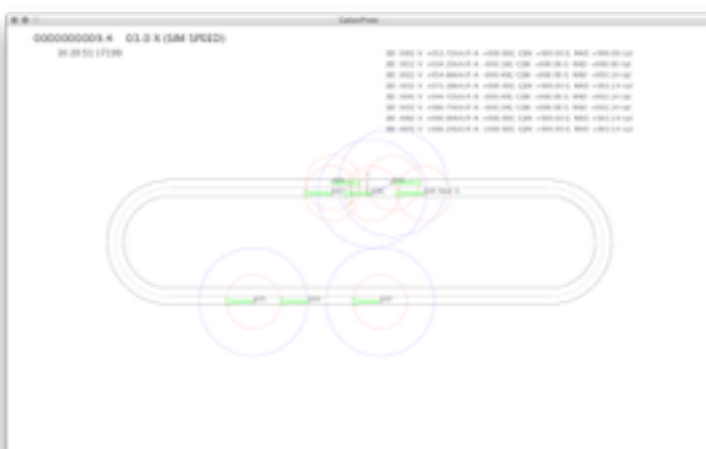
How can we deal with the concept phase in the functional safety standard for automobiles



Nil Software Corp.
Masao Ito

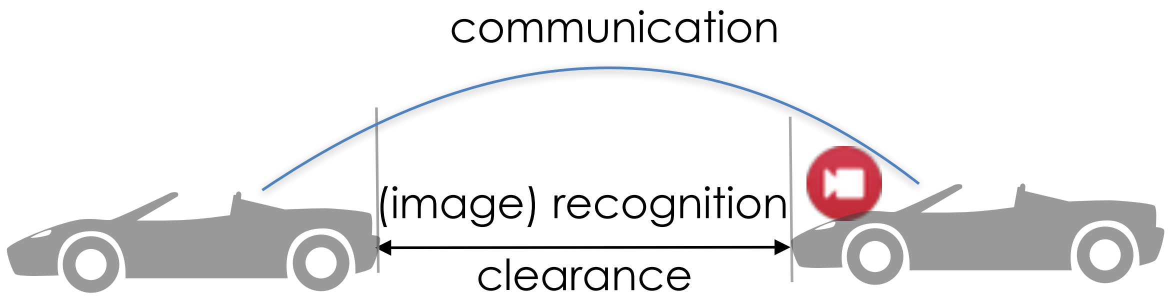
Useless concept phase ?!

擦り合わせ (Su-ri-awa-se)

- People in the automobile field always say that there is no chance to develop an item from scratch. Because currently the most important activity is *Su-ri-awa-se* (closely coordination). And they sometimes set aside the importance of the concept phase.
- But, I think we will have to think the new systems in the future automated driving car. In that time, I believe we need the coherent approach for establishing safety in the new car.

Example

- I use CACC as an example to explain our approach
  - CACC is an enhancement of ACC that enables more accurate gap control and operations at smaller gaps by adding communication using the forward vehicle information. In this type, we use the LIDAR for recognition of the target car



Simple image of CACC (it has two mechanism to get the forward car information)

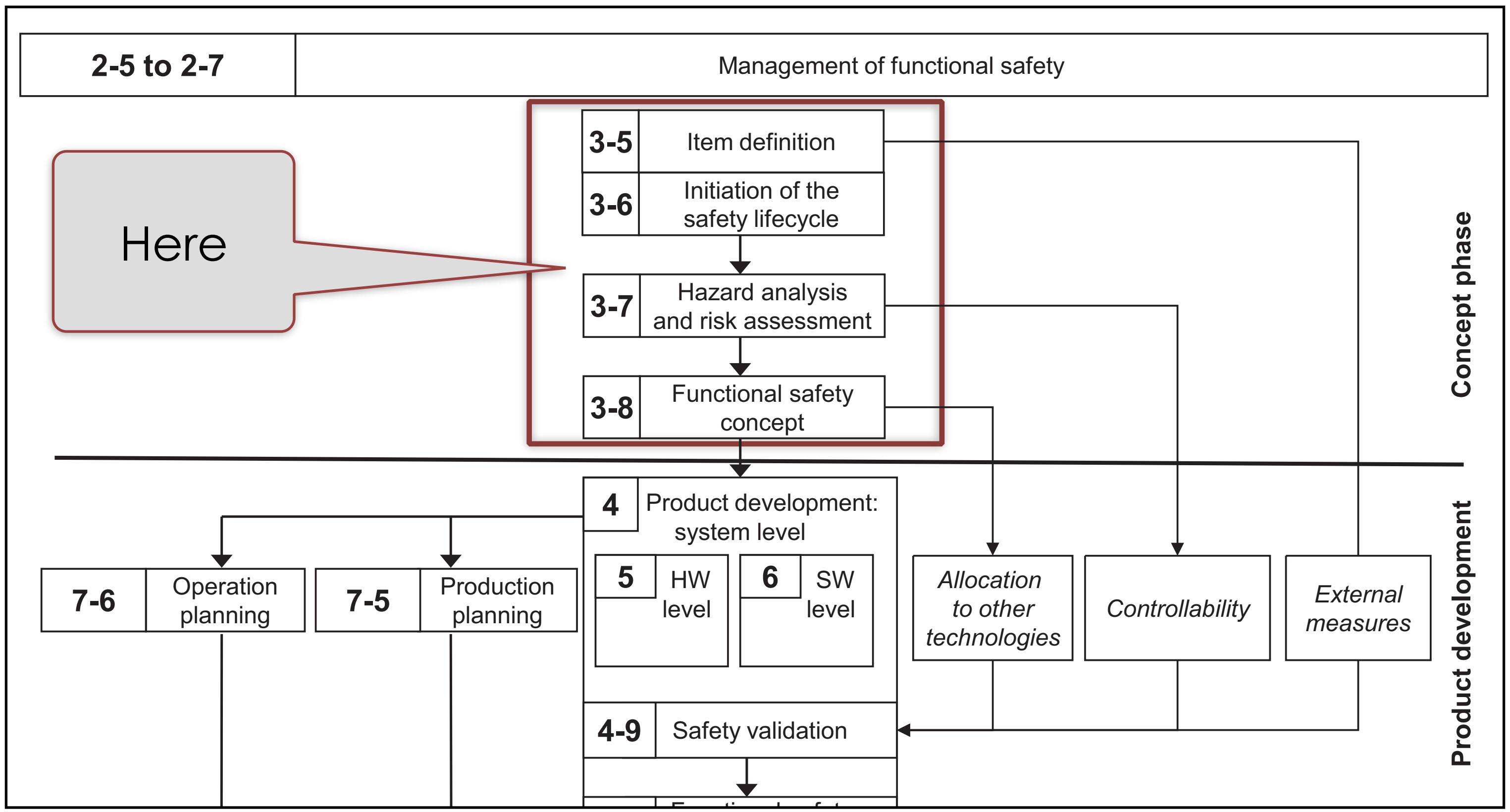
CACC: Cooperative Adaptive Cruise Control

Concept phase ?

- Part 3 of ISO 26262 is for the concept phase.
- This phase has four sub-phases:
  - Item definition
  - Initiation of software lifecycle
  - Hazard analysis and risk assessment (HARA)
  - Functional safety concept

Where is the Concept Phase ?

