



D-Case Modeling Guide for Target System



Table of Contents

1	Scope	4
2	Overview of D-Case and SysML Modeling Guide	4
2.1	Background and Purpose	4
2.2	Target System of Modeling Guide	5
2.3	Constitution of Modeling Guide	6
3	Configuration Elements of D-Case	9
4	Structure of D-Case	15
4.1	Definition of Items	15
4.2	Identification of Hazards	18
4.3	Decomposition Based on Functional Safety Requirements	19
4.4	Decomposition Based on Technical Safety Requirements	22
4.5	Guaranty by Verification Results	24
5	Notations for D-Case Nodes	26
5.1	Notation for Goals to Achieve, Environment and Restriction	26
5.2	Notation for Environment and Operation of System	27
5.3	Notation for Detailed Cause to Take Actions	27
5.4	Notation for System Requirements	29
5.5	Notation for Information Required for Architecture	31
5.6	Notation for Information Required for Verification	32



Revision History

Revised Date	Description
2014/01/27	Created



1 Scope

This document guides the D-Case notation and definition in order to enable the collaboration of D-Case and SysML.

2 Overview of D-Case and SysML Modeling Guide

2.1 Background and Purpose

Recently, embedded systems are used by users in many fields. They have become complicated to satisfy various demands. The demands consist not only of functional demands from users but also of non-functional demands related to dependability. Dependability includes attributes of safety, reliability, availability, integrity, maintainability.

This guide shows an approach by D-Case which consistently realizes the dependability of the target system from upper process to lower process. Table 2-1 shows what is asked for developing a dependable system.

Table 2-1 What is asked for developing a dependable system

Development Phase	What is asked
Requirements Definition	System demands should satisfy dependability
System Design	Design specifications should reflect the demands correctly
System Verification	Verification results should account for satisfying dependability

In requirements definition phase, system demands should be derived by removing all the factors which inhibit dependability so that system demands satisfy dependability. D-Case is utilized for just enough derivation of system demands. D-Case decomposes dependability based on defining threat to dependability, scene of threat, cause, and provision. This decomposition extracts all the factors which inhibit dependability and marshals their provisions as system demands.

In system design phase, system design should be performed by utilizing the design information which is included in the demands based on dependability so that design specifications reflect the demands correctly. Derived design specifications should also be verified just enough by checking with the demands based on dependability. To realize this, design information like functional or non-functional demand, system element, restriction to element, and verification condition is extracted from the demands derived from dependability, and design specifications are correctly derived. Next, design specifications are analyzed by correspondence check with the demands based on dependability.



In system verification phase, verification results should be associated to the demands or design specifications, and their positions should be clarified so that verification results satisfy dependability.

The flow of the method is shown in Figure 2-1.

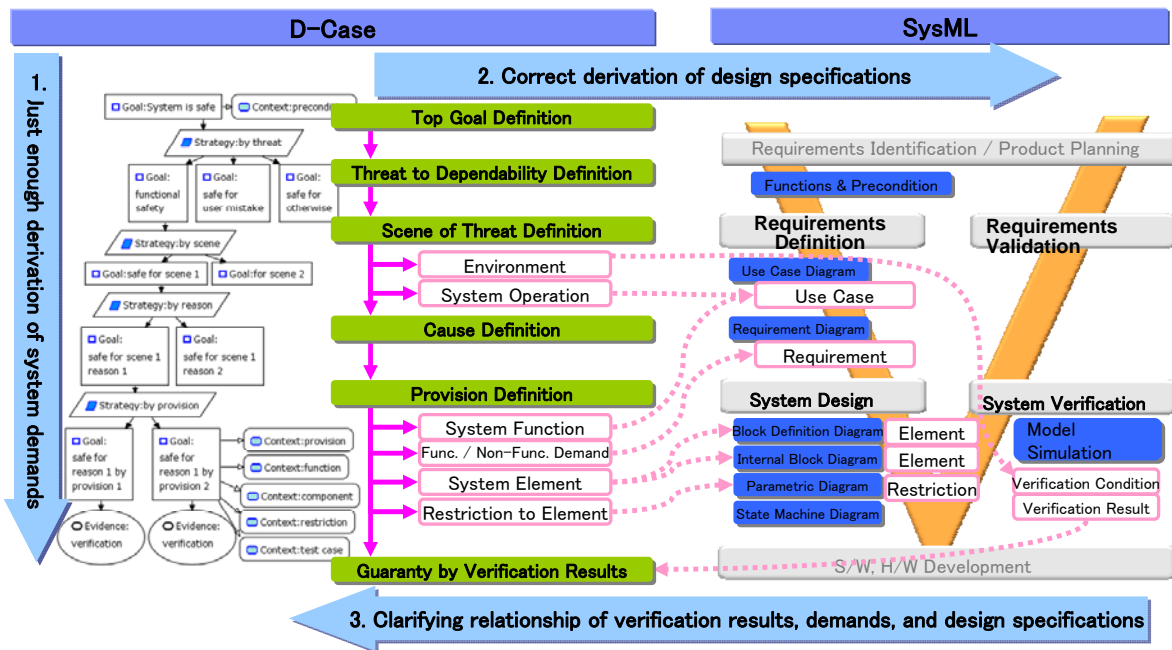


Figure 2-1 Dependable System Development Method by D-Case and SysML Collaboration

In this method, D-Case improves quality by reflecting development intents from upper process to lower process. SysML models can be made by reflecting the system demands derived from D-Case decomposition. As the sub-goals derived from the D-Case decomposition contain activities needed by the development, the accuracy of development plan can be enhanced by reflecting the activities to the plan.

This method is guided in the following documents:

- D-Case Modeling Guide for Target System
- SysML Modeling Guide for Target System
- D-Case Template
- SysML Template

2.2 Target System of Modeling Guide

Target of this guide is the in-vehicle system complying with ISO26262, the global standard of functional safety for vehicles. The derivational development is assumed in



that the intents such as safety demands or reliability demands are reflected to previous model already developed to adapt functional safety.

2.3 Constitution of Modeling Guide

Relationship of modeling flow by this method and ISO26262 safety lifecycle is shown in Figure 2-2. The constitution of D-Case decomposition corresponds to ISO26262 part 3 concept phase and part 4 product development: system level.

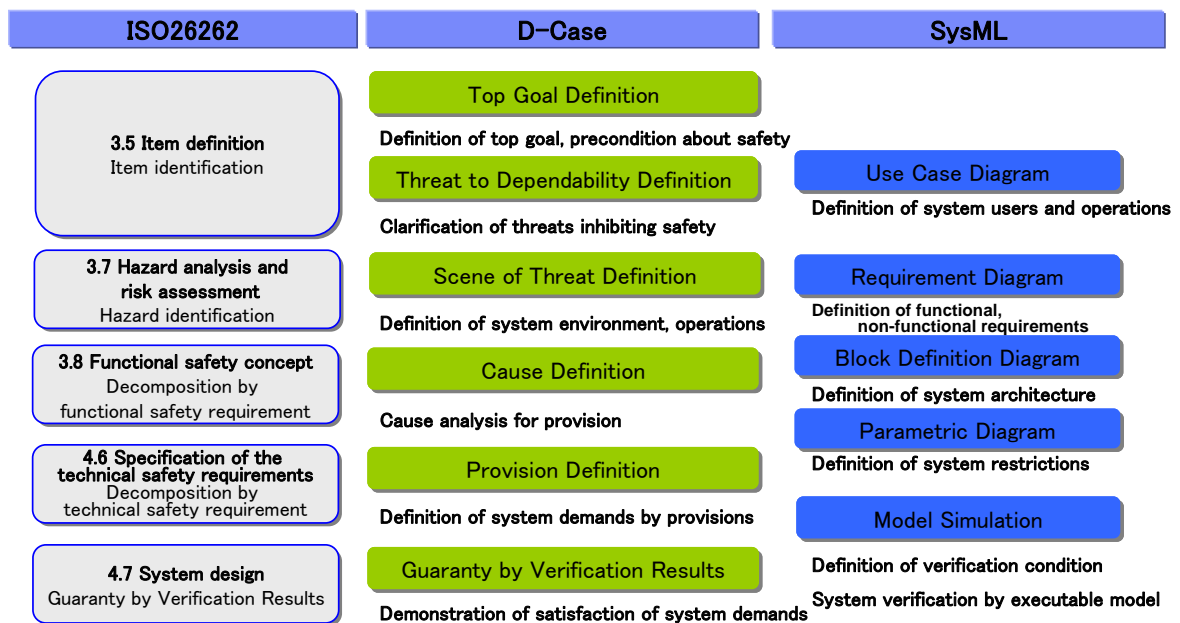


Figure 2-2 Relationship of Modeling Flow and ISO26262



Constitution of D-Case, SysML modeling guide is shown in Table2-2.

