Dependable System Development by D-Case and SysML Collaboration

Demonstration

Index

1. Abstract
2. Dependable System Development
   by D-Case and SysML Collaboration
3. Sample of Applying this Method
   ～ In-Vehicle System Development Complying with ISO26262 ～

Abstract

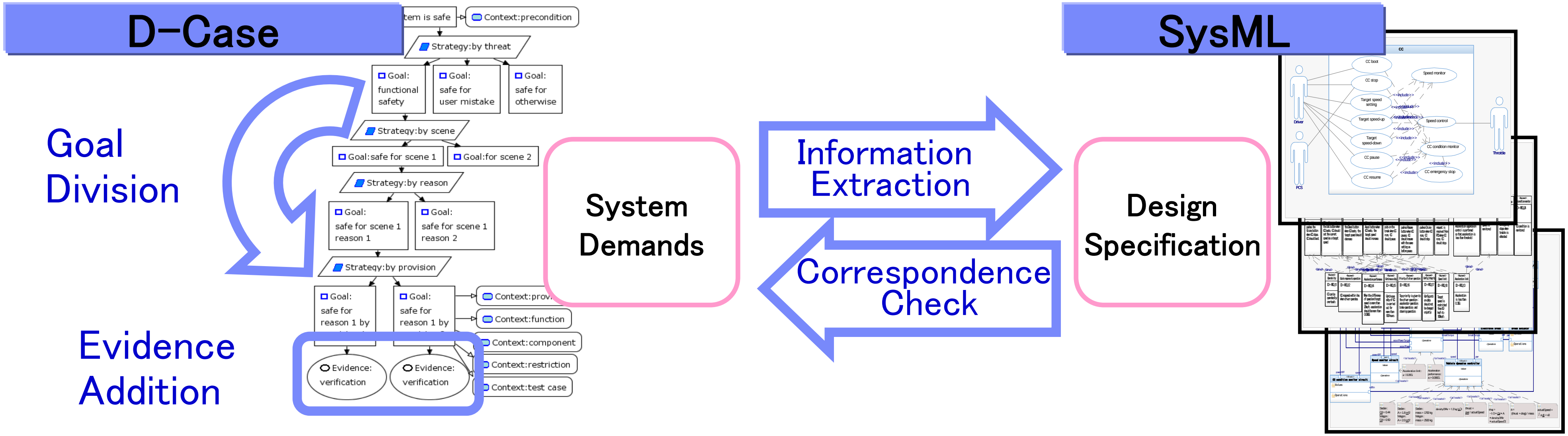
Scope and Approach

Scope

This document shows a method realizing the development of the dependable system.

Approach

This documents shows an approach which consistently realizes collaboration of D-Case and SysML model from upper process to lower process.



D-Case

Goal Division

Evidence Addition

System Demands

Information Extraction

Correspondence Check

Design Specification

SysML

Achievement

Guide and Example for D-Case and SysML model collaboration are provided.

Modeling Guide

- Modeling Flow of D-Case and SysML
- Style Guide for D-Case Node
- Collaboration of D-Case and SysML Model Description

Example

This document shows an example of the application of in-vehicle system which complies with ISO26262, the global standard of functional safety for vehicles.

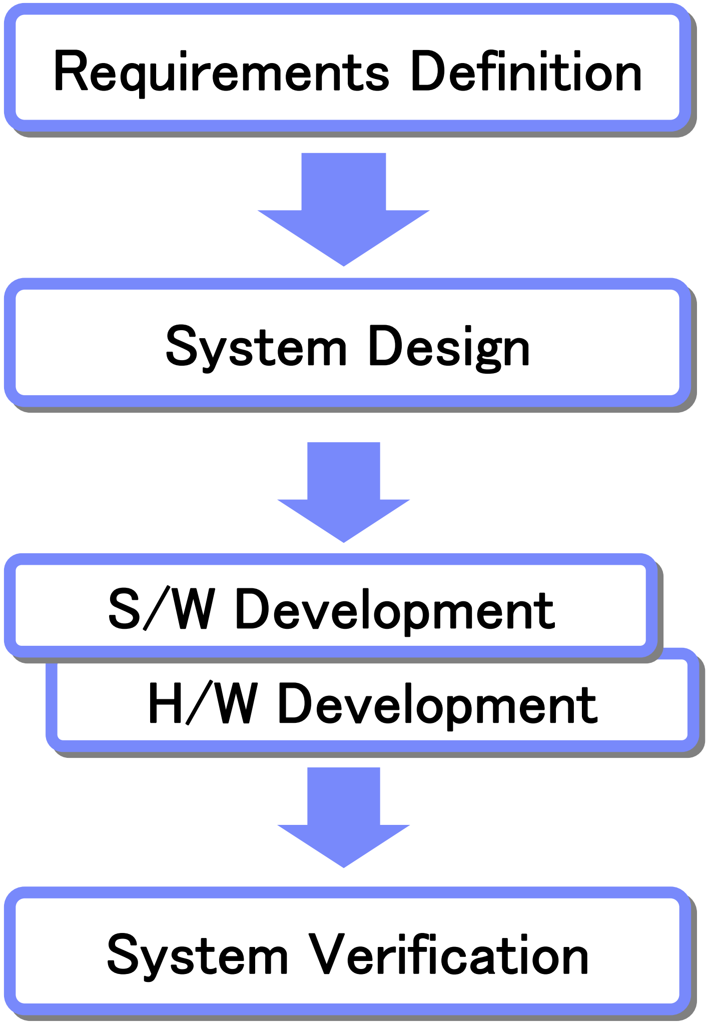
Target Reader

This document assumes the developers as readers who examine developing a dependable system.

Dependable System Development by D-Case and SysML Collaboration

What is Asked for Developing a Dependable System

Development Phase



What is Asked

System Demand

System demands should satisfy dependability.

Design Specification

Design specifications should reflect the demands correctly.

Verification Result

Verification results should account for satisfying dependability.

Issue of Dependable System Development

1 System demands are not derived just enough

System demands should be derived by removing all the factors which inhibit dependability.

2 Design specifications are not derived just enough

- System design should be performed by utilizing the design information which is included in the demands based on dependability.
- Derived design specifications should be verified just enough by checking with the demands based on dependability.

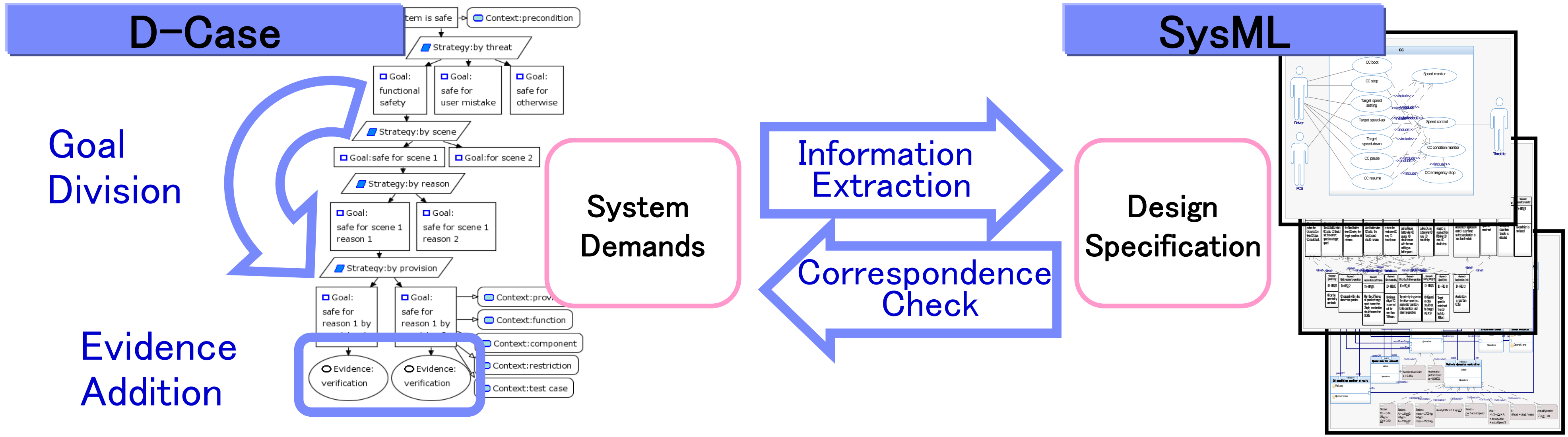
3 Verification results are not accounted how they satisfy the demands or design specifications

Verification results should be associated to the demands or design specifications and their positions should be clarified.

Approach by D-Case

D-Case is Utilized to Realize the Dependability of the Target System Consistently from Upper Process to Lower Process

1	System demands should be derived by removing all the factors which inhibit dependability	Goal division by D-Case
2	System design should be performed by utilizing the design information which is included in the demands based on dependability	Information extraction by D-Case
	Derived design specifications should be verified just enough by checking with the demands based on dependability	Correspondence check by D-Case
3	Verification results should be associated to the demands or design specifications and their positions should be clarified	Evidence addition to D-Case


D-Case

SysML

Goal Division

Evidence Addition

System Demands

Information Extraction

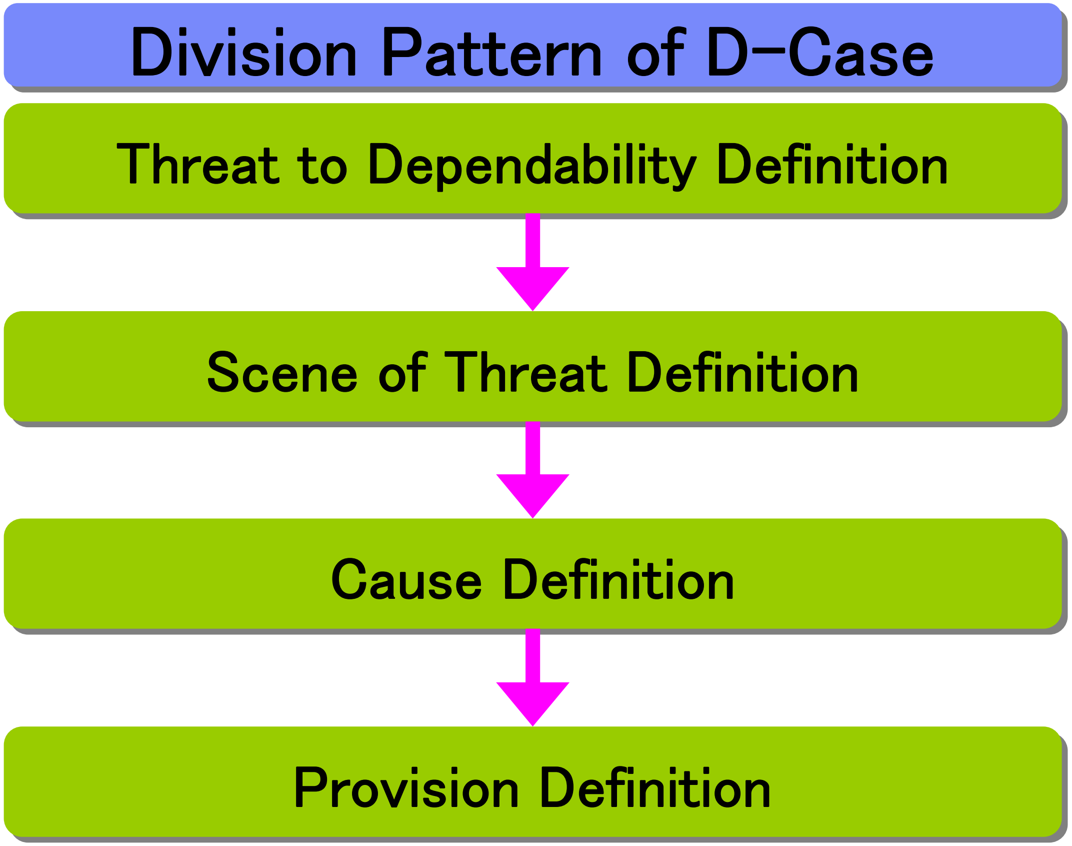
Correspondence Check

Design Specification

Approach ～ Goal Division by D-Case

1 Just enough derivation of system demands

By extracting all the factors which inhibit dependability and marshaling their provisions as system demands.



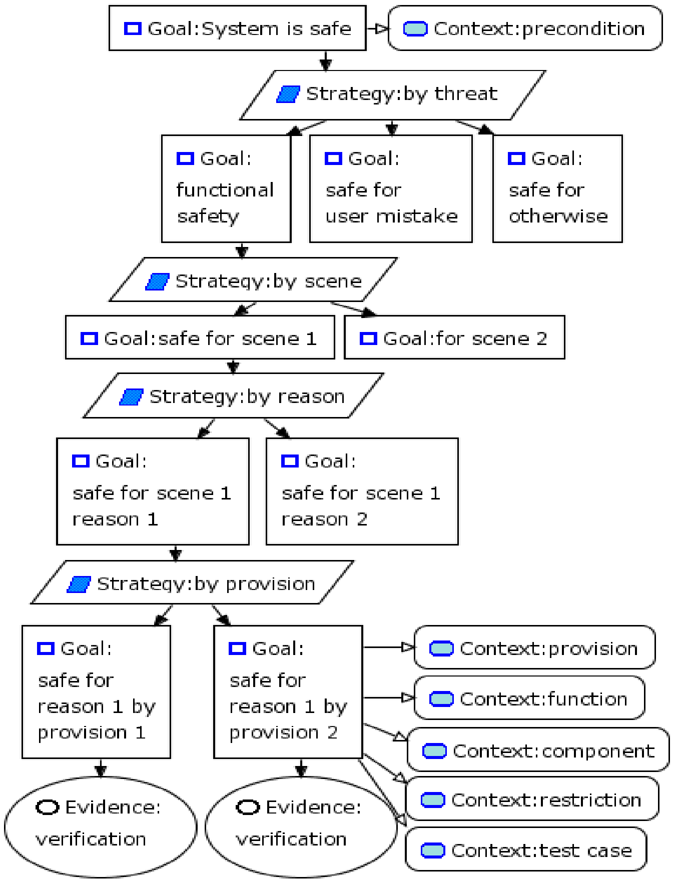
Division Pattern of D-Case

Threat to Dependability Definition

Scene of Threat Definition

Cause Definition

Provision Definition

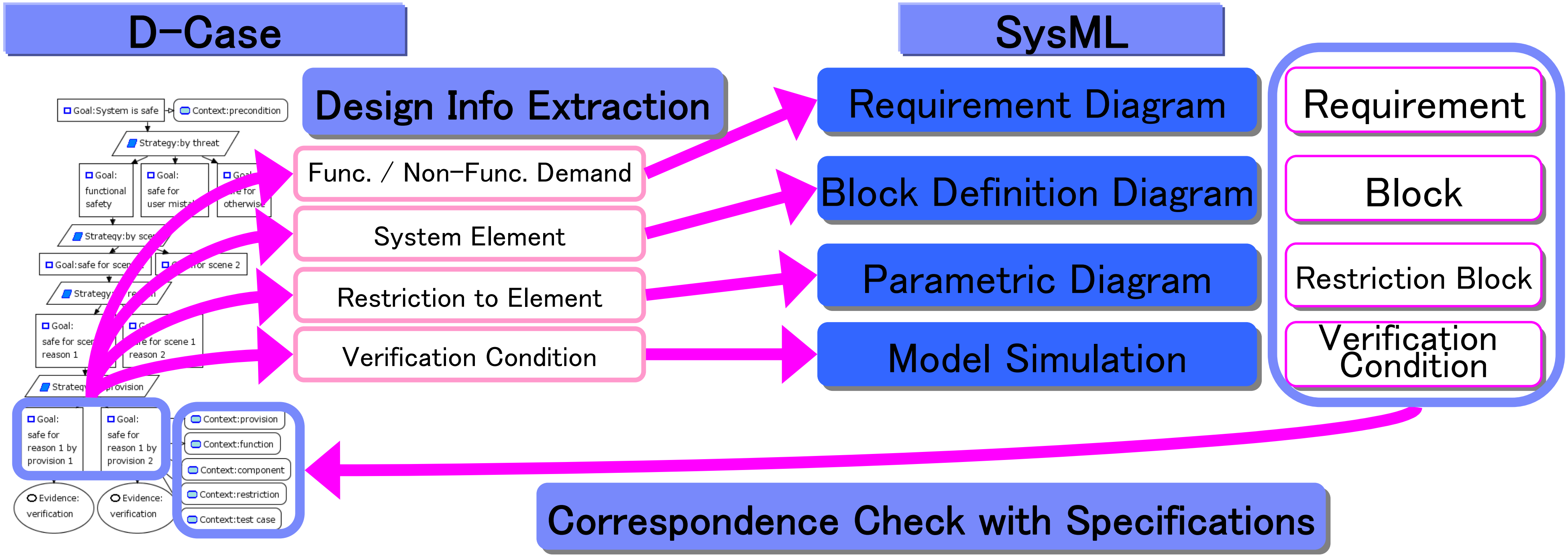


Approach ～ Info Extraction and Correspondence Check by D-Case

2 Correct derivation of design specifications

By extracting design information contained by demands derived from dependability.

By correspondence check between design specifications derived and demands based on dependability.



D-Case

SysML

Design Info Extraction

Func. / Non-Func. Demand

System Element

Restriction to Element

Verification Condition

Requirement Diagram

Block Definition Diagram

Parametric Diagram

Model Simulation

Requirement

Block

Restriction Block

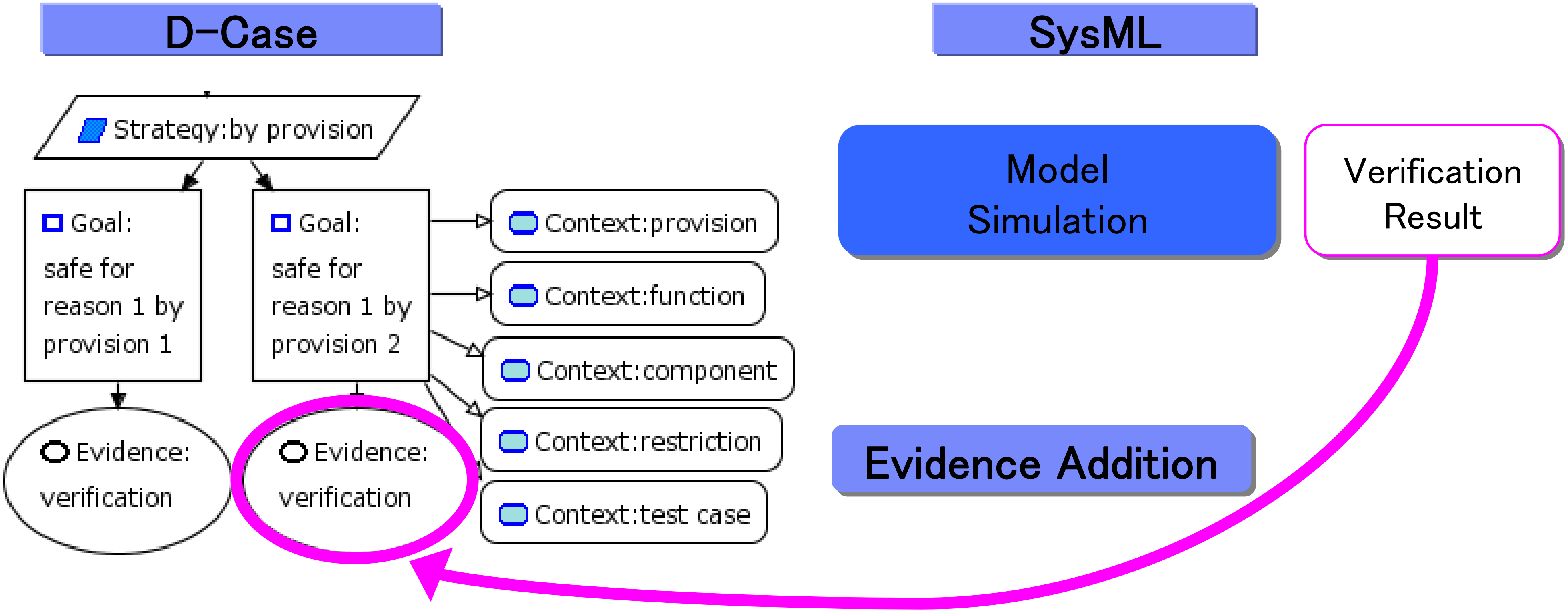
Verification Condition

Correspondence Check with Specifications

Approach ～ Evidence Addition to D-Case

3 Clarifying relationship of verification results, demands, and design specifications

By associating verification result to goal as evidence and correspondence check with relate demands or design specifications



