SCDL 仕様書

Version 1.5

2020/01/06

権利関係、免責、商標などについて

※本書に記載している内容や参照文献、URL等は予告なく変更する場合があります。
※本書では正確な記述につとめましたが、安全コンセプト記法研究会は本書の内容に対してなんらかの保証をするものではなく、内容やサンプルに基づくいかなる運用結果に関してもいっさいの責任を負いません。
※本書ではTM、®、©などの表記は割愛しております。

【本書に関するお問い合わせ先】
本書の2次利用などのお問い合わせは
研究会URL　http://www.scn-sg.com　サイト内の「コンタクト」フォームをご利用ください。

まえがき

本仕様書は２つのパートで構成される。第 1 部では Safety Concept Description Language(SCDL)の目的を示し、文法[1]を定義する。第 2 部では SCDL の利用ガイダンスを提供する。

本仕様書は 2015 年から 2019 年にかけて、SCN-SG に参画した有志によって作成された。

本仕様書の変更履歴は 3 ページに、目次を 8 ページに示す。

[1] ここでの文法は、必ずしもメタモデルによって定義される厳密なシンタックスのみを意味しない。記述法・表記法・記法などの意味を含む。

変更履歴

Version	作成日	概要
1.0	2016/04/04	初版公開
1.1	2016/11/11	誤記修正（4.3.4 無干渉要求表記にて、無干渉の始点をエレメントに限定するよう修正） 脚注一部修正
1.2	2017/02/17	● CONTRIBUTORS のアップデート、および附属書 A の Editor 追加 ● 用語見直し(機能要求、非機能要求 脚注 5,6) ● 3.5 節を構造図の概要説明に留め、4.4 節にて構造図の詳細を記述するように修正 ● 4.1 構造図の種類をアップデート ● 4.2 色の範囲(塗りつぶし色、線色、文字色など)を記載 ● 4.2 表 3 の要求グループ 箇条書き 3 に補足説明を追加 ● 4.2 表 4 にて、関係を表す要素であることを強調するため、無干渉線、要求遇ループペアリング線、引き出し線から"線"を削除 ● 4.2.1 非機能要求の定義見直し ● 4.3.1 要求グループペアリングの表記方法を補足。また、すべての要求グループの表記方法ごとの例を記載 ● 4.4 構造図の種類から相互関係の説明を削除し、それぞれの構造図の詳細記述を見直し ● 5.6 外部プラントの例を改善 ● 附属書 A ユースケースを追加 ● 9 参考文献の追加
1.3	2017/07/28	● 3.3 節 機能要求、非機能要求に対する追記 ● 非干渉を無干渉に修正 ● 附属書 A ユースケース(自動車分野への適用事例)での図番号参照の誤記修正 ● 附属書 B ユースケース(ソフトウェア編)を追加 ● 附属書 C メタモデルを追加
1.4	2018/08/13	● 附属書 D モデル入出力・データ交換を追加 ● 誤記修正
1.5	2020/01/06	● 附属書に、ユースケース(ハードウェア編)を追加。追加による附属書の章番号修正

CONTRIBUTORS

本仕様書は、安全コンセプト記法　研究会（SCN-SG）の活動により策定された。なお、本項記載の所属および氏名は発行当時のものである。

(50 音順)

青峰　亮子　　（株）東芝
板本　英則　　（株）ジェイテクト
岩崎　保　　ＫＹＢ（株）
内山　幹康　　アンソレイエ（株）
梅田　崇史　　三菱自動車工業（株）
益　　啓純　　（株）ジェイテクト
小黒　龍一　　（株）ニッキ
小田　祐司　　オートリブ（株）
川口　貴正　　（株）日立製作所
河野　岳史　　スパークスシステムズジャパン（株）
河野　文昭　　（株）アドヴィックス
小島　仙睦　　（株）ニッキ
酒井　英子　　（株）デンソー
佐々木　喜好　　カルソニックカンセイ（株）
島中　茂樹　　ジヤトコ（株）
島村　俊一　　日産自動車（株）
末冨　隆雅　　マツダ（株）
杉村　嘉秋　　住友電装（株）
関　　康大　　日本精工（株）
東道　徹也　　（株）デンソー
宮崎　義弘　　日立オートモティブシステムズ（株）
宮本　秀徳　　（株）構造計画研究所
村松　稔久　　スズキ（株）
森　　広樹　　（株）デンソー
山下　修平　　DNV GL ビジネス・アシュアランス・ジャパン（株）
湯山　俊夫　　（株）東芝

ガイオ・テクノロジー(株)
キャッツ（株）
ＳＯＬＩＺＥ　Ｅｎｇｉｎｅｅｒｉｎｇ（株）
(株)チェンジビジョン

ベクター・ジャパン（株）
三菱電機（株）
ヤマハ発動機（株）

1.1 Editors

- 第 1 章－第 5 章
  （主査）岩永　寿来　（株）チェンジビジョン
  以下、50 音順
  内山　幹康　アンソレイエ（株）
  大西　建児　ガイオ・テクノロジー（株）
  小田　祐司　オートリブ（株）
  河野　文昭　（株）アドヴィックス
  佐々木　喜好　カルソニックカンセイ（株）
  杉村　嘉秋　住友電装（株）
  村松　稔久　スズキ（株）
  山下　修平　DNV GL ビジネス・アシュアランス・ジャパン（株）

- 附属書 A ユースケース（自動車分野への適用事例）
  （主査）河野 文昭　（株）アドヴィックス
  以下、50 音順
  川口　貴正　（株）日立製作所
  佐々木　喜好　カルソニックカンセイ（株）
  島中　茂樹　ジヤトコ（株）
  杉村　嘉秋　住友電装（株）
  村松　稔久　スズキ（株）
  森田　徹　三菱電機（株）
  山下　修平　DNV GL ビジネス・アシュアランス・ジャパン（株）
  湯山　俊夫　（株）東芝

- 附属書 B ユースケース（ハードウェア編）
  （主査）河野 文昭　（株）アドヴィックス
  以下、50 音順
  内山 幹康　アンソレイエ（株）
  小田　祐司　ヴィオニアジャパン（株）
  川口 貴正
  佐々木 喜好　マレリ（株）

島中 茂樹　ジヤトコ（株）
杉村 嘉秋　住友電装（株）
関　康大　日本精工（株）
村松 稔久　スズキ（株）
森田 徹　三菱電機（株）
湯山 俊夫　（株）東芝

- 附属書 C　ユースケース（ソフトウェア編）
  （主査）宮崎　義弘　日立オートモティブシステムズ(株)
  以下、50 音順
  青峰　亮子　(株)東芝
  安部　秀二　パナソニック(株)
  岩崎　保　ＫＹＢ(株)
  岩永　寿来　(株)チェンジビジョン
  大西　建児　ガイオ・テクノロジー(株)
  小澤　弘正　三菱電機(株)
  河野　文昭　(株)アドヴィックス
  板本　英則　(株)ジェイテクト
  佐々木　喜好　カルソニックカンセイ(株)
  島中　茂樹　ジヤトコ(株)
  高山　剛　ガイオ・テクノロジー(株)
  東道　徹也　(株)デンソー
  中村　伸彦　ベクター・ジャパン(株)
  山下　修平　DNV GL ビジネス・アシュアランス・ジャパン(株)

- 附属書 D　SCDL メタモデル
  （主査）河野 岳史　スパークスシステムズジャパン(株)
  以下、50 音順
  岩永　寿来　(株)チェンジビジョン
  高山　剛　ガイオ・テクノロジー(株)
  佐々木 喜好　カルソニックカンセイ(株)
  宮本　秀徳　(株)構造計画研究所
  山下　修平　DNV GL ビジネス・アシュアランス・ジャパン(株)

- 附属書 E　モデル入出力・データ交換 仕様書
  （主査）河野 岳史　スパークスシステムズジャパン(株)

以下、50 音順

岩永 寿来　(株)チェンジビジョン
兼平 靖夫　ダッソー・システムズ(株)
仮屋 義明　(株)DTS インサイト
妹尾 覚　(株)DTS インサイト
高山　剛　ガイオ・テクノロジー(株)
佐々木 喜好　カルソニックカンセイ(株)
光山 栄太　ガイオ・テクノロジー(株)
宮本 秀徳　(株)構造計画研究所
山下 修平　DNV GL ビジネス・アシュアランス・ジャパン(株)

目次

2 はじめに

2.1 背景

現在、社会基盤を支える大規模システムから、生活関連製品にいたるまで、製品の安全性の論拠を示すことが強く求められている。ISO/IEC 一連の安全規格の発行により、化学プラント、原子力プラント、工作機械などの産業用機器を対象とした分野における安全活動をきっかけとして、システムの安全設計の見える化など、第三者への説明性向上が図られてきたことからも窺い知ることができる。この流れは、自動車分野においても ISO 26262[2] の発行をきっかけに一段の広がりを見せている。

安全設計の見える化ではシステムの安全性構想、安全性設計、安全性検証といった一連の安全活動に対する説明、すなわち、システムの安全性をどのようにして設計に織り込んだのか、安全性をどのようにして担保しているのかについての分かりやすい説明が求められる。

加えて、このことはシステムの保守性や、システムのセキュリティ確保のための対策を論じる際にも重要なテーマとなる。

安全性構想、安全性設計、安全性検証といった安全活動に必要な主要な要素は、要求とアーキテクチャである。安全活動においては、要求とアーキテクチャがどのような関係になっているかをビジュアルに表現するための表記法と検証するための方法論が望まれている。SCN-SG は、製品開発に携わる設計者や検証者、あるいは、製品評定に携わる第三者（評価者）にとって有益な表記方法、すなわち Safety Concept Description Language(SCDL)を提供することで、社会的ニーズに応えようとするものである。

2.2 目的

本仕様書は、SCDL の文法を定義し、利用ガイダンスを提供する。SCDL はシステムの安全設計をアーキテクチャ視点から整理し、論じるための表記法である。安全設計は既存の様々な記法でも表記できるが、安全設計の検討や論証をアーキテクチャ視点から一貫して連続的にサポートし、理解しやすく安全設計に着目したレビューが行えることが、より求められている。

このような課題に対応するため、SCDL では安全設計においてアーキテクチャを主題として扱い、安全機構と安全に関連する要素の識別を階層的に論じる。

SCDL で設計されるアーキテクチャは、要求[3]と、要求が配置されるシステムの構成要素

[2]ISO 26262：2011. Road Vehicles -- Functional Safety. ISO Standard.
[3]意図機能といった機能要求や非機能要求、安全要求など

[4]を明確に区別して扱え、要求間の依存関係や、要求のグルーピングやグループ間の制約となる要求（以降は制約条件と称する）、要求の配置と重み付け（例えば、ISO 26262 ではASIL）の波及範囲などが表現できる。

2.3 仕様書の構成

本仕様書は、以下に示す 5 つの章で構成される

- 3 章: SCDL とは
  - 本章は SCDL と安全設計での役割について概要とコンセプトを示す
- 4 章: SCDL の基本定義
  - 本章は SCDL の文法や表記といった最小限必要な要素を定義する
- 5 章: SCDL の拡張定義
  - 本章は SCDL の拡張定義を示す
- 6 章: SCDL のユースケース
  - 本章は SCDL を用いて安全設計を有効に行うための参考事例を示す
- 7 章: 自動車分野への適用事例
  - 本章は ISO 26262 への適用事例を示す

[4] ソフトウェア、ハードウェア部品など

3 SCDL とは

3.1 概要

本章では、SCDL の背景、目的やコンセプトについて示す。

3.2 イントロダクション、コンセプト

ISO/IEC 一連の機能安全規格の発行をきっかけにシステムの安全設計の考え方、言い換えると安全アーキテクチャのありかたが議論されるようになってきた。安全アーキテクチャが安全設計の説明性や評価・検証の容易性を確保する上で重要だという考え方である。安全設計の確からしさを論証するためには、少なくとも要求とシステムの構成要素の関係が曖昧ではなく、分かりやすく階層的に表現されていることが必要となる。

安全アーキテクチャは主に次の 1、2、3 のトピックを中心に論じられることが多い。

1. 想定されるリスクに対する十分な手当て
   安全上の手当ては一般的には安全方策・安全対策と呼ばれる概念だが、特にシステム上に実装することで効果を得るものを安全機構と呼ぶことが多く、SCDL ではこれを扱う。また安全機構は、図 1 に記載された属性を考慮する必要がある。

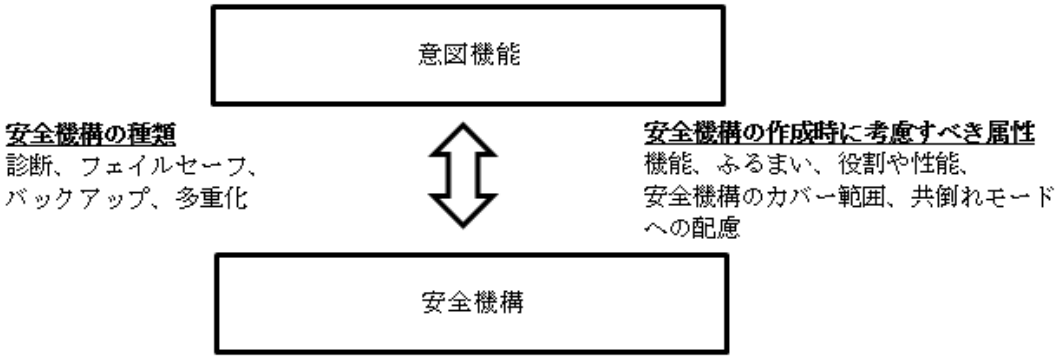


図 1 安全機構の仕様

2. 安全関連エレメントの識別と分離
   安全関連エレメントに関しては、重要度に応じた手厚い取り扱いが必要となるため、安全関連エレメントと非安全関連エレメントの識別や、安全関連エレメント内の重み付けをすることで、安全設計の論証が可能となる。

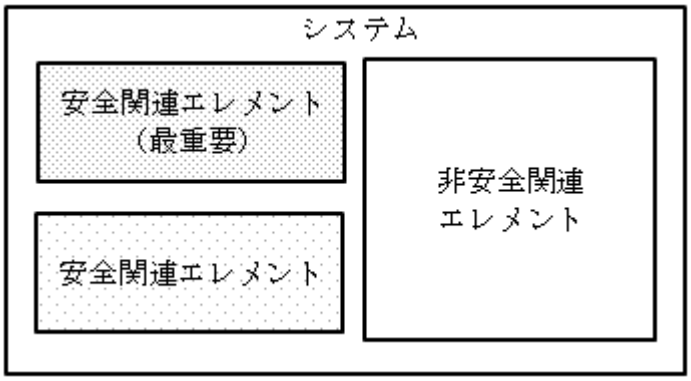


図 2　安全関連エレメントの識別と分離

3.　1 と 2 の体系的、構造的、階層的論拠

設計対象となるシステムではアーキテクチャはトップダウンで階層的に詳細化される。安全アーキテクチャもトップダウンで、例えば製品コンセプト、システムコンセプト、ハードウェアコンセプトおよびソフトウェアコンセプトといった形で階層的に具現化されていく。このように、どのレベルにおいても、安全機構の構造と安全関連エレメントの識別を階層的に論じることで、安全設計の論証を行う。

安全アーキテクチャは既存のシステム仕様の記述法:UML、SysML、ADL、EAST-ADL、DFD、 FBD 等を用いることで表現されている事例は少なくない。

しかしながら、安全設計の検討作業や論証を一貫して連続的にサポートするには、どの記法や言語を適用しても不足感があり、曖昧さのない準形式的言語と方法論が必要である。記述法は個々の技術論のニーズに応じた工夫の結果から生まれたものであり、安全アーキテクチャを適切に扱う道具立てとして準備されたものではないため無理があるのは致し方ない。そこから、安全アーキテクチャを分かりやすく図解しようとする試みの中から見えてくる安全設計オリエンテッドな準形式記法の検討が議論されるようになってきた。

限られたリソースの中で行われるシステム開発において、関係者間のコミュニケーションやレビュー時の負荷を考えると、安全アーキテクチャ用の記法は独自で特殊な記号の集合になることを避けなくてはならない点も重要である。これまでに使い慣れたグラフィカルな各種表現法の拡張版として直感的に仕様理解を助けることで、エンジニアが安全設計に着目した会話や議論に注力できることも、目指すべき記述法の重要な特性といえる。

3.3 適用範囲

本記法（または記述法、記述言語）は、安全設計に関する成果物を効果的かつ効率的に安全アーキテクチャとして作成、検証、再利用するための仕様記述に適用することを意図している。

さらに、重み付けなど、安全設計上のなんらかのパラメータを持つ、安全関連エレメントのアーキテクチャ記述に適用することを想定している。技術分野によっては、安全性に関する要求を非機能要求と分類する場合もあるが、本記法では、表 1 に示す要求、および安全・非安全関連を扱う。

表 1 SCDL 適用範囲

	機能要求 [5]	**非機能要求** [6]
安全関連	適用対象	必要に応じて記述する場合がある
非安全関連	必要に応じて記述する場合がある	原則、適用対象外

3.4 SCDL 記法の基本概念

前述の背景を踏まえ安全要求仕様や安全アーキテクチャを適切に論じようと考えた時に記述言語に求められる要件には以下が必要となる。

- 要求とエレメント（サブシステムやコンポーネントなど）を明確に区別して扱えると同時に要求とエレメントを同一のビューに重ね書きができる
- 要求間のインタラクション（情報・信号・メッセージ等の授受など）、機能的な依存関係を従来の DFD、FBD の表記で図示できる
- 安全機構の役割を要求の冗長化と、両要求間の制約条件で分かりやすく説明できる
- 安全機構に関する要求の導出背景を、安全分析を交えて説明できる
- 安全機構として守る側、安全機構に守られる側といった、要求のグルーピングを明示できる
- 要求のグルーピング間の従属故障分析における、対象や範囲の特定をサポートできる
- エレメントの重み付けが、どの要求の配置によってもたらされたか、どの要求への無干渉要求からもたらされたのかを可視化できる
- システムレベルからハードウェア、ソフトウェアといった実装レベルまで一貫して扱える
- ソフトウェアツール上に実現されるモデリング機能を前提にした文法・記法である

[5] 利害関係者が期待するサービス(補足）機能 とは「もののはたらきのこと。相互に連関し合って全体を構成しているものの各要素や部分が、それぞれ荷っている固有の役割、作用。」である。（広辞苑 第 6 版）

[6] 機能要求を実現する上での制約条件。例えば、独立要求、非干渉要求がある。ただし、独立要求、非干渉要求は、詳細化の過程で機能要求となり得る。

ソフトウェアツールで作成した安全アーキテクチャの仕様情報をシステムのモデルととらえることで、当該ツールは適切な支援を提供することにより、その意味論に基づくアーキテクチャの最適化・合理化のための有効な手段ともなりえる。このことから言語仕様としては多くのモデリング言語同様、ソフトウェアツール上に実現されるモデリング機能を前提にした文法・記法とすることとした。

3.5 構造図の種類の概要

SCDL では要求層、エレメント層、要求層とエレメント層を組み合わせた、各構造図を用いる（表 2、図 3)。各構造図の詳細については、4.4 節にて後述する。

表 2 SCDL 構造図

構造図名	説明
要求構造図	要求間の関係を示す
安全要求構造図	安全要求を含む要求構造図
エレメント構造図	エレメントの構造を示す
要求配置図	要求をエレメントに配置したもの
安全要求配置図	安全要求を含む要求配置図
コンセプト図	すべての要求配置図を一体化した結果を示す
安全コンセプト図	安全要求を含むコンセプト図

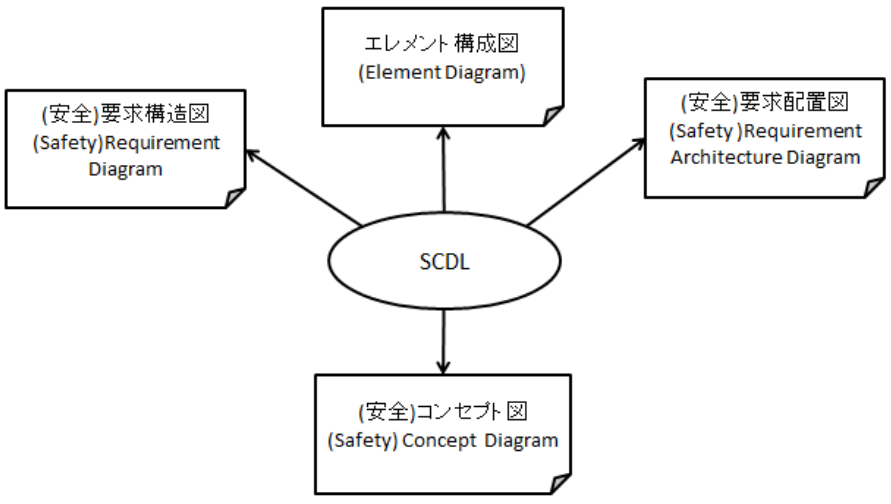


図 3 SCDL 構成図の関係

4 SCDL の基本定義

4.1 概要

本章では、SCDL の定義として、記号の定義、組み合わせた時の意味、構造図の種類の詳細を示す。

- 4.2 節 記号の定義では、要求、要求グループ、要求グループの冗長関係、エレメント、インタラクション、システムバウンダリインタラクションを示す
- 4.3 節 上記の記号を組み合わせた時の意味では、要求のエレメントへの配置、要求グループの括り方、ペアリング時の制約条件導出の仕方、無干渉要求表記、要求間インタラクション線の分岐を示す
- 4.4 節 構造図の種類の詳細では、要求構造図、安全要求構造図、エレメント構造図、要求配置図、安全要求配置図、コンセプト図、安全コンセプト図、各構造図間の関係を示す

4.2 記号の定義

本節にて、SCDL の基本的な表記法に用いる基本要素（要求、エレメント、それらのインタラクション、および無干渉）を定義する。すべての記号の線種・太さ・色（塗りつぶし色、線色、文字色など）は任意とするが、それぞれ記号を区別できるように、利用者がルールを規定すること。

SCDL では以下の要素を定義する。

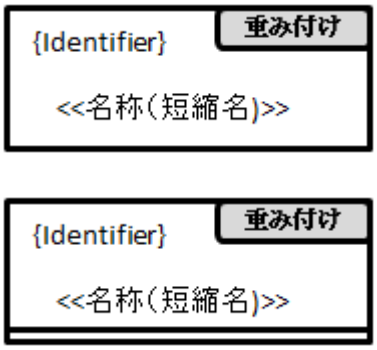
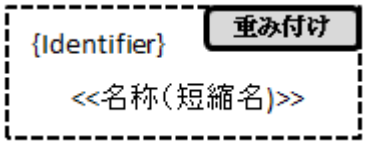
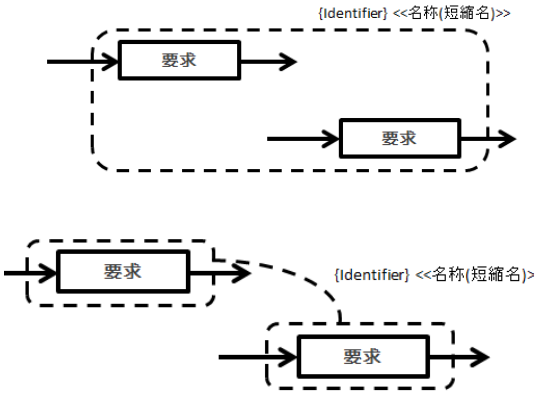
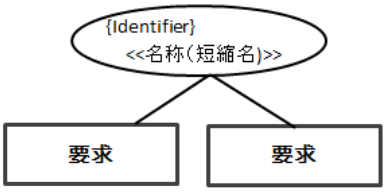
- 要求
- 要求グループ
- エレメント

また以下の要素間の関連を定義する。

- インタラクション
- システムバウンダリインタラクション
- 無干渉線
- 要求グループペアリング線
- 制約条件からの引き出し線

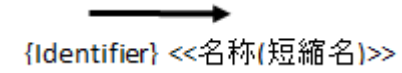
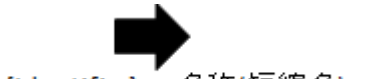
要素の表記を表 3 に示す。表中の{ } <<>> ( )は、項目の記入欄を示し図上での表記は不要である。

表 3

	要求は長方形で表記し、右肩に内包される長方形で重み付けを表記する。 なお、短縮名は名称の短縮名称を意味し、理解を助けるための省略表現である。 要求のエレメントへの配置先が決定されていない、または配置できない場合は、左下図のように要求の下辺を二重線で表記する。
	エレメントは長方形で表記し、右肩に内包される長方形で重み付けを表記する。 エレメントと要求は異なる線種や色で区別できるようにする。
1.  2.  3. 	要求グループは 3 種類の表記を持つ。どの表記を利用するかは任意とする。 1．(Enclosure type)同じグループに属する要求を枠線で囲む。同じグループであるものを線で紐付けてもよい 2．(Balloon type)要求グループを楕円で表記し、同グループに属する要求を線で結ぶ 3．(Tab type)要求の長方形の上辺にタブを表記し、要求が属する要求グループを表記する。名称には同じ要求グループに属することがわかる識別子を設定すること

関連の表記を表 4 に示す。

表 4

	“インタラクション”は２つの要求間に一方向の矢印として表記する。
	“システムバウンダリインタラクション”は２つの要求間に一方向のブロック矢印として表記する。
	“無干渉”は、稲妻型の矢印で表記する。矢印始点はエレメントとし、矢印終点は要求、要求グループ、またはエレメントとする。
	“要求グループペアリング”は、要求グループ間に双方向の破線矢印にて表記する。
	制約条件からの”引き出し”は、両端を菱型端点で表記する。

4.2.1 要求表記

要求は長方形で示す。一部特定の非機能要求を除き、要求は基本的に機能・役割・振る舞いを扱うこととする。

要求を示す長方形内には、要求の識別子として“ID“、“名称（短縮名）“のいずれかを必ず表記し、両方表記してもよい。記載内容の命名則は、SCDL では定義しない。要求を示す長方形の線種は任意であるが、次節にて定義するエレメントを示す長方形とは区別できるようにする。

要求を示す長方形は、重み付けを表記する記入欄を持つ。重み付け記入欄は、要求の右肩を標準位置として、2 辺を要求と内接する小さな長方形とする。重み付けは、各種安全規格が定める規則に従い表記する。ただし、重み付けの付与が行われるまでは、表記されない。

- 規定項目
  - 識別子として“ID“、“名称（短縮名）“のいずれかの表記すること
  - 形状は長方形とすること
  - 重み付けを表記する記入欄を持つこと
  - 重み付け欄は、要求の右肩を標準位置として持ち、重み付けの 2 辺を要求と内接する小さな長方形とすること
  - 重み付けの付与が行われるまでは、重み付けは表記されないこと
  - エレメントを示す長方形とは区別できること
- 任意項目
  - ID、名称の命名規則
  - 重み付けの表記、非表記
  - 線種、太さ、色

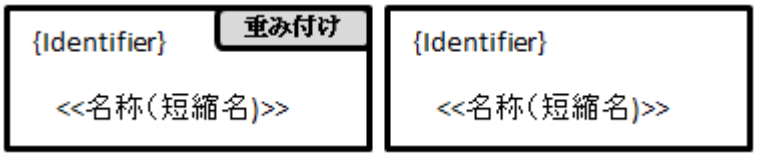


図 4

4.2.2 インタラクション表記

SCDL での要求間の情報・信号・メッセージ等の授受をインタラクションと呼ぶ。インタラクションの代表的な表記を図 5 に示す。

要求を示す長方形は、原則１つ以上の入力と１つの出力で表されるインタラクションを持つ。インタラクションは“矢印(→)“で表し、“矢無し端点“を要求の出し側、“矢有り端点“を要求の受け側の要求に接続する。 “矢印(→)“には、インタラクションの識別子として“ID“、“名称（短縮名）“のいずれか、または両方表記してもよい。

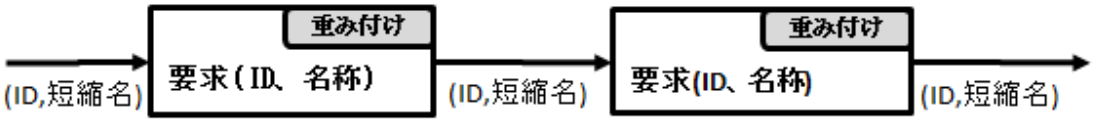


図 5　要求および要求間インタラクションの代表表記

- 規定項目
  - 識別子として“ID“、“名称（短縮名）“のいずれかの表記すること
  - 形状は矢印線とすること
  - 線の両端は要求と接続されること
- 任意項目
  - ID、名称の命名規則
  - 線種、太さ、色

要求は、１つ以上の複数のインタラクション（入力）を持つことができる。

図 6

同じ情報（インタラクション）を下流の２つ以上の要求で共有する場合は、インタラクションを分岐させる方法で表記することができる。

図 7

4.2.3 システムバウンダリインタラクション表記

インタラクションの特殊事例として、記述対象としているシステムの外界から、または外界への働きかけを表すインタラクションを、システムバウンダリインタラクションと呼ぶ。システムバウンダリインタラクションの表記法を図 8 に示す。

システムバウンダリインタラクションは“ブロック矢印“で示し、入力を表現するには“矢側端点“を要求に接続し、“矢無端点“はシステムバウンダリの範囲外の要求と接続する。出力を表現するには“矢無端点“を要求に接続し、“矢側端点“はシステムバウンダリの範囲外の要求と接続する。なお、システムバウンダリの範囲外の要求との接続は省略し、どちらかの端点が要求に接続していない表現としてもよい。接続を省略している場合でも、接続していない端点はシステム外界に配置する。

システムバウンダリインタラクションの識別子として“ID“、“名称（短縮名）“のいずれか、または両方表記してもよい。表記位置は任意とする。

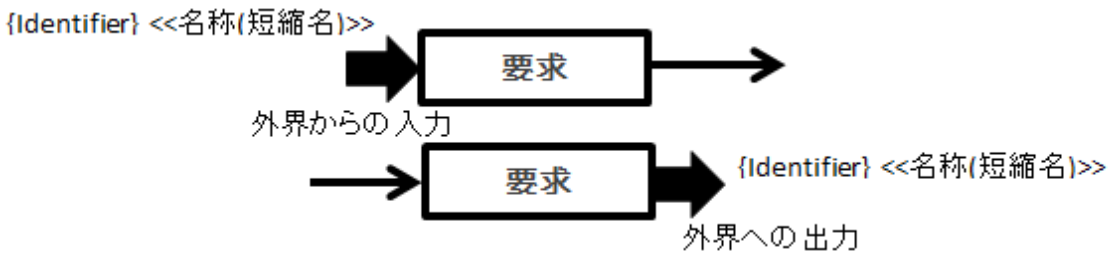


図 8　システムバウンダリインタラクションの表記法

- 規定項目
  - 識別子として“ID“、“名称（短縮名）“のいずれかの表記すること
  - 形状はブロック矢印とすること
  - 線のどちらか一方は要求と接続されること
- 任意項目
  - ID、名称の命名規則
  - 線種、太さ、色
  - システムバウンダリの範囲外の要求との接続

4.2.4 インタラクション表記の禁止事項や例外事項

要求間のインタラクション表記での代表的な禁止事項や例外事項を下記に示す。

1. 要求は、要求がアトミックでなくなるため、２つ以上の出力のインタラクションを持ってはならない

図 9

2. 入力のインタラクションは、合流のロジック自体を要求化しなくてはならないため、合流表記をしてはならない

図 10

3. インタラクションは、アイテム/システム/サブシステムにおいて、一般的に入力部から出力部までつながった経路を持つが、入力のない要求や出力のない要求は例外として存在する（例：初期値設定、乱数発生、リセット等）

図 11　入力を持たない例

図 12　出力を持たない例

4.2.5 エレメント表記

SCDL でのエレメントの代表的な表記を図 13 に示す。

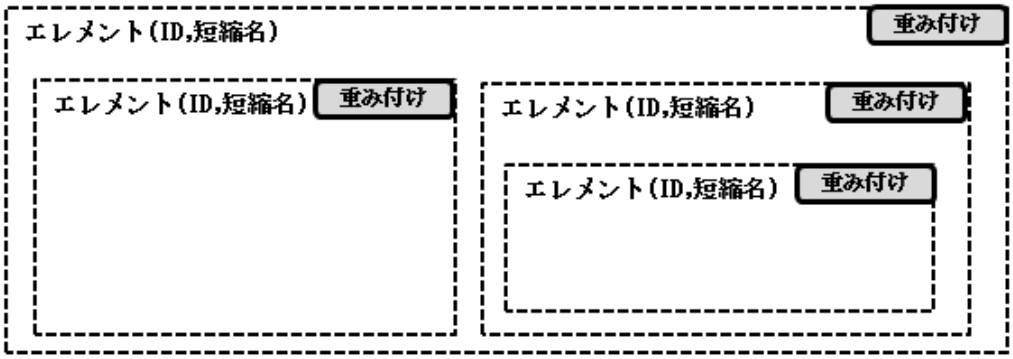


図 13　SCDL でのエレメントの代表表記

エレメントの線種は任意であるが、要求を示す長方形とは区別できるようにする。エレメントはシステム、サブシステム、コンポーネント、ユニット、モジュール、パーツ、回路ブロック等および、それらの包括関係を示す。

包括関係表記として、サブエレメントは親エレメントの内部に隙間を設けて表記する。一番外側のエレメントがシステムバウンダリを示す。

エレメントには、エレメントの識別子として”ID”、”名称（短縮名）”のいずれか、または両方表記してもよい。記載内容の命名則については、SCDL では定義せず任意とする。

エレメントは、重み付けを表記する記入欄を持つ。重み付け記入欄は、エレメントの右肩を標準位置として、2 辺を要求と内接する小さな長方形とする。

エレメント表記の禁止事項を図 14 および下記に示す。

1. エレメント間を線で分割して２つのエレメントとしてはならない
2. エレメントは包括図として表記し、エレメントを部分的に重なる様に表記してはならない
3. 上位エレメントと下位エレメントを内接させてはならない

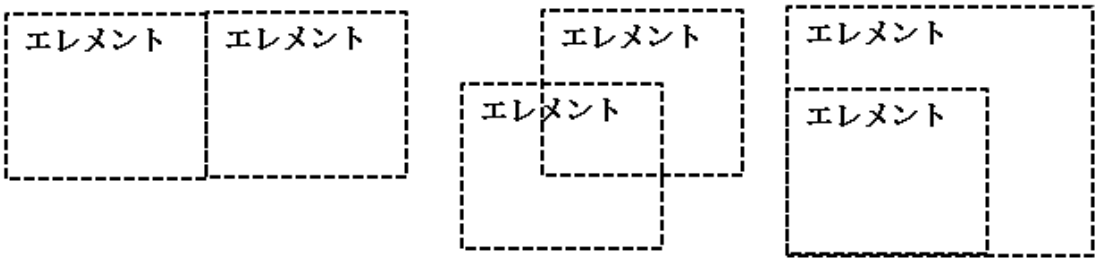


図 14　エレメント表記の禁止例

- 規定項目
  - 識別子として“ID“、“名称（短縮名）“のいずれかの表記すること
  - 形状は長方形とすること
  - 重み付けを表記する記入欄を持つこと
  - 重み付け欄は、要求の右肩を標準位置として持ち、重み付けの 2 辺を要求と内接する小さな長方形とすること
  - 重み付けの付与が行われるまでは、重み付けは表記されないこと
  - 要求を示す長方形とは区別できること
  - エレメントの線は、他のエレメントの線と重ねないこと
- 任意項目
  - ID、名称の命名規則
  - 重み付けの表記、非表記
  - 線種、太さ、色

4.3 表記を組み合わせた時の意味

SCDL は各要素を組み合わせて表記するため、本節にて代表的な表記の組み合わせを示す。

4.3.1 要求のエレメントへの配置

要求はエレメント内に配置される。

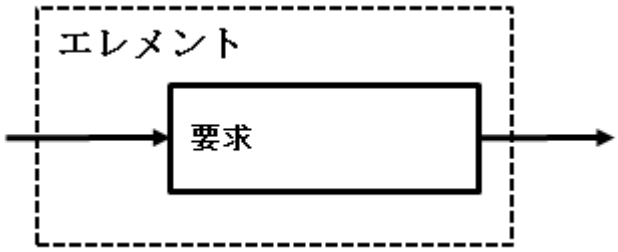


図 15

1. エレメントの境界線とは内接、交差してはならない

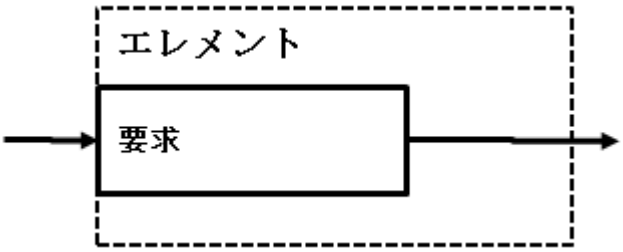


図 16 内接の禁止例

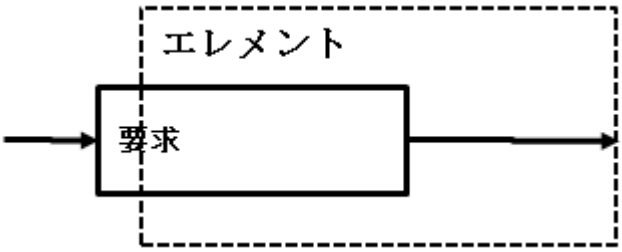


図 17 交差の禁止例

2. エレメントが入れ子状態になる場合にも上位エレメントおよび下位エレメントのどちらの境界線とも要求は交わらない

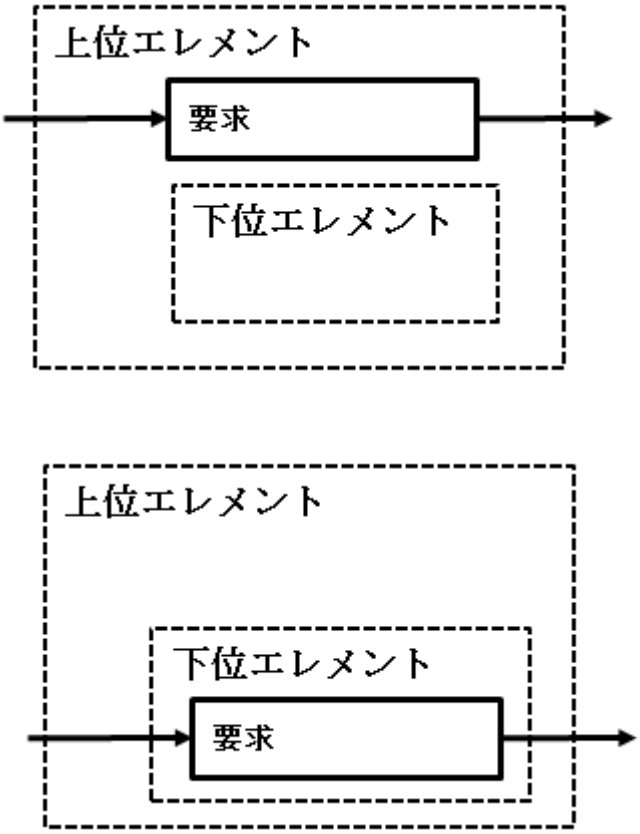


図 18　入れ子状態のエレメントに対する要求の配置例

3. エレメントの重み付けは、要求配置の結果として各種安全規格が定める規則に従う。ただし、重み付けの割付が行われるまでは表記されない

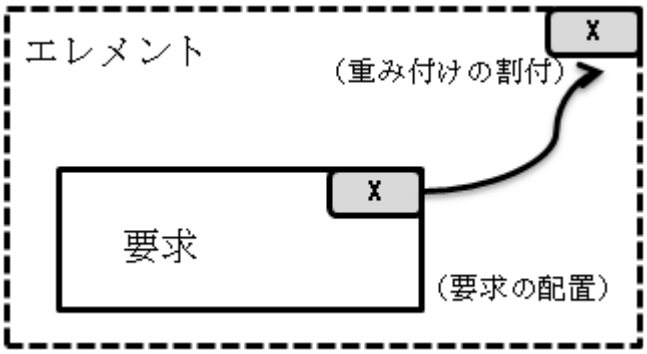


図 19

4.3.2 要求グループの括り方

1. 要求をグルーピングする場合には角丸の長方形、または多角形と、それらを結ぶ接続線で表す。グループは一群の要求グループを示す、ID、 短縮名を与え、どちらかをまたは両方を表記する。
   グルーピング表記に適用する線種、太さ、色はいずれも任意とする

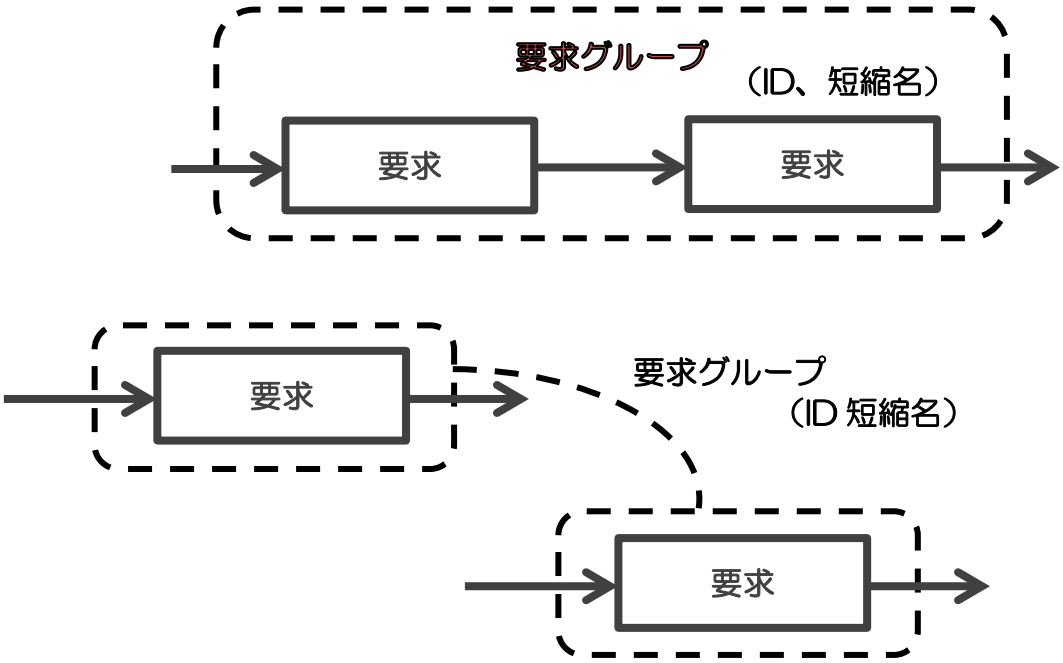


図 20

2. 冗長な要求グループ間のペアリングは双方向矢印で示し、ペアの ID、 短縮名を与え、どちらかをまたは両方を表記する

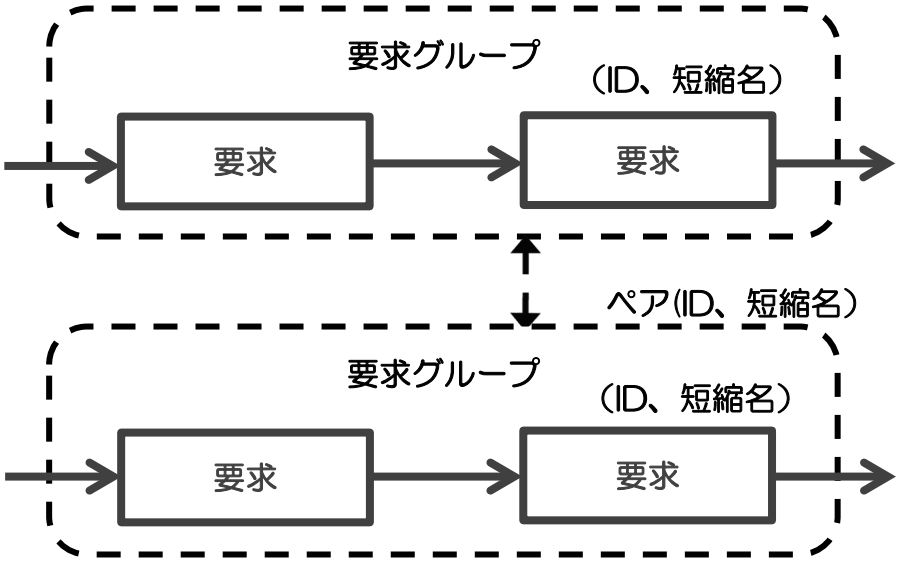


図 21

冗長な要求グループ間のペアリングを示す双方向矢印は、要求グループペアリングと呼称し、要求グループ間に双方向の破線矢印にて表記する。
要求グループペアリングの紐づけ箇所は、要求をグルーピングする枠線に紐づける（図 22）、またはグルーピングの枠線を結ぶ接続線（図 23）に紐づいてよい。

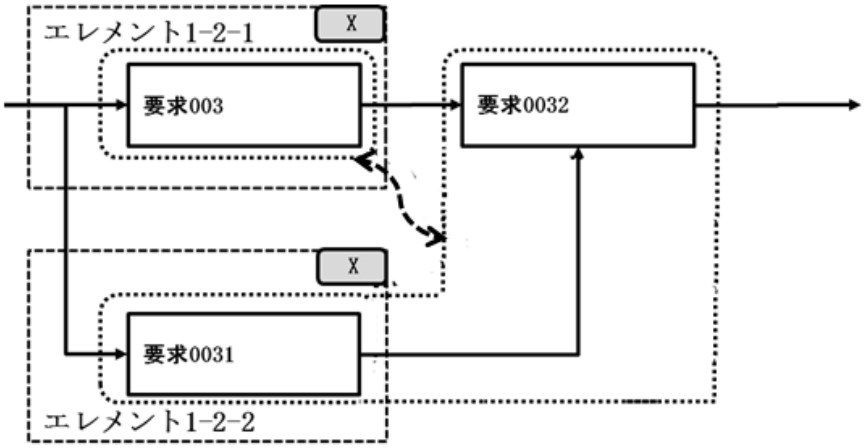


図 22 枠線での要求グループ表記

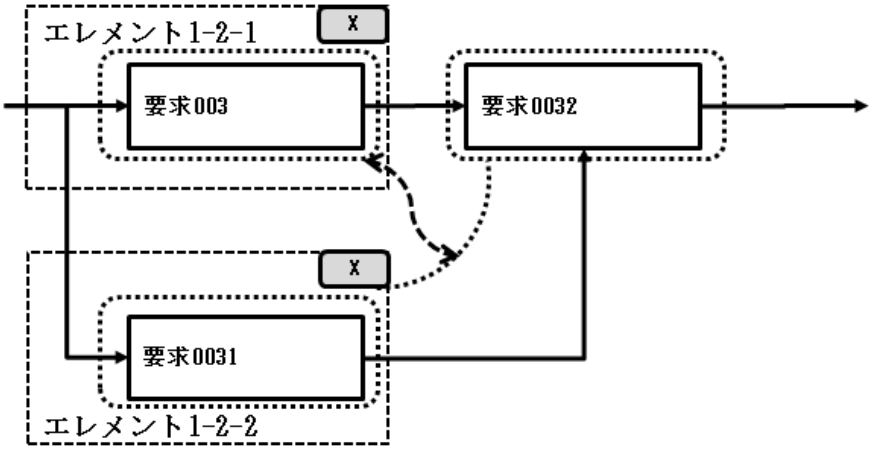


図 23　枠線と接続線での要求グループ表記

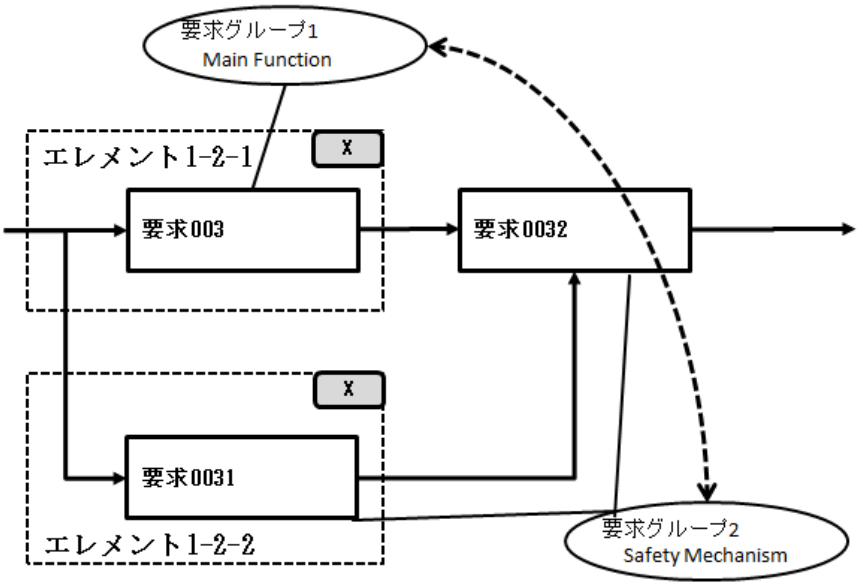


図 24　楕円での要求グループ表記

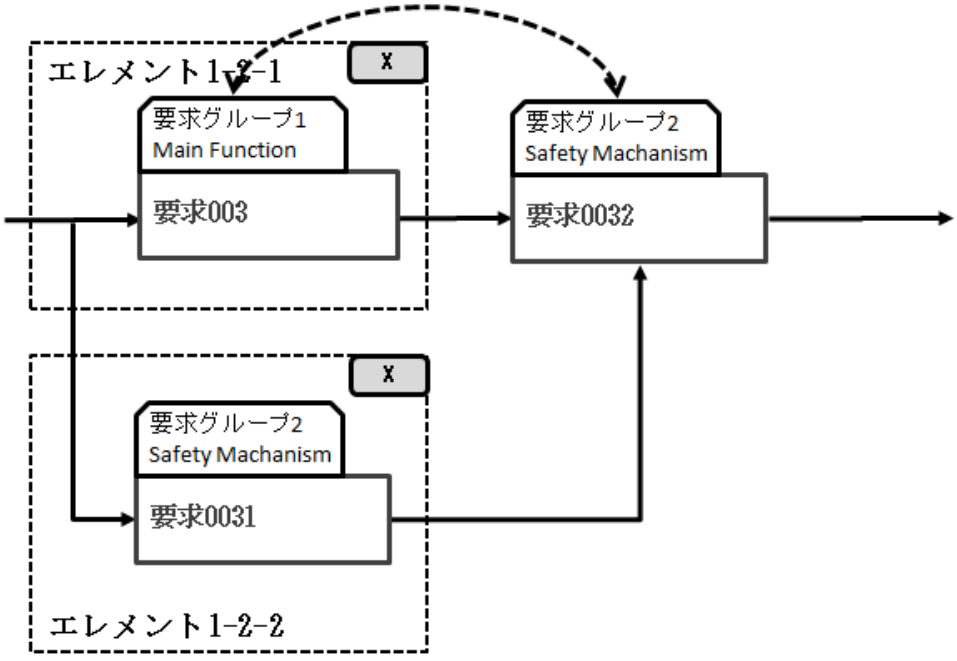


図 25　タブでの要求グループ表記

4.3.3 ペアリング時の制約条件導出の仕方

1. 冗長な要求グループ間のペアリングに制約条件がある場合には、ペアリングを示す双方向矢印からの引き出し線で表記する（制約条件は要求を用いる）

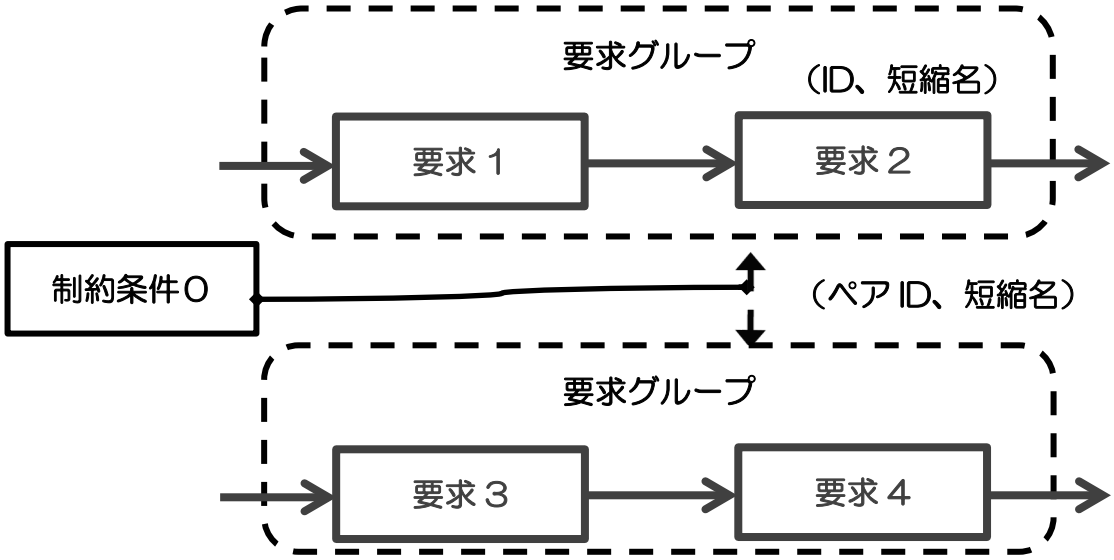


図 26

制約条件のエレメントへの配置先が決定されていない、または配置できない場合は、以下のように要求の下辺を二重線で表記する。

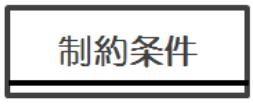


図 27

2. 要求グループ間の制約条件は２つの要求グループに属するそれぞれの要求間の制約条件に分解できる（原則全部の組み合わせを配慮する。図 28 は図 26 の制約条件０を詳細化した場合を示す）