Table2-2. Constitution of Modeling Guides

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	Func. / non-func. requirement, System elements, Restriction	Use Case, Requirement, Component, and Restriction
	Guaranty by Verification Results	Information required to Verification	Condition and Processing of Control	Verification Result

This guide mainly guides as follows:

1. D-Case structure based on ISO26262 safety lifecycle

The property which the system for development should satisfy is described as D-Case top goal in the form of proposition. The whole structure of D-Case decomposition is considered to accomplish the D-Case. The top goal is divided into ISO26262 part and other part based on ISO26262. The D-Case of ISO26262 is divided by utilizing work products made in the activities of safety lifecycle. The decomposition flow of D-Case is explained in chapter 4.

2. Notation of D-Case nodes providing information needed by SysML model

The information needed by SysML model is described in the process of D-Case decomposition. The notation which is suitable for SysML collaboration is explained in chapter 5.



3. Writing procedure of SysML model based on the information of D-Case

SysML models are created or updated by extracting information needed for SysML model. Procedures are explained in “SysML Modeling Guide for Target System”.

Figure 2-3 shows the modeling flow of D-Case and SysML.

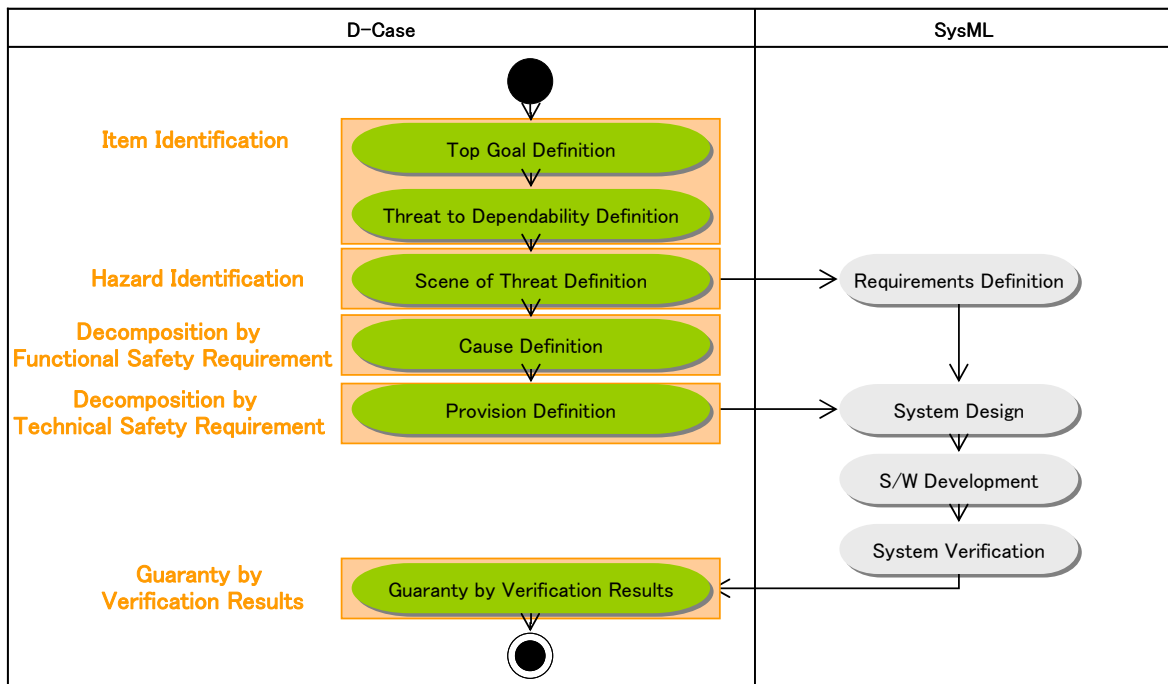


Figure 2-3 Modeling Flow of D-Case and SysML



3 Configuration Elements of D-Case

In this document, D-Case is linked with artifacts of ISO26262 safety lifecycle and with SysML model contents, and is placed as the main role for development process.

Basic structure of D-Case is shown in Figure3-1. Then describe the major nodes composing D-Case.

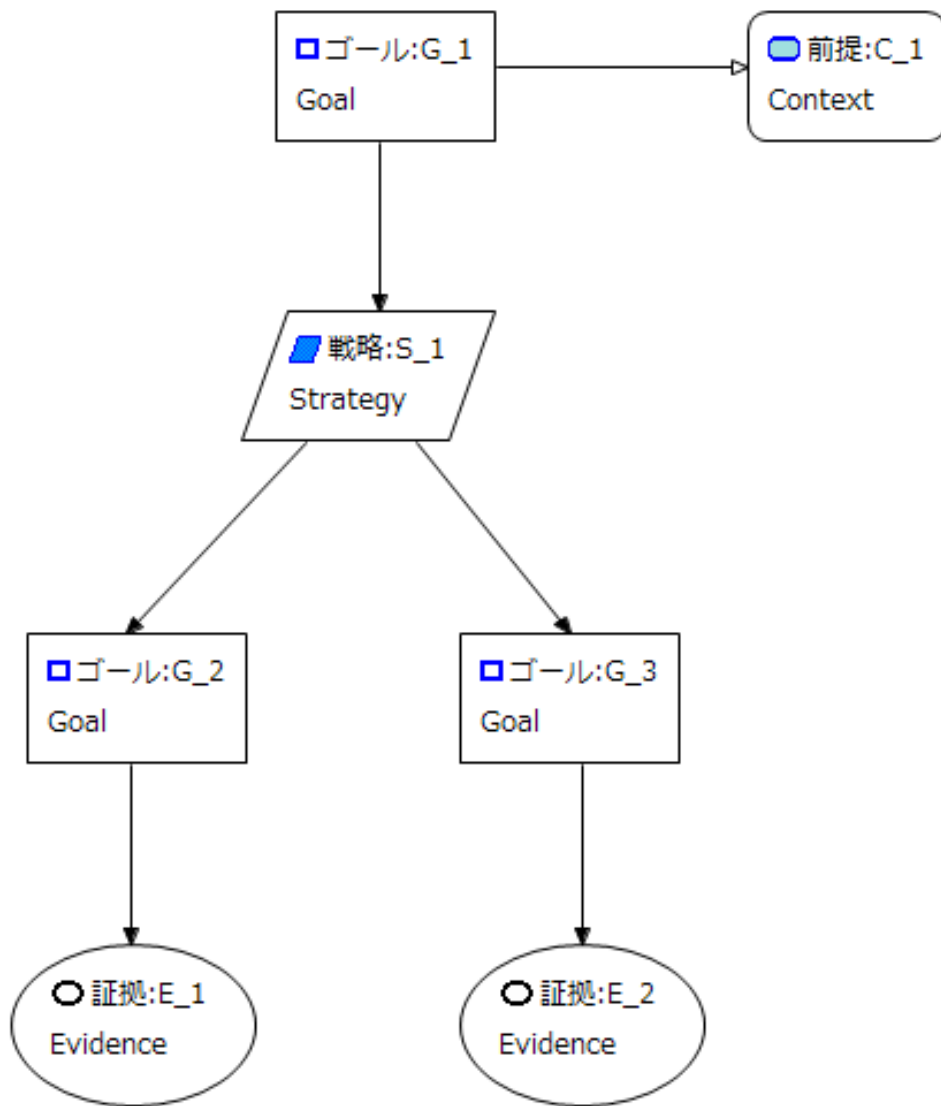


Figure3-1 Basic Structure of D-Case



Goal

Goals represent the behavior target systems need to accomplish (Figure3-2) .

Higher goals are decomposed to Lower sub-goals, and detailed. Goal nodes are stated in propositional formats such as “system is safe”. In order to apply to ISO26262, we focus on safety and reliability requirements of functional safety and describe that the system is meeting these requirements in higher goals. Next, decomposition to sub-goals is done following the artifacts created through safety lifecycle such as hazard analysis result, determined in ISO26262. Then finally, through realization components and verification results, sub-goals having specific behavior to accomplish goals defined in top goals are reached. The description on goal is explained in chapter4.