- It is the first phase in the development process
  - from item definition (3-5) to functional safety concept (3-8)



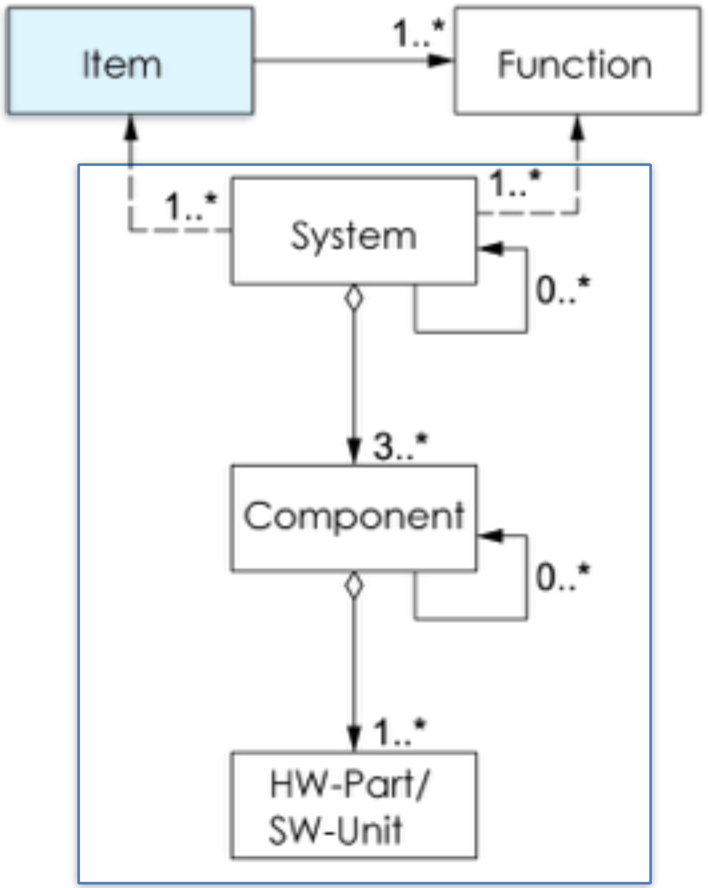
(ISO 26262 Part 2 Figure 2)

five issues

- Item ?
- Safety activity and other development activity
- Finding Hazards
- How to calculate the controllability for ASIL
- Several "times"

item ?

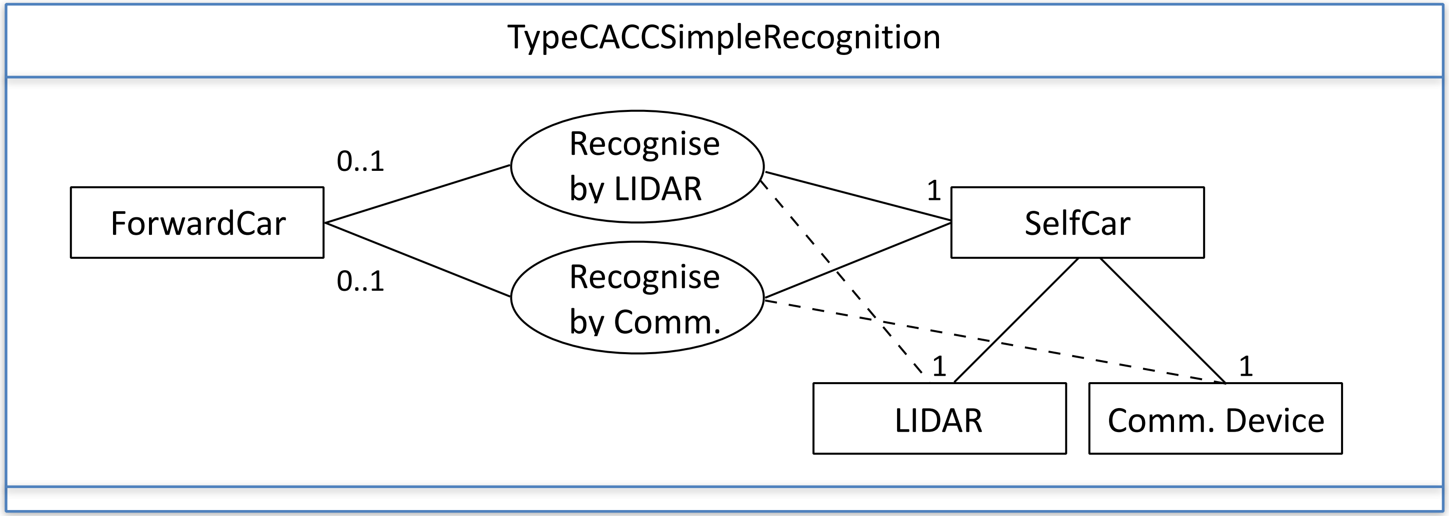
- The item is not a system. It is an abstract object, and a system is generated from the item.
  - e.g.
  - The auto-cruise control system is an item
  - The ACC in the toyota camry is a system
- As for system, we have many analyzing method. But I think there is no good approach of the item.



ISO 26262 Part 10 Fig. 3

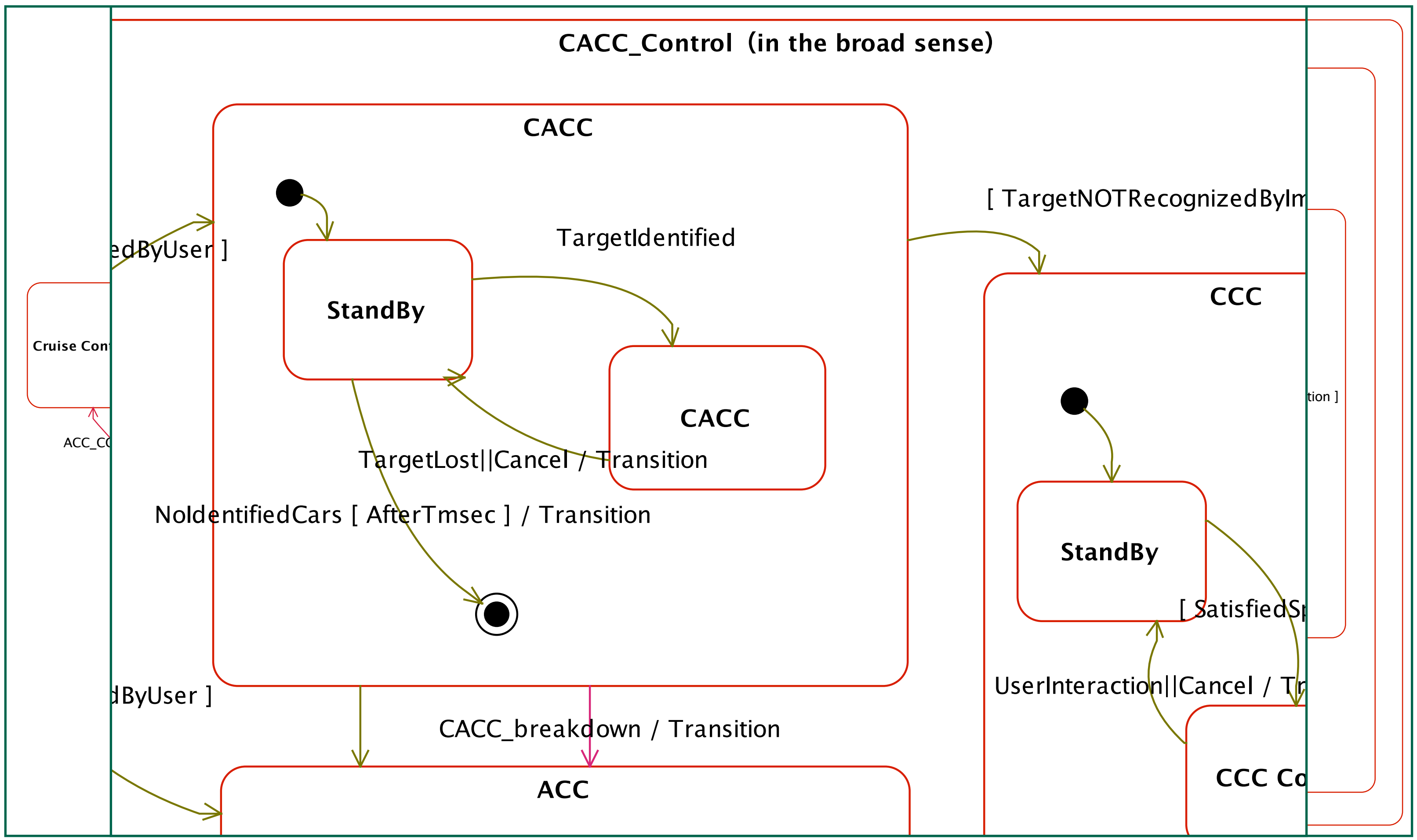
Item Sketch

- We use the item sketch to represent the static and dynamic model of an item
  - As the static representation, we use the type model of catalysis (, but uml class model is enough in this phase)
  - As the dynamic model, we can use the statechart as a finite state machine