D-Case

Strategy:by provision

Goal: safe for reason 1 by provision 1

Goal: safe for reason 1 by provision 2

Evidence: verification

Evidence: verification

Context:provision

Context:function

Context:component

Context:restriction

Context:test case

SysML

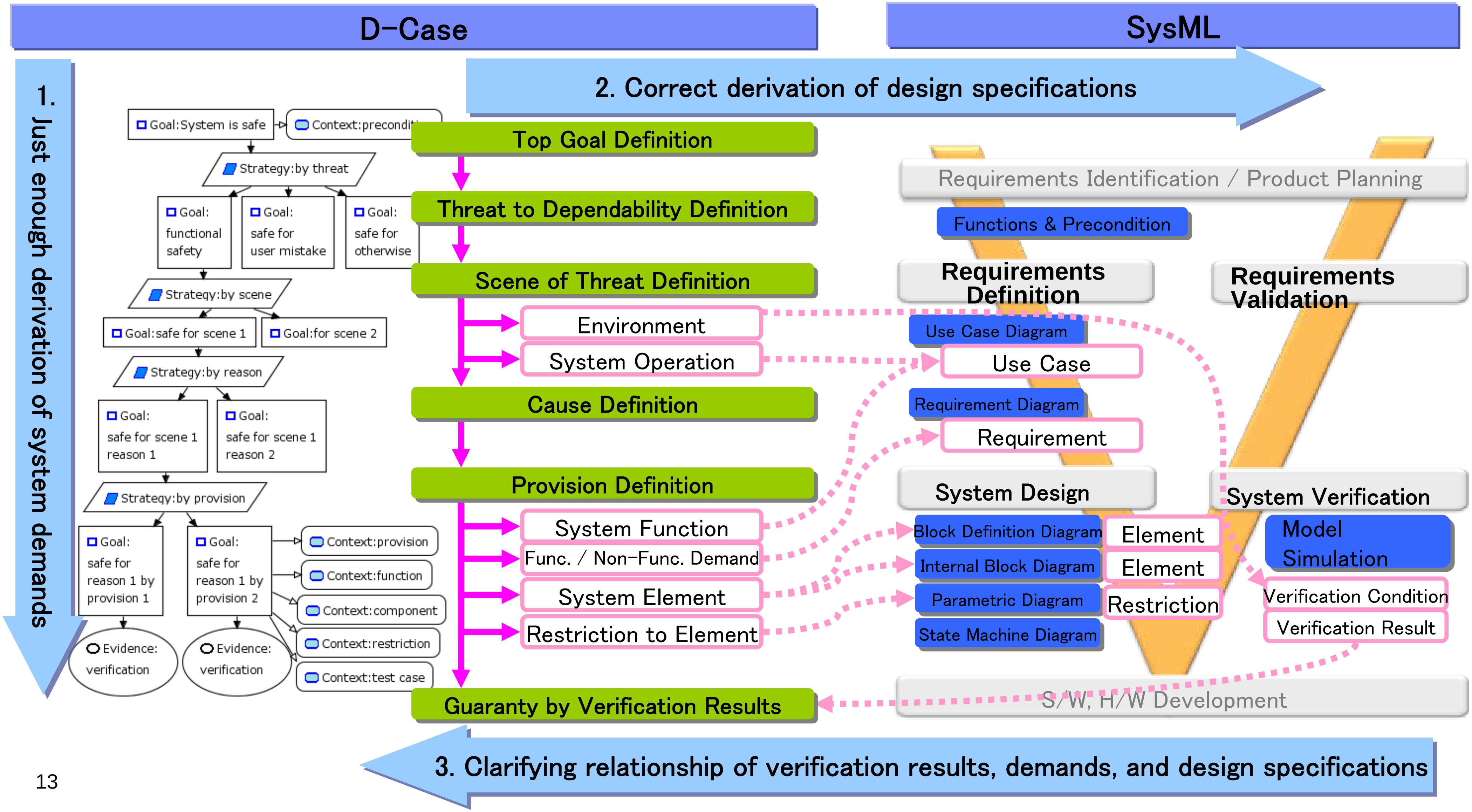
Model Simulation

Verification Result

Evidence Addition

Outline of Dependable System Development Method

Collaboration of D-Case and SysML Modeling



D-Case

SysML

1. Just enough derivation of system demands

2. Correct derivation of design specifications

3. Clarifying relationship of verification results, demands, and design specifications

Goal:System is safe

Context:precondit

Strategy:by threat

Goal: functional safety

Goal: safe for user mistake

Goal: safe for otherwise

Strategy:by scene

Goal:safe for scene 1

Goal:for scene 2

Strategy:by reason

Goal: safe for scene 1 reason 1

Goal: safe for scene 1 reason 2

Strategy:by provision

Goal: safe for reason 1 by provision 1

Goal: safe for reason 1 by provision 2

Context:provision

Context:function

Context:component

Context:restriction

Context:test case

Evidence: verification

Evidence: verification

Top Goal Definition

Threat to Dependability Definition

Scene of Threat Definition

Environment

System Operation

Cause Definition

Provision Definition

System Function

Func. / Non-Func. Demand

System Element

Restriction to Element

Guaranty by Verification Results

Requirements Identification / Product Planning

Functions & Precondition

Requirements Definition

Requirements Validation

Use Case Diagram

Use Case

Requirement Diagram

Requirement

System Design

System Verification

Block Definition Diagram

Element

Internal Block Diagram

Element

Parametric Diagram

Restriction

State Machine Diagram

Model Simulation

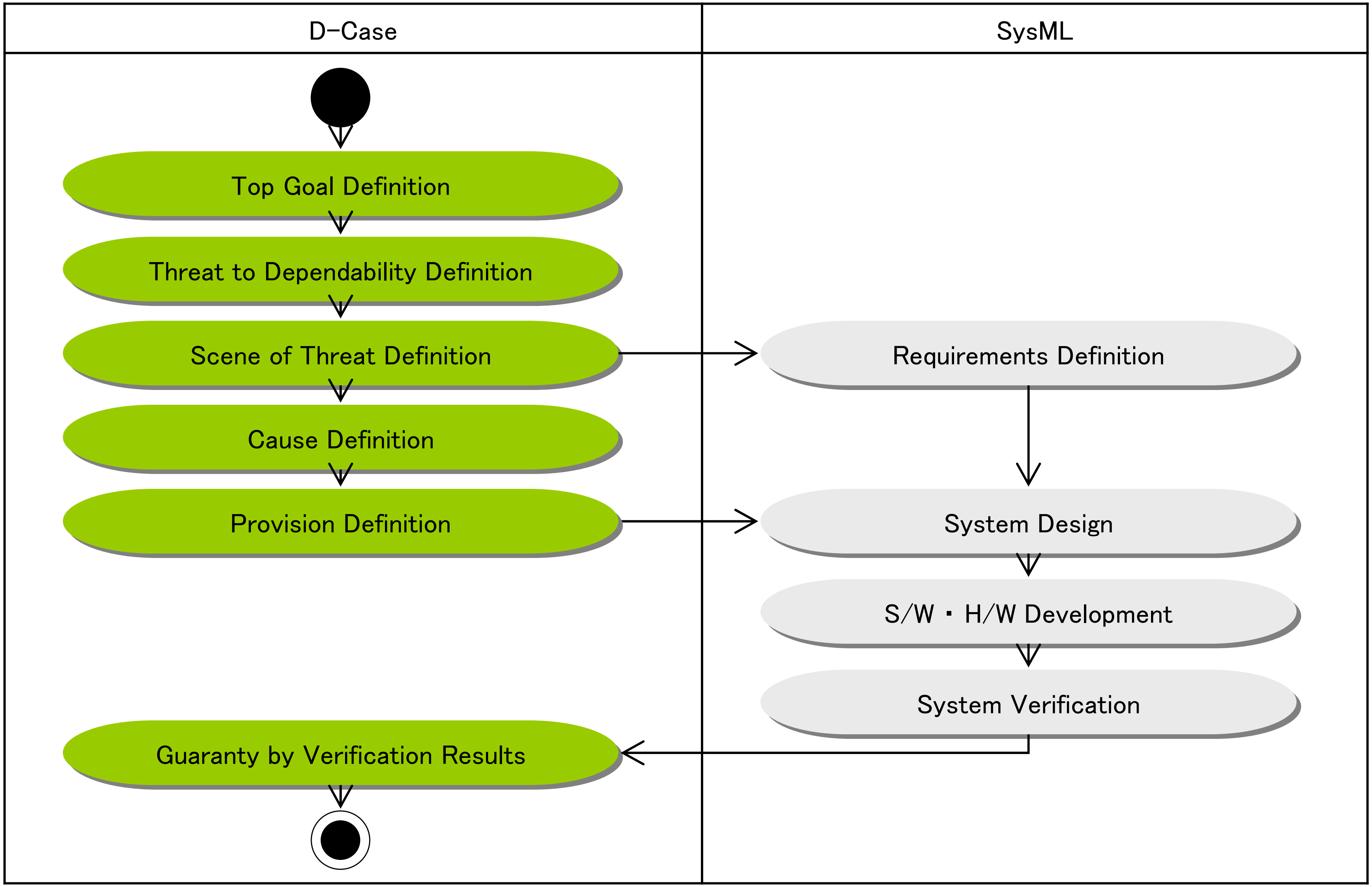
Verification Condition

Verification Result

S/W, H/W Development

Flow of Dependable System Development Method

Dependable System Development by D-Case and SysML Collaboration



Flow Detail – Top Goal and Threat to Dependability Definition

Define Top Goal in Terms of Dependability, Categorize Threats

D-Case

Top Goal Definition

Define top goal and precondition on dependability

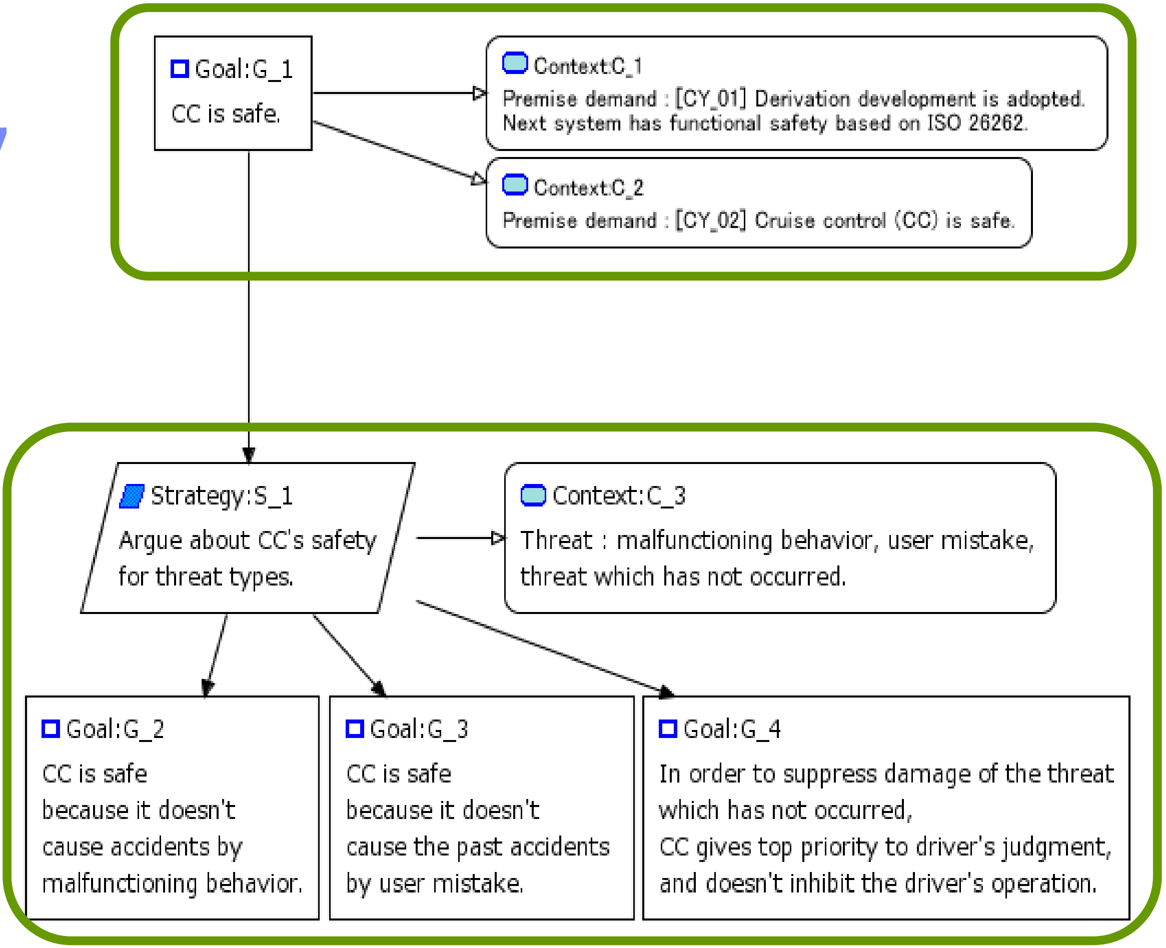
Extract characteristics system needs and preconditions for dependability.

Threat to Dependability Definition

Marshal all threats inhibiting dependability

Decompose top goal to threats by malfunctions, user mistakes, and others.

Example. Safety of In-Vehicle Cruise Control (CC) System



Goal:G_1
CC is safe.

Context:C_1
Premise demand : [CY_01] Derivation development is adopted. Next system has functional safety based on ISO 26262.

Context:C_2
Premise demand : [CY_02] Cruise control (CC) is safe.

Strategy:S_1
Argue about CC's safety for threat types.

Context:C_3
Threat : malfunctioning behavior, user mistake, threat which has not occurred.

Goal:G_2
CC is safe because it doesn't cause accidents by malfunctioning behavior.

Goal:G_3
CC is safe because it doesn't cause the past accidents by user mistake.

Goal:G_4
In order to suppress damage of the threat which has not occurred, CC gives top priority to driver's judgment, and doesn't inhibit the driver's operation.

Flow Detail – Scene of Threat Definition

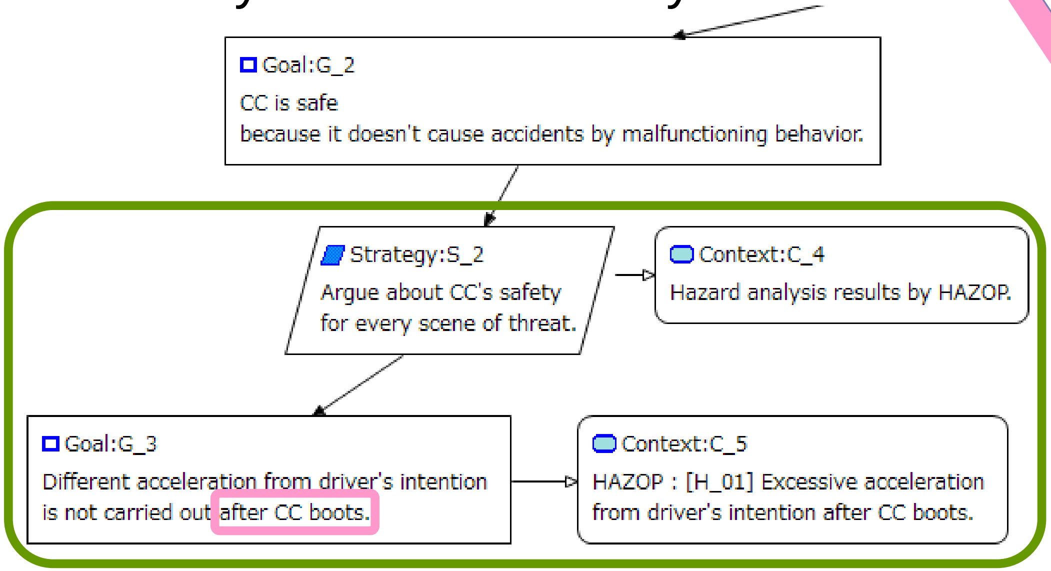
Define System Environment and Operations by Clarifying Scenes of Threat

D-Case

Scene of Threat Definition

Clarify all the scenes inhibiting dependability

- ●Clarify operations causing threats
- ●Clarify environment of system


Goal:G_2
CC is safe
because it doesn't cause accidents by malfunctioning behavior.

Strategy:S_2
Argue about CC's safety
for every scene of threat.

Context:C_4
Hazard analysis results by HAZOP.

Goal:G_3
Different acceleration from driver's intention
is not carried out after CC boots.

Context:C_5
HAZOP : [H_01] Excessive acceleration
from driver's intention after CC boots.

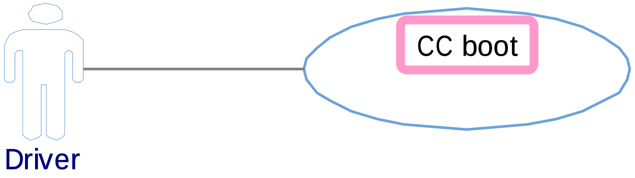
Reflect

SysML

Use Case Diagram | **Use Case**

Clarify association between user and operation of system

Update use case by operations causing threats



Model Simulation | **Verification Condition**

Clarify verification condition

Update condition of verification scenario by environment of system

Flow Detail – Cause Definition and Provision Definition

Analyze Cause of Threat, Make Provisions

D-Case

Cause Definition

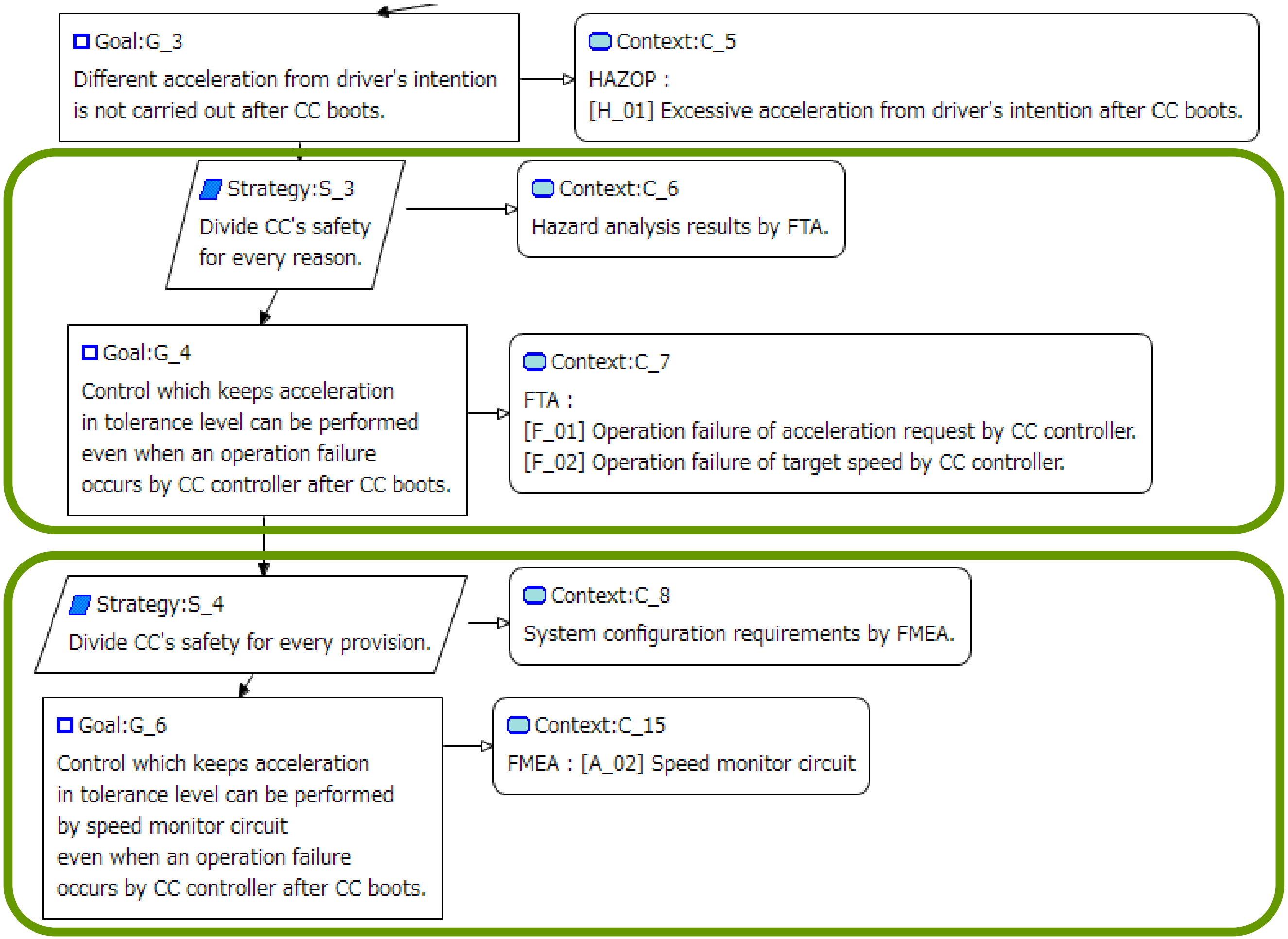
Cause analysis for provision

Analyze cause of threats for each scene

Provision Definition

Make provision for each cause

Marshal system demands from provisions for each cause of threat.



Goal:G_3
Different acceleration from driver's intention is not carried out after CC boots.

Context:C_5
HAZOP :
[H_01] Excessive acceleration from driver's intention after CC boots.

Strategy:S_3
Divide CC's safety for every reason.

Context:C_6
Hazard analysis results by FTA.

Goal:G_4
Control which keeps acceleration in tolerance level can be performed even when an operation failure occurs by CC controller after CC boots.

Context:C_7
FTA :
[F_01] Operation failure of acceleration request by CC controller.
[F_02] Operation failure of target speed by CC controller.

Strategy:S_4
Divide CC's safety for every provision.

Context:C_8
System configuration requirements by FMEA.

Goal:G_6
Control which keeps acceleration in tolerance level can be performed by speed monitor circuit even when an operation failure occurs by CC controller after CC boots.

Context:C_15
FMEA : [A_02] Speed monitor circuit

Flow Detail – Requirements Definition

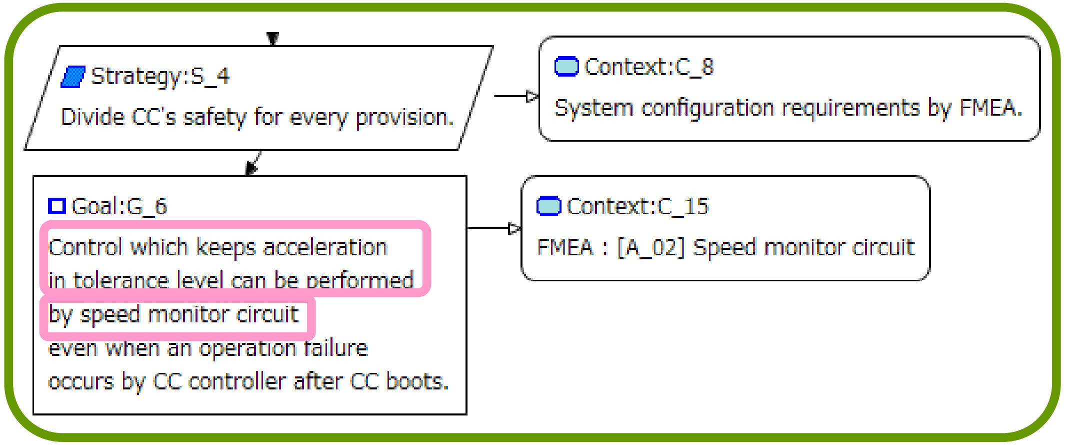
Clarify System Demands from Provisions, Derive Requirements

D-Case

Provision Definition

Clarify system demands for each provision

- Clarify functional demand
- Clarify non-functional demand
- Clarify system configuration requirement
- Clarify restriction to element


Strategy:S_4
Divide CC's safety for every provision.

Context:C_8
System configuration requirements by FMEA.

Goal:G_6
Control which keeps acceleration in tolerance level can be performed by speed monitor circuit even when an operation failure occurs by CC controller after CC boots.