In relation with SysML model, there are cases when goals correspond to functional or non-functional requirements in higher layer of D-Case. In these cases, from the decomposition structure of goals corresponding to requirements, impact scale of requirements or corresponding relationship with functions can be traced.

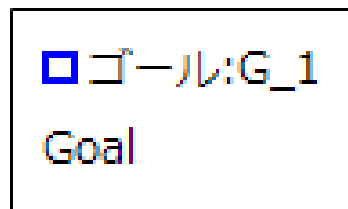


Figure3-2 Goal



Strategy

Strategies represent the idea of decomposing goals into lower sub-goals (Figure3-3) . In order to apply to ISO26262, since goals are decomposed following the safety lifecycle, goal decomposition strategies should be stated as “strategies”, using the result of hazard analysis such as HAZAP. For example, statement should be like “Argue about CC's safety for every scene of threat”. You can check if the system’s development flow is compliant to safety lifecycle by tracing the strategies from higher to lower layers of D-Case. The description on strategy in each decomposition step is explained in chapter4.

If the strategy decomposes the goal by system architecture, association with upper goal to lower goal expresses layer of system elements. Design intentions can be clearly stated by describing the name of the decomposition patterns in strategy nodes.



Figure3-3 Strategy



Context

Contexts represent precondition for goals or strategies (Figure3-4). In order to apply to ISO26262, artifacts outputted from safety lifecycle activities such as “results of hazard analysis” are stated. The description on context is explained in chapter4.

It is similar for relationship with SysML model, and system demands or system configuration requirements outputted from safety lifecycle activities are reflected to the context. Since the contexts are likely to contain wide variation of contents, you can clarify the corresponding relation with SysML model, by adding managing IDs or labels like “Use case”, “Requirement”, “Block”, “PAR”, and “Test case” for node statements.

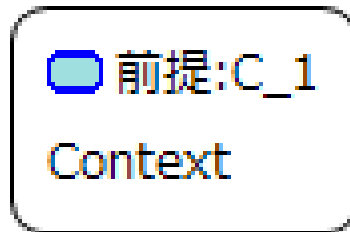


Figure3-4 Context

Evidence

Evidences represent the trails proving that the goals can conclusively be accomplished (Figure3-5) . In order to apply to ISO26262, state verification results indicating that top goals can be accomplished. Results of model simulations are reflected to the evidence. The description on evidence is explained in chapter4.

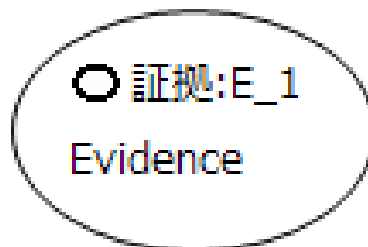


Figure3-5 Evidence



Figure3-6 represents the overall structure of D-Case, build by applying the configuration elements of D-Case and procedures explained in this document. In the higher layer, we defined the top goals as “Cruise Control System (CC) is safe”, which is our target development system. In order to determine the basic structure of D-Case, we defined the top goal’s decomposition strategy as “Consider actions for accidents caused by incomplete functions, accidents caused by user mistakes and unknown threats”, and decomposed into ISO26262 related parts and non-related other parts.

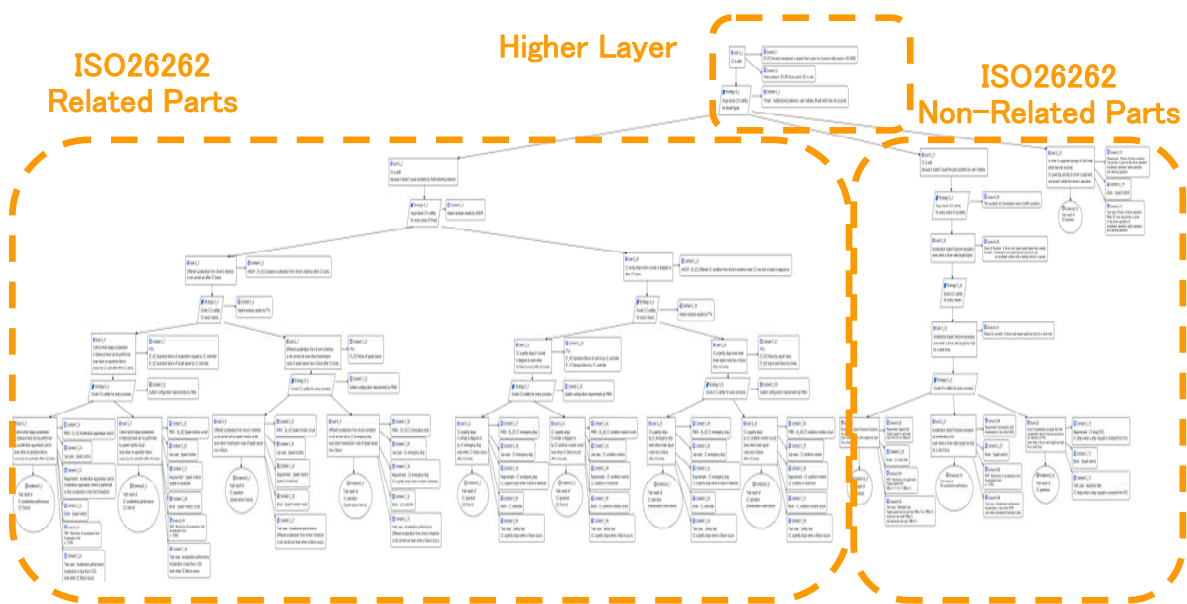


Figure3-6 Overall Structure of D-Case Developed



Decompose the ISO26262 related parts (Figure3-7) following the artifacts of ISO26262 safety lifecycle such as result of hazard analysis, and provides evidence from components conclusively realizing the requirements and result of verification.

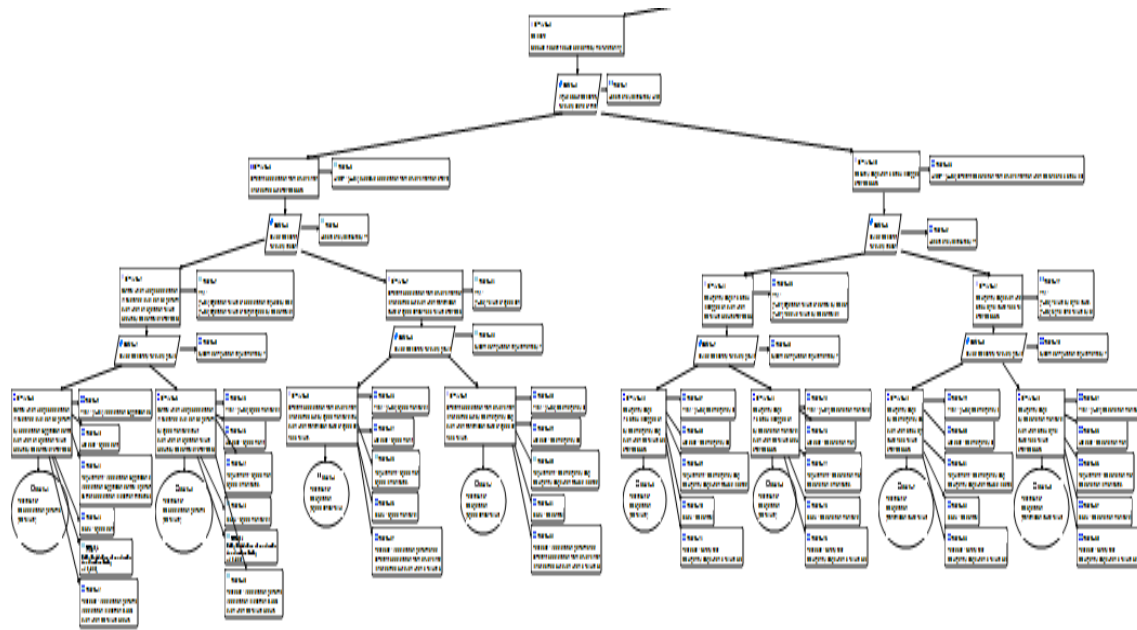


Figure3-7 Structure of ISO26262 Related Parts



4 Structure of D-Case

Decompose the goal system needs to accomplish into ISO26262 related and non-related parts, and then refine the D-Case for ISO26262 related parts basing on the safety lifecycle (Table2-2).