Example of static item sketch

Item Sketch


CACC_Control (in the broad sense)
CACC
StandBy
TargetIdentified
CACC
TargetLost||Cancel / Transition
NoIdentifiedCars [AfterTmsec] / Transition
[TargetNOTRecognizedByIm
CCC
StandBy
[SatisfiedSp
UserInteraction||Cancel / Tr
CCC Co
edByUser]
Cruise Cont
ACC_CC
dByUser]
CACC_breakdown / Transition
ACC
tion]

five issues

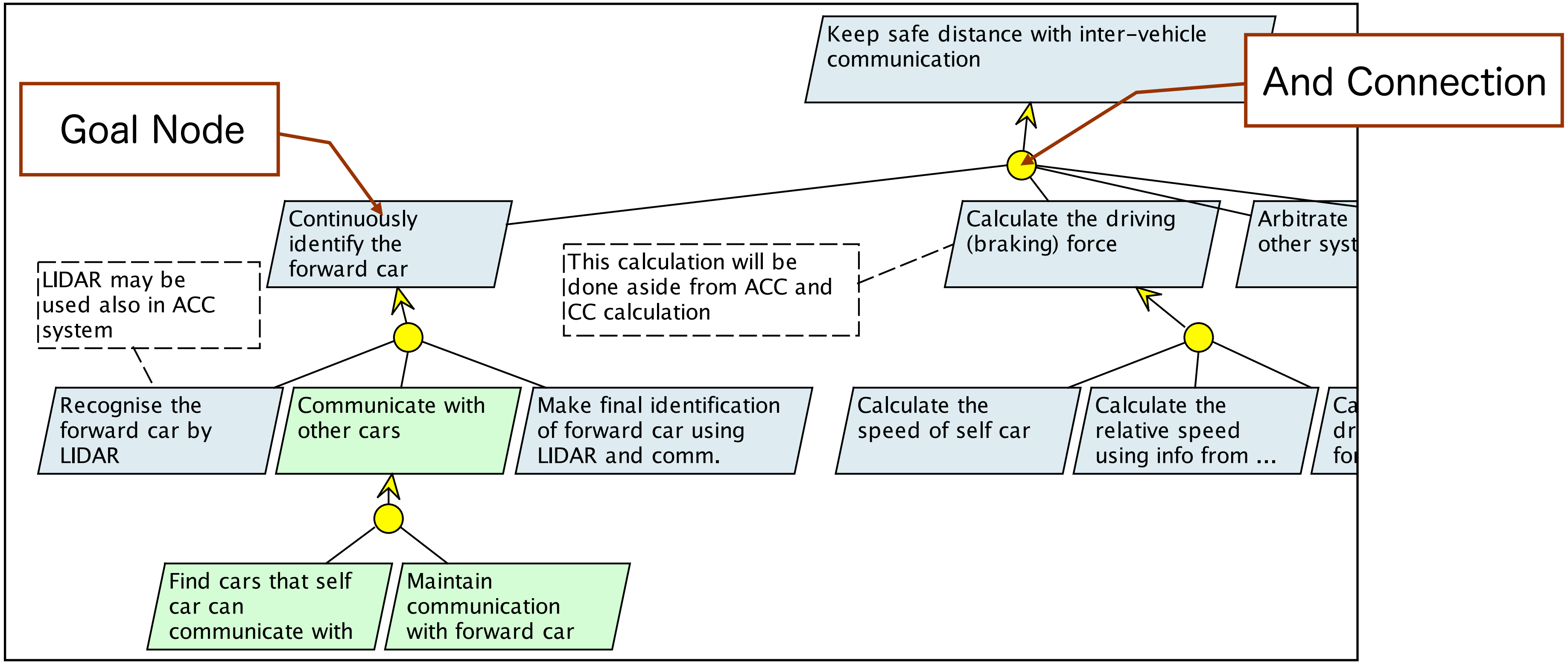
- Item ?
  - Item sketch (static & dynamic model)
- Safety activity and other development activity
- Finding Hazards
- How to calculate the controllability for ASIL
- Several “times”

Safety activity and other development activity

- No separation
  - ISO 26262 is the standard for functional safety. We would like to locate it in the whole development process, because in the early phase (i.e. concept phase) it is hard to divide it into the development and safety activity
- Solution: Goal Model
  - To consolidate the requirements in the abstract level, we use the KAOS approach
  - (Obstacle node is a candidate of hazard)

Goal model

- The goal of an item is the top goal. We decompose it into the sub-goals. We can also write the non-functional requirement, for example, as a soft goal



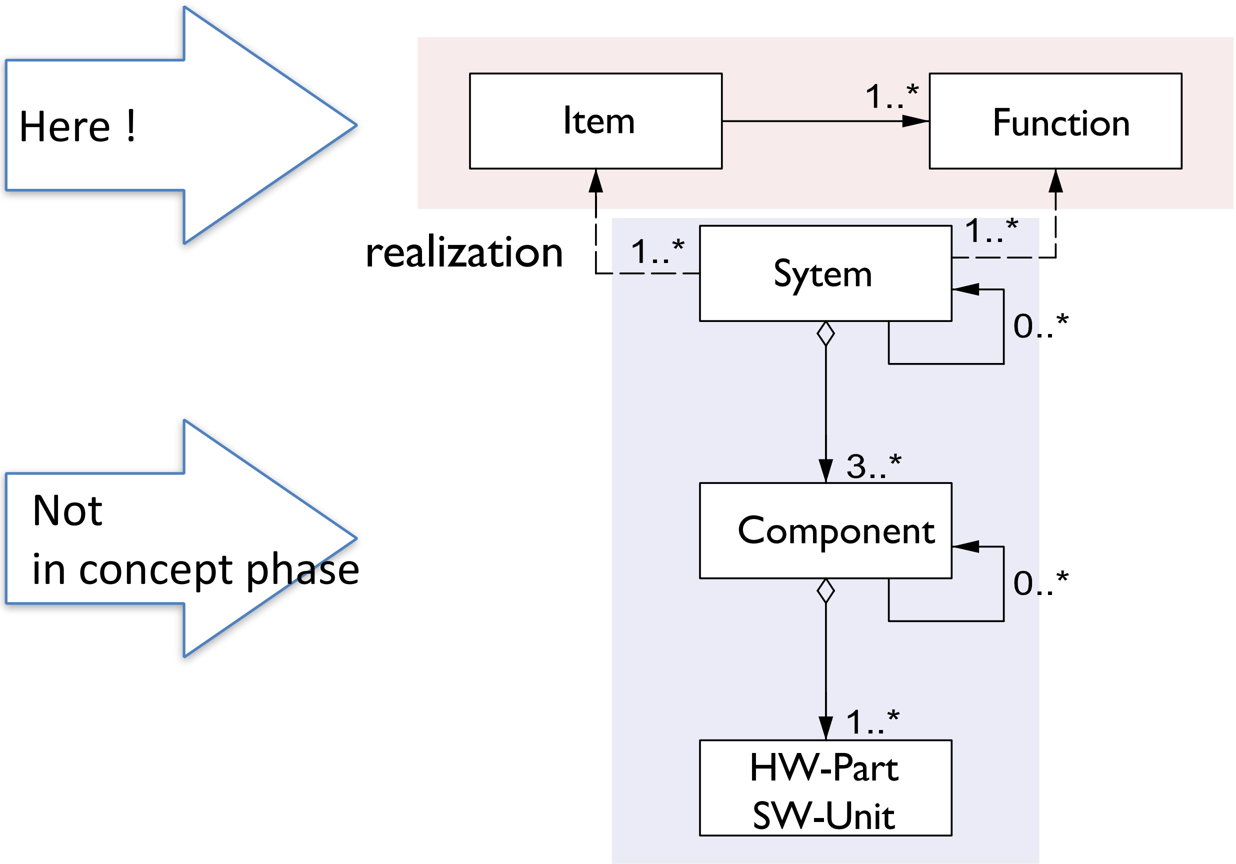
example of goal modeling by goal decomposition

five issues

- Item ?
  - Item sketch (static & dynamic model)
- Safety activity in the whole development process
  - Use goal model
- Finding Hazards
- How to calculate the controllability for ASIL
- Several “times”

