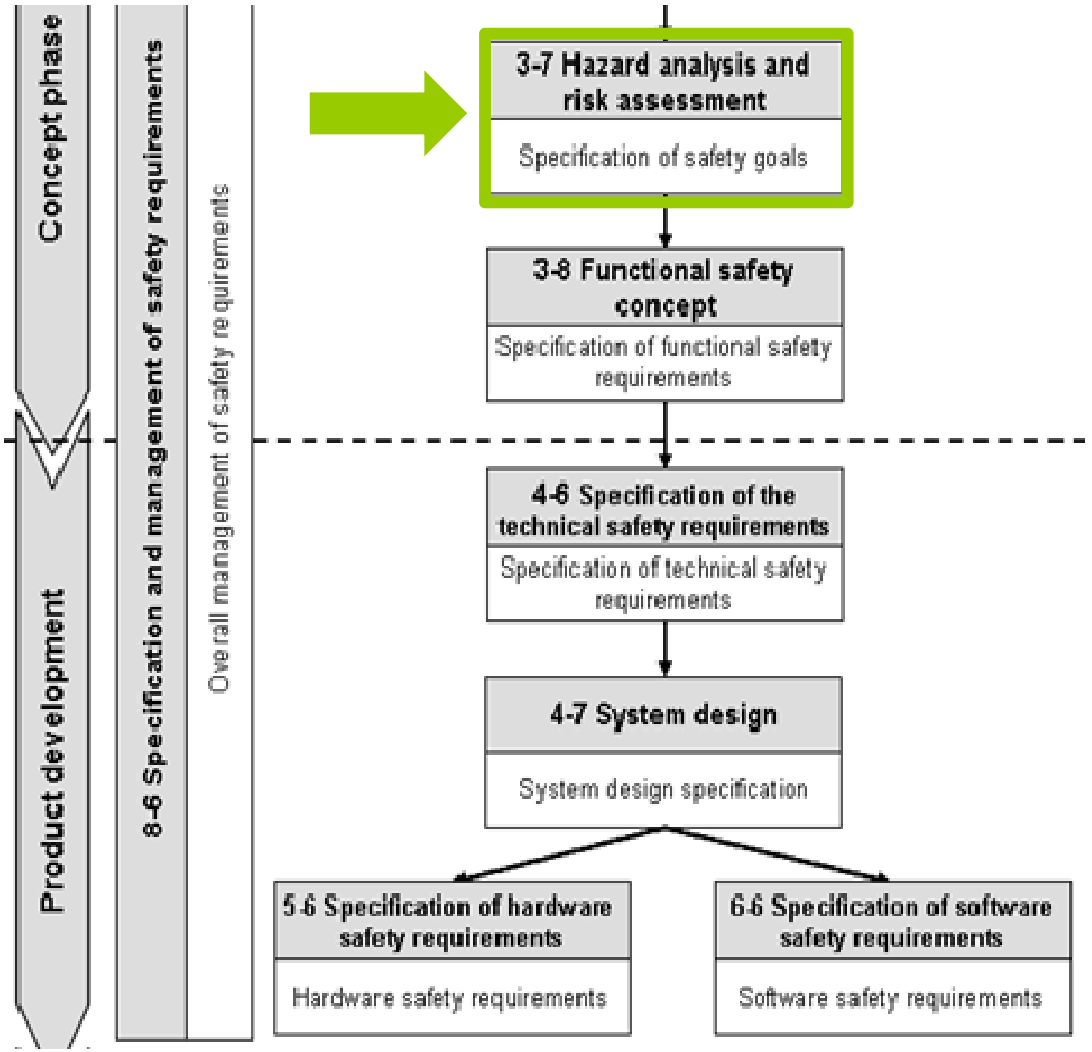
In previous work:
A method to specify **safety goals** and **ASILs** related to the prevention or mitigation of hazards.

Mauborgne, P., et al. (2016). Operational and system hazard analysis in a **safe systems requirement engineering process** – Application to automotive industry. ***Safety Science***, *87*, 256-268.

Automotive Safety Integrity Level (ASIL) is a measure of criticity of each hazard including the probability of exposure, controllability and severity, with A representing the least stringent level and D the most stringent one.

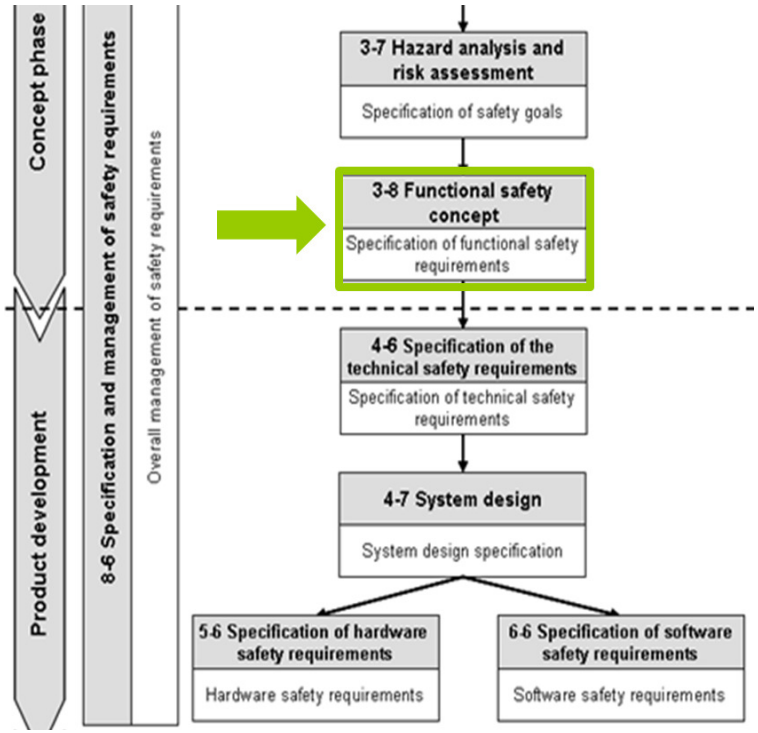
A **safety goal** is specified **for each hazard** with its respective ASIL.