Context:C_15
FMEA : [A_02] Speed monitor circuit

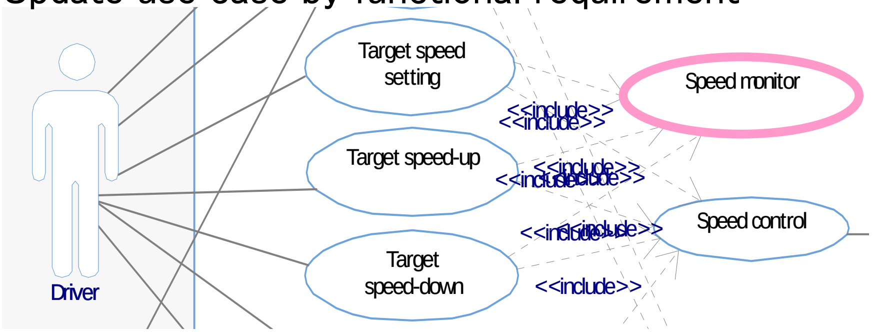
Reflect

SysML

Use Case Diagram | **Use Case**

Clarify relation between user and system operation

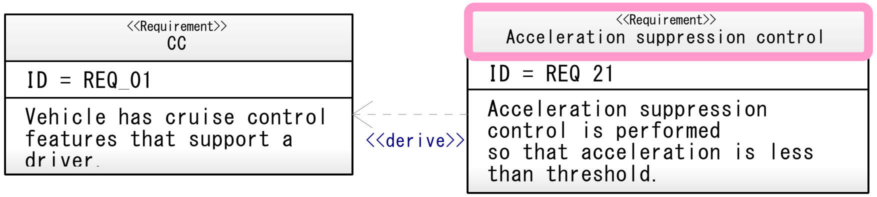
Update use case by functional requirement



Requirement Diagram | **Func. / Non-func. requirement**

Clarify func. / non-func. requirement

Update requirement by func./non-func. demand


<<Requirement>>
CC
ID = REQ_01
Vehicle has cruise control features that support a driver.

<<derive>>

<<Requirement>>
Acceleration suppression control
ID = REQ 21
Acceleration suppression control is performed so that acceleration is less than threshold.

Flow Detail – System Design

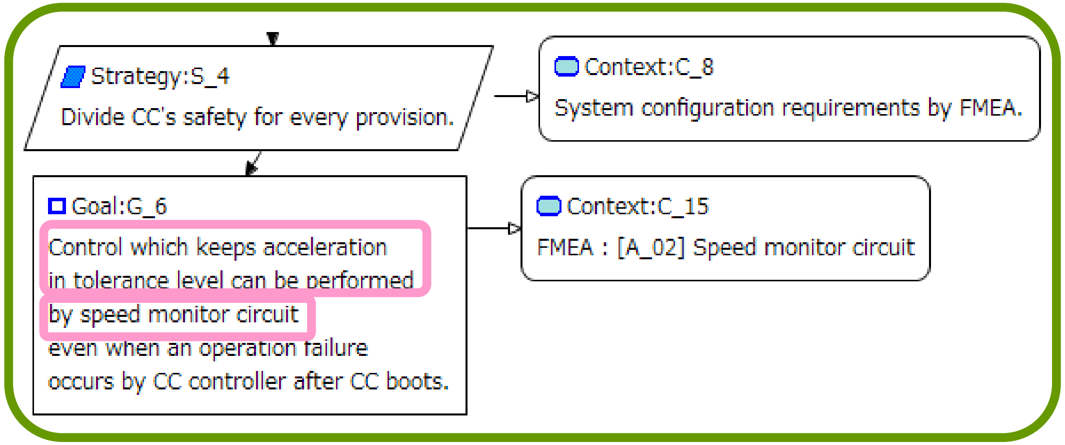
Clarify Design Specifications based on System Demands and Requirements

D-Case

Provision Definition

Clarify system demands for each provision

- Clarify functional demand
- Clarify non-functional demand
- Clarify system configuration requirement
- Clarify restriction to element



Strategy:S_4
Divide CC's safety for every provision.

Context:C_8
System configuration requirements by FMEA.

Goal:G_6
Control which keeps acceleration in tolerance level can be performed by speed monitor circuit even when an operation failure occurs by CC controller after CC boots.

Context:C_15
FMEA : [A_02] Speed monitor circuit

Reflect

SysML

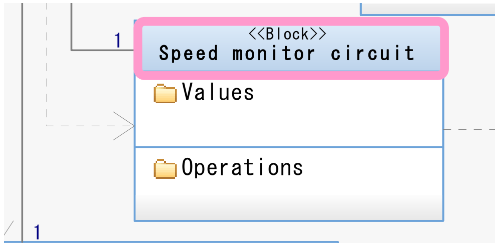
Block Definition Diagram

Internal Block Diagram

Element

Clarify system architecture

Update element by system configuration requirement

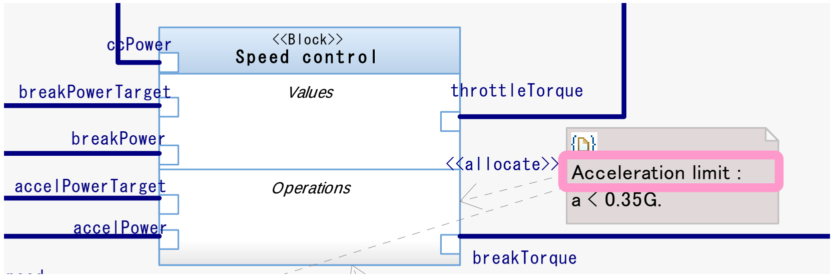


Parametric Diagram

Restriction

Clarify system precondition and restriction

Update restriction to element



Acceleration limit :
a < 0.35G.

Flow Detail – Guaranty by Verification Result

Guaranty of Satisfaction of System Demands

D-Case

SysML

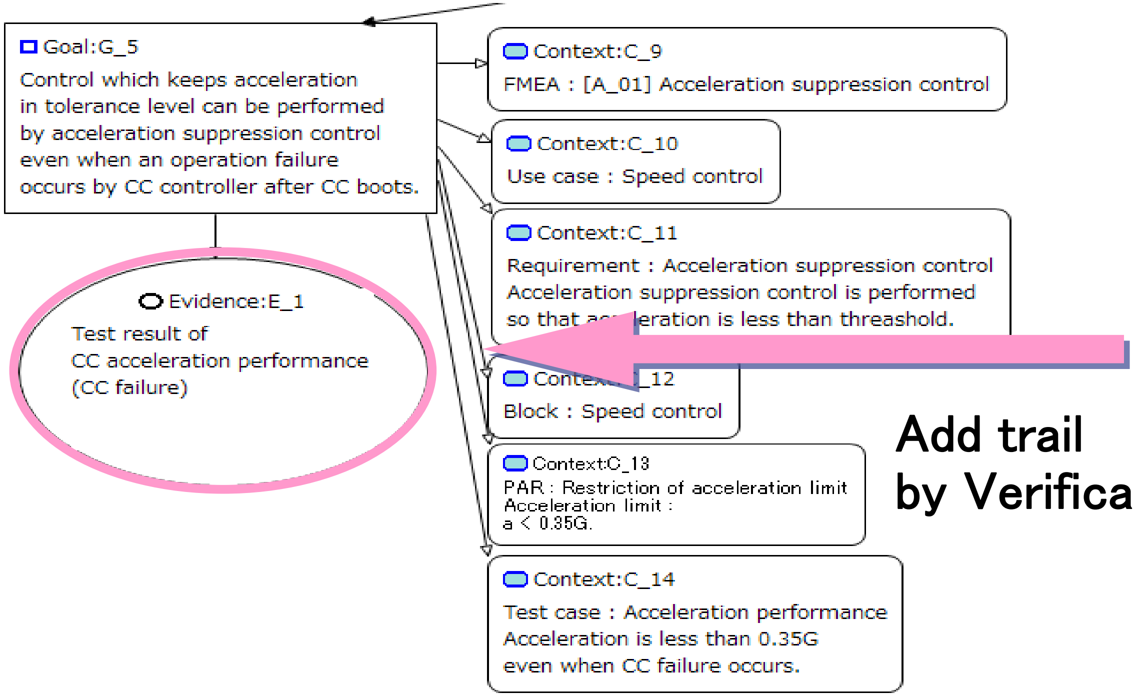
Guaranty by Verification Result

Model Simulation

Verification Result

Clarification of satisfaction of system demands

Associate verification result to goal



Goal:G_5
Control which keeps acceleration in tolerance level can be performed by acceleration suppression control even when an operation failure occurs by CC controller after CC boots.

Evidence:E_1
Test result of CC acceleration performance (CC failure)

Context:C_9
FMEA : [A_01] Acceleration suppression control

Context:C_10
Use case : Speed control

Context:C_11
Requirement : Acceleration suppression control
Acceleration suppression control is performed so that acceleration is less than threashold.

Context:C_12
Block : Speed control

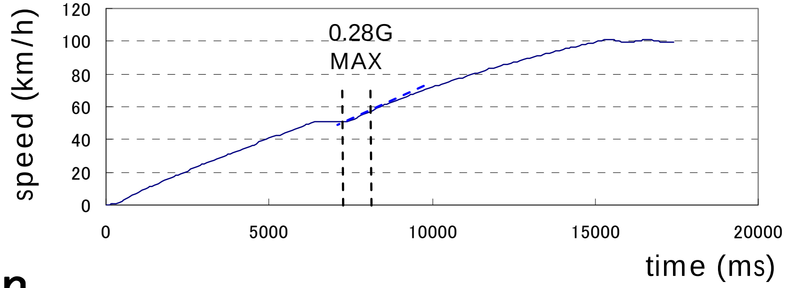
Context:C_13
PAR : Restriction of acceleration limit
Acceleration limit :
a < 0.35G.

Context:C_14
Test case : Acceleration performance
Acceleration is less than 0.35G even when CC failure occurs.

Add trail by Verification

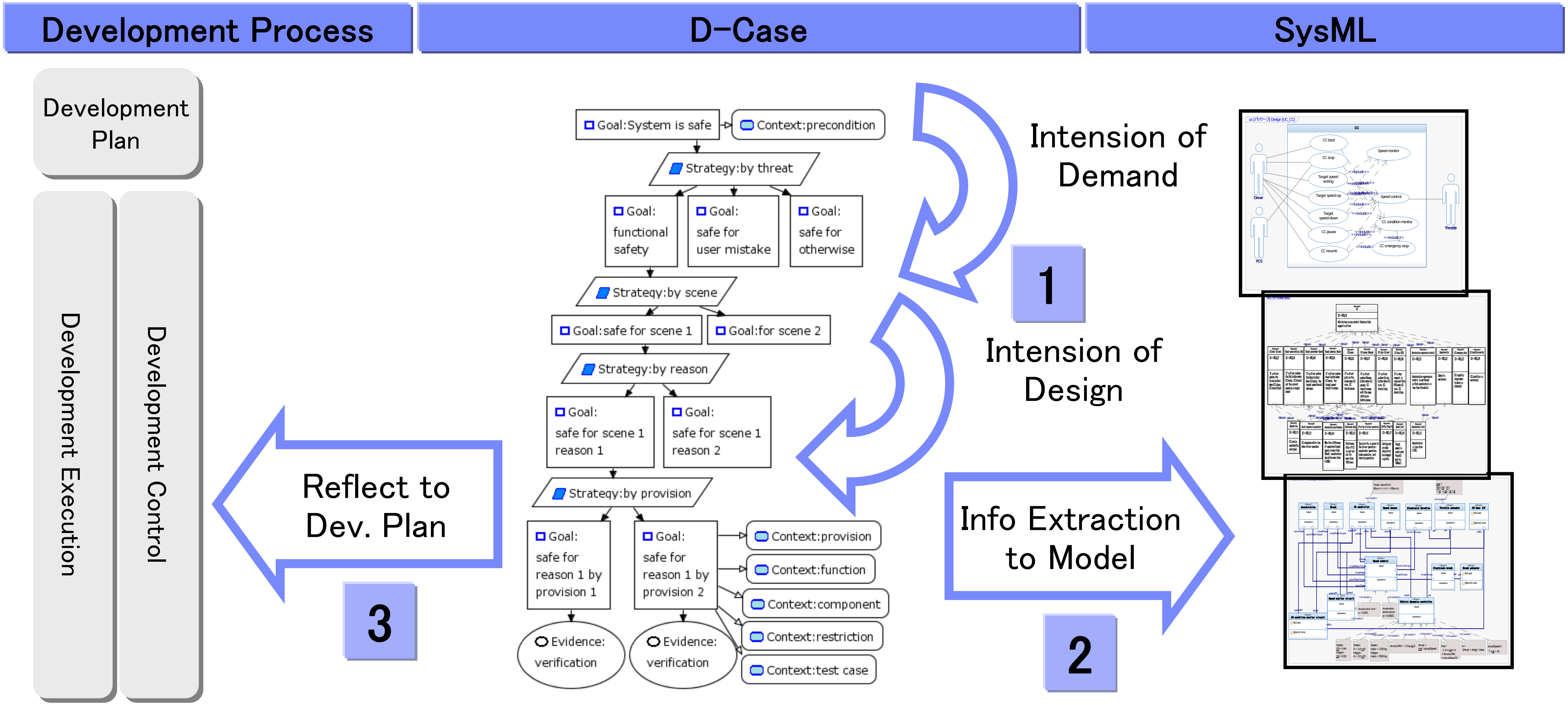
System verification by executable model

Verification result



Advantage of applying this method

1. Improvement of Quality by Reflecting Development Intents from Upper Process to Lower Process
2. Modeling from Information on D-Case
3. Improvement of Plan by Controlling Development Process by D-Case



Development Process

D-Case

SysML

Development Plan

Development Execution

Development Control

Reflect to Dev. Plan

3

Goal:System is safe

Context:precondition

Strategy:by threat

Goal: functional safety

Goal: safe for user mistake

Goal: safe for otherwise

Strategy:by scene

Goal:safe for scene 1

Goal:for scene 2

Strategy:by reason

Goal: safe for scene 1 reason 1

Goal: safe for scene 1 reason 2

Strategy:by provision

Goal: safe for reason 1 by provision 1

Goal: safe for reason 1 by provision 2

Context:provision

Context:function

Context:component

Context:restriction

Context:test case

Evidence: verification

Evidence: verification

Intension of Demand

1

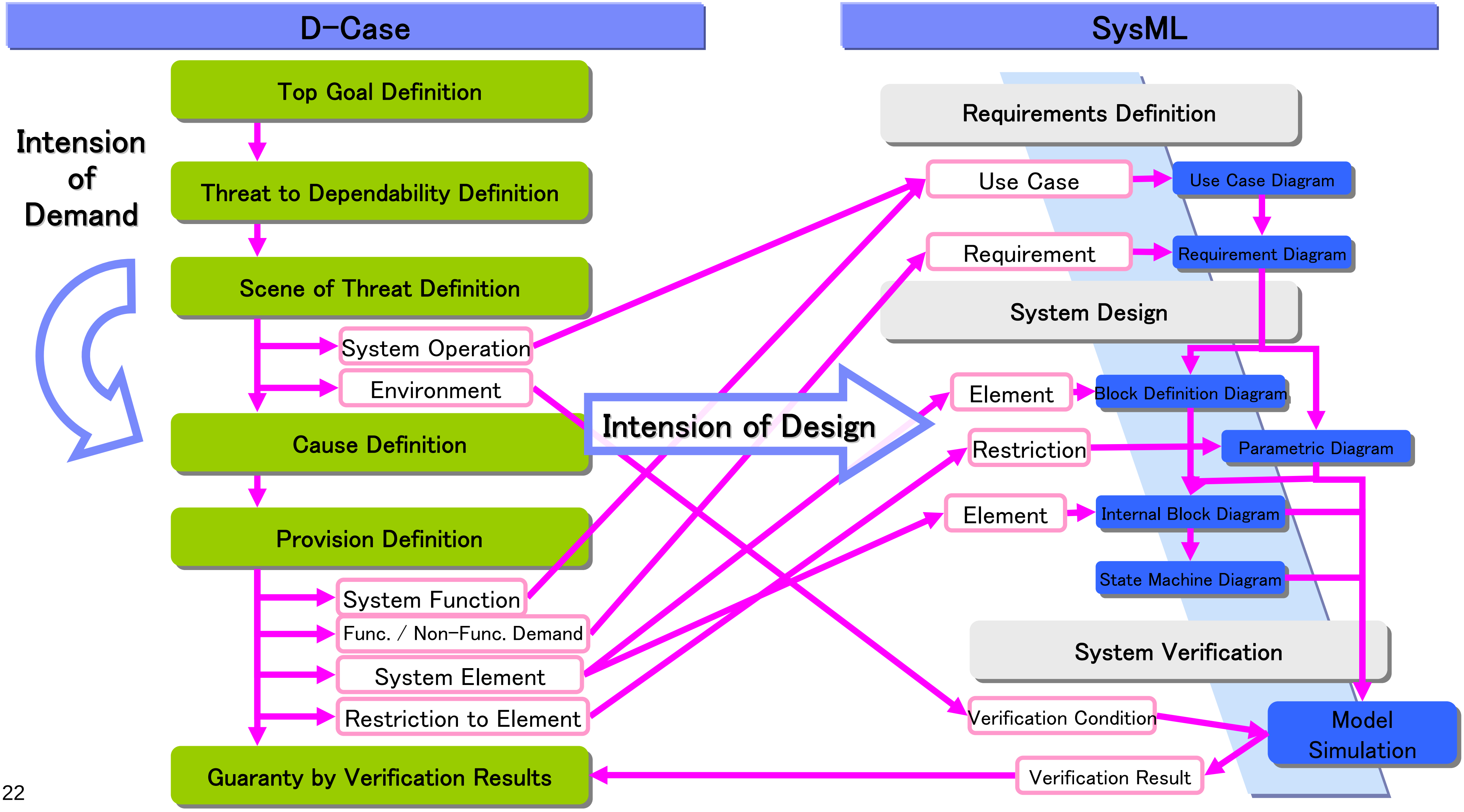
Intension of Design

Info Extraction to Model

2

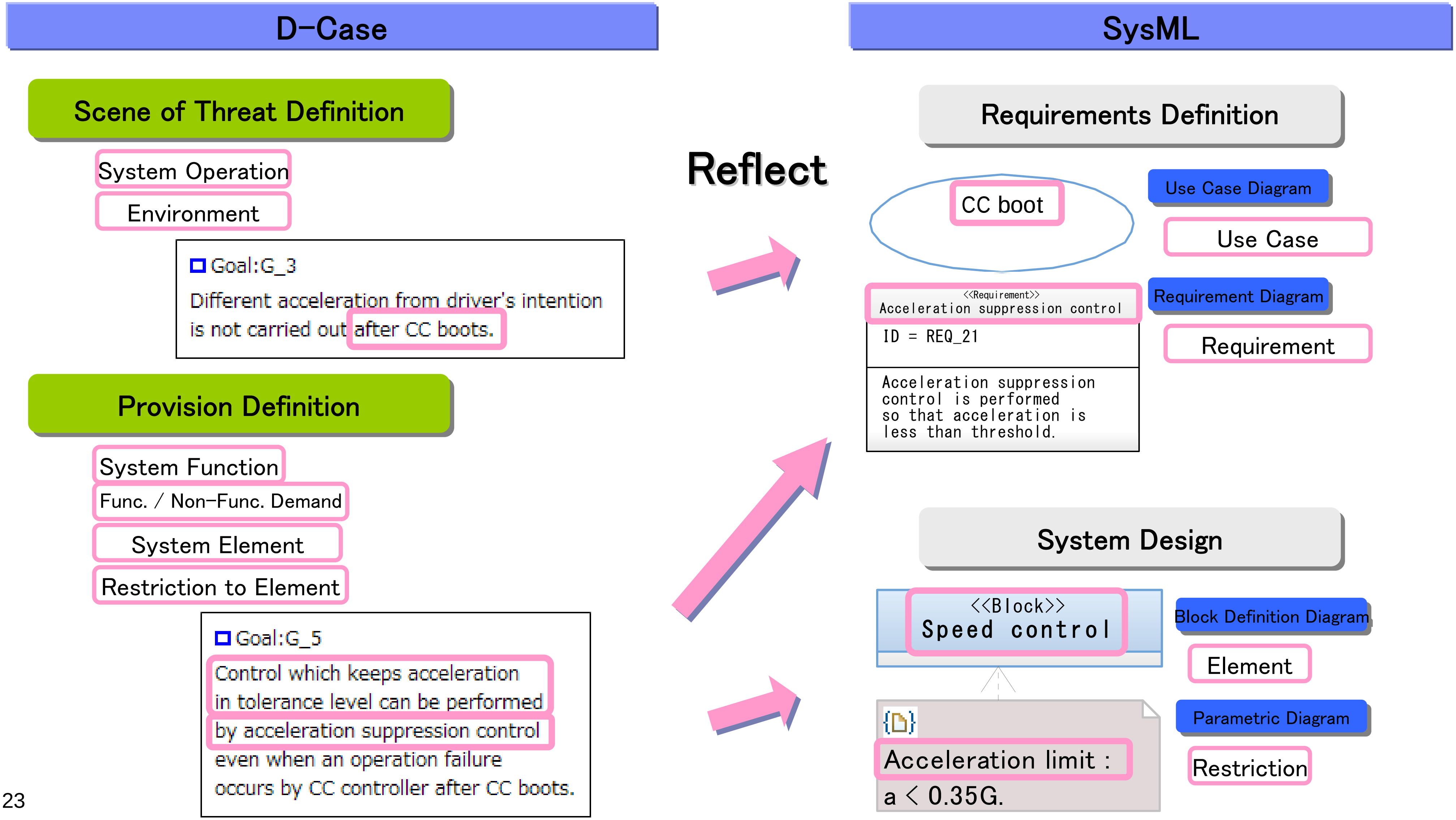
Advantage 1. Improvement of Quality by Reflecting Development Intents

Clarify Development Intents by D-Case and Reflect to SysML Model



Advantage 2. Modeling from Information on D-Case

Update SysML Model Elements based on Descriptions of D-Case



D-Case

Scene of Threat Definition

- System Operation
- Environment

□ Goal:G_3

Different acceleration from driver's intention is not carried out after CC boots.