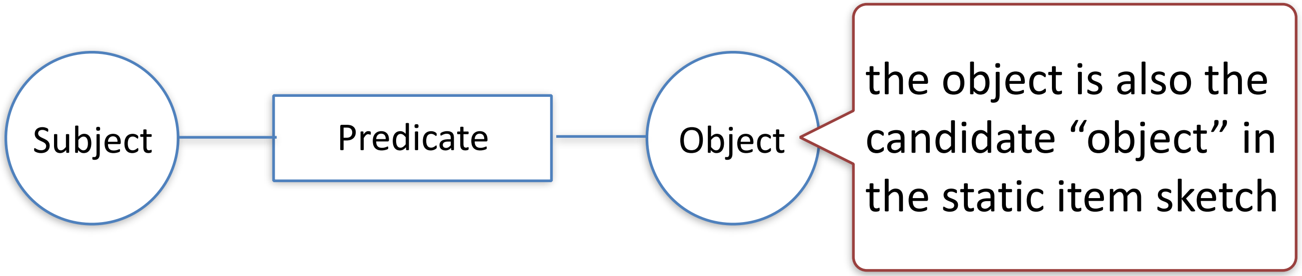
Finding Hazards

- The item is an abstract object and it is not a system
- So, It is hard to use the conventional method (such as FTA).



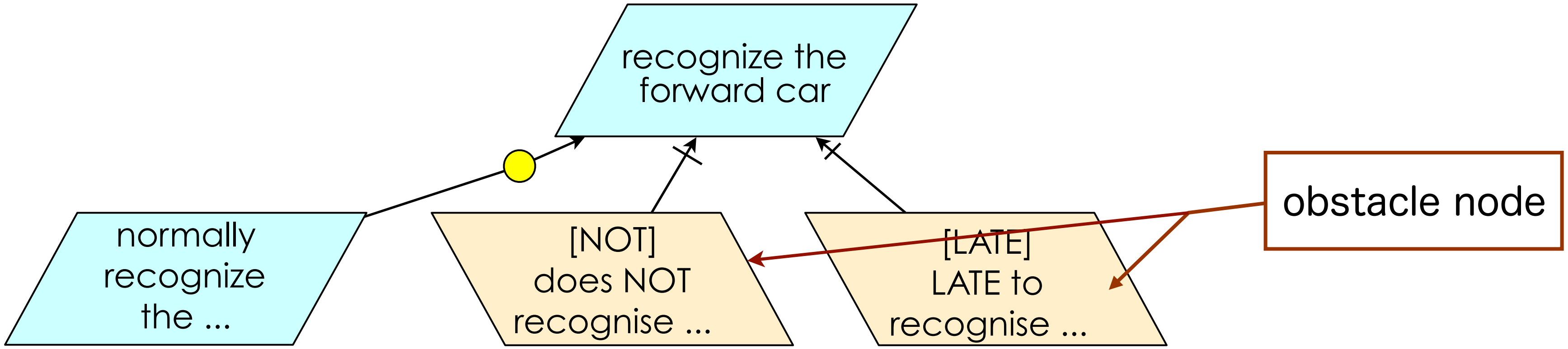
Finding Hazards

- We use the description of a goal, it is compromising semi-formal approach
  - Because,
    - In concept phase, it is hard to describe the formal model
    - But, the graphical representation of item sketch (UML and specification type) help us to think correctly.
  - If sentence consist of <Subject> <Verb> <Object>, we can write:
    - e.g. The subject car can recognize the car ahead by LIDAR.
  - Insert the guide word (of HAZOP) or change the predicate/object.
    - e.g. The subject car can NOT recognize the car ahead by LIDAR



Finding Hazards

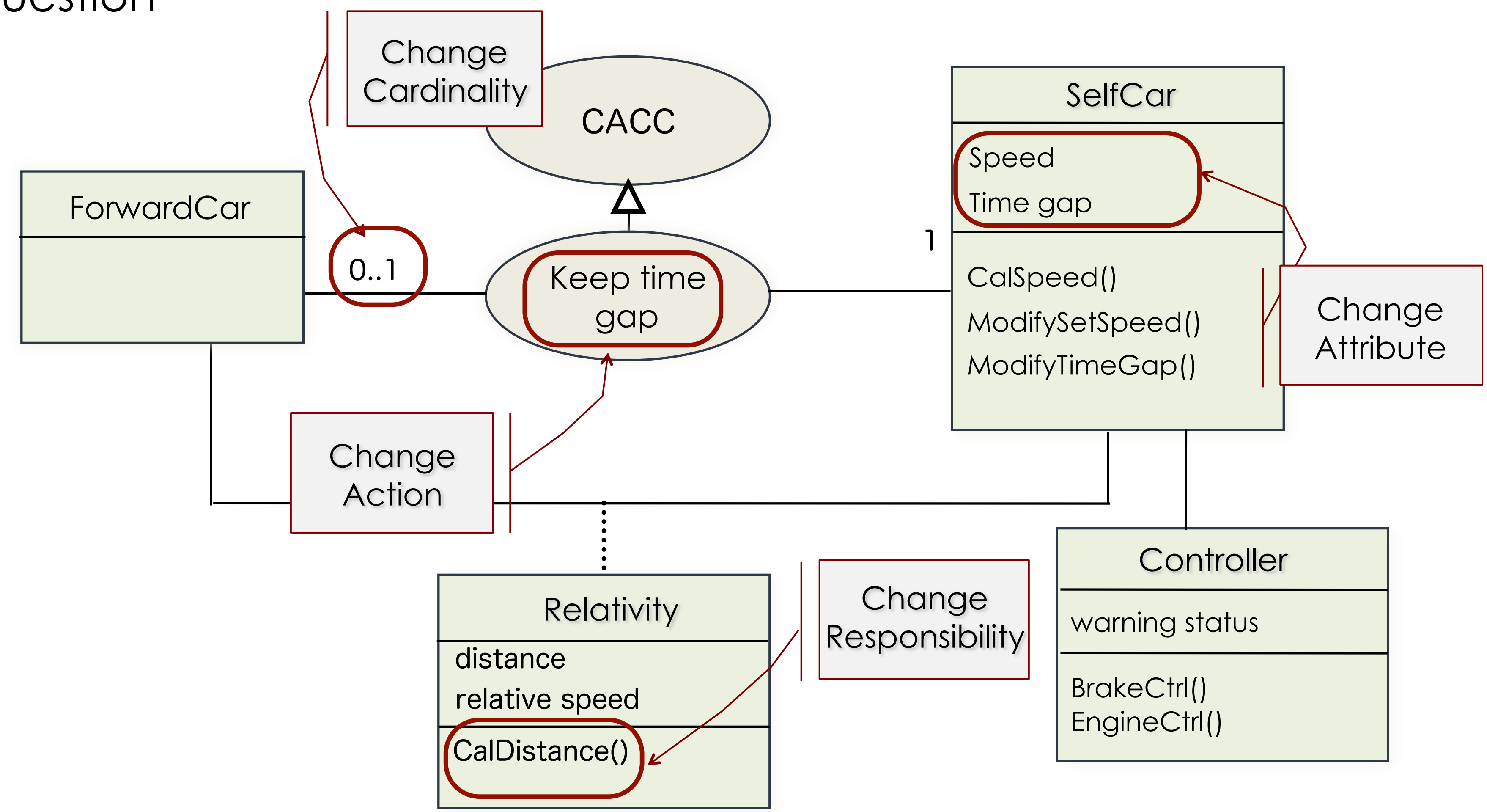
- Use sentence in the goal node
- Apply the what-if question to the goal node
  - e.g.: "recognize the forward car"
    - (system) does **NOT** recognize the forward car
    - (system) is **LATE** to recognize the forward car