Table4-1 Structure of D-Case

Category	Structure of D-Case
Guide Items	Definition of Items
	Identification of Hazards
	Decomposition Based on Functional Safety Requirements
	Decomposition based on Technical Safety Requirements
	Guaranty by Verification Results

4.1 Definition of Items

For top goals, based on the preconditions for the system, extract the characteristics the system needs to accomplish to goals in propositional format. Also, extract preconditions system needs to meet to context. Next, decompose the top goal, and design the basic structure to complete the D-Case based on threats inhibiting safety (Figure 4-1).

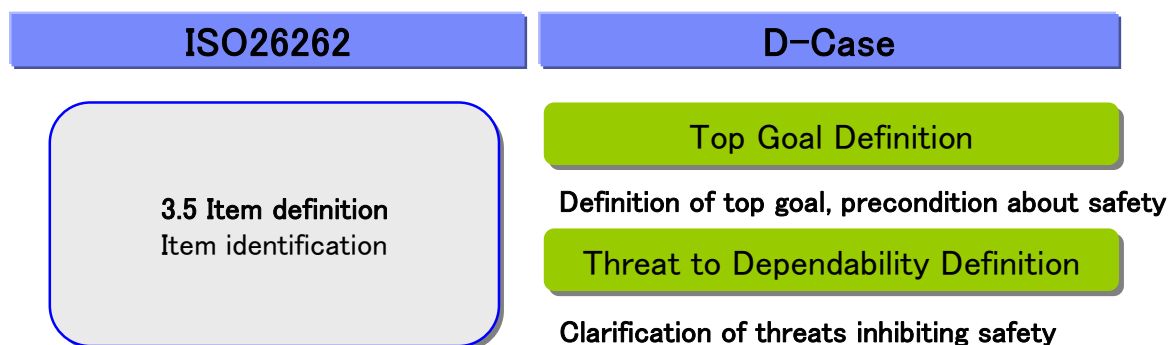


Figure 4-1 Definition of Items

(Definition of Items ISO26262-3-5)

Consider the configuration of the target system. Clarify environment and constraints surrounding the system and system boundary. State this information in the context



associated with the goal, and place them as preconditions for refining the D-Case.

- Focusing on safety, top goal has description “● system is safe”, and context associated to top goal has description “Premise demand : ● system is safe”.
- Focusing on threats inhibiting safety in functional safety, that is the target of derivational development, decompose the top goal to ISO26262 related part “malfunctioning behavior”, and others “user mistake”, “threat which has not occurred”.
- On this decomposition, strategy has description “Argue about ●'s safety”, and context associated to the goal has description “Threat : malfunctioning behavior, user mistake, threat which has not occurred”.
- Decomposed goals have description “● system is safe because it doesn't cause accidents by malfunctioning behavior.”, “● system is safe because it doesn't cause the past accidents by user mistake.”, and “In order to suppress damage of the threat which has not occurred, ● system does ● function.”.

Figure 4-2 shows extracting top goal. As the characteristics the system needs to accomplish to meet ISO26262, “CC is safe” is described in top goal.

Top Goal of D-Case

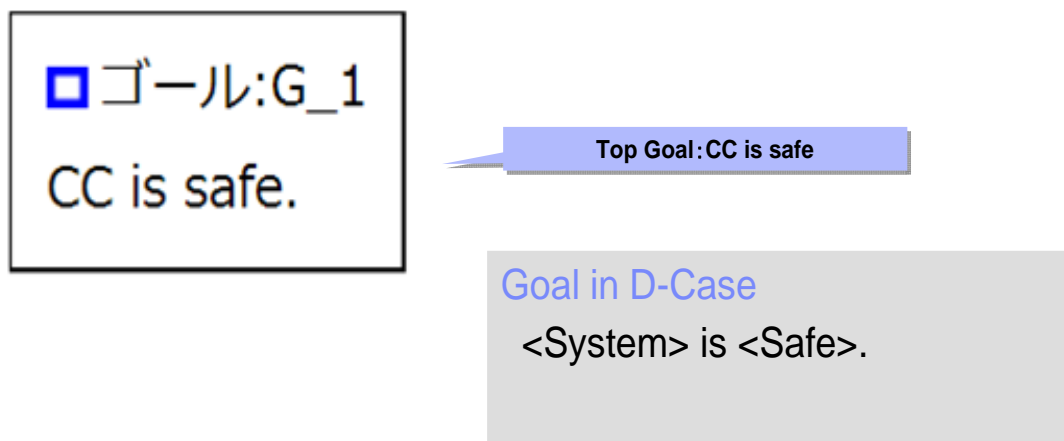


Figure 4-2 Extract top goal

(Example)

Figure 4-3 shows extracting system requirements like environment, restriction or border of systems. To achieve top goal “CC is safe”, extract requirements “[CY_01]



Derivation development is adopted. Next system has functional safety based on ISO 26262” and “Premise demand : [CY_02] Cruise control (CC) is safe.”. Describe these requirements into contexts related to the top goal.

Pre-requirement

- [CY_01] focus on the safety in this demonstration.
- [CY_02] CC is safe.

Top Goal

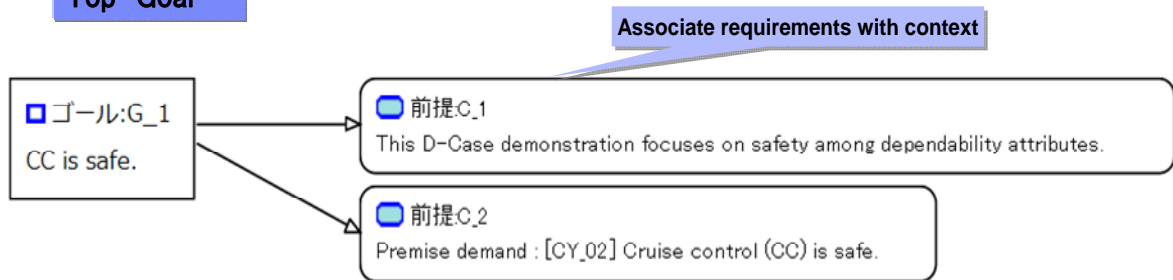


Figure 4-3 Extract system requirements

Figure 4-4 shows decomposition by ISO26262 related parts and others.

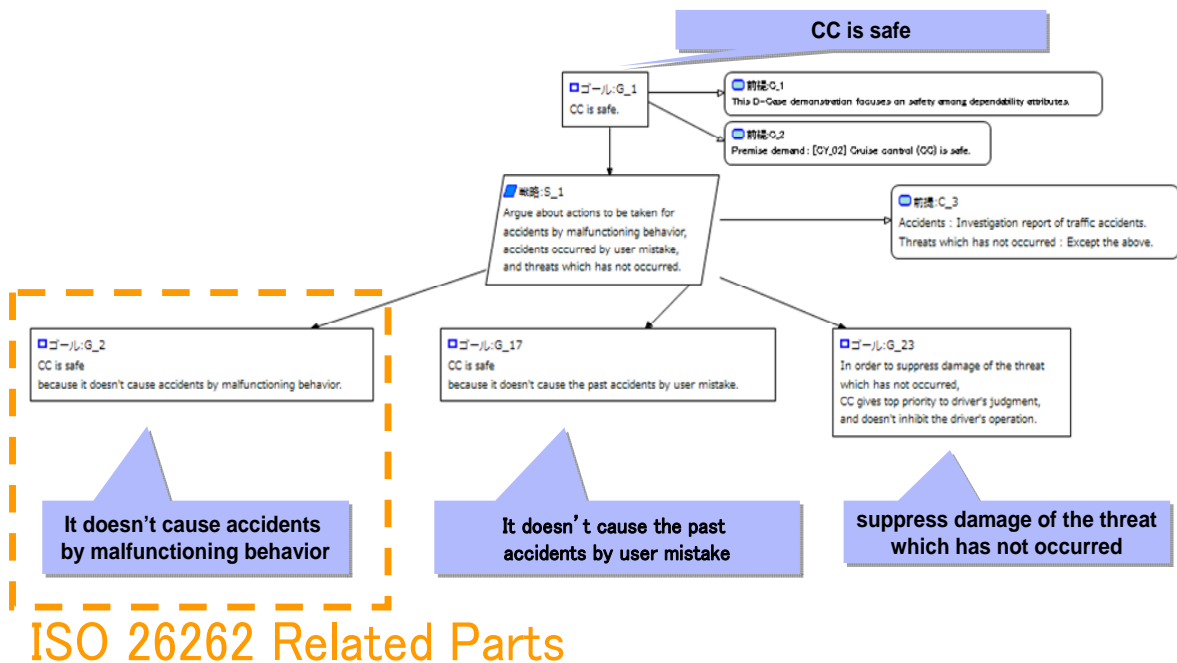


Figure 4-4 Decomposition by ISO26262 related parts and others



4.2 Identification of Hazards

To refine the goals for ISO26262 related parts, decompose by scene of hazards inhibiting safety of the target system (Figure 4-5).