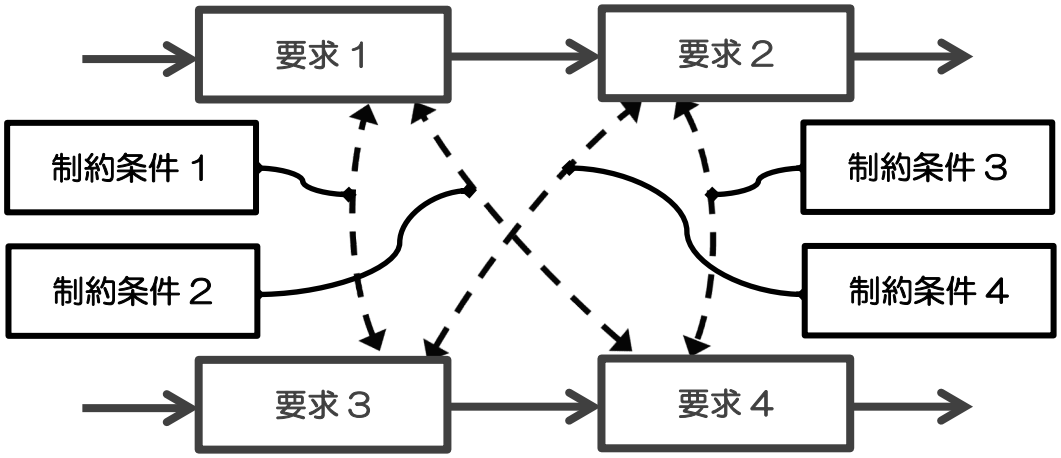


図 28

4.3.4 無干渉要求表記

SCDL での無干渉要求の代表的な表記を図 29、図 30、図 31 に示す。あるエレメントから要求（図 29)、要求グループ（図 30)、および他のエレメントに含まれるすべての要求とエレメント間に”無干渉要求”がある場合（図 31）には、要素間を”稲妻型矢印”にて表記することができる。

あるエレメントから、他のエレメントに含まれるすべての要求への無干渉を表現する場合は、稲妻型矢印終点のエレメントに含まれるすべての要求に向けて、始点エレメントから稲妻型矢印が表記されるべきだが、簡易表現として、始点・終点のエレメント間に稲妻型矢印を表現できる。なお始点はエレメントの内部、外周、終点は要求、要求グループ、エレメントの内部、外周とする。

無干渉要求の制約条件は”稲妻型矢印”からの引き出し線でつながった要求（下図、制約条件 001）に表記する。この場合、制約条件を記載した要求は、無干渉要求がある上位エレメントに配置される。

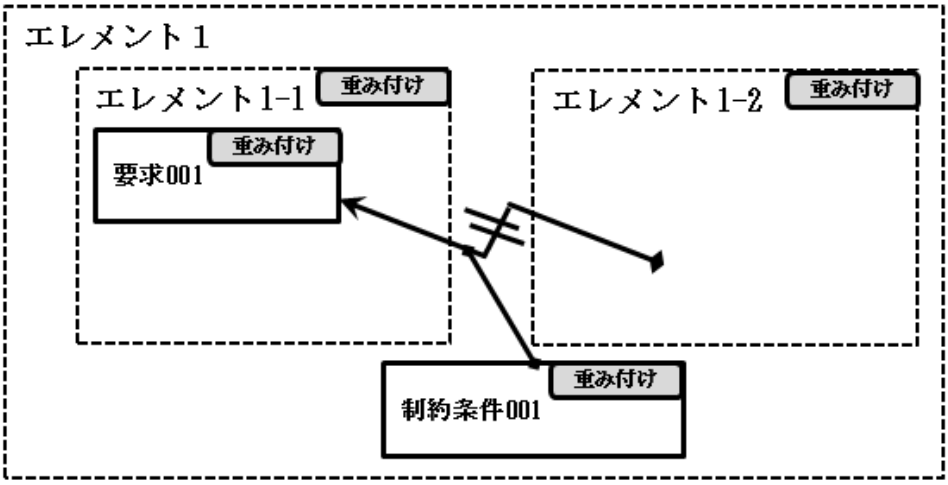


図 29 エレメント―要求間の無干渉の表記例

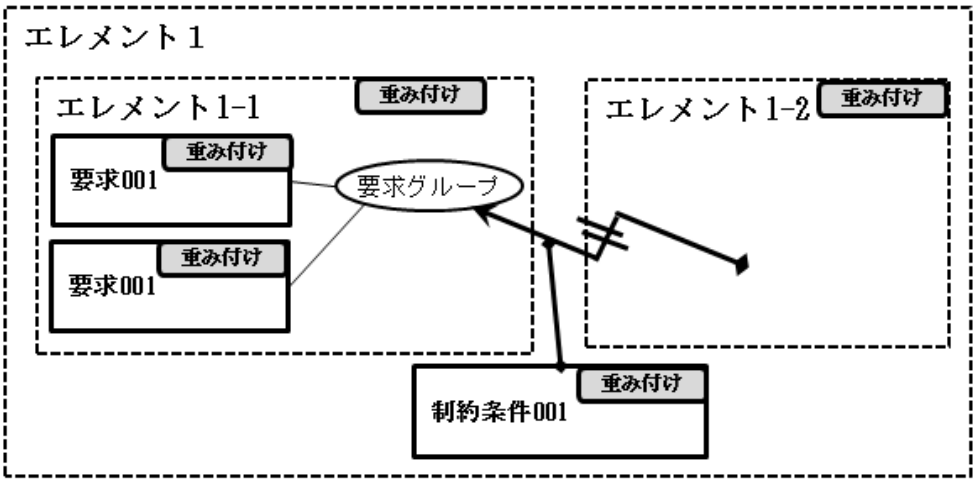


図 30　エレメント―要求グループ間の無干渉の表記例

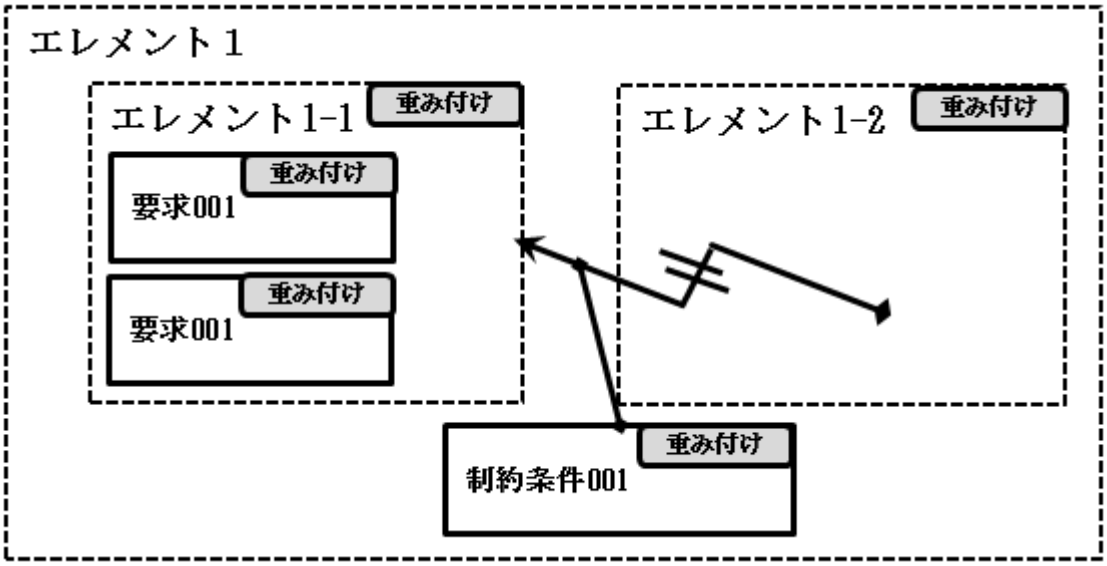


図 31　エレメント―エレメント間の無干渉の表記例

4.3.5 要求間インタラクション線の分岐

エレメントに配置された後の要求間のインタラクション線の注意事項を下記に示す。

エレメントに配置された後の要求間のインタラクション線の分岐点位置は、インターフェース定義（後述）されるまでは意味を持たないため、図 32 の左右に違いはない。その際の接続点を図 33 の左右のように、白丸で表記してもよい。

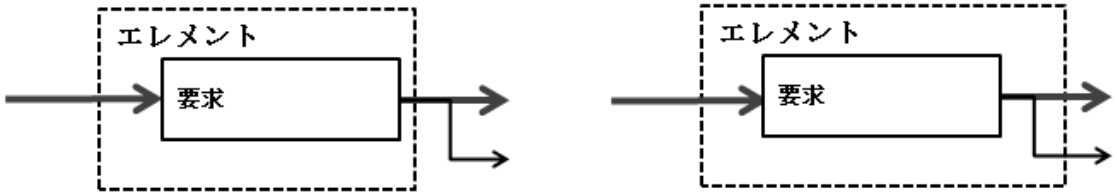


図 32

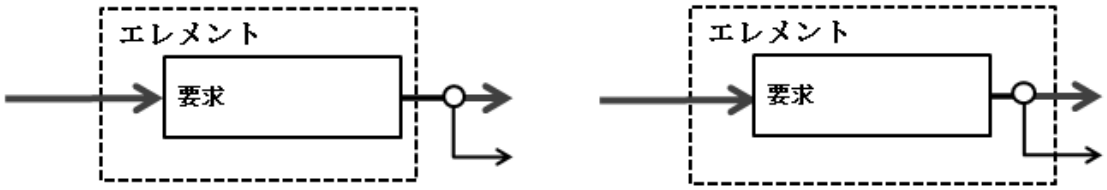


図 33

安全アーキテクチャを検討する上で、要求間のインタラクション線の分岐点位置を規定したい場合は、インタラクション線上に分岐点表記”黒丸”を表記し規定する。エレメントの外側で分岐する場合は図 34 の左図のように表記し、エレメントの内部で分岐する場合は、図 34 の右図のように表記する。

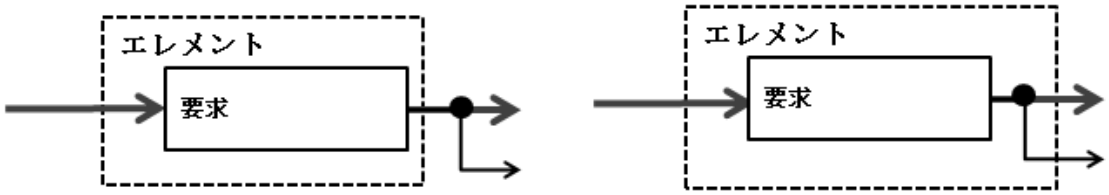


図 34

4.4 構造図の種類の詳細

本節は、SCDL を理解する上での参考として、SCDL を用いた安全設計の手順例と、その結果得られる各種構造図について説明する。本章の内容は SCDL 文法には含まれない。

各構造図の表記には要求層、エレメント層、要求層とエレメント層を組み合わせた各構造図を用いる。要求層では要求間の関係を要求構造図として定義する。エレメント層ではエレメントの階層構造の関係をエレメント構成図として定義する。

要求層とエレメント層の組み合わせとして、要求とエレメントの配置関係を要求配置図として定義する。言い換えると、要求配置図とは要求構造図とエレメント構造図を重ね合わせた構造図である。コンセプト図とは、すべての要求配置図を一体化した結果を示す構造図である。

4.4.1 要求構造図

アーキテクチャの主要なトピックの１つは要求の仕様と構造である。要求構造図（図 35）では、機能要求の仕様と構造を示す。要求構造図は要求記述にて表現し、要求間のインタラクションを明確にする。

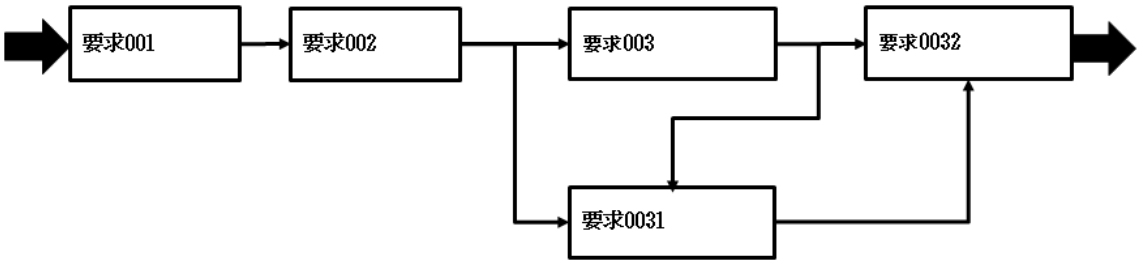


図 35 要求構造図

4.4.2 安全要求構造図

安全要求構造図（図 36）では、安全アーキテクチャにおける安全要求の仕様と構造として、多重化や検出と処置などの機能的／論理的な構造を論じる。SCDL 上はあくまで要求層の議論であるが、表現は一般的な機能ブロックダイアグラムの部類といえ、実運用上は安全機構毎に提示され議論されることが推奨される。時には、独立した冗長設計を適用して要求間の重み付けの分配や制約条件の導出を論じることも求められる。
安全に関わる要求を表現する場合は、安全要求構造図として示す。

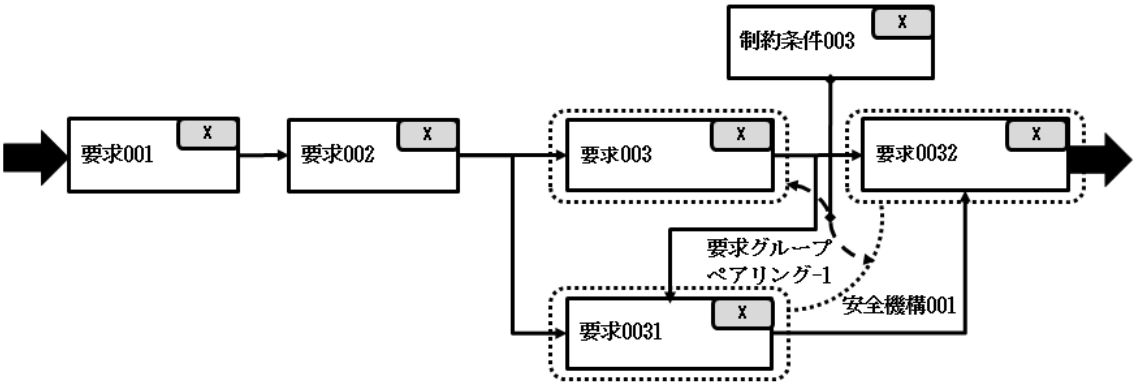


図 36 安全要求構造図

4.4.3 エレメント構造図

エレメント構造図（図 37）では、要求の受け皿となるエレメント層の構造を論じる。エレメントはシステムを構成する部品（コントローラ、マイクロコンピュータ、組み込みソフトウェア等）などが該当する。エレメント構造図では、エレメントの階層構造を包含図にて明確にする。要求の配置と重み付けの波及範囲を可視化する際の土台となり、要求構造図もしくは安全要求構造図と組み合わせて要求配置図/安全要求配置図を表現する際に使用する。

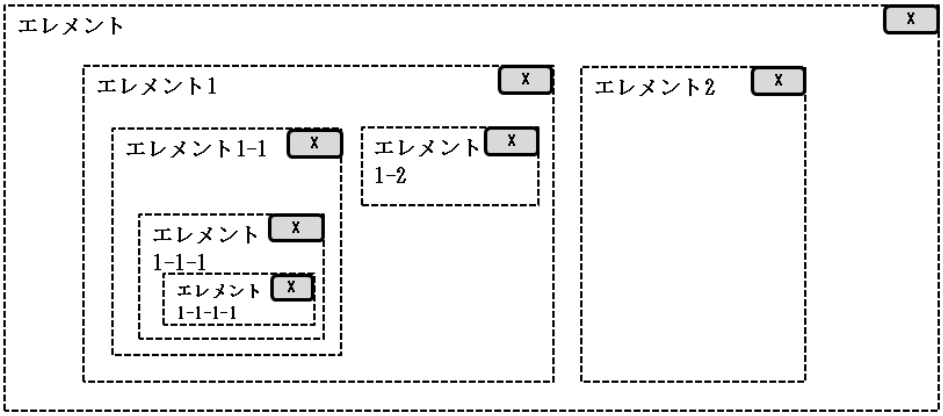


図 37 エレメント構造図

4.4.4 要求配置図

要求配置図（図 39）は要求構造図（図 35）とエレメント構造図（図 37）を重ね合わせて表現し、要求をどのエレメントへ配置するかを明確にする。

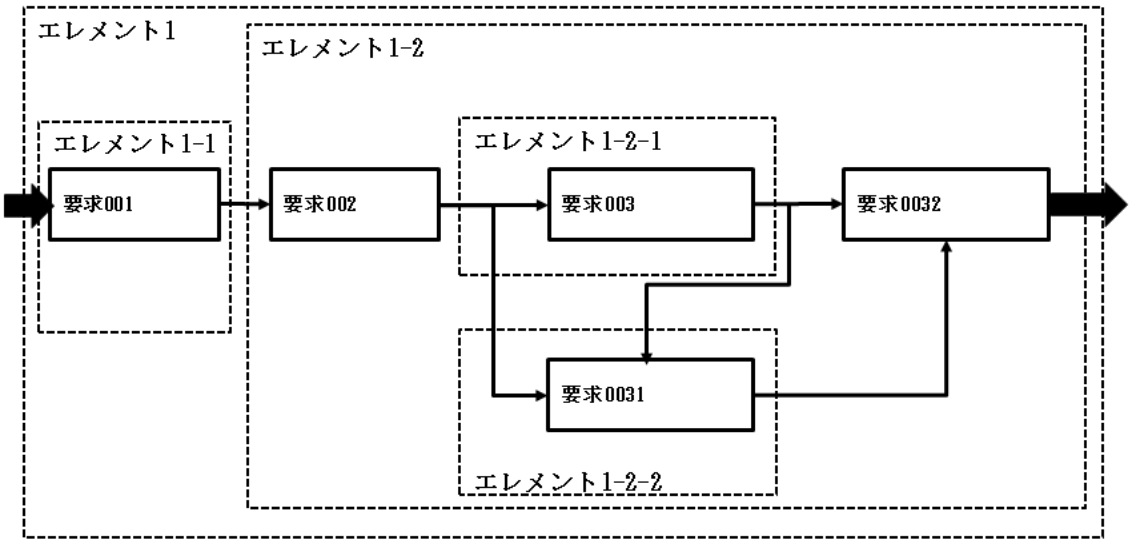


図 38 要求配置図

4.4.5 安全要求配置図

安全要求配置図（図 39）は安全要求構造図（図 36）とエレメント構造図（図 37）の重ね書きによって得られる。安全要求のエレメントへの配置や、エレメント間での冗長関係と独立要求を明確にする。この安全要求のエレメントへの配置によりエレメントが受け取る暫定的な重み付けを論じることも可能となる。

安全要求配置図は安全機構毎に表現することを推奨する。

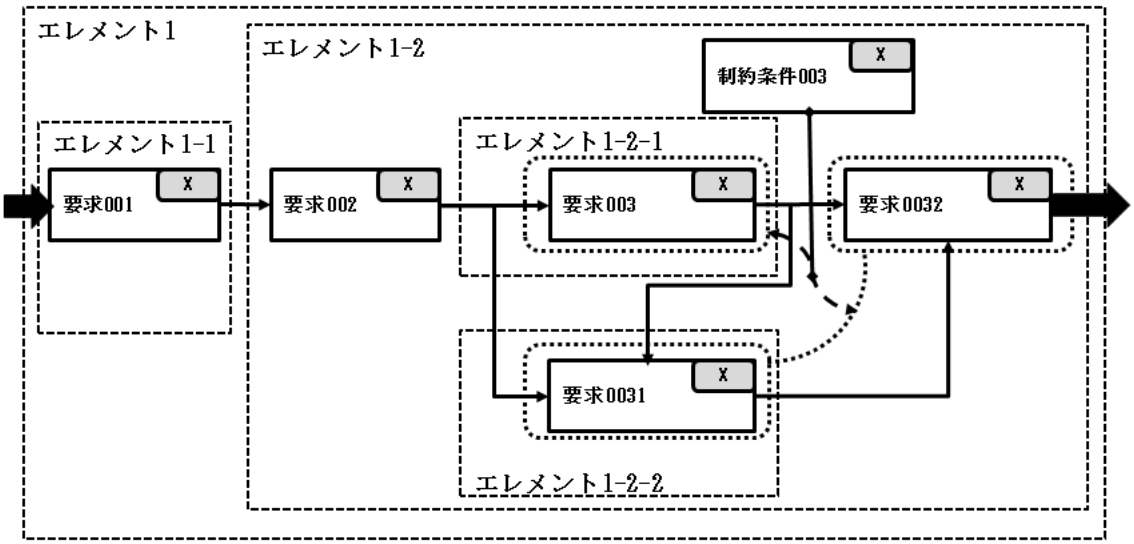


図 39 安全要求配置図

4.4.6 コンセプト図

コンセプト図（図 40）は、設計者の関心がある観点毎に作成された、いくつかの要求配置図を一体化した構成図である。コンセプト図は、要求のエレメント上への最終配置結果を明確にする。要求構造図で表現していた要求の構造を、要求配置図で組み合わせたエレメント構造図のエレメントへ展開して再配置を施した構造図である。

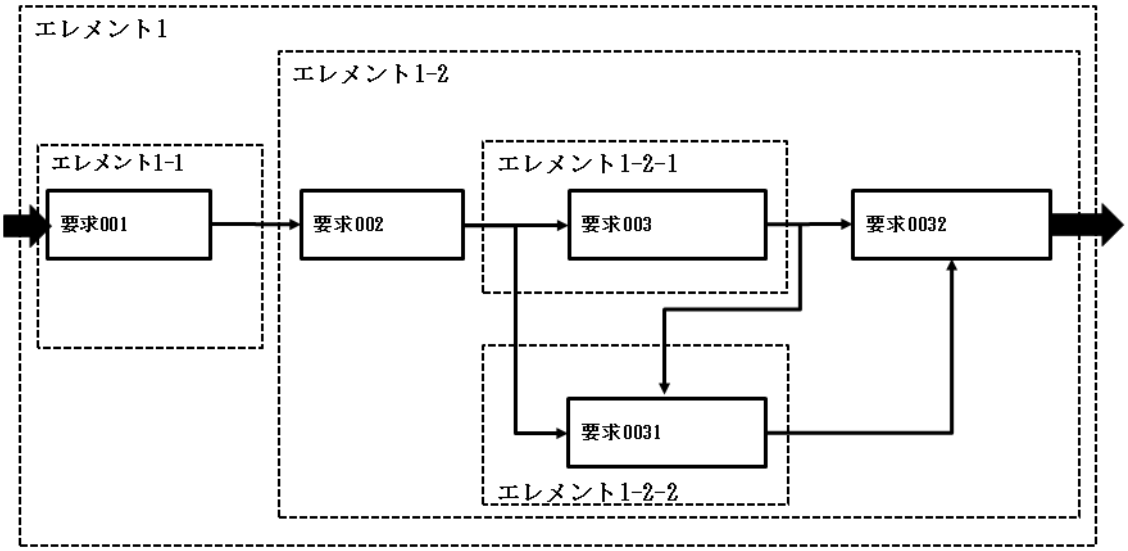


図 40 コンセプト図

4.4.7 安全コンセプト図

安全コンセプト図（図 41）は、複数の安全要求配置図で示される安全機構に関する安全要求をエレメントへ配置した構成図である。安全コンセプト図は、重み付けのエレメント上への最終配置結果を明確にする。安全要求構造図で表現していた重み付けを、安全要求配置図で組み合わせたエレメント構造図のエレメントへ展開して再配置を施した構造図である。

安全コンセプト図では、各エレメントの最終的な重み付けや同居するエレメントの重み付けに波及に関する無干渉要求などについても論じることができる。安全機構の合理化、最適化(複数の安全機構の競合解消など)の作業もこの段階で行う。安全コンセプト図は重み付けのエレメント上への最終配置結果を明確にする。要求構造図で表現していた重み付けを、安全要求配置図で組み合わせたエレメント構造図のエレメントへ展開して再配置を施した構造図である。

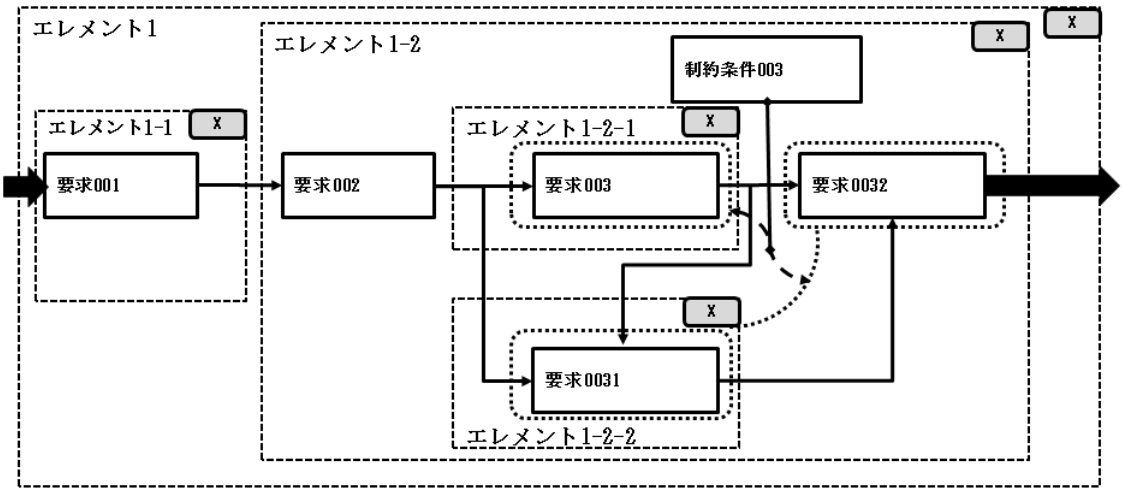


図 41 安全コンセプト図

5 SCDL の拡張定義

5.1 概要

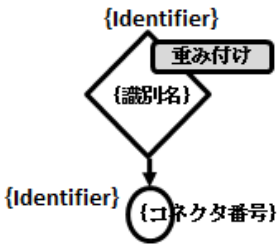
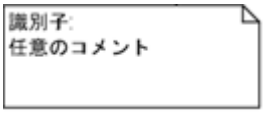
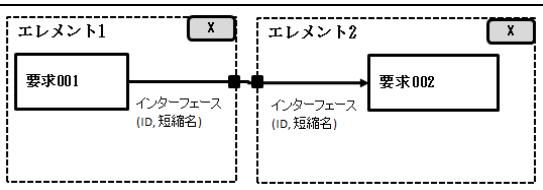
これまでの章では、安全アーキテクチャを表記するための基本的な記法や文法を定義している。これらの記法および文法に従うことで、要求構造図や、安全コンセプト図を作成することが可能となる。本章では、SCDL の拡張的な表記法に用いる要素を定義する。

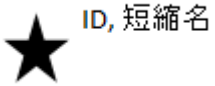
SCDL では、以下の表記を拡張定義する。

- リソース共有
- 分岐コネクタ
- コンストレインツ
- インターフェース
- 外部プラント
- アザーテクノロジーリンク

各表記を以下に示す。表中の{ } << >> ( )は、項目の記入欄を示し図上での表記は不要である。

表 5

	リソース共有はリソースを共有する機能ブロックを、楕円または円で接続させ表記する
	分岐コネクタは分岐判定を菱形で表記し、またコネクタを円形で表記し、コネクタ番号を円形内に表記する
	コンストレインツは識別子とコメント等を表記する
	インターフェースは要求間インタラクションがエレメント境界線をまたぐ箇所に、■を用いて表記する

	外部プラントは星形を用いて表記する
	アザーテクノロジーリンクはシステムバウンダリインタラクション上に重なるように黒丸を置き、リンクしている黒丸を線にて結合し表記する

5.2 リソース共有表記

直接のインタラクションはないが、診断目的などでリソースを共有する場合の要求間の関係を以下の表記とする。 共有リソースは識別子として”ID”、”名称（短縮名）”のいずれか、または両方表記してもよい。詳細定義などを付与してもよい。

5.2.1 記法

リソース共有する要求を、下図に示すように、楕円または円で接続させる。楕円または円の中に、リソース ID を表記する。

図 42

5.2.2 使用例

識別子(ID)	要求
要求 003	XX 演算をする
要求 004	XX 演算と共通のリソースを使う例題演算を行う
要求 005	例題を出す
要求 006	例題演算の結果を検証する

識別子(ID)	詳細定義
RSC001	共通メモリ

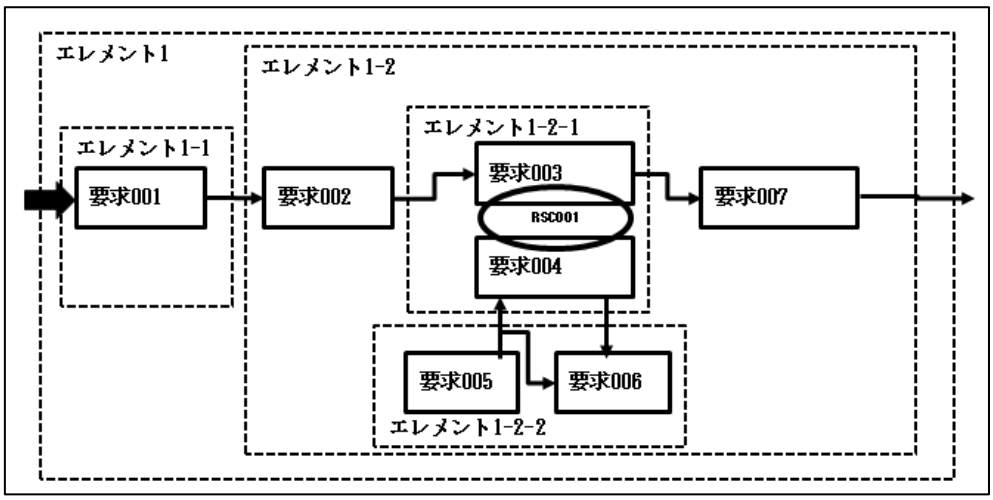


図 43

5.3 分岐コネクタ表記

条件判定や状態遷移により対となる要求構造図をグループ化する際の分岐判定を菱形、コネクタを円形で表記し、コネクタ番号を円形内に表記する。

識別子として”ID”、”名称（短縮名）”のいずれか、または両方表記してもよい。分岐判定詳細等を付与してもよい。重み付けを表記する場合には、分岐コネクタの右上辺上を標準位置とする小さい長方形を表記する。状態遷移図等を併用して遷移条件や状態を定義してもよい。分岐コネクタ表記の範囲を明確にするため、コンストレインツ記法を用いて補足してもよい。

5.3.1 記法

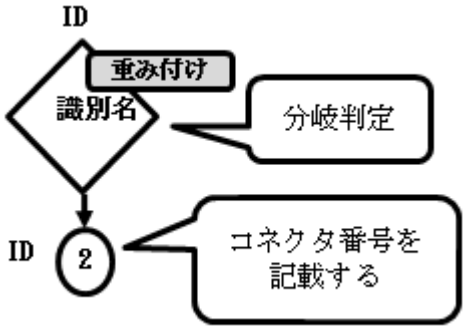


図 44

5.3.2 使用例

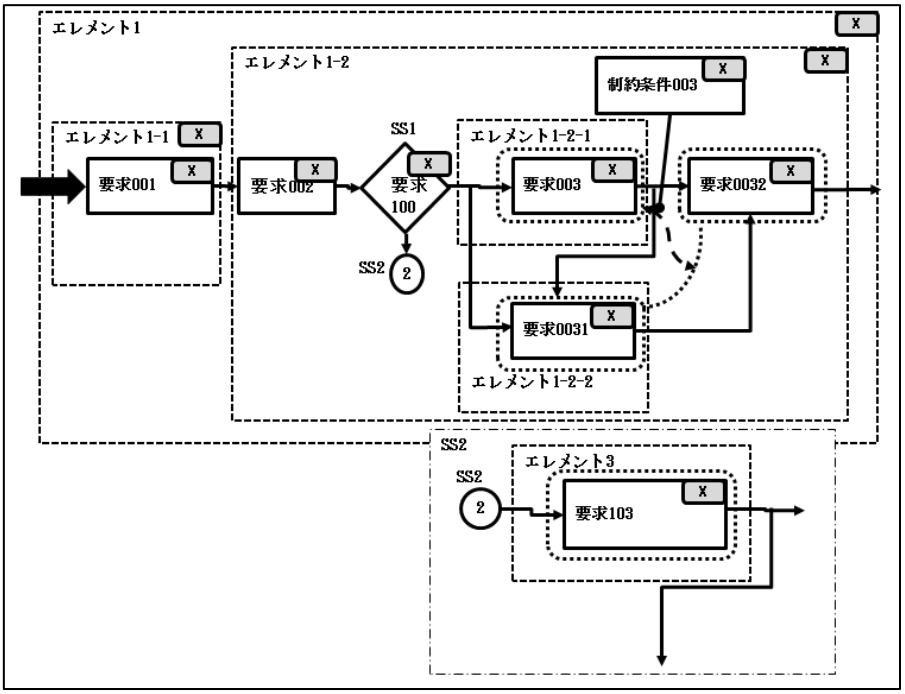


図 45

5.4 コンストレインツ表記

顧客要望、実装要件、方式指定等をコメントとして任意に表記する。識別子として”ID”、”名称（短縮名）”のいずれか、または両方表記してもよい。詳細定義等を付与してもよい。

5.4.1 記法

識別子とコメント等を図 46 のように表記する。エレメント、要求などの SCDL の要素を特定するために、ノートと要素を線でつなげてもよい。

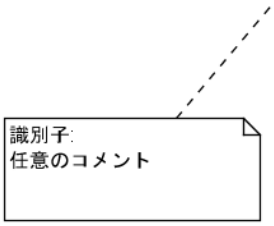


図 46

5.4.2 使用例

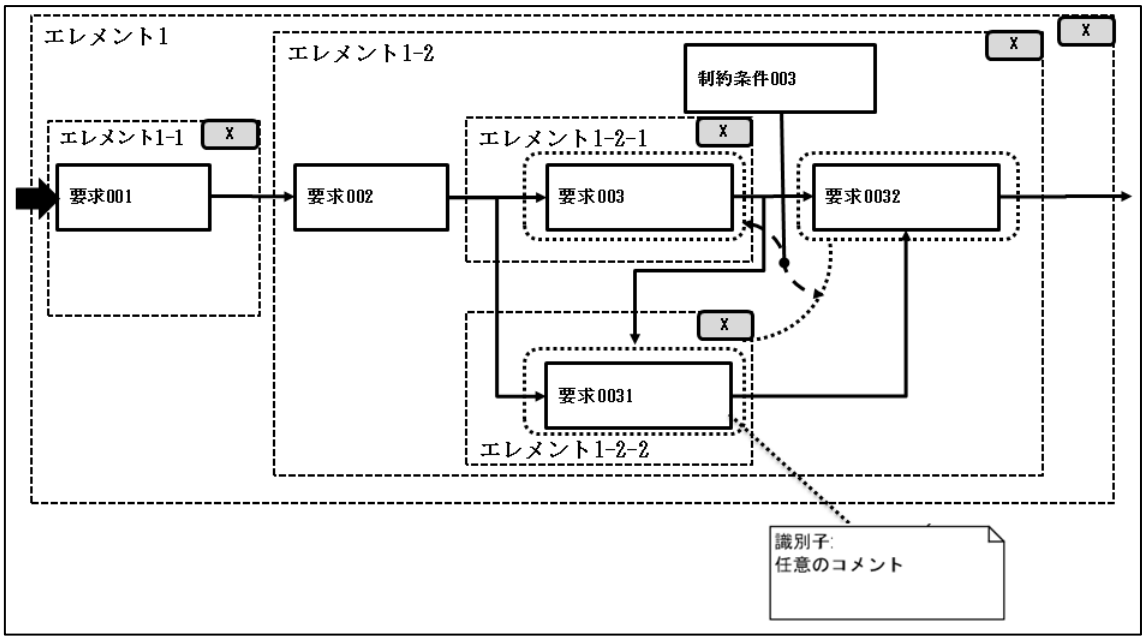


図 47

5.5 インターフェース(I/F)表記

要求間インタラクションがエレメント境界線をまたぐ箇所に、インターフェース仕様を定義したい場合は、■を用いて表記する。ID、短縮名を付与し、表形式等のインターフェース仕様定義と併用できる。

5.5.1 記法

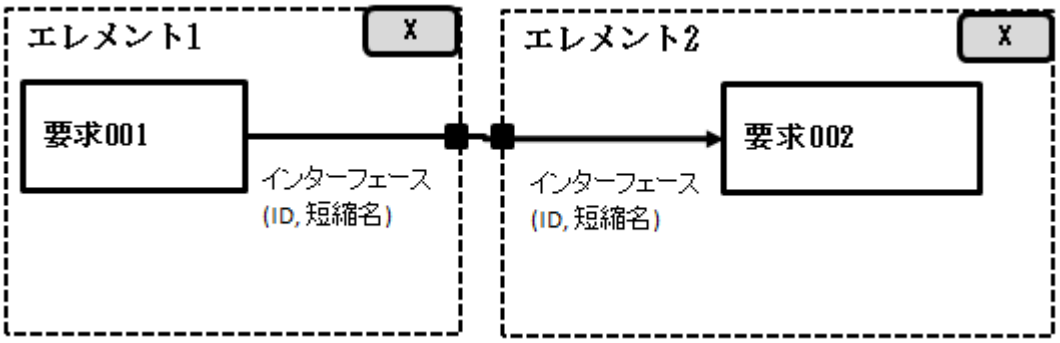


図 48

5.5.2 使用例

ID	説明要求	分類	属するエレメント
IF001	センサ出力	シリアル通信	エレメント 1-1
IF002	センサ入力	シリアル通信	エレメント 1-2

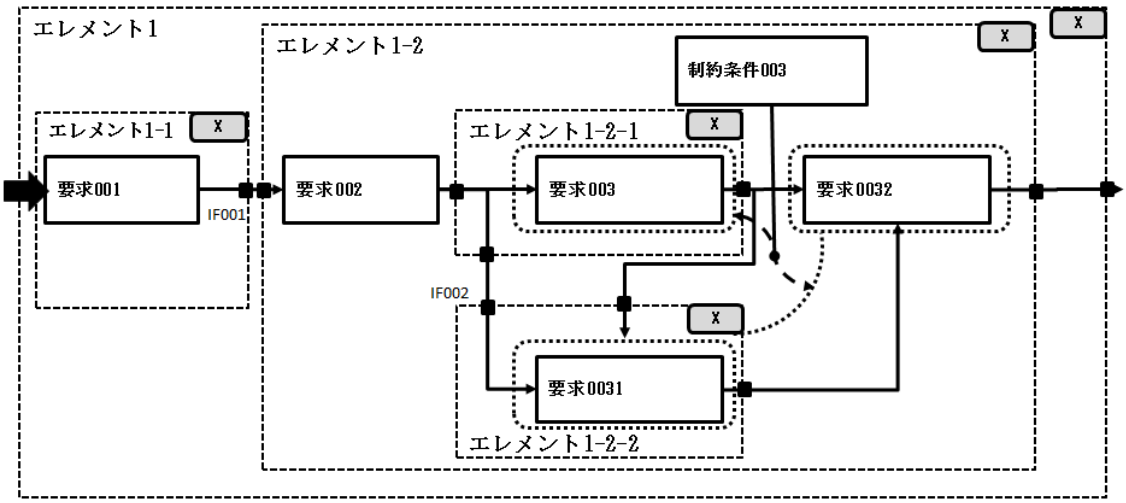


図 49

5.6 外部プラント

記述対象としたシステムの外側での接続やインタラクション、 因果関係は星形を用いて表現する（制御結果をセンサでモニタする、あるいは制御結果が挙動に反映され結果がセンシングされる場合など)。

識別子、短縮名を付与したり、特性等詳細な定義、記述、モデルを関連付けたりしてもよい。

5.6.1 記法

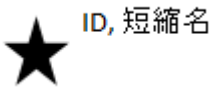


図 50

5.6.2 使用例

下図では、エレメント 1-4 によって、制御結果が制御対象の挙動に反映され、その挙動が、エレメント 1-2 にてセンシングされていることを示している。

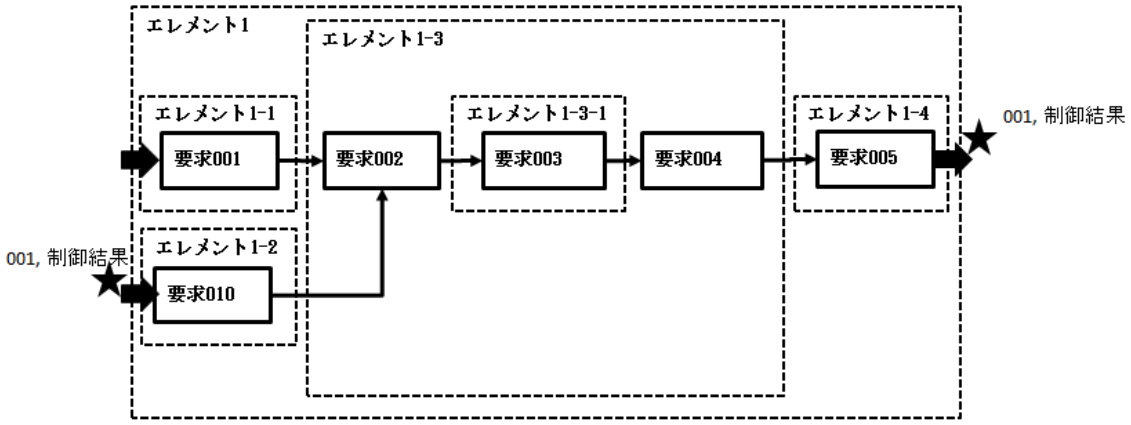


図 51

5.7 アザーテクノロジーリンク

機械構造などによりリンクしている入力などの表記法を以下とする。必要に応じて識別子、短縮名を付与し、詳細定義などを付与できる。

5.7.1 記法

システムバウンダリインタラクションを表す矢印記号上に重なるように黒丸を置き、リンクしている黒丸を線にて結合する。

図 52

5.7.2 使用例

センサが機械的にリンクされている場合の事例を、図 53 に示す。

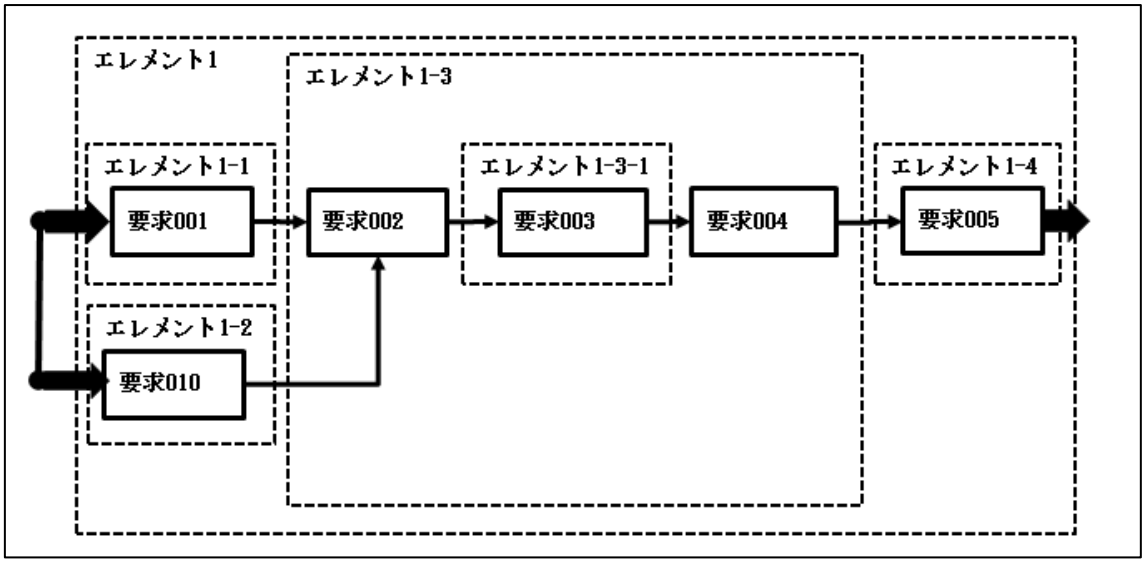


図 53

5.8 ペアリング時の制約条件導出の仕方

要求グループと制約条件の表記方法については、以下の表記も利用できる。

5.8.1 要求グルーピングと冗長をバルーンで表記する場合

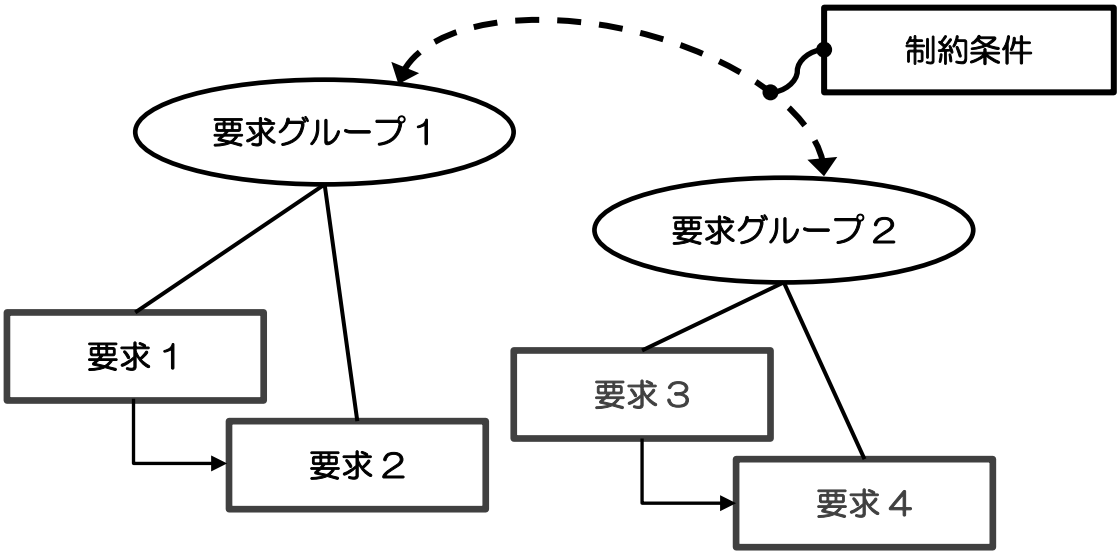


図 54

5.8.2 要求グルーピングと冗長をタグで表記する場合

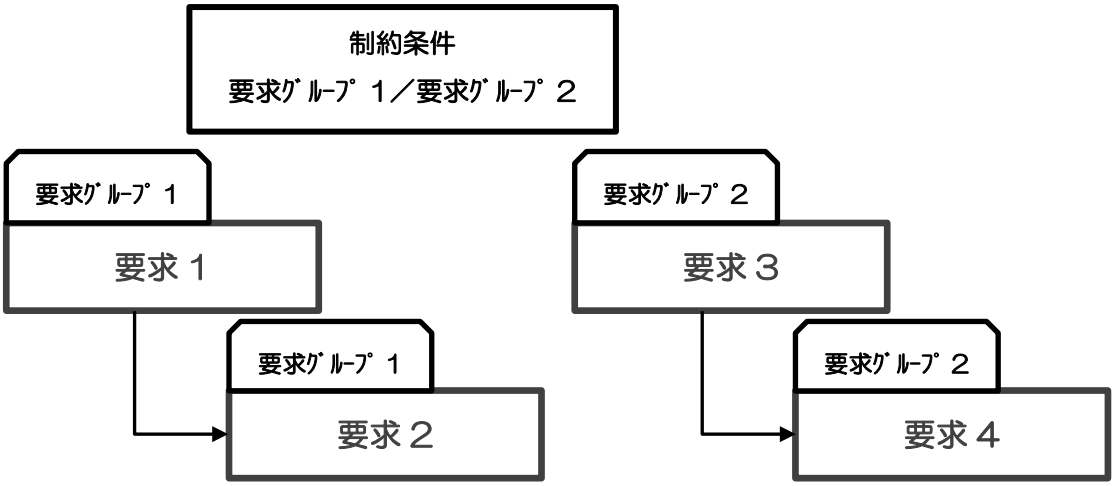


図 55

6 用語、略語集

6.1 用語集

用語（日本語）	用語（英語）	用語の説明
割付	Assignment	要求の重み付けのエレメントに対する付与
エレメント	Element	システムまたはシステムの構成単位
無干渉	Freedom from interference	2 つ以上のエレメント間において安全要求の侵害につながる可能性のあるカスケード故障が存在しない状態
機能要求	Functional requirement	仕様書の中では、利害関係者が期待するサービス。(補足) 機能 とは「もののはたらきのこと。相互に連関し合って全体を構成しているものの各要素や部分が、それぞれ荷っている固有の役割、作用。」である。（広辞苑 第 6 版）
機能安全	Functional safety	機能による安全、機能の安全
独立要求	Independence requirement	複数の要求間に共通原因による、それらの同時侵害がないことを意味する要求
意図機能	Intended function	エレメントに対して指定される、安全機構以外の振る舞い
非機能要求	Non-functional requirement	仕様書の中では、機能要求に対して、機能要求を実現する上での制約条件。例えば、独立要求、無干渉要求がある。ただし、独立要求、無干渉要求は、詳細化の過程で機能要求となり得る。
安全機構	Safety mechanism	安全要求の達成のために、主に電気電子技術で実装される技術的な解決策
安全要求	Safety requirement	エレメントの安全に関わる役割、機能、振る舞い
重み付け	Weighting	安全上の重要さの度合い
アザーテクノロジー	Other Technology	電気電子分野以外の技術分野

6.2 略語集

略語	正式名称
ADL	Architecture Description Language
ASIL	Automotive Safety Integrity Level
DFD	Data Flow Diagram
FBD	Function Block Diagram
UML	Unified Modeling Language
SysML	Systems Modeling Language
SCDL	Safety Concept Description Language
SCN-SG	Safety Concept Notation - Study Group

7 参考文献

- ISO 26262：2011. Road Vehicles -- Functional Safety. ISO Standard.
- ISO 26262：2011. Road Vehicles -- Functional Safety. ISO Standard. 英和対訳版
- 新村出(2008)「広辞苑 第6版」岩波書店
- **図** 78 調達のプロセス：坂口孝則，調達・購買の教科書，日刊工業新聞，2013，pp3

附属書 A ユースケース（自動車分野への適用事例）

A.1 はじめに

SCDL は特定のドメインや特定の開発工程への適用に限られたものではなく、安全設計の検討や論証を必要とするドメインの各開発工程において、アーキテクチャ視点で安全設計を捉えることができるように提供された表記法である。したがって、SCDL の適用範囲や用途は広く、適用の仕方は使用者の意思に委ねられているが、実際に適用する使用者がどの開発工程で、どのように使うのが有効であるかを知っておくことは、SCDL を効果的に使うための指標となる。

本章では、自動車分野の機能安全規格（ISO 26262：2011）を参照した開発フローに適用した場合の SCDL のユースケースを示す。

A.2 自動車機能安全規格の安全コンセプトにおける論証

ISO 26262：2011 を参照した安全活動による開発フローの一例を図 56 に示す。

図 56 内のゴールコンセプトやハードウェア安全コンセプト、ソフトウェア安全コンセプトについては、ISO 26262：2011 に定義されている用語ではないが、SCDL のユースケースにおける説明のために定義した語句である（表 6）。