Overview of the safety requirements process (ISO 26262)



Focus of this paper

- Based on the safety goals, a **functional safety concept** (ISO°26262-3) is specified considering preliminary architectural assumptions. The functional safety concept is detailed and specified by **functional safety requirements** that are allocated to the elements of the item.

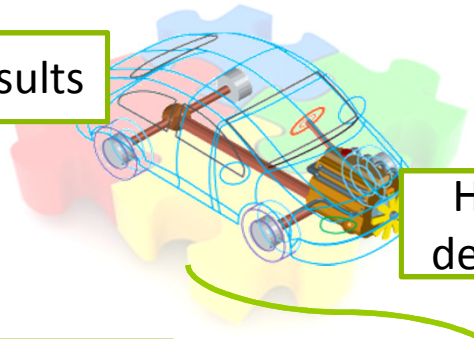


- **In this paper, we focus on the proposal of a Safe Logical Architecture Definition Process that enables to define the functional safety requirements**
- In the case study presented in this paper, we will limit the specification of a functional safety concept to the introduction of “safety functions”

Steps for the Safe Logical Architecture (SLA) Process Definition

Safe Logical Architecture

Expected results



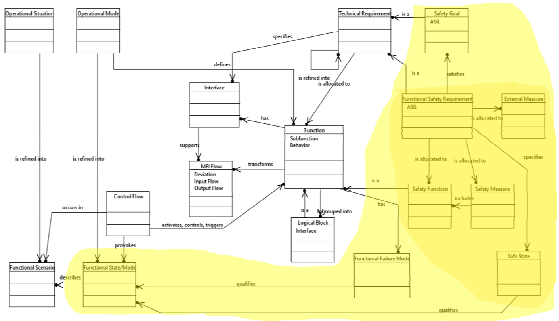
How to define it?

SSE Processes

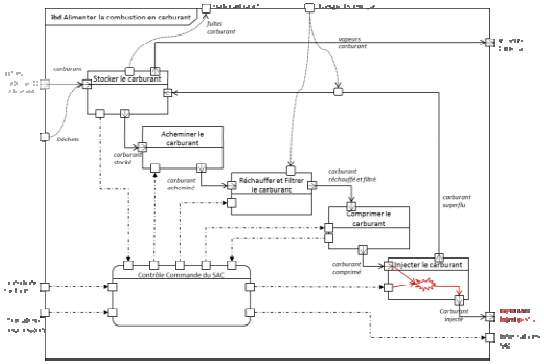
Which are the key concepts?

Which methods to integrate these activities?