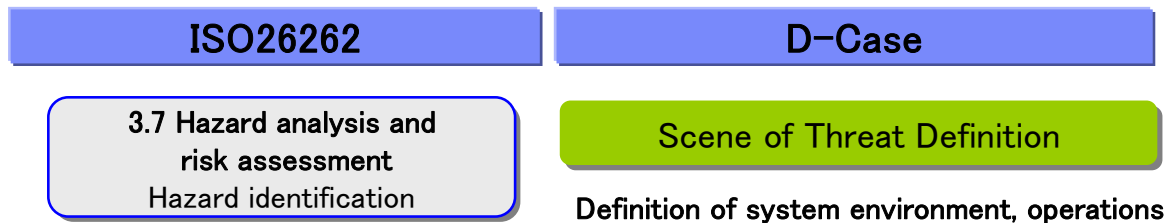


Figure 4-5 Identification of Hazards

(Hazard ISO26262-3-7)

Analyze the cause of danger triggered from behavior due to system functions' incompleteness, and decompose the D-Case based on the result of hazard analysis. Example of hazard analysis are as follows; HAZOP, FTA and FMEA (Table4-2) . By utilizing these analysis results, goals system need's to accomplish can be decomposed without any lack, covering the possible hazards (Figure4-6) .

Table4-2 Example of Hazard Analysis Methods

Method	Description
HAZOP	Identification of hazards by checking the derivation from expected value
FTA	Top down analysis on root cause of hazards
FMEA	Bottom up analysis on impact to failure modes and higher items

- Focusing on scene of threat, decompose the goal, caused by malfunctioning behavior.
- On this decomposition, strategy has description “Argue about ● system's safety for every scene of threat”, and context associated to the strategy has description “Hazard analysis results by HAZOP”.
- Decomposed sub goal has description “When ▼, ● system is safe for hazard ”, and context associated to the goal has description “HAZOP : Excessive ● when ▼”.



(Example)

Figure4-6 shows decomposition of ISO26262 related parts based on HAZOP analysis result. Hazards, which are related to sub goals, are described in context node, and have a relation to sub goals.

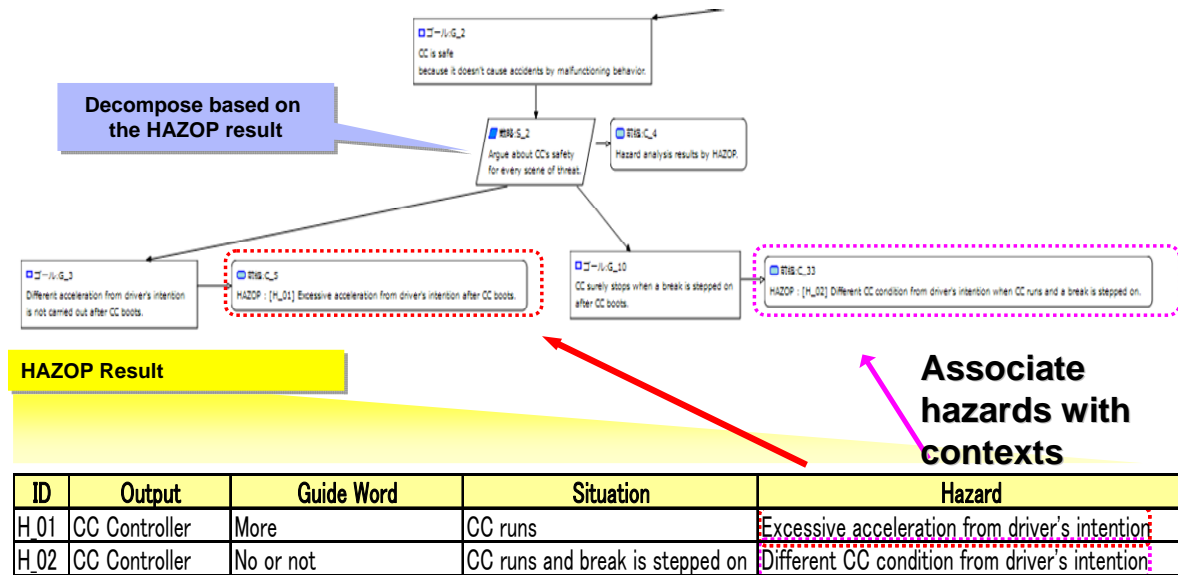


Figure4-6 Decomposition by Hazards

4.3 Decomposition Based on Functional Safety Requirements

Analyze causes of hazards for provision from goal decomposed by hazards (Figure 4-7).

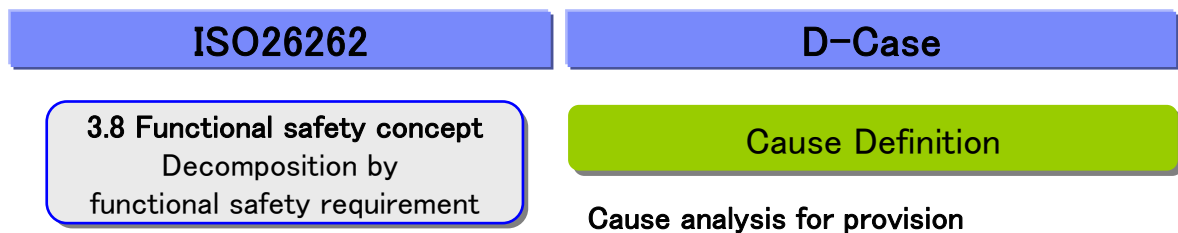


Figure 4-7 Decomposition Based on Functional Safety Requirements

(Safety Objectives ISO26262-3-7)

Decompose D-Case by extracting requirements the system needs to meet in order to avoid hazards, basing on the result of hazard analysis and risk assessment. From the result of FMEA analysis and risk assessment, analyze the impact and consider the actions needed for each of the failure modes.



(Functional Safety Concept ISO26262-3-8)

Consider the prior assumption of system configuration, and decompose the requirements into requirements corresponding to components, based on the architecture.

- Focusing on cause of hazard, decompose the goal.
- Strategy has description “Divide ● system's safety for every reason”, and context associated to the strategy has description “Hazard analysis results by FTA”.
- Decomposed sub goal has description “When ▼, control which keeps ● value in tolerance level can be performed even when a failure occurs at ▲ block.”, and context associated to the goal has description “FTA : Failure at ▲ block”.

(Example)

Figure4-8 shows Definition of cause of threats based on FTA analysis.

Based on FTA result, extract functional safety requirements from analysis result for the cause of hazard occurrence.