尚、本章において図中に示される(3-5)は ISO 26262：2011 Part 3 Clause 5 を表す。

表 6

ゴール コンセプト	アイテムの安全目標を導出するための一連のアクティビティを実施した結果としての作業成果物の総称とする
機能 安全コンセプト	ISO 26262-1:2011 英和対訳版には「機能安全要求の仕様、関連情報、アーキテクチャエレメントへの配置及び安全目標達成に必要な相互作用」と定義されているが、他のコンセプトと整合させるため、本章では安全目標を達成するためのアーキテクチャおよび機能安全要求を開発する一連のアクティビティを実施した結果としての作業成果物の総称とする
技術 安全コンセプト	ISO 26262-1:2011 英和対訳版には「技術安全要求の仕様と、それらのシステム設計によって実装するシステムエレメントへの配置」と定義されているが、他のコンセプトと整合させるため、本章では機能安全要求を達成するためのアーキテクチャおよび技術安全要求を開発（ハードウェアとソフトウェアへの分配を含む）する一連のアクティビティを実施した結果としての作業成果物の総称とする
ハードウェア 安全コンセプト	ハードウェア安全要求からハードウェアを開発する一連のアクティビティを実施した結果としての作業成果物の総称とする
ソフトウェア 安全コンセプト	ソフトウェア安全要求からソフトウェアを開発する一連のアクティビティを実施した結果としての作業成果物の総称とする

安全設計の確からしさを論証するためには、システムにおける要求とアーキテクチャ構成要素の関係が曖昧ではなく、分かり易く階層的に表現されていることが必要と考えられる。SCDL を用いれば、図 56 に示すゴールコンセプト、機能安全コンセプト、技術安全コンセプト、ハードウェア安全コンセプト、ソフトウェア安全コンセプトのいずれの段階（フェーズ）においても安全アーキテクチャを描くことで各階層の要求とアーキテクチャ構成要素の関係を明確に表現でき、安全設計の論証に役立てることができる。

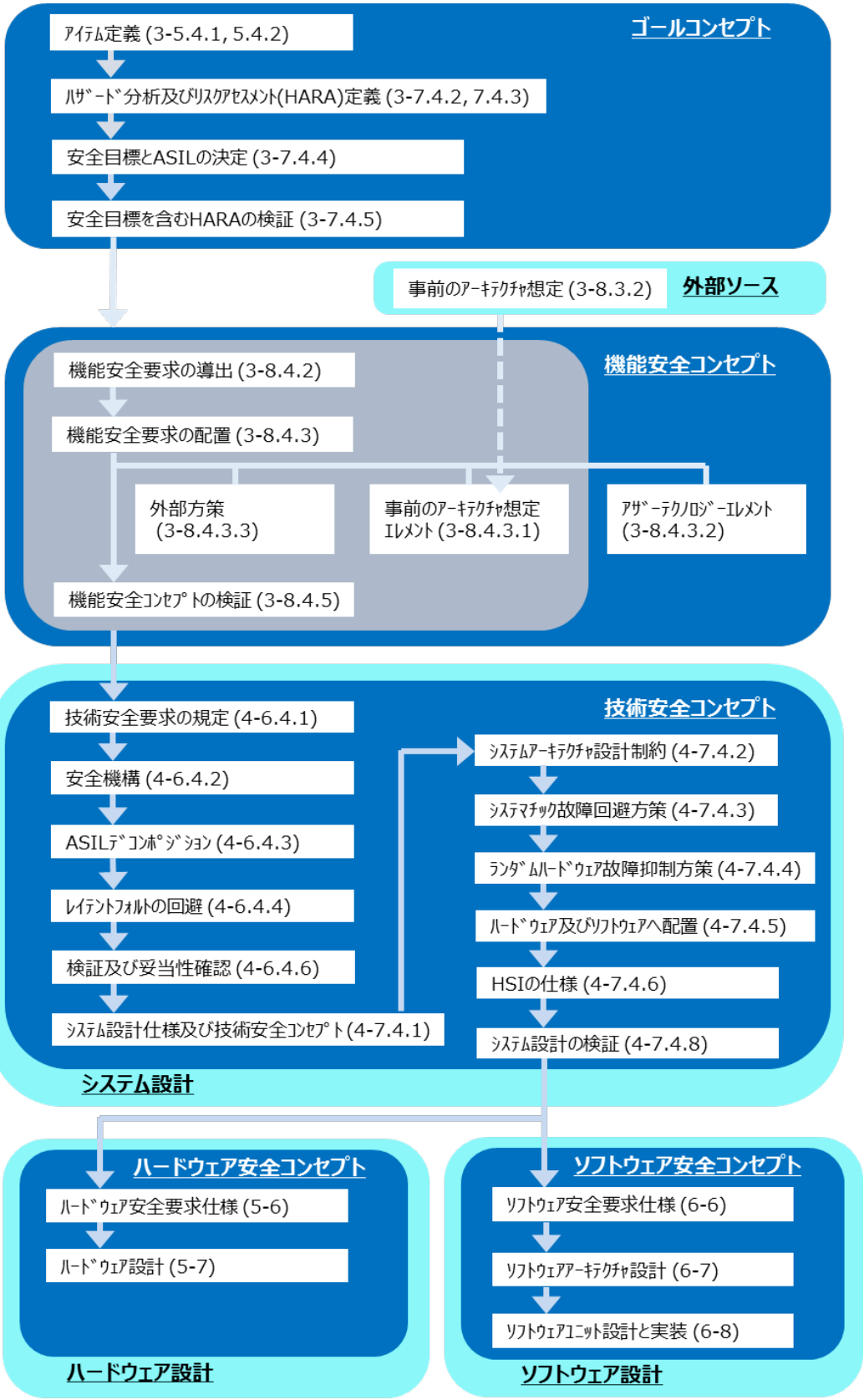


図 56　安全活動による開発フローの一例

A.3 安全設計階層モデル

前項における自動車機能安全規格の開発プロセスで特に重要となるのが、「要求」、「アーキテクチャ」、「分析」の 3 軸である。この 3 つの軸を相互に関連させながらゴールコンセプトの安全目標を起点としたトップダウンアプローチで段階的に実装レベルにまで落とし込むことで、安全目標の確実な実装が可能となる。階層モデルのイメージを図 57 に、各階層における対象を表 7 にそれぞれ示す。

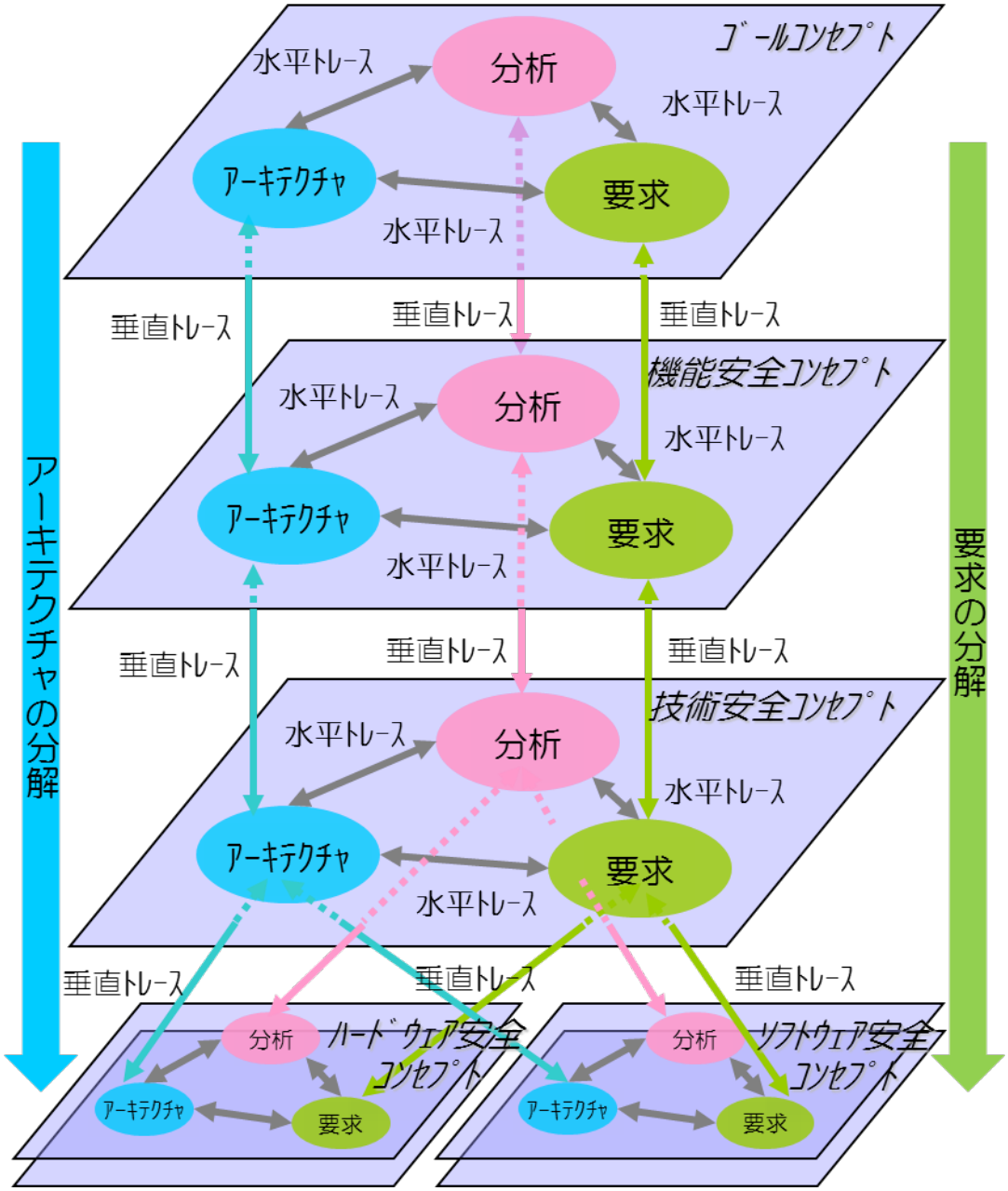


図 57　安全設計階層モデル

表 7

階層	要求	アーキテクチャ	分析
ゴールコンセプト	・アイテムへの要求 ・安全目標	・アイテム	・ハザード分析 → 安全目標導出
機能安全コンセプト	・意図した機能要求 ・機能安全要求	・PAA のエレメント	・安全分析＋従属故障分析 → 機能安全要求抽出
技術安全コンセプト	・意図した機能要求 ・技術安全要求	・システム･アーキテクチャのエレメント	・安全分析＋従属故障分析 → 技術安全要求抽出
ハードウェア安全コンセプト	・意図した機能要求 ・ハードウェア安全要求	・ハードウェア･アーキテクチャのエレメント	・安全分析＋従属故障分析 → ハードウェア安全要求抽出
ソフトウェア安全コンセプト	・意図した機能要求 ・ソフトウェア安全要求	・ソフトウェア･アーキテクチャのエレメント	・安全分析＋従属故障分析 → ソフトウェア安全要求抽出

注釈）・PAA：Preliminary architectural Assumptions

・機能安全コンセプトでは外部方策やアザーテクノロジーエレメントにも安全要求が配置されるが表 7 では対象アイテムのアーキテクチャのみにフォーカスしている

各コンセプトレベルにおいて、3 つの軸となる「要求」の定義、「アーキテクチャ」の構成要素への要求の割り付け、ハザードまたは安全に関する「分析」といった作業を繰り返し、各コンセプトレベルにおける安全コンセプトを導出していく。各階層での作業においては、常に上位との繋がりに不整合がないかを確認するとともに、各階層にわたって段階的に詳細化する上で適した粒度となるよう留意する。

この安全コンセプトの形成における安全アーキテクチャの作図に SCDL を適用すると要求とアーキテクチャ構成要素の包含関係や、要求間、アーキテクチャ構成要素間の繋がりが明確となり、安全分析や従属故障分析がしやすくなる。

また、トップダウンアプローチで安全コンセプトを段階的に詳細化していく過程において、各要素の一貫性や完全性を保証するために 3 つの軸間の双方向トレーサビリティを確保することが必要となる。たとえば、「意図した機能要求」と「アーキテクチャ構成要素」、「意図した機能要求」と「機能不全モード」、「機能不全モード」と「安全要求」、「安全要求」と「アーキテクチャ構成要素」など、「要求」と「アーキテクチャ」と「分析」間において、双方向のトレーサビリティを確保することが期待される。この横のつながりを本章では双方向の水平トレーサビリティと称する。

さらに、上位階層となるゴールコンセプトから、下位階層となるハードウェアコンセプ

トやソフトウェアコンセプトに至る縦のつながりを本章では双方向の垂直トレーサビリティと称する。たとえば、3 つの軸のうち「要求」の軸においては、「安全目標」と「機能安全要求」、「機能安全要求」と「技術安全要求」、「技術安全要求」と「ハードウェア安全要求」、「技術安全要求」と「ソフトウェア安全要求」などにおいて、双方向の垂直トレーサビリティを確保することが期待される。

このように、各階層の双方向の水平トレーサビリティと各軸の双方向の垂直トレーサビリティによる 3 次元トレーサビリティの確保により、安全目標の確実な実装の支援および、安全性の論証がしやすくなる。

A.4 機能安全コンセプトにおけるユースケース

図 57 安全設計階層モデルのゴールコンセプトから機能安全コンセプトに至る安全アーキテクチャのユースケースを紹介する。図 58 は、安全活動における開発フローにおいて、SCDL の各ダイアグラムを各フェーズに段階的に適用するイメージを示している。

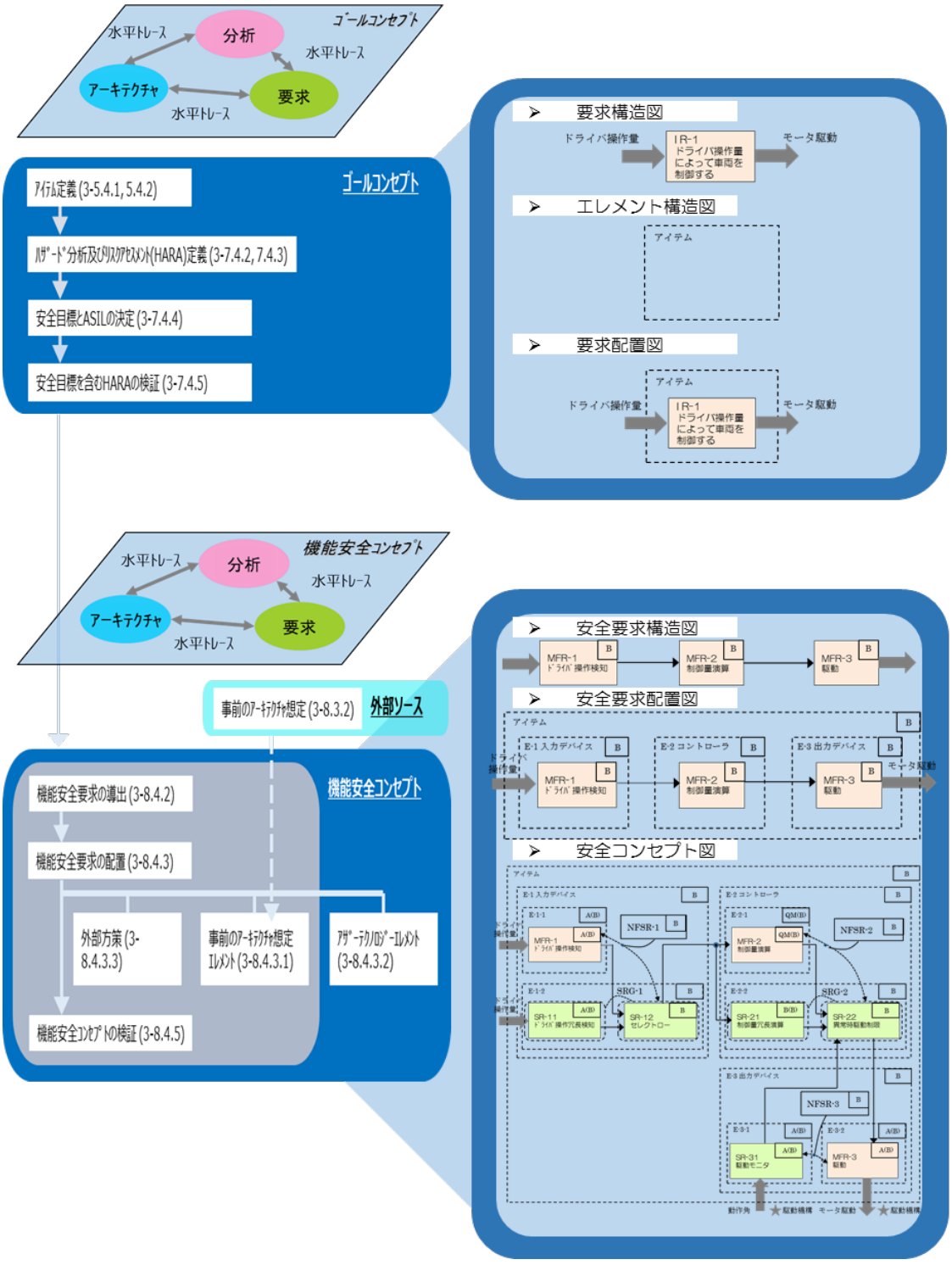


図 58 SCDL ダイアグラムの適用イメージ

図 58 のエレメント構造図、安全要求配置図、安全コンセプト図などの概要説明が本仕様書の 3.5 項に記載されているので参照するとよい。

尚、ここで紹介するユースケースは、初期段階で意図した機能に ASIL を付与し、デコンポジションをおこなうアプローチ例となっている。

A.5 アイテム定義～安全目標の検証

ドライバ操作量によって車両を制御するための車両制御システムをアイテムと定義する。図 59 にアイテムの要求構造図、図 60 にアイテムの要求配置図を示す。

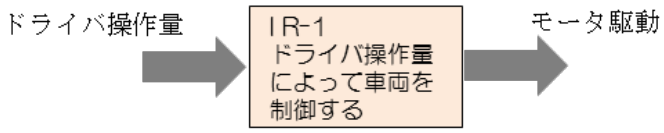


図 59 アイテムの要求構造図

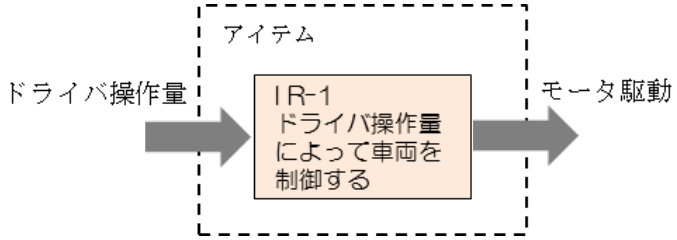


図 60 アイテムの要求配置図

図 60 アイテムの要求配置図に対するハザード分析とリスクアセスメントを実施した結果を表 8 に示す

表 8

ハザード	車両が意図せず大きく偏向する
安全目標	車両が意図せず大きく偏向するような過大な出力を発生させない
安全状態	車両の意図しない偏向が、3deg/s 以下
時間的制約	安全状態に 500ms 以内で移行すること
ASIL	B

図 61 にハザード分析とリスクアセスメントの結果を反映したアイテムの安全要求配置図（安全目標配置図）を示す。

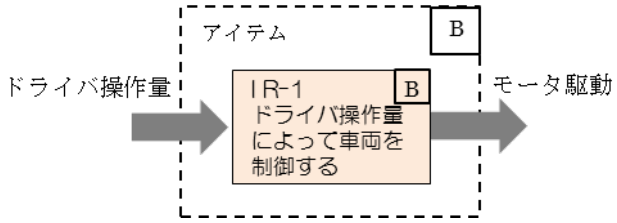


図 61　アイテムの安全要求配置図（安全目標配置図）

表 9 に示すようにアイテムには 3 つの意図した機能要求 MFR-1～3 が定義されている。また、意図した機能要求間の関係を図 62　要求構造図に示す。

表 9

ID	名称	意図した機能要求
MFR-1	ドライバ操作検知	ドライバの操作量を検知してコントローラへ送る
MFR-2	制御量演算	ドライバの操作量に従った制御量を演算し、演算値を出力デバイスへ送る
MFR-3	駆動	演算値に応じて駆動をおこなう

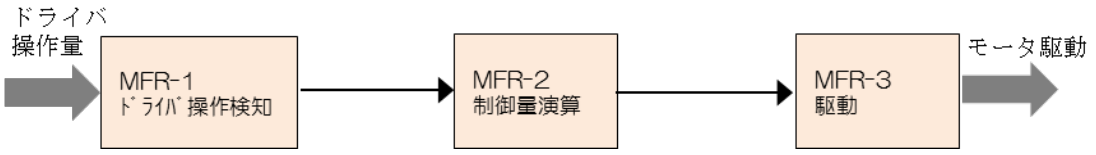


図 62　要求構造図

図 63 に示すようにシステムにはアイテムを構成するエレメントとして、E-1 入力デバイス、E-2 コントローラ、E-3 出力デバイスがある。

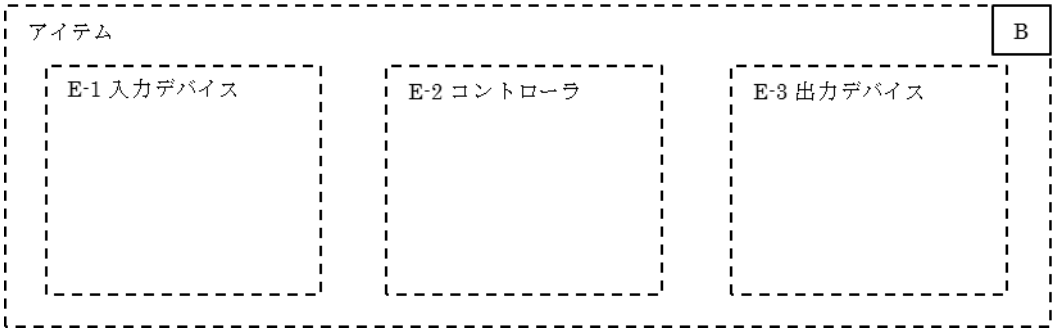


図 63　アイテムを構成するエレメント構造図

図 62　要求構造図と図 63　アイテムを構成するエレメント構造図を統合すると図 64 要求配置図ができる。

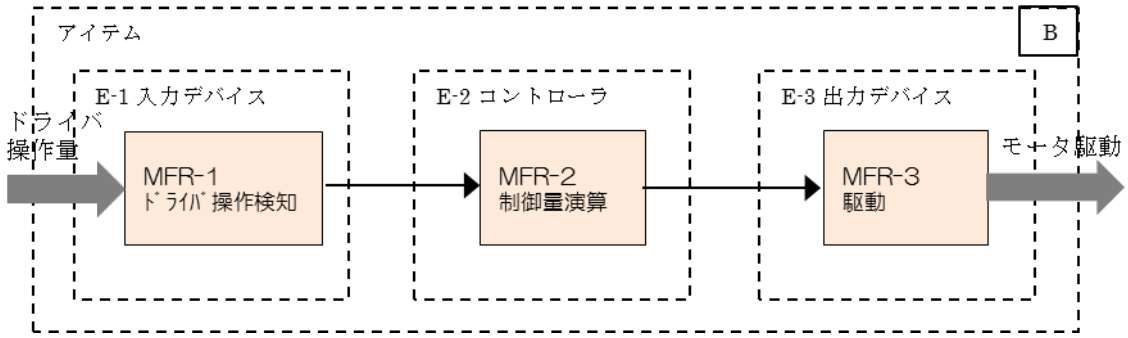


図 64　要求配置図

A.6 機能安全要求の導出

図 64　要求配置図において、入力デバイス･エレメントに配置されたドライバ操作検知要求 MFR-1、コントローラ･エレメントに配置された制御量演算要求 MFR-2、出力デバイス･エレメントに配置された駆動要求 MFR-3 を安全分析した結果、各要求の機能不全モードが ASIL-B の安全目標を侵害することが判明した（表 10)。

表 10

ID	名称	安全目標侵害の可能性がある機能不全
MFR-1	ドライバ操作検知	過大側に入力値を誤る
MFR-2	制御量演算	過大側に演算値を誤る
MFR-3	駆動	過大な駆動をおこなう

したがって、表 9 は表 11 に示すように意図した機能要求が安全の要求を併せ持つことになる。

表 11

ID	名称	意図した機能要求に由来する安全要求
MFR-1	ドライバ操作検知	ドライバの操作量を(正しく)検知してコントローラへ(正しく)送る
MFR-2	制御量演算	ドライバの操作量に従った制御量を(正しく)演算し、演算値を出力デバイスへ(正しく)送る
MFR-3	駆動	演算値に応じて(正しく)駆動をおこなう

注釈）ここでいう“正しく”とは、安全目標を侵害しないように”正しく”という意味である

このようにして、図 64　要求配置図は、図 65　安全要求配置図としてリファインされる。

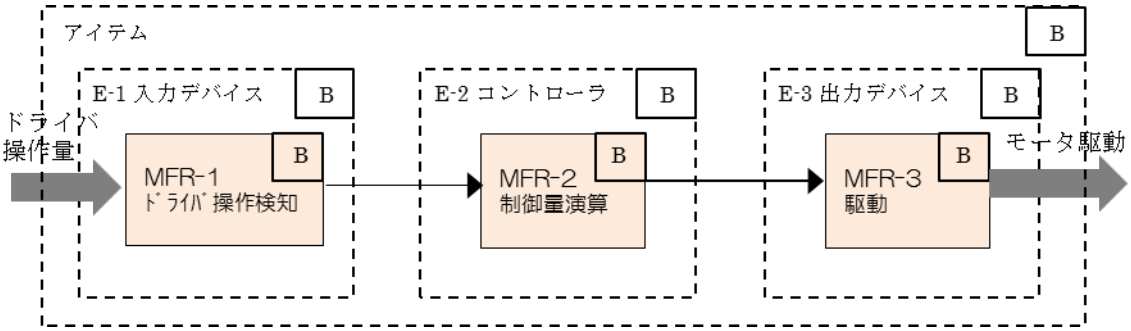


図 65　安全要求配置図

機能要求の機能不全により安全目標を侵害するため、安全目標を侵害しないようにするための安全機構を含む安全方策を検討することになるが、これらを実現する方法についてはある程度の自由度がある。

A.7 機能安全要求の配置～機能安全コンセプト検証

次に、図 65　安全要求配置図における E-1～3 の各エレメント単位で安全方策を検討していくことになるが、ここでは事例として 2 例のユースケースを示す。安全方策には自由度がある（解が一つとは限らない）ため、過去実績、造り易さ、コスト等を考慮し、複数の選択肢からベストなソリューションを選択するとよい。

尚、2 例のユースケースにおける安全方策の検討は、E-1 エレメントのみを代表例として扱い、安全要求構造図や安全要求配置図を示す。E-2 と E-3 のエレメントについては安全方策の検討経緯を割愛し、E-1、E-2、E-3 の全エレメントに関する安全方策の検討結果を安全コンセプト図として統合する。

(1) ユースケース 1

入力デバイス･エレメントのドライバ操作検知要求 MFR-1 をデコンポジションして冗長設計とするための安全要求グループ SRG-1（ドライバ操作冗長検知要求 SR-11、セレクトロー要求 SR-12）および独立要求 NFSR-1、コントローラ･エレメントの制御量演算要求 MFR-2 をデコンポジションして冗長設計とするための安全要求グループ SRG-2（制御量冗長演算要求 SR-21、異常時駆動制限要求 SR-22）および独立要求 NFSR-2、出力デバイス･エレメントの駆動要求 MFR-3 のフィードバックを監視して状態を通知する駆動モニタ要求 SR-31 および独立要求 NFSR-3 を配置する。

安全分析および従属故障分析によって抽出した安全要求を表 12、独立要求を表 13、安全要求構造図を図 66、図 67 に示す（SR-12 の出力側インタラクション先の要素は省略）。また、機能安全要求をエレメントへ配置した結果として安全要求配置図を図 68 に示す。

表 12

ID	名称	要求グループ	安全要求（安全目標侵害なきよう要求を満たす）
SR-11	ドライバ操作冗長検知	SRG-1	ドライバの操作量を検知する
SR-12	セレクトロー		2 つのドライバ操作量を比較し、ロー側を選択する
SR-21	制御量冗長演算	SRG-2	ドライバの操作量に従った制御量を演算する
SR-22	異常時駆動制限		2 つの演算値比較や駆動モニタ SR-31 からの異常通知時に駆動を制限する
SR-31	駆動モニタ	―	駆動量を監視して、SR-22 に異常を通知する

表 13

ID	独立要求
NFSR-1	MFR-1 と SRG-1 は従属故障を持たないこと
NFSR-2	MFR-2 と SRG-2 は従属故障を持たないこと
NFSR-3	MFR-3 と SR-31 は従属故障を持たないこと

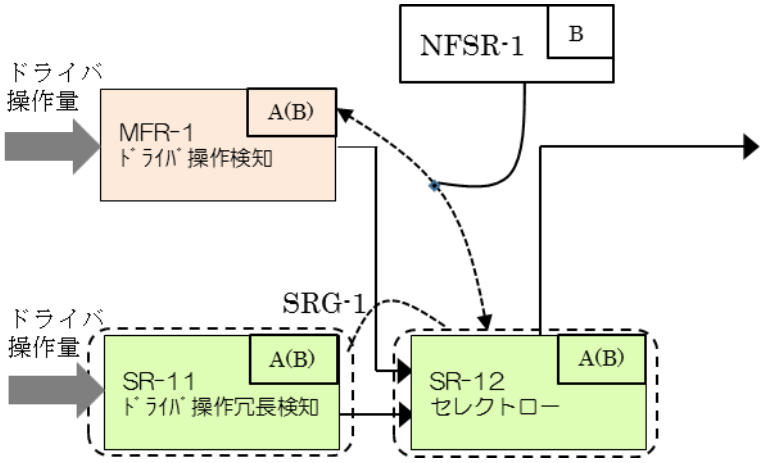


図 66　安全要求構造図

図 66 におけるデコンポジションの SR-12 セレクトローは安全分析の結果、単独で安全目標を侵害することが判明したため、安全目標の ASIL B を割り当てる（図 67）。

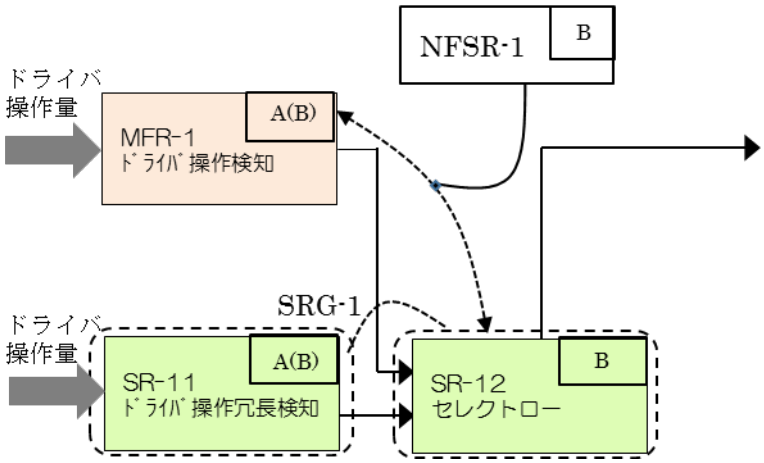


図 67　安全要求構造図

注釈）本章では「意図した機能要求に由来する安全要求：ペールオレンジ」と
「安全機構として追加した安全要求：グリーン」を区別するため、色分けして示す

安全要求構造図をエレメント構造図に統合すると図 68　安全要求配置図となる。

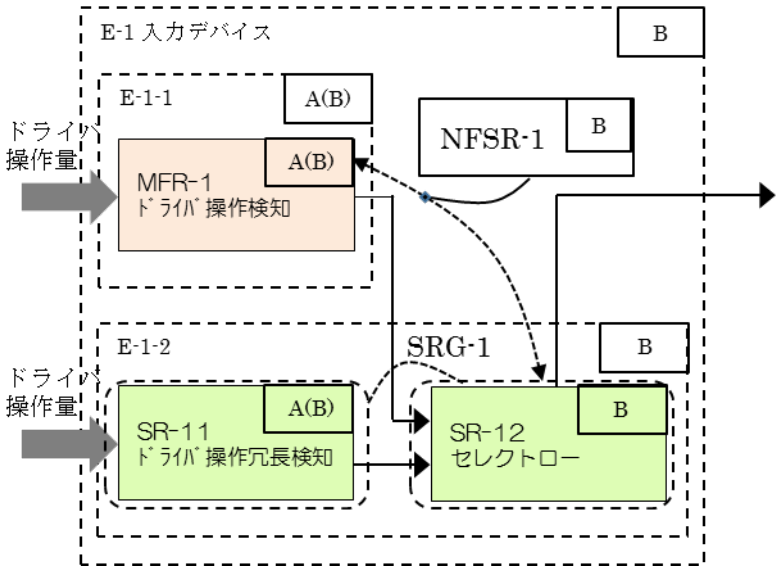


図 68　安全要求配置図

このように意図した機能を実現する単位で安全要求配置図を作成し、意図した機能単位での安全方策の有効性や適切性を検証するとよい。しかしながら、システムが小規模な場合には、必ずしも意図した機能単位で安全要求配置図を作成して個々に安全方策を検証していくアプローチではなく、全体の安全要求図とエレメント図を統合して、次に示す安全コンセプト図を作成してから安全方策の検証をしてもよい。

導出した全ての機能安全要求をエレメントに配置し、安全方策の確からしさの検証および安全アーキテクチャの最適化を実施することで、最終的な機能安全コンセプトを作成することができる。すなわち、すべての機能単位の安全要求配置図が出揃い、それらを統合して安全コンセプト図としてまとめることで、機能安全コンセプトを示すことができるのである。

SCDL による安全アーキテクチャを図 69　安全コンセプト図に示す。

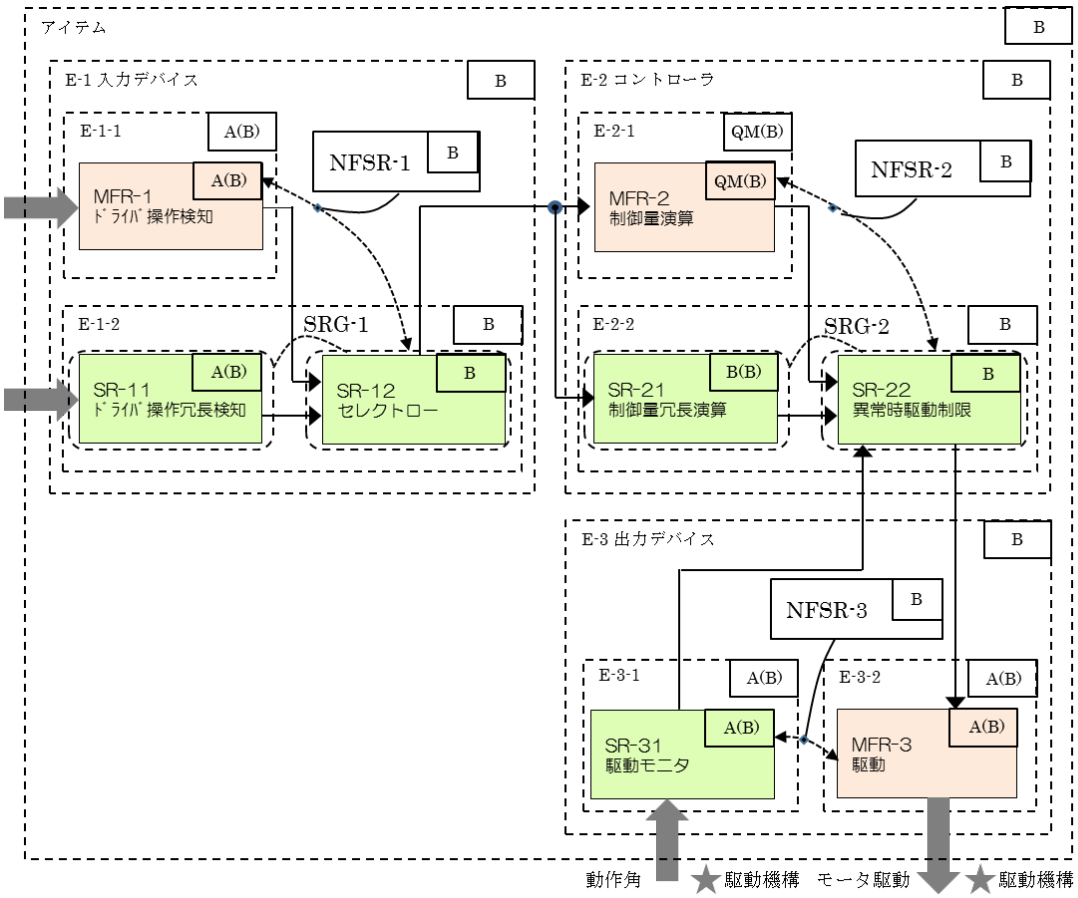


図 69　安全コンセプト図

(2) ユースケース 2

入力デバイス･エレメントのドライバ操作検知要求 MFR-1 とコントローラ･エレメントの制御量演算要求 MFR-2 からなる要求グループ RG-1 の出力値を規定値内に抑えるための制御量上側制限要求 SR-21 および独立要求 NFSR-1、出力デバイス･エレメントの駆動要求 MFR-3 の駆動量をモニタして異常時に駆動を制限するための安全要求グループ SRG-1（駆動モニタ要求 SR-31 と異常時駆動制限要求 SR-32）および独立要求 NFSR-2 を配置する。

安全分析および従属故障分析によって抽出した安全要求を表 14、独立要求を表 15 に、安全要求構造図を図 70 に示す。図 70 における SR-21 制御量上域制限は安全分析の結果、単独で安全目標を侵害することが判明したため、安全目標の ASIL B を割り当てる。また、機能安全要求をエレメントへ配置した結果として安全要求配置図を図 71 に示す。

表 14

ID	名称	要求グループ	安全要求（安全目標侵害なきよう要求を満たす）
SR-21	制御量上域制限	－	制御量を上限値で制限する
SR-31	駆動モニタ	SRG-1	駆動量を監視して、SR-32 に異常を通知する
SR-32	異常時駆動制限		駆動モニタで異常検知時に駆動を制限する

表 15

ID	独立要求
NFSR-1	RG-1 と SR-21 は従属故障を持たないこと
NFSR-2	MFR-3 と SRG-1 は従属故障を持たないこと

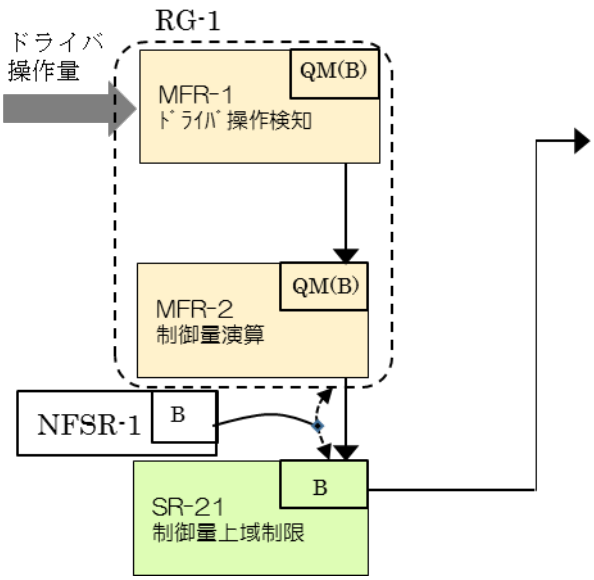


図 70　安全要求構造図

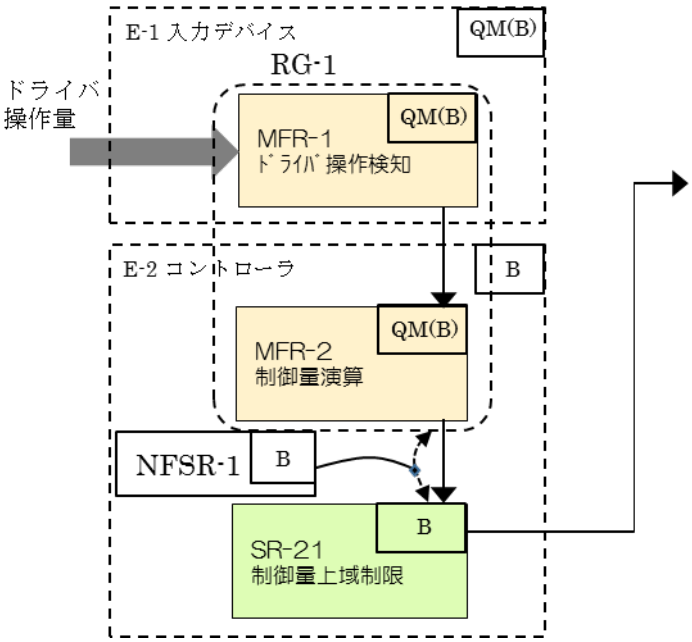


図 71　安全要求配置図

導出した全ての機能安全要求をエレメントに配置し、安全方策の確からしさの検証および安全アーキテクチャの最適化を実施することで、最終的な機能安全コンセプトを作成することができる。すなわち、すべての機能単位の安全要求配置図が出揃い、それらを統合して安全コンセプト図としてまとめることで、機能安全コンセプトを示すことができるのである。

SCDL による安全アーキテクチャを図 72　安全コンセプト図に示す。

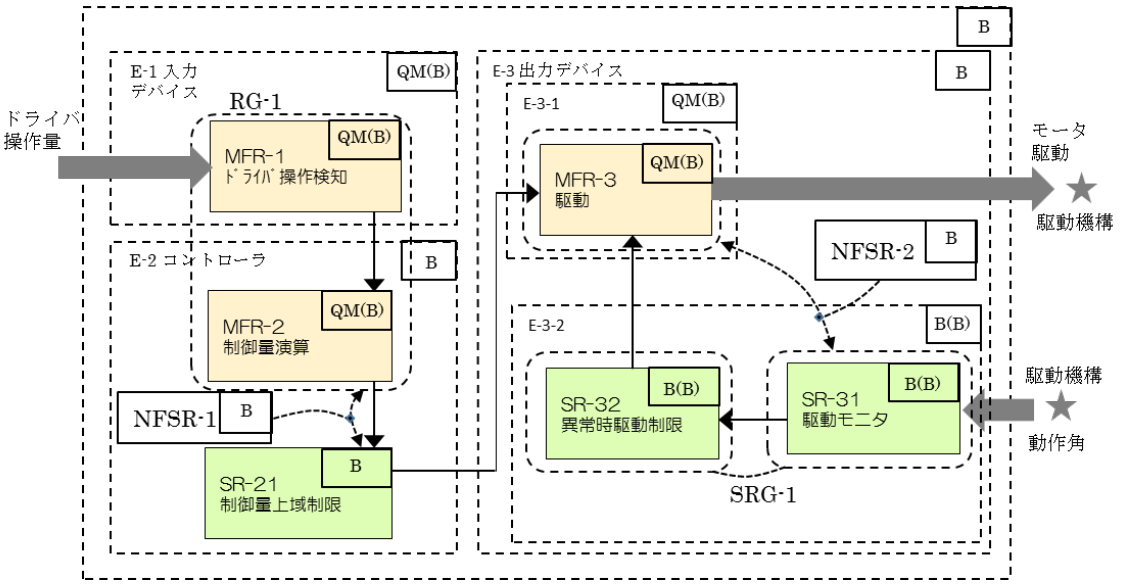


図 72　安全コンセプト図

付属書B ユースケース（ハードウェア編）

B.1 はじめに

附属書A(SCDL仕様書 Ver. 1.2以降)では、自動車分野の機能安全規格(ISO 26262：2011)を参照した開発フローにおいて、ゴールコンセプト(附属書 A.2 参照)から機能安全コンセプトに至る範囲の安全アーキテクチャの設計にSCDLを適用したユースケースを示した。本編では、同じく自動車分野における機能安全規格（ISO 26262 ： 2011）の技術安全コンセプトからハードウェアレベルの安全コンセプト（本仕様書ではハードウェア安全コンセプトと称す）に至る範囲の安全アーキテクチャ設計にSCDLを適用したユースケースを示す。なお、自動車分野の機能安全規格は2018年版（ISO 26262 ： 2018）が発効済であるが2011年版の規格要求内容は基本的に2018年版に引き継がれているため、本SCDL仕様書では、その作成過程当初に発効済であった自動車分野の機能安全規格の 2011 年版（ISO 26262 ： 2011）をベースとしている。

B.2 ハードウェアにおける安全アーキテクチャ設計の考え方

自動車分野の機能安全規格においては、附属書 A.3 の安全設計階層モデルに示すようにゴールコンセプト、機能安全コンセプト、技術安全コンセプト、ハードウェア安全コンセプト、ソフトウェア安全コンセプトといった流れのトップダウンアプローチが示されている。しかしながら、既製品をベースに次の製品を派生開発したり、軸となる開発品をベースにシリーズ品を派生開発したりする場合には、過去に実績のあるコンポーネントが流用される。特に自動車分野の機能安全規格におけるハードウェアの安全設計においては、「ハードウェアアーキテクチャメトリックの評価」や「ランダムハードウェア故障による安全目標侵害の評価」において、ハードウェア部品の故障率を扱うため、故障率に基づく評価が規格要件（ISO 26262-5:2011）を満たさなければならない。そのため、安全目標からのトップダウンアプローチおよび実績のあるハードウェアコンポーネントからのボトムアップ（部品調達などのハードウェア上の制約や素子構成などの情報を上位に持ち上げること）を併用することも少なくない。したがって、本編では、トップダウンとボトムアップを併用して総合（Synthesis）し、技術安全コンセプトからハードウェア安全コンセプトに至る安全アーキテクチャ設計を構築する場合のユースケースを取り扱うこととする(図 73 参照)。

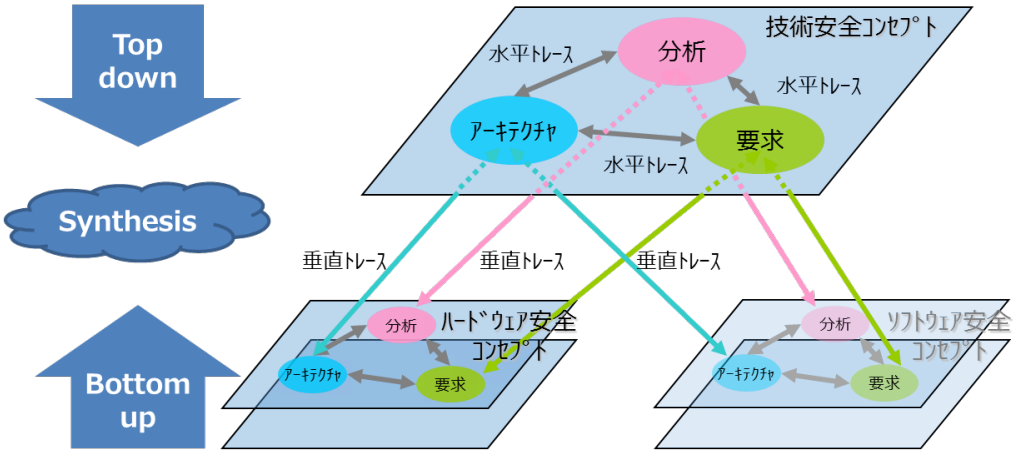


図 73　トップダウンアプローチとボトムアップの総合

B.3 ハードウェア設計

ハードウェア設計をおこなう上で、エンジニアは様々な情報を取り扱う。その際に考慮するとよい事項の例として、表 16 に示すような項目がある。多くの項目は一般的に取り扱われる事項であるが、中には ISO 26262:2011 の規格要件として要求されている特有な項目もあるため、それらについては、分類欄に ISO 26262 と記している。エンジニアはこれらの項目を踏まえて、ハードウェア設計の戦略を立て、ハードウェア安全コンセプト（ハードウェア安全アーキテクチャを含む）を構築していく。また、ハードウェア設計の戦略は上流にエスカレーションされ、機能安全コンセプトや技術安全コンセプトの構築において考慮される場合がある。

表 16　ハードウェア設計時の検討事項

	項　目	分　類
1	環境制約	一般
2	搭載スペース	一般
3	調達	一般
4	新規／流用	一般
5	安全設計の定量評価	ISO 26262
6	独立要求の配慮	一般/ISO 26262
7	従属故障	一般/ISO 26262
8	ハードウェア部品認定	一般/ISO 26262

上記の各項目は、ハードウェア設計戦略を検討する上で重要な項目の例である。また、これらの項目は、ある項目が他の項目の入力または出力となるような相関関係を持っている。これらの関係について図 74 に示す。

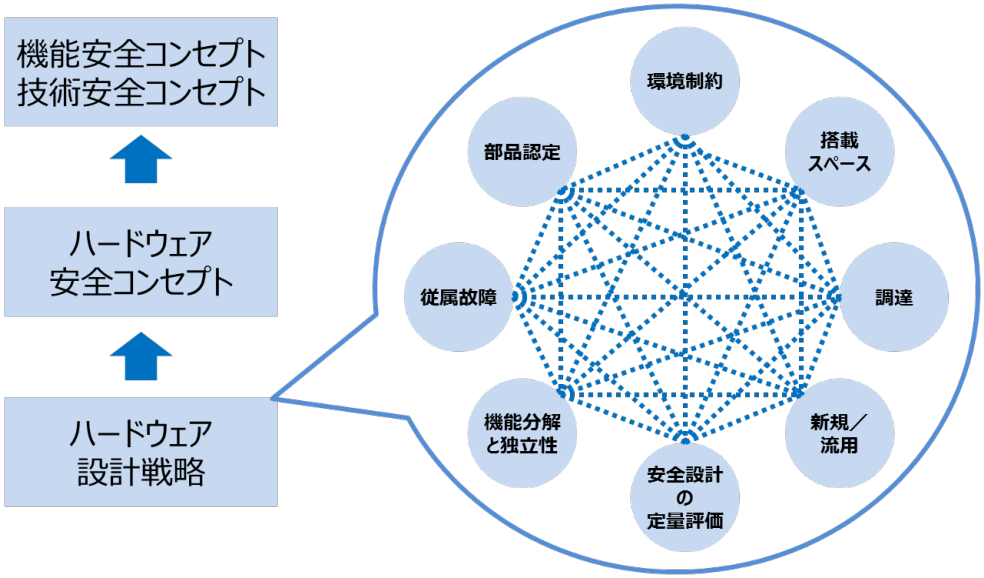


図 74　ハードウェア設計戦略導出における各事項の相互関係

次に、ハードウェア設計を進めるにあたり、表 16 の各項目で考慮するとよい事項について、B. 3. 1～B. 3. 8 に述べる。

B.3.1 環境制約

自動車に搭載される電気/電子システムの開発において、アイテム開発の企画・構想を検討する際には、その製品の仕向地の計画や ECU（Electronic Control Unit）レイアウトなどの情報を基に初期アーキテクチャ構想図が設計ドキュメントとして起こされる。そのドキュメントには、温度、湿度、振動などの環境制約としての仕様も示される。機能安全における安全目標や機能安全コンセプトを記述する段階で、ハードウェア部品の故障率を想定したレイアウトを検討し、環境制約や動作条件の充足が評価されることがある（図 75 参照）。

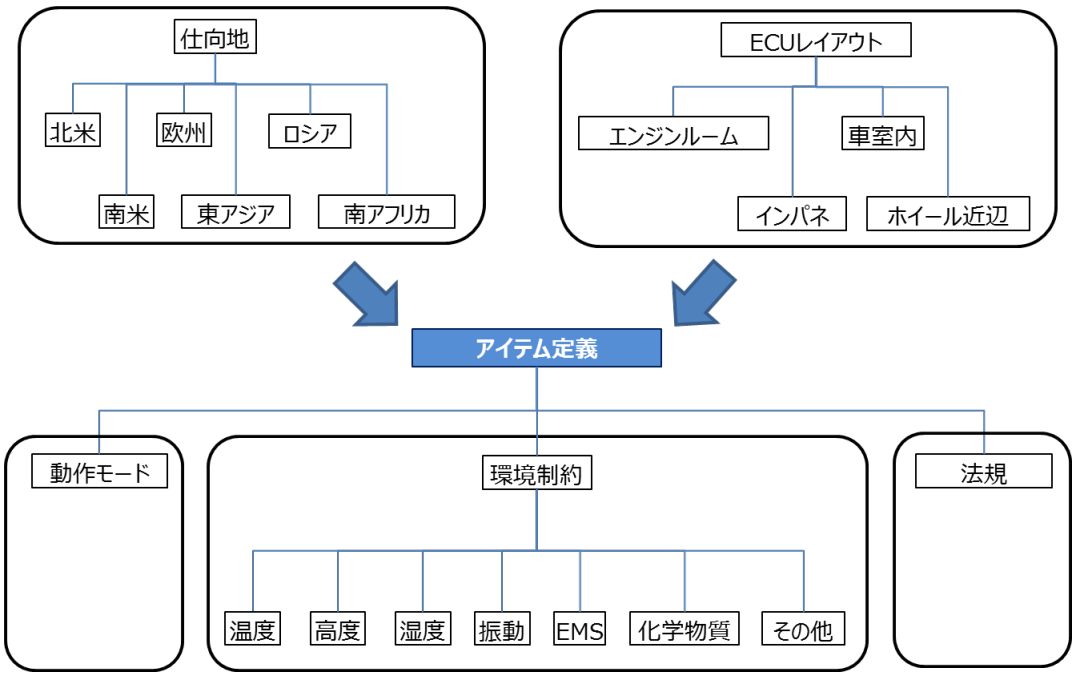


図 75　アイテム定義に関する環境制約

システム設計の段階で各エレメントへの要求の配置が決まり、ハードウェアおよびソフトウェアの仕様が導出されるタイミングで、PMHF（Probabilistic Metric for random Hardware Failures）などの故障率の概算計算をおこなう必要が出てくる。故障率の算出に必要なパラメータの中には、環境温度や動作タイミングなどがあり、それらをベースに故障率データベースのミッションプロファイルを決めたり、ハードウェア統合テストにおけるテスト条件に反映したりする。そのため、根拠となる数値については完成車メーカとサプライヤ間で合意しておくことが必要となる（図 76 参照）。