SLA Conceptual Model

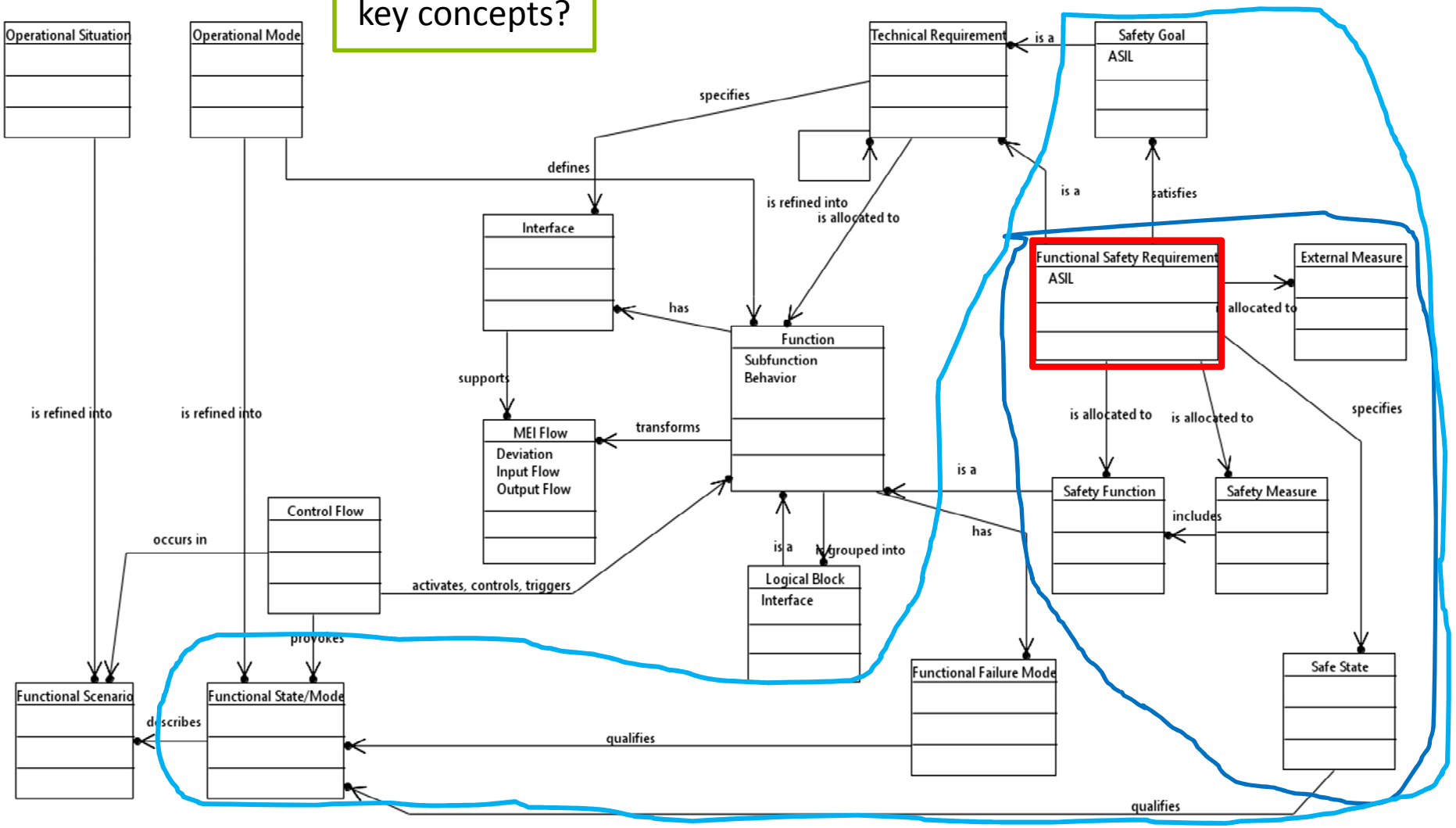


SLA Definition Method



Conceptual model – Logical View

Which are the key concepts?



functional safety requirement: specification of implementation-independent safety behavior, or safety measure

functional safety concept: specification of the **functional safety requirements**, with associated information, their assignment to **architectural elements**, and their interaction necessary to achieve the **safety goals**

Proposed Safe Logical Architecture (SLA) Definition Process

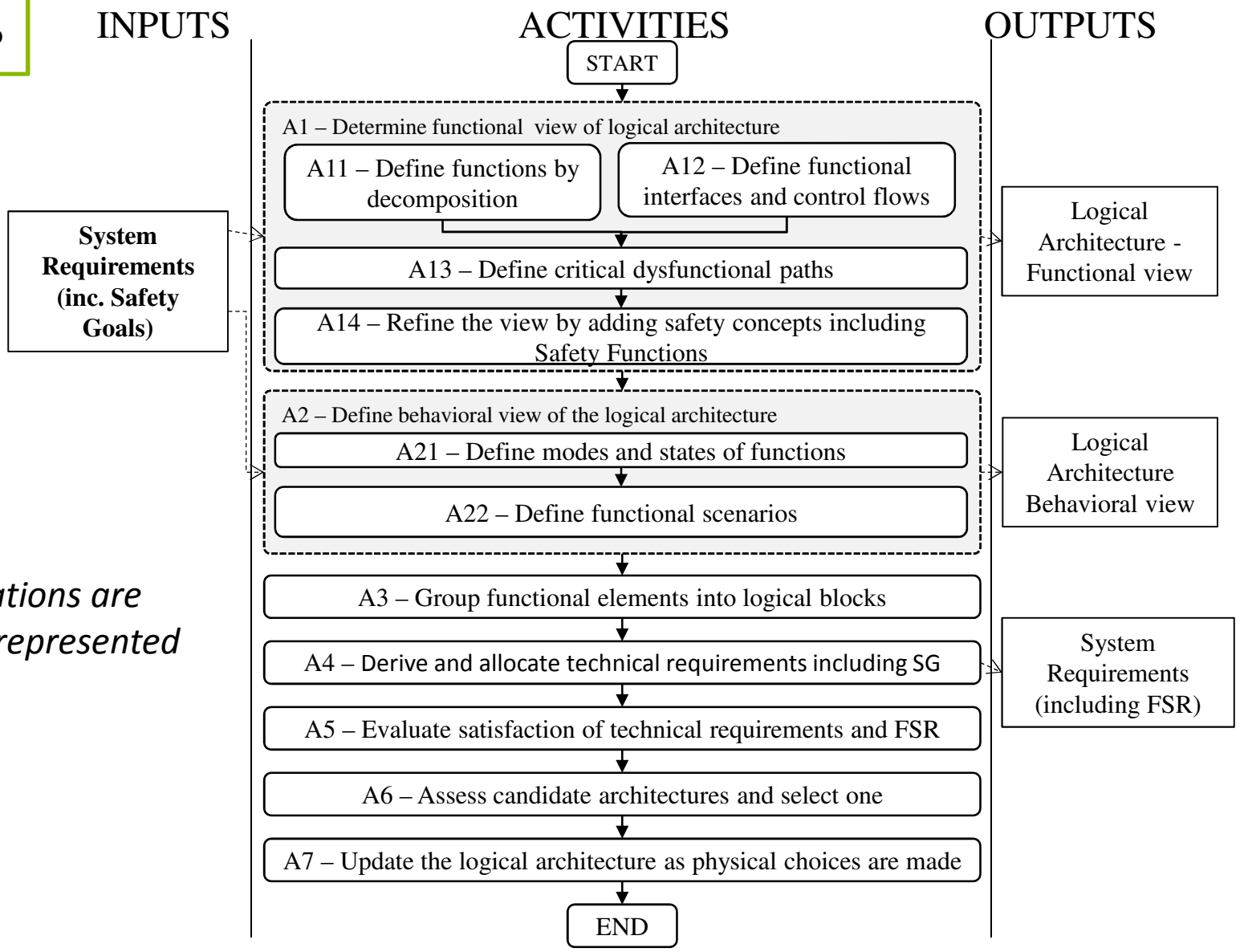
How to define it?