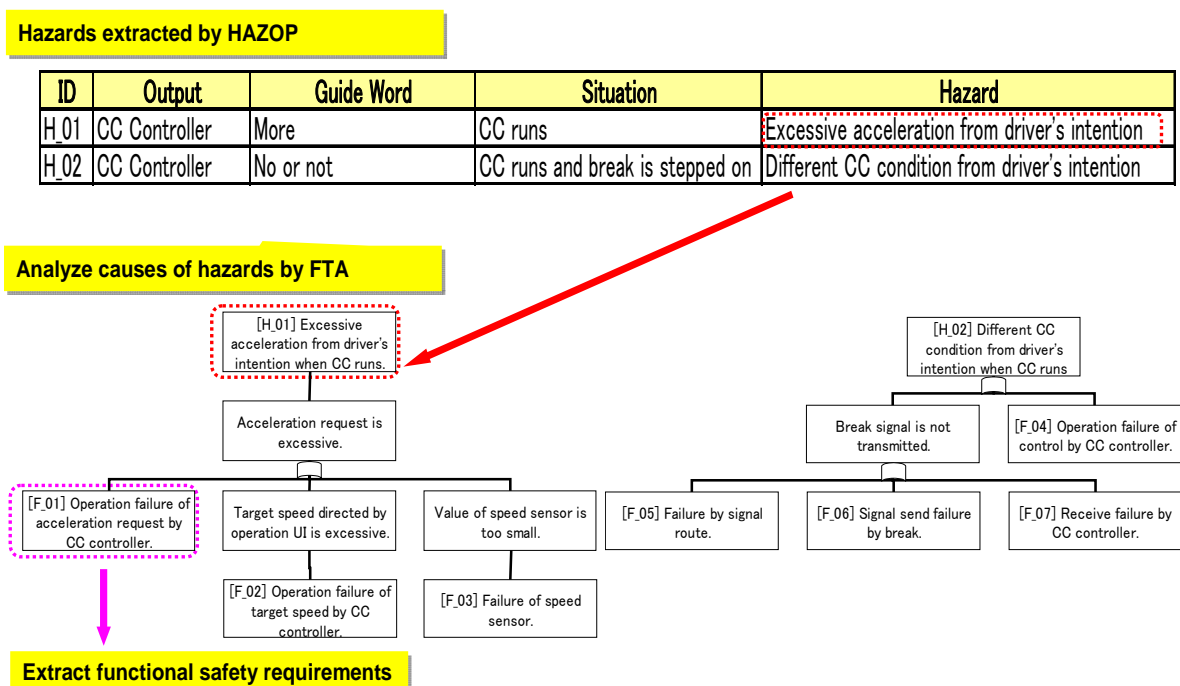


Figure4-8 Definition of Cause of Threats Based on FTA Analysis



Figure 4-9 shows decomposition based on defined cause of threats. Decomposing goal, which are developed by hazard, based on FTA result, goals are decomposed cyclopaedically.

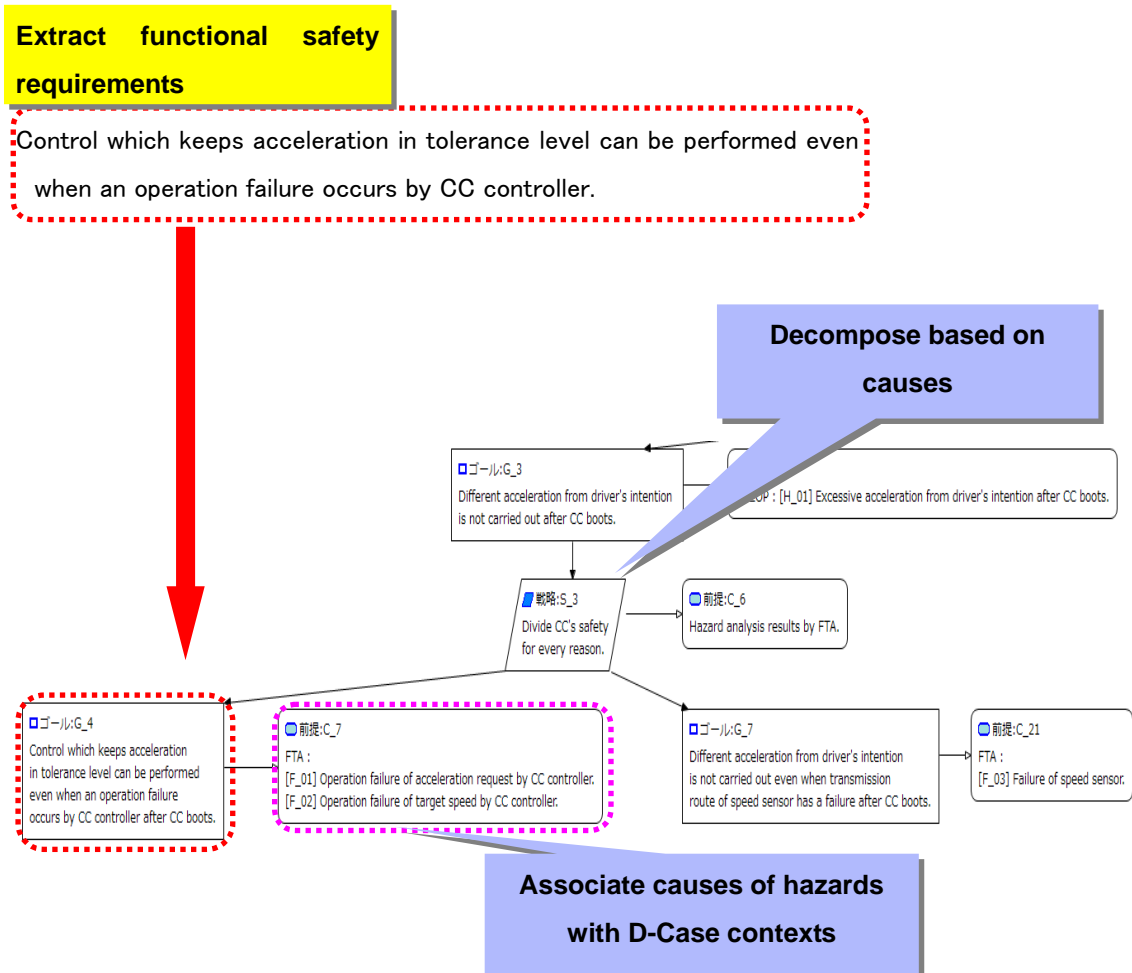


Figure 4-9 Decomposition based on Defined Cause of Threats



4.4 Decomposition Based on Technical Safety Requirements

Refine system demands, requirements, and design specifications based on provisions for causes (Figure 4-10).

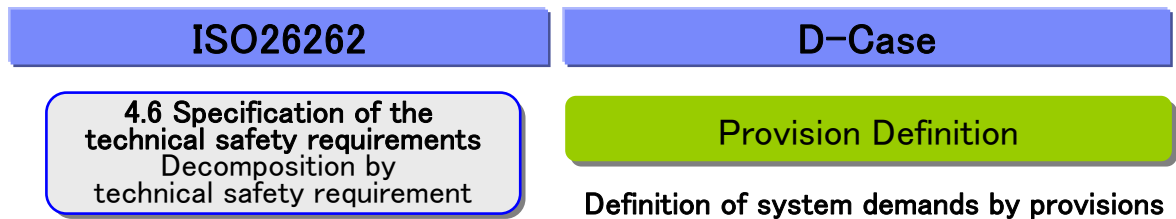


Figure 4-10 Decomposition Based on Technical Safety Requirements

(Technical Safety Requirements ISO26262-4-6)

Create specifications for each of the components with a requirement allocated, by considering the system's surrounding environment and constraints. Here, the system's surrounding environment and constraints stated in the contexts of upper hierarchy of D-Case are composed into component level, corresponding to the context of lower hierarchy of D-Case.

- Focusing on provision of hazard, decompose the goal.
- Strategy has description “Divide ● system's safety for every provision.”, and context associated to the strategy has description “System configuration requirements by FMEA”.
- Decomposed sub goal has description “When ▼, control which keeps ● value in tolerance level can be performed by ■ block even when a failure occurs at ▲ block”, and context associated to the goal has description “When ▼, control which keeps ● value in tolerance level can be performed by ■ block even when a failure occurs at ▲ block”.



(Example)

Figure 4-11 shows definition of provision based on FMEA analysis.

Based on FMEA result, make provision for each cause, and marshal system demands from provisions for each cause of threat.