Provision Definition

- System Function
- Func. / Non-Func. Demand
- System Element
- Restriction to Element

□ Goal:G_5

Control which keeps acceleration in tolerance level can be performed by acceleration suppression control even when an operation failure occurs by CC controller after CC boots.

Reflect

SysML

Requirements Definition

Use Case Diagram

- Use Case

```
<<Requirement>>
Acceleration suppression control
ID = REQ_21
Acceleration suppression
control is performed
so that acceleration is
less than threshold.
```

Requirement Diagram

- Requirement

System Design

```
<<Block>>
Speed control
```

Block Definition Diagram

- Element

Acceleration limit :
a < 0.35G.

Parametric Diagram

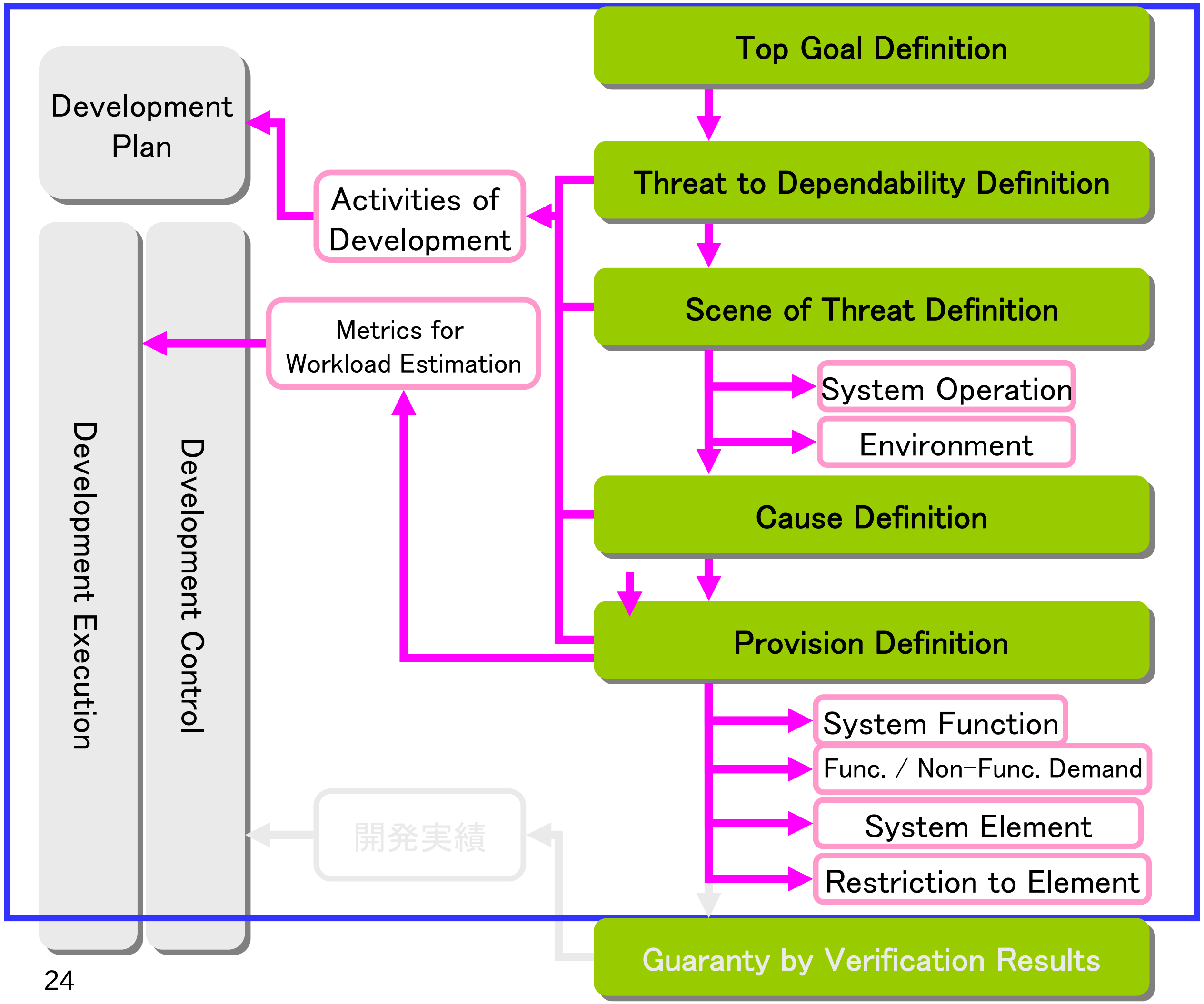
- Restriction

Advantage 3. Improvement of Plan by Controlling Development Process

Improve Development Plan by Just Enough Provisioning

Development Plan

D-Case

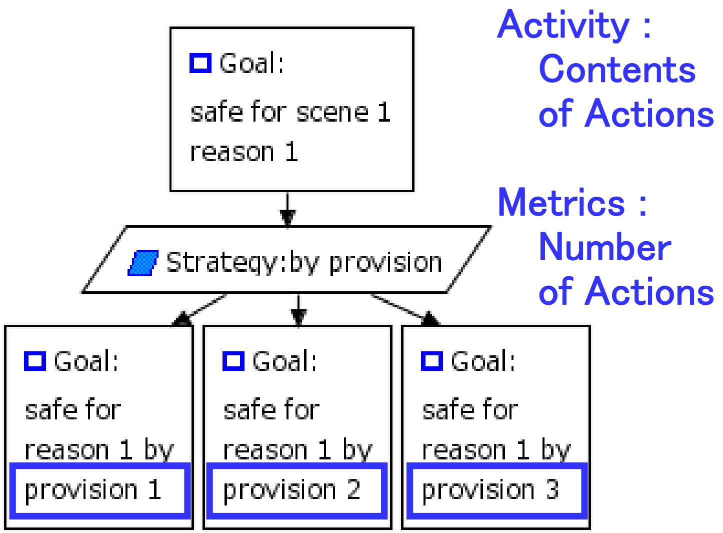


- ● Extract development activities cyclpaefically

  ➡ Improve development plan

- ● Extract metrics for workload estimation

  ➡ Improve man-hour plan



Goal:
safe for scene 1 reason 1
Strategy:by provision
Goal:
safe for reason 1 by provision 1
Goal:
safe for reason 1 by provision 2
Goal:
safe for reason 1 by provision 3
Activity : Contents of Actions
Metrics : Number of Actions

Guide for Supporting this Method

Modeling Guide and Template are Provided

Modeling Guide

- Modeling Flow of D-Case and SysML
- Style Guide for D-Case Node
- Collaboration of D-Case and SysML Model Description

Template

- Pattern of Goal Structure in D-Case and Node Notation
- Structure of SysML Model

Target of Guide

Derivational Development of System in the Automotive Domain to apply Functional Safety Requirements

Sample of Applying this Method

～ In-Vehicle System Development Complying with ISO26262 ～

Correspondence Relation between this Method and ISO26262

This Method Corresponds ISO26262 Safety Lifecycle

ISO26262

3.5 Item definition
Item identification

3.7 Hazard analysis and risk assessment
Hazard identification

3.8 Functional safety concept
Decomposition by functional safety requirement

4.6 Specification of the technical safety requirements
Decomposition by technical safety requirement

4.7 System design
Guaranty by Verification Results

D-Case

Top Goal Definition

Definition of top goal, precondition about safety

Threat to Dependability Definition

Clarification of threats inhibiting safety

Scene of Threat Definition

Definition of system environment, operations

Cause Definition

Cause analysis for provision

Provision Definition

Definition of system demands by provisions

Guaranty by Verification Results

Demonstration of satisfaction of system demands

SysML

Use Case Diagram

Definition of system users and operations

Requirement Diagram

Definition of functional,
non-functional requirements

Block Definition Diagram

Definition of system architecture

Parametric Diagram

Definition of system restrictions

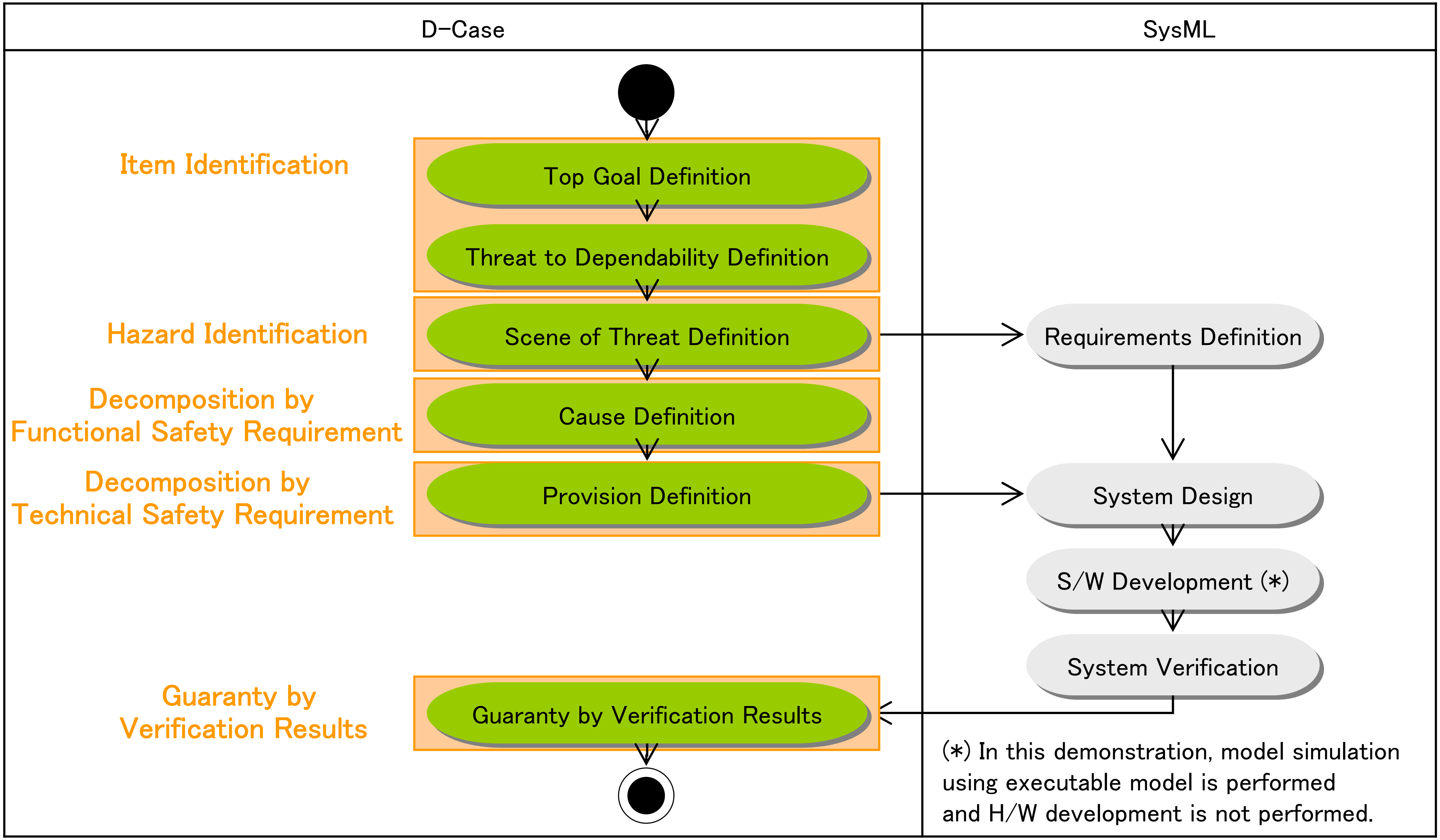
Model Simulation

Definition of verification condition

System verification by executable model

Development flow in this Demonstration

Develop a System Complying with ISO26262 by D-Case and SysML Collaboration


D-Case
SysML
Item Identification
Top Goal Definition
Threat to Dependability Definition
Hazard Identification
Scene of Threat Definition
Decomposition by Functional Safety Requirement
Cause Definition
Decomposition by Technical Safety Requirement
Provision Definition
Guaranty by Verification Results
Guaranty by Verification Results
Requirements Definition
System Design
S/W Development (*)
System Verification

(*) In this demonstration, model simulation using executable model is performed and H/W development is not performed.

D-Case and SysML Modeling based on ISO26262

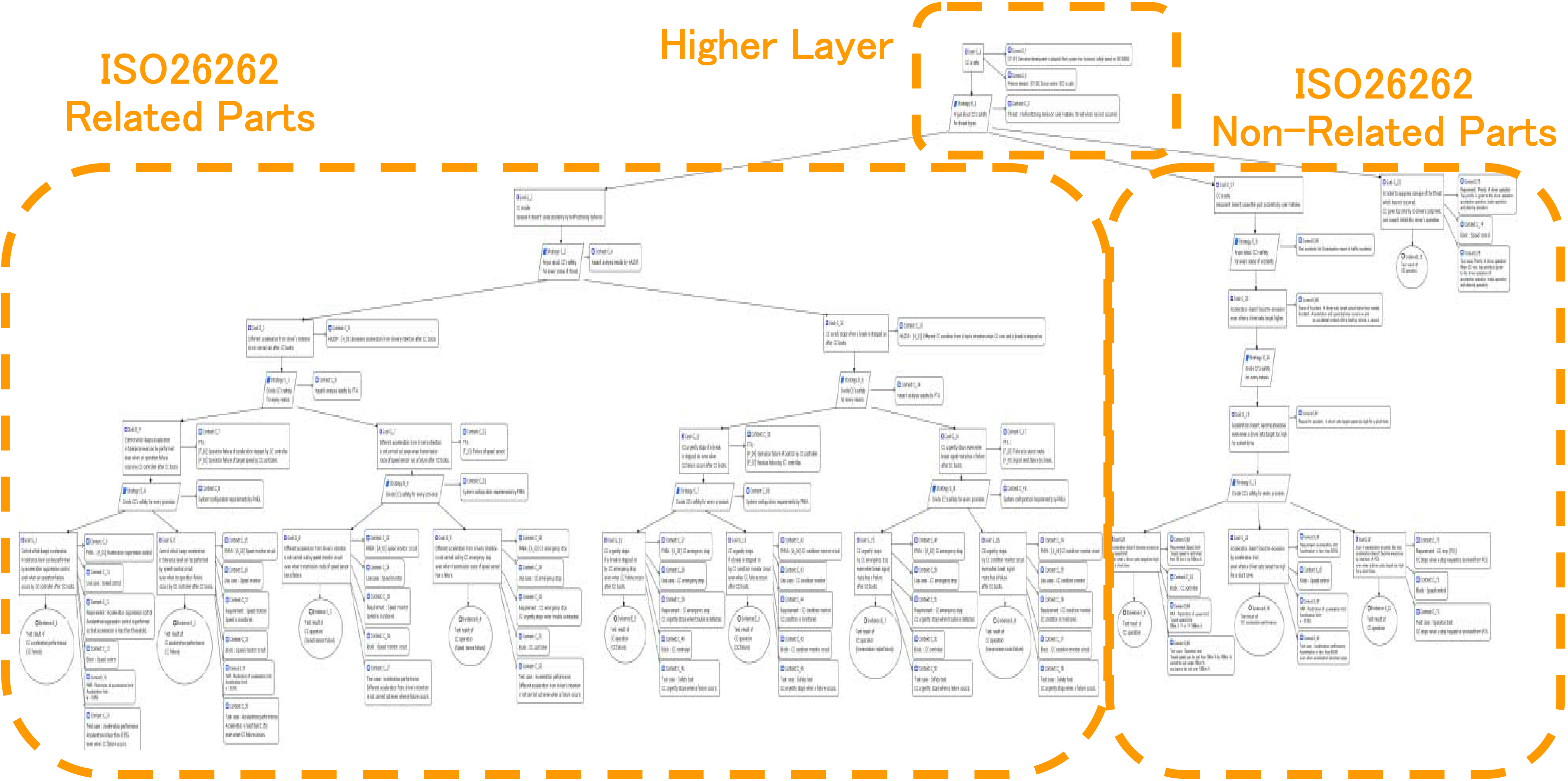
- D-Case Structure based on Safety Lifecycle
- SysML Modeling Collaborated with D-Case

Contents of Modeling Guide

Target	D-Case		SysML	
Category	D-Case Structure	Node Notation	Association from D-Case	Association to D-Case
Item	Item Definition	Goal to Achieve, Environment and Restriction	–	–
	Identification of Hazards	Environment and Operation of System	Environment and Operation of System	Use Case Verification condition
	Decomposition by Functional Safety Requirements	Detailed Cause to Take Actions	–	–
	Decomposition by Technical Safety Requirements	System Requirement	Functional and non-functional requirement, Functional Block and Restriction	Use Case, requirement, Functional Block and Restriction
	Guaranty by Verification Results	Information required to Verification	Condition and Processing of Control	Verification Result

Overall Structure of D-Case Developed

Decompose the Top Goals to ISO26262 Related Parts and Non-Related Parts



Safety and Reliability Requirements on ISO26262

Safety and Reliability Required by Functional Safety

ISO26262

Absence of unreasonable risk due to hazards caused by malfunctioning behavior of E/E systems

Decompose to Safety and Reliability Requirement

Safety Requirement

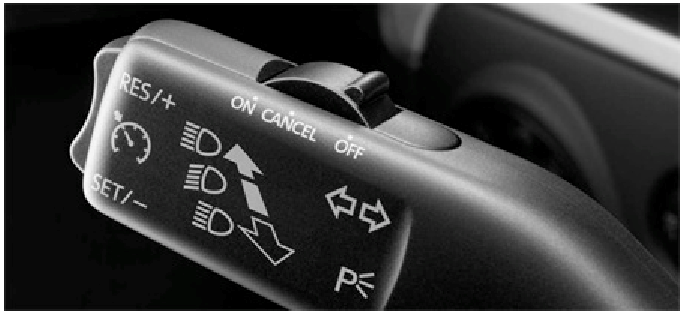
Reduce the risk so that hazards do not arise or result in an accident, even when system has a failure

Reliability Requirement

System can continue to operate correctly

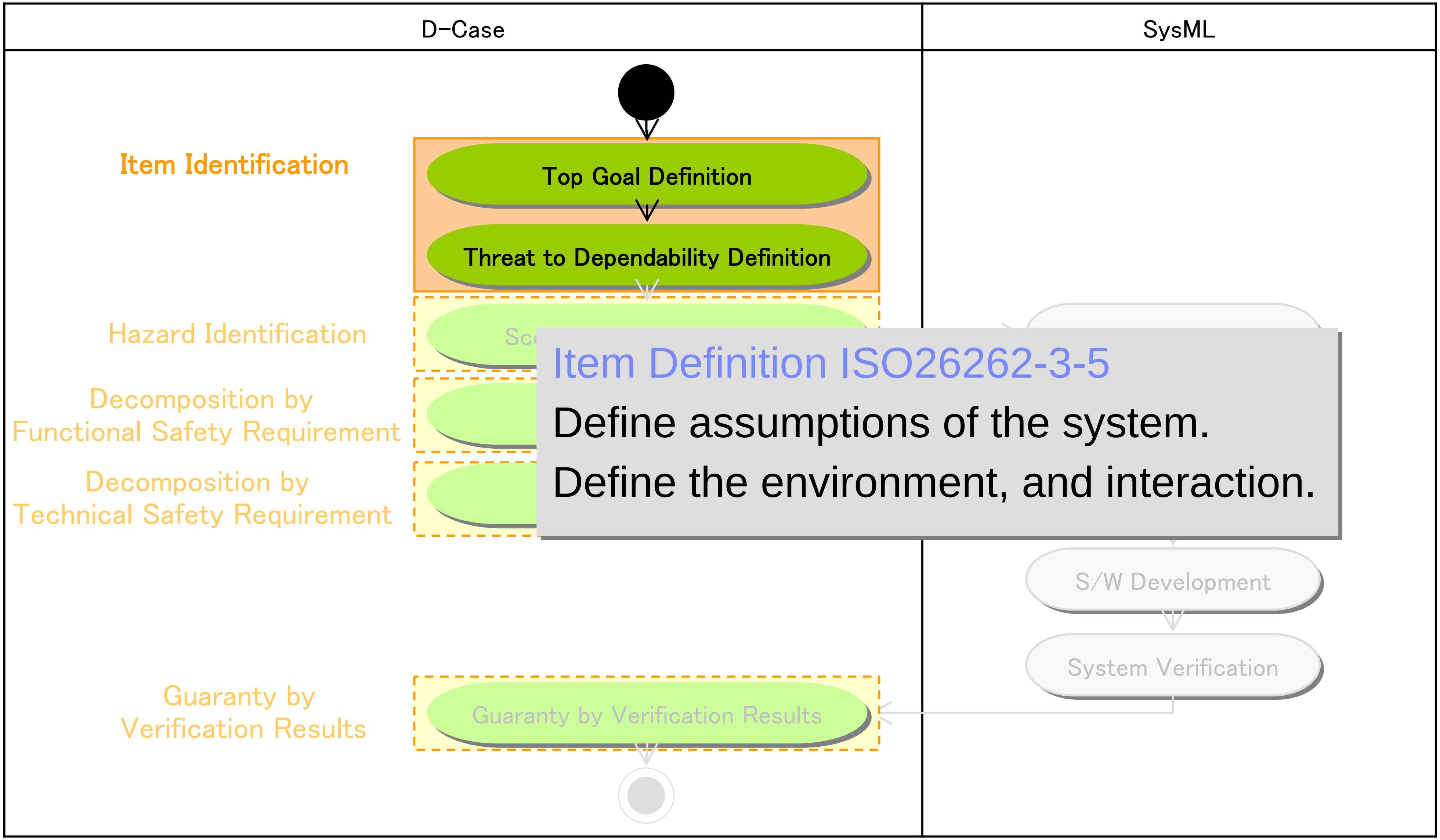
Demonstration – Target System

System	Cruise Control System (CC)
Precondition	Derivational development of system to apply functional safety requirements
Function	Speed control set by driver
Defective	System failure or calculation error
Hazard	Excessive acceleration from driver’ s intension
Functional Safety Requirement	Different acceleration from driver’ s intension is not carried out
Safety Requirement	Control which keeps acceleration in tolerance level can be performed even when an operation failure occurs by CC controller
Reliability Requirement	Different acceleration from driver’s intention is not carried out even when transmission route of speed sensor has a failure



Development Flow – Item Identification

Define Top Goal and Precondition about Safety, Extract Threats


D-Case
SysML
Item Identification
Top Goal Definition
Threat to Dependability Definition
Hazard Identification
Sc
Decomposition by Functional Safety Requirement
Decomposition by Technical Safety Requirement
Guaranty by Verification Results
Guaranty by Verification Results
S/W Development
System Verification

Item Definition ISO26262-3-5

Define assumptions of the system.