Goal VS. Obstacle

Finding Hazards

- Another method: item sketch is helpful to apply the what-if type question



SSM

- Situation-Scenario Matrix
  - We can express the usage of an item by the scenario and the situation.
- Example: CACC
  - Road type
  - Structure on the road
  - Neighboring car
  - Degree of jam
  - Climate visibility
  - Non-automobile perimeter objects
  - Regulation

SSM

- Example

Situations

Attrib. / Time (HM:S)	Road				Structure		Neighboring Car		F
	Type*	State*	Lane#	Curve (m)	Light-ing	Guard Rail	Front Dist. (m)	Rear Dist. (m)	F
1010:00	RT_SB	GR(0), GG(0), MU(0.8)	2	-	Y	Y	30	20	
1012:00	↑	↑	↑	-	↑	↑	30	20	
…									
1030:00	RR_CL	GR(0), GG(0), MU(0.6)	1	-	N	N	150	200	

A Scenario

situation category

situation attribute

*: appendix

An Example SSM of CACC

five issues

- Item ?
  - Item sketch (static & dynamic model)
- Safety activity in the whole development process
  - Use goal model
- Finding Hazards
  - guide words, Situation-Scenario Matrix (SSM)
- How to calculate the controllability for ASIL
- Several "times"

ASIL and Controllability

- We need three factors to calculate ASIL

CACC		**B**
Scenario	In highway, (AND) driving at high velocity in CACC mode	
Malfunction	Identified, but there are differences in both information. If this situation continues, controller may indicate the wrong time gap.	
Severity	It may lead to crash with the forward car in larger velocity than expected	S3
Exposure	E3: Highway E4: High velocity	E3
Controllability	If driver notices the wrong behavior of CACC, he can put on the brake and he can escape from the CACC control.	C2

It comes from SSM

from Obstacle

ASIL definition of an item

Controllability

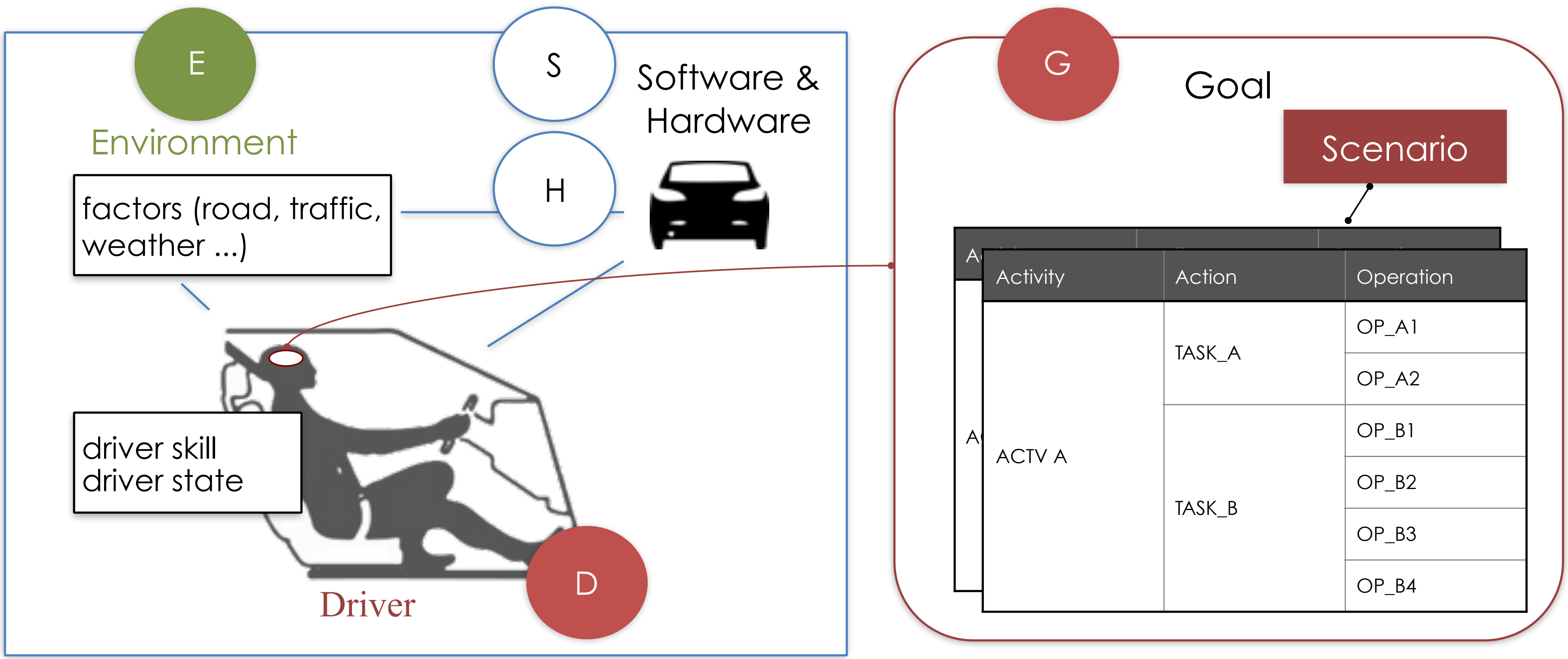
- Controllability is the "ability to avoid a specified harm or damage through the timely reactions of the persons involved, possibly with support from external measures" (ISO26262 1-19)

⇩

How to calculate ?

Big Picture with driver and environment model

- DESH-G schema covers the environment, driver and goal as well as hardware and software.



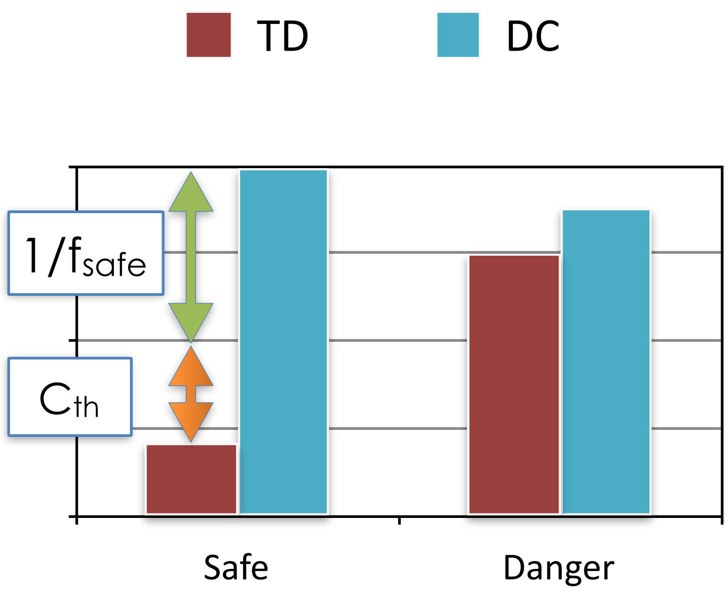
Driving Difficulty : DD

Driving Difficulty (DD) is given by the difference between the value of Driver Capability (DC) and the value of the Task Demand (TD) to achieve the driver goal.