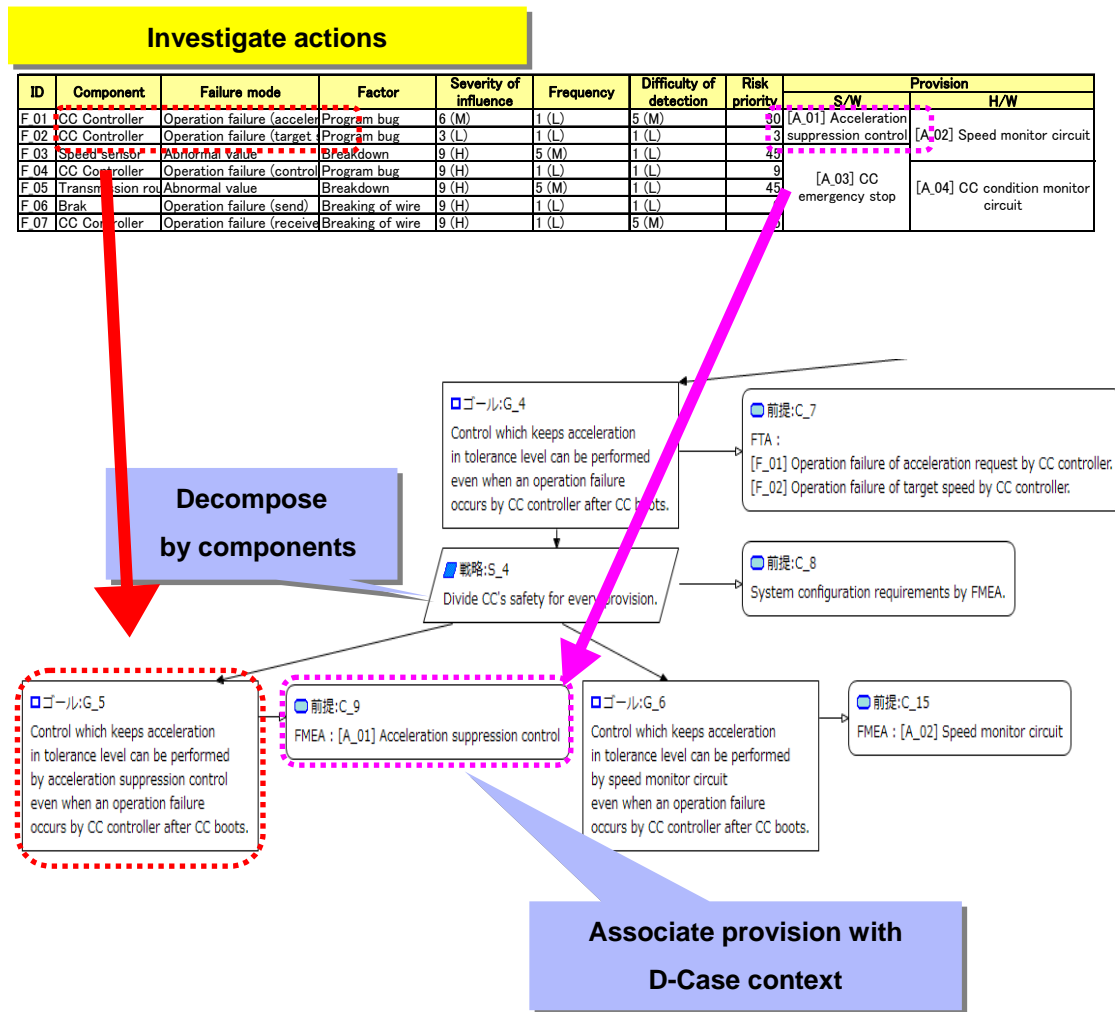


Figure 4-11 Decomposition based on realization component



4.5 Guaranty by Verification Results

Add trails according to verification results, to the specifications allocated. Avoid lacks in system verification by checking if each of requirement and of design specification is associated to one or more system verification results (Figure 4-12).

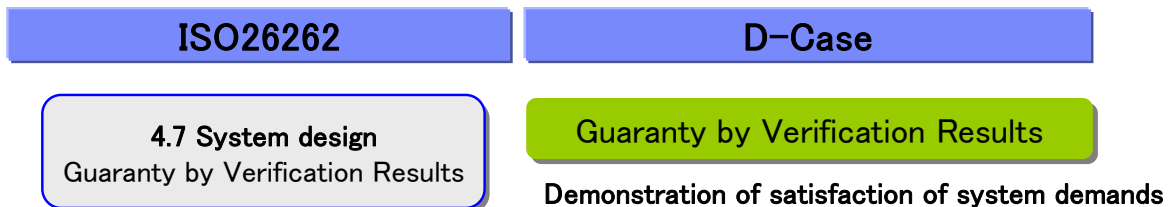


Figure 4-12 Guarranty by Verification Results

(Verification ISO26262-4-8)

Provide information required for verification from environment, constraints and criteria based on D-Case statements, and from component structure and behavior defined in D-Case decomposition structure and SysML model.

Table4-3 Available Information for Verification

Information	Source
System environment and constraints	Context of upper hierarchy of D-Case
Criteria for realization component to meet	Context of lower hierarchy of D-Case
Structure and behavior of realization components	Decomposition structure of D-Case and SysML models developed

- Verify the system satisfies the specification by verification.
- Evidence has description “Test result for ● function”, and associated to the verification result.



(Example)

Figure 4-13 shows guaranty by verification results.

SysML modeling environment provides model simulation. Develop architecture, behavior models, and test case based on restriction or other information described in D-Case. Model simulation is executed using this information. Add simulation result in evidence of D-Case

Avoid lacks in system verification by checking if each of the goals divided by D-Case is associated to one or more system verification results.

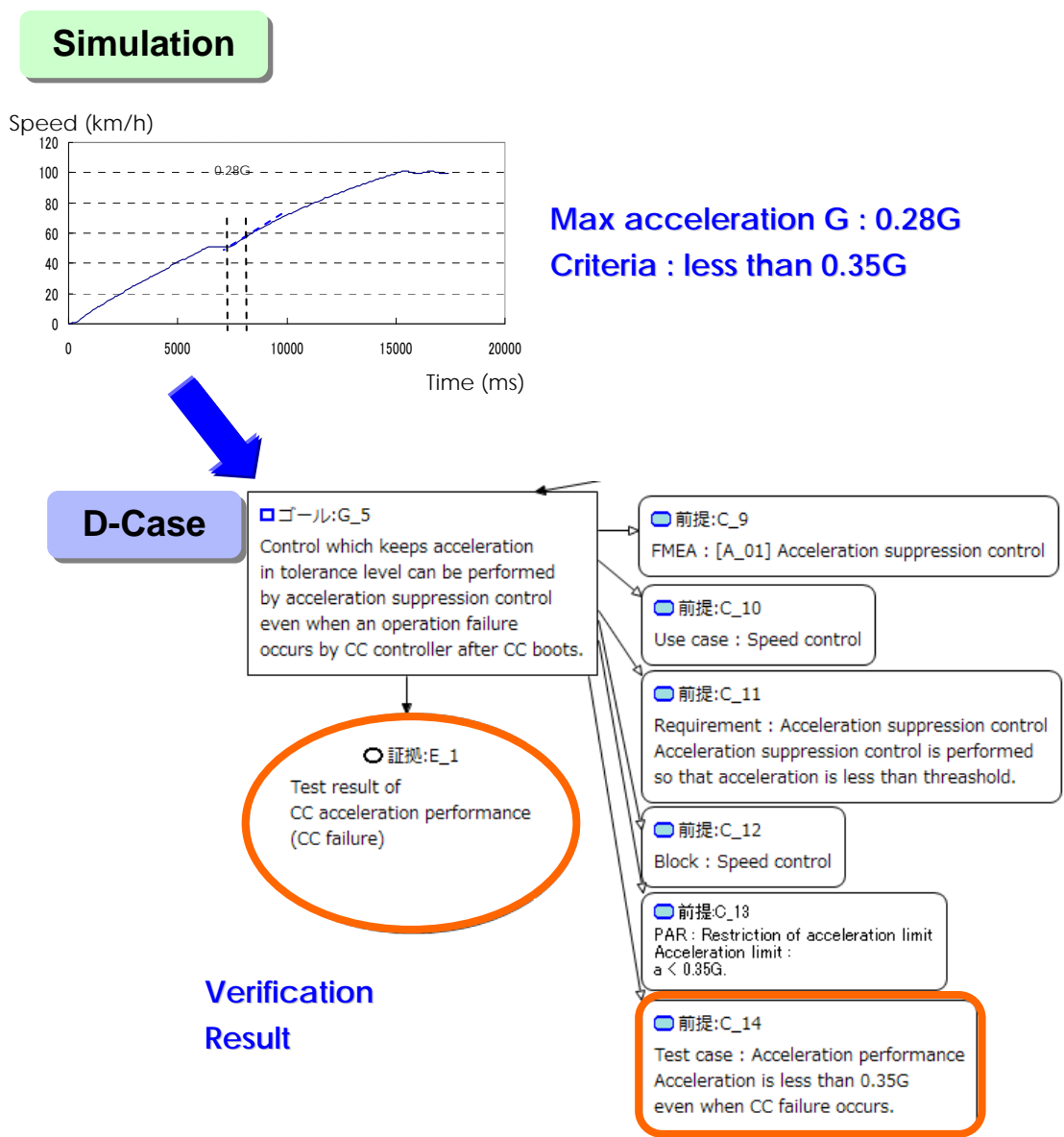


Figure 4-13 Guaranty by Verification Results



5 Notations for D-Case Nodes

This chapter guides the notational methods for using the information stated in D-Case nodes in SysML modeling (Table5-1).