Define the environment, and interaction.

Top Goal Definition based on Safety

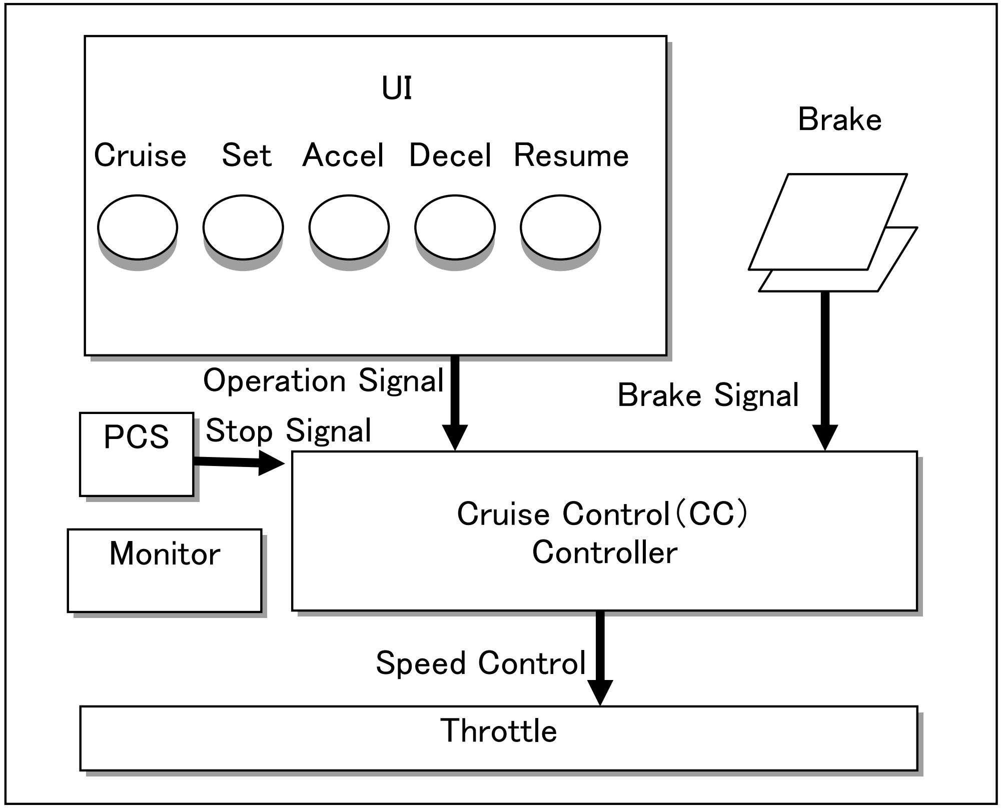
Clarify the precondition of the target system

Function

Cruise Control (CC) system controls speed set by driver.

Precondition

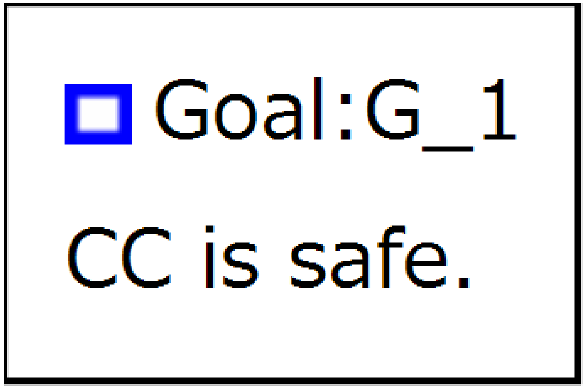
- [CY_01] Derivation development is adopted.
  Next system has functional safety
  based on ISO 26262.
- [CY_02] CC is safe.
- [CY_11] CC has 5 buttons on UI:
  Cruise, Set, Accel, Decel, and Resume.
- [CY_12] Driver controls CC via UI and brake pedal.
- [CY_13] Driver can always set CC in driving the car.
- [CY_21] OS is xx OS.



Top Goal Definition

Focusing on the safety required to ISO26262, define the top goal as “<system> is safe”.

Top Goal of D-Case



Goal:G_1

CC is safe.

Top Goal：CC is safe

Goal in D-Case

<System> is <Safe>.

（index）

D-Case **SysML** **ISO 26262**

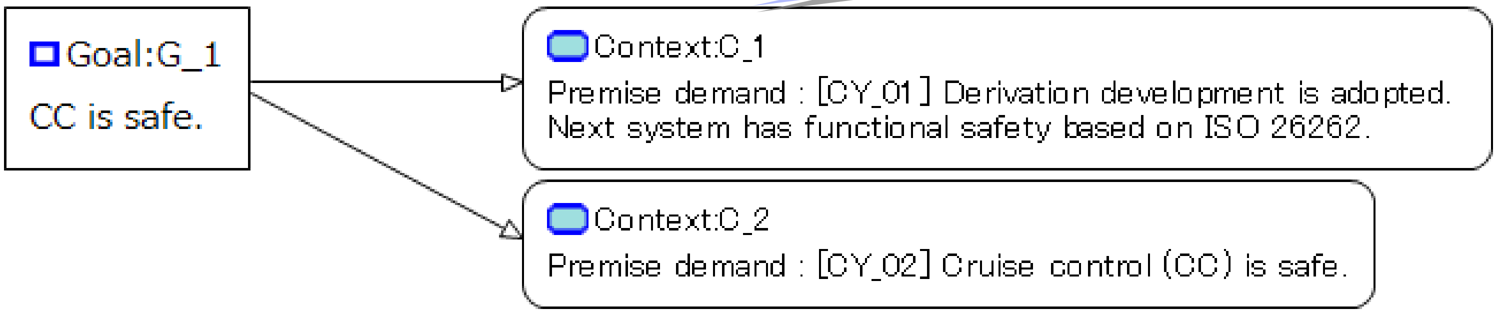
Association of Top Goal to Preconditions

Extract requirements to D-Case context to achieve the top goal

Precondition

·[CY_01] Derivation development is adopted.
Next system has functional safety based on ISO 26262.
·[CY_02] CC is safe.

Top Goal


Associate preconditions as context

Goal:G_1
CC is safe.

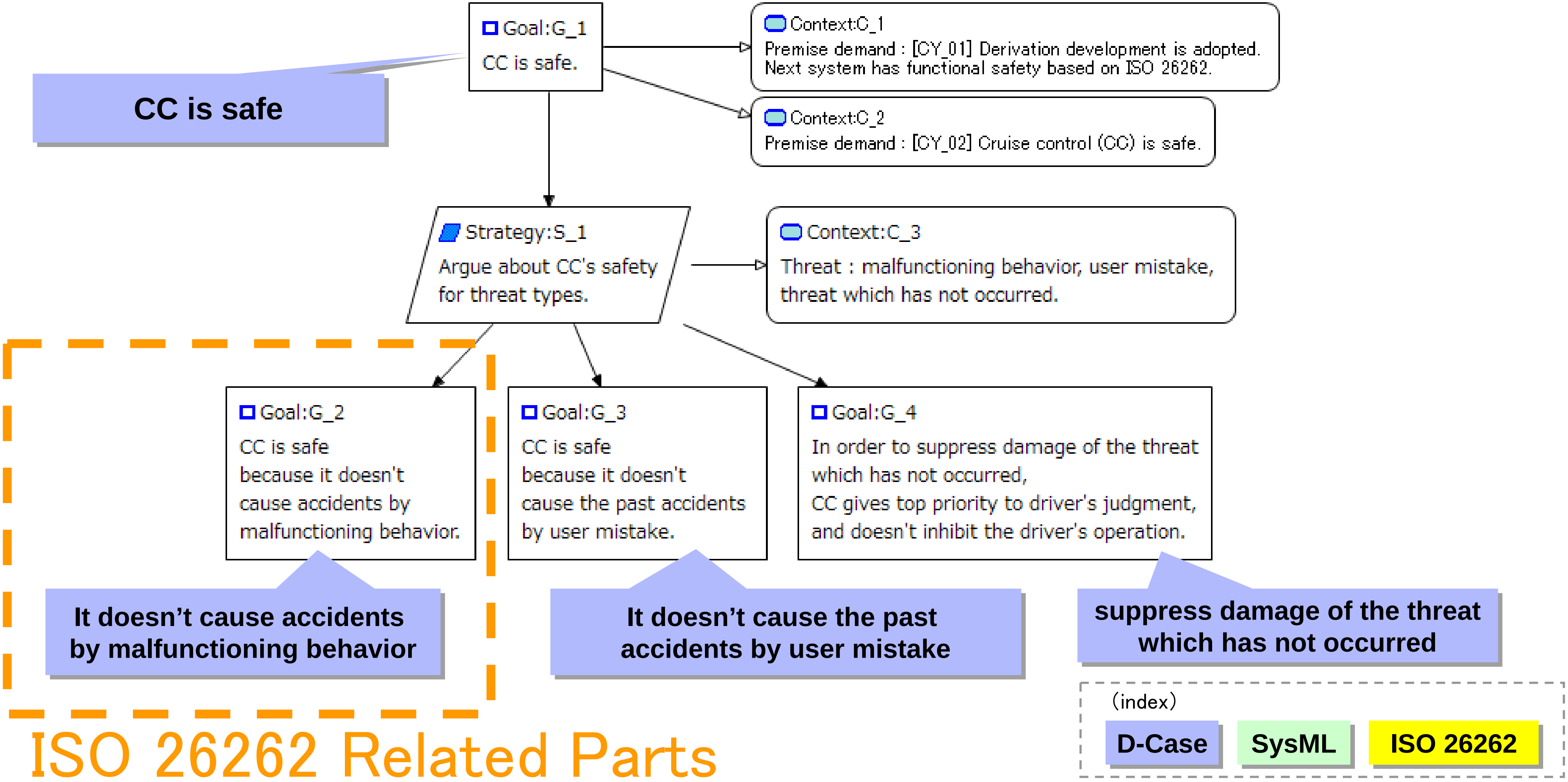
Context:C_1
Premise demand : [CY_01] Derivation development is adopted.
Next system has functional safety based on ISO 26262.

Context:C_2
Premise demand : [CY_02] Cruise control (CC) is safe.

(index)
D-Case SysML ISO 26262

Categorization of Threats Inhibiting Safety

Decompose top goal based on functional safety for the target system



Goal:G_1
CC is safe.

Context:C_1
Premise demand : [CY_01] Derivation development is adopted. Next system has functional safety based on ISO 26262.

Context:C_2
Premise demand : [CY_02] Cruise control (CC) is safe.

Strategy:S_1
Argue about CC's safety for threat types.

Context:C_3
Threat : malfunctioning behavior, user mistake, threat which has not occurred.

Goal:G_2
CC is safe because it doesn't cause accidents by malfunctioning behavior.

Goal:G_3
CC is safe because it doesn't cause the past accidents by user mistake.

Goal:G_4
In order to suppress damage of the threat which has not occurred, CC gives top priority to driver's judgment, and doesn't inhibit the driver's operation.

CC is safe

It doesn't cause accidents by malfunctioning behavior

It doesn't cause the past accidents by user mistake

suppress damage of the threat which has not occurred

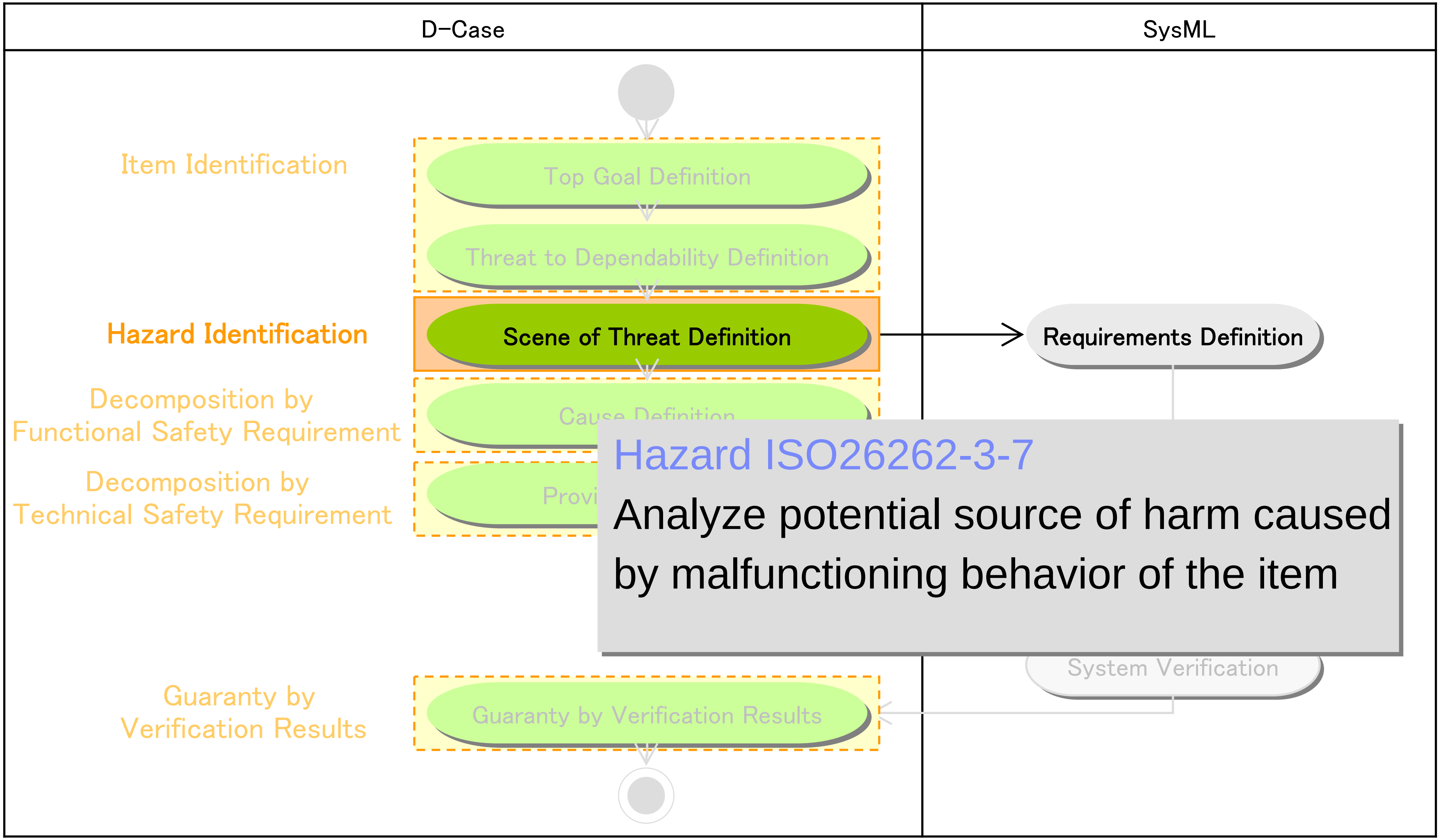
ISO 26262 Related Parts

（index）

D-Case	SysML	ISO 26262

Development Flow – Hazard Identification

Extract All the Scenes Inhibiting Safety



Hazard ISO26262-3-7

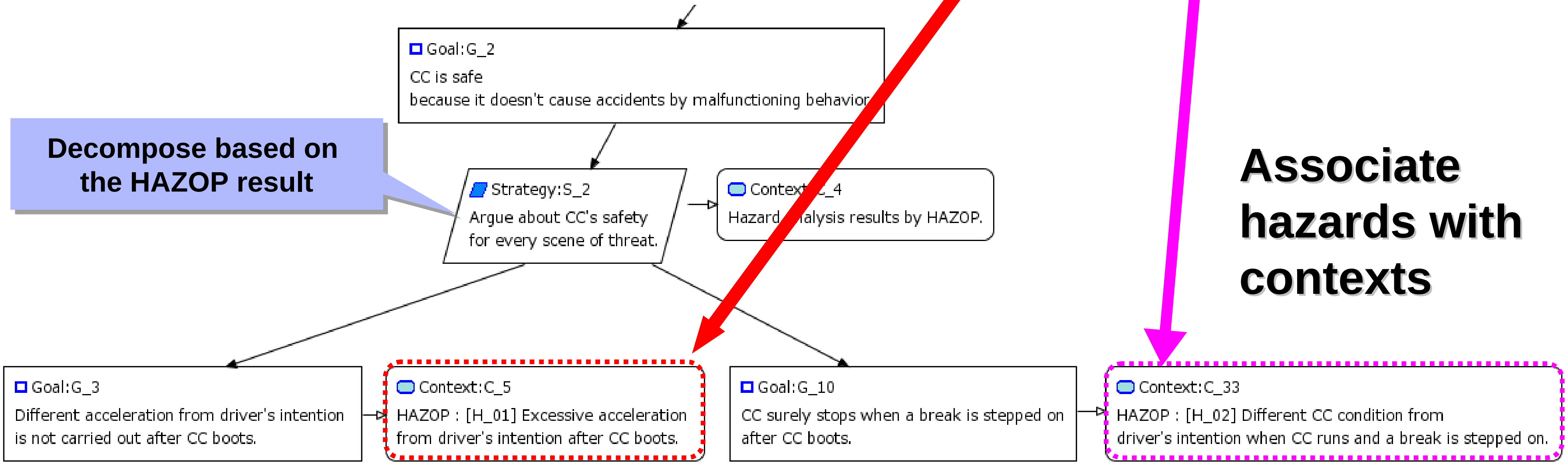
Analyze potential source of harm caused by malfunctioning behavior of the item

Clarification of Scene of Threat based on HAZOP

Decompose the goal by scenes of hazards derived from HAZOP

Clarify scenes of hazards by HAZOP

ID	Output	Guide Word	Scene	Hazard
H_01	CC Controller	More	After CC boots	Excessive acceleration from driver’s intention
H_02	CC Controller	No or not	Break is stepped on after CC boots	Different CC condition from driver’s intention



Decompose based on the HAZOP result

Goal:G_2
CC is safe
because it doesn't cause accidents by malfunctioning behavior

Strategy:S_2
Argue about CC's safety
for every scene of threat.

Context:C_4
Hazard analysis results by HAZOP.

Goal:G_3
Different acceleration from driver's intention is not carried out after CC boots.

Context:C_5
HAZOP : [H_01] Excessive acceleration from driver's intention after CC boots.

Goal:G_10
CC surely stops when a break is stepped on after CC boots.

Context:C_33
HAZOP : [H_02] Different CC condition from driver's intention when CC runs and a break is stepped on.

Associate hazards with contexts

(index)

D-Case **SysML** **ISO 26262**

Requirements Definition – Update Use Case Diagram and Verification Condition

Reflect users and system operations to use case diagram and verification conditions to test cases

D-Case

□ Goal:G_3

Different acceleration from driver's intention is not carried out after CC boots.

Context:C_5

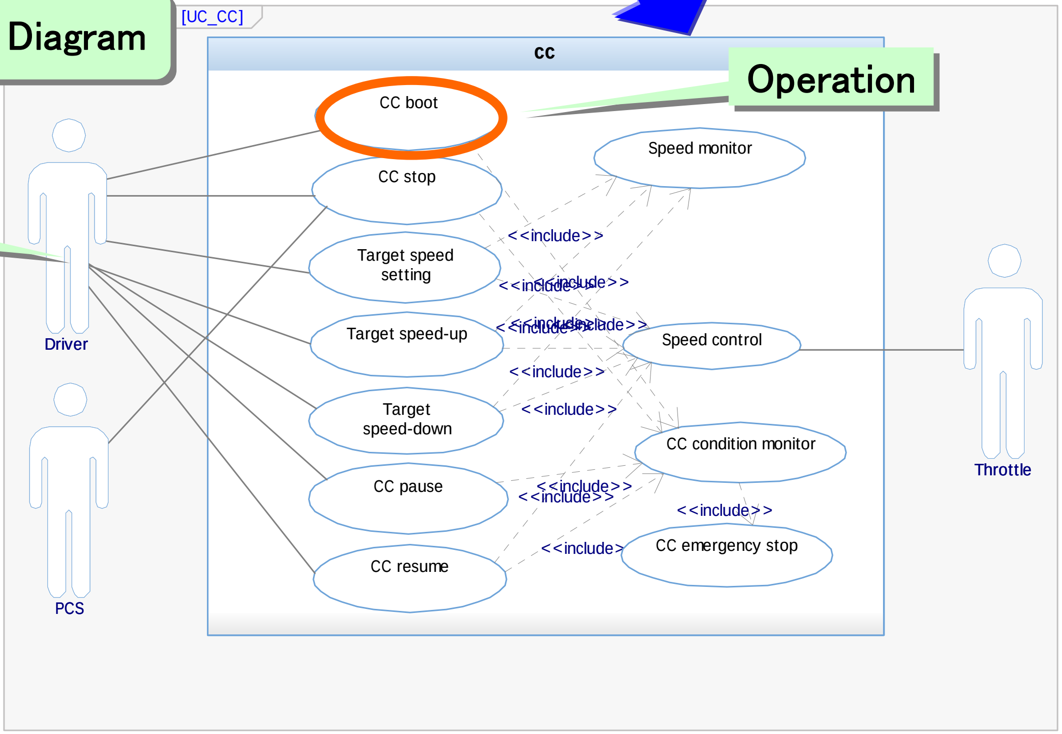
HAZOP : [H_01] Excessive acceleration from driver's intention after CC boots.