$$f_{safe}(dc,td,c_{th}) = \begin{cases} f_{mrg}(dc,td) - c_{th} & when \quad f_{safe} \geq c_{th} \\ 0 & when \quad f_{safe} < c_{th} \end{cases}$$

$$f_{mrg}(dc,td) = dc - td$$

$$INV : dc > td$$

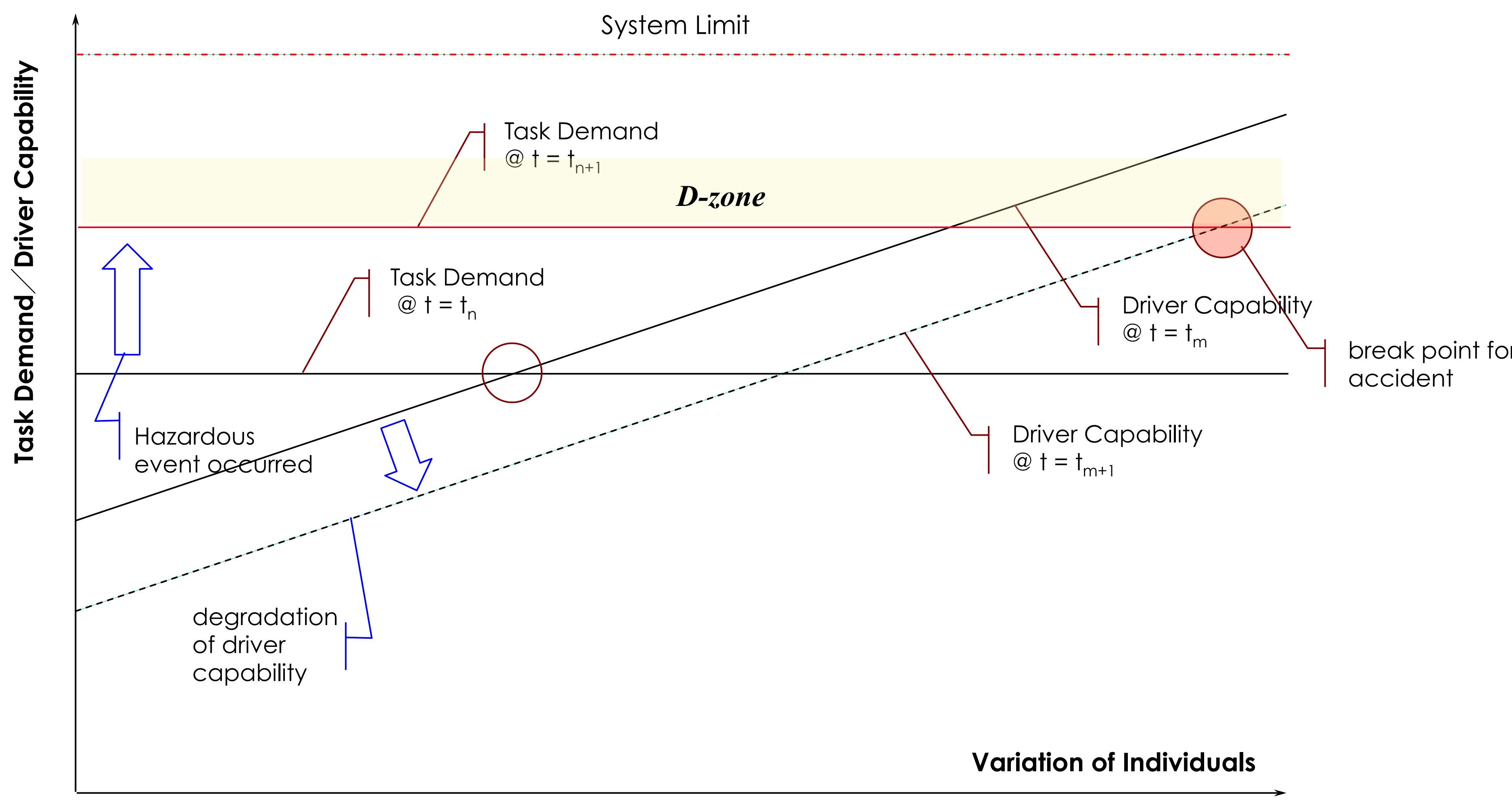


dc : DC (Driver Capability)

td : TD (Task Demand)

c_{th} : threshold

Safety vs Harm

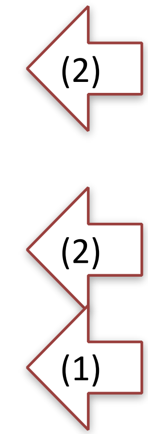


five issues

- Item ?
  - Item sketch (static & dynamic model)
- Safety activity in the whole development process
  - Use goal model
- Finding Hazards
  - Guide words, Situation-Scenario Matrix (SSM)
- How to calculate controllability for ASIL
  - Driver model, SSM
- Several "times"

Several "times"

- Functional Safety Requirement (FSR) has followings:
  a) operating modes
  b) fault tolerant time interval (FTTI)
  c) safe states
  d) emergency operation interval, and
  e) functional redundancies (e.g. fault tolerance)

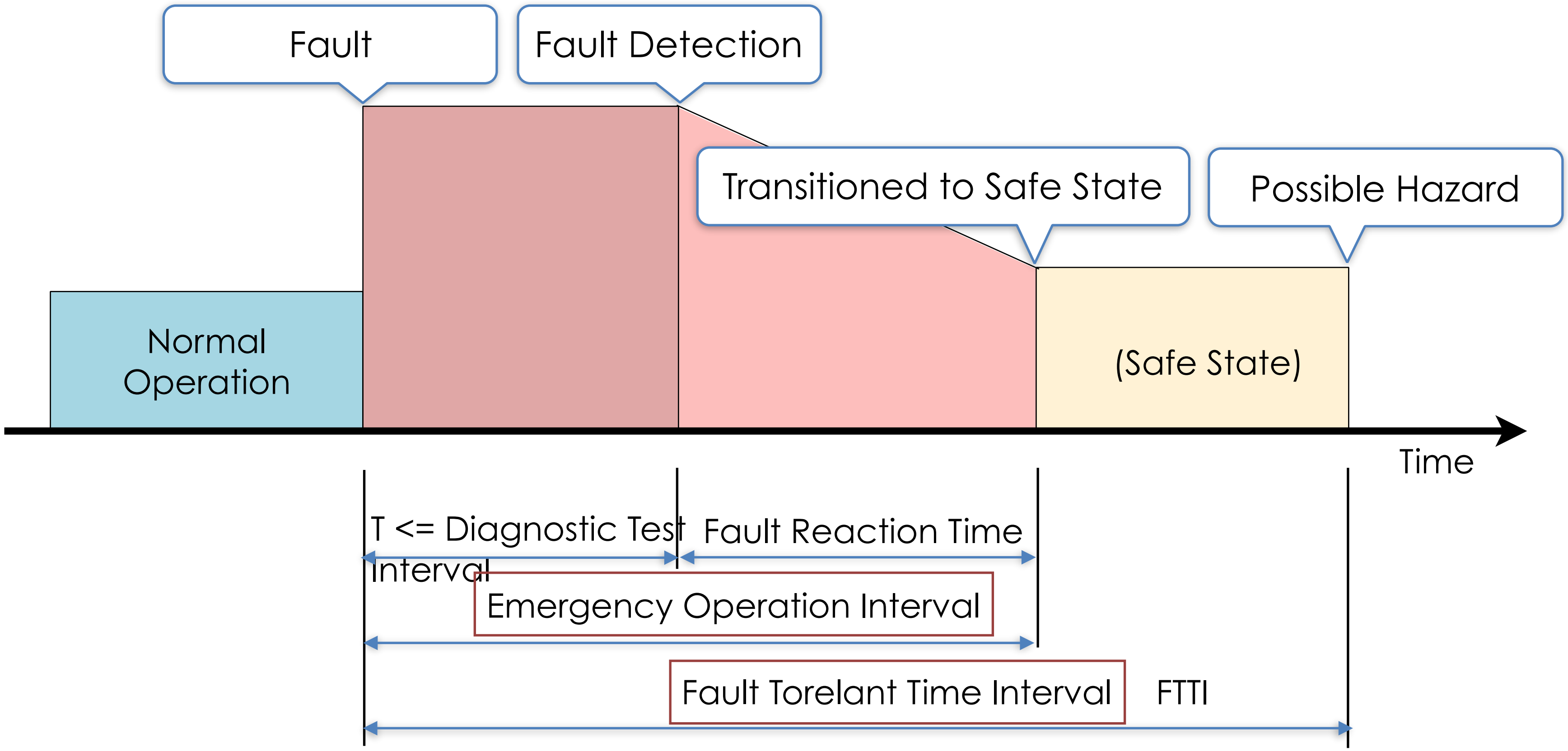


Points:

(1) Abstract Functional Safety Mechanism
(2) Flow Analysis and error description by AADL