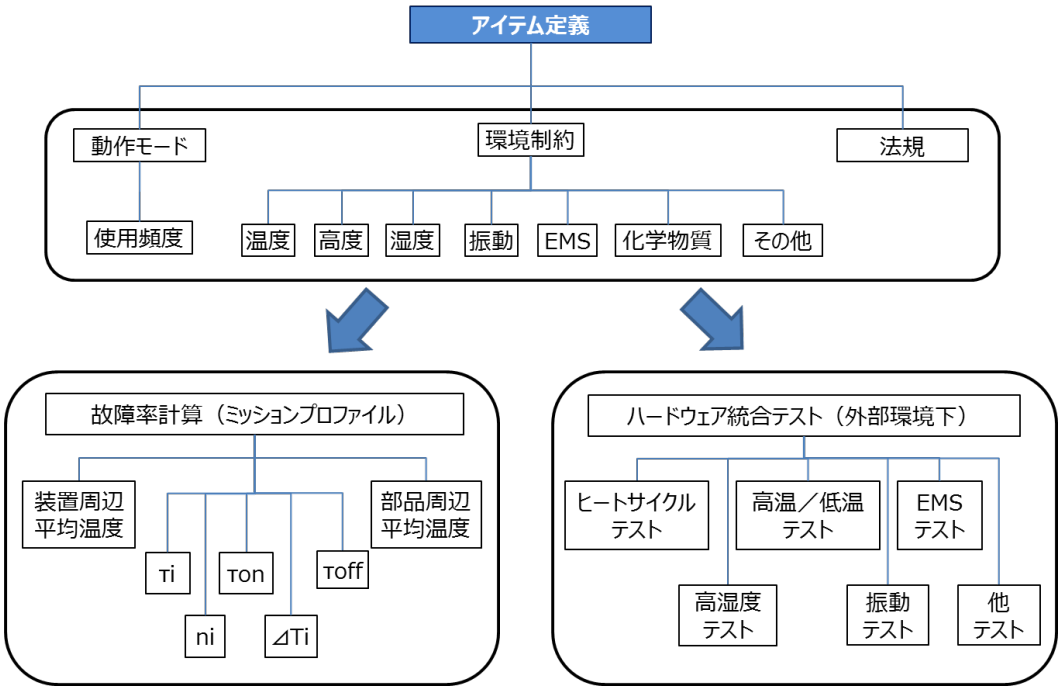


図 76 ミッションプロファイルおよびテスト条件の導出

B.3.2 搭載スペース

搭載スペースの検討においては、システムレベルで決められた制約条件にしたがって、ハードウェア部品の構成や配置を最適化しなければならない。

製品や部品を供給する役割ごとに検討する対象が異なるため、本項では ECU メーカ（No. 2）および半導体メーカ（No. 3）を扱うこととして、搭載スペースの検討で考慮すべきことを示す（図 77 参照）。

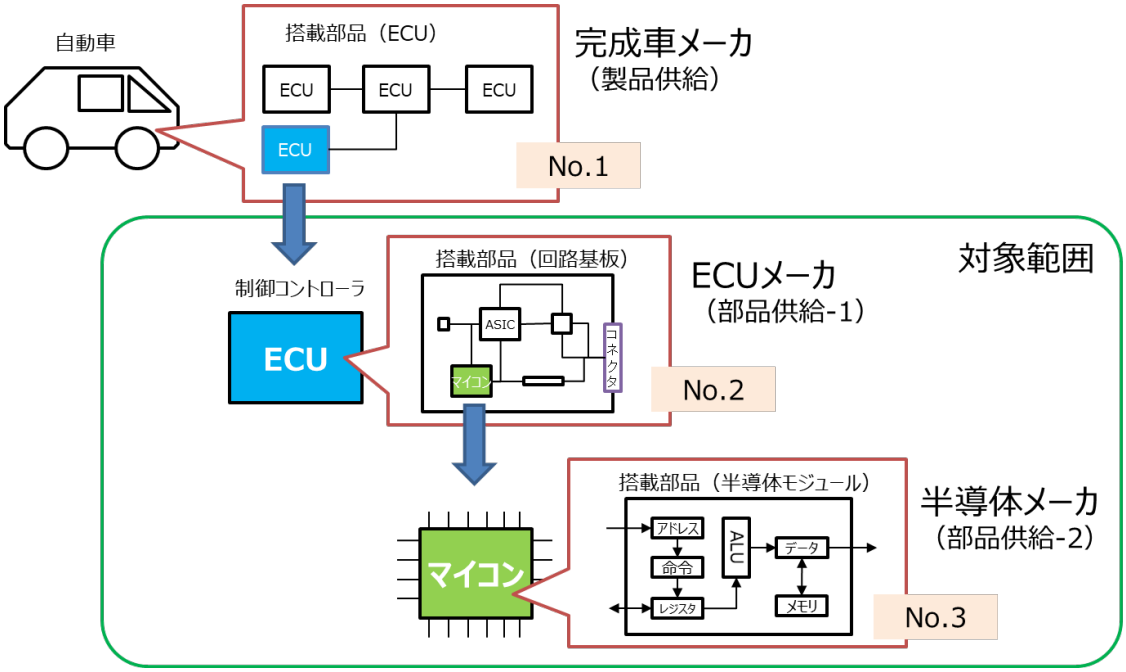


図 77　プロダクトの関係

搭載スペースの観点からハードウェア設計を決定する上での主な要素（検討項目）を表17 に示す。各要素が相互に関連してハードウェア設計が決まるため、多視点でのバランスをとる必要がある。第一指標としている各特性は、品質特性モデルのディジュールスタンダード（De jure standard）などを参考にしている。また、関連する項目は、表 1 ハードウェア設計時の検討事項で挙げた項目である。

表 17　搭載スペースを決定する上で考慮すべきハードウェア設計の要素

特性	考慮要件	主な要素［例］	関連する項目
機能性	性能	要求仕様、目標性能	
	安全	デコンポジション／独立要求のさばき	独立要求の配慮
		カスケード故障、共通原因故障	従属故障
		故障率	安全設計の定量評価（部品故障率）
			安全設計の定量評価（安全機構カバレッジ）
信頼性	物理	温度、湿度、力、摩擦、振動	環境制約
	電気	ノイズ、電波、電圧	
	環境	化学物質、設置場所	
	耐久	振動、使用頻度、稼働率、負荷	
実現性	干渉	コネクタ・ピン位置、レイアウト、部品間隔	
	配置	電源位置、寸法、重量、部品位置	
	確からしさ	部品認定	新規／流用
			ハードウェア部品認定
	高密度実装	ASIC 化、高密度実装技術、冷却技術	
保守性	生産	製造時検査、組み付け	
	維持	交換などのアフターサービス	
収益性	コスト	調達	調達

表 17 は搭載スペースを決定する上でのハードウェア設計に関する主な特性とそれらに関連する要素を示しており、表 18 では表 17 の特性を踏まえて、具体的に搭載スペースを決定するために必要なハードウェア設計者が持つべき視点の例を示している。また、「収益性」のように国際規格の品質特性モデルにはない特性も、搭載スペースを決定する上での特性として採用している。

表 18　搭載スペースの検討に適用する特性

特性	考慮要件	要素［例］	部品供給検討の視点
機能性	性能	要求仕様	要求を漏れなく実装できるか
	安全	デコンポジション	安全系の独立要求を実装できるか
		カスケード故障	安全系、非安全系の無干渉が実装できるか
		故障率	目標故障率を達成する冗長機能を実装できるか
信頼性	物理	温度	放熱板のレイアウトが成立するか
	電気	ノイズ	ノイズを受けにくいレイアウトを実装できるか
	環境	化学物質	外的要因を回避する策を実装できるか
	耐久	振動	破損または変形を起こさないレイアウトを実装できるか
実現性	干渉	コネクタ・ピン位置	実装レイアウトが成立するか
	配置	電源位置	ノイズ源とならない実装レイアウトが成立するか
	確からしさ	部品認定	認定された部品を採用しているか
保守性	生産	製造時検査	製造時のテストができるか
	維持	交換などのアフターサービス	メンテナンス時に容易にアクセスできるか
収益性	コスト	調達	コスト要求を満たす部品を採用しているか

B.3.3 調達

自動車に搭載される電気/電子システムの開発では、システム実現に向けたハードウェア設計をおこなうことになるが、必ずしも機能安全に関する要求を満足するだけで、システム構築ができるわけではない。

機能要求を満たすシステムを継続的に市場に供給するためには

・機能要求を満足しつつ、コスト要求を満たす

・機能要求を満足するとともに部品の供給能力を満たすサプライヤの選定

などの調達要素も考慮する必要がある。そのためここでは、調達の観点から、前述表 16 のハードウェア設計時の検討事項との関連とハードウェア設計への入力情報の事例について述べる。なお、ここで調達とは、ハードウェア設計に関し、システムレベルで定められた制約事項にしたがい、仕様・コスト・品質の最適解を実現するハードウェア部品の安定供給を受けるための活動である。

ハードウェア設計部品の供給を受けるまでの調達のプロセスは、図 78 に示すように整理することができる。つまり調達のプロセスでは、サプライヤの業界調査、調達に関する戦略目標、調達仕様の決定、部品の安定供給や品質に対する安全性などの各ステップを経て、調達先であるサプライヤを決定する。その後、部品供給に向けた量産準備、数量および納期調整の後、部品納入、検収、支払などの各ステップを経て、必要に応じ、サプライヤパフォーマンスのフィードバックをおこなう。このため調達のプロセスは、次の二つのフェーズに大別できる。

- Sourcing：サプライヤの選定、調達仕様、価格の決定
- Purchasing：調達の実行

調達のプロセスでは、サプライヤの業界調査、調達に関する戦略目標、折衝+サプライヤ決定などの段階で、調達の各ステップとは別にサプライヤの収益調査やサプライヤの安全性を監視および確認しておくケースが想定される。

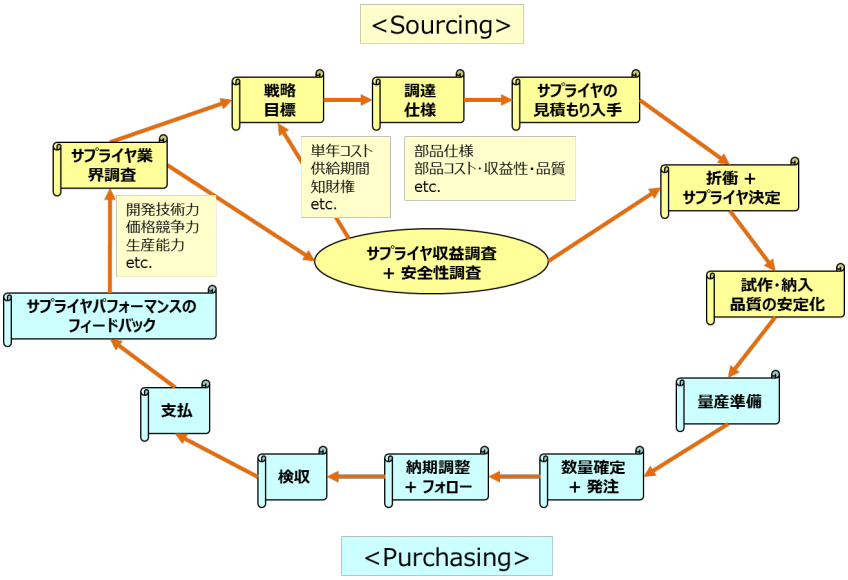


図 78　調達のプロセス（Sourcing & Purchasing）例

本編でスコープとするハードウェア設計は､｢設計｣であるため､図 78 のうち､｢Purchasing（調達の実行)｣より､｢Sourcing（サプライヤの選定・調達仕様・価格の決定)｣の各ステップに関連が深い。

この｢Sourcing（サプライヤの選定・調達仕様・価格の決定)｣の各ステップに関し、ハードウェア設計時の検討事項との関連と、ハードウェア設計への入力情報の事例について、表 19 に整理する。

表 19　サプライヤ業界調査のステップで関連するハードウェア設計への入力情報

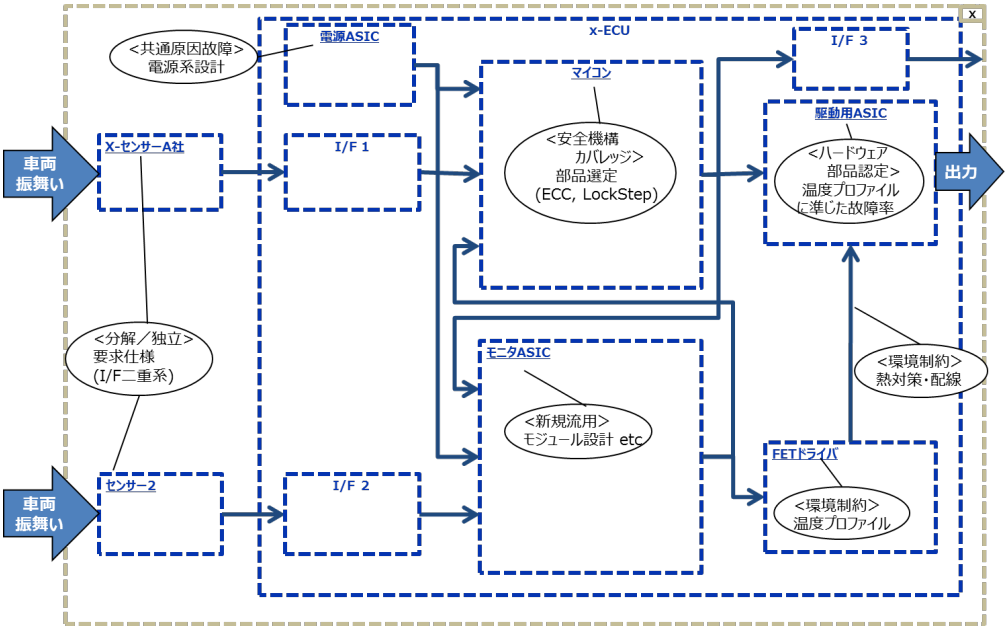
ステップ	考慮要件	主な要素	関連する項目	ハードウェア設計の入力情報
サプライヤ業界調査	開発 技術力	要求仕様、 部品仕様、 目標性能、 安全要求仕様、 部品信頼性、	搭載スペース	要求仕様（部品サイズ etc.）
			環境制約	発熱対策:配置、配線 温度プロファイル etc.
			安全設計の定量評価（定量目標故障率）	目標値（SPFM、LFM、PMHF etc.）
			安全設計の定量評価（安全機構カバレッジ）	部品選択（ECC、lockstep、etc.）
			従属故障（共通原因故障、カスケード故障）	電源系設計 etc.
			独立要求の配慮（分解/独立）	要求仕様（I/F 二重系 etc.）
	価格 競争力 (Cost)	生産コスト、 輸送コスト、 輸入コスト	(調達)	- （部品選定）
	生産能力 (Delivery)	納期、安定供給、 アフターサービス （補用品） サプライヤ評価 （供給能力）	(調達)	- （部品選定）
	品質 安定力 (Quality)	歩留り サプライヤ評価 （ISO 9001、 IATF 16949 取得）	(調達)	- （部品選定）

次に表 20 に品質の安定化までに関連する項目とハードウェア設計への入力情報の例を示す。

表 20　品質安定化のステップまでに関連するハードウェア設計への入力情報

ステップ	考慮要件	主な要素	関連する項目	ハードウェア設計への入力情報
戦略目標	知財権 単年コスト 供給期間 (含 QCD)	新規部品開発（汎用性／カスタム)	新規／流用	モジュール設計, 適度な粒度, 単純性の確保 etc.
			ハードウェア部品認定	温度プロファイルに準じた故障率 etc.
		単年コスト、供給期間（量産効果)	(調達)	- (部品選定)
調達仕様 ↓ 見積もり ↓ ｻﾌﾟﾗｲﾔ決定 ↓ 品質安定化	部品仕様	部品仕様評価結果	(調達)	- (部品選定)
	コスト	コスト査定結果	(調達)	- (部品選定)
	収益性	収益性評価結果	(調達)	- (部品選定)
	品質	納入部品品質評価結果	(調達)	- (部品選定)

表 19 および表 20 から、調達要素は、環境制約、搭載スペースなど前述表 16 の全ての検討事項と関連し、ハードウェア設計への入力情報にもつながることから、ハードウェア設計は、調達要素を考慮しながら検討をする必要がある。
以下に、表 19 および表 20 記載のハードウェア設計への入力情報を元に、ハードウェアアーキテクチャレベルでのハードウェア設計の事例を示す（図 79 参照)。



注）SCDL の文法に基づくエレメントの入れ子図を回路ブロック図として転用した例

図 79　アーキテクチャ設計と部品調達時の検討事項

B.3.4 新規／流用

ビジネスやプロジェクトの目標を達成するために、ハードウェアアーキテクチャを構成するエレメント（パーツや、モジュールなど）を「新規で開発するか」または「既存のものを流用するか」の判断は、長期的な視野と複合的思考で判断する必要がある。そこで、ハードウェアアーキテクチャ構築時に考慮されるであろう、新規／流用の判断について、どのようなアプローチで、どのように考慮するのかについてのヒントを示す。

図 80 に示すように、エレメントの新規／流用の判断は、ビジネス方針やプロジェクトの目標、およびその他のハードウェア開発戦略から作成された「当該プロジェクトの設計基準」を基に行われる。その際の設計基準は、ディジュールスタンダードにおける品質モデルを参考にしてもよい。このようにして作成された設計基準は、ハードウェアアーキテクチャ設計に引き渡され、ハードウェア要求や制約を達成することが可能となる複数のエレメントの選択肢（量産実績のある流用エレメントや、新規開発のエレメント）の選択基準として用いられる。図 80 中の「ハードウェアアーキテクチャ設計」には、選択した流用エレメントのインパクト分析などの設計活動が含まれている。

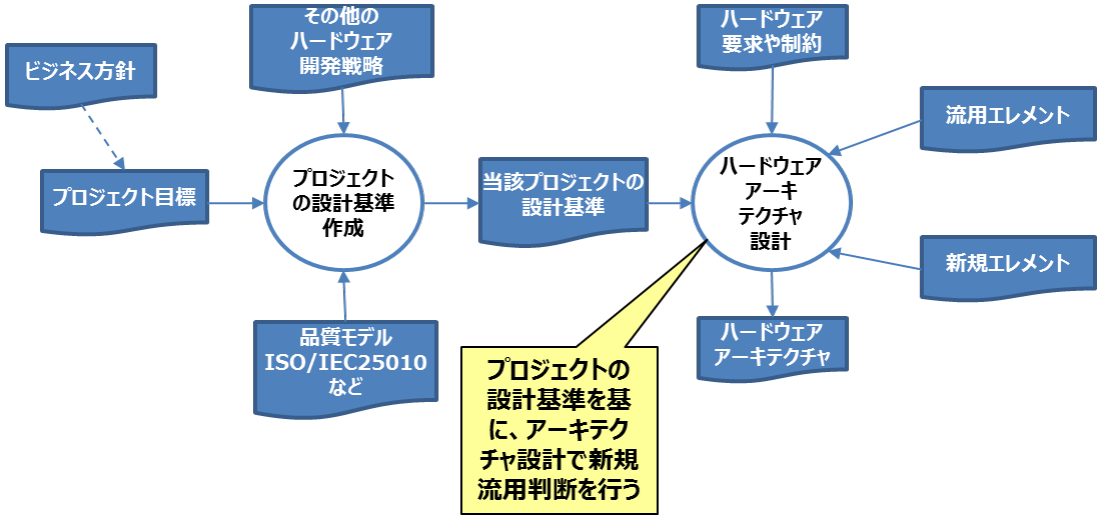


図 80　ハードウェアエレメントの新規／流用判断

B.3.5 安全設計の定量評価

機能安全規格（ISO 26262:2011）におけるハードウェア設計では、アーキテクチャを定量的に評価することが求められている。規格要件としてアイテムの定量目標値（PMHF、SPFM:Single point fault metric、LFM:Latent fault metric）の達成を求めているが、規格要件に準拠するための計算値自体は安全コンセプトに大きく依存するため、システム設計領域などの上位工程においても、定量目標値の故障率などの概算をおこなう必要が出てくる。そこで、例えば、次の情報を使って開発コストや安全目標への寄与度を勘案しながら定量目標値の達成に向けた戦略を検討する必要がある。

※本件で扱う故障率とは、部品（エレメント等）単体の故障率を下げる内容ではなく、安全機構の有効性を含めたコンセプトつまり、アイテム全体の定量目標値のことを指している。

- 定量目標値（PMHF、SPFM、LFM）
- 従来情報（派生元の製品情報など）
- 安全関連範囲の見込み
- ハードウェア部品認定情報（セーフティマニュアルやレポート等の入手状況）
- 主要エレメント故障率の見込み
- 安全機構の期待カバレッジ（DC）および性能に対する論証手法の選択
- 安全機構の SG に対する寄与度および実装コストを考慮し、適切に安全機構を選択
- 部品故障率／故障モード算出の基となるハンドブックおよび規格の選択
- 算出ツール選択
- 算出手法の選択（1st-method or 2nd-method, 定量 FTA or 近似式の利用等）

図 81 に示すように、ハードウェア設計に大きな影響を与えるシステム設計レベルでの PMHF 目標値との関連について簡単に取り上げる。また、安全機構カバレッジの検討については、図 83 で簡単に触れる。SPFM、LFM は実際に使用する部品の故障率などをベースに算出することになるため、一般的にはハードウェア設計完了時におこなう作業となる。しかしながら、システム設計の初期段階において、どのような故障を持った部品を使うのかを想定し、どのような安全機構でそれらの故障を検出および対処するのかを検討することで、PMHF 目標値達成の目途付けがしやすくなると考える。

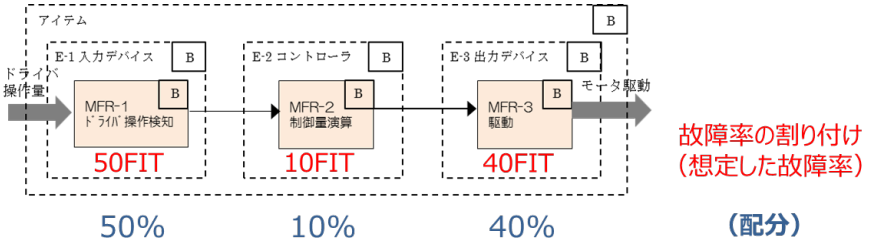


図 81　システム設計レベルの PMHF 目標値の割り付け（数字は仮定）

図 81 の E-1 入力デバイスは 50FIT 以内に、E-2 コントローラは 10FIT 以内に、E-3 出力デバイスは 40FIT 以内にそれぞれ収める必要があるため、E-１から E-3 の各々の構成要素となるハードウェアエレメントの故障率を勘案しながら、ハードウェア設計を進めることとなる。

図 82 に SPFM の算出の手順を簡単に紹介する。はじめに、対象となるシステムで使用される各部品の故障率の見積もりをおこなうことが必要となる。使用する故障率の根拠に関しては、ISO 26262:2011 を参照されたい。次に、故障モードと故障率の分布を想定する。さらに続いて、部品の故障モードが安全機構なしと想定したときに安全目標を侵害するかどうかを判断することになる。この部分は判断根拠を示すことが求められているため、根拠が明確に示せない場合には厳しめ側を選択することとなる。例えば、ロジック回路においては取り得る値が‘０’と’1’の二つの状態となるため、均等に発生すると考えることが妥当な場合には、安全側と危険側の配分を 50%ずつとする。さらに、安全を侵害する故障モードに関して、安全機構によるダイアグカバレッジを判断し、安全を侵害する Single Point Fault および Residual Fault の故障率を求め、アイテムの SPFM を算出する。

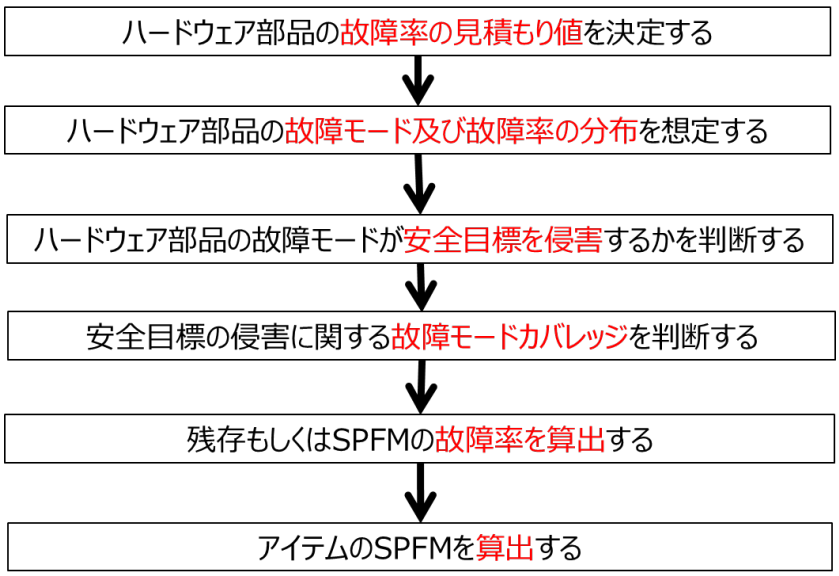


図 82　SPFM の算出例

図 83 に安全コンセプト図におけるダイアグカバレッジの検討例を示す。

表 D.11 – センサー

安全機構/方策	技術の概観を参照	達成可能であるとみなされる典型的なダイアグカバレッジ	備考
入力比較/多数決	D.2.6.5	高	診断テスト間隔内でデータフローが変更する場合のみ

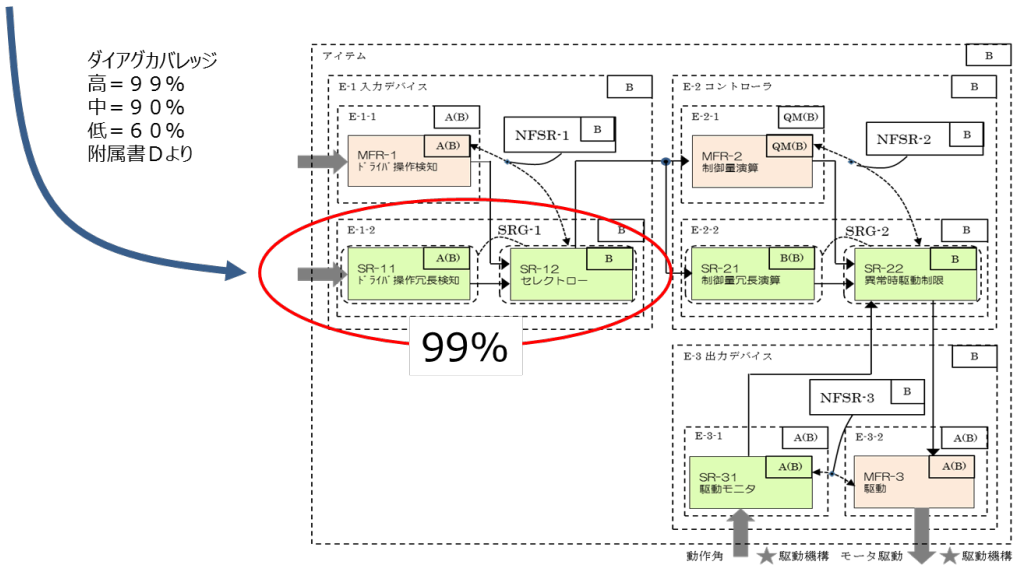


図 83　安全コンセプト図におけるダイアグカバレッジ（安全機構カバレッジ）の表現

B.3.6 独立要求の配慮

ハードウェア部品のコストや省スペース化などの理由により、ハードウェア構成に設計制約が課せられることは少なくない。上流のシステムレベルにおけるデコンポジションを継承してハードウェア設計をする際に、例えば、図 84 に示すようなシステム構成に対して、なんらかの理由で二つのセンサを同一品（A 社）のものにしなければならない場合の独立要求の取り扱いについて取り上げる。

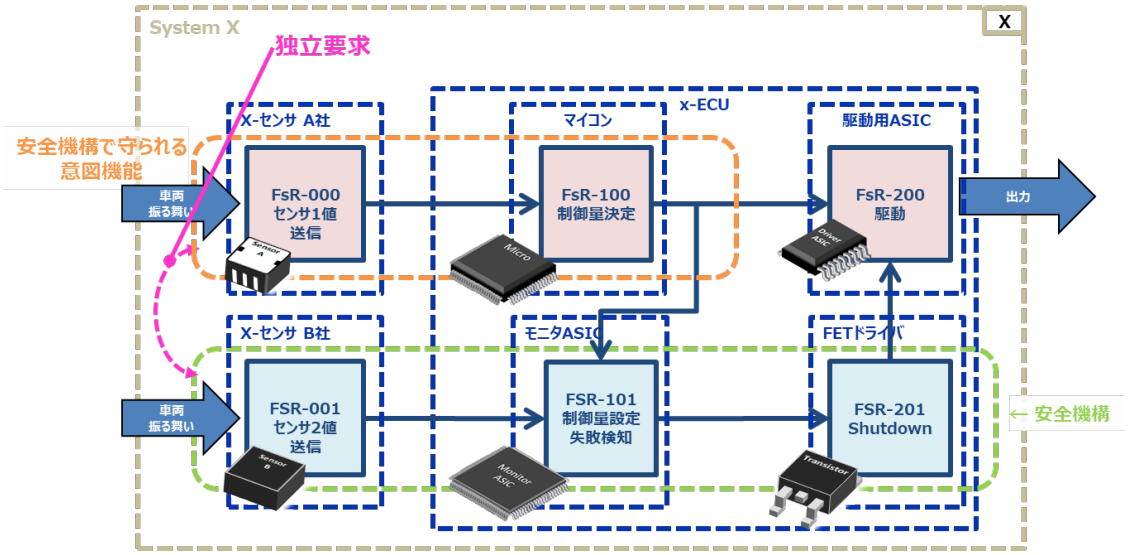


図 84　従来開発品のハードウェア構成

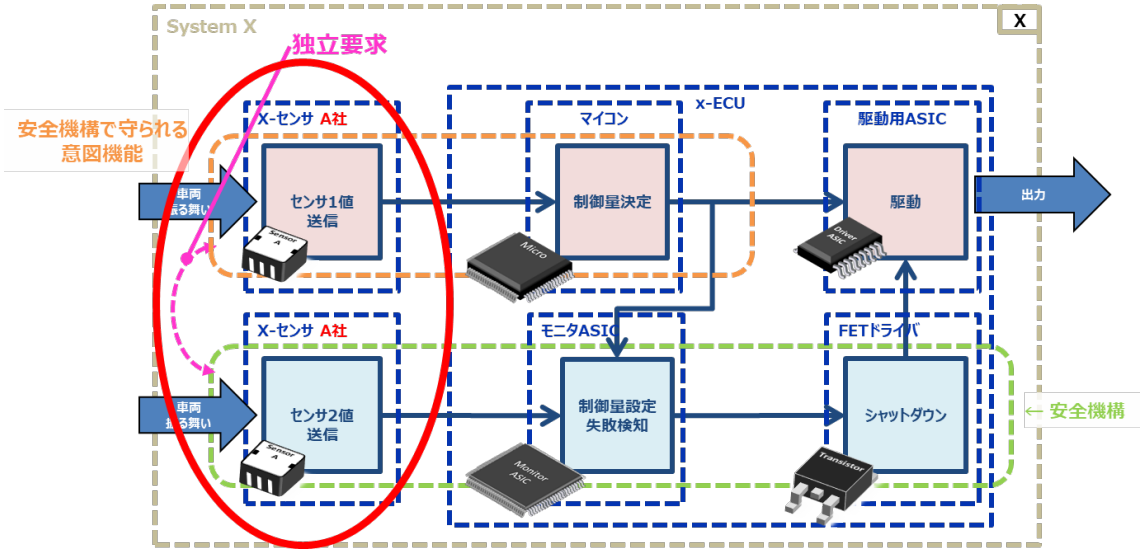


図 85　センサを A 社製に統一

図 85 への変更は、意図機能に対する安全機構のハードウェア冗長性に影響が出るため、機能安全エンジニアは安全設計に対する再検討（安全分析や従属故障分析）が必要となる。例えば、分析の結果は表 21 に示すように整理し、第三者にも説明しやすいように工夫するとよい。

表 21　設計変更に伴う、従属故障分析の再検討

分析エレメント		分析観点							
第1エレメント	第2エレメント	a）ランダムハードウェア故障	b）開発フォールト	c）製造フォールト	d）インストールフォールト	e）修理フォールト	f）環境要因	g）共通外部リソースの故障	h）特定状況での過負荷
センサ1値送信	センサ2値送信								
	制御量設定失敗検知								
	シャットダウン								
制御量決定	センサ2値送信								
	制御量設定失敗検知								
	シャットダウン								
		例 （マイクロコントローラ，ASICなどのような）大規模集積回路におけるクロック，テストロジック，内部降圧回路のような共通ブロックの故障	例 要求フォールト，設計フォールト，実施フォールト、新技術使用の結果としてのフォールト，改良時にとり込まれるフォールト	例 プロセス，手順、訓練に関連するフォールト，コントロールプランや特別な特性を監視するために関するフォールト、ソフトウェア書き換えやライン端のプログラミングに関連するフォールト	例 ワイヤハーネスの引き回しに関連するフォールト，交換可能部品に関連するフォールト，隣接するアイテム又はエレメントの故障	例 プロセス，手順，訓練に関連するフォールト，故障点検に関連するフォールト，交換可能部品に関連するフォールト、過去への非互換性によるフォールト	例 温度，振動，圧力，湿度/結露，公害，腐食，汚染，EMCなど	例 パワーサプライ，入力データ，システム間のデータバス，通信	例 磨耗，経年劣化

B.3.7 従属故障

従属故障分析では、起こりうる従属故障（共通原因故障およびカスケード故障）を洗い出し、それらの従属故障に対して最善な対策を立て、安全性を説明できるようにしたい。そこで、必要十分な従属故障の故障モードを抽出するためには、FTA、HAZOP、FMEA などの様々な分析手法を併用することが有用となる。その際、SCDL はこれらの従属故障分析に必要なアーキテクチャ（要求がエレメントに配置された構造）を提供する手法として位置付けられる。また、ISO 26262-9:2011 7 節には従属故障分析をする上での観点が記述されているので参考にするとよい。

従属故障対策は安全性だけでなく、実現性や競争力の観点なども含めて総合的に最善策を決定する。例えば、従属故障の共通原因故障対策は、同質冗長（ホモジーニアス型）よりも異質冗長(ヘテロジーニアス型)の方が、高い安全性を確保する可能性があるが、異なるメーカや異なる型式の部品使用は実現性や競争力を低下させる可能性もある。したがって、多面的な視点で必要十分な最善の対策を選択することが重要である。

図 14 に従属故障分析の課題と戦略を示す。従属故障の抽出や従属故障対策の有効性説明に、SCDL は有用である。

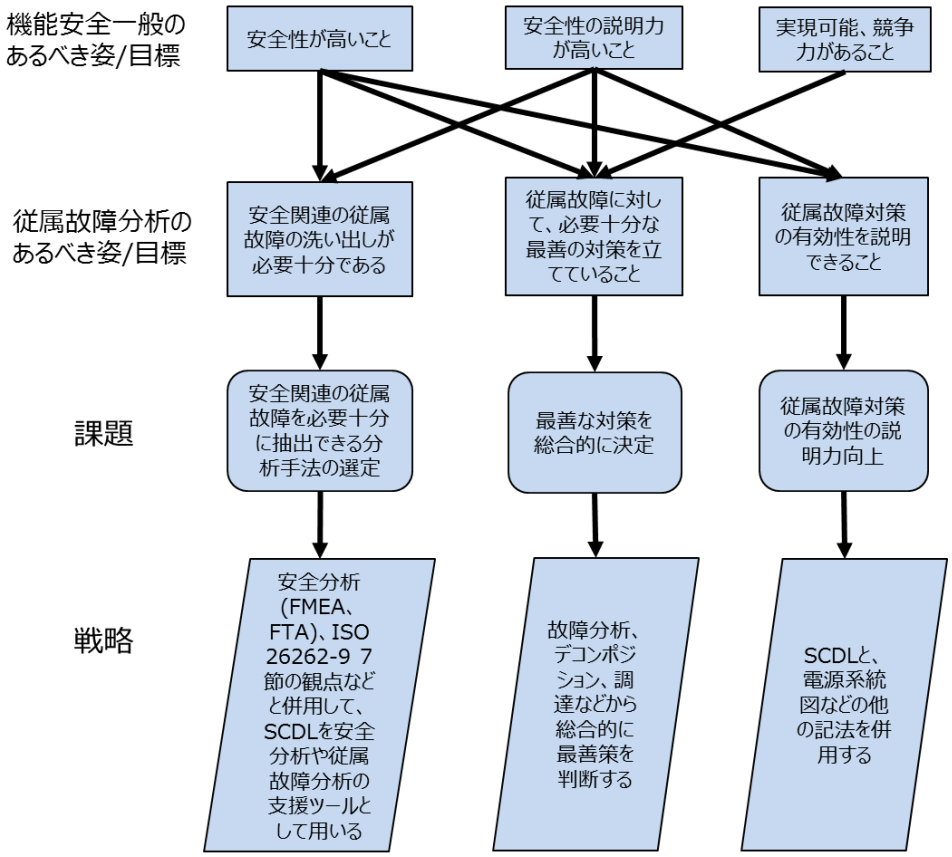


図 86　従属故障分析の課題と戦略

B.3.8 ハードウェア部品認定

ハードウェア構成の ECU を例としてハードウェア部品認定について述べる（図 87）。ECU は、センサからの電気信号を入力処理する回路（入力に関するエレメント）、マイコンと呼ばれる演算およびメモリから構成されるチップ（制御に関するエレメント）および、モータまたはリレーなどを駆動する出力回路（出力に関するエレメント）など、多くの部品により構成されている。

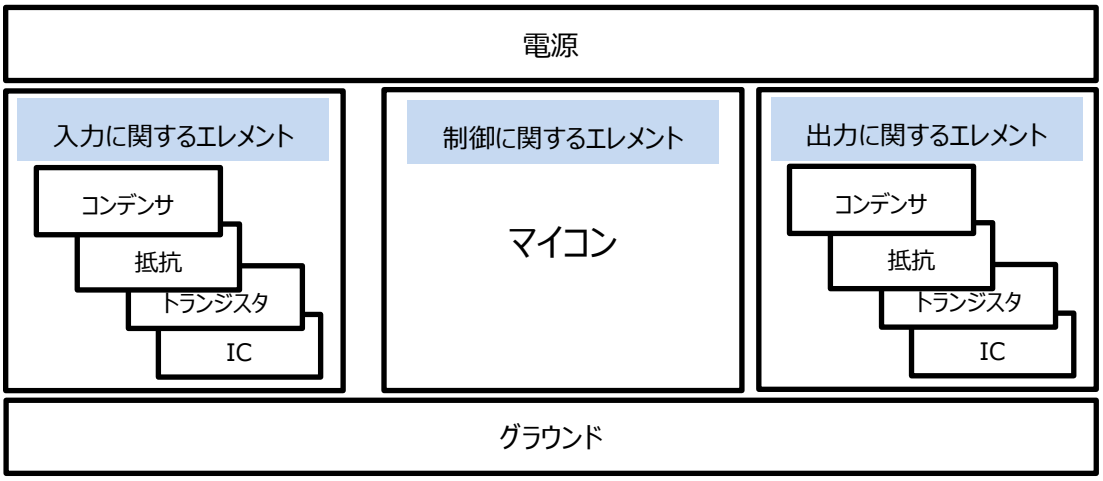


図 87　ハードウェア構成の例

ECU など車載電子装置は、一般民生品と比べると高い品質と耐久性が求められている。車両メーカ各社は、ECU に対して JASO などにあるような負荷試験および合格基準を設定し、ECU の設計および製造メーカに実施を求めている。ECU の設計および製造メーカのハードウェア設計においては、実績のある部品や標準化され試験仕様を満たす（社内基準にて認定された）部品を採用することが前提となっている。図 88 にハードウェアの素子およびコンポーネント認定の流れの例を示す。

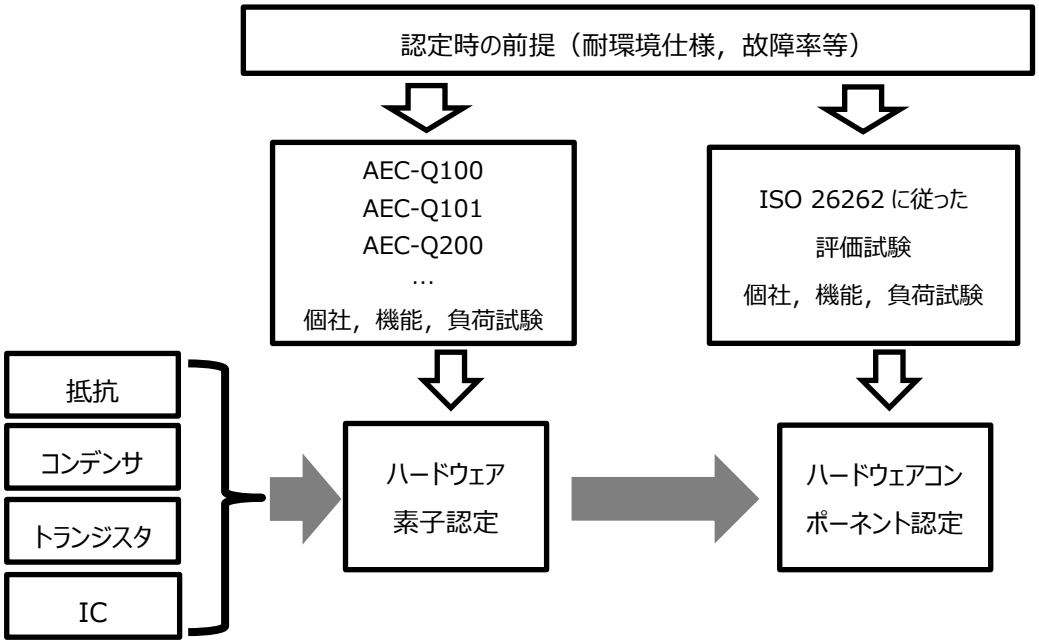


図 88　ハードウェア部品認定の流れの例

ISO 26262:2011 においては、エレメント（素子、コンポーネント）単位でのハードウェアコンポーネント認定の規定がある。コンポーネント認定は、一般的には再利用性を考慮し、設計、評価の工数削減につなげる。

例えば、ハードウェアアーキテクチャ上で、エレメントごとにハードウェア部品認定済であるか否かを明確にすることで、認定済エレメントの単体評価を割愛することが可能となるため、ハードウェア全体の検証計画策定時の有用な情報となる。ただし、ハードウェア部品認定時の前提への適合性を確認する必要がある。

B.4 ユースケース

B. 3 に記述した項目を考慮しながら、図 73 の技術安全コンセプトからハードウェア安全コンセプトに至る安全アーキテクチャのユースケースを紹介する。

なお、ここで紹介するユースケースは、初期段階において、意図機能に対し ASIL を付与するアプローチを例として選択した。

本ユースケースでは、最終的に図 89 のようなエレメント全体構成で示される 4 階層の構成要素を最上位層の Sys01 から順に示していく。

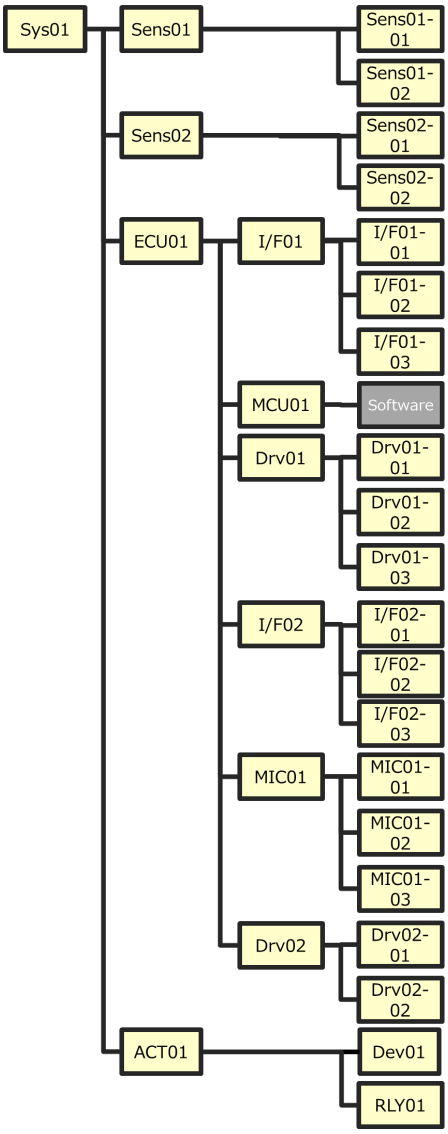


図 89　エレメント全体構成

B.4.1 ゴールコンセプト～機能安全コンセプト

技術安全コンセプトからハードウェア安全コンセプトに至る安全アーキテクチャを検討するにあたり、入力情報として、アイテム、安全目標、機能安全コンセプトを定義する。

アイテム：車両挙動（X）によってアクチュエータ制御（Z）する車両制御システム Sys01（アイテムの境界を定義、図 90）

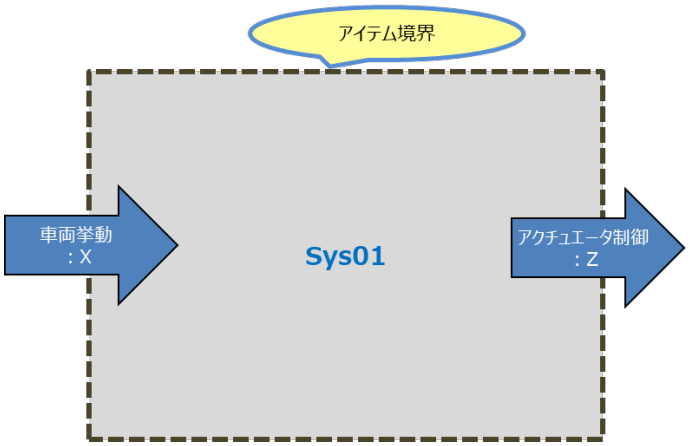


図 90　アイテム定義

安全目標 SG01：ドライバの意図しない過大な出力を発生させない（ASIL D）
安全状態 SS01：出力停止状態

アイテムは、図 91 のようなエレメント構成である。これを SCDL で表現すると、図 92 のようなエレメント構造図となる。各エレメントの説明を表 22 に示す。

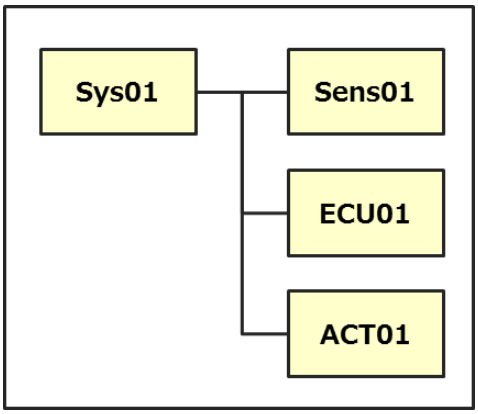


図 91　エレメント構成

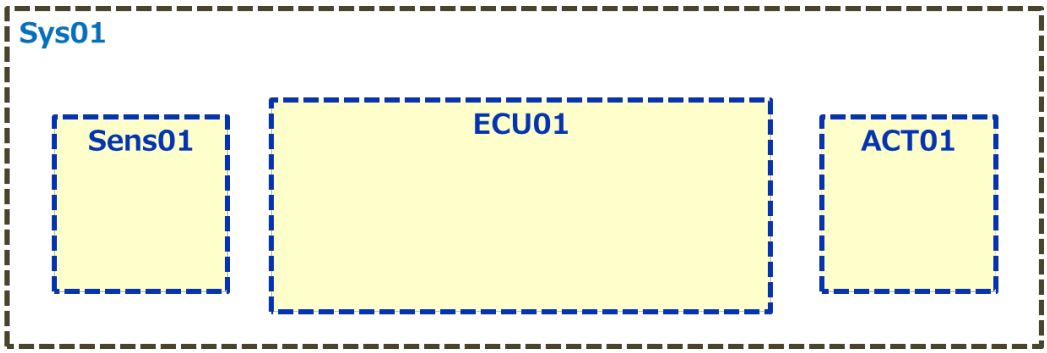


図 92　アイテムのエレメント構造図

表 22　エレメントの説明

ID	名称	説明
Sys01	システム 01	車両挙動パラメータ X によって ACT01 を制御するシステム
Sens01	センサ 01	車両挙動を検知して、ECU01 に送る
ECU01	電子制御ユニット 01	Sens01 からの入力から制御量を決定し ACT01 に出力する電子制御ユニット
ACT01	アクチュエータ 01	エネルギーをアクチュエータ制御 Z に変換する駆動装置

意図機能の SG01 侵害についての安全分析を実施した結果としての機能安全要求（以下、意図機能由来の機能安全要求）を表 23 に示す。また、図 21 に要求構造を、図 94 に意図機能由来の機能安全要求をエレメントに配置した要求配置図を示す。

表 23　意図機能由来の機能安全要求

ID	名称	内容
FsR101	センシング	車両の挙動を正しく検知し、物理量を正しく変換して、FsR102 に正しく送ること
FsR102	制御量決定	FsR101 から正しく受け取った値をもとに、制御量を正しく算出して、制御量を FsR103 に正しく送ること
FsR103	制御量出力	FsR102 より正しく受信した制御量を FsR104 に正しく出力すること
FsR104	駆動	FsR103 から正しく受け取った物理量にて正しく駆動すること

※FsRxxx：意図機能由来の安全要求

ここでいう「正しく」とは、「上位安全要求を満足するよう」という意味である。

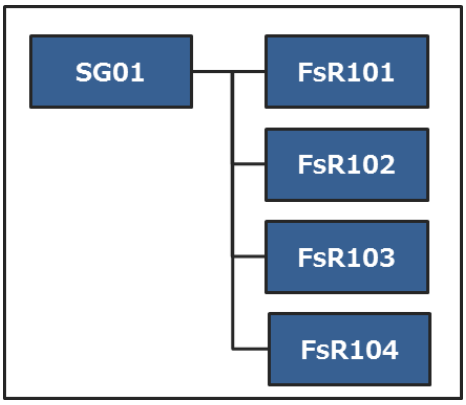


図 93　要求構造

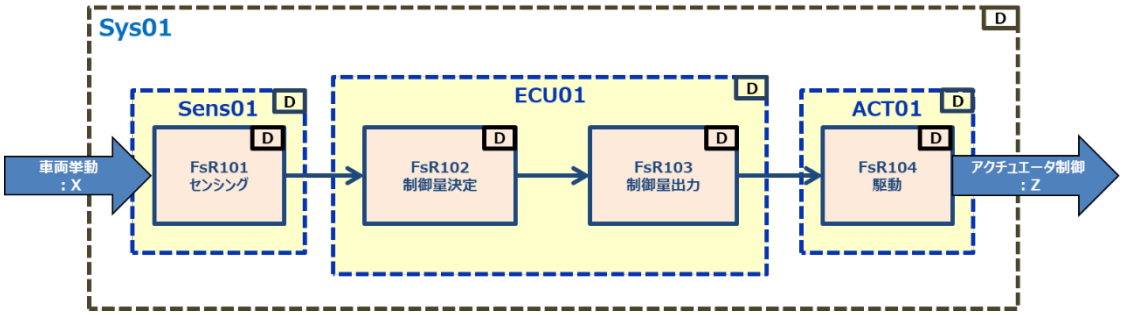


図 94　意図機能由来の機能安全要求をエレメントに配置した要求配置図

アーキテクチャおよび機能安全要求に基づき、安全分析（トップダウン、ボトムアップ）を行って安全機構の必要性、有効性を検討し、機能安全コンセプトを作成する。

本ユースケースでは、図 94 のような入力されるセンサ値を含めた制御結果の正当性を確認し、安全目標を達成するシステムを検討する。この段階からハードウェア設計戦略が、アーキテクチャに考慮される場合もある。