Update use case diagram by referring description about function

Extract verification conditions

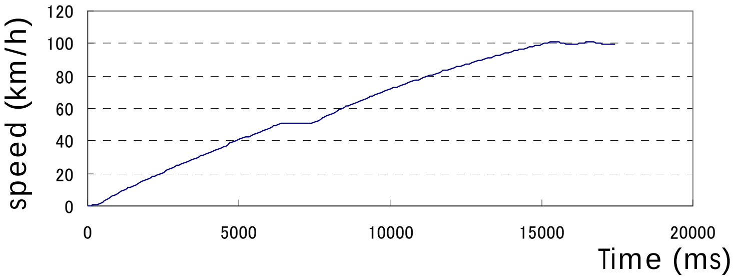
Use Case Diagram

User



Operation

Verification Condition



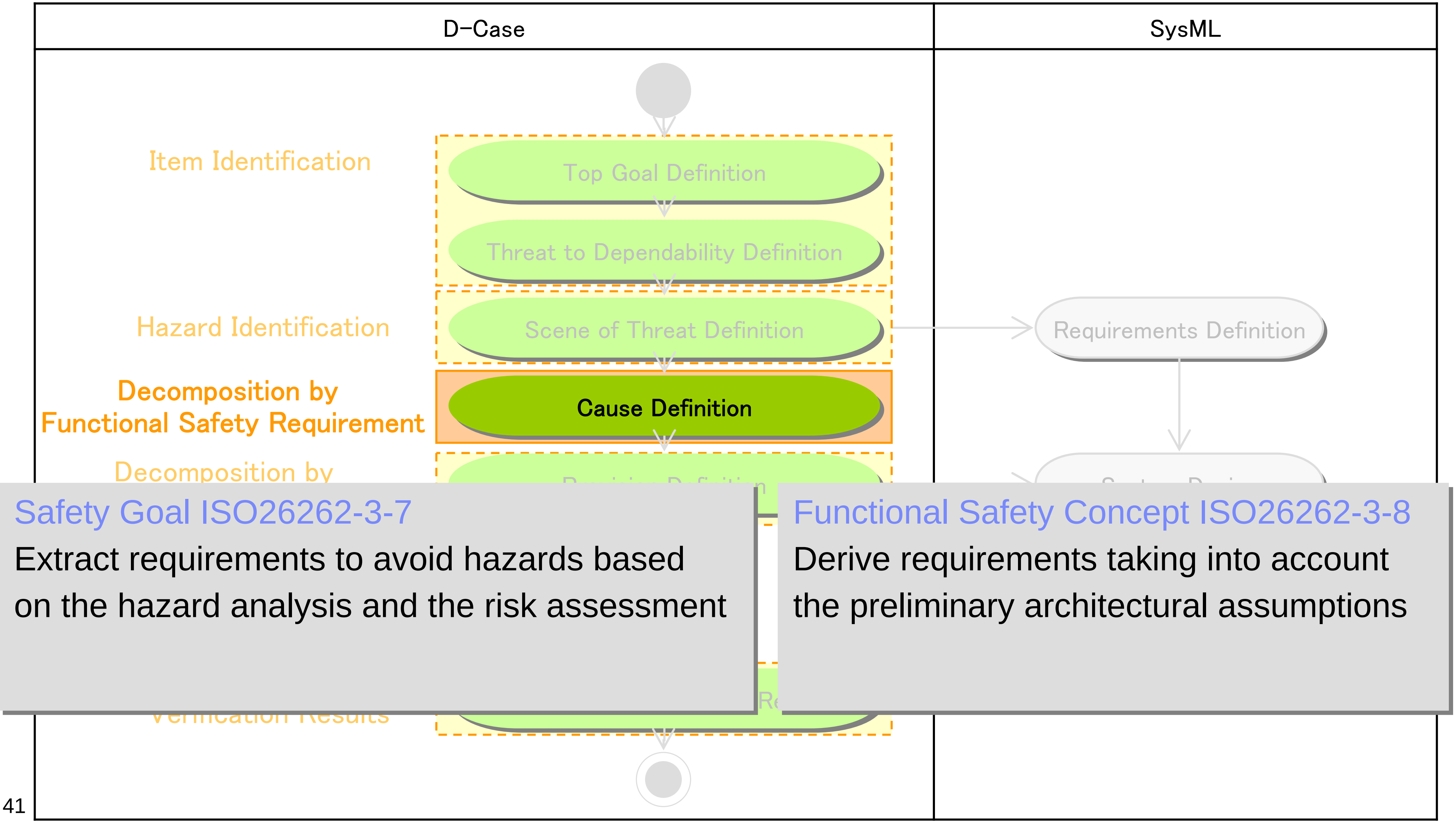
After CC boots, acceleration should change as a driver intends.

(index)

D-Case	SysML	ISO 26262

Development Flow – Decomposition by Functional Safety Requirement

Analyze Cause for Provision



Safety Goal ISO26262-3-7

Extract requirements to avoid hazards based on the hazard analysis and the risk assessment

Functional Safety Concept ISO26262-3-8

Derive requirements taking into account the preliminary architectural assumptions

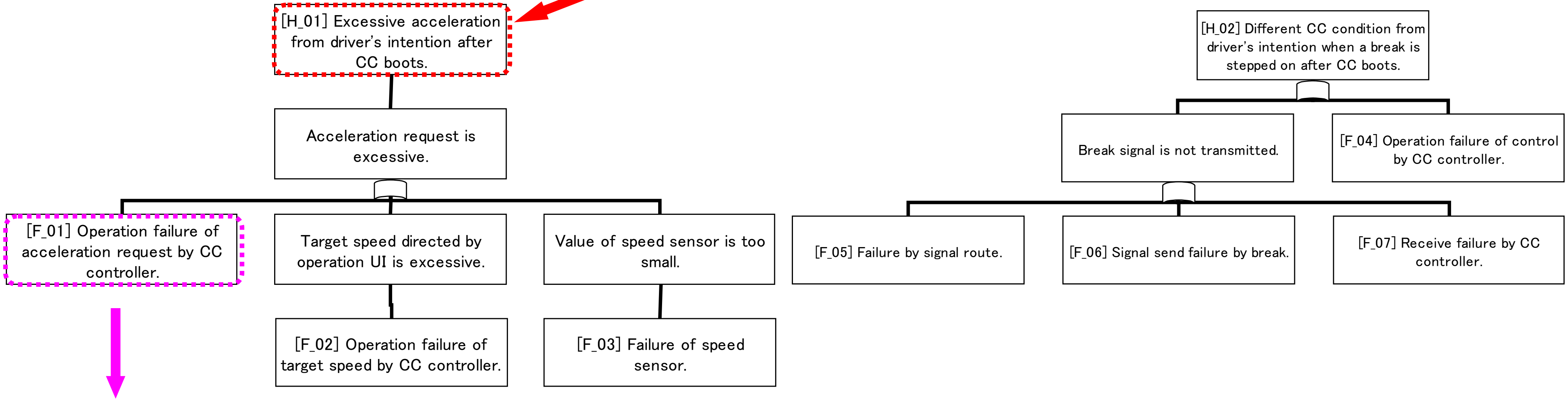
Clarification of Cause of Threat based on FTA

Analyze causes of hazards by FTA

Hazards extracted by HAZOP

ID	Output	Guide Word	Scene	Hazard
H_01	CC Controller	More	After CC boots	Excessive acceleration from driver’s intention
H_02	CC Controller	No or not	Break is stepped on after CC boots	Different CC condition from driver’s intention

Analyze causes of hazards by FTA



[H_01] Excessive acceleration from driver’s intention after CC boots.

Acceleration request is excessive.

[F_01] Operation failure of acceleration request by CC controller.

Target speed directed by operation UI is excessive.

Value of speed sensor is too small.

[F_02] Operation failure of target speed by CC controller.

[F_03] Failure of speed sensor.

[H_02] Different CC condition from driver’s intention when a break is stepped on after CC boots.

Break signal is not transmitted.

[F_04] Operation failure of control by CC controller.

[F_05] Failure by signal route.

[F_06] Signal send failure by break.

[F_07] Receive failure by CC controller.

Extract functional safety requirements

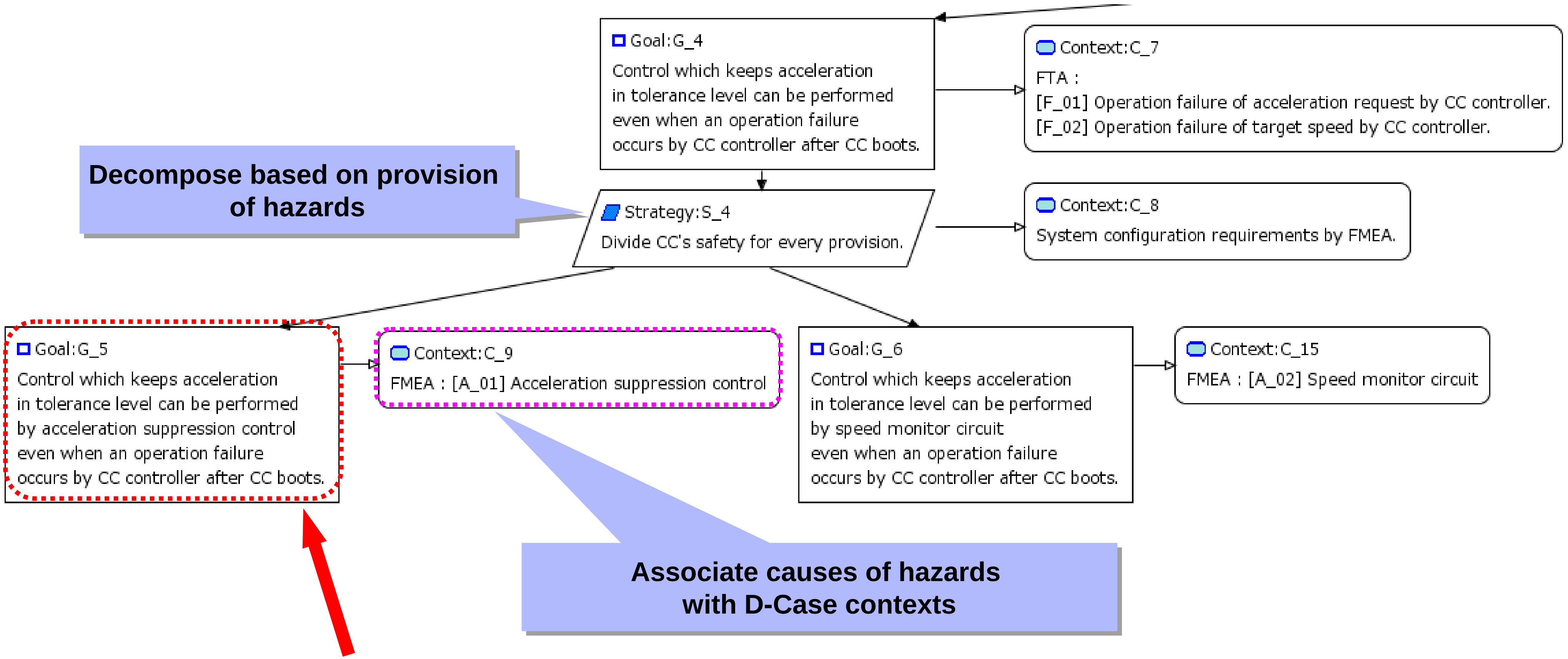
Control which keeps acceleration in tolerance level can be performed even when an operation failure occurs by CC controller.

（index）

D-Case SysML ISO 26262

Clarification of Cause of Threat based on FTA

Decompose the goal by causes of hazard based on FTA



Goal:G_4
Control which keeps acceleration in tolerance level can be performed even when an operation failure occurs by CC controller after CC boots.

Context:C_7
FTA :
[F_01] Operation failure of acceleration request by CC controller.
[F_02] Operation failure of target speed by CC controller.

Decompose based on provision of hazards

Strategy:S_4
Divide CC's safety for every provision.

Context:C_8
System configuration requirements by FMEA.

Goal:G_5
Control which keeps acceleration in tolerance level can be performed by acceleration suppression control even when an operation failure occurs by CC controller after CC boots.

Context:C_9
FMEA : [A_01] Acceleration suppression control

Goal:G_6
Control which keeps acceleration in tolerance level can be performed by speed monitor circuit even when an operation failure occurs by CC controller after CC boots.

Context:C_15
FMEA : [A_02] Speed monitor circuit

Associate causes of hazards with D-Case contexts

Functional safety requirements

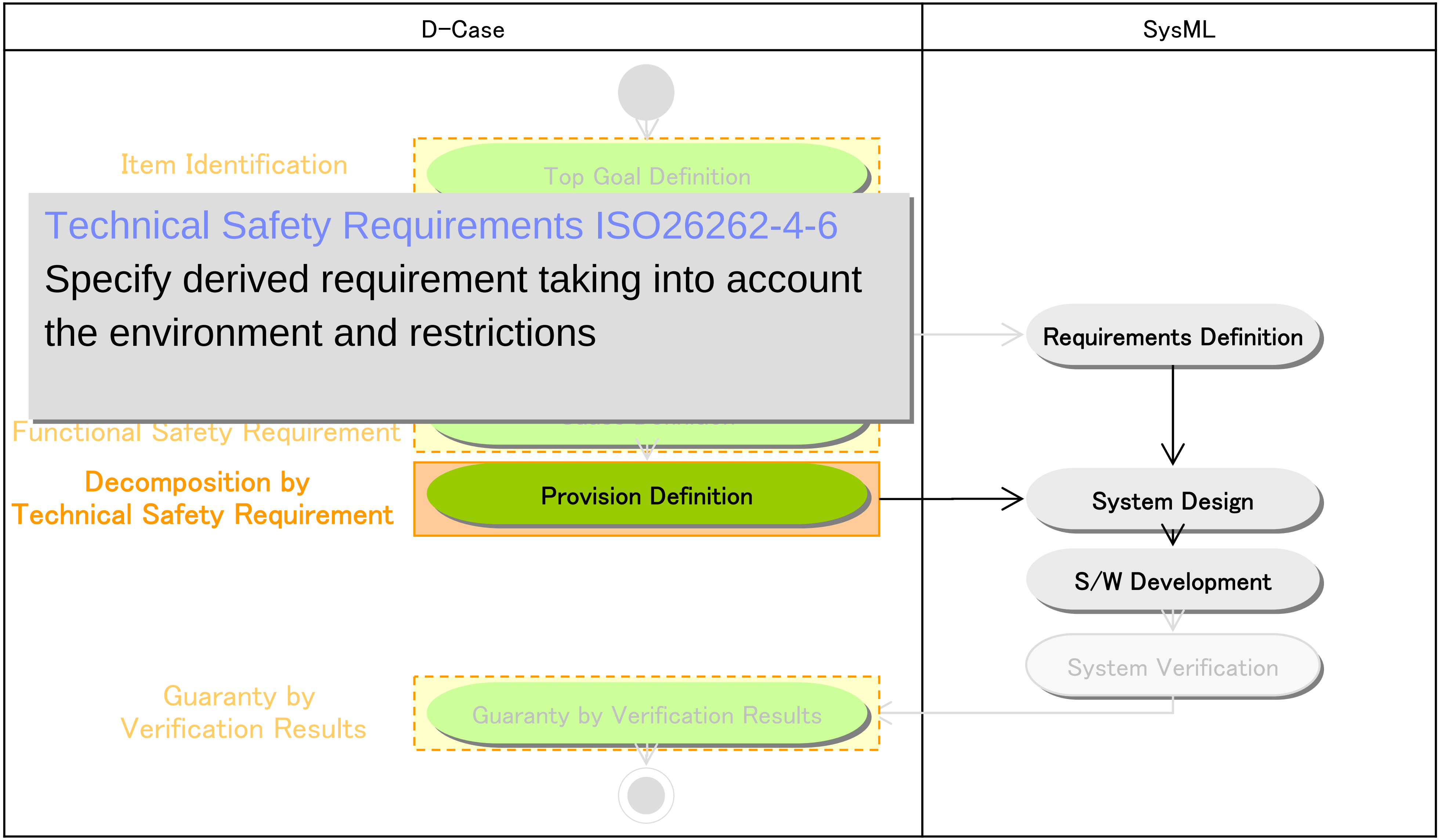
Control which keeps acceleration in tolerance level can be performed even when an operation failure occurs by CC controller.

（index）

D-Case | SysML | ISO 26262

Development Flow – Decomposition by Technical Safety Requirement

Refine System Demands based on Provisions for Causes



Technical Safety Requirements ISO26262-4-6

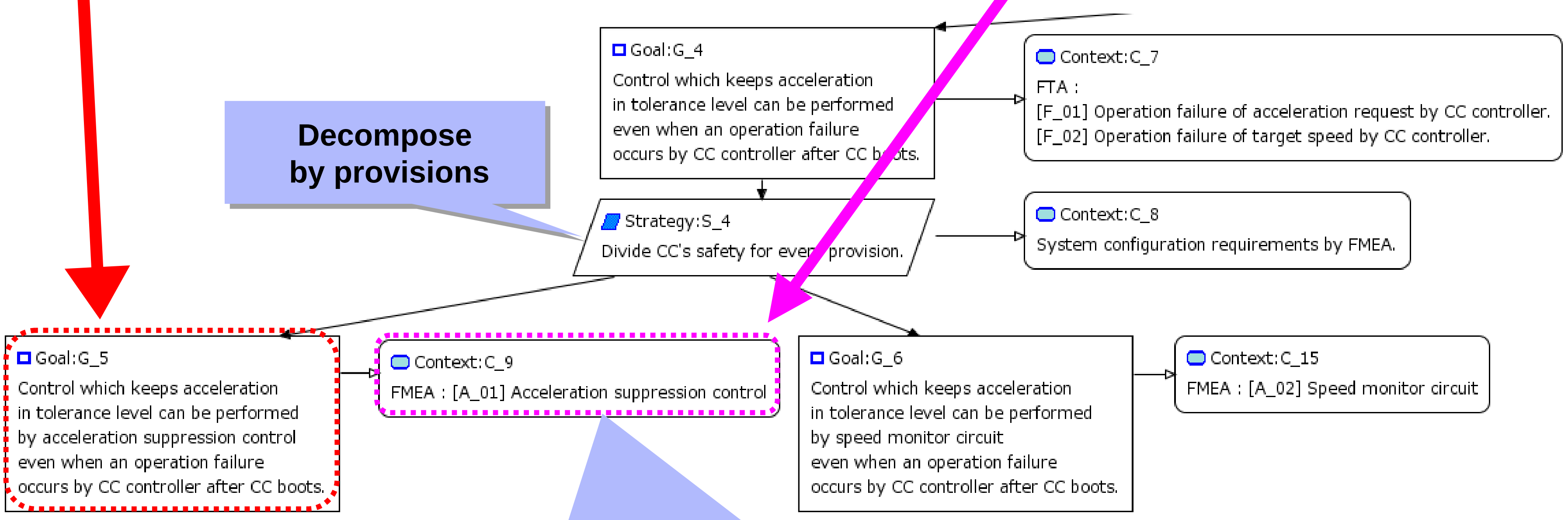
Specify derived requirement taking into account the environment and restrictions

Clarification of Provision based on FMEA

Clarify system demands just enough using provisions by FMEA

Investigate provisions by FMEA

ID	Component	Failure mode	Factor	Severity of influence	Frequency	Difficulty of detection	Risk priority	Provision	
								S/W	H/W
F_01	CC Controller	Operation failure (acceler	Program bug	6 (M)	1 (L)	5 (M)	30	[A_01] Acceleration suppression control	[A_02] Speed monitor circuit
F_02	CC Controller	Operation failure (target	Program bug	3 (L)	1 (L)	1 (L)	3		
F_03	Speed sensor	Abnormal value	Breakdown	9 (H)	5 (M)	1 (L)	45	[A_03] CC emergency stop	
F_04	CC Controller	Operation failure (control	Program bug	9 (H)	1 (L)	1 (L)	9		[A_04] CC condition monitor circuit
F_05	Transmission ou	Abnormal value	Breakdown	9 (H)	5 (M)	1 (L)	45		
F_06	Brak	Operation failure (send)	Breaking of wire	9 (H)	1 (L)	1 (L)	9		
F_07	CC Controller	Operation failure (receive	Breaking of wire	9 (H)	1 (L)	5 (M)	45		

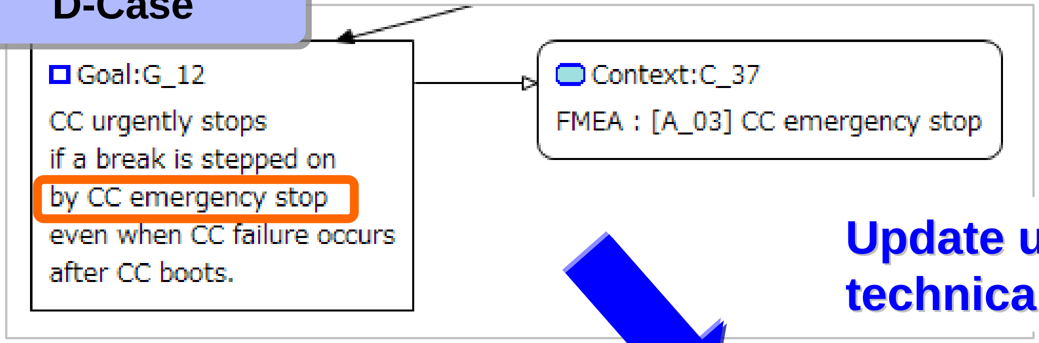

Goal:G_4
Control which keeps acceleration in tolerance level can be performed even when an operation failure occurs by CC controller after CC boots.
Context:C_7
FTA :
[F_01] Operation failure of acceleration request by CC controller.
[F_02] Operation failure of target speed by CC controller.
Decompose by provisions
Strategy:S_4
Divide CC's safety for every provision.
Context:C_8
System configuration requirements by FMEA.
Goal:G_5
Control which keeps acceleration in tolerance level can be performed by acceleration suppression control even when an operation failure occurs by CC controller after CC boots.
Context:C_9
FMEA : [A_01] Acceleration suppression control
Goal:G_6
Control which keeps acceleration in tolerance level can be performed by speed monitor circuit even when an operation failure occurs by CC controller after CC boots.
Context:C_15
FMEA : [A_02] Speed monitor circuit

Associate the provision with context and clarify correspondence relation with demands

Requirements Definition – Update Use Case Diagram

Basing on system demands, update users and system operations to use case diagram

D-Case



□ Goal:G_12

CC urgently stops
if a break is stepped on
by CC emergency stop
even when CC failure occurs
after CC boots.