SEBoK: Systems Engineering Body of Knowledge (www.sebokwiki.org)

SEBoK Process	Proposed SLA Definition Process
Determine functional view of the logical architecture	**Define technical functions by decomposition & Identify the functional failure modes**
	Define functional interfaces and control flows
	Define critical dysfunctional paths
	Refine the view by adding safety concepts including Safety Functions
Define behavioral view of the logical architecture	**Define modes and states of functions (nominal, degraded, safe)**
	Define functional scenarios (nominal and dysfunctional)
Group functional elements into logical blocks	
Derive and allocate technical requirements of higher level including Safety Goals	
Assess candidate architectures and select one	**Evaluate satisfaction of technical requirements and Functional Safety Requirements**
	Assess candidate architectures and select one
Update the logical architecture as physical architecture choices are made	

Proposed SLA Definition Process

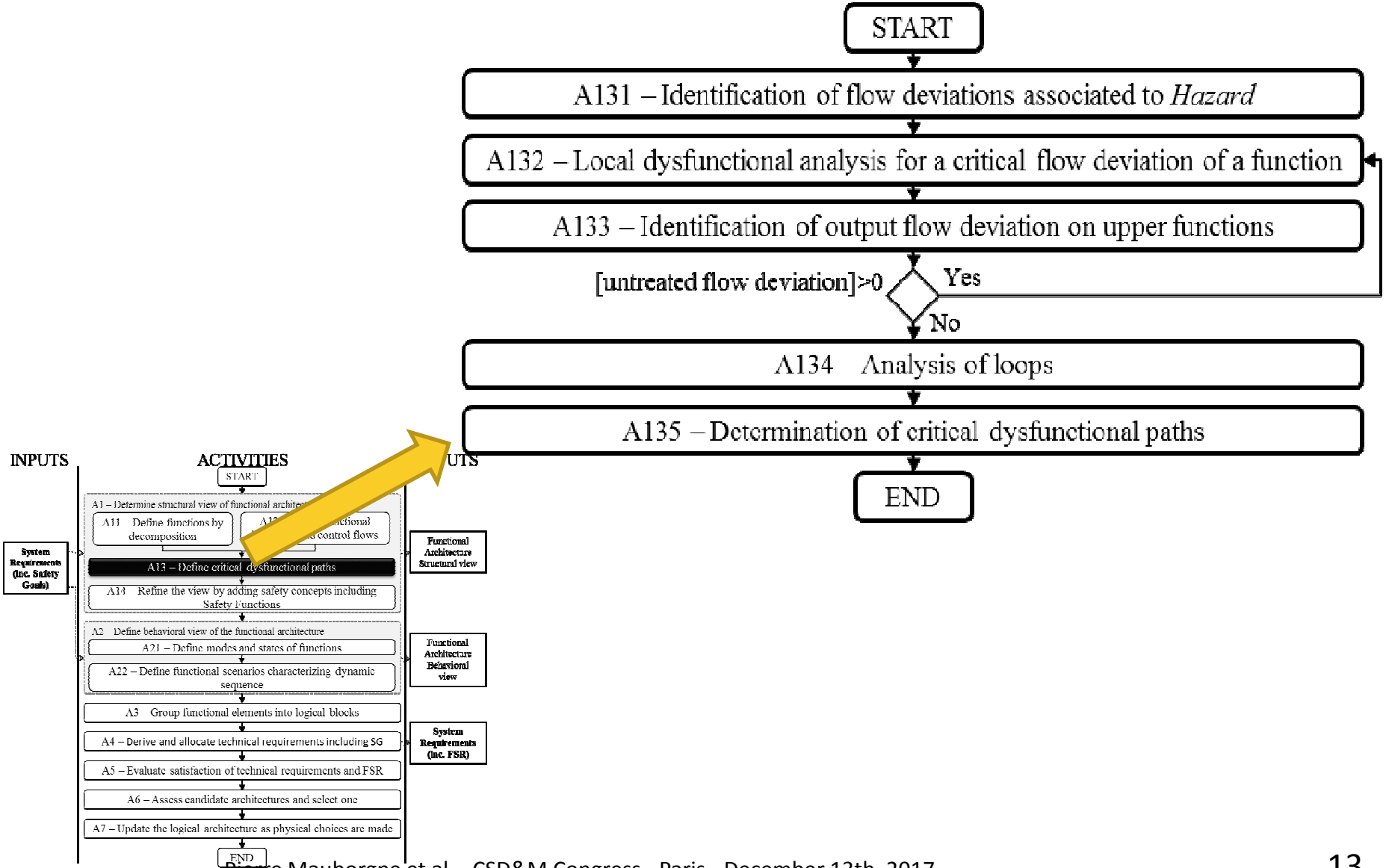
How to define it?



Iterations are not represented

SLA definition method – Functional view

- A13 – Define critical dysfunctional paths



Application to a case study

- In previous work, we have proposed a method to define Safety Goals (SG) related to the operational (external) view of the system (Mauborgne, et al., 2016).