最初にシステム全体の安全性を確保するために、冗長性を踏まえて評価する。この際に用いられるのが、前項で紹介した

- 1.3.6 独立要求の配慮
- 1.3.7 従属故障分析

などの項目になる。これらの項目にしたがってデコンポジション、従属故障分析を取り入れて検討を進める。

まず、例として図 94 を基に図 95 のような帰納的分析を実施し、安全目標 SG01 を達成するための安全機構の検討を進める。

部位	内容	故障モード（ガイドワード/経験値）	原因・要因	安全目標SG１侵害可能性	ASIL	対策	妥当性その他要件	備考
FsR101 センシング	車両の挙動を正しく検知し、物理量を正しく変換して、FsR102に正しく送ること	出力が不正など（信号遅れ、出力小さい、大きいなど。）	-センサ破壊（EMC、ESDなど） -センサ製造不良など	Yes	D	-SM1:FSR111センシング2とFSR112制御量失敗検知を追加して、FSR111とFsR103からくる制御量をFSR112で比較して、不正を検知する。不正な出力量の時には、FSR113によって出力を停止する。	-下位レベルの安全分析にて確認。 -共倒れ対策のために、独立要求が必要	
FsR102 制御量決定	FsR101から正しく受け取った値をもとに、制御量を正しく算出して、制御量をFsR103に正しく送ること	制御量の決定値が不正など（遅れ、決定値がハンチング、異常値出力）	-入力が不正 -この機能上でのシステマチック故障。（ソフトウェアのバグ、仕様間違いなど） -関連部分のランダムハードウェア故障	Yes	D			
FsR103 制御量出力	FsR102より正しく受信した制御量をFsR104に正しく出力すること（ECU01）	制御量の出力値が不正など（遅れ、出力値がハンチング、出力不足、固着など）	-入力が不正 -この機能上でのシステマチック故障。（ソフトウェアのバグ、仕様間違いなど）	Yes	D			
FsR104 駆動	FsR103から正しく受け取った物理量にて正しく駆動すること	駆動量の不正など（意図しない量の出力、固着で駆動できない、）	-入力が不正 -駆動用のHWの破壊。（製造不良、ESD破壊など）	Yes	D	-SM２：FSR111センシング2とFSR112制御量失敗検知を追加して、FsR104の駆動量をFSR111でモニターしその駆動量とFsR103からくる制御量をFSR112で比較して、意図した制御量が、FsR104から駆動されていない場合、FSR113によって出力を停止する。	-下位レベルの安全分析にて確認。 -共倒れ対策のために、独立要求が必要	

SM1

SM２

図 95　帰納的分析の例

図 95 の帰納的分析によって安全機構 SM1 および SM2 が導き出されるが、本ユースケースでは、SCDL の使用例を示すことが目的であるため、安全機構 SM1 を代表例として取り上げて、詳細化していく。

図 96 に、安全機構 SM1 の実装のために追加された機能安全要求(FSR111、FSR112、FSR113)を示す。

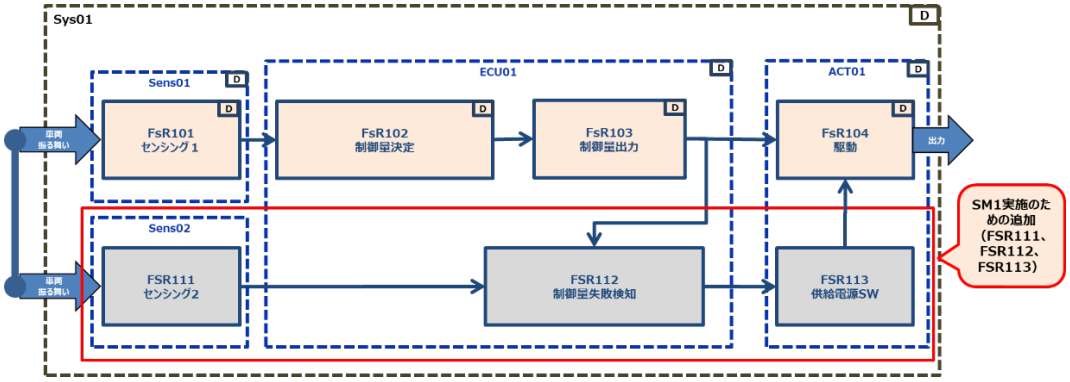


図 96　SM1 実装のために追加された機能安全要求

以上のような安全分析を経て、図 97、表 24、表 25 に示すように、機能安全コンセプトが構成される。

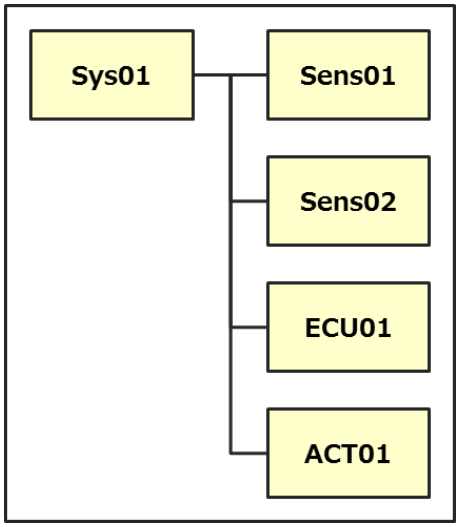


図 97　機能安全エレメント構成

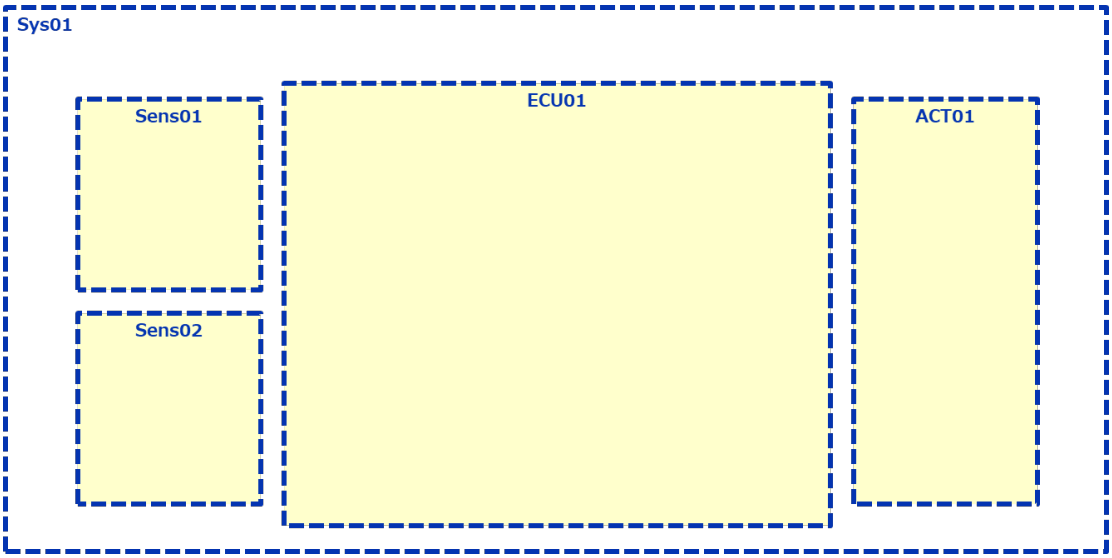


図 98　機能安全エレメント構造図

表 24　エレメントの説明

ID	名称	説明
Sys01	システム 01	車両挙動 X によって ACT01 を制御するシステム
Sens01	センサ 01	車両の挙動を検知して、ECU01 に送る
Sens02	センサ 02	車両の挙動を検知して、ECU01 に送る
ECU01	電子制御ユニット 01	Sens01/Sens02 の入力から制御量を決定し ACT01 に出力する電子制御ユニット
ACT01	アクチュエータ 01	エネルギーをアクチュエータ制御 Z に変換する駆動装置と供給電源停止用のスイッチを持つユニット

表 25　機能安全要求（安全分析後）

ID	名称	内容
FsR101	センシング 1	車両の挙動を正しく検知し、物理量を正しく変換して、FsR102 に正しく送信する
FsR102	制御量決定	FsR101 から正しく受け取った値をもとに、制御量を正しく算出して、制御量を FsR103 に正しく送る
FsR103	制御量出力	FsR102 より正しく受信した制御量を ACT01 に正しく出力する
FsR104	駆動	正しく物理量にして駆動する
FSR111	センシング 2	車両の挙動を正しく検知し、物理量を正しく変換して、FSR112 に正しく送信する
FSR112	制御量失敗検知	FsR102 の制御量が正しいかを判断するために、FSR111 の値と FsR102 の算出ロジックと同等な処置によって、期待値である正しい制御量を導出し、FsR102 の制御量を FsR103 制御量出力を通して比較をおこなう FsR102 の制御量が安全範囲を超えている場合に FSR113 に出力停止信号を正しく送る
FSR113	供給電源 SW	FsR104 の出力を停止させるために、FSR112 から出力停止信号を受信した場合、FsR104 への電源供給をやめ、出力を停止させる

※FsRxxx：意図機能由来の安全要求、FSRxxx：安全機構の安全要求

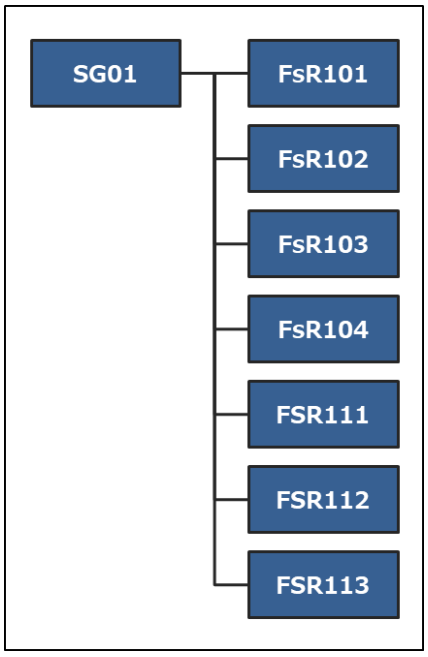


図 99　要求構造　（安全分析後）

ここで，安全機構 SM1 が共通原因故障およびカスケード故障によって失陥した場合に安全目標 SG01 を侵害することを防ぐため、デコンポジション戦略を取り入れ、戦略の実現に必要となる独立要求 NFSRxxx をシステム Sys01 に配置する。

表 26　機能安全要求（独立要求追加後）

ID	名称	内容
FsR101	センシング 1	車両の挙動を正しく検知し、物理量を正しく変換して、FsR102 に正しく送信する
FsR102	制御量決定	FsR101 から正しく受け取った値をもとに、制御量を正しく算出して、制御量を FsR103 に正しく送る
FsR103	制御量出力	FsR102 より正しく受信した制御量を ACT01 に正しく出力する
FsR104	駆動	正しく物理量にして駆動する
FSR111	センシング 2	車両の挙動を正しく検知し、物理量を正しく変換して、FSR112 に正しく送信する
FSR112	制御量失敗検知	FsR102 の制御量が正しいかを判断するために、FSR111 の値と FsR102 の算出ロジックと同等な処置によって、期待値である正しい制御量を導出し FsR102 の制御量を FsR103 制御量出力を通して比較をおこなう FsR102 の制御量が安全範囲を超えている場合に FSR113 に出力停止信号を正しく送る
FSR113	供給電源 SW	FsR104 の出力を停止させるために、FSR112 から出力停止信号を受信した場合、FsR104 への電源供給をやめ、出力を停止させる
NFSRxxx	独立要求	意図機能由来の機能安全要求（FsR101、FsR102、FsR103）と安全機構 SM1（FSR111、FSR112、FSR113）の共通原因故障およびカスケード故障なきこと

※NFSR：非機能要求。xxx については、表 27 参照のこと。

表 26 の要求を図 98 の機能安全エレメントに配置し、デコンポジションスキームにしたがって重み付けを分解すると、図 100 機能安全コンセプト図が定義できる。

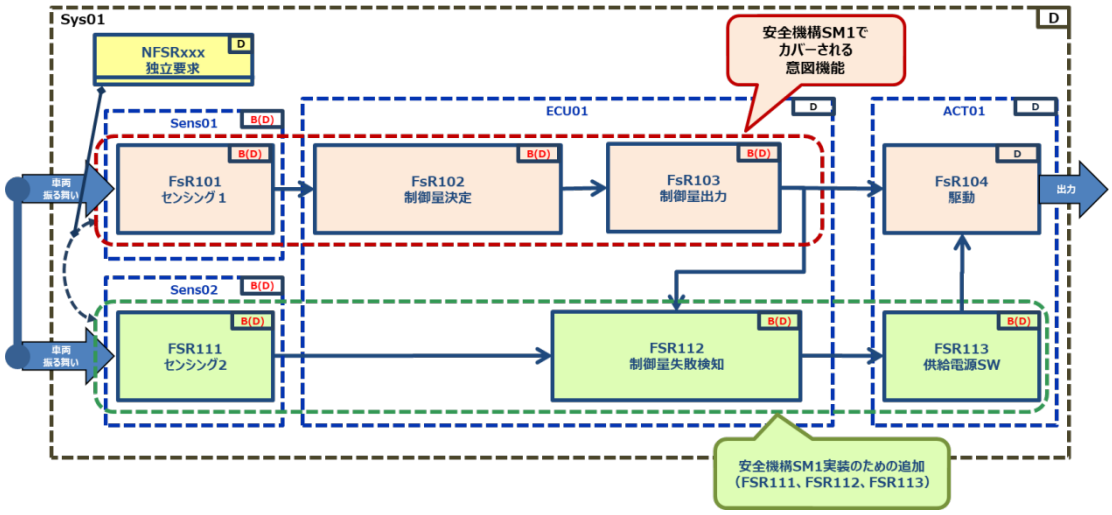


図 100　機能安全コンセプト図
（ペールオレンジ：意図機能由来の機能安全要求、
グリーン：安全機構、イエロー：独立要求）

この例では、ASIL D を ASIL B(D)と ASIL B(D)に分解した。

機能安全コンセプト図を作成後、従属故障分析を実施する。表 27 は独立要求の展開例である。この例では独立要求がある安全要求の組み合わせの一覧を作成し、各々の組み合わせについての従属故障分析を実施している。

表 27　独立要求の展開

NFSRxxx

	FSR111	FSR112	FSR113
FsR101	NFSR001	NFSR002	NFSR003
FsR102	NFSR101	NFSR102	NFSR103
FsR103	NFSR201	NFSR202	NFSR203

図 101 は独立要求の組み合わせのそれぞれを機能安全コンセプト図に配置した例である。この例では独立要求 NFSR001 / 002 / 003 / 101 / 103 / 201 / 203 がアイテムの Sys01 上に、NFSR102 / 202 が ECU01 上に配置されている。このように SCDL で記載することにより、独立要求のエレメントの相関関係およびその独立要求がどこに配置されるかを明示できる。

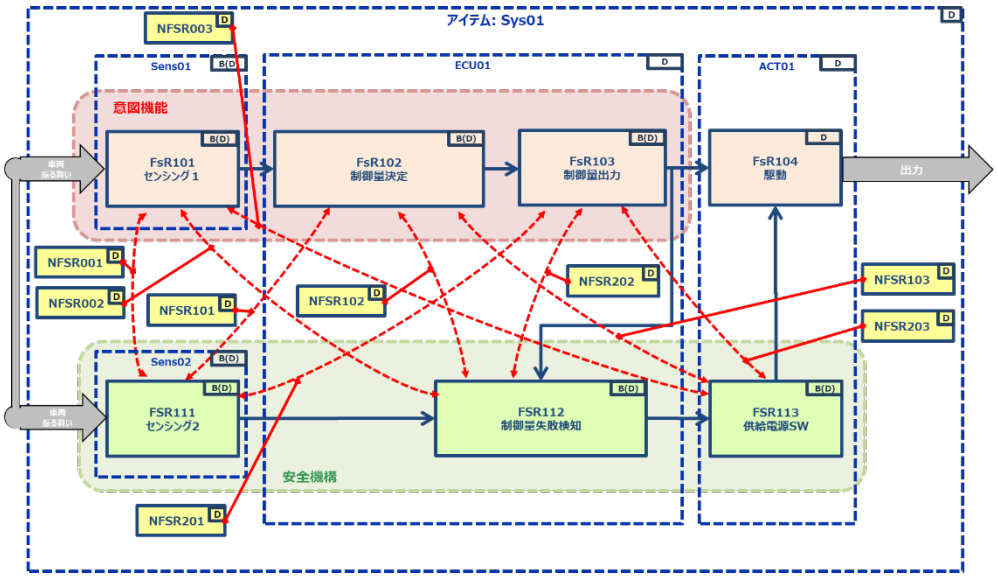


図 101　機能安全コンセプトの独立要求 NFSRxxx の分解例（機能安全コンセプト図）

図 102 は FsR101 と FSR111 間の従属故障分析を実施し、独立要求 NFSR001 実現のための対策を導出した例である。FsR101 と FSR111 間の独立要求は、アイテムに配置されるため、図 102 のように従属故障分析をおこなう。

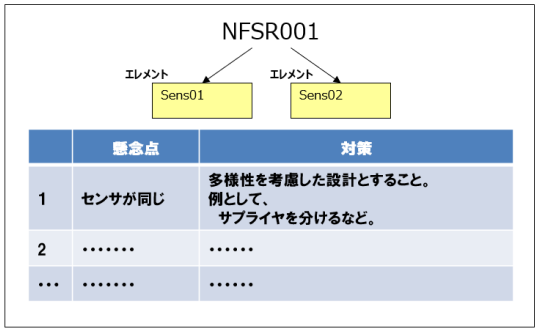


図 102　独立要求 NFSR001 の例

その他の独立要求についても同ように、従属故障分析をおこなう。図 103 は ECU 内に配置される NFSR102、NFSR202 の例である。

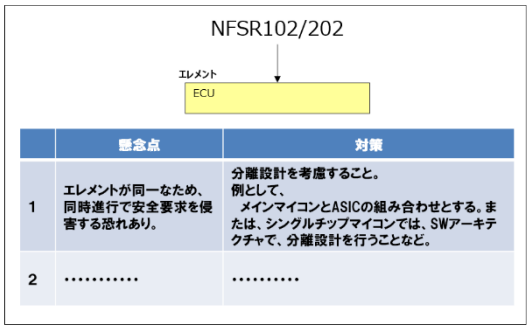


図 103　独立要求、その他例

従属故障分析の結果、エレメントの分離設計や安全機構の多様性を考慮することを技術安全コンセプトの検討に申し送り、新たに安全要求を追加する必要がないことを確認し、図 101 の機能安全コンセプト図を確定する。

B.4.2 技術安全要求

上位の機能安全要求に基づき、技術安全要求を導出する。
技術安全要求を導出する際に重要なのは、下位のレベルで実現可能な実装を考慮することである。　例えば、個社が持っている既存の製品技術、または軸となる開発品の派生技術などを基に、ある程度のハードウェアの構成を想定する。センサ技術はどのようなものとするのか、マイコンにはどのような機能およびどのくらいの容量が必要なのか、駆動部分はアクチュエータに合わせてどのようなドライバを使うのかなどを検討する。

この際に用いられるのが、前項で紹介した

- B.3.2 搭載スペース
- B.3.3 調達
- B.3.4 新規／流用
- B.3.8 ハードウェア部品認定

などの項目になる。

開発上のリスクを最小化するために、図 104 に示すようにマイコンなどの主要部品については、すでに調達ルートが確保されていることがあり、認定された部品メーカのラインナップから、ROM/RAM 容量などを踏まえて選択するのが一般的である。　また、マイコンの Pin 数、パッケージなどは、搭載スペースを考慮して、選択される。

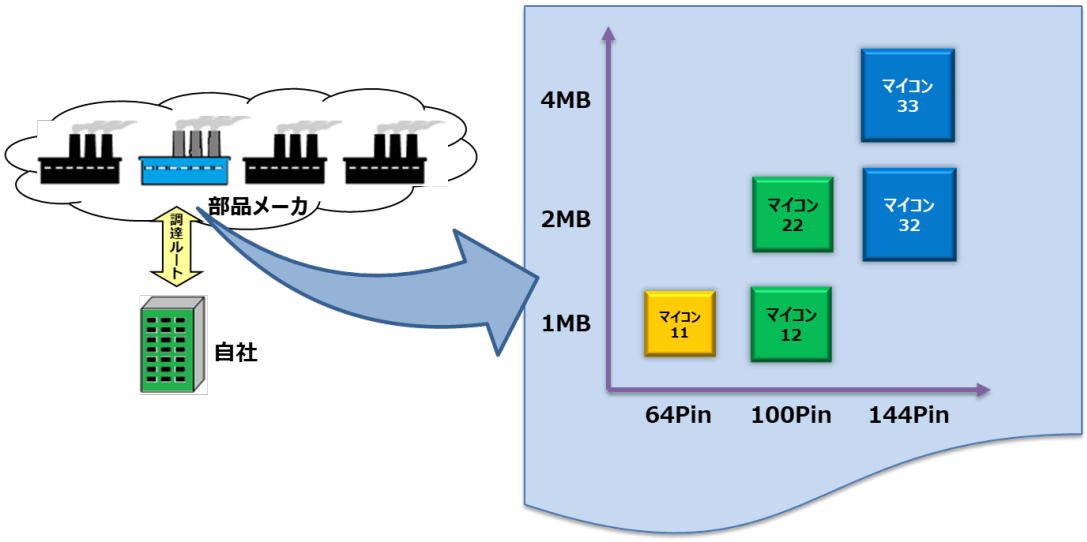


図 104　調達ルートとラインナップ

また、個社にはすでに量産したハードウェア設計資産がエレメントごとに存在することもあるため、図 105 に示すように設計者はその中からインタフェースエレメントなど、汎用性の高いブロックの流用を考慮することも有用である。

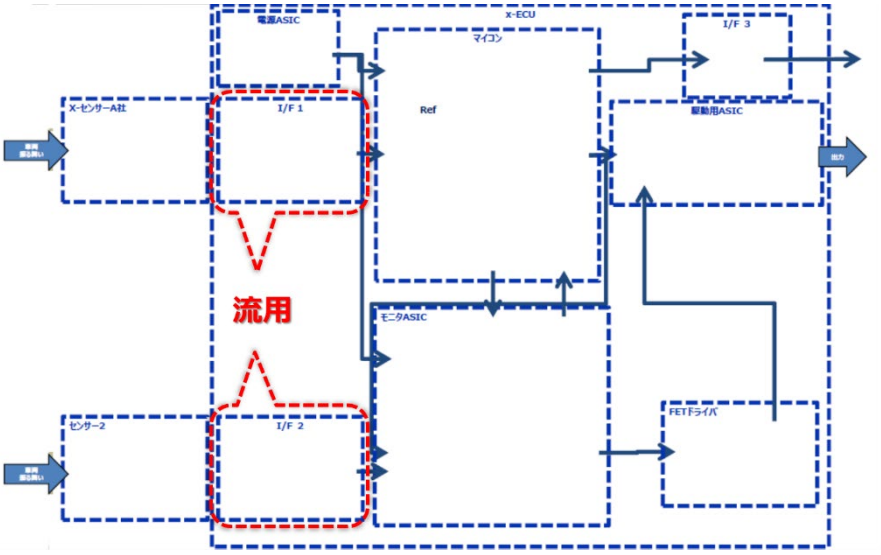


図 105　既存のハードウェア設計資産の流用検討

これらの情報を基に、機能安全コンセプトから、エレメント情報、技術要求を含んだ技術安全要求を作成し、エレメントに配置する（流用したセンサにはイニシャルチェック機構が存在しているので、レイテント要求を配置できないか検討する）。例えば、表 28 機能安全要求 FsR101 センシング 1 は、図 106 の想定されるセンサ技術を基に表 29 および図 107 のような技術安全要求に展開される。

表 28　機能安全要求＜センシング 1 ＞

ID	名称	内容
FsR101	センシング 1	車両の振る舞いを正しく検知し、物理量を正しく変換して、FsR102 に正しく送信する

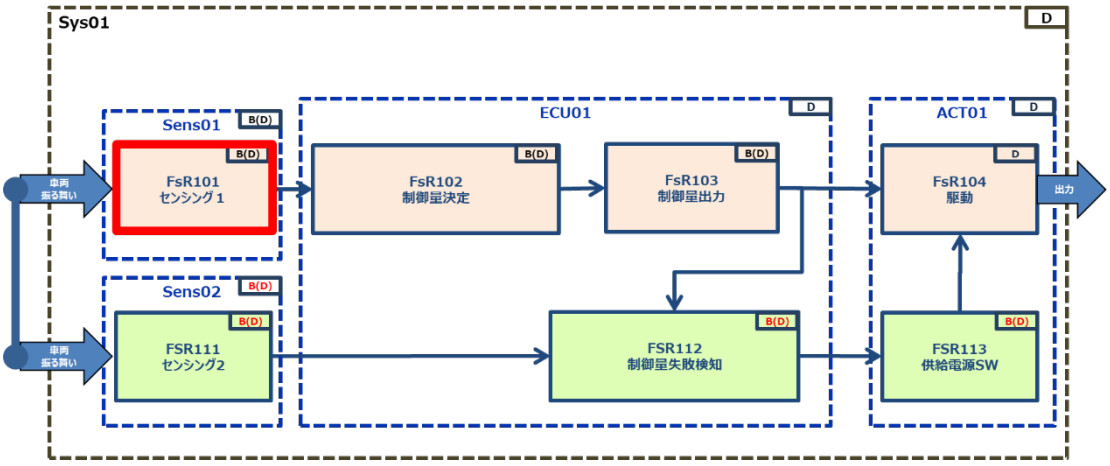


図 106　機能安全要求 FsR101 センシング１

表 29　技術安全要求＜センシング 1 ＞

ID	名称	内容
TsR101-1-1	センシング 1	Sens01 は車両挙動にしたがって正しく静電容量を変化させ、TsR-101-1-2 へ正しく送信すること
TsR101-1-2	変換 1	Sens01 は信号変換器を設けて、静電容量を正しく電圧に変換し、TsR101-1-3 へ正しく送信すること
TsR101-1-3	デジタイズ 1	Sens01 は A/D 変換により、正しくデジタル量に変換し、TsR101-1-4 へ正しく送信すること
TsR101-1-4	コミュニケーション 1	Sens01 はデジタル量をシリアル通信によって TsR101-1-5 へ正しく送信すること
TsR101-1-5	センシングダイアグ 1	Sens01 の異常を正しく監視して、センシング部異常の場合は、故障状態であることを ECU01 に正しく知らせるため（イニシャル時）TsR102-1-1 正しく送信すること

※TsRxxx：意図機能由来の技術安全要求

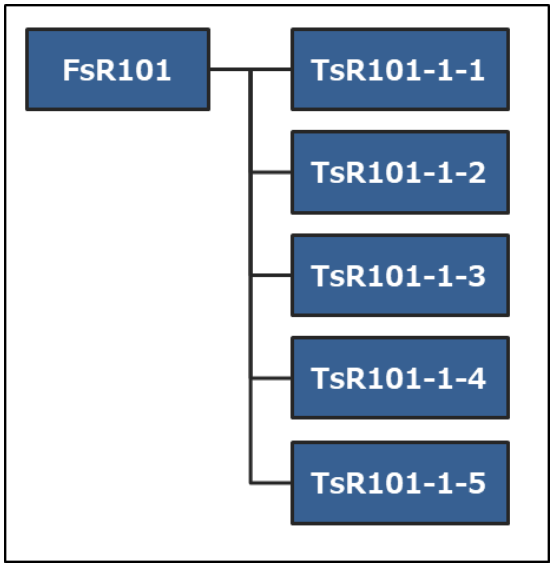


図 107　要求構造（FsR101 を技術安全要求に展開）

このように機能安全要求を実装可能な技術安全要求として導出していく。前述のように、技術安全要求を導出する際に重要なことは、下位のレベルで実現可能な実装を考慮することである。

今回の場合、SCDL で記載された機能安全コンセプトに基づいて、実現可能な実装を意識してレビューすると、表 30 に示すように FSR112<制御量失敗検知>の実現方法のアイデアが出た。

表 30　FSR112<制御量失敗検知>

ID	名称	内容
FSR112	制御量失敗検知	FsR102 の制御量が正しいかを判断するために、FSR111 の値と FsR102 の算出ロジックと同等な処置によって、期待値である正しい制御量を導出し FsR102 の制御量との比較をおこなう FsR102 の制御量が安全範囲を超えている場合に FSR113 に出力停止信号を正しく送る

図 108 では、ECU01 から ACT01 への出力をモニタリングし、制御量失敗検知をしている。この場合のインタラクションは、ハードウェアの実装を考慮すると、ACT01 のドライバの出力をモニタリングすることになり、ソフトウェアの処理に負担がかかることが指摘された。また、ECU01 の外部 I/F から近く、外乱影響の保護対策が必要になることが予想されることも指摘された。

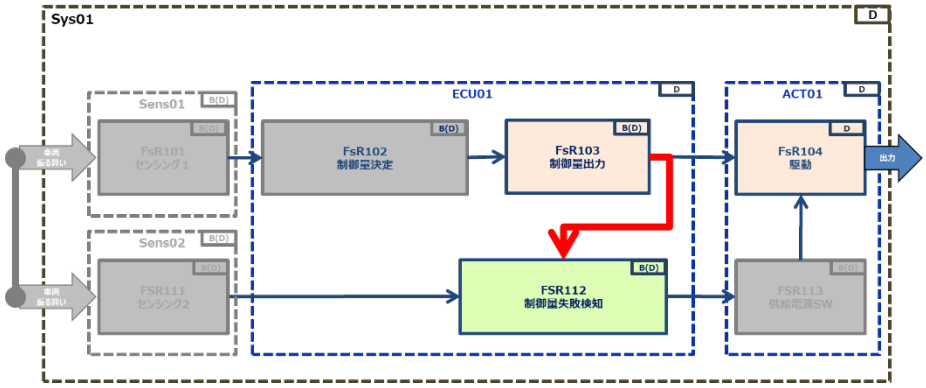


図 108　機能安全コンセプト　安全機構　SM1

そこで改善案として、図 109 のようにモニタリングするインタラクションを変更することにした。

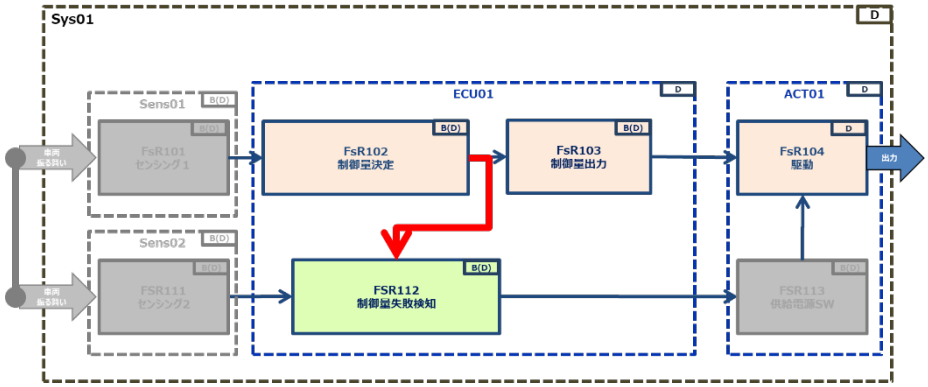


図 109　機能安全コンセプト　安全機構　SM1 変更後

この場合、ECU01 の内部で情報を送るため、実際の制御量をモニタリングできる。また、懸念された外乱の影響は受けにくくなる。例えばシリアル通信などを用いれば、デジタル量でのモニタリングが可能になる。懸念点としては、FsR103 が SM1 の対象から外れるので、安全目標の達成のためには他の安全機構での担保が必要になる。

この新しい機能安全コンセプトに基づいて、再度、機能安全要求として帰納的分析を更新する。

部位	内容	故障モード（ガイドワード/経験値）	原因・要因	安全目標SG1侵害可能性	ASIL	対策	妥当性その他要件	備考
FsR101 センシング	車両の挙動を正しく検知し、物理量を正しく変換して、FsR102に正しく送ること	出力が不正など（信号遅れ、出力小さい、大きいなど。）	-センサ破壊（EMC、ESDなど） -センサ製造不良など	Yes	D	-SM1:FSR111センシング2とFSR112制御量失敗検知を追加して、FSR111とFsR102からくる制御量をFSR112で比較して、不正を検知する。不正な出力量の時には、FSR113によって出力を停止する。	-下位レベルの安全分析にて確認。 -共倒れ対策のために、独立要求が必要	
FsR102 制御量決定	FsR101から正しく受け取った値をもとに、制御量を正しく算出して、制御量をFsR103に正しく送ること	制御量の決定値が不正など（遅れ、決定値がハンチング、異常値出力）	-入力が不正 -この機能上でのシステマチック故障。（ソフトウェアのバグ、仕様間違いなど） -関連部分のランダムハードウェア故障	Yes	D			
FsR103 制御量出力	FsR102より正しく受信した制御量をFsR104に正しく出力すること（ECU01）	制御量の出力値が不正など（遅れ、出力値がハンチング、出力不足、固着など）	-入力が不正 -この機能上でのシステマチック故障。（ソフトウェアのバグ、仕様間違いなど）	Yes	D	-SM２：FSR113を常時Off、FSR112の演算結果から、出力するときのみ、FSR113をOnにしてFsR104に通電して、FsR103/FsR104で不正が起こっても、安全目標を侵害する出力が出ないようにする。	-下位レベルの安全分析にて確認。 -共倒れ対策のために、独立要求が必要。 -SM３：FSR113が必要時にOff→Onにできるように、初期診断時に診断が必要	
FsR104 駆動	FsR103から正しく受け取った物理量にて正しく駆動すること	駆動量の不正など（意図しない量の出力、固着で駆動できない、）	-入力が不正 -駆動用のHWの破壊。（製造不良、ESD破壊など）	Yes	D			

SM1
SM２ 常時
SM３ 初期診断時

図 110　帰納的分析の更新例

SM1 については、モニタの位置を FsR102 に修正した。SM2 は内容を見直し、FsR103 と FsR104 の意図しない出力を防ぐため、その出力源を FSR113 で常時 Off とし、FSR112 が正常と判断したときのみ On にするように変更した。また、FsR104 単独で SG を侵害するリスクがあるため、一旦 ASIL D とし、下位レベルにて確認することとした。

更新された要求を、機能安全エレメントに配置し、デコンポジションスキームにしたがって ASIL の重み付けを分解すると、機能安全コンセプト図が図 111 のように再度定義できる。

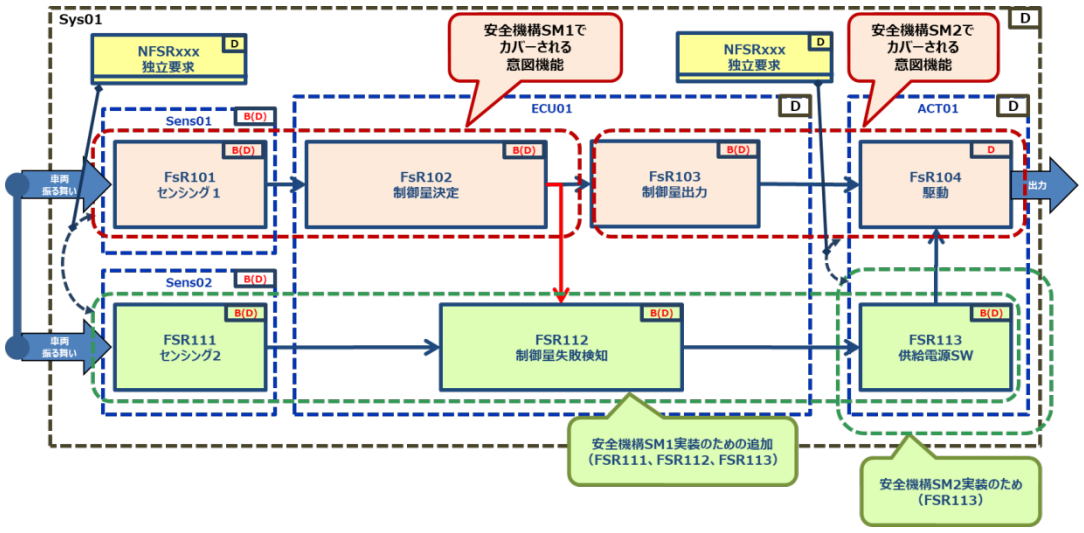


図 111　機能安全コンセプト図

これらの安全分析結果より、安全機構 SM1、SM2 が更新された。またこの分析結果により FSR113 の Off 固着の潜在故障対策のため、安全機構 SM3 を初期診断時に追加することにした。これにより，図 101　機能安全コンセプトの独立要求 NFSRxxx の分解例（機能安全コンセプト図）は，今回更新した SM1/SM2 によって次の図 112 となる。SM3（初期診断時）の表現方法については、0B.4.4 初期診断の表現方法についての一例において示す。

なお、本ユースケースでは、引き続き安全機構 SM1 を代表例として取り上げる。

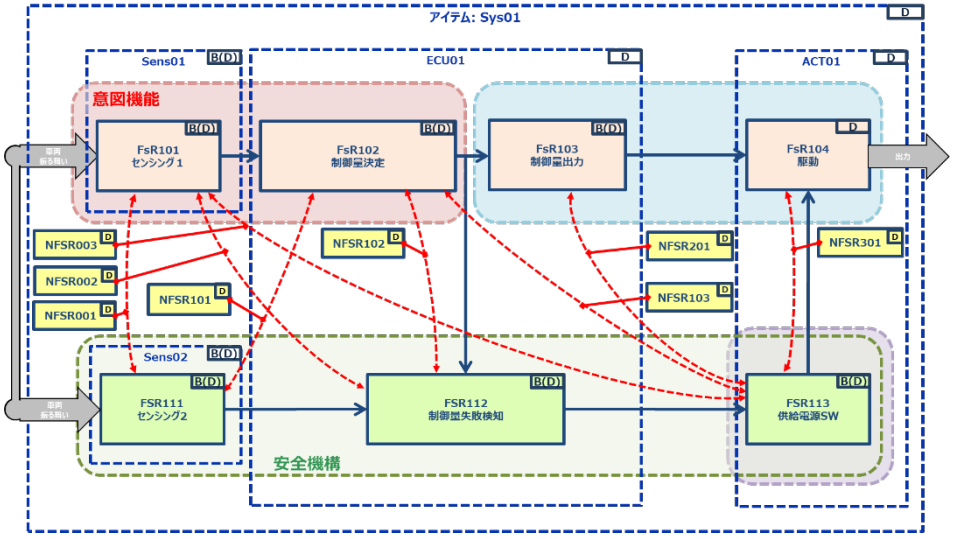


図 112　SM1/SM2 更新後の機能安全コンセプトの独立要求　（機能安全コンセプト図）

この例のように SCDL を用いて、レビューを行って、その安全対策の結果を表現し、開発チームで履歴を共有できることも SCDL の利点であると考える。

この新しい機能安全コンセプト図に基づいて、前述の FsR101 以外の部分も展開するとともに、従属故障分析結果として申し送られた図 30、図 31 の内容も考慮すると、技術安全要求は表 31 および図 113 に示すようになる。そして、エレメントは図 114 および図 115 に示すようになる。

表 31　技術安全要求＜全体：一部抜粋＞

ID	名称	内容
TsR101-1-1	センシング 1	Sens01 は車両挙動にしたがって正しく静電容量を変化させ、TsR101-1-2 へ正しく送信すること
TsR101-1-2	変換 1	Sens01 は信号変換器を設けて、静電容量を正しく電圧に変換し、TsR101-1-3 へ正しく送信すること
TsR101-1-3	デジタイズ 1	Sens01 は A/D 変換により、正しくデジタル量に変換し、TsR101-1-4 へ正しく送信すること
TsR101-1-4	コミュニケーション 1	Sens01 はデジタル量をシリアル通信によって TsR101-1-5 へ正しく送信すること
〃	〃	〃
TSR111-1-1	センシング 2	Sens02 は車両挙動にしたがって正しく静電容量を変化させ TSR111-1-2 へ正しく送信すること
TSR111-1-2	変換 2	Sens02 は信号変換器を設けて、静電容量を正しく電圧に変換し、TSR111-1-3 へ正しく送信すること
TSR111-1-3	デジタイズ 2	Sens02 は A/D 変換により、正しくデジタル量に変換し、TSR111-1-4 へ正しく送信すること
TSR111-1-4	コミュニケーション 2	Sens02 はデジタル量をシリアル通信によって TSR111-1-5 へ正しく送信すること
TSR111-1-5	センシングダイアグ 2	Sens02 の異常を正しく監視して、センサ部が異常の場合には故障状態であることを ECU01 に正しく知らせるため TSR112-1-1 へ正しく送信すること
TSR112-1-1	センサデータ受信 2	Sens02 のデジタルデータを正しく受け取り、TSR112-1-2 へ正しく送信すること
TSR112-1-2	外乱保護 2	ECU01 の外部の外乱から Sens02 のデータを正しく保護し、TSR112-1-3 へ正しく送信すること
TSR112-1-3	データ送信 2	Sens02 のデジタルデータは内部 Bus を通して MIC01 に送るため TSR112-2-1 へ正しく送信すること
TSR112-2-1	データ確認	I/F02 から受け取ったデータの正当性を正しく確認し、

		TSR112-2-2 へ正しく送信すること
TSR112-2-2	制御量決定 2	Sens02 のデータを基に、あらかじめ設定された制御量を正しく決定し、TSR112-3-1 へ正しく送信すること
TSR112-3-1	制御量検証	MCU01 の制御量と TSR112-2-2 の制御量を正しく比較し TSR112-3-2 へ正しく送信すること
TSR112-3-2	制御時操作	比較結果が許容値を 3 秒超えた場合には、Drv01 出力停止信号を、TSR113-1-1 へ正しく送信すること
TSR113-1-1	Drv02 駆動信号受信	TSR112-3-2 の信号を正しく受け取り、その情報に基づいて TSR113-1-2 を正しく駆動させること
TSR113-1-2	アクチュエータ出力	TSR113-1-1 の信号を正しく受け取り、TSR114-1-1 を正しく駆動すること
TSR114-1-1	電源供給スイッチ	TSR113-1-2 の信号を正しく受け取り、TsR104-1-1 への電源供給を正しく実施すること
NTSR00x	独立要求	意図機能由来の技術安全要求（Sens01 と I/F01、MCU01）と安全機構（Sens02、I/F02、MIC01、Drv02、ACT01）は独立していることを保証すること

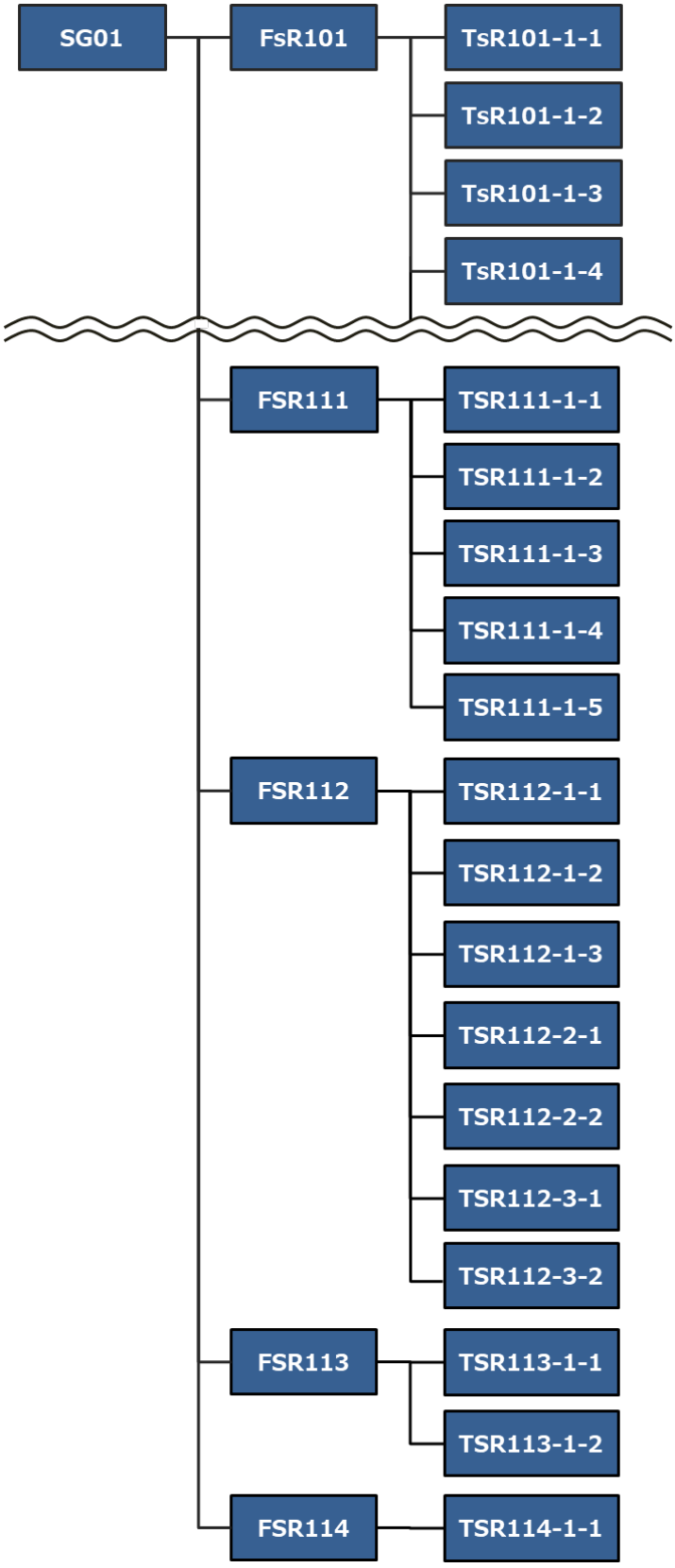


図 113　要求構造（全体：一部抜粋）

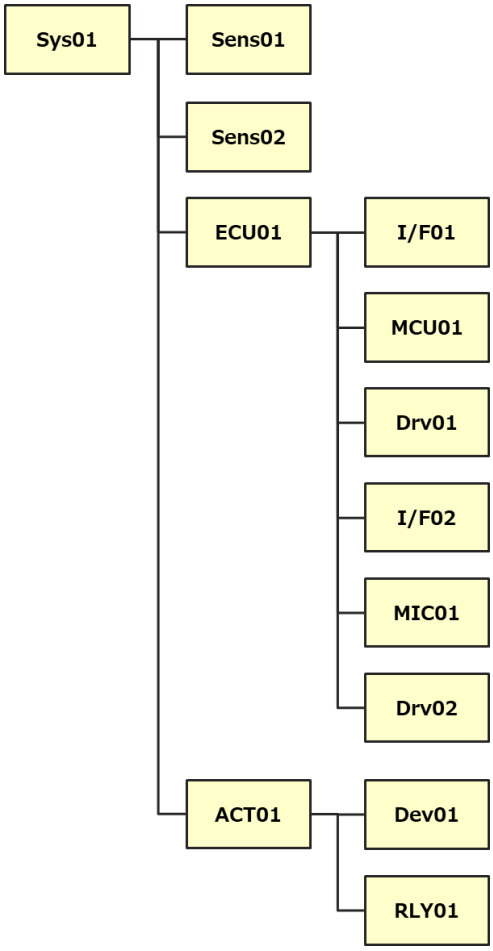


図 114　技術安全エレメント構成

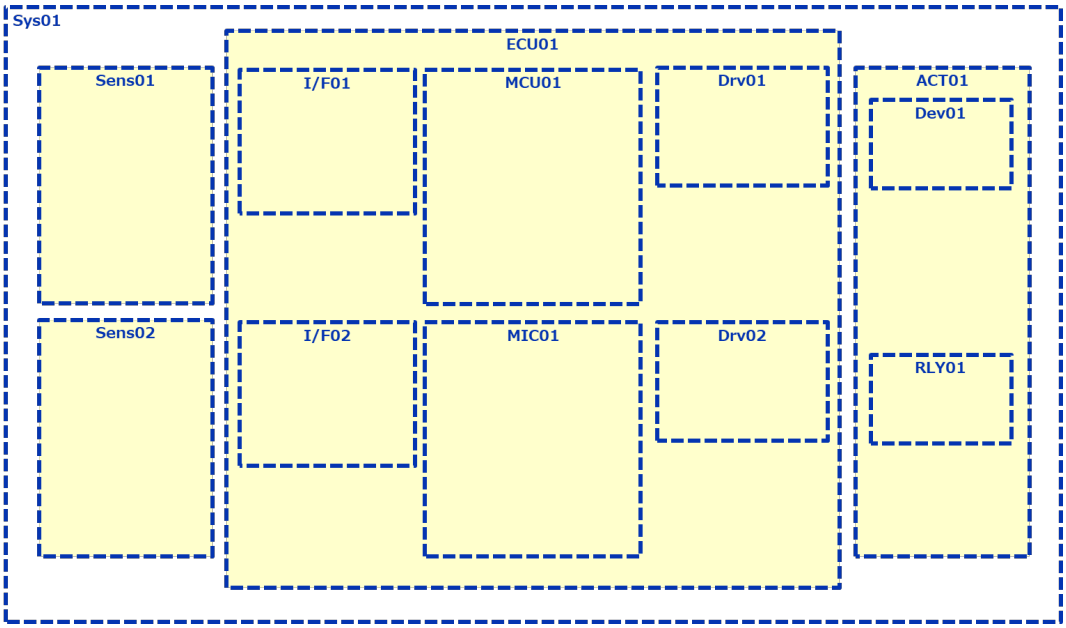


図 115　技術安全エレメント構造図

B.4.3 技術安全要求の配置および技術安全コンセプトの検証

図 114 および図 115 にてアーキテクチャを細分化したので、アーキテクチャに基づいた安全分析を実施して、機能安全コンセプトの段階でおこなった安全分析（ボトムアップ/トップダウン）結果を更新していく。表 31 にて導出された技術安全要求が、構造化され、エレメントに配置されると、図 116 に示す技術安全コンセプト図となる。

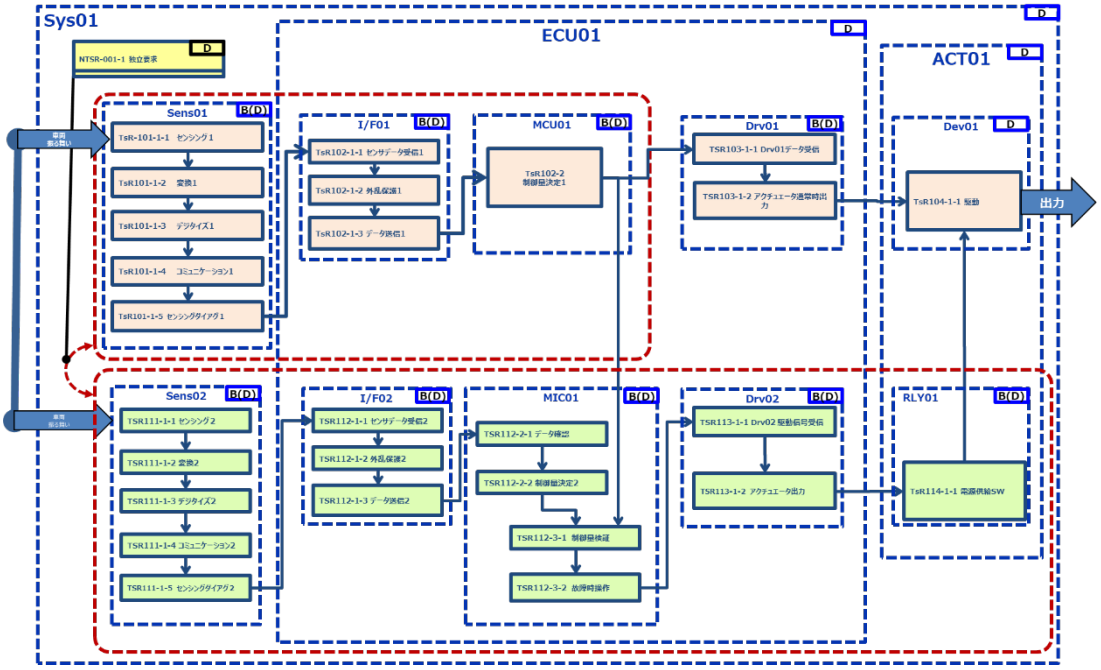


図 116　技術安全コンセプト図
（ペールオレンジ：意図機能由来の技術安全要求、
グリーン：安全機構、イエロー：独立要求）

SCDL は要求間のインタラクションを明確にすることができるので、その情報を使って技術安全要求に基づいた帰納的分析（例：FMEA）をおこなうことができる。例えば、安全要求から出ている SCDL のインタラクション線ごとに、安全要求を HAZOP のガイドワードや個社の過去の教訓をもとに帰納的分析を行い、図 117、図 118 に示すように一覧表にまとめることができる。また、意図機能、安全機能（安全機構）をアーキテクチャとして表現することができるので、アイテムの安全目標侵害について、演繹的分析（例：FTA）を行い、安全機構で意図機能が守られているかどうかを容易に確認することができる。その分析結果に基づき、最小カットセット数の検証、共通原因故障などの分析もおこなうことができる。