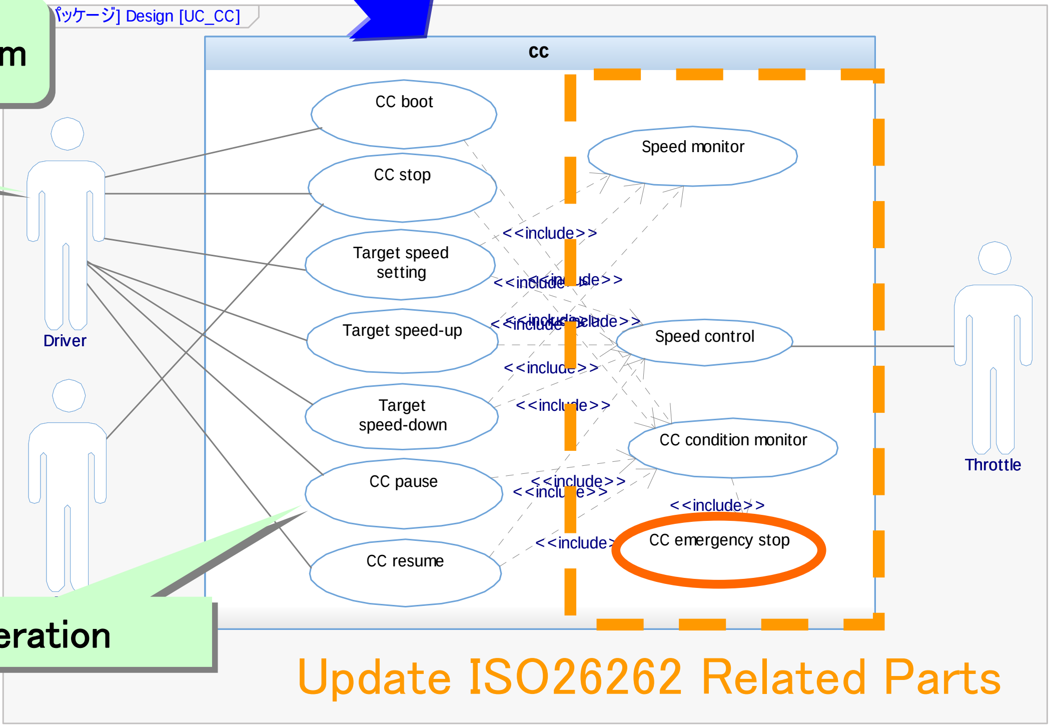
Context:C_37

FMEA : [A_03] CC emergency stop

Update use case diagram by referring a technical safety requirement about function

Use Case Diagram

User



Operation

Update ISO26262 Related Parts

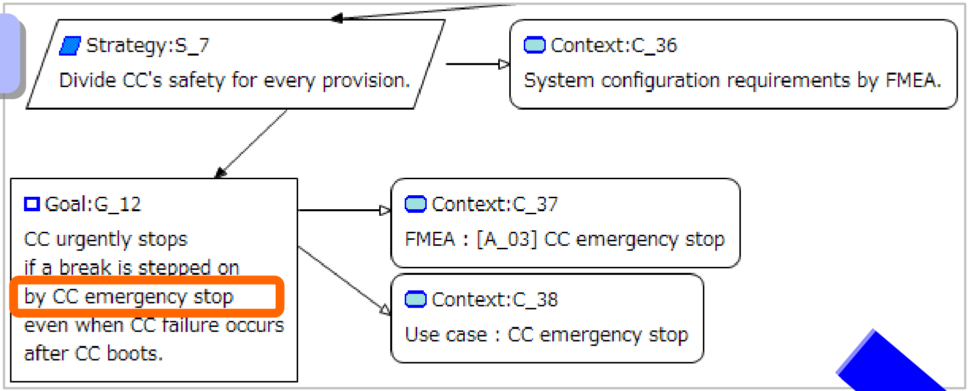
(index)

D-Case **SysML** **ISO 26262**

Requirements Definition – Update Requirement Diagram

Basing on system demands, update requirements to requirement diagram

D-Case



Strategy:S_7
Divide CC's safety for every provision.

Context:C_36
System configuration requirements by FMEA.

Goal:G_12
CC urgently stops if a break is stepped on by CC emergency stop even when CC failure occurs after CC boots.

Context:C_37
FMEA : [A_03] CC emergency stop

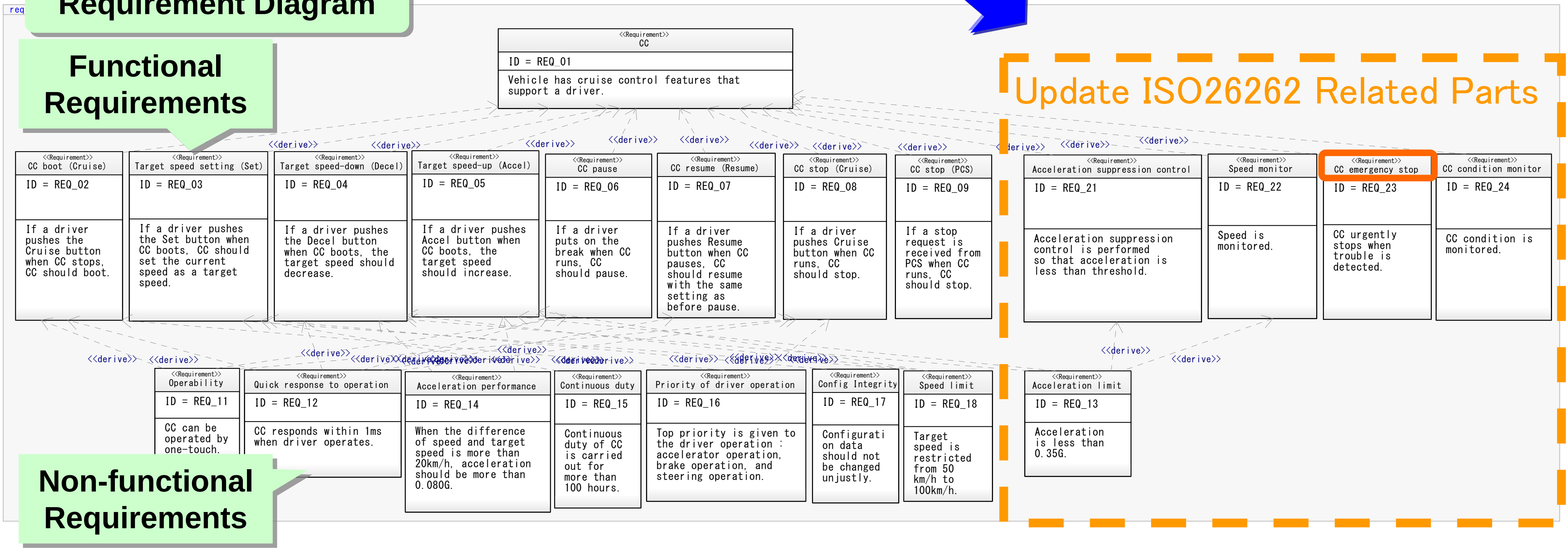
Context:C_38
Use case : CC emergency stop

Update requirement diagram related to functional safety

Requirement Diagram

Functional Requirements

Non-functional Requirements



<<Requirement>>
CC
ID = REQ_01
Vehicle has cruise control features that support a driver.

<<Requirement>>
CC boot (Cruise)
ID = REQ_02
If a driver pushes the Cruise button when CC stops, CC should boot.

<<Requirement>>
Target speed setting (Set)
ID = REQ_03
If a driver pushes the Set button when CC boots, CC should set the current speed as a target speed.

<<Requirement>>
Target speed-down (Decel)
ID = REQ_04
If a driver pushes the Decel button when CC boots, the target speed should decrease.

<<Requirement>>
Target speed-up (Accel)
ID = REQ_05
If a driver pushes Accel button when CC boots, the target speed should increase.

<<Requirement>>
CC pause
ID = REQ_06
If a driver puts on the break when CC runs, CC should pause.

<<Requirement>>
CC resume (Resume)
ID = REQ_07
If a driver pushes Resume button when CC pauses, CC should resume with the same setting as before pause.

<<Requirement>>
CC stop (Cruise)
ID = REQ_08
If a driver pushes Cruise button when CC runs, CC should stop.

<<Requirement>>
CC stop (PCS)
ID = REQ_09
If a stop request is received from PCS when CC runs, CC should stop.

Update ISO26262 Related Parts

<<Requirement>>
Acceleration suppression control
ID = REQ_21
Acceleration suppression control is performed so that acceleration is less than threshold.

<<Requirement>>
Speed monitor
ID = REQ_22
Speed is monitored.

<<Requirement>>
CC emergency stop
ID = REQ_23
CC urgently stops when trouble is detected.

<<Requirement>>
CC condition monitor
ID = REQ_24
CC condition is monitored.

<<Requirement>>
Operability
ID = REQ_11
CC can be operated by one-touch.

<<Requirement>>
Quick response to operation
ID = REQ_12
CC responds within 1ms when driver operates.

<<Requirement>>
Acceleration performance
ID = REQ_14
When the difference of speed and target speed is more than 20km/h, acceleration should be more than 0.080G.

<<Requirement>>
Continuous duty
ID = REQ_15
Continuous duty of CC is carried out for more than 100 hours.

<<Requirement>>
Priority of driver operation
ID = REQ_16
Top priority is given to the driver operation : accelerator operation, brake operation, and steering operation.

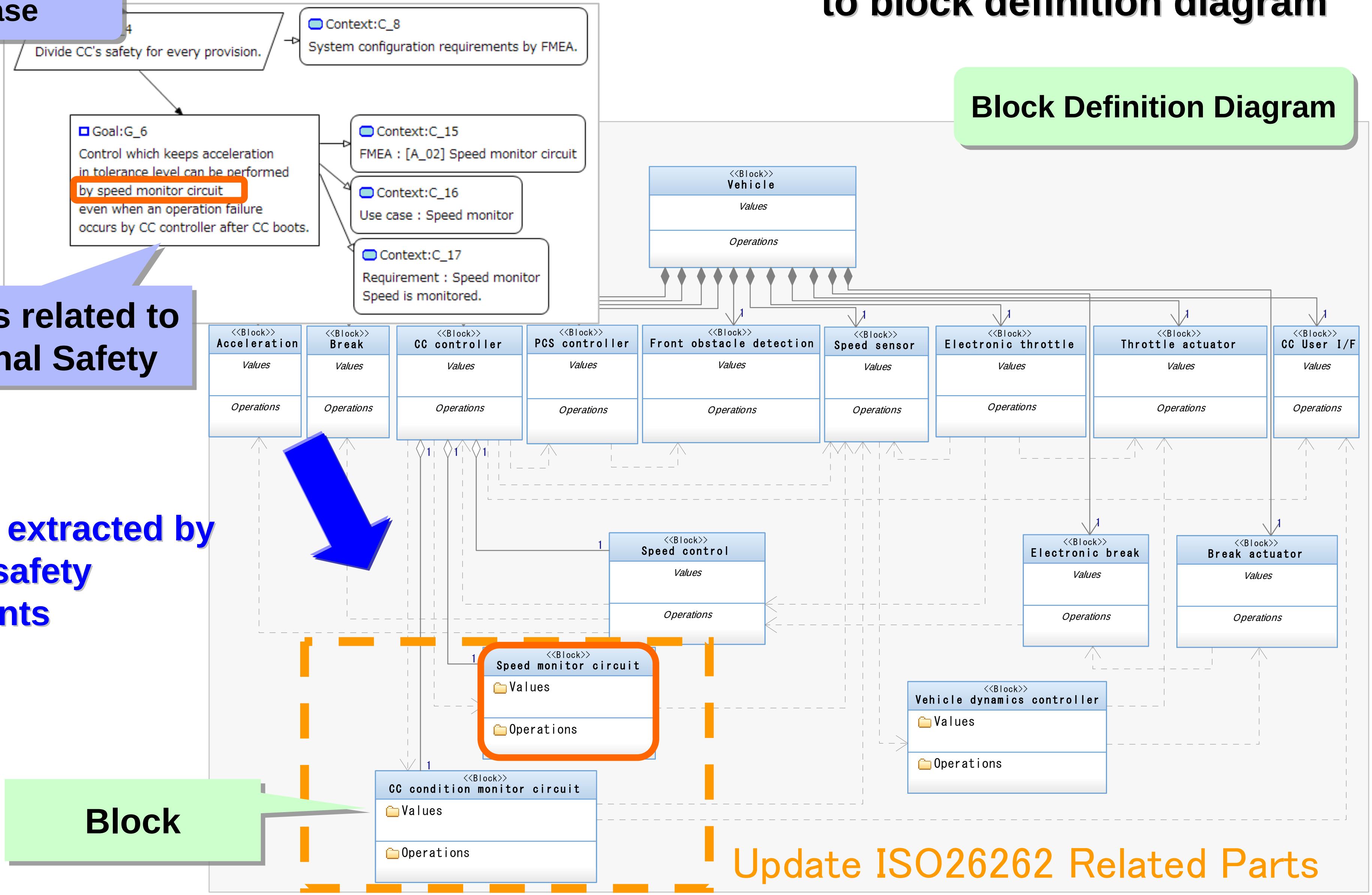
<<Requirement>>
Config Integrity
ID = REQ_17
Configuration data should not be changed unjustly.

<<Requirement>>
Speed limit
ID = REQ_18
Target speed is restricted from 50 km/h to 100km/h.

<<Requirement>>
Acceleration limit
ID = REQ_13
Acceleration is less than 0.35G.

System Design – Update Block Definition Diagram

Basing on system demands, update system architecture to block definition diagram



D-Case

4
Divide CC's safety for every provision.

Context:C_8
System configuration requirements by FMEA.

Goal:G_6
Control which keeps acceleration in tolerance level can be performed by speed monitor circuit even when an operation failure occurs by CC controller after CC boots.

Context:C_15
FMEA : [A_02] Speed monitor circuit

Context:C_16
Use case : Speed monitor

Context:C_17
Requirement : Speed monitor
Speed is monitored.

Functions related to Functional Safety

Block Definition Diagram

Add block extracted by technical safety requirements

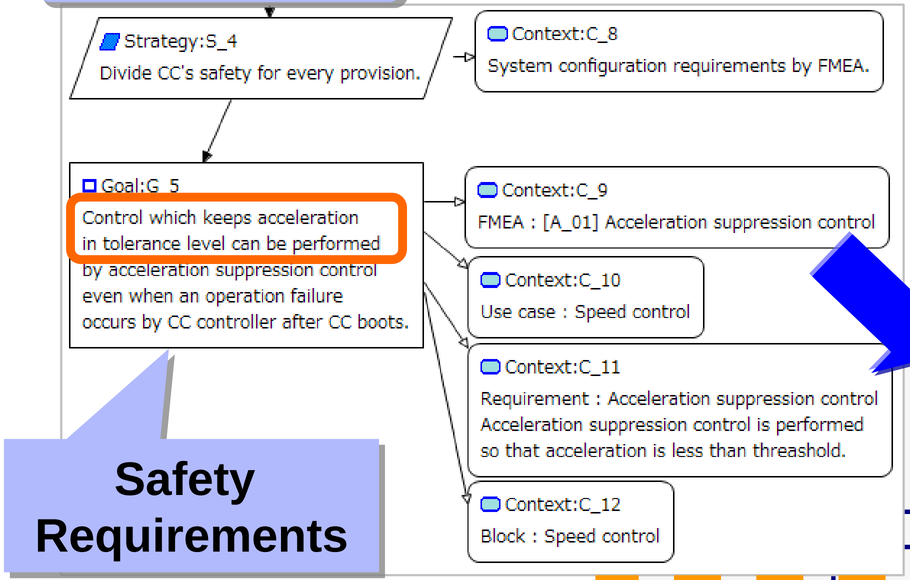
Block

Update ISO26262 Related Parts

System Design – Update Parametric Diagram

Update restrictions, constant value and formula to parametric diagram

D-Case



Strategy:S_4
Divide CC's safety for every provision.

Context:C_8
System configuration requirements by FMEA.

Goal:G_5
Control which keeps acceleration in tolerance level can be performed by acceleration suppression control even when an operation failure occurs by CC controller after CC boots.

Context:C_9
FMEA : [A_01] Acceleration suppression control

Context:C_10
Use case : Speed control

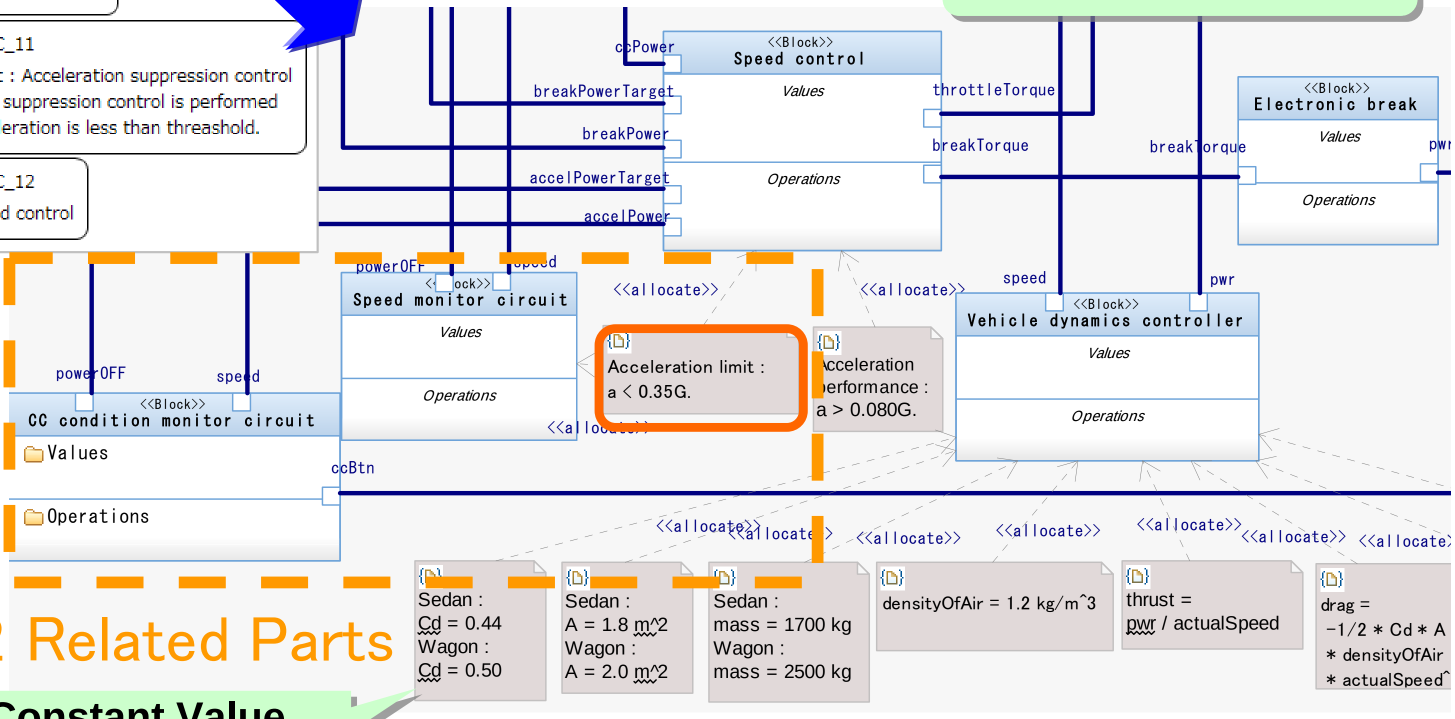
Context:C_11
Requirement : Acceleration suppression control
Acceleration suppression control is performed so that acceleration is less than threashold.

Context:C_12
Block : Speed control

Safety Requirements

Update restriction by safety requirement

Parametric Diagram



Acceleration limit :
a < 0.35G.

Acceleration performance :
a > 0.080G.