- We focus now on possible critical dysfunctional paths of the hazard: unintended acceleration of the vehicle.

- As an input of the current process, there is a Safety Goal concerning an output flow of the function of the powertrain system which is called *'Generate Mechanical Energy'* (GME) which is one of the principal function of the vehicle.

PHA of "Unintended acceleration during driving".

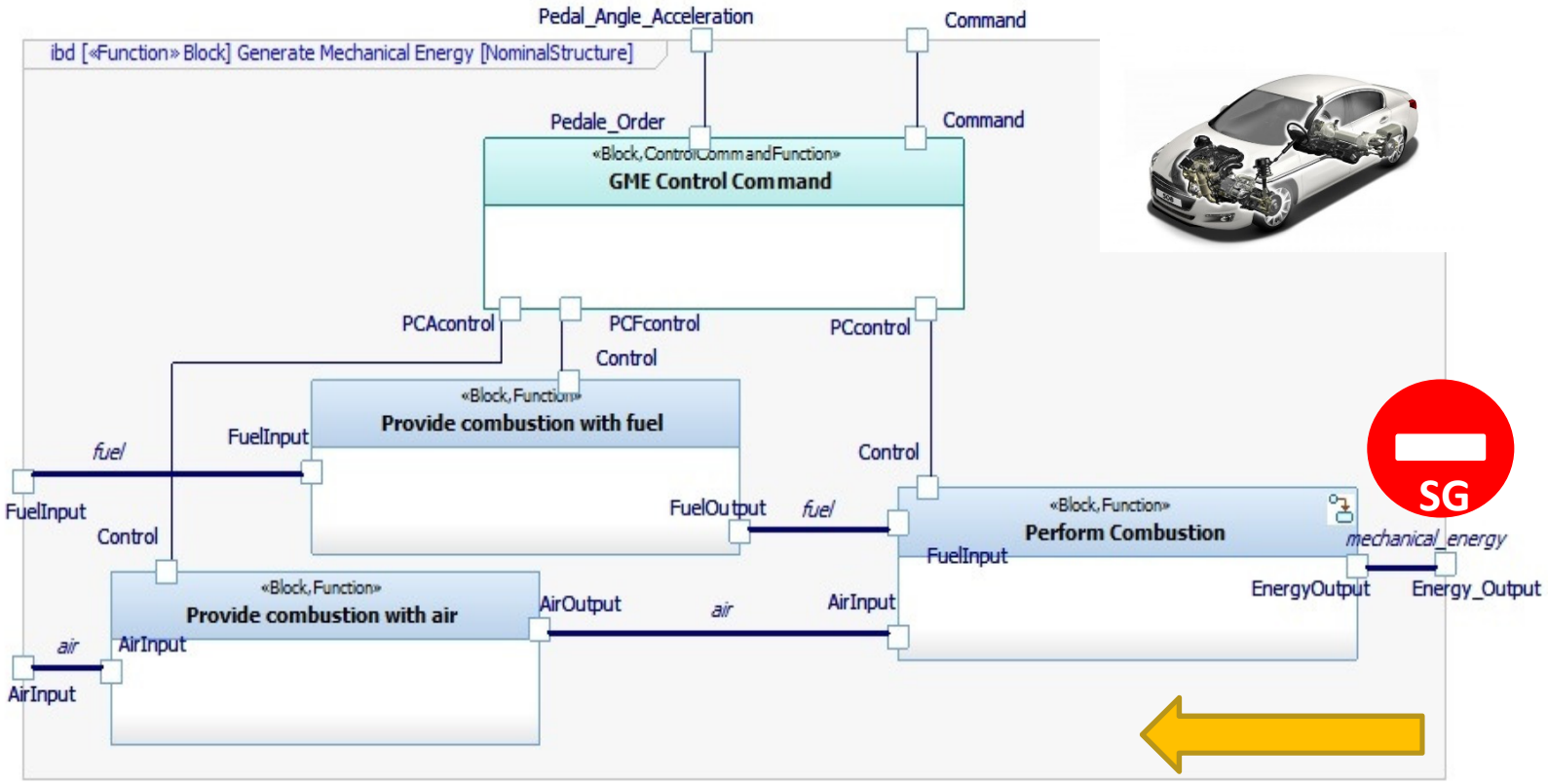
System	Phase	Output flow	Hazard	Hazardous event	Harmful event	Situation with harm
Vehicle	Driving	Torque to road	Unintended acceleration	Unintended acceleration during driving	Shock	Critical injuries of the driver

Prob. of exposure	Sev.	Control.	ASIL	Avoidance scenario	Safety Goal
E4	S3	C2	C	Reduction of performances	The difference between driver's intend and the acceleration of the vehicle must be less than Y %

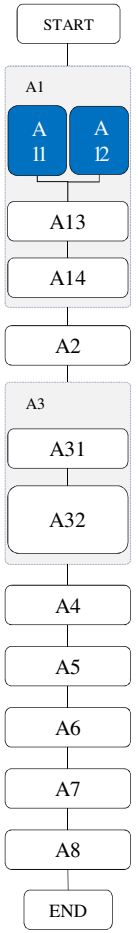
Mauborgne, P., et al. (2016). Operational and system hazard analysis in a safe systems requirement engineering process – Application to automotive industry. *Safety Science, 87*, 256-268.