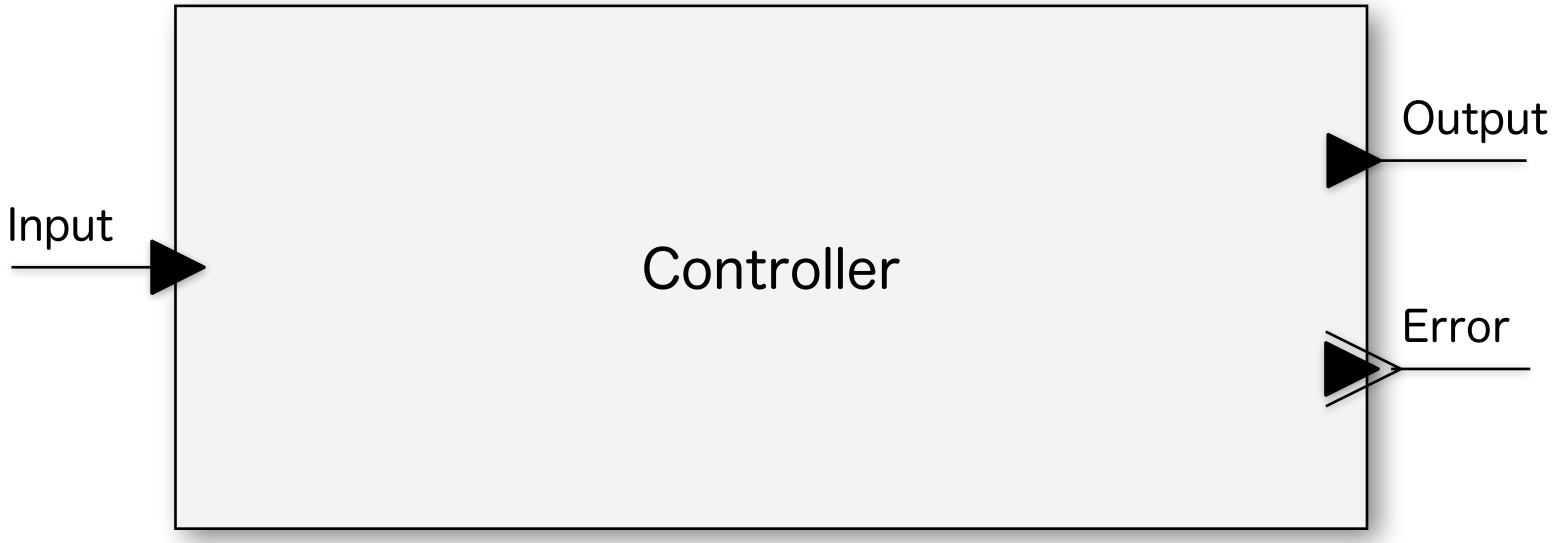
FTTI & Emergency Operation Interval

- Fault and Transition



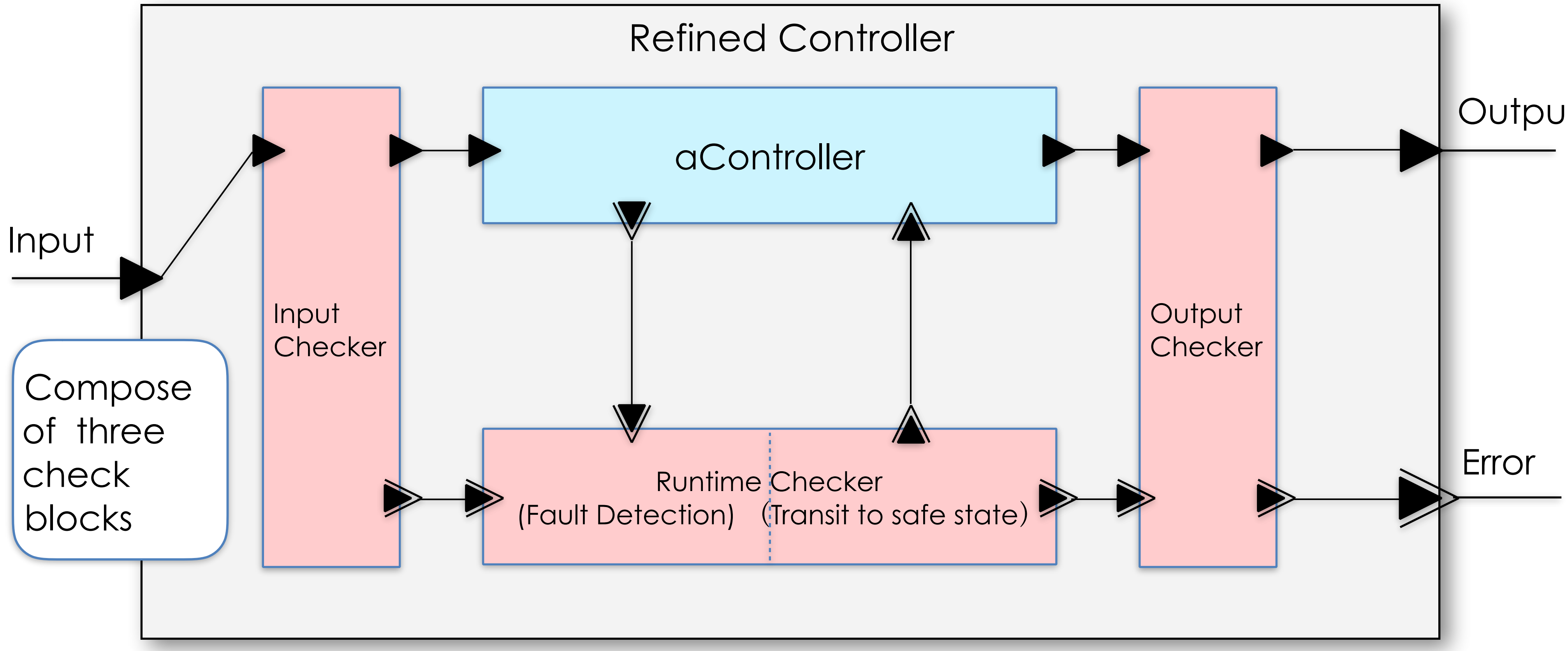
Fault reaction time and fault tolerant time interval (ISO26262-1 Fig.4)

Abstract Functional Safety Mechanism



Generic initial architecture w/ safety mechnism

- For functional redundancy, we have to several checker/verifier for the target controller



Initial Architecture

```
system implementation comp0.i
        subcomponents
                c : system pcontroller
                        {ISO26262::ASIL => LEVEL_B;};
                i : system pfsminp.i;
                s : system pfsmcre.i;
                o : system pfsmout.i;
        connections
                c0 : port i.p_out -> s.p_in;
                c1 : port s.p_out -> o.p_in;
                ce : port o.p_err -> p_err;
        annex EMV2 {**
                use types errorlibrary;
                use behavior
NILErrorModelLibrary::Basic_behave;
                …
                -- state transition --
                composite error behavior
                states
                        [o.failed]->failed;
                end composite;
        **};
end comp0.i;
```