Table5-1 Notations for Nodes

Category	Structure of D-Case	Notation for Nodes
Guide Items	Item Definition	Goal to Achieve, Environment and Restriction
	Identification of Hazards	Environment and Operation of System
	Decomposition by Functional Safety Requirements	Detailed Cause to Take Actions
	Decomposition by Technical Safety Requirements	System Requirement
	Item Definition	Goal to Achieve, environment and Restriction

5.1 Notation for Goals to Achieve, Environment and Restriction

(Association 4.1)

State behaviors system needs to accomplish in D-Case goals.

State environment and criteria for target system in context node of D-Case (Figure 5-1).

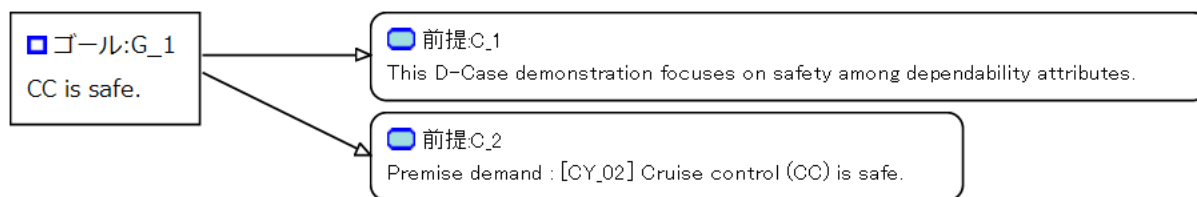


Figure 5-1 Description in goal and context



Context node has information as shown in Table5-2.

Table5-2 Information for Environment and Criteria

Information	Example
System Environment	Road Conditions, Eyesight and etc
System's External Boundary	User's Behavior and etc.
Criteria for System to Meet	Parameters for Constrains and Preconditions and etc.

5.2 Notation for Environment and Operation of System

Decomposing based on scene of threats, goal has description “When ▼, ● system is safe for hazard ”. “When ▼” provides information about users and external system of the target system. It is applied to actor in Use case diagram.

5.3 Notation for Detailed Cause to Take Actions

State the perspective for goal refinement in D-Case strategy nodes. State preconditions and related information used for goal refinement in D-Case context nodes.



(Association 4.2)

In goal decomposition based on identification of hazards, state the result of hazard analysis in context nodes, and decompose the goals into lower level goals following the strategy executing decomposition by hazards (Figure 5-2).

Structure of goal decomposition by strategy can be applied to hierarchical structure of requirement blocks of SysML's requirement diagrams.

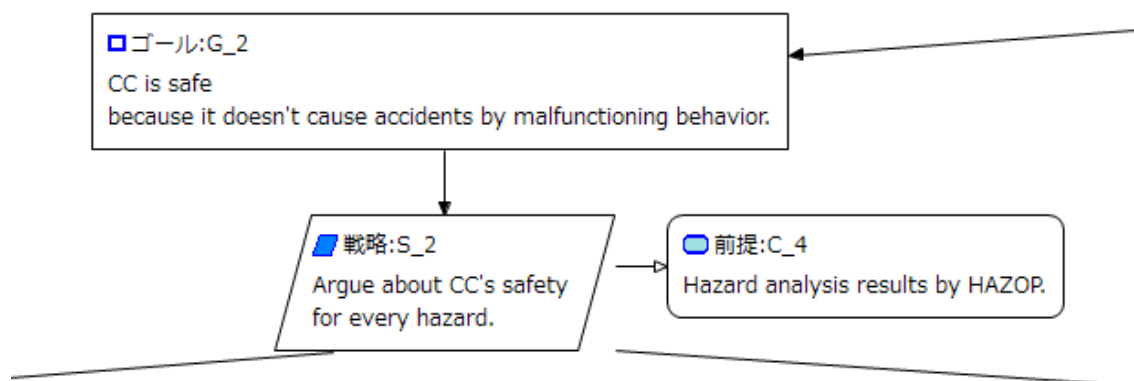


Figure 5-2 Decomposition based on hazard



5.4 Notation for System Requirements

(Quality of Documents)

It is important that the contents of the requirements are transferrable to lower part of the development process, since the characteristics of requirements will work as the base for system development. Standards for document quality are as follows; unambiguity, verifiability, completeness and simplicity (Table 5-3). By writing documents carefully following these benchmarks, documents have exact meaning.

Table 5-3 Benchmark of Document Quality

Benchmark	Standard
Unambiguity	No unclear relations No inconsistency between subjects and predicates No unclear postpositional particles No unclear signs (arrows, etc)
Verifiability	No statements without comparison objectives No unclear checkpoints No inadequate references
Completeness	No lacks of subjects or objects No statements on unsolved items ¹ No mistakes
Brevity	No redundant expressions No causative expressions or passive expressions No desiderate expressions No compound noun with multiple words are used or created

1

¹ If there is undeveloped item in developing D-Case, use undeveloped node temporarily to identify that consideration is needed later.



(Syntax for Requirement Statement)

EARS (Easy Approach to Requirements Syntax) template is used as the syntax for clearly writing the information required for system development in requirement statements (Table5-4) . By applying this syntax, it is possible to correctly extract the intentions from requirement statements and create specifications.

Table5-4 Syntax Table

Requirement Type	Writing Pattern
Ubiquitous	<System name> shall <system response>.
Event Driven	When <optional preconditions> <trigger> the <system name> shall <system response>.
Unwanted Behavior	If <optional preconditions> <trigger>, then the <system name> shall <system response>.
State Driven	While <in a specific state> the <system name> shall <system response>.
Optional Feature	Where <feature is included> the <system name> shall <system response>.



(Relationship between Statements)

Next, it is important to clarify the relationship between statements, since in many cases, system requirements are written crossing multiple statements. Five patterns are defined for relationship between statements (Table5-5) . These patterns are categorized by Input, Output, and Process of the requirement describes. Requirements must have description about Input, Output, and Process, and be consistent.

Table5-5 Relationship between Statements

Relationship Pattern	Description
Parallel (Arbitration)	Group and definition of statements with common process and output Statement that execute process with different value by different conditions
Parallel (Different Process)	Group and definition of statements with common input Cases where same input is used by multiple processes
Sequence	Group and definition of statements where one's output becomes the input for the other Cases where one's process result is used by the process of the other
Conversion	Group and definition of statements with common input, process and output Assume as same specification
Hierarchy	Definition of statements that is in the relation of overview and detail of specification Overview is used as the explanation of detailed specification

(Allocation of Management IDs)

From traceability's point of view, allocate control number to demands, requirements, and specifications.

5.5 Notation for Information Required for Architecture

(Association 4.4)

(Notation for Control Flow)

In order to write about system control, it is important to clarify inputs for the control, output and contents of the process.



(Hierarchical Structure used for Refinement of D-Case)

Refinement structure of goals using strategy can also be common with the architecture's hierarchical structure, in addition to the system structure information stated in the contexts of D-Case. Structure information becomes reusable to SysML model, by writing that the structure is decomposed based on the architectural hierarchy to strategy nodes.

5.6 Notation for Information Required for Verification

(Association 4.5)

(Omit Ambiguity in Behaviors Clarify Conditions and Responses)

For information used for verification, it is important that inputs used as conditions and expected responses are clarified.

(Quantitative Criteria)

For verifications, quantitative criteria are required to assess the responses outputted from verifications.

On the other hand, not only quantitative criteria work for non-functional requirements. For these types of items, it is possible to leave trails by writing consensus-building processes and results in D-Case.