Application to a case study

- Preliminary logical architecture: GME is broken down into four sub-functions that are '*Provide combustion with fuel*', '*Provide combustion with air*', '*Perform combustion*', and a control sub-function (Activities A11 & A12) (according to an architectural pattern - not described in this talk).



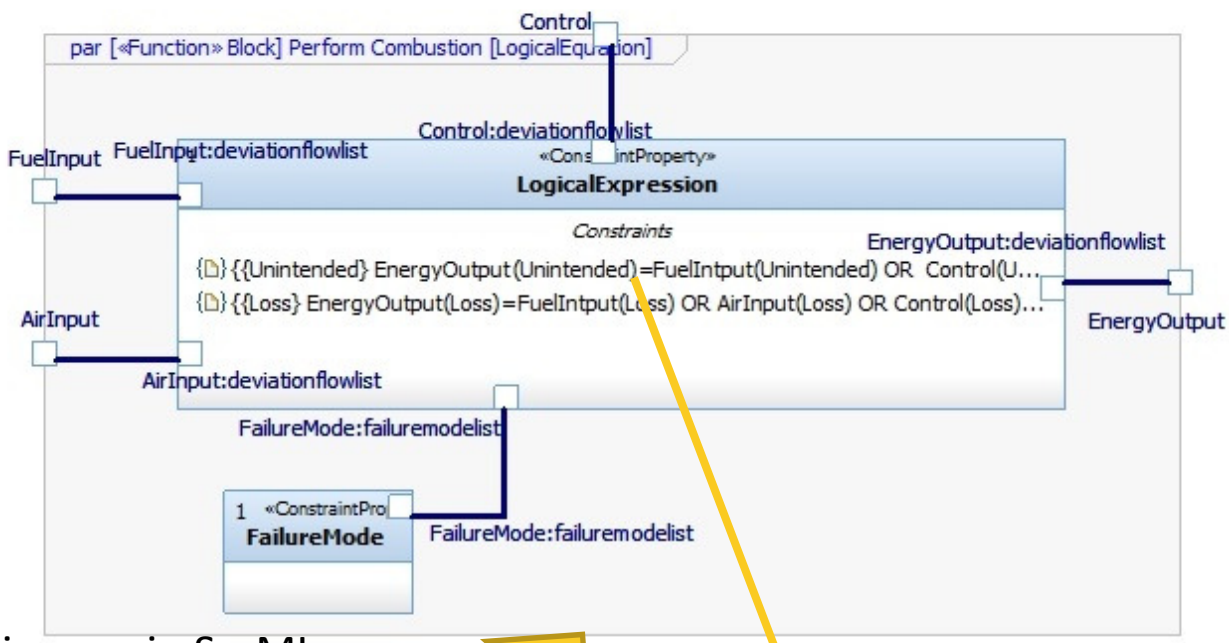
IBD in SysML



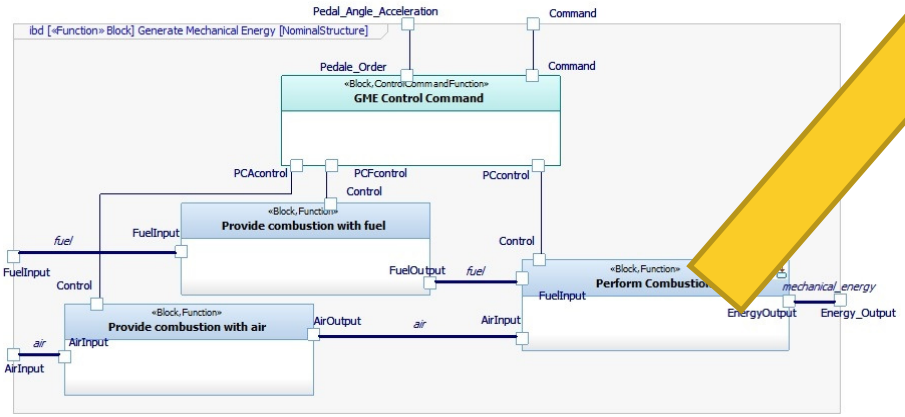
The first sub-function that has to be considered when studying the dysfunctional aspect is *'Perform combustion'*.

Application to a case study

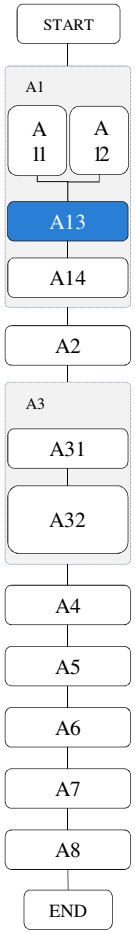
- Local dysfunctional analysis: definition of logical equations