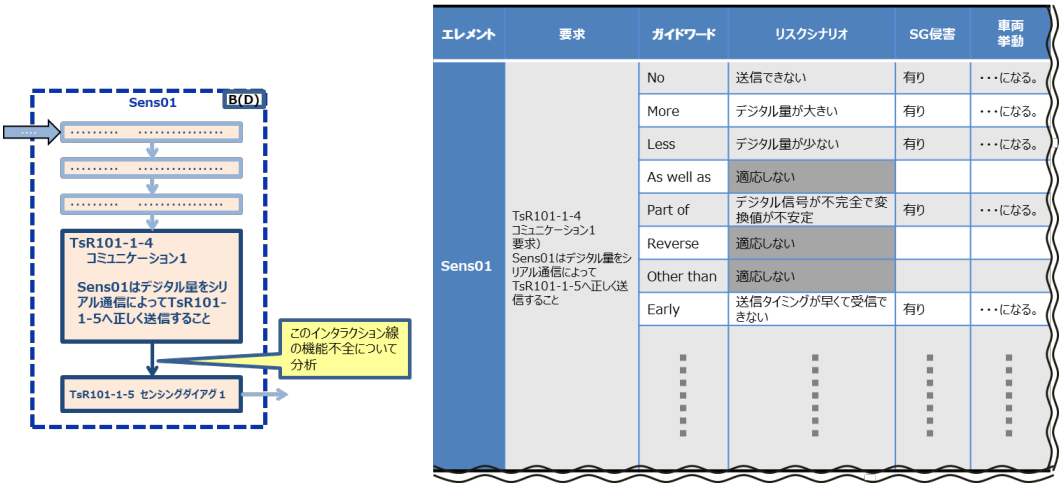


図 117　HAZOP ガイドワードによる SCDL のインタラクション線を使った帰納的分析のイメージ

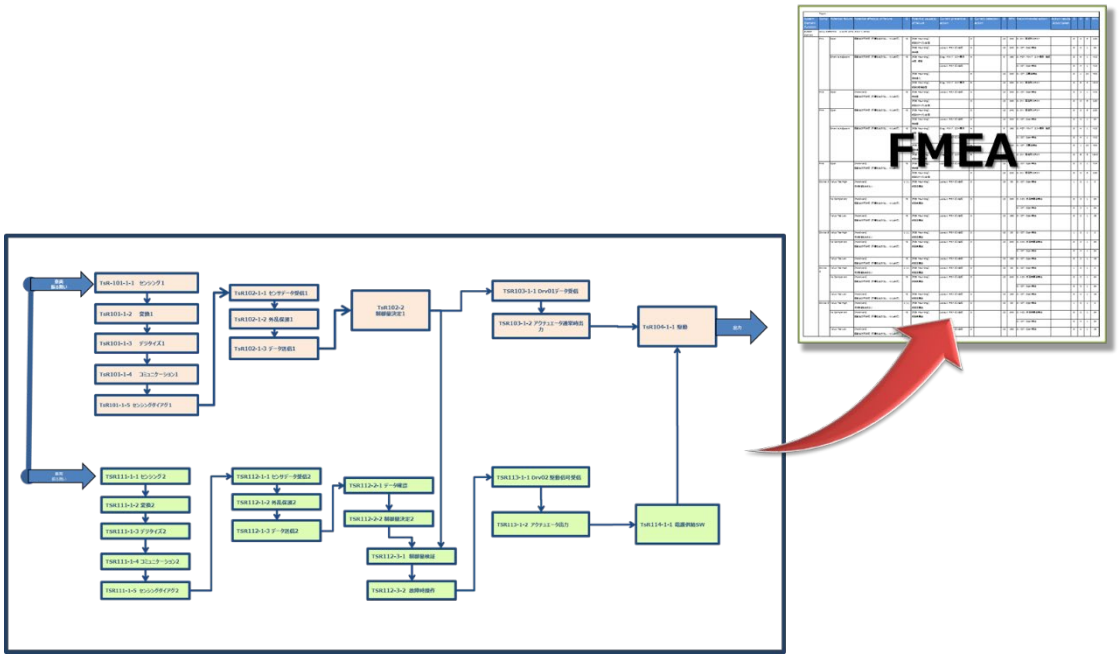


図 118　SCDL の安全要求インタラクションを使った帰納的分析イメージ

機能安全コンセプトでおこなった従属故障分析（表 27）にて、考慮点として抽出された要求やテストの検証結果などを技術安全コンセプトに引き継ぎ、さらに従属故障分析を実施して検証結果を更新する。すなわち、機能安全要求での従属故障分析を継承し、技術安全要求での従属故障分析をおこなう。例として、安全機構 SM1 の ECU 内部の独立要求の詳細化の一例を図 119 に示す。

NFSRxxx分解	FSR111	FSR112
FsR101	NFSR001	NFSR002
FsR102	NFSR101	NFSR102

NFSR102 分解	TSR112-1-x	TSR112-2-x	TSR113-1-x
TsR102-1-x	NTSR102-1	NTSR102-2	NTSR102-3
TsR102-2-x	NTSR102-4	NTSR102-5	NTSR102-6

図 119　独立要求 NFSR102 の分解例

図 119 では上位の要求 NFSR102 にしたがって、独立要求が技術安全要求レベルに適切に分解されている。SCDL を適用した図 116 の技術安全コンセプト図では、ECU 内の構造がエレメント表記で示されているため、独立要求の対象として分離設計を考慮すべきエレメントの組み合わせを特定することができる。図 120 に独立要求 NFSR102 の例を示す。このように SCDL を用いれば、ASIL の重み付けを継承しながら次の実装レベル（ハードウェア）の安全設計アーキテクチャを導出できる。

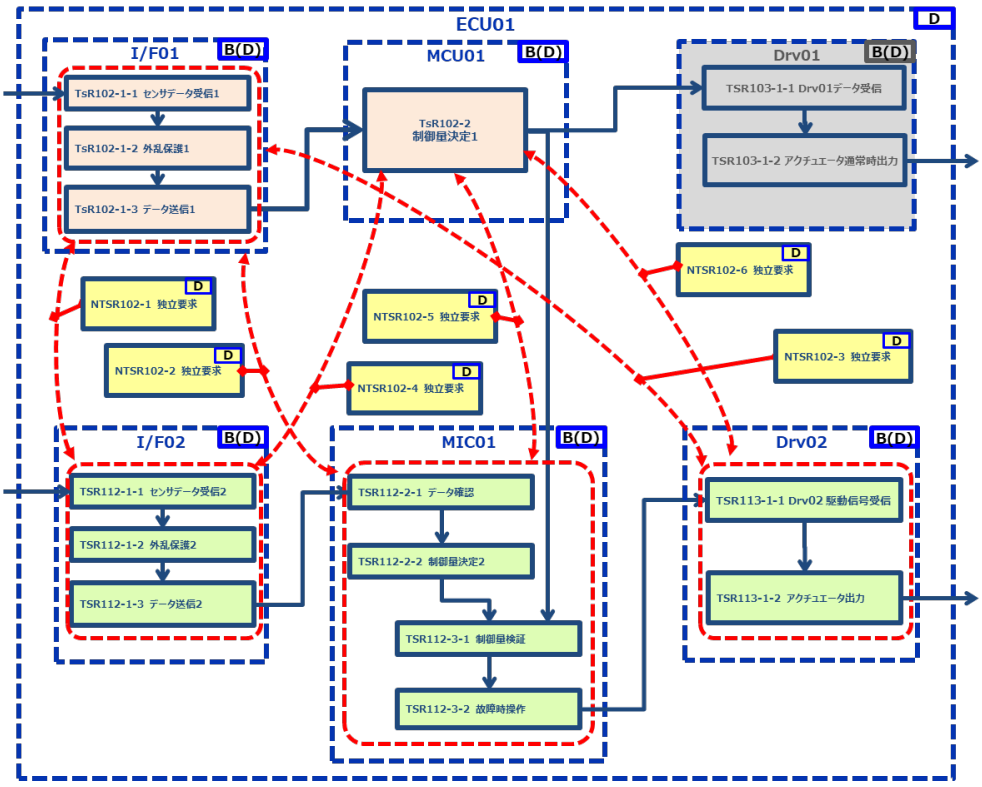


図 120　機能安全コンセプトの独立要求 NFSR102 の分解例（技術安全コンセプト図）

上記のように従属故障分析の結果から、エレメントの分離設計や安全機構の多様性を考慮することをハードウェア設計の検討者に申し送り、新たに安全要求を追加する必要がないことを確認し、図 116 の技術安全コンセプト図を確定する。技術安全コンセプトを受けて、ハードウェア安全要求導出の前段階で、機能安全における安全設計の定量評価のための目標値を設定し、過去の実績を踏まえた設計および検証の戦略を立ててハードウェアの評価をおこなう。このアイテムの安全目標 SG01 は前述のように ASIL D である。この例ではアイテムとシステム Sys01 は同一であるので、その目標値は分配されることなく、そのまま Sys01 の目標値となる。すなわちハードウェアアーキテクチャメトリック評価の目標値は、

Single Point Fault Metrics ≧ 99%

Latent Fault Metrics ≧ 90%

ランダムハードウェア故障による安全目標侵害の評価（PMHF）の目標値は、

$<10^{-8}$h （10FIT）

となる。

目標値は安全目標ごとにエレメントに分配される。この例では安全目標が一つなので PMHF の ASIL D の合計目標値 10FIT は、図 121 に示すように Sens01:A FIT, Sens02:B FIT, ECU:C FIT, ACT01:D FIT に分配される。

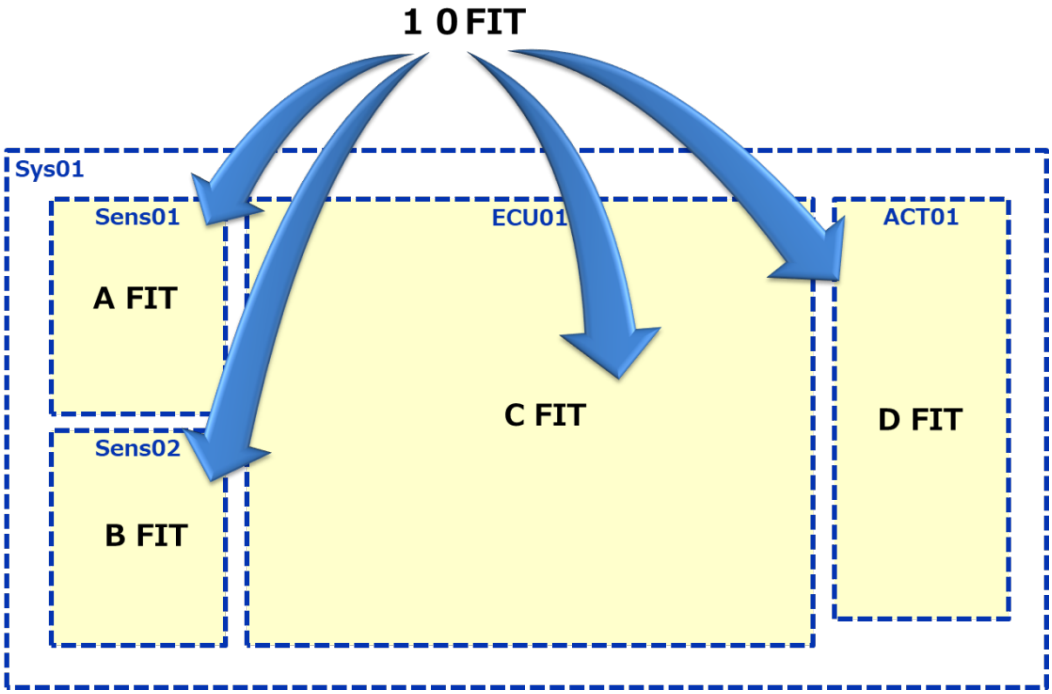


図 121 PMHF 目標値の分配例

図 115 に示すように ECU の内部には複数のエレメントが存在し、その目標値を達成するためにそれぞれのエレメントに分配されるため、図 121 は図 122 に示すようになる。なお、今回、2nd-method は取り扱わない。

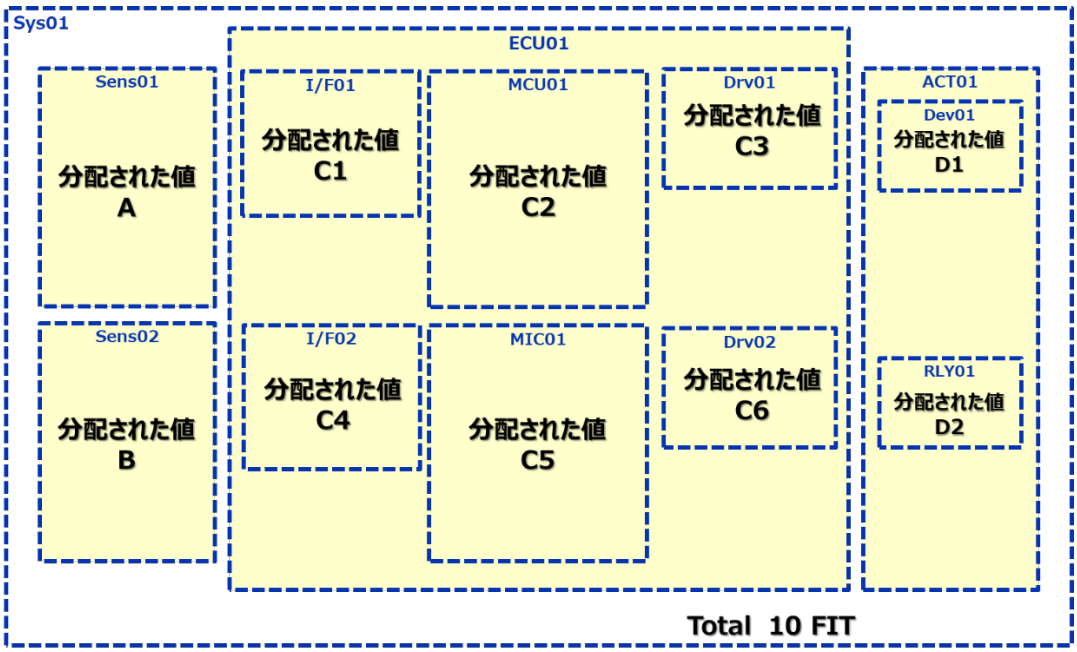


図 122　SCDL のエレメントを使った PMHF 目標値の分配例

本編の実装事例では、マイコンに配置可能な IF（意図機能）と SM（安全機構）を別のマイコン（またはハードウェア）に配置した。機能安全要求における IF と SM との間の独立要求を、技術安全要求にそのまま展開し、実装されるマイコン（またはハードウェア）を独立にした事例である。しかし、機能安全要求における IF と SM との間の独立要求を満たしつつ、IF と SM を 1 つのマイコン上に配置することも可能である。

B.4.4 初期診断の表現方法についての一例

ここで、安全機構 SM3 をもとに初期診断の表現方法についての一例を挙げる。通常、車両に搭載されるユニットは、始動時に初期診断を実施して、通常状態に検出できない潜在故障を検出する。それを他の表現方法と合わせて SCDL を使用する例を紹介する。

エレメント Drv01 およびエレメント Drv02 が正しく機能しない場合に、Sys01 からの不正な出力を防止し、安全状態に移行させるためのエレメント Drv02 を図 123 に示し、車両の始動時に ECU 内部で故障診断をおこなう例を記載する。Sys01 の状態遷移を図 124 に示し、エレメント Drv02 をどのように診断するかについて、フローチャートを使った設計を図 125 に示す。

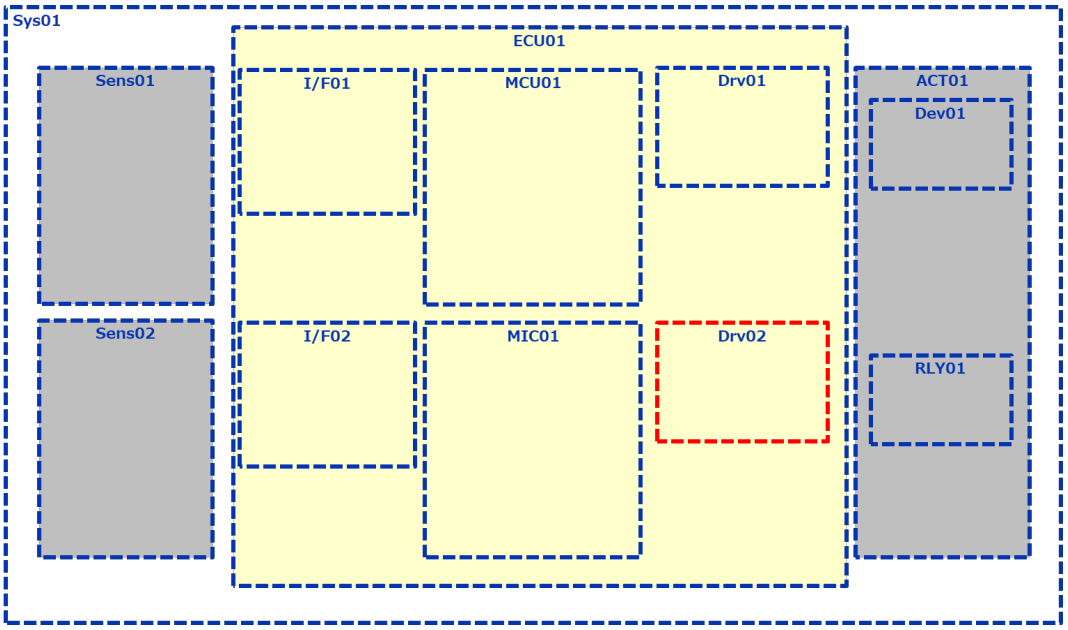


図 123　故障時に安全状態に移行するためのエレメント Drv02

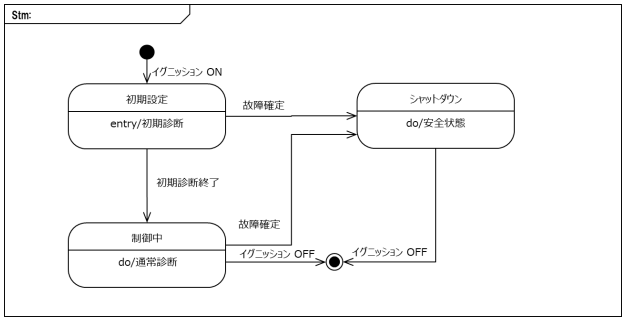


図 124　ステートマシン図

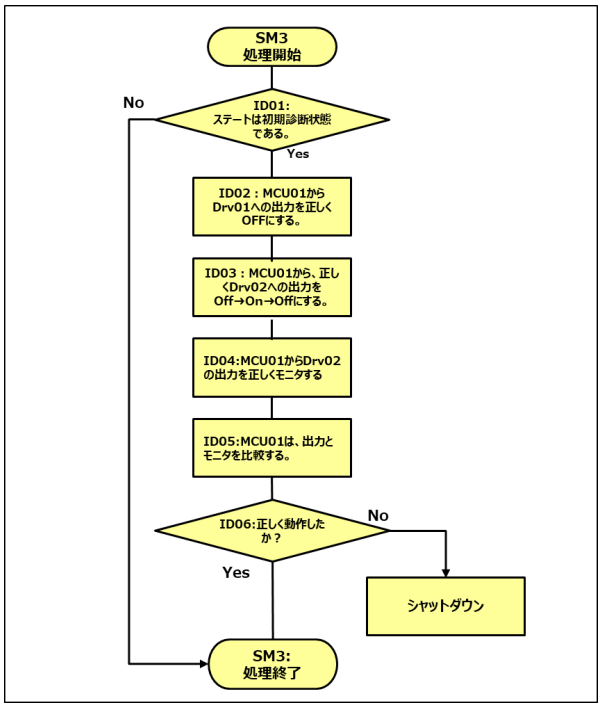


図 125　初期診断：フローチャート

これらの情報をもとに、要求を図 126 に示すように SCDL で記述する。このようにして、SCDL の分岐を用いて、要求仕様を表現することもできる。また、分岐を用いない場合には、図 127、図 128 のような表記となる。

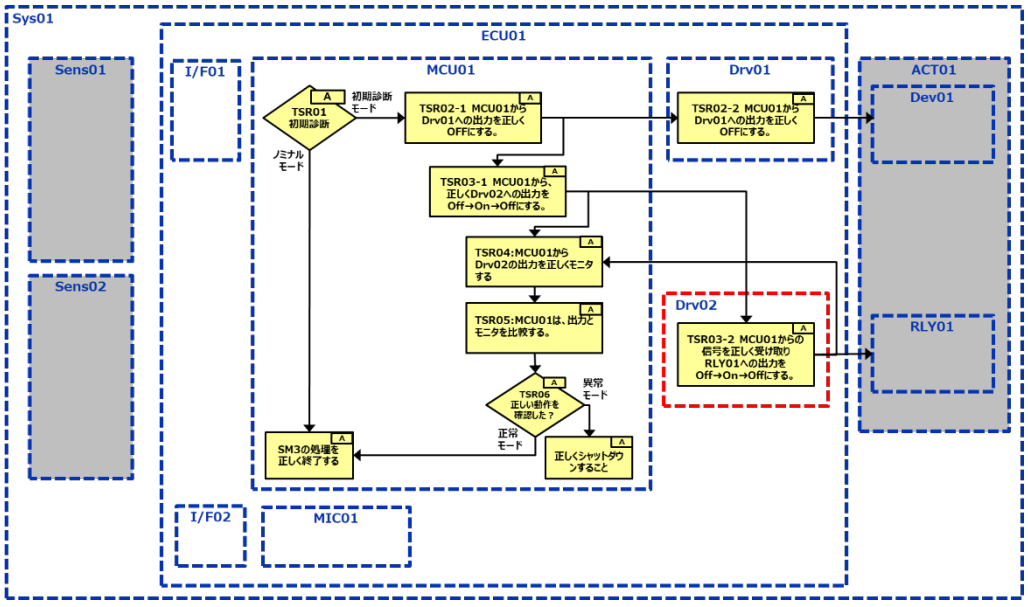


図 126　フローチャートの内容を表現した SCDL 例

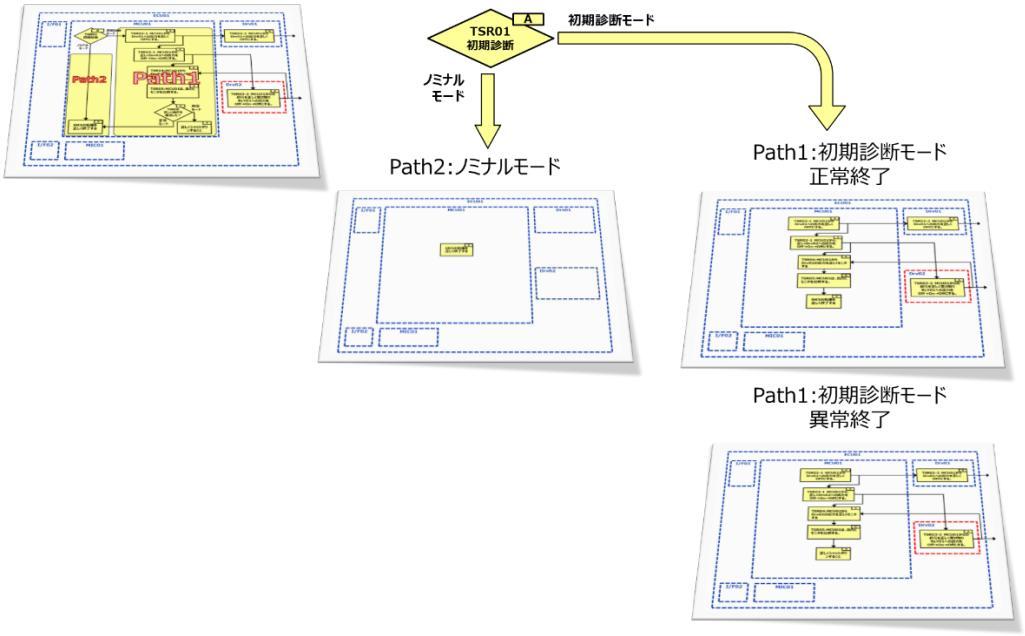


図 127　分岐を用いない場合のビュアイメージ

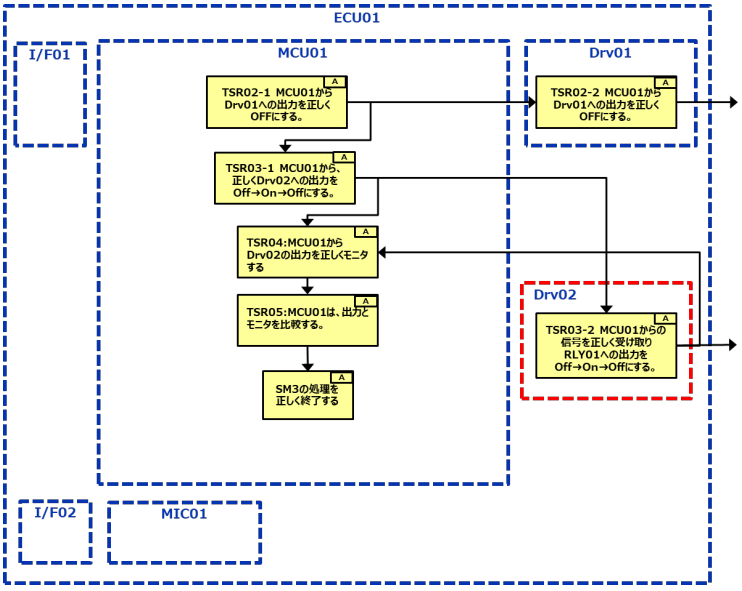


図 128　Path1 正常終了の Case の SCDL 表記イメージ

B.4.5 ハードウェア安全要求

上位の技術安全コンセプトに基づき、ハードウェア安全要求を導出する。ハードウェアは、システム、ソフトウェアとは違って実体を持ち、その特性から故障率の目標を定め、全体として規格要件の目標値の達成が求められ、その検討が必要である。

この際に用いられるのが、前項で紹介した

B.3.1 環境制約 B.3.1 環境制約

B.3.2 搭載スペース

B.3.3 調達

B.3.5 安全設計の定量評価

B.3.7 従属故障

B.3.8 ハードウェア部品認定などの項目になる。

今回のユースケースでは、技術安全コンセプトで目標値から各エレメントに分配された値があるので、その達成のため、あらかじめ顧客や規格などに指定されている要件にしたがって、対象のシステムで使用される各部品のカバレッジを含んだ故障率（FSC や TSC の段階で検討してきた安全機構のカバレッジを反映した故障率）を見積る。これには部品サプライヤからの情報や、すでに量産したハードウェアのエレメントごとのデータを基に算出を行い、図 129 に示すように、導出された結果（安全、非安全を含んだ）をエレメントごとに整理する。

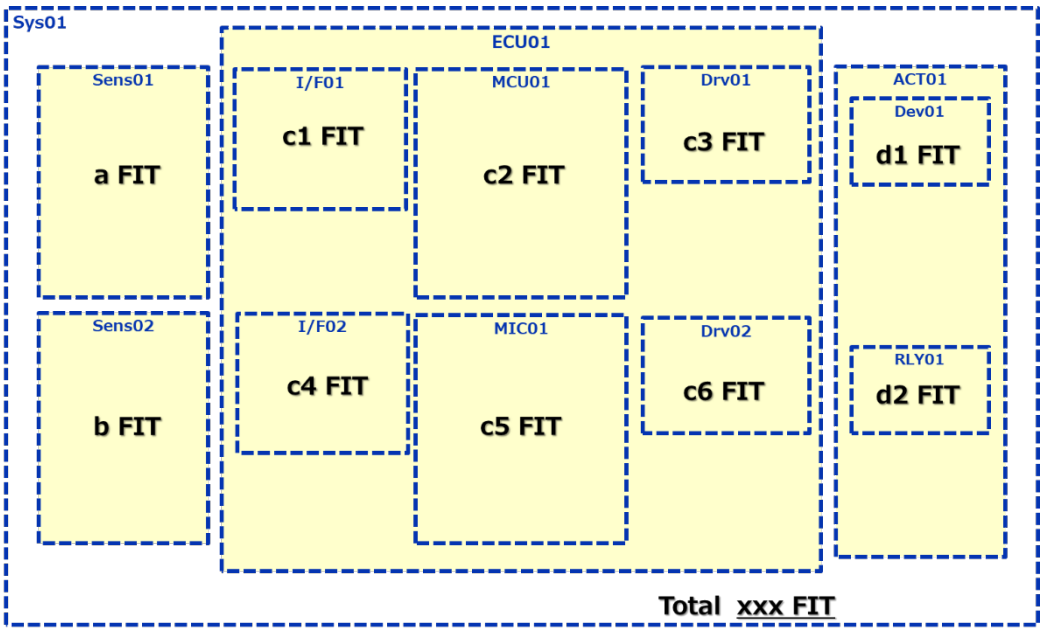


図 129　SCDL のエレメントを使った（安全機構のカバレッジを含んだ）故障率の状態

現状のハードウェア設計の段階では、(安全機構のカバレッジを含んだ) 故障率をエレメントに分配する際に、図 122 の各エレメントに分配された目標値に整合するように、個社で管理されているハードウェア部品、またはハードウェア回路ブロックを適用する。

図 130 に示すように、個社で管理しているハードウェア部品、またはハードウェア回路ブロックの故障率が分配されたエレメントの目標値 C5 を満たさない場合 (C5 < c5) は適用する部品を再考したり，安全機構を追加したりする工夫が必要となる。

図 130　目標値 C5 を満たさない場合の対応

これらの情報および、非機能要求 (環境) 、ASIL によるレイテント要求 (マルチプル検出間隔を含む) 、ハードウェア各部の故障影響を考慮した安全機構の設置 (安全分析) に基づいて、技術安全コンセプトから、表 32 に示すようにハードウェア安全要求を作成し、図 132 と図 133 に示すようにエレメントに展開する。

表 32　ハードウェア安全要求＜全体：一部抜粋＞

ID	名称	内容
HsR101-1-1-1	センシング 1	HW は車両挙動を静電容量に正しく変換し、HsR101-1-2-1 へ正しく受け渡すこと
HsR101-1-2-1	変換 1	HW は静電容量を電圧に正しく変換し、HsR101-1-3-1 へ正しく受け渡すこと
HsR101-1-3-1	デジタイズ 1	HW は A/D 変換を正しく行って正しくデジタル量にし、HsR101-1-4-1 へ正しく受け渡すこと
HsR101-1-4-1	コミュニケーション 1	HW はデジタル量をシリアル通信 (通信 Format ISO xxx) にて、HsR101-1-5-1 へ正しく送信する

		こと
ﾍ	ﾍ	ﾍ
HSR111-1-2-1	変換 2	HW は静電容量を電圧に正しく変換し、HSR111-1-3-1 へ正しく受け渡すこと
HSR111-1-3-1	デジタイズ 2	HW は A/D 変換を正しく行って正しくデジタル量にし、HSR111-1-4-1 へ正しく送信すること
HSR111-1-4-1	コミュニケーション 2	HW はデジタル量をシリアル通信（通信 Format ISO xxx）にて、HSR111-1-5-1 へ正しく送信すること。
HSR111-1-5-1	センシングダイアグ/センシング部診断 2	HW はエレメントの異常を検出するため、振動が規定値内であることを正しく確認し、HSR111-1-5-2 へ正しく受け渡すこと
HSR111-1-5-2	センシングダイアグ/センシング部異常処理 2	HW は、エレメント異常時には、直ちに通信を正しく停止し、HSR111-1-5-3 へ正しく受け渡すこと
HSR111-1-5-3	センシングダイアグ/オフセット異常検出 2	HW は、Sens02 のオフセットを正しく監視して、x 秒間で xxbit の振れを検出したら、正しくエレメントの情報送信を停止し、リセットまでゼロ値を正しく送信し、HSR112-1-1-1 へ正しく受け渡すこと
HSR112-1-1-1	センサデータ受信 FORMAT 検証 2	HW は通信 Format ISO xxx に従ったインタフェース（I/F）を正しく有し、HSR112-1-1-2 へ正しく送信すること
HSR112-1-1-2	センサデータ受信 FORMAT 異常処理 2	HW はエレメント、デジタル I/F の異常を正しく検出し、HSR112-1-2-1 へ正しく送信すること
HSR112-1-2-1	外乱保護-静電気 2	HW は ISO xxx に従った ESD 保護

		(xxkV)を正しく有し、HSR112-1-2-2 へ正しく受け渡すこと
HSR112-1-2-2	外乱保護-電界強度 2	HW は ISO xxx にしたがって xxxV/m の電界でデータを正しく入力し、HSR112-1-2-3 へ正しく受け渡すこと
HSR112-1-2-3	外乱保護-ノイズ保護 2	HW は ECU01 と Sens02 間のノイズ保護機能を正しく有し、（社内規格 xxx に従うこと）HSR112-1-2-4 へ正しく受け渡すこと
HSR112-1-2-4	外乱保護-短絡保護 2	HW は、外部ハーネスが GND、+B ショートしても I/F の破壊がなく、HSR112-1-3-1 へ正しく送信すること
HSR112-1-3-1	データ渡し-FORMAT2	HW は、シリアル・ペリフェラル・インタフェース（SPI）を正しく有し、HSR112-1-3-2 へ正しく受け渡すこと
HSR112-1-3-2	データ渡し-内容 2	HW は、SPI 通信を通して、State とデータを正しく送信し、HSR112-1-3-3 へ正しく受け渡すこと
HSR112-1-3-3	データ渡し-タイミング 2	HW は SPI 通信を通して、要求されれば直ちにデータを正しく送信し HSR112-2-1-1-へ正しく送信すること
HSR112-2-1-1	データ確認-手段	HW は、シリアル・ペリフェラル・インタフェース（SPI）を正しく有し、HSR112-2-1-2 へ正しく受け渡すこと
HSR112-2-1-2	データ確認-FORMAT	HW は SPI で、送信されたデータのフォーマットを正しく確認し、HSR112-2-2-1 へ正しく送信すること
HSR112-2-2-1	制御量決定-データ比	HW は、正しく受信したデータか

	較	ら、設定値テーブルと正しく比較して、制御量を正しく取得し、HSR112-3-1-1 へ正しく送信すること
HSR112-3-1-1	制御量検証-ルート	HW は、SPI 通信を通して MCU01 からの制御量を正しく取得し、HSR112-3-1-2 へ正しく受け渡すこと
HSR112-3-1-2	制御量検証-比較	HW は、MCU01 のデータと HSR112-2-2-1 のデータを正しく比較し、HSR112-3-2-1 へ正しく送信すること
HSR112-3-2-1	フェイル時操作 故障確定診断	HW は、比較した結果、許容値 zz を 3 秒超えたら、シャットダウンを正しく確定し、HSR113-1-1-1 へ正しく受け渡すこと
HSR113-1-1-1	MIC01 データ受信　ノイズ除去	HW は MIC01-03 と Drv02 間のノイズ保護機能を正しく有し、（社内規格 xxx に従うこと）HSR113-1-1-2 へ正しく受け渡すこと
HSR113-1-1-2	MIC01 データ受信確定	HW は、ハードワイヤを通して MIC01 からの制御量を正しく取得し、HSR113-1-2-1 へ正しく受け渡すこと
HSR113-1-2-1	受信結果出力	HW は HSR113-1-1-2 からの制御量を正しく取得し、RLY01 を正しく駆動すること
HSR114-1-1-1	制御量結果受信　ノイズ除去	HW は ECU01 と RLY01 間のノイズ保護機能を正しく有し、（社内規格 xxx に従うこと）HSR114-1-1-2 へ正しく受け渡すこと
HSR114-1-1-2	制御量受信	HW は HSR114-1-1-1 からの制御量を正しく取得し、ACT01 への供給電電を正しく駆動すること
HSR114-1-1-3	Drv01 電源供給/停止	HW は RLY01 と ACT01 間を正しく接

		続し、HsR104-1-1-2 へ正しく電源供給をおこなうこと
NHSR101-1	独立要求	HW は、「Sens01、I/F01、MCU01」と「Sens02、I/F02、MIC01、Drv02、RLY01」は、それぞれ独立した別部品で構成すること

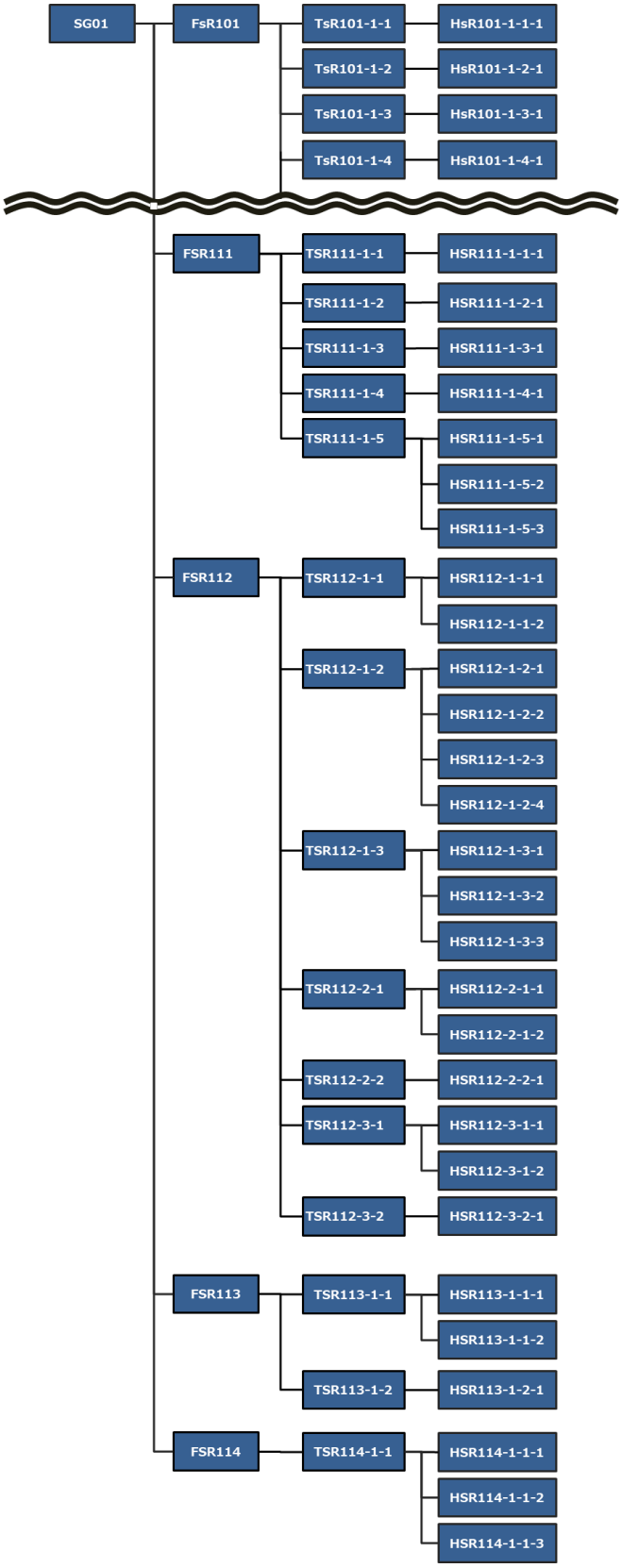


図 131　要求構造（全体：一部抜粋）

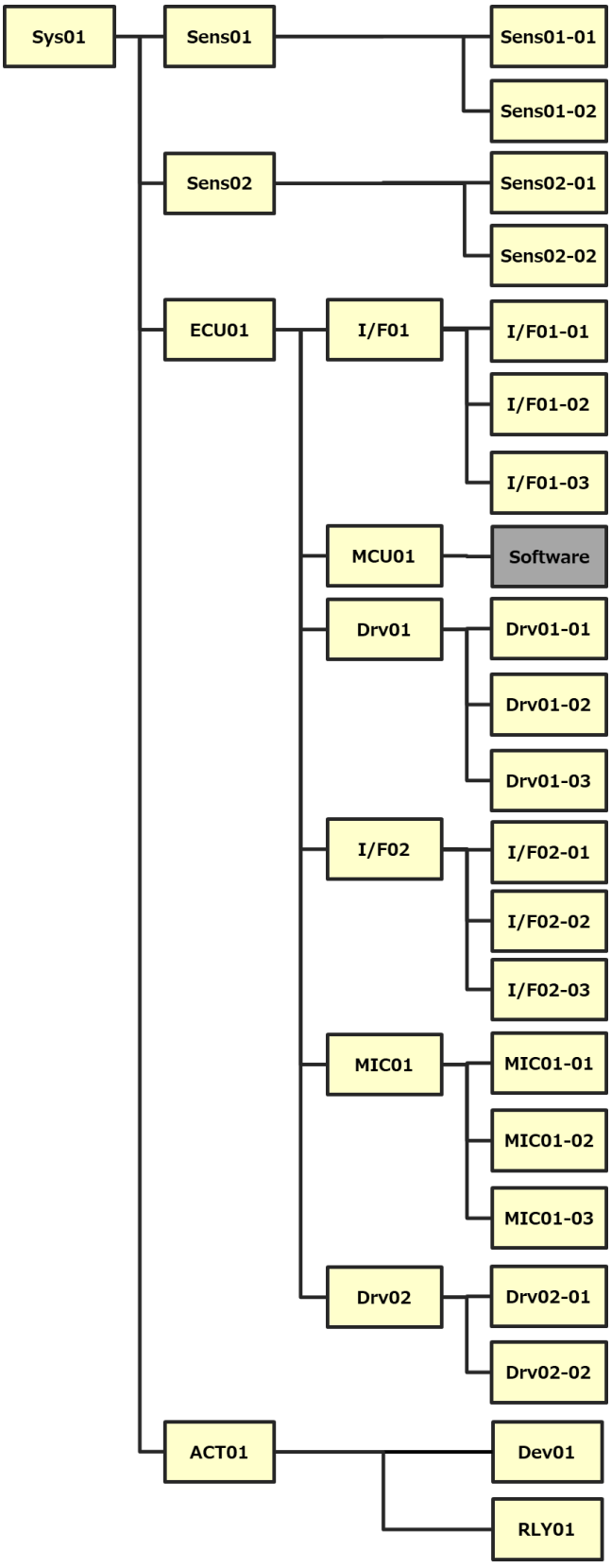


図 132　ハードウェア安全エレメント構成

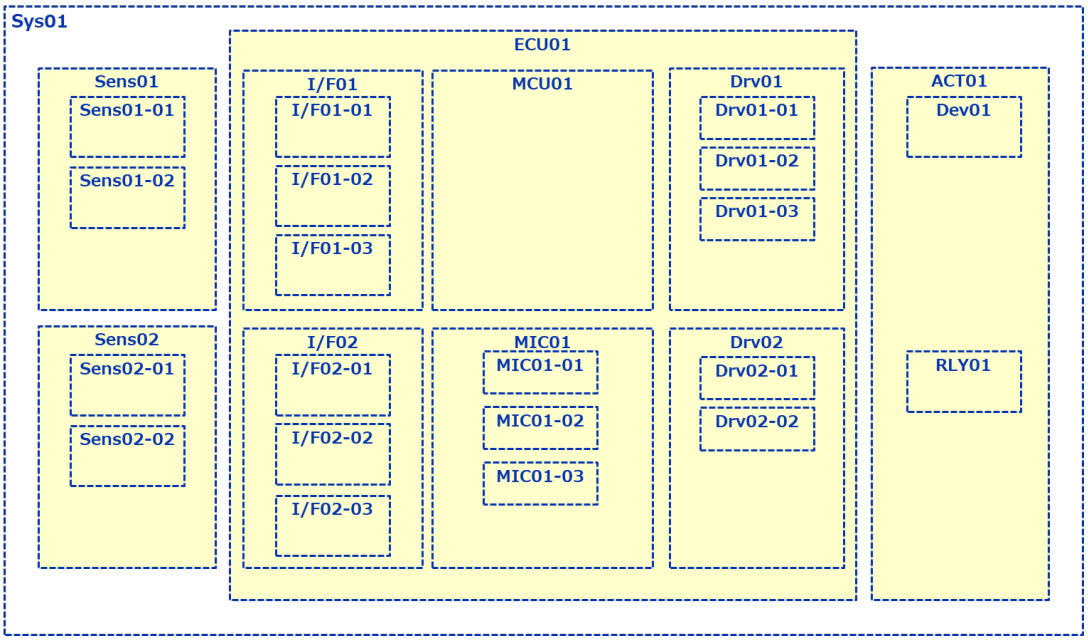


図 133　ハードウェア安全エレメント構造図

B.4.6 ハードウェア安全要求の配置およびハードウェア安全コンセプトの検証

B. 4. 4 までの活動によってハードウェア安全要求は導出され、構造化され、エレメントに配置され、図 134 に示すようにハードウェア安全コンセプト図となる。

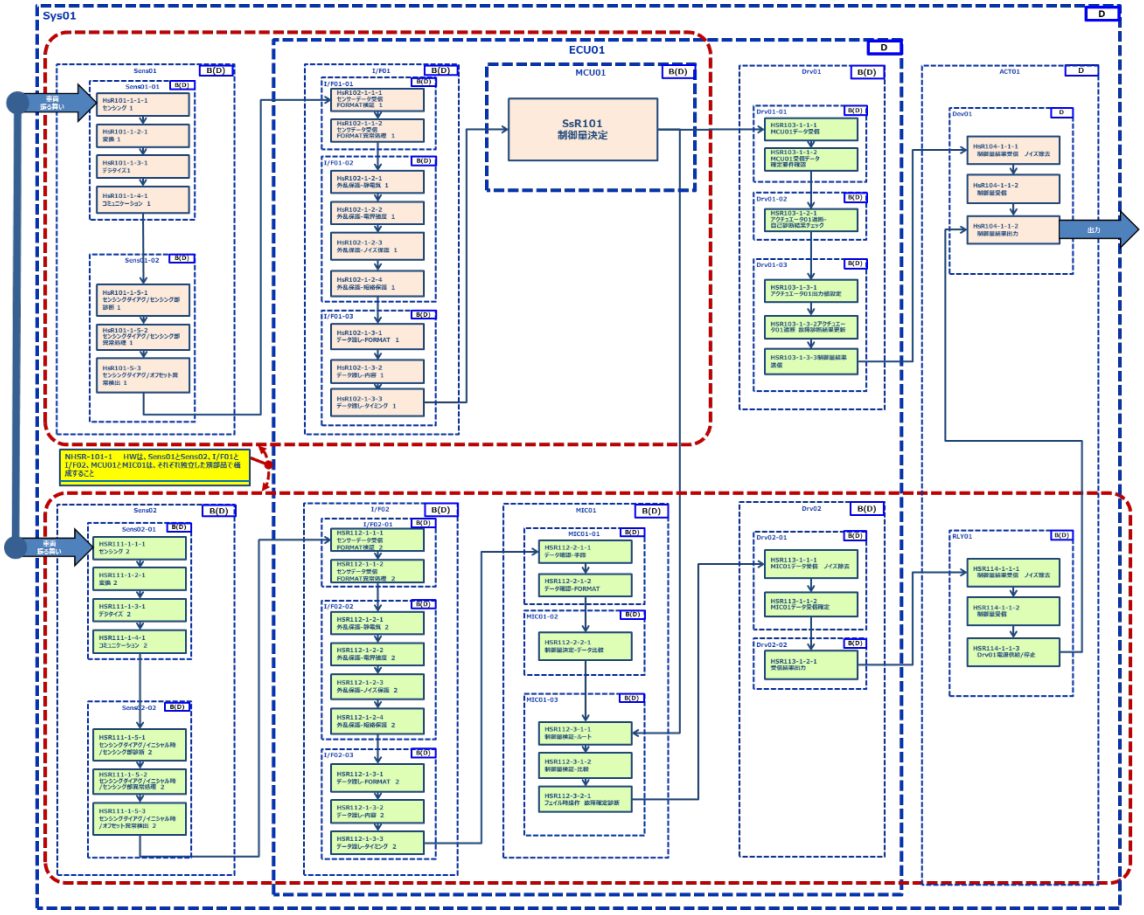


図 134　ハードウェア安全コンセプト図（全体）
（ペールオレンジ：意図機能由来の技術安全要求、
グリーン：安全機構、イエロー：独立要求）

アーキテクチャを細分化したので、アーキテクチャに基づいた安全分析を実施し、技術安全コンセプトの段階で行った安全分析（ボトムアップ/トップダウン）の結果を更新する。さらにハードウェアレベルの従属故障分析を実施し、技術安全コンセプトと同ように考慮点として抽出された内容を更新する。SCDL は要求間のインタラクションが明確になっているので、その情報を基にハードウェア安全要求レベルについても同ように分析が行える。その際に、基本的な構成は、多くの場合、図 135 の例に示すように個社ですでに管理されている HW ライブラリ（HW コンポーネント認定）をベースにすることもできる。

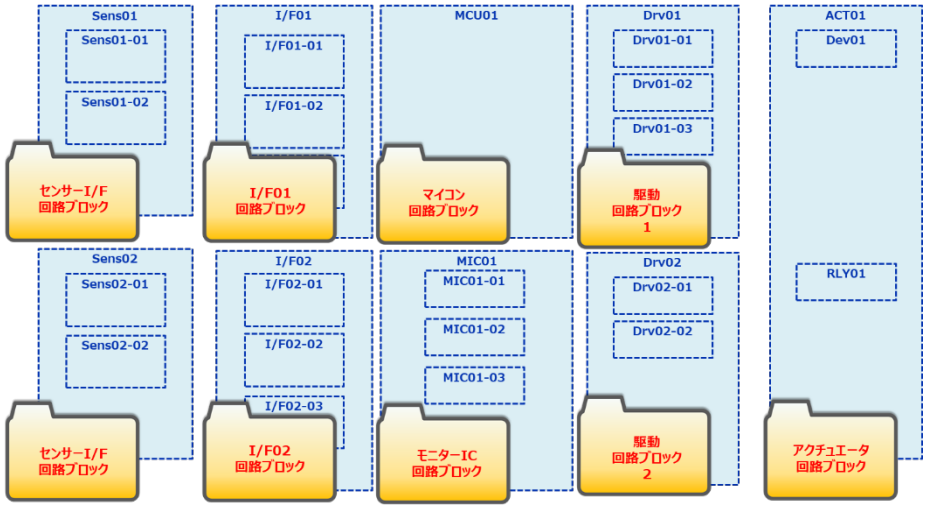


図 135　ハードウェアライブラリの例

図 135 のように、個社ではそれぞれの回路ブロックで分析結果も管理されており、それを基に回路ブロックごとに図 136 の例に示すように部品レベルの FMEA のブロックを管理することもできる。

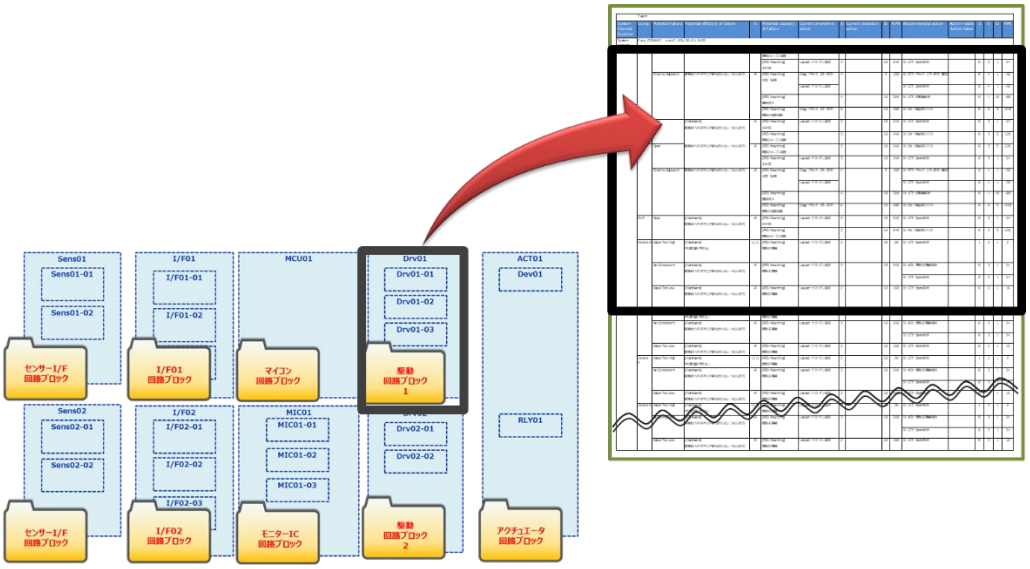


図 136　ハードウェア回路ブロックごとに FMEA ブロックを管理した例

同ように、回路ブロック単位で、信頼性データ、故障カバレッジ、故障率目標値、そして部品 FMEA の結果を踏まえて図 137 の例に示すようにハードウェアアーキテクチャメトリックの評価を実施することもできる。SCDL は、エレメントと要求の配置関係が明確であり、それぞれのインタラクションが明示されているため、定量的評価の分析を容易に実施することができる。