Implementation part

ISO 26262 property set

Three checkers

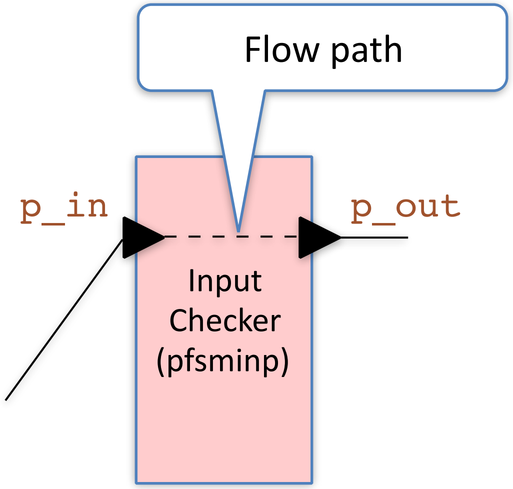
in/out

Use error annex

Error relating behavior

Describing estimated latency

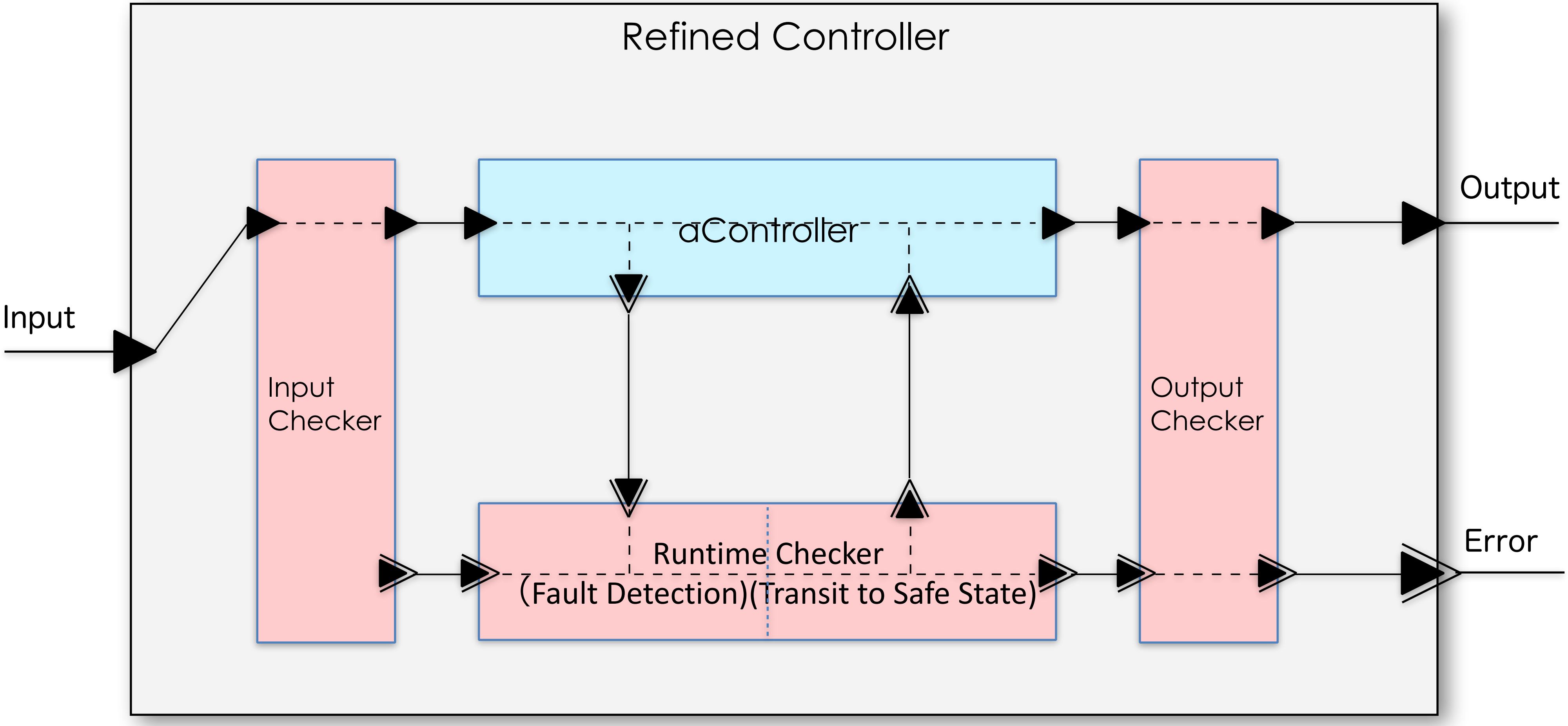
```
system pfsminp
        features
                p_in  : in  event data port;
                p_out : out event data port;
        flows
                fl10 : flow path p_in -> p_out
                { latency => 1 Ms .. 4 Ms; };
```



Describe estimated Latency in the flow path

Calculation of FTTI

- To calculate FTTI we need the various flow paths



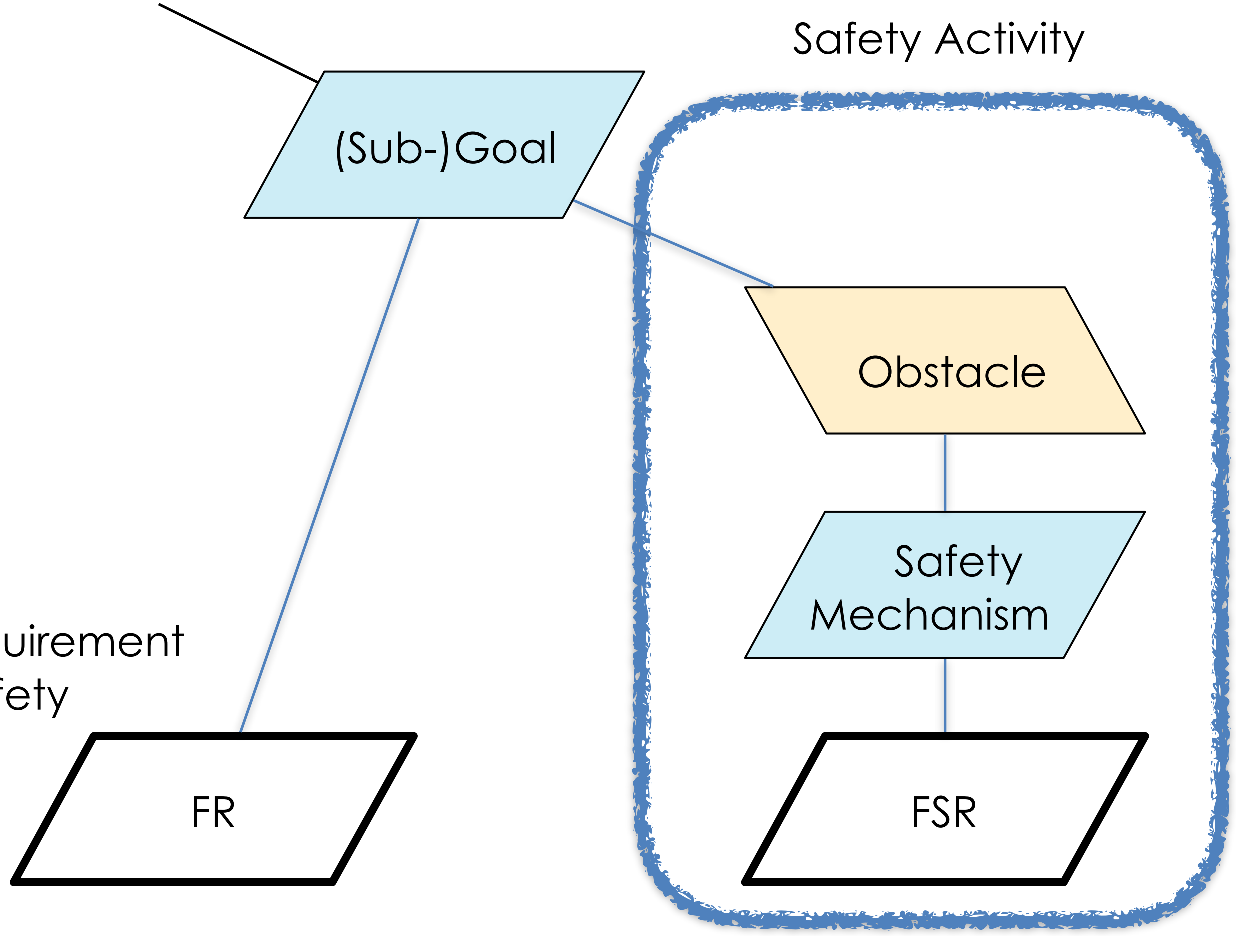
five issues

- Item ?
  - Item sketch (static & dynamic model)
- Safety activity in the whole development process
  - Use goal model
- Finding Hazards
  - Guide words, Situation-Scenario Matrix (SSM)
- How to calculate controllability for ASIL
  - Driver model, SSM
- Several “times”
  - AADL and flow model

Conclusion

- To support the concept phase of ISO 26262, we propose the practical approach. This is manly based on the goal model and we add new features.
  - Item Sketch
  - Scenario-Situation Matrix (SSM)
  - Driver Model
  - General functional safety mechanism

Summarize by goal model



FR: Functional Requirement
FRS: Functional Safety Requirement

Development and Safety Activity by the KAOS Goal model