Parametric diagram in SysML

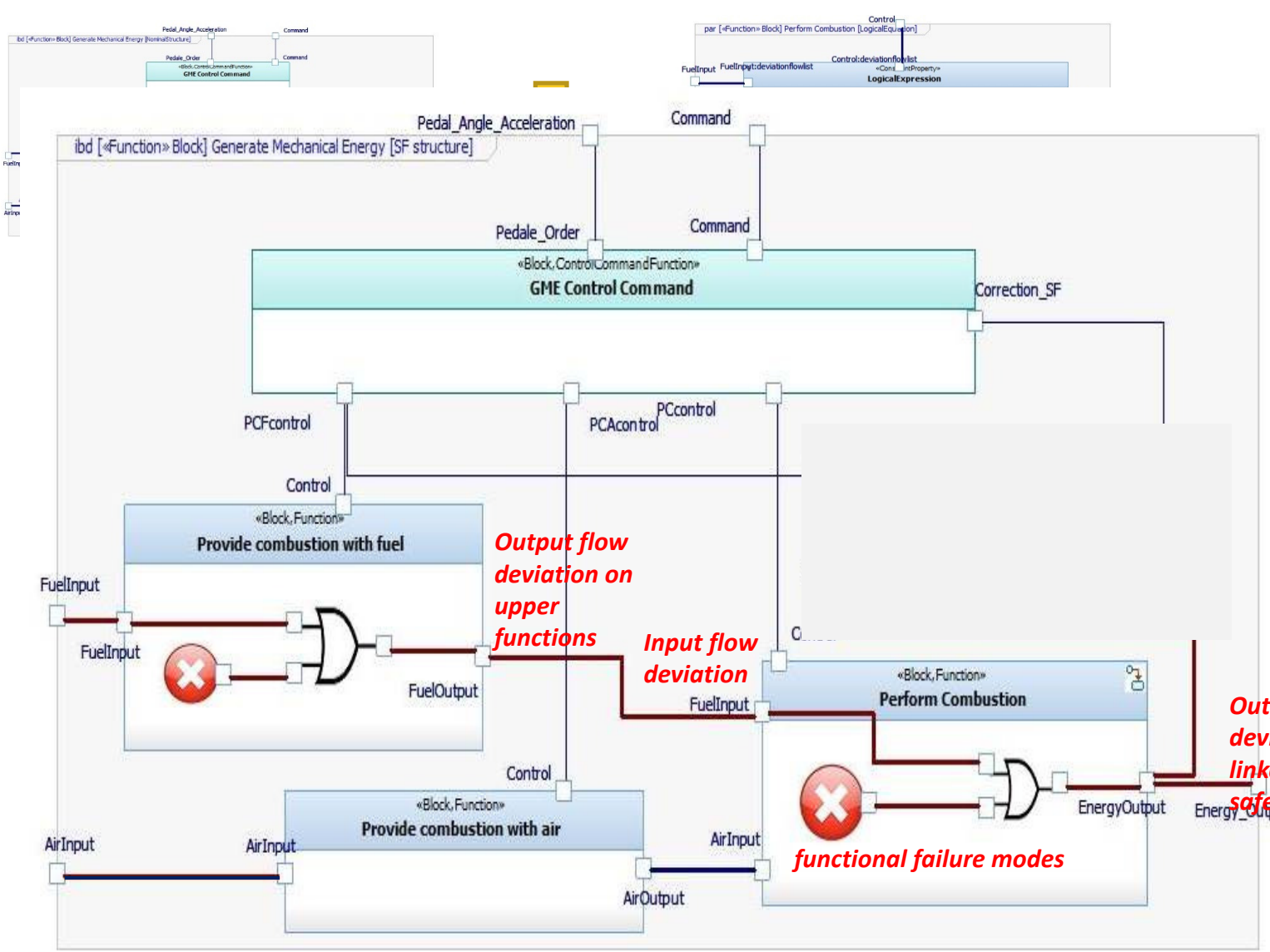


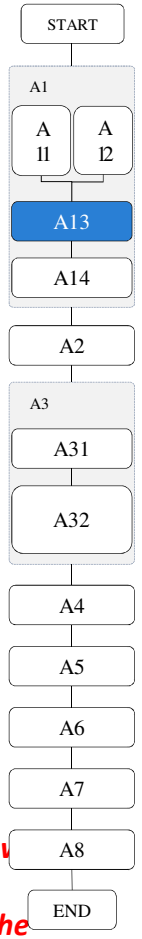
EnergyOutput(Unintended)=
FuelInput(Unintended)
OR Control(Unintended)
OR FailureMode (Unintended).