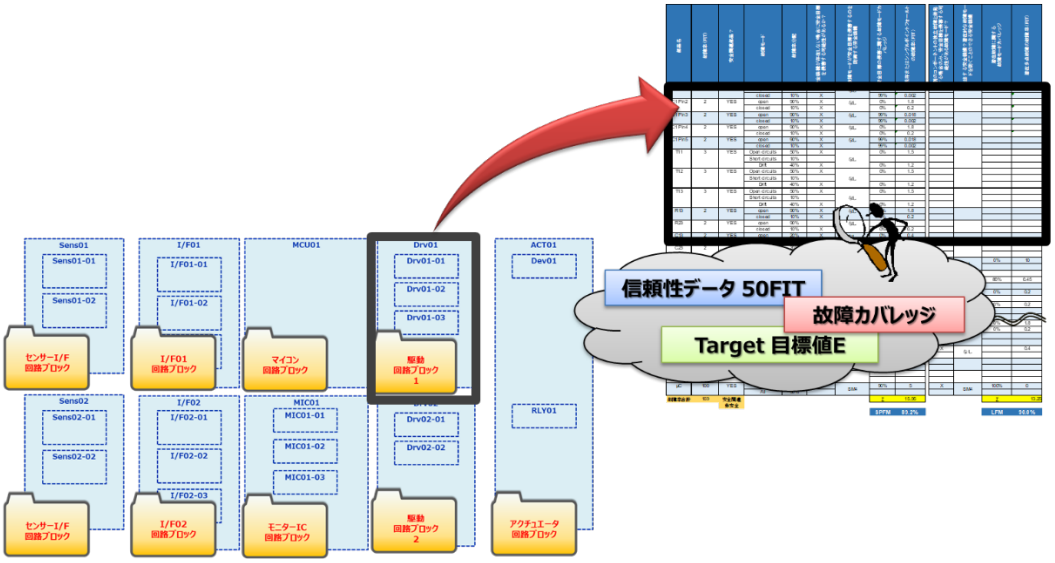


図 137　ハードウェア回路ブロックごとに
ハードウェアアーキテクチャメトリックを管理した例

次にランダムハードウェア故障による安全目標侵害の評価（PMHF）について、例えば下記の図 138 に示すように守られる機能（意図機能由来の機能安全要求）と、守る機能（安全機構）が配置されている場合について、分かりやすいよう FSR ベースで取り上げてみる（この例では SM1 のみ）。

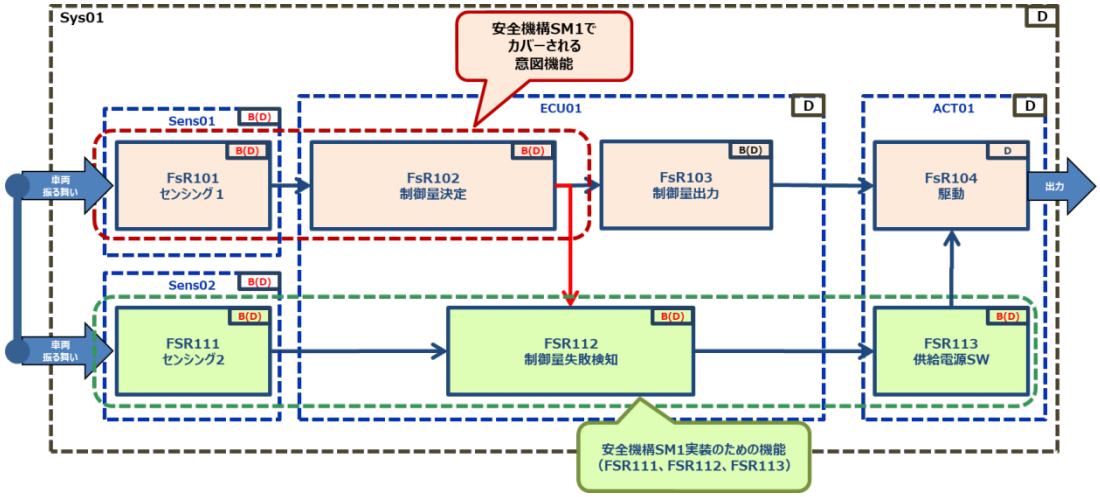


図 138　意図機能と安全機構の関係（例）

まず、ボトムアップアプローチの帰納的分析手法の代表例として、ISO 26262-5:2011 Annex E に要求されている FMEA 事例を示す（FMEA を実施すればよいという意図で例示しているわけではない）。次の表 33 の事例では、図 138 の意図した機能と安全機構の関係に基づいて、エレメントを基軸に安全分析を行った。

表 33　帰納的分析（例）

				SMがない場合にSGに違反する可能性があるか？			別の独立故障と合わせてSGに違反する可能性があるか？		
エレメント ID	役割 (安全要求等)	故障モード	故障率 λ	SG侵害 有/無	安全機構 (SM) 有/無	安全機構 ID	SG侵害 有/無	安全機構 (SM) 有/無	安全機構 ID
Sens01	FsR-101	A	λ_A	無	–	–	無	–	–
		B	λ_B	有	有	SM1	無	–	–
Sens02	FSR111 (SM1)	C	λ_C	無	–	–	有	無	–
		D	λ_D	無	–	–	無	–	–
ECU01	FsR-102	E	λ_E	無	–	–	無	–	–
		F	λ_F	有	有	SM1	無	–	
	FsR-103	G	λ_G	有	無	–	無	–	–
		H	λ_H	無	–	–	無	–	–
	FSR112 (SM1)	I	λ_I	無	–	–	有	無	–
		J	λ_J	無	–	–	無	–	–
ACT01	FsR104	K	λ_K	有	無	–	無	–	–
		L	λ_L	無	–	–	無	–	–
	FSR113 (SM1)	M	λ_M	無	–	–	有	無	–
		N	λ_N	無	–	–	無	–	–

次にトップダウンアプローチの演繹的分析手法の代表例として、FTA 事例を示す（FTA を実施すればよいという意図で例示しているわけではない）。演繹的分析により、図 139 に示すように定量的 FTA が作成でき、ISO 26262 が要求する「ランダムハードウェア故障の確率的メトリック(PMHF)の評価」を容易におこなうことができる。

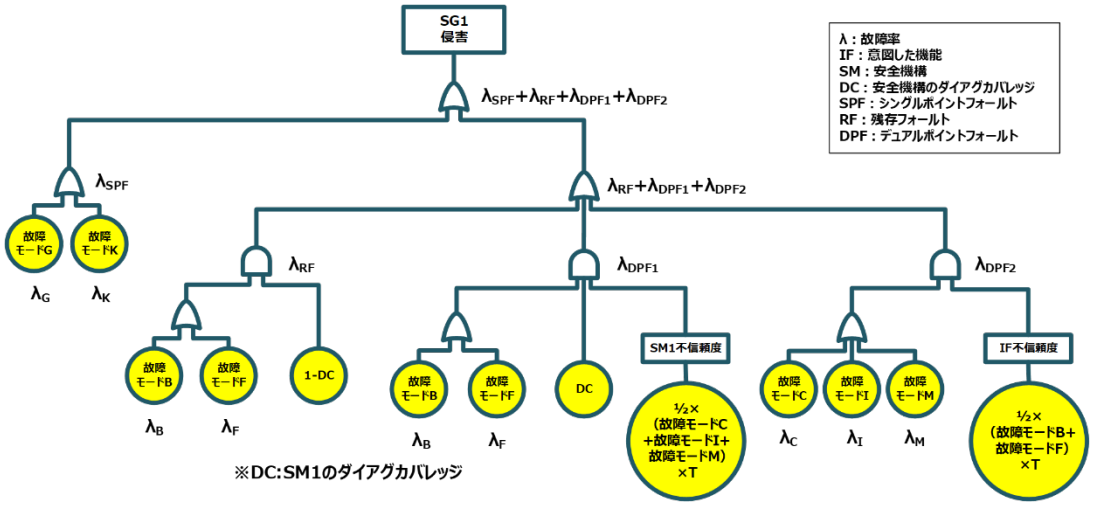


図 139　演繹的分析の定量的 FTA（例）

以上のように、SCDL を用いて、ハードウェア安全要求の詳細化、分配化、そしてそのハードウェアの安全性定量評価を合理的に実施することができ、その際のトレーサビリティの結果から完全性を満たすことが期待できる。

B.4.7 アイテムのエレメント間に配置された非機能要求の検証

ここまでは、各エレメントに配置される要求について検討してきた。一方で、個別のエレメントではなく、アイテムに配置されたエレメント間の非機能要求（独立要求）については、例えば前述の図 112 の例では、図 140 で示すように 6 つの独立要求がアイテムに配置されている。これらの非機能要求についてもこのレベルの従属故障分析で確認するとよい。

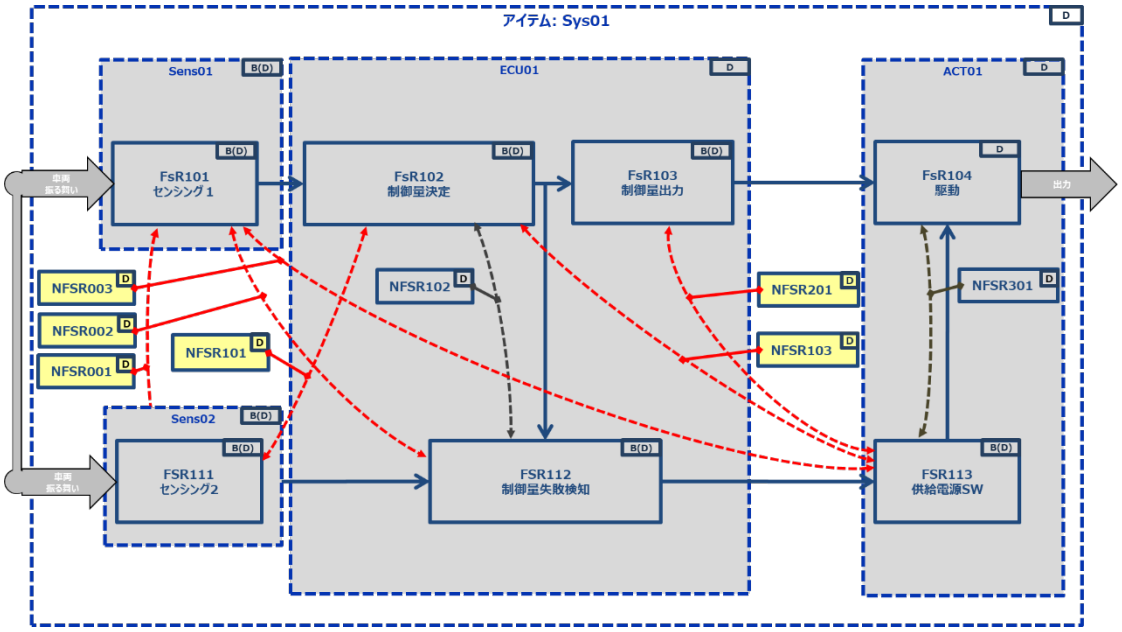


図 140　アイテムに配置された非機能要求

表 34　SG01:アイテムに配置された非機能要求の例

ID	名称	内容
NFSR001	独立要求	意図機能由来の機能安全要求(FsR101)と安全機構 SM1(FSR111)の共通原因故障およびカスケード故障がなきこと
NFSR002	独立要求	意図機能由来の機能安全要求(FsR101)と安全機構 SM1(FSR112)の共通原因故障およびカスケード故障がなきこと
NFSR003	独立要求	意図機能由来の機能安全要求(FsR101)と安全機構 SM1(FSR113)の共通原因故障およびカスケード故障がなきこと
NFSR101	独立要求	意図機能由来の機能安全要求(FsR102)と安全機構 SM1(FSR111)の共通原因故障およびカスケード故障がなきこと
NFSR103	独立要求	意図機能由来の機能安全要求(FsR102)と安全機構 SM1(FSR113)の共通原因故障およびカスケード故障がなきこと
NFSR201	独立要求	意図機能由来の機能安全要求(FsR103)と安全機構 SM2(FSR113)の共通原因故障およびカスケード故障がなきこと

B.4.8 TSR のリファインおよび HSI、HSR、SSR の導出例

B.4.1 ～ B.4.7 では、技術安全コンセプトからハードウェア安全コンセプトに至る安全アーキテクチャ設計を構築する場合のユースケースを紹介してきた。本来であれば、技術安全要求（TSR）は直接、あるいはさらにリファインされて、ハードウェアおよび／またはソフトウェアに配置され、ハードウェア-ソフトウェアインタフェース(HSI)の仕様化をおこなった後、ハードウェア安全要求（HSR）および／またはソフトウェア安全要求（SSR）に展開され実装されていく。

ここまでハードウェアに焦点を絞って説明してきたが、本項では、仮想の A/D 変換モジュールに関連する安全要求を採り上げ、技術安全要求から HSR、SSR、HSI への導出例を補足として紹介する。今回の事例における初期の技術安全要求を表 35 にまとめた。また、技術安全コンセプト図を図 141 に示す。

表 35 技術安全要求＜一部抜粋＞

ID	名称	内容
TSR-SN1	センシング	SEN1 は温度を電圧に正しく変換すること。
TSR-DIG1	デジタイズ	DIG1 は電圧を温度として、正しくデジタル値に変換すること。
TSR-CM1	通信	CM1 は温度値を正しくし、TSR-*へ送信すること。

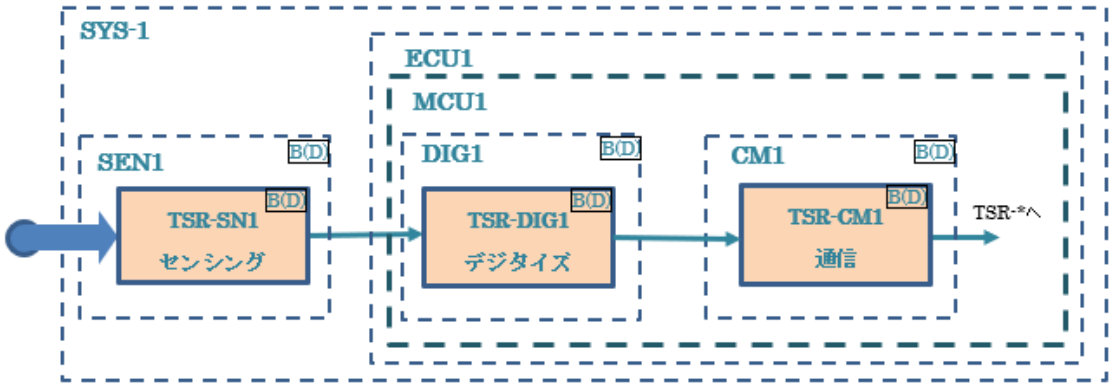


図 141 技術安全コンセプト図<一部抜粋>

以降はマイコン（MCU1）内の DIG1 のみに着目して詳細化していく。技術安全要求 TSR-DIG1 に対し、性能要求の達成見込み、過去の採用実績や種々の情報などから実現可能な A/D 変換モジュールを想定する。ハードウェアとソフトウェアの役割分担を考慮しながら技術安全要求をどちらに割り付けるかを検討し、TSR をリファインする（表 36）。

デジタイズ機能に対する性能要求として以下を想定している。

- 温度のダイナミックレンジ（分解能）を考慮し 10bit で A/D 変換する。
- 温度は-100～200℃の範囲をデジタイズする。
- 1msec サンプリング間隔でデジタイズする。
- A/D 変換結果は所定の式を用い温度値に変換する。

今回の要求が実現可能な A/D 変換モジュールとして以下を想定している。

- アナログ入力は 4 チャネルある。
- マルチプレクサ後に一つの A/D 変換器により変換する。
- 基準電圧は外部／内部の切り替えが可能である。
- 分解能は 10bit／12bit の切り替えが可能である。
- 変換結果はレジスタに格納される。

表 36 技術安全要求のリファイン

ID	名称	ﾊｰﾄﾞｳｪｱ	ｿﾌﾄｳｪｱ	内容
TSR-DIG1-HW1	A/D 変換	○	-	入力電圧を正しく A/D 変換する。
TSR-DIG1-SW1	温度変換	-	○	入力電圧を正しく温度値に変換する。

次に、それらに対する HSI を書き出す（表 37）。

表 37 ハードウェア-ソフトウェア インタフェース

ID	ｺﾝﾌｨｸﾞﾚｰｼｮﾝ	ﾀｲﾐﾝｸﾞ	ﾓｰﾄﾞ	分解能	ﾒﾓﾘ	配置先
DIG1-HSI -1	変換分解能を設定	初回	初期	10bit	—	I/F-1
DIG1-HSI -2	—	変換周期 1msec	変換	—	—	I/F-1

DIG1 に対するハードウェア、ソフトウェアそれぞれの技術安全要求を安全要求配置図（図 142）に示す。

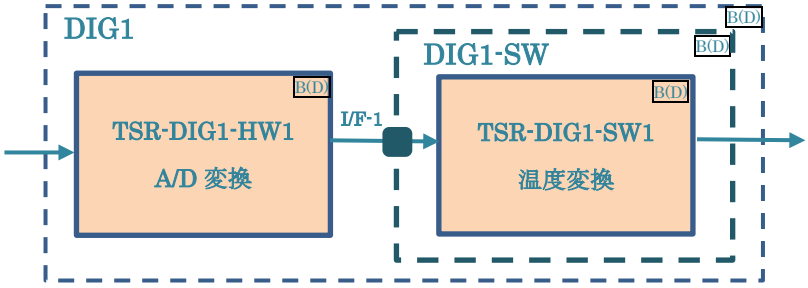


図 142 TSR-HW、TSR-SW

他の技術安全要求も含めて、安全分析および従属故障分析の結果、ハードウェアアーキテクチャメトリックの達成目処などを考慮し、必要に応じて TSR や HSI のリファインおよび分析を繰り返す。今回はリファインの必要がなかったため、表 36 のハードウェアに割り付けられた TSR に基づいて HSR に分解し（表 38）、さらにソフトウェアに割り付けられた TSR に基づいて SSR に分解した（表 39）。

表 38 ハードウェア安全要求

ID	名称	内容
HSR-DIG1-HW1-1	A/D 変換 1	1msec 周期のサンプリング間隔にて、A/D 変換時間 100 μsec 以内で正しく A/D 変換する。

表 39 ソフトウェア安全要求

ID	名称	内容
SSR-DIG1-SW1-1	初期設定	起動初回時に正しく A/D 変換モジュールの初期設定をする。 ・CH1 を使用 ・内部基準電圧を使用 ・分解能を 10bit に設定
SSR-DIG1-SW1-2	A/D 値処理	A/D 変換結果を正しく受け渡す。
SSR-DIG1-SW1-3	温度変換 1	1msec 周期のサンプリング間隔にて、900 μsec 以内で正しく温度値に変換する。

表 37 の HSI を HSR、SSR の粒度に合わせて詳細化し、書き出す。（表 40）

表 40 ハードウェア-ソフトウェア インタフェースのリファイン

ID	ｺﾝﾌｨｸﾞﾚｰｼｮﾝ	ﾀｲﾐﾝｸﾞ	ﾓｰﾄﾞ	分解能	ﾒﾓﾘ
DIG1-HSI-1-1	ﾏﾙﾁﾌﾟﾚｸｻの設定：CH1	初回	初期	—	—
DIG1-HSI-1-2	基準電圧：内蔵基準電圧	初回	初期	—	—
DIG1-HSI-1-3	分解能の設定	初回	初期	10bit	—
DIG1-HSI-2-1	変換周期：1msec	A/D 変換：100μsec 温度変換：900μsec	変換	0.5℃/bit	符号付 2byte：RAM

DIG1 の HSR、SSR をエレメントに配置した安全要求配置図（一部抜粋）を図 143 に示す。ここでは、通常動作時の要求とは分けて、A/D 変換モジュールに対する初期設定として、起動初回時のみのソフトウェア技術安全要求として SSR-DIG1-SW1-1 を配置した。

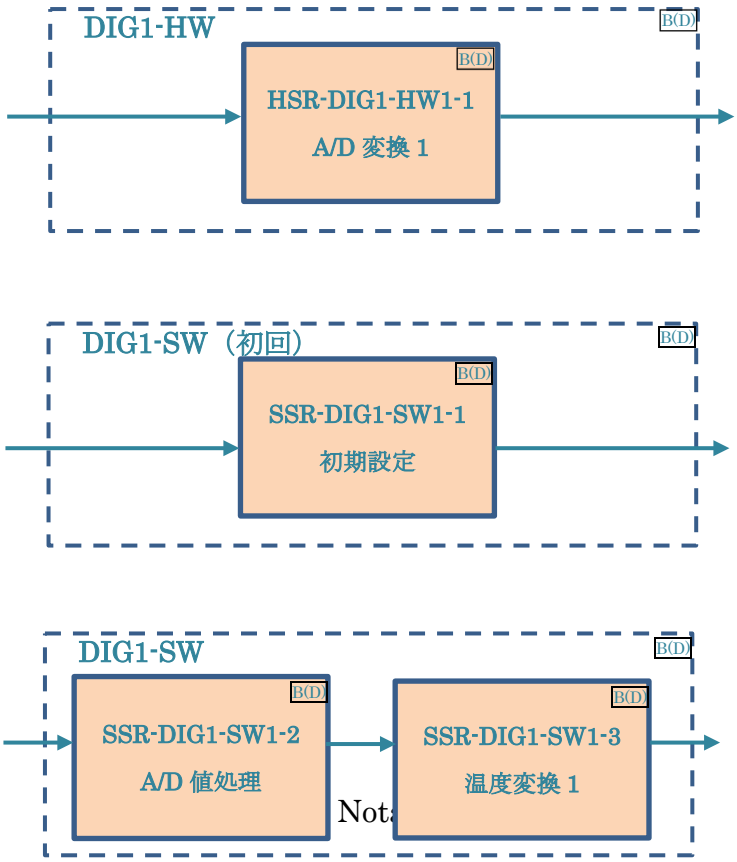


図 143 HSR,SSR

この一連の作業結果を要求ツリーとして図 144 に示す。

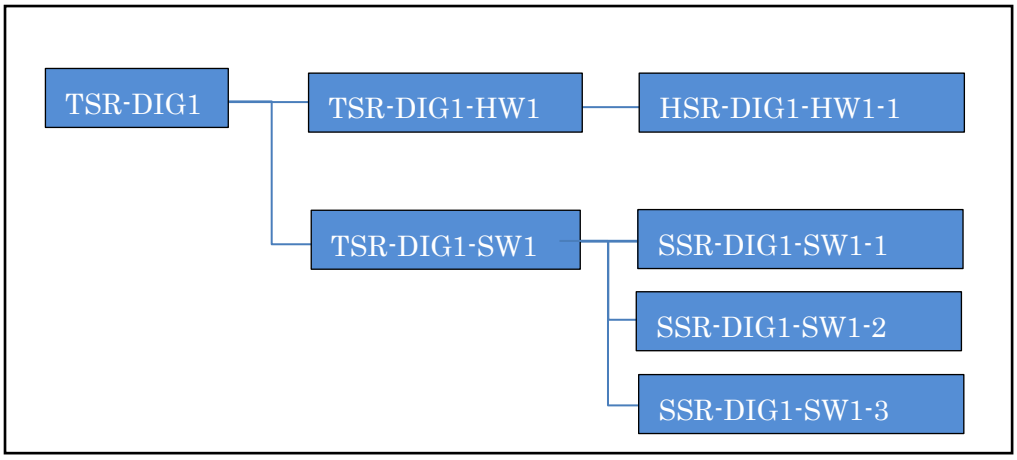


図 144 要求ツリー

以上が技術安全要求から HSR、SSR、HSI を導出する例である。実際にはハードウェアだけでなくソフトウェアにも技術安全コンセプトが割り付けられる場合があることを踏まえて本編を参照されたい。また、HSI を考慮しながら TSR をリファインし、ハードウェアおよびソフトウェアに要求を割り付けるには専門知識が必要となる場合もあることから、各分野の専門知識を持つメンバの協力によるリファイン作業をお勧めする。

B.5 まとめ

本編では、安全目標からのトップダウンアプローチと、実績のあるハードウェア安全コンセプト設計（ハードウェアコンポーネント）からのボトムアップアプローチを総合（Synthesis）し、技術安全コンセプトからハードウェア安全コンセプトに至る安全アーキテクチャ設計を構築する場合のユースケースを紹介した。具体的には、1.4 節 ユースケースにおいて、1.3 節の表 1 に示すようなハードウェア設計時の検討事項がどのようなシーンで必要になるのかを示し、これまで積み上げてきた種々のハードウェア設計時の資産を活用しながら、上位要求を戦略的にハードウェア安全コンセプトとして構築していくことが効果的かつ効率的な安全設計となるかを説明した。

図 145 に示すように、技術安全コンセプトからハードウェア安全コンセプト設計に至る設計工程に SCDL を用いることで、各段階の要求間のインタラクションおよび各エレメントに配置された要求間の関係を容易に把握することができるようになる。さらに、要求分析、ランダムハードウェア故障の目標値をエレメントへ分配、安全分析といった一連の安全活動を一貫しておこなうことが可能となる。

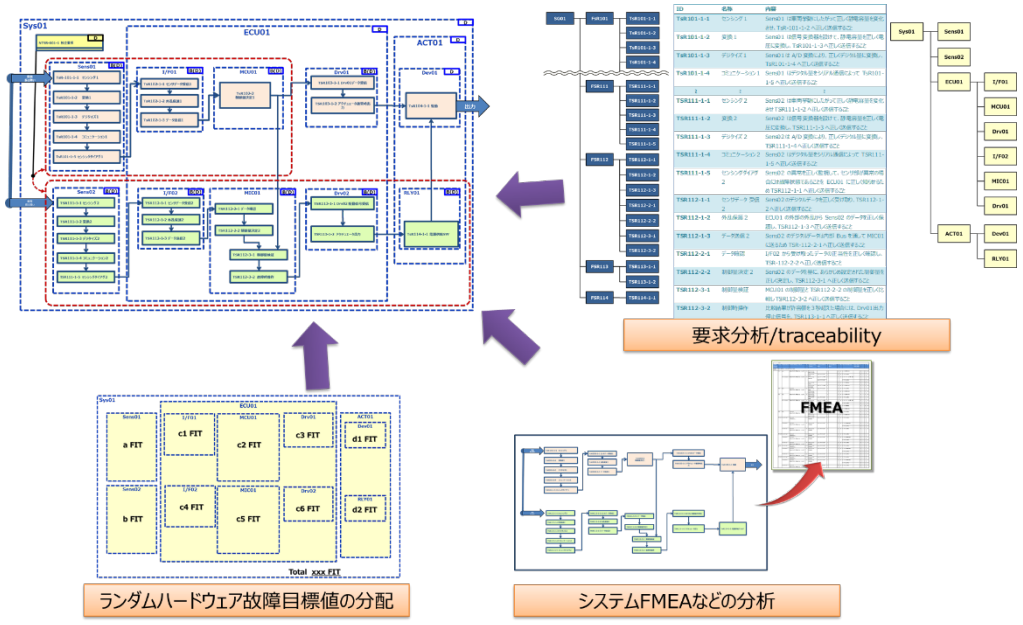


図 145 SCDL を基本とした様々な分析（例）

附属書C ユースケース（ソフトウェア編）

C.1 目的

本章では、SCDL のユースケースの中で、自動車用車載制御システムのソフトウェアに特有の記載例を紹介する。ソフトウェアの場合はハードウェアの場合と異なり、コンポーネント間のつながりが複雑であり、マイコンハードウェアを含めた共有リソースも多い。安全度水準(ASIL)の異なるソフトウェアが混在して、ソフトウェアパーティショニングを考慮せねばならない場合に、低 ASIL 部位から高 ASIL 部位に対するカスケード故障を防止するための安全要求の記述においては、その特殊性を考慮する必要がある。記載例は具体的に以下の２例である。

例１：ソフトウェアパーティショニングの SCDL 記述例

例２：共通ライブラリの SCDL 記述例

例１（C.2 節で詳述）では、自動車用車載制御システムにおける仮想的なコントローラにおいて、ASIL の異なるソフトウェアが混在して、ソフトウェアパーティショニングを考慮せねばならない事例について、実装を意識した技術安全コンセプト(TSC)に利用できる SCDL 記述例を紹介する。

例２（C.3 節で詳述）では、ASIL の異なるソフトウェア間で共通に利用される共通ライブラリの実装方式の検討と、その SCDL 記述例を紹介する。

【備考】

本書では、事例を、一旦、非形式記法による記述で示した。その後に、その内容を SCDL で表現可能かを検討し、SCDL で記述した事例(ユースケース)を示すものである。

C.2 ソフトウェアパーティショニングの SCDL 記述例

安全コンセプトの検討において、ソフトウェアに関連する課題として、ソフトウェアパーティショニングによる無干渉(FFI: Freedom From Interference)要求をいかに記述するかがある。

本節では、ソフトウェアパーティショニングを考慮したソフトウェア構成図の例題を仮定して、その記載事例を紹介する。

C.2.1 SCDL 記述例で取り上げるシステムの説明

低 ASIL が配されている機能（以降、低 ASIL 機能、例：QM 機能）と高 ASIL が配されている機能（以降、高 ASIL 機能、例；ASIL-X 機能）の２種の安全度水準のソフトウェアコンポーネントが混在し、低 ASIL 機能から高 ASIL 機能へのカスケード故障を防止するために、ソフトウェアパーティショニングによる FFI を要求するものと想定する。なお、ASIL-X は、ASIL-A～D のいずれかと想定する。

本節では、FFI に関する要求を、一旦、非形式記法による記述で示す。このとき、FFI に関する要求を、まず機能レベルの要求から検討開始し、最終的に実装レベルの検討結果を示した。次節では、その最終検討結果（実装レベルの検討結果）を SCDL で表現可能かを検討し、SCDL で記述した事例(ユースケース)を示す。

まず、機能レベルで簡易構成を描いたものが、図 C- 1 である。簡易構成とは、機能ブロックの配置のみを記述し、機能ブロック間のインタラクションは記述省略したものである。

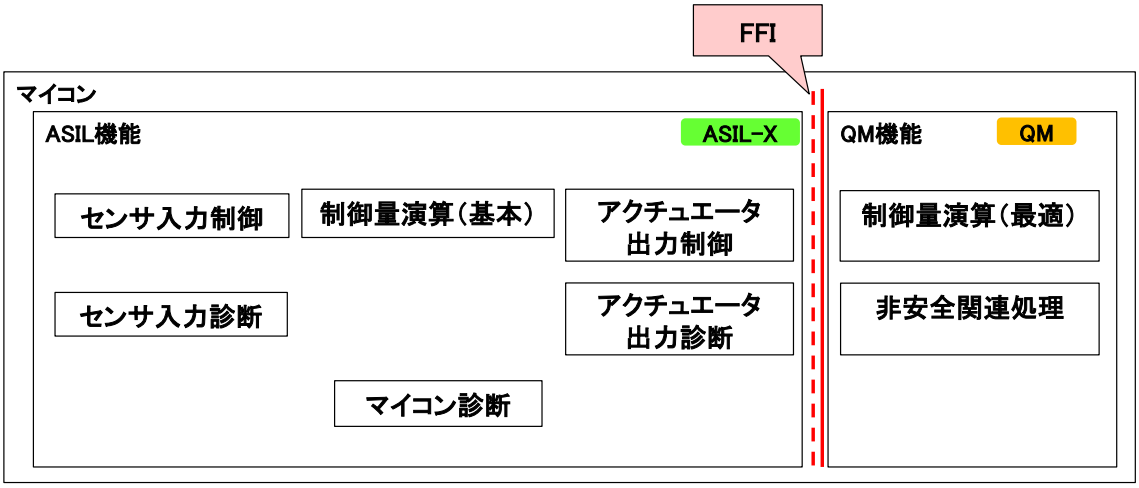


図 C- 1 仮想コントローラのマイコン内部の簡易構成 (機能レベル)

図中の FFI の仕切り線は、実線側の部位から破線側の部位に干渉しないことを意味する

次に、実装レベル（技術安全コンセプトのレベル）で簡易構成を描いたものが、図 C 2 である。本例では、ソフトウェアは、アプリケーションソフトウェア（アプリソフト）（ASW: Application Software)と基盤ソフトウェア（基盤ソフト）（BSW: Basic Software)に分離して実装する構成とした。また、本例では、ソフトウェアパーティショニングの実装構成例として、アプリソフトは ASIL アプリソフトと QM アプリソフトに分離し、基盤ソフトとハードウェアは ASIL 共通基盤として実装する構成とした。

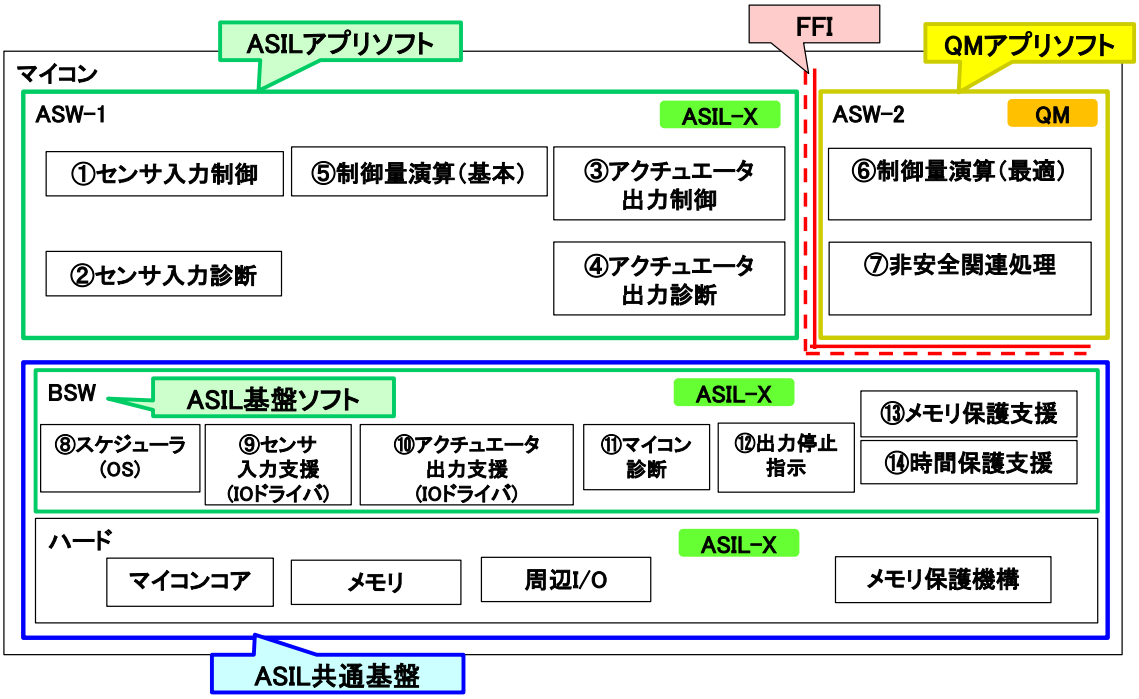


図 C 2　仮想コントローラのマイコン内部簡易構成　(実装レベル)

図 C 2 における各ブロックの機能を表 C- 1 に示す。

表 C- 1　仮想コントローラのマイコン内ブロックの機能

No.	ブロック名称	機能	ASIL
①	センサ入力制御 [ASW-1]	センサ i からの入力値を受信する。(i=1～N)	ASIL-X
②	センサ入力診断 [ASW-1]	センサ i からの入力値を診断する。(i=1～N)	ASIL-X
③	アクチュエータ出力制御 [ASW-1]	アクチュエータへの出力指示値を送信する。	ASIL-X
④	アクチュエータ	アクチュエータの動作を診断する。	ASIL-X

	出力診断 [ASW-1]		
⑤	制御量演算（基本） [ASW-1]	アクチュエータへの制御量を演算し指示値を出力する。通常は、制御量演算（最適）からの指示値（最適値）をそのまま出力する。しかし、制御量演算（最適）からの指示値が許容最大値以上の場合は、許容最大値に置き換えて出力する。	ASIL-X
⑥	制御量演算（最適） [ASW-2]	センサ入力情報や経緯情報を元に、アクチュエータへの最適な制御量を演算して指示値を求め、制御量演算（基本）に渡す。	QM
⑦	非安全関連処理 [ASW-2]	安全に関わる制御には直接関連しない処理を行う。（例：盗難防止機能）	QM
⑧	スケジューラ(OS) [BSW]	アプリタスクのスケジューラ（ディスパッチャー、割込処理、システムコール受付処理）	ASIL-X
⑨	センサ入力支援(IO ドライバ) [BSW]	センサｉからの入力信号を受けてアプリソフト(センサ入力制御）への入力値に変換する。(マイコン周辺 IO（ADC など）の入出力制御ソフト)	ASIL-X
⑩	アクチュエータ出力支援(IO ドライバ) [BSW]	アプリソフト(アクチュエータ出力制御）からの出力指示値をアクチュエータへの出力信号に変換する。(マイコン周辺 IO（PWM など）の入出力制御ソフト)	ASIL-X
⑪	マイコン診断 [BSW]	マイコンの診断を行い、診断結果や異常検出情報を収集する。正常時にはマイコン外部の監視装置に正常であることを定期的に伝える。(マイコン外部の監視装置は定期的な正常通知が途絶えた場合にはマイコンの出力指示をカットし安全状態に遷移する。)	ASIL-X
⑫	出力停止指示 [BSW]	センサ入力診断またはアクチュエータ出力診断からの異常通知を受けたら、アクチュエータへの制御指示をカットさせる信号を出力する	ASIL-X
⑬	メモリ保護支援 [BSW]	マイコンハードのメモリ保護機構を使用して、QM アプリから ASIL 部位に対するメモリ保護を支援する。	ASIL-X
⑭	時間保護支援 [BSW]	OS(スケジューラ）と連携して、QM アプリから ASIL 部位に対する時間保護を支援する。	ASIL-X

次に、安全分析に必要な機能ブロック間のインタラクションを検討する。本例では簡易化

のためにアプリソフトに関わるインタラクションに着目した。本例におけるインタラクションを、図 C- 3 に示す。

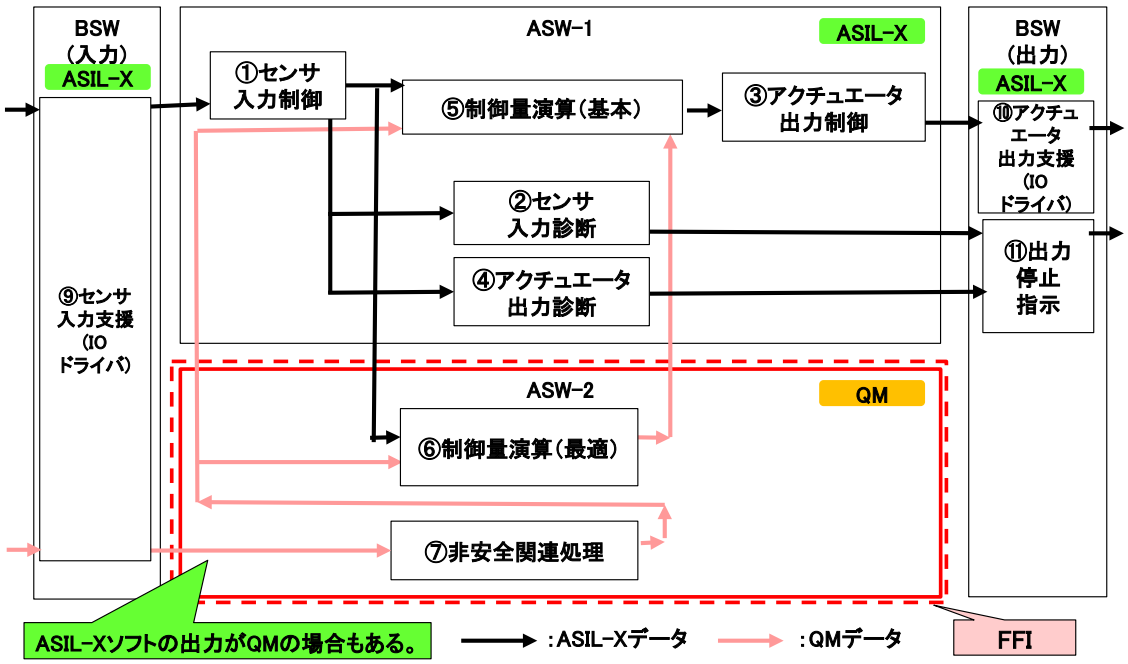


図 C- 3　仮想コントローラのマイコン内インタラクション　(アプリソフトに着目)

C.2.2　SCDL 記述例

仮想コントローラにおけるマイコン内ソフトウェアの SCDL 記述例を図 C- 4 に示す。本例では簡易化のために、アプリソフトに関わるインタラクションにのみ着目して記載した。ソフトウェアパーティショニングによる FFI 要求は、QM アプリソフトで構成されるエレメント「EL3」から、ASIL アプリソフトで構成されるエレメント「EL2」および ASIL 基盤ソフトで構成される「EL1」(および EL4) に対して、指定される。

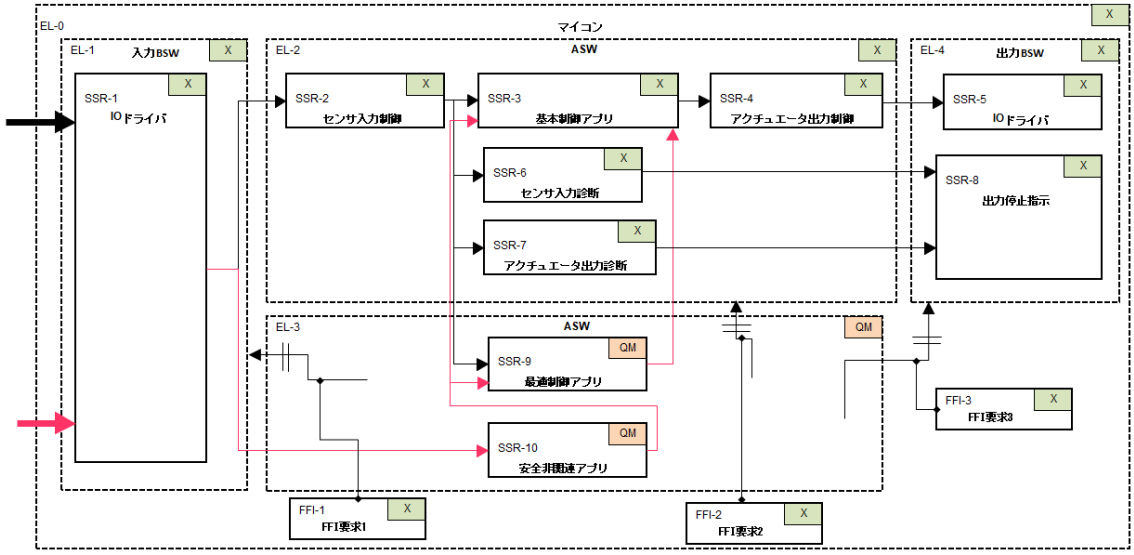


図 C- 4 仮想コントローラにおけるマイコン内ソフトウェア SCDL 記述例

ソフトウェアパーティショニングに関わる安全分析・従属故障分析では、低 ASIL アプリソフトから高 ASIL アプリソフトおよび高 ASIL 共通基盤部位へのカスケード故障の防止が重要な検討対象項目である。カスケード故障の防止（すなわち FFI 要求の満足）には、以下の３つの要素が必要である。

1) メモリ保護
2) 時間保護
3) データ保護

上記の３つの要素の中で、データ保護については、SCDL の持つ機能ブロック間のインタラクションの記述が安全分析・従属故障分析の支援に有効利用できる。データ保護では、低 ASIL 部位からの入力信号に対して、後段の高 ASIL 部位で、何らかの診断や保護を行う必要がある。低 ASIL 部位からの入力信号に対する診断や保護が、高 ASIL 部位の安全要求として適切に定義され配置されているかを、SCDL 上で見通しよくレビューできることが期待される。

さらに留意すべき点として、入力データに対する診断や保護を検討する上で、高 ASIL 部位からの入力データであっても、低 ASIL 部位からの入力データに対するのと同様の診断や保護が必要な場合があることである。

例えば、低 ASIL 部位からの入力データが、一旦、高 ASIL 基盤ソフトに入力され、その出力が、さらに高 ASIL アプリに入力されたとする。高 ASIL 基盤ソフトでは、低 ASIL 部位からの入力データに対して、ディフェンシブプログラミング（異常値であっても高 ASIL 基盤ソフトが誤動作しないための保護）は行っていても、当該入力データ値の最終的な安全性

の診断は行っていないケースが多い。このようなケースでは、当該入力データ値の最終的な安全性の診断は、後段の高 ASIL アプリソフトで行う必要がある。

具体例として、ASIL-B のペダルセンサからの入力データが、ASIL-D 基盤ソフトを経由してノイズ除去などされて、後段の ASIL-D アプリソフトに渡される場合に、その安全レベルは ASIL-D ではなく、ASIL-B のままである。後段の ASIL-D アプリソフトにおいて、他のペダルセンサからの入力データ(ASIL-B)と比較され、低い値が選択されたときに、その低い値に対して、ASIL-D の安全レベルが担保できる。

以上のように、安全分析・従属故障分析を検証する際に、各インタラクションで渡されるデータの安全レベルを色分けなどで区別表示することで、当該データの安全上の診断が適切に定義され配置されていることを、ビジュアル的に、容易に確認できることが期待される。なお、「図 C-3　仮想コントローラのマイコン内インタラクション　(アプリソフトに着目)」と、「図 C-4 仮想コントローラにおけるマイコン内ソフトウェア SCDL 記述例」では、各インタラクションで渡されるデータの安全レベルによって色分け表示した例を示した。

C.3 共通ライブラリの SCDL 記述例

安全コンセプトの検討において、ソフトウェアに関連する課題として、異なる ASIL ソフトで共通に利用される共通ライブラリをいかに記述するかがある。

本節では、共通ライブラリの実装方法を整理し、その記載事例を紹介する。

C.3.1 実装方式の説明

異なる ASIL ソフトで共通に利用される共通ライブラリの実装方式は２種類考えられる。本章では、高 ASIL ソフト側を ASIL-X と、低 ASIL ソフト側を QM と仮定して解説する。

第１の方式「分離実装方式」は、同一のライブラリを異なる ASIL に別々に実装する方式である。制約条件は、ASIL-X ソフトと QM ソフトとで RAM エリアを分けることである。なお ROM エリアは実装上共通／分離いずれでもよい。なお、コードは共通なので高 ASIL 開発プロセスを適用する必要がある。SCDL 記述上の特徴として、絵はシンプルであるが、実装とはイメージが異なる可能性がある。

第２の方式「共通実装方式」は、共通ライブラリを ASIL-X ソフト側に実装し、QM ソフト側からはスイッチングして使う方式である。制約条件は、QM ソフト側からコールされても安全目標侵害せぬこと、具体的には、QM 側からコール時にメモリ保護スイッチングを行う（OS マクロなどで対応する）ことや、ASIL 側の共通ライブラリにおいて、QM 側からの入力値に対してディフェンシブプログラミングでデータ保護することなどが必要である。第１の方式に比較すると、スイッチングなどが必要となり、処理負荷は増加する。SCDL 記述上の特徴として、実装とイメージのズレは出にくい。

両方式の相違点を比較したものを、図 C- 5 に示す。

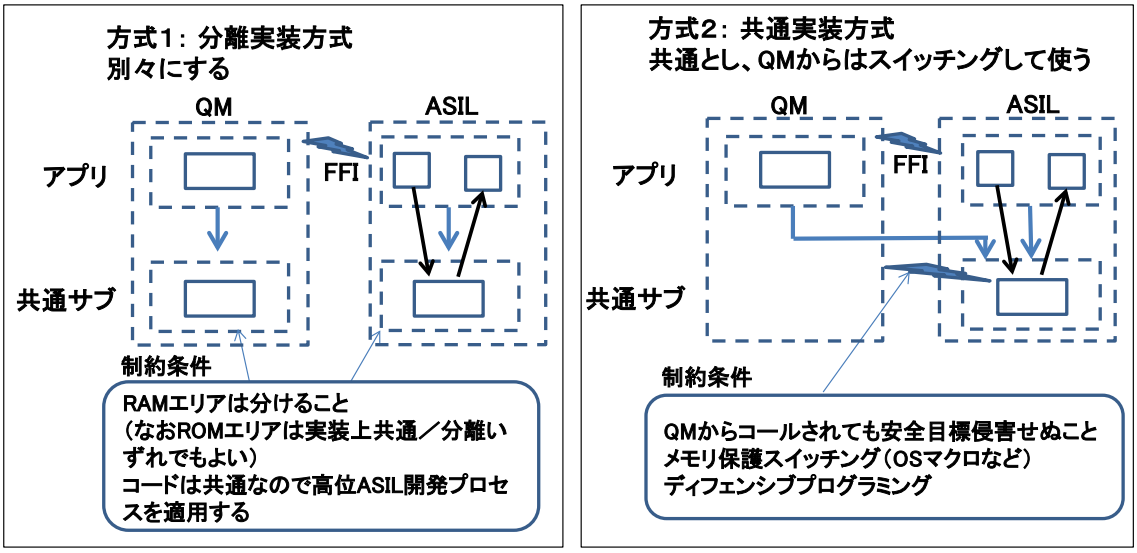


図 C- 5　異種 ASIL 間の共通ライブラリの実装方式比較

C.3.2 SCDL 記述例

異種 ASIL 間の共通ライブラリの SCDL 記述例を、図 C- 6 に示す。なお記述例にて示した関数コールは現時点での SCDL 仕様書では定義されていない記法であるが、今後仕様への反映も検討する。

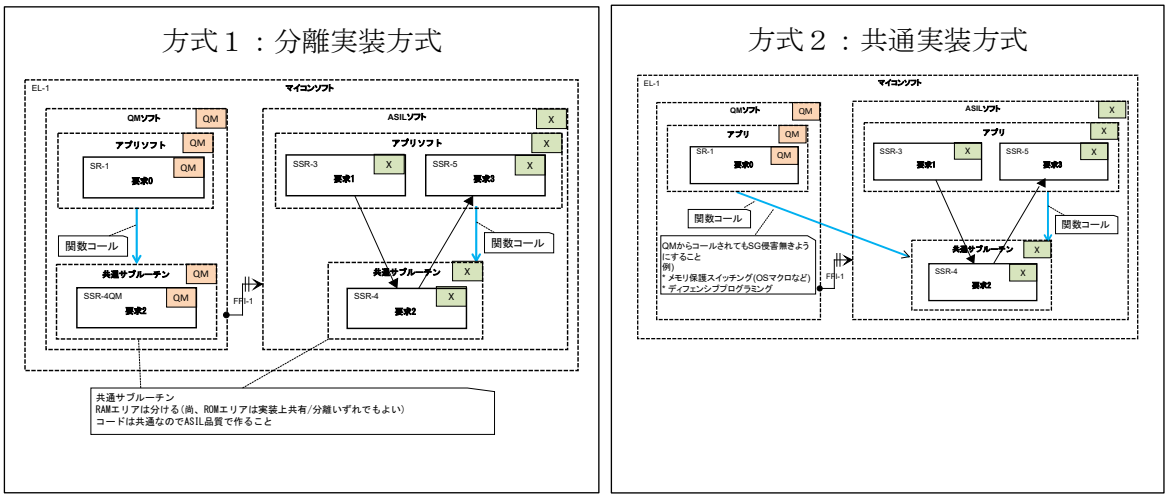


図 C- 6 異種 ASIL 間の共通ライブラリの記述例

SCDL 記述において、特殊な制約条件はコメントで付記した。第１の方式（分離実装方式）では、「RAM エリアを分けること」などを制約条件として記述した。また、第２の方式（共通実装方式）では、「メモリ保護スイッチングやディフェンシブプログラミング」などを制約条件として記述した。このように、異種 ASIL 間の共通ライブラリであっても、特殊な制約条件をコメントで付記することにより、SCDL 上で記述することができる。その結果、安全要求が適切に定義され配置されているかを、SCDL 上でビジュアルにイメージできるので、必要な要求の過不足を容易に確認できることが期待される。