Sedan :
Cd = 0.44
Wagon :
Cd = 0.50

Sedan :
A = 1.8 m^2
Wagon :
A = 2.0 m^2

Sedan :
mass = 1700 kg
Wagon :
mass = 2500 kg

densityOfAir = 1.2 kg/m^3

thrust =
pwr / actualSpeed

drag =
−1/2 * Cd * A
* densityOfAir
* actualSpeed^

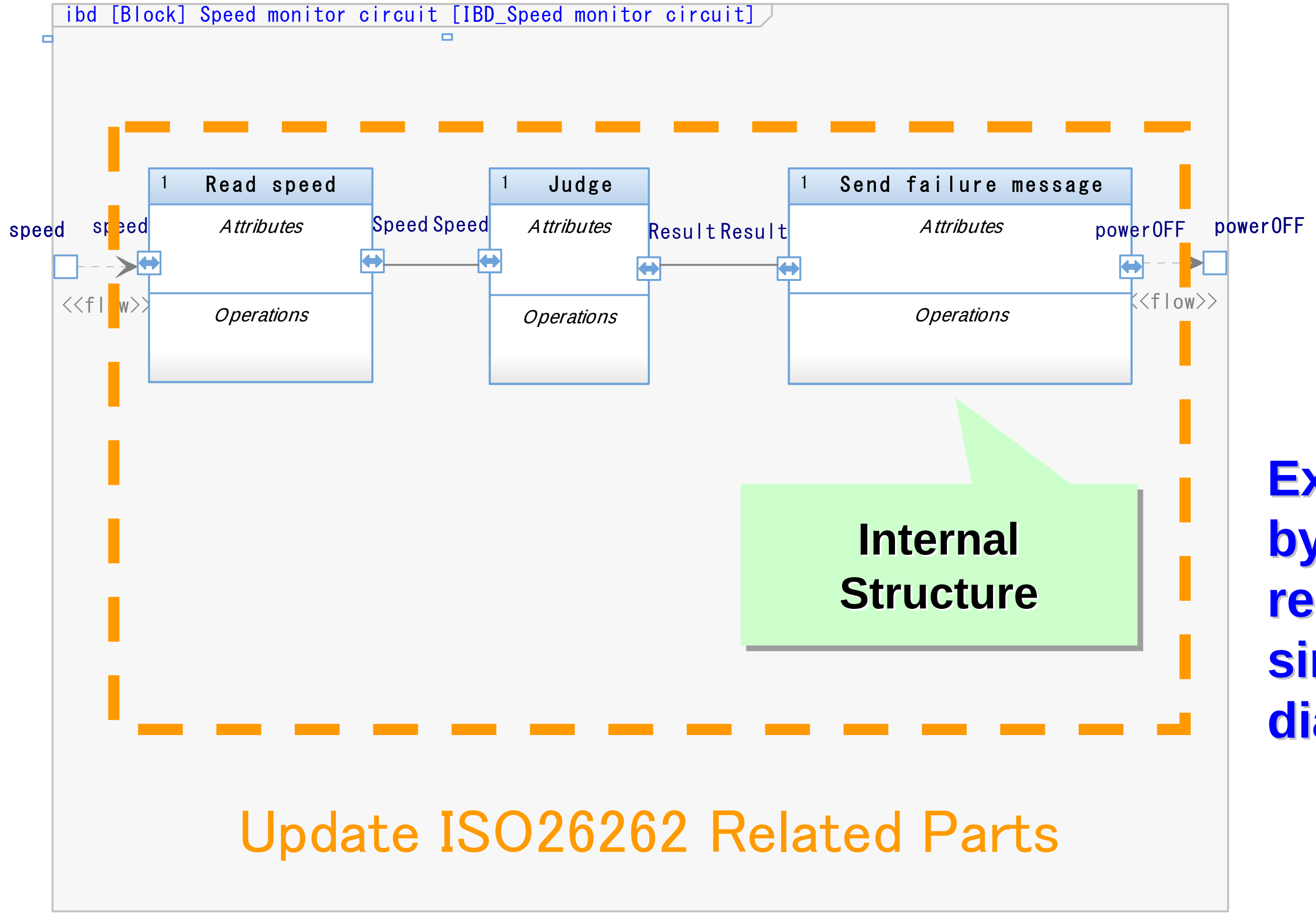
Update ISO26262 Related Parts

Restrictions, Constant Value, Formula, Intentions

System Design – Update Internal Block Diagram

Update internal structure of blocks to internal block diagram

Internal Block Diagram



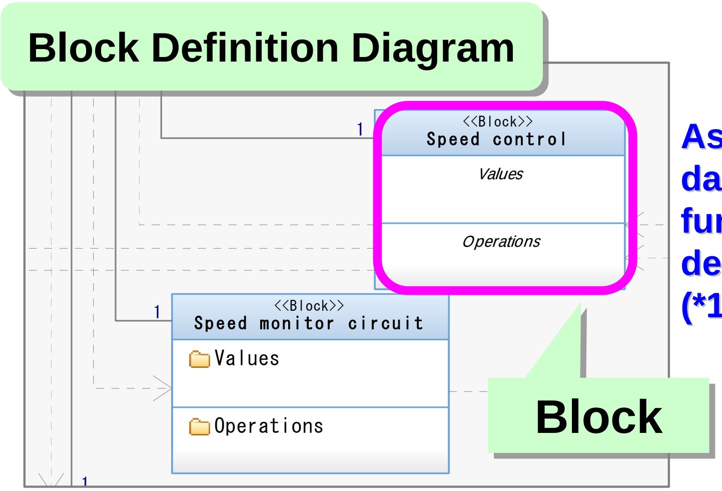
Internal Structure

Update ISO26262 Related Parts

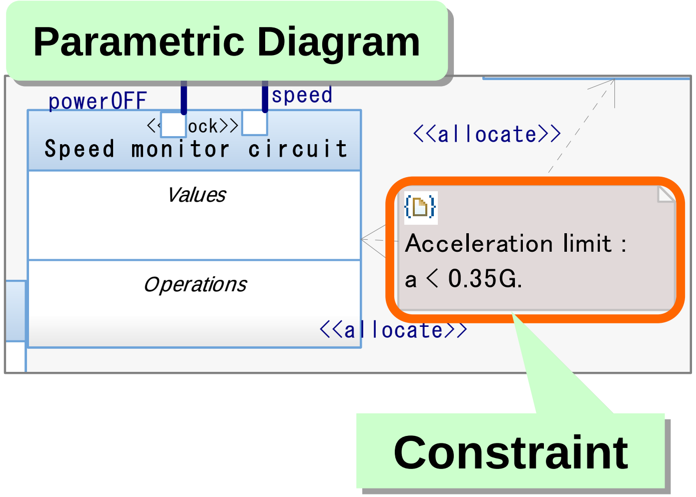
Extract system element by technical safety requirements from D-Case, similar to block definition diagram

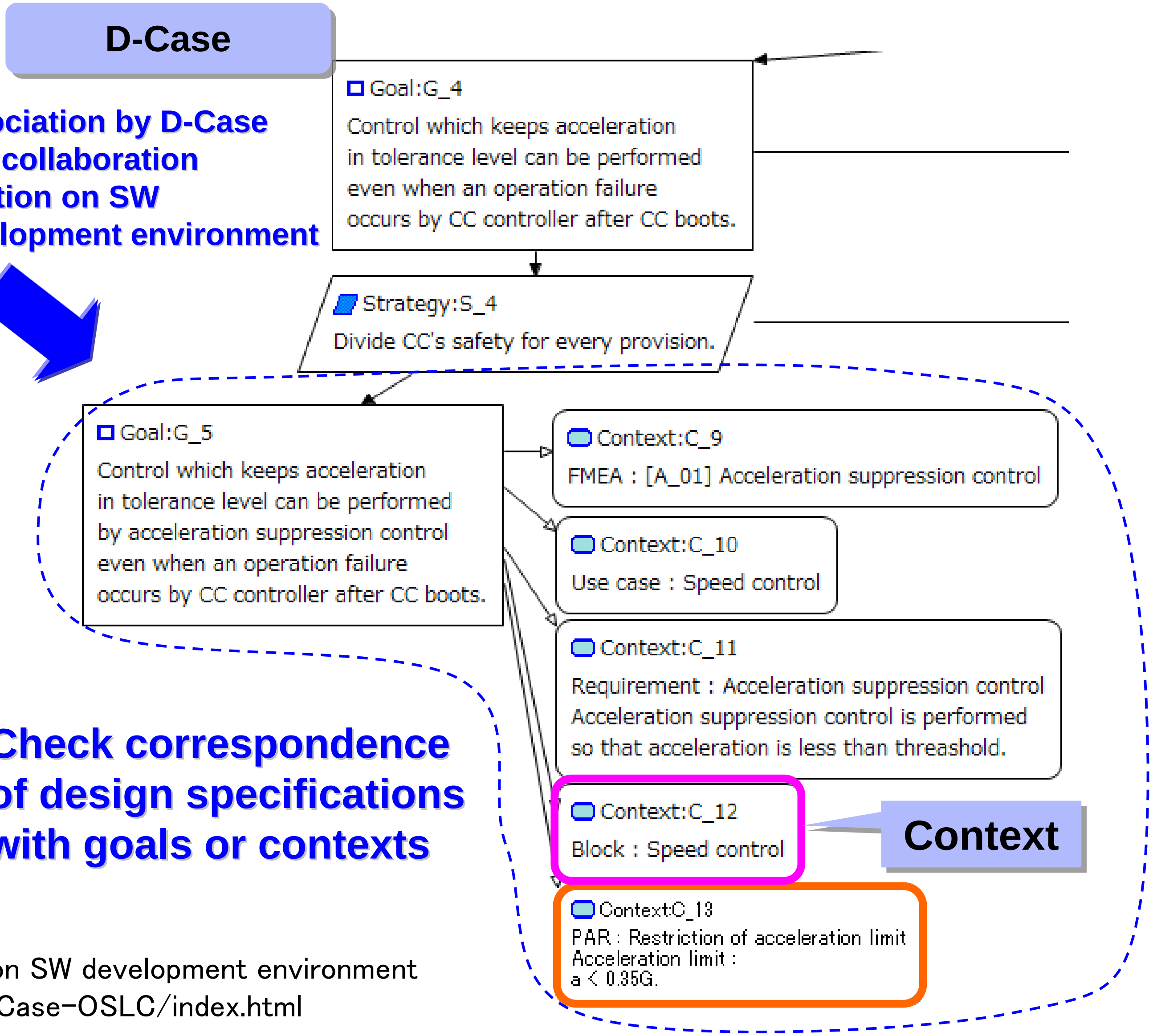
Correspondence Check between System Demands and Design Specifications

Check correspondence of design specifications by associating blocks of block definition diagram and restrictions of parametric diagram



Association by D-Case data collaboration function on SW development environment (*1)




D-Case
Goal:G_4
Control which keeps acceleration in tolerance level can be performed even when an operation failure occurs by CC controller after CC boots.
Strategy:S_4
Divide CC's safety for every provision.
Goal:G_5
Control which keeps acceleration in tolerance level can be performed by acceleration suppression control even when an operation failure occurs by CC controller after CC boots.
Context:C_9
FMEA : [A_01] Acceleration suppression control
Context:C_10
Use case : Speed control
Context:C_11
Requirement : Acceleration suppression control Acceleration suppression control is performed so that acceleration is less than threashold.
Context:C_12
Block : Speed control
Context
Context:C_13
PAR : Restriction of acceleration limit
Acceleration limit :
a < 0.35G.

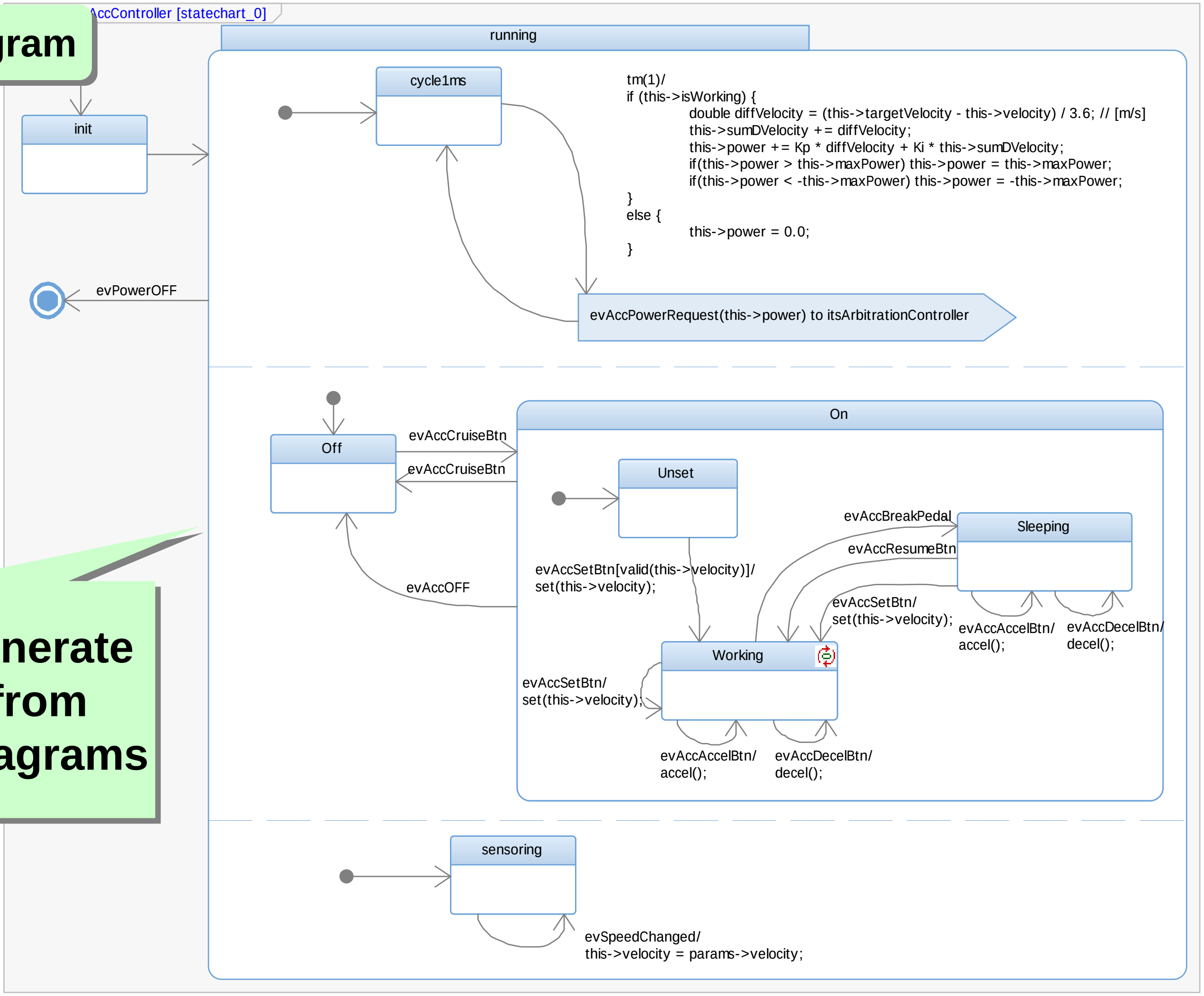
Check correspondence of design specifications with goals or contexts

(*1)D-Case data collaboration function on SW development environment
http://www.dependable-os.net/tech/D-Case-OSLC/index.html

S/W Development

Generate source codes from state machine diagrams defining dynamical behavior of the system

State Machine Diagram

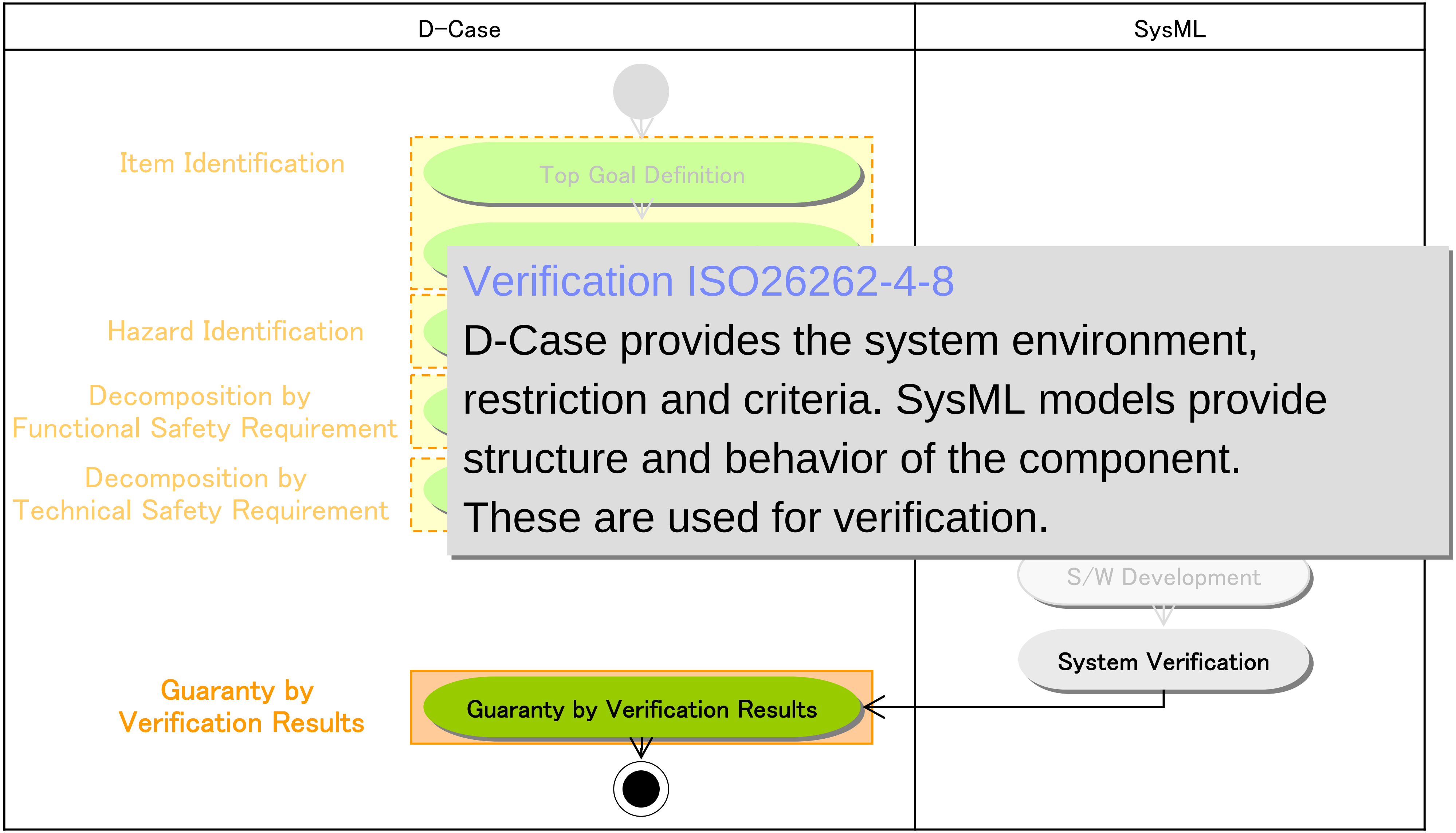


```
tm(1)/
if (this->isWorking) {
        double diffVelocity = (this->targetVelocity - this->velocity) / 3.6; // [m/s]
        this->sumDVelocity += diffVelocity;
        this->power += Kp * diffVelocity + Ki * this->sumDVelocity;
        if(this->power > this->maxPower) this->power = this->maxPower;
        if(this->power < -this->maxPower) this->power = -this->maxPower;
}
else {
        this->power = 0.0;
}
```

Automatically generate source codes from state machine diagrams

Development Flow – Guaranty by Verification Results

Clarify Satisfaction of System Demands



Verification ISO26262-4-8

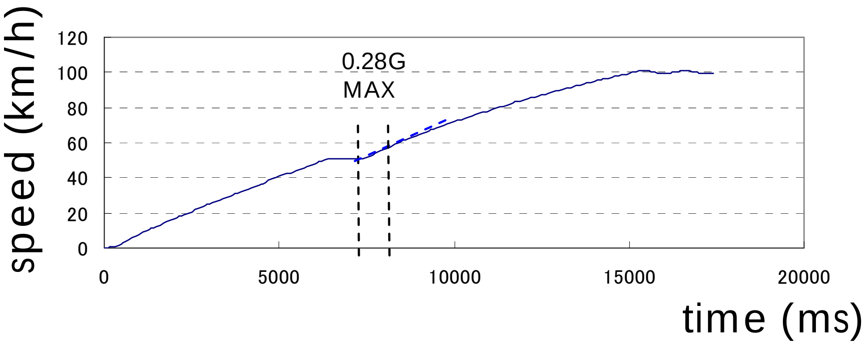
D-Case provides the system environment, restriction and criteria. SysML models provide structure and behavior of the component. These are used for verification.

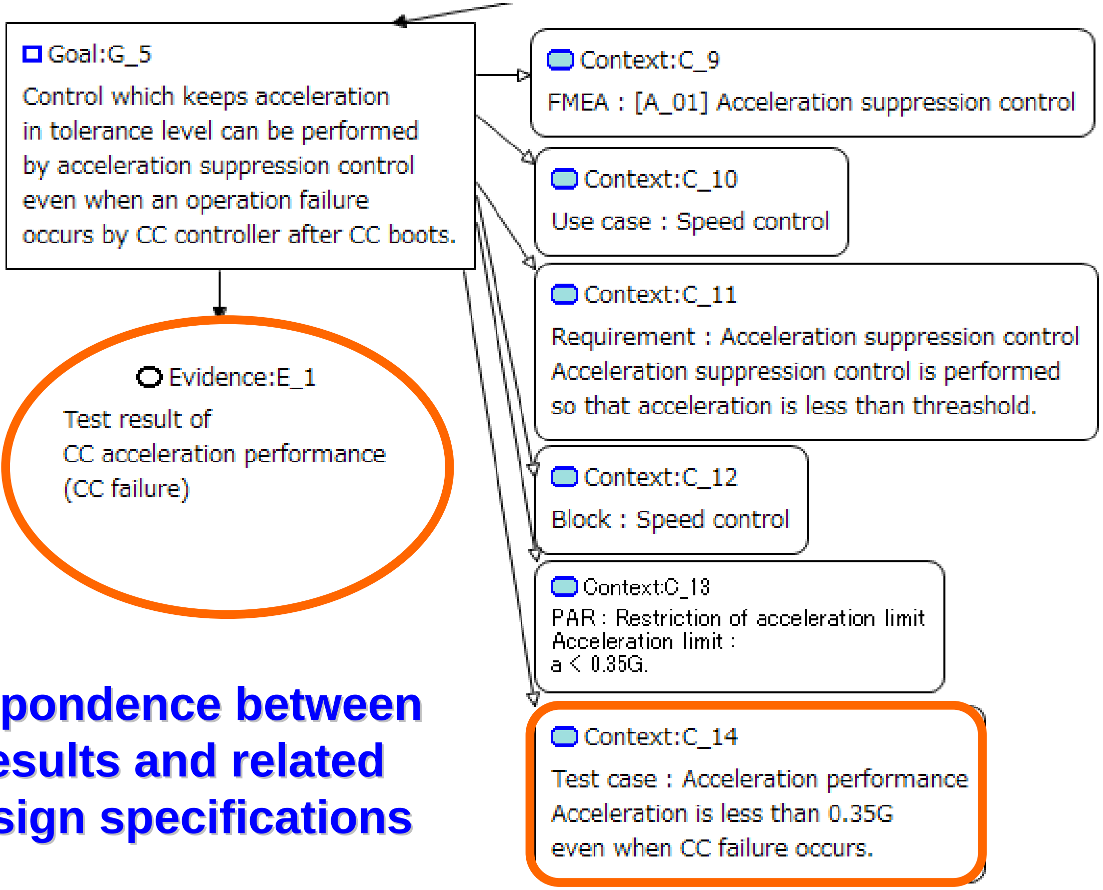
System Verification

Clarify relation of verification results and demands or design specifications by associating model simulation results to goals as evidences

Simulation Result

D-Case





□ Goal:G_5
Control which keeps acceleration in tolerance level can be performed by acceleration suppression control even when an operation failure occurs by CC controller after CC boots.

○ Evidence:E_1
Test result of CC acceleration performance (CC failure)

Context:C_9
FMEA : [A_01] Acceleration suppression control

Context:C_10
Use case : Speed control

Context:C_11
Requirement : Acceleration suppression control
Acceleration suppression control is performed so that acceleration is less than threashold.

Context:C_12
Block : Speed control

Context:C_13
PAR : Restriction of acceleration limit
Acceleration limit :
a < 0.35G.

Context:C_14
Test case : Acceleration performance
Acceleration is less than 0.35G even when CC failure occurs.

Verification Result : Clear !

Check correspondence between verification results and related demands, design specifications

Conclusion

This Documents Shows the Method and Guide of Developing Dependable System from upper process to lower process by D-Case and SysML Collaboration

Development of Dependable System

Development of Dependable System is realized by D-Case and SysML Collaboration

- Just enough derivation of system demands
- Correct derivation of design specifications by D-Case data collaboration function on SW development environment
- Clarifying relationship of verification results, demands, and design specifications

Guide

- Modeling Flow of D-Case and SysML
- Style Guide for D-Case Node
- Collaboration of D-Case and SysML Model Description

END OF PACKAGE