Application to a case study

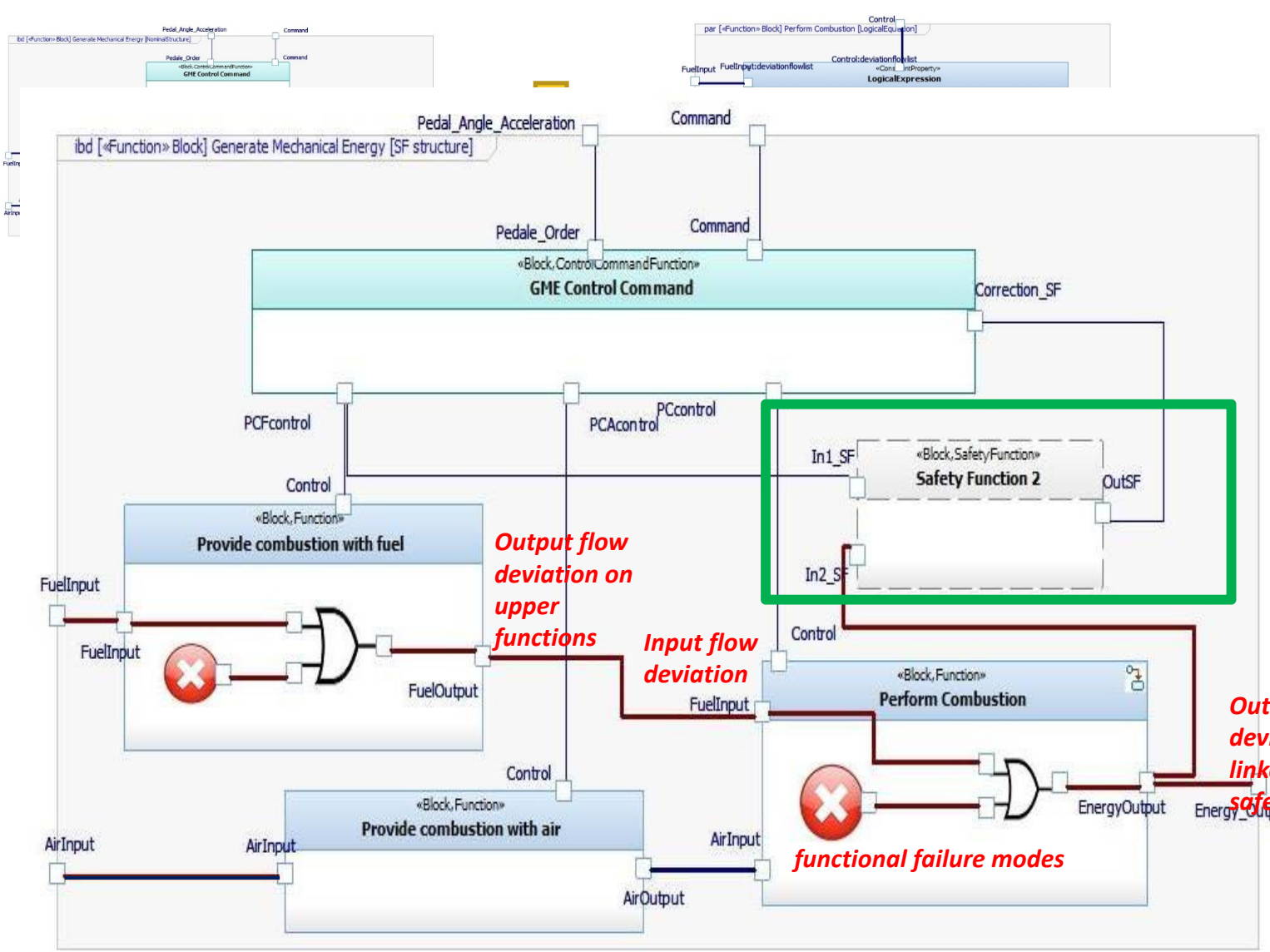
- Definition of enriched logical architecture – functional/dysfunctional view

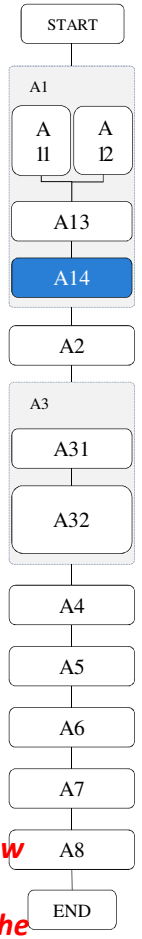




Application to a case study

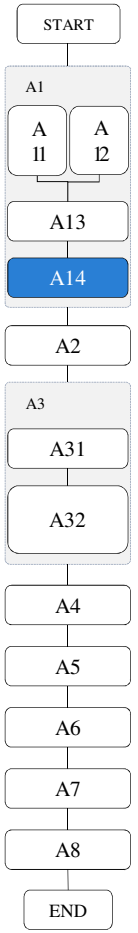
- Definition of enriched logical architecture – functional/dysfunctional view





A14 – Refine the view by adding safety concepts including Safety Functions: Consistency checking from a safety viewpoint

- We have created a prototype with the language Prolog. It consists in formalizing by means of logical equations the value of an output flow that can lead to a hazard, based on the values of the inputs and the functional failure modes of the functional view of the logical architecture.
- The different programs that have been developed allow to perform these analyses:
  - Verification of the adequacy of the behaviors of the sub-functions with the behavior of the composed function (verify the compositional laws)
  - Search for critical dysfunctional paths. In logic programming, the fault modes of the subfunctions and / or their inputs on the critical path are identified. The result is the set of dysfunctional critical paths for a chosen hazard.
  - Verification of candidate architectures after the introduction of different safety functions. The search for critical dysfunctional paths is performed again to determine which ones have been deleted. This makes it possible to check the effectiveness of a Safety Function. The system architect can then compare (1) the properties before, then after the introduction of the Safety Function and (2) several candidate architectures.
- Details: Pierre Mauborgne's PhD thesis (Mauborgne, 2016).



Conclusions & Further Works

- Conclusion
  - a method
    - systematic, not dependent on the user
    - consistent with the ISO 26262 standard
    - A step towards a safe systems engineering process
- Advantages
  - Enriched logical architecture model
    - No redundant analysis - No loss of information
    - Introduction of a Safety Function ➔ reduced loop time
    - Safety specialists can focus on quantitative analyses.
- Further Works
  - Consider different patterns of safety functions
  - Extend this method to the physical architecture

Questions?

Definition and verification of functional safety concepts for the definition of safe logical architectures

Pierre Mauborgne, Éric Bonjour, Christophe Perrard, Samuel Deniaud,
Éric Levrat, Jean-Pierre Micaëlli, Dominique Loise

Automotive Safety Integrity Level

Automotive Safety Integrity Level (ASIL) is a measure of criticity of each hazard including the probability of exposure, controllability and severity,

with A representing the least stringent level and D the most stringent one.

The class QM (Quality Management) denotes no requirement in accordance with ISO 26262.

Severity	Exposition	Controlability		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	**C**
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D