附属書 D SCDL メタモデル

D.1 概要

本節では SCDL のメタモデルを示す。

図 D- 1 では、SCDL のメタモデルを構成する要素(以下、メタモデル要素と記載)間の基礎的な階層関係を表す。なお、SCDLType 要素以外のメタモデル要素の属性は省略した。

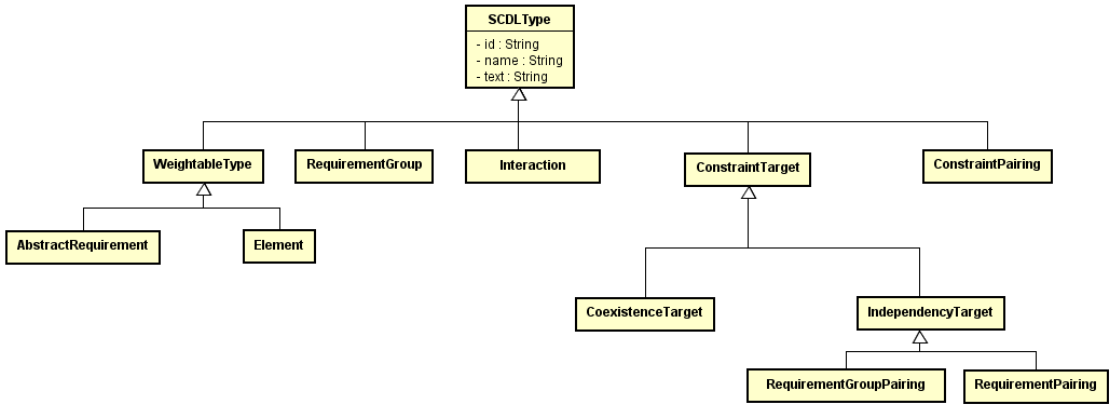


図 D- 1

次の図 D- 2 にて、SCDL のメタモデル要素間の関係と属性を表す。ただし、SCDLType 要素をこの図に含めると図が煩雑になるため、SCDLType 要素は省略した。

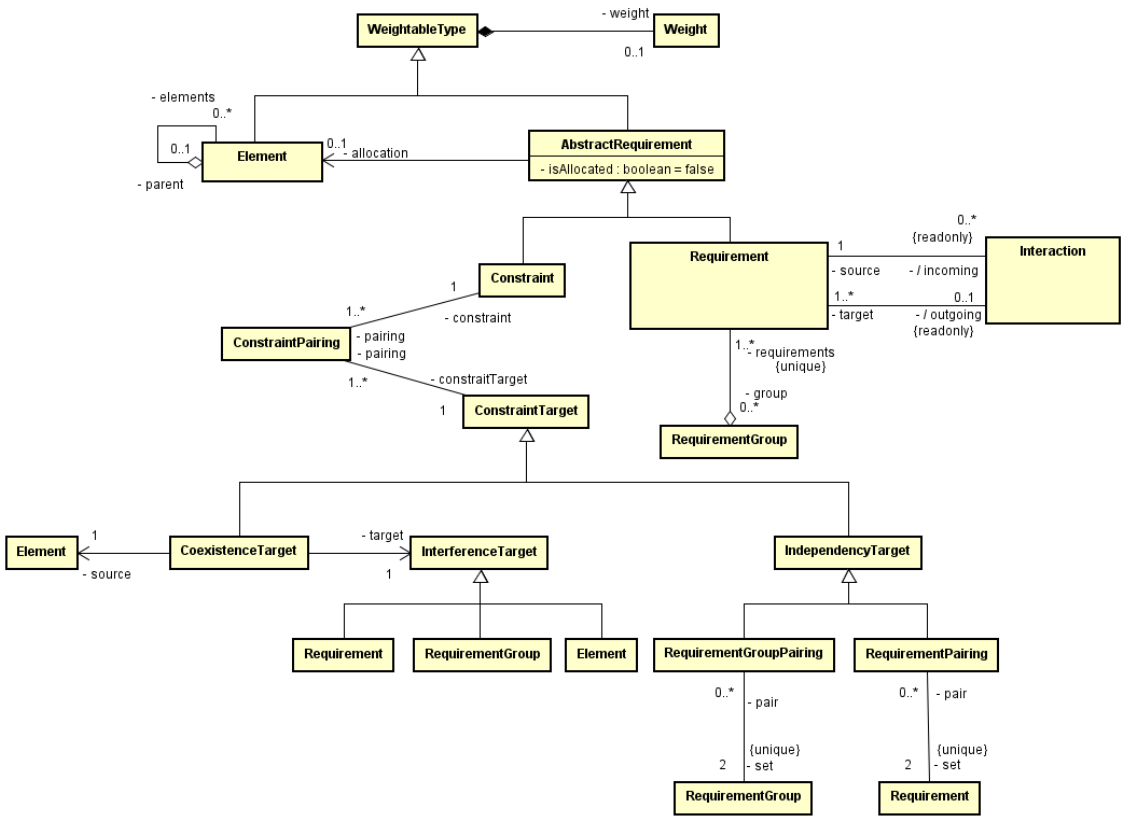


図 D- 2

次の表 D- 1 は、メタモデルで定義されるメタモデル要素と、仕様書記載の内容(表現)との対応表である。この表にないメタモデル要素は、メタモデルを構成するために追加した抽象的な要素(図には表現されない要素)である。

表 D- 1 仕様とメタモデルの対応付け

メタモデル	SCDL 仕様要素
Requirement	要求
RequirementGroup	要求グループ
Element	エレメント
Interaction	インタラクション システムバウンダリインタラクション
Constraint	制約条件
RequirementGroupPairing	要求グループペアリング
ConstraintPairing	制約条件からの引き出し線
CoexistenceTarget	無干渉
Weight	重み付け

D.2. 範囲

本節にて定義される SCDL のモデル範囲を示す。

- SCDL の基本定義
  - 要求
  - 要求グループ
  - エレメント
  - インタラクション
  - システムバウンダリインタラクション
  - 制約条件
  - 要求グループペアリング
  - 制約条件からの引き出し線
  - 無干渉
  - 重み付け

D.3. SCDLType (SCDL 型)

SCDLType は SCDL で定義される要素や関係のうち、id や名称などを持つ要素の基底となる抽象的な概念である。

D.3.1. Specializations

WeightableType, RequirementGroup, Interaction, ConstraintTarget, ConstraintPairing

D.3.2. Attributes

- id：String
  識別子
- name：String
  名称、もしくは短縮名称
- text：String
  備考

D.3.3. WeightableType (重み付け可能型)

WeightableType は、重み付けの情報を保持できる要素を示す抽象的な概念である。

D.3.4. Generalizations

SCDLType

D.3.5. Specializations

Element, AbstractRequirement

D.3.6. Association Ends

- weight: Weight[0..1]

D.4. Weight (重み付け)

重み付け要素。位置付けについては、仕様書 2.2 節を参照のこと。

D.5. AbstractRequirement (抽象要求)

AbstractRequirement は、Requirement 要素と Constraint 要素の共通部分を示す抽象的な概念である。

D.5.1. Generalizations

WeightableType

D.5.2. Constraints

- isAllocated が true の場合、allocation が存在する。
- isAllocated が false の場合には、allocation は存在しない。

D.5.3. Specializations

Requirement, Constraint

D.5.4. Attributes

- isAllocated: boolean = false
  false の場合、要求のエレメントへの配置先が決定されていないことを示す。true の場合は、要求はエレメントへ配置されている。

D.5.5. Association Ends

- allocation : Element[0..1]
  要求の配置先エレメント

D.6. Element (エレメント)

Element は、4.2.5 項記載のエレメントに対応する要素である。

D.6.1. Generalizations

WeightableType, InterferenceTarget

D.6.2. Constraints

- elements 内あるいは、再帰的に elements に含まれる Element の elements に自分自身の Element 要素が含まれてはならない。

D.6.3. Association Ends

- parent: Element[0..1]

エレメントの親エレメント

- elements: Element[0..*]
  エレメントが内包するエレメント

D.7. Requirement (要求)

Requirement は、4.2.1 項記載の要求に対応する要素である。

D.7.1. Generalizations

AbstractRequirement, InterferenceTarget

D.7.2. Association Ends

- /incoming : Interaction[0..*]
  要求に入るインタラクション
- /outgoing : Interaction[0..1]
  要求から出るインタラクション
- /group : RequirementGroup[0..*]
  要求が所属するグループ

D.8. Constraint (制約条件)

Constraint は、4.3.3 項記載の制約条件に対応する要素である。

D.8.1 Generalizations

AbstractRequirement

D.8.2 Association Ends

- pairing : ConstraintPairing[1..*]
  制約条件の対象を示す。

D.9. ConstraintPairing (制約条件との関連)

ConstraintPairing は、4.3.3 項で定義されるペアリングと制約条件間や、4.3.4 項で定義される非干渉要求と制約条件間を表す抽象的な概念である。

D.9.1. Generalizations

SCDLType

D.9.2. Association Ends

- constraint：Constraint[1..1]
  紐づく制約条件
- constraintTarget：ConstraintTarget[1..1]
  制約条件と紐づく対象

D.10. Interaction (インタラクション)

Interaction は、4.2.2 項記載のインタラクションに対応する要素である。

D.10.1. Generalizations

SCDLType

D.10.2. Constraints

- source と target は同じ Requirement 要素であってはならない。

D.10.3. Association Ends

- source：Requirement [1..1]
  インタラクションの入力側/From 側の要求
- target：Requirement [1..*]
  インタラクションの出力側/To 側の要求

D.11. RequirementGroup (要求グループ)

RequirementGroup は、4.3.2 項記載の要求グループに対応する要素である。

D.11.1 Generalizations

SCDLType, InterferenceTarget

D.11.2. Constraints

- requirements に含まれる Requirement 要素は重複してはならない。

D.11.3. Association Ends

- requirements：Requirement[1..*]
  要求グループに属する要求

D.12. ConstraintTarget (制約条件紐づけ対象)

ConstraintTarget は、ある制約条件に対して、その制約の適用先となる対象を示す抽象的な概念である。

D.12.1. Generalizations

SCDLType

D.12.2. Specializations

CoexistenceTarget, IndependencyTarget

D.12.3. Association Ends

- pairing: ConstraintPairing [1..*]

D.13 IndependencyTarget (独立対象)

IndependencyTarget は、4.3.3 項で定義されるペアリングと制約条件間を表す抽象的な概念である。

D.13.1. Generalizations

ConstraintTarget

D.13.2. Specializations

RequirementGroupPairing, RequirementPairing

D.14. CoexistenceTarget (共存対象)

CoexistenceTarget は、4.3.4 項記載の無干渉関連に対応する要素である。

D.14.1. Generalizations

ConstraintTarget

D.14.2. Association Ends

- target：InterferenceTarget [1..1]
  干渉先の要素を示す。
- source：Element[1..1]
  干渉元の要素を示す。

D.15. InterferenceTarget (干渉波及先)

InterferenceTarget は、無干渉関連の干渉先を示す抽象的な概念である。

D.15.1. Specializations

Requirement, RequirementGroup, Element

D.16. RequirementGroupPairing (要求グループペアリング)

RequirementGroupPairing は、4.3.2 項記載の要求グループペアリングに対応する要素である。

D.16.1. Generalizations

IndependencyTarget

D.16.2. Constraints

set の 2 つの RequirementGroup 要素は、異なる要素でなければならない。

D.16.3. Association Ends

- set：RequirementGroup[2..2]
  ペアリング関係にある 2 つの要求グループ

D.17. RequirementPairing (部分ペアリング)

RequirementPairing は、4.3.3 項の 2 に記載の要求グループペアリングを詳細化した要求間の部分的な関連に対応する要素である。

D.17.1. Generalizations

IndependencyTarget

D.17.2. Constraints

set の 2 つの Requirement 要素は、異なる要素でなければならない。

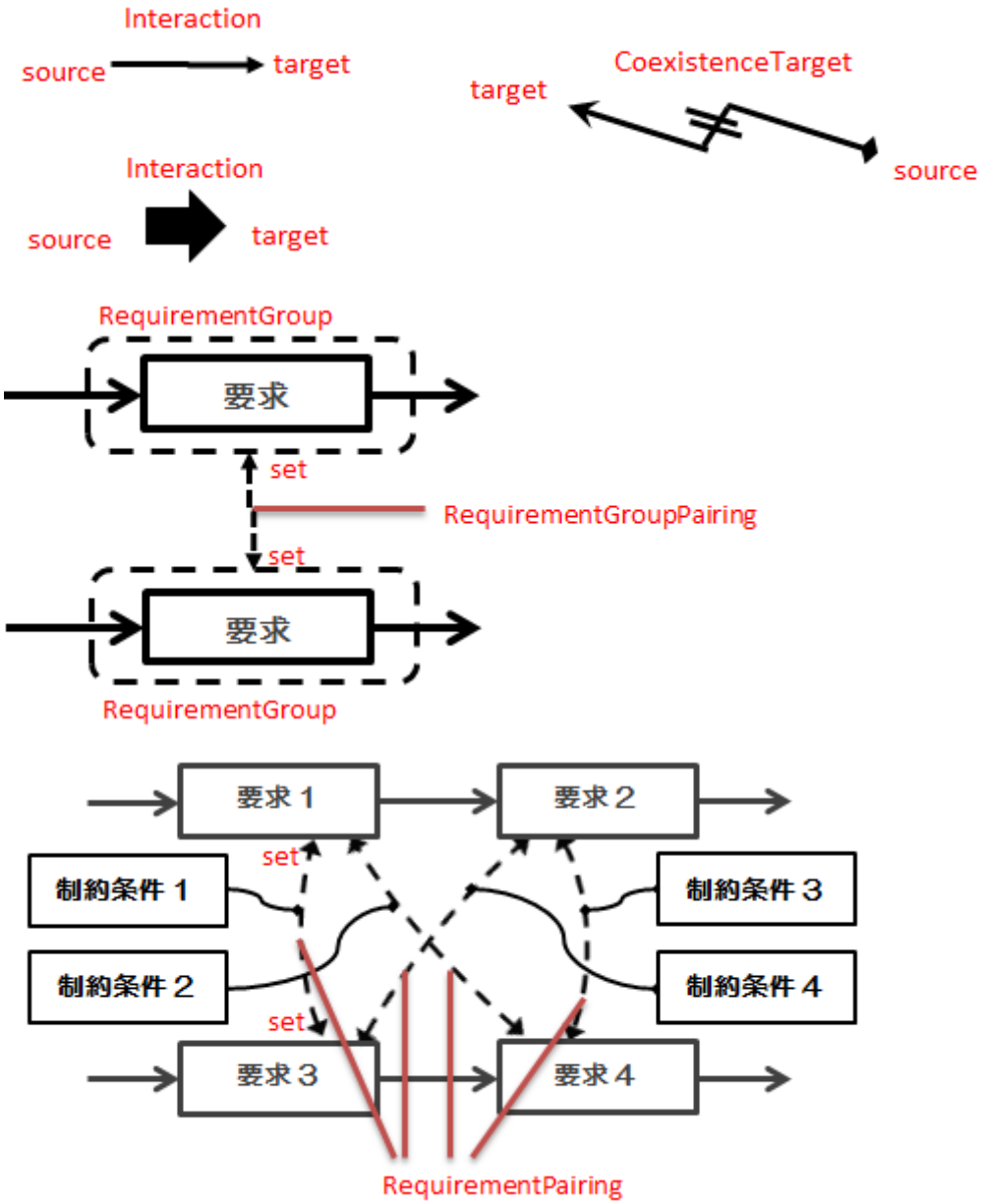
D.17.3. Association Ends

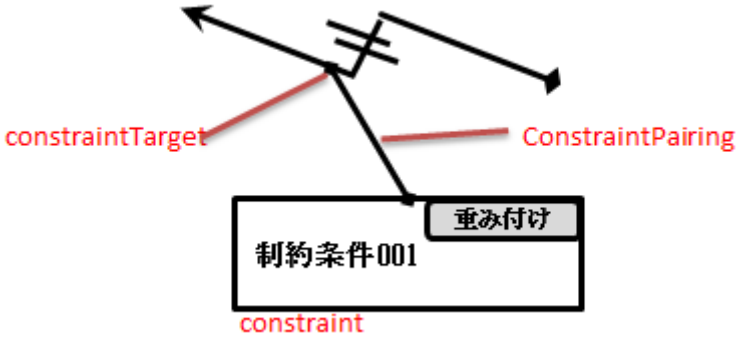
- set：Requirement[2..2]
  独立条件関係にある 2 つの要求

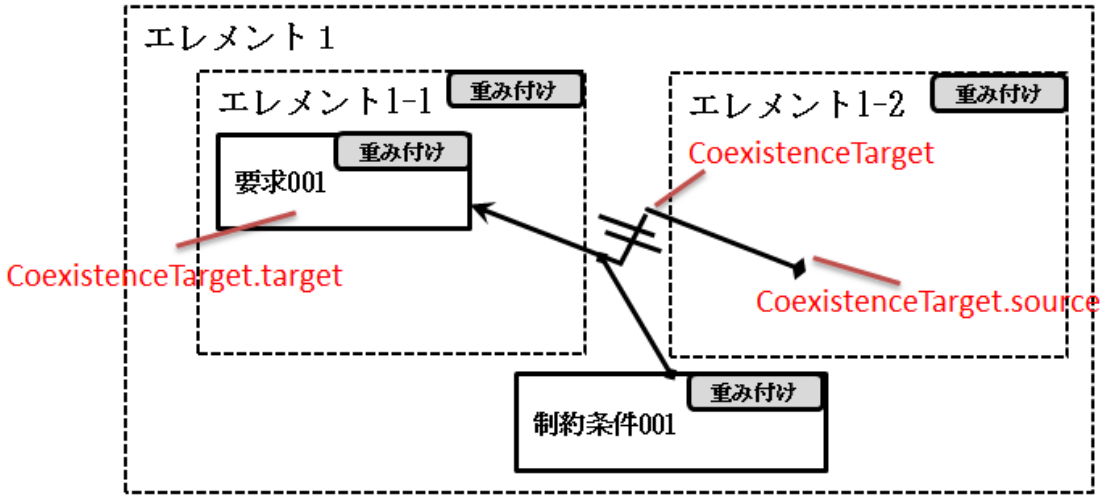
D.18. SCDL メタモデルと図の対応

本節では、メタモデルの理解の助けとなるために、本仕様書に記載のいくつかの図について、メタモデル要素との対応付けを示す。

D.18.1 仕様書における図での対応

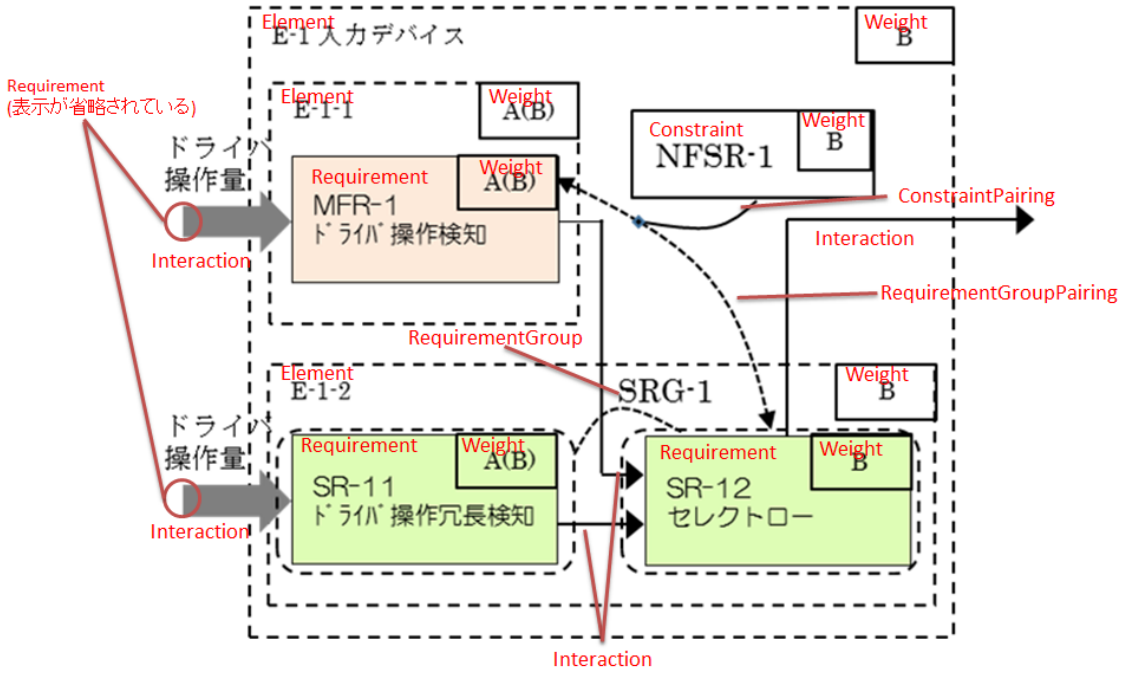




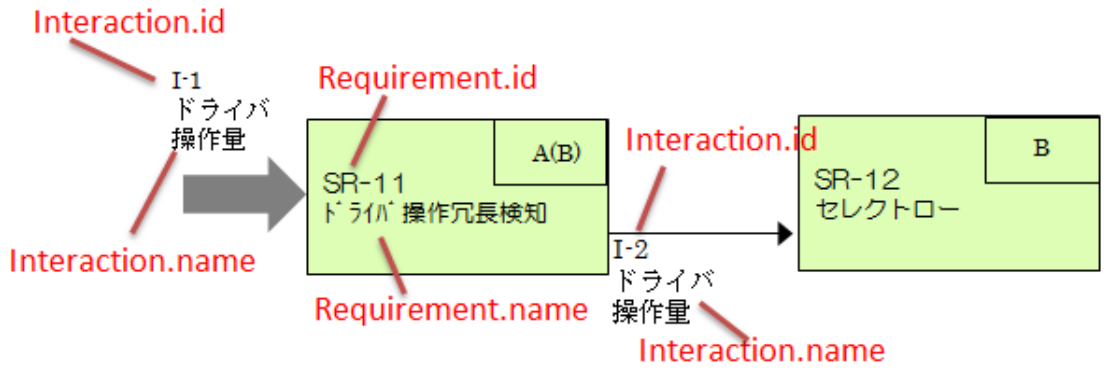


D.18.2. ユースケースで利用されている図を例にした対応

メタモデル要素との対応:



メタモデル要素の属性との対応:



附属書 E　モデル入出力・データ交換

E.1. 概要

本節では SCDL のモデルの入出力・データ交換の概要を示す。

SCDL を利用して記述したモデルは、機能安全を実現する最上流のデータとして、重要な価値を持つ。このデータをさらに設計など後工程につなげて活用することは、それら安全設計にかかわるユーザにとって、正確性・効率性・トレーサビリティなど非常にメリットが大きくなる。

本モデル入出力・データ交換では、前述の実現のために、SCDL ツール間の互換性のみならず、後工程など様々なツールへのオープンな連携を目指した。

本ドキュメントでは SCDL のメタモデルの構成を基礎としたモデル入出力・データ交換の内容・形式・構成を定義する。

定義の形式としては、XML を採用した。これは、XML が現在書式として広く利用されており、プログラム等で XML 形式のデータを扱うためのライブラリも充実していることが理由である。

なお、入出力の対象は、SCDL メタモデルとして定義されている範囲とする。

E.2. 全体の階層構造

本ドキュメントにて定義される XML の構成を示す。なお、それぞれのノードが持つ属性は省略し、階層構造のみを示している。

- scdl
  - model
    - documentation
    - diagrams
      - diagram
        - type
          - element
            - graphicsinfo
            - extensions
              - extension

- requirement
  - graphicsinfo
  - extensions
    - extension
- constraint
  - graphicsinfo
  - extensions
    - extension
- requirementGroup
  - graphicsinfo
  - extensions
    - extension

- relation
  - containerElementOwnElement
    - extensions
      - extension
  - containerElementOwnRequirement
    - extensions
      - extension
  - containerRequirementGroupOwnRequirement
    - points
      - point
    - extensions
      - extension
  - interaction
    - points
      - point
    - extensions
      - extension
  - coexistenceTarget
    - points
      - point
    - extensions
      - extension
  - requirementGroupPairing
    - points

        - point
      - extensions
        - extension
    - requirementPairing
      - points
        - point
      - extensions
        - extension
    - constraintPairing
      - points
        - point
      - extensions
        - extension
  - extensions
    - extension
- extensions
  - extension

次の節にて、それぞれのノードが持つ属性を定義する。

E.3. 共通の制約事項

以下の内容は、この仕様書を通しての共通の制約事項である。

- すべての属性は必ず定義すること。値がない場合には空文字列を持つ属性を定義すること。ただし、「必須」と定義されている属性については、空文字列は許容されない。
- “uuid”の名前を持つ属性は、対応する要素や関係などを一意に示すための文字列である。1 つの XML ファイル内において、同一の要素や関係が複数箇所に現れる場合、同一の uuid の値を持たなければならない。
  具体的な形式には制限はない。文字列の長さは、1 以上とする。
- ダイアグラム内での位置情報および長さ・幅は、任意の正の数値(小数値も可能)とする。なお、位置情報は、描画領域の左上を(0,0)とし、右方向および下方向に数値が増大するような関係で示すものとする。
  なお、数値は相対的なものとし、実際の画面上の表示における大きさとの相関関係は本ドキュメントでは定義しない。
- この仕様では、データの利用者の便宜を図った結果、同じ情報を異なる 2 カ所のノードに出力する情報がある。この情報について、内容に矛盾があってはならないものとする。
  (例: エレメントの上位・下位関係は、element.parent_uuid と、container の両方で示される。)

E.3.1. scdl

scdl ノードは、最上位となるノードである。

E.3.1.1. 下位ノード

model : [1]

E.3.1.2. 属性

- version (必須)
  データ形式のバージョンを示す固定文字列。本ドキュメントに準拠する場合には、“1.0”を設定すること。

E.3.2. model

model ノードは、実質的な最上位となるノードである。

E.3.2.1. 下位ノード

documentation: [0..1]

diagrams: [1]

extensions: [0..1]

E.3.2.2. 属性

- id
  出力対象を識別するための ID。
- name
  出力対象を識別するための文字列。

E.3.3. documentation

documentation ノードは、出力ファイルに関係する追加情報を格納するノードである。

E.3.3.1. 下位ノード

なし

E.3.3.2. 属性

- documentVersion
  出力内容のバージョン(履歴)を示す文字列。
- description
  出力内容の説明を示す文字列。
- author
  出力内容の作者を示す文字列。
- timestamp
  出力時刻を示す文字列。時刻の表現形式については規定しない。

E.3.4. diagrams

diagram ノードを格納するためのコレクションとなるノードである。

E.3.4.1. 下位ノード

diagram: [1..*]

E.3.4.2. 属性

なし

E.3.5. diagram

diagram ノードは、それぞれの図に対応する情報を持つノードである。

E.3.5.1. 下位ノード

type: [1]

relation: [1]

extensions: [0..1]

E.3.5.2. 属性

- uuid (必須)
  ダイアグラムを識別するための文字列。
- name
  ダイアグラムの名称を示す文字列。
- text
  ダイアグラムの説明を示す文字列。

E.3.6. type

type ノードは、ダイアグラム内に配置された要素を格納するためのコレクションとなるノードである。

E.3.6.1. 下位ノード

element: [0..*]

requirement: [0..*]

constraint: [0..*]

requirementGroup: [0..*]

E.3.6.2. 属性

なし

E.3.7. element

element ノードは、エレメントに対応するノードである。

E.3.7.1. 下位ノード

graphicsinfo: [0..1]

extensions: [0..1]

E.3.7.2. 属性

- uuid (必須)
  エレメントを識別するための文字列。
- id
  設計者がエレメントを識別するための文字列。
- name
  エレメントの名称を示す文字列。
- text
  エレメントの説明を示す文字列。
- weight
  エレメントの重み付けを示す文字列。
- parent_uuid
  このエレメントが他の(上位の)エレメントに配置されている場合に、そのエレメントを示すエレメントの uuid を示す文字列。

E.3.8. graphicsinfo

graphicsinfo ノードは、ダイアグラム内での位置情報を示すためのノードである。

E.3.8.1. 下位ノード

なし

E.3.8.2. 属性

- x, y (必須)
  要素の左上の位置を示す数値。
- width (必須)
  要素の幅を示す数値。
- height (必須)
  要素の高さを示す数値。

E.3.9. requirement

requirement ノードは、要求に対応するノードである。

E.3.9.1. 下位ノード

graphicsinfo: [0..1]

extensions: [0..1]

E.3.9.2. 属性

- uuid (必須)
  要求を識別するための文字列。
- type
  要求の種類を示す文字列。
- id
  設計者が要求を識別するための文字列。
- name
  要求の名称を示す文字列。
- text
  要求の説明を示す文字列。
- weight
  要求の重み付けを示す文字列。
- parent_uuid
  この要求がエレメントに配置されている場合に、そのエレメントを示す uuid を示す文字列。

E.3.10. constraint

constraint ノードは、制約条件に対応するノードである。

E.3.10.1. 下位ノード

graphicsinfo: [0..1]
extensions: [0..1]

E.3.10.2. 属性

- uuid (必須)
  制約条件を識別するための文字列。
- type
  制約条件の種類を示す文字列。
- id
  設計者が制約条件を識別するための文字列。
- name
  制約条件の名称を示す文字列。
- text
  制約条件の説明を示す文字列。

- weight
  制約条件の重み付けを示す文字列。
- parent_uuid
  この制約条件がエレメントに配置されている場合に、そのエレメントを示す uuid を示す文字列。

E.3.11. requirementGroup

requirementGroup ノードは、楕円形式で表現する要求グループに対応するノードである。

E.3.11.1. 下位ノード

graphicsinfo: [0..1]
extensions: [0..1]

E.3.11.2. 属性

- uuid （必須）
  要求グループを識別するための文字列。
- id
  設計者が要求グループを識別するための文字列。
- name
  要求グループの名称を示す文字列。
- text
  要求グループの説明を示す文字列。

E.3.12. relation

relation ノードは、ダイアグラム内に配置された要素間の関係を格納するためのコレクションとなるノードである。

E.3.12.1. 下位ノード

containerElementOwnElement: [0..*]
containerElementOwnRequirement: [0..*]
containerRequirementGroupOwnRequirement: [0..*]
interaction: [0..*]
coexistenceTarget: [0..*]
requirementGroupPairing: [0..*]
requirementPairing: [0..*]
constraintPairing: [0..*]

E.3.12.2. 属性

なし

E.3.13. containerElementOwnElement

containerElementOwnElement ノードは、エレメントとエレメント間の関係を示すためのノードである。

E.3.13.1. 下位ノード

extensions: [0..1]

E.3.13.2. 属性

- parent_uuid （必須）
  上位側エレメント(含む側のエレメント)の uuid を示す文字列。
- child_uuid （必須）
  下位側エレメント(含まれる側のエレメント)の uuid を示す文字列。
- parent_id
  上位側エレメント(含む側のエレメント)の、設計者がエレメントを識別するための文字列。
- child_id
  下位側エレメント(含まれる側のエレメント)の、設計者がエレメントを識別するための文字列。

E.3.14. containerElementOwnRequirement

containerElementOwnRequirement ノードは、エレメントと要求間の関係を示すためのノードである。

E.3.14.1. 下位ノード

extensions: [0..1]

E.3.14.2. 属性

- parent_uuid （必須）
  エレメントの uuid を示す文字列。
- child_uuid （必須）
  要求の uuid を示す文字列。
- parent_id

設計者がエレメントを識別するための文字列。

- child_id
  設計者が要求を識別するための文字列。

E.3.15. containerRequirementGroupOwnRequirement

containerRequirementGroupOwnRequirement ノードは、要求グループと要求間の関係を示すためのノードである。

E.3.15.1. 下位ノード

points: [0..1]
extensions: [0..1]

E.3.15.2. 属性

- uuid (必須)
  関係を識別するための文字列。
- id
  設計者が関係を識別するための文字列。
- name
  関係の名称を示す文字列。
- text
  関係の説明を示す文字列。
- parent_uuid (必須)
  要求グループの uuid を示す文字列。
- child_uuid (必須)
  要求の uuid を示す文字列。

E.3.16. points

points ノードは、point ノードを格納するためのコレクションとなるノードである。

point ノードは 2 つ以上が必要であり、向きがある関係の場合には、元側の要素から先側の要素に向かって順番に位置情報を含む。結果的に、コレクションの 1 番目の point ノードは、元側の要素との端点を示し、コレクションの最後の point ノードは先側の要素との端点を示すことになる。

E.3.16.1. 下位ノード

point: 2..*

E.3.16.2. 属性

なし

E.3.17. point

point ノードは、関係の両端あるいは折れ曲がる点のダイアグラム内での位置情報を示すためのノートである。

E.3.17.1. 下位ノード

なし

E.3.17.2. 属性

- x, y (必須)
  点の位置を示す数値。

E.3.18. interaction

interaction ノードは、インタラクションおよびシステムバウンダリインタラクションに対応するノードである。

E.3.18.1. 下位ノード

points: [0..1]
extensions: [0..1]

E.3.18.2. 属性

- uuid (必須)
  インタラクションを識別するための文字列。
- id
  設計者がインタラクションを識別するための文字列。
- name
  インタラクションの名称を示す文字列。
- text
  インタラクションの説明を示す文字列。
- source_id
  インタラクションの送信元側の要求を識別するための文字列。
- targete_id
  インタラクションの送信先側の要求を識別するための文字列。

- source_uuid
  インタラクションの送信元側の要求の uuid を示す文字列。
- target_uuid
  インタラクションの送信先側の要求の uuid を示す文字列。

E.3.19. coexistenceTarget

coexistenceTarget ノードは、無干渉に対応するノードである。

E.3.19.1. 下位ノード

points: [0..1]
extensions: [0..1]

E.3.19.2. 属性

- uuid （必須）
  無干渉を識別するための文字列。
- id
  設計者が無干渉を識別するための文字列。
- name
  無干渉の名称を示す文字列。
- text
  無干渉の説明を示す文字列。
- source_uuid
  無干渉の送信元側の要素の uuid を示す文字列。
- target_uuid
  無干渉の送信先側の要素の uuid を示す文字列。

E.3.20. requirementGroupPairing

requirementGroupPairing ノードは、要求グループペアリングに対応するノードである。

E.3.20.1. 下位ノード

points: [0..1]
extensions: [0..1]

E.3.20.2. 属性

- uuid （必須）
  要求グループペアリングを識別するための文字列。

- id
  設計者が要求グループペアリングを識別するための文字列。
- name
  要求グループペアリングの名称を示す文字列。
- text
  要求グループペアリングの説明を示す文字列。
- set1_uuid
  要求グループペアリングの対象の要求の uuid を示す文字列。
- set2_uuid
  要求グループペアリングの対象の要求の uuid を示す文字列。

E.3.21. requirementPairing

requirementPairing ノードは、部分ペアリングに対応するノードである。

E.3.21.1. 下位ノード

points: [0..1]

extensions: [0..1]

E. 3.21.2. 属性

- uuid （必須）
  部分ペアリングを識別するための文字列。
- id
  設計者が部分ペアリングを識別するための文字列。
- name
  部分ペアリングの名称を示す文字列。
- text
  部分ペアリングの説明を示す文字列。
- set1_uuid
  部分ペアリングの対象の要求の uuid を示す文字列。
- set2_uuid
  部分ペアリングの対象の要求の uuid を示す文字列。

E.3.22. constraintPairing

constraintPairing ノードは、制約条件と要求グループペアリングとの関係(引き出し線)に対応するノードである。

E.3.22.1. 下位ノード

points: [0..1]

extensions: [0..1]

E.3.22.2. 属性

- uuid （必須）
  引き出し線を識別するための文字列。
- id
  設計者が引き出し線を識別するための文字列。
- name
  引き出し線の名称を示す文字列。
- text
  引き出し線の説明を示す文字列。
- constraint_uuid
  引き出し線の対象の制約条件の uuid を示す文字列。
- constraintTarget_uuid
  引き出し線の対象の要求グループペアリングの uuid を示す文字列。

E.3.23. extensions

extensions ノードは、extension ノードを格納するためのコレクションとなるノードである。

E.3.23.1. 下位ノード

extension: [0..*]

E.3.23.2. 属性

なし

E.3.24. extension

extension ノードは、このデータを扱うそれぞれのツールごとの独自の情報を格納するためのノードである。

このデータを扱うそれぞれのツールは、exportVendor 属性として格納する文字列を定めて情報を格納する。既存のデータを編集する際には、exportVendor 属性の値が自分自身のツールを示す値ではない場合、そのノードの編集・削除は行ってはならない。

E.3.24.1. 下位ノード

任意（ツールごとに独自に定義可能）

E.3.24.2. 属性

- exportVendor （必須）
  独自情報を設定するツールなどを示す文字列。

E.4. スキーマ

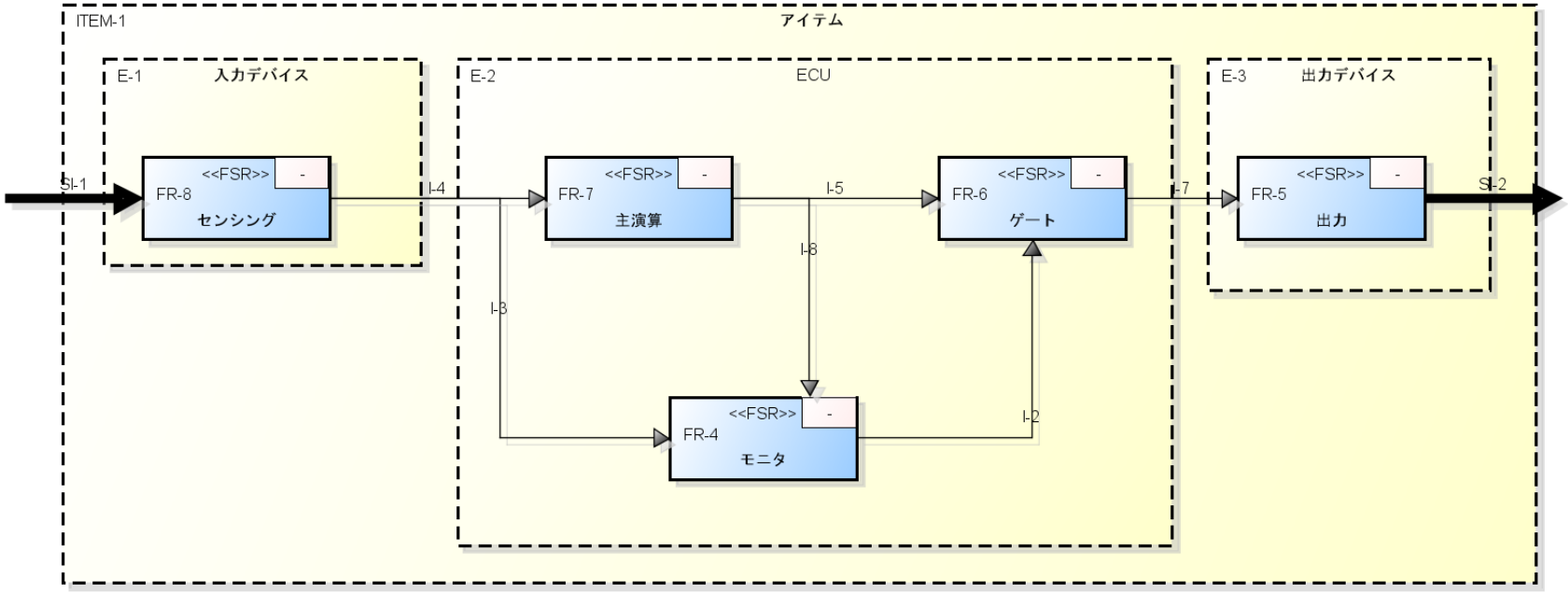
この SCDL モデル入出力・データ交換のスキーマ(XSD 形式のファイル)は、安全コンセプト記法研究会（SCN-SG）の Web サイトからダウンロード可能である。

なお、スキーマの namespace は、”http://www.scn-sg.com/2018/04/scdl” である。

E.5. サンプル

本仕様に基づくサンプルデータを次ページ以降に示す。

○SCDL モデル



○XML 形式の表現

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<scdl version="1.0">
  <model id="" name="XML サンプル 1.0.scdl">
    <diagrams>
      <diagram name="Sample1.0" text="" uuid="1h9m-bd9dbaca97bf99bc1b3de839343d0157">
        <type>
          <requirement id="FR-8" name="センシング" parent_uuid="2oyn-bd9dbaca97bf99bc1b3de839343d0157" text="" type="FSR" uuid="2oz5-bd9dbaca97bf99bc1b3de839343d0157" weight="-">
            <graphicsinfo height="61.5" width="140.0" x="232.75" y="290.5"/>
            <extensions>
              <extension exportVendor="Sample Company">
                <graphicsinfo bgcolor="#FF99CCFF" linecolor="#FF000000" linetype="line"/>
              </extension>
            </extensions>
          </requirement>
          <requirement id="FR-7" name="主演算" parent_uuid="2oyw-bd9dbaca97bf99bc1b3de839343d0157" text="" type="FSR" uuid="2ozg-bd9dbaca97bf99bc1b3de839343d0157" weight="-">
            <graphicsinfo height="61.5" width="140.0" x="534.75" y="290.5"/>
            <extensions>
              <extension exportVendor="Sample Company">
                <graphicsinfo bgcolor="#FF99CCFF" linecolor="#FF000000" linetype="line"/>
              </extension>
            </extensions>
          </requirement>
          <requirement id="FR-6" name="ゲート" parent_uuid="2oyw-bd9dbaca97bf99bc1b3de839343d0157" text="" type="FSR" uuid="2ozr-bd9dbaca97bf99bc1b3de839343d0157" weight="-">
            <graphicsinfo height="61.5" width="140.0" x="829.75" y="290.5"/>
            <extensions>
              <extension exportVendor="Sample Company">
                <graphicsinfo bgcolor="#FF99CCFF" linecolor="#FF000000" linetype="line"/>
              </extension>
            </extensions>
          </requirement>
          <requirement id="FR-5" name="出力" parent_uuid="2oye-bd9dbaca97bf99bc1b3de839343d0157" text="" type="FSR" uuid="2p02-bd9dbaca97bf99bc1b3de839343d0157" weight="-">
            <graphicsinfo height="61.5" width="140.0" x="1054.25" y="290.5"/>
            <extensions>
              <extension exportVendor="Sample Company">
                <graphicsinfo bgcolor="#FF99CCFF" linecolor="#FF000000" linetype="line"/>
              </extension>
            </extensions>
          </requirement>
          <requirement id="FR-4" name="モニタ" parent_uuid="2oyw-bd9dbaca97bf99bc1b3de839343d0157" text="" type="FSR" uuid="2p0d-bd9dbaca97bf99bc1b3de839343d0157" weight="-">
            <graphicsinfo height="61.5" width="140.0" x="628.25" y="470.0"/>
            <extensions>
```

```
        <extension exportVendor="Sample Company">
          <graphicsinfo bgcolor="#FF99CCFF" linecolor="#FF000000" linetype="line"/>
        </extension>
      </extensions>
    </requirement>
    <requirement id="" name="" parent_uuid="" text="" type="ExternalRequirement" uuid="69c9e302-d06c-38ab-b3b2-b0af7f82931e" weight="-">
      <graphicsinfo height="1.0" width="1.0" x="114.25" y="313.75"/>
    </requirement>
    <requirement id="" name="" parent_uuid="" text="" type="ExternalRequirement" uuid="367c0a32-e582-3295-aaea-52ce92b3112f" weight="-">
      <graphicsinfo height="1.0" width="1.0" x="1296.75" y="313.75"/>
    </requirement>
    <element id="ITEM-1" name="アイテム" parent_uuid="" text="" uuid="1i9l-bd9dbaca97bf99bc1b3de839343d0157" weight="not specified">
      <graphicsinfo height="433.0" width="1105.0" x="172.6500000000001" y="176.0"/>
      <extensions>
        <extension exportVendor="Sample Company">
          <graphicsinfo bgcolor="#FFFFFFCC" linecolor="#FF000000" linetype="dash2"/>
        </extension>
      </extensions>
    </element>
    <element id="E-3" name="出力デバイス" parent_uuid="1i9l-bd9dbaca97bf99bc1b3de839343d0157" text="" uuid="2oye-bd9dbaca97bf99bc1b3de839343d0157" weight="not specified">
      <graphicsinfo height="173.0" width="210.9000000000001" x="1031.75" y="217.0"/>
      <extensions>
        <extension exportVendor="Sample Company">
          <graphicsinfo bgcolor="#FFFFFFCC" linecolor="#FF000000" linetype="dash2"/>
        </extension>
      </extensions>
    </element>
    <element id="E-1" name="入力デバイス" parent_uuid="1i9l-bd9dbaca97bf99bc1b3de839343d0157" text="" uuid="2oyn-bd9dbaca97bf99bc1b3de839343d0157" weight="not specified">
      <graphicsinfo height="154.0" width="237.60000000000002" x="203.64999999999998" y="217.0"/>
      <extensions>
        <extension exportVendor="Sample Company">
          <graphicsinfo bgcolor="#FFFFFFCC" linecolor="#FF000000" linetype="dash2"/>
        </extension>
      </extensions>
    </element>
    <element id="E-2" name="ECU" parent_uuid="1i9l-bd9dbaca97bf99bc1b3de839343d0157" text="" uuid="2oyw-bd9dbaca97bf99bc1b3de839343d0157" weight="not specified">
      <graphicsinfo height="364.09375" width="535.9" x="468.75" y="216.90625"/>
      <extensions>
        <extension exportVendor="Sample Company">
          <graphicsinfo bgcolor="#FFFFFFCC" linecolor="#FF000000" linetype="dash2"/>
        </extension>
      </extensions>
    </element>
  </type>
  <relation>
```

```
        <containerElementOwnElement child_id="E-3" child_uuid="2oye-bd9dbaca97bf99bc1b3de839343d0157" parent_id="ITEM-1" parent_uuid="1i9l-bd9dbaca97bf99bc1b3de839343d0157"/>
        <containerElementOwnElement child_id="E-1" child_uuid="2oyn-bd9dbaca97bf99bc1b3de839343d0157" parent_id="ITEM-1" parent_uuid="1i9l-bd9dbaca97bf99bc1b3de839343d0157"/>
        <containerElementOwnElement child_id="E-2" child_uuid="2oyw-bd9dbaca97bf99bc1b3de839343d0157" parent_id="ITEM-1" parent_uuid="1i9l-bd9dbaca97bf99bc1b3de839343d0157"/>
        <containerElementOwnRequirement child_id="FR-8" child_uuid="2oz5-bd9dbaca97bf99bc1b3de839343d0157" parent_id="E-1" parent_uuid="2oyn-bd9dbaca97bf99bc1b3de839343d0157"/>
        <containerElementOwnRequirement child_id="FR-7" child_uuid="2ozg-bd9dbaca97bf99bc1b3de839343d0157" parent_id="E-2" parent_uuid="2oyw-bd9dbaca97bf99bc1b3de839343d0157"/>
        <containerElementOwnRequirement child_id="FR-6" child_uuid="2ozr-bd9dbaca97bf99bc1b3de839343d0157" parent_id="E-2" parent_uuid="2oyw-bd9dbaca97bf99bc1b3de839343d0157"/>
        <containerElementOwnRequirement child_id="FR-5" child_uuid="2p02-bd9dbaca97bf99bc1b3de839343d0157" parent_id="E-3" parent_uuid="2oye-bd9dbaca97bf99bc1b3de839343d0157"/>
        <containerElementOwnRequirement child_id="FR-4" child_uuid="2p0d-bd9dbaca97bf99bc1b3de839343d0157" parent_id="E-2" parent_uuid="2oyw-bd9dbaca97bf99bc1b3de839343d0157"/>
        <interaction id="I-2" name="" source_id="FR-4" source_uuid="2p0d-bd9dbaca97bf99bc1b3de839343d0157" target_id="FR-6" target_uuid="2ozr-bd9dbaca97bf99bc1b3de839343d0157" text=""
uuid="2p32-bd9dbaca97bf99bc1b3de839343d0157">
          <points>
            <point x="698.25" y="500.75"/>
            <point x="899.75" y="500.75"/>
            <point x="899.75" y="321.25"/>
          </points>
        </interaction>
        <interaction id="I-3" name="" source_id="FR-8" source_uuid="2oz5-bd9dbaca97bf99bc1b3de839343d0157" target_id="FR-4" target_uuid="2p0d-bd9dbaca97bf99bc1b3de839343d0157" text=""
uuid="2p47-bd9dbaca97bf99bc1b3de839343d0157">
          <points>
            <point x="302.75" y="321.25"/>
            <point x="500.5" y="321.25"/>
            <point x="500.5" y="500.75"/>
            <point x="698.25" y="500.75"/>
          </points>
        </interaction>
        <interaction id="I-4" name="" source_id="FR-8" source_uuid="2oz5-bd9dbaca97bf99bc1b3de839343d0157" target_id="FR-7" target_uuid="2ozg-bd9dbaca97bf99bc1b3de839343d0157" text=""
uuid="2p5c-bd9dbaca97bf99bc1b3de839343d0157">
          <points>
            <point x="305.25" y="321.25"/>
            <point x="604.75" y="321.25"/>
          </points>
        </interaction>
        <interaction id="I-5" name="" source_id="FR-7" source_uuid="2ozg-bd9dbaca97bf99bc1b3de839343d0157" target_id="FR-6" target_uuid="2ozr-bd9dbaca97bf99bc1b3de839343d0157" text=""
uuid="2p6h-bd9dbaca97bf99bc1b3de839343d0157">
          <points>
            <point x="607.25" y="321.25"/>
            <point x="899.75" y="321.25"/>
          </points>
        </interaction>
        <interaction id="I-7" name="" source_id="FR-6" source_uuid="2ozr-bd9dbaca97bf99bc1b3de839343d0157" target_id="FR-5" target_uuid="2p02-bd9dbaca97bf99bc1b3de839343d0157" text=""
uuid="2p7m-bd9dbaca97bf99bc1b3de839343d0157">
          <points>
            <point x="902.25" y="321.25"/>
            <point x="1124.25" y="321.25"/>
          </points>
```

```
            </interaction>
            <interaction id="I-8" name="" source_id="FR-7" source_uuid="2ozg-bd9dbaca97bf99bc1b3de839343d0157" target_id="FR-4" target_uuid="2p0d-bd9dbaca97bf99bc1b3de839343d0157" text=""
uuid="2p8r-bd9dbaca97bf99bc1b3de839343d0157">
              <points>
                <point x="578.25" y="321.25"/>
                <point x="732.65" y="321.25"/>
                <point x="732.65" y="470.615"/>
              </points>
            </interaction>
            <interaction id="SI-1" name="" source_id="" source_uuid="69c9e302-d06c-38ab-b3b2-b0af7f82931e" target_id="FR-8" target_uuid="2oz5-bd9dbaca97bf99bc1b3de839343d0157" text=""
uuid="2p9w-bd9dbaca97bf99bc1b3de839343d0157">
              <points>
                <point x="121.75" y="321.25"/>
                <point x="302.75" y="321.25"/>
              </points>
            </interaction>
            <interaction id="SI-2" name="" source_id="FR-5" source_uuid="2p02-bd9dbaca97bf99bc1b3de839343d0157" target_id="" target_uuid="367c0a32-e582-3295-aaea-52ce92b3112f" text=""
uuid="2pbc-bd9dbaca97bf99bc1b3de839343d0157">
              <points>
                <point x="1124.25" y="321.25"/>
                <point x="1304.25" y="321.25"/>
              </points>
            </interaction>
          </relation>
          <extensions>
            <extension exportVendor="Sample Company">
              <item_uuid id="ITEM-1" uuid="1i9l-bd9dbaca97bf99bc1b3de839343d0157"/>
              <highestRequirement id="FR-1" uuid="yr-30f7815327a5ef303213dfb2aadbe149"/>
              <DebugInformation CONTAINMENT="0" EXTERNAL_PLANT="0" Element="3" FFI="0" INTERACTION="6" Item="1" OTHER_TECHNOLOGY_LINK="0" REQUIREMENT_ANCHOR="0"
Requirement="5" RequirementGroup="0" RequirementGroupsParing="0" RequirementParing="0" SYSTEM_BOUNDARY_INTERACTION="2" id="" name="Sample1.0"/>
            </extension>
          </extensions>
        </diagram>
      </diagrams>
      <documentation author="Sample User" description="" documentVersion="1.0" timestamp="Thu Apr 19 10:04:12 JST 2018"/>
    </model>
</scdl>
```