

モデルベース設計を用いた ISO26262 対応 システム設計プロセス

System Design Process in Compliance with ISO26262 Using Model-Based Design

相 模 宏 樹*

Hiroki SAGAMI

森 良 司*

Ryoji MORI

広 中 慎 司*

Shinji HIRONAKA

向 井 拓 幸*

Takuyuki MUKAI

要 旨

自動車に搭載される電気、電子システムは、機能安全規格 ISO26262 への準拠が必要である。ISO26262 では、安全性を証明するためのプロセスの実施、および成果物の作成が要求されている。一方で、ISO26262 の基本は V 字開発プロセスであり、開発全体の工数増加の有無は設計段階での構造と要求の定義に左右される。そこで、工数増加を抑制するための手段として、上流工程にあるシステム設計段階に着目し、モデルベース設計を用いた ISO26262 対応のシステム設計プロセスを構築した。その結果、規格に準拠するためのプロセスの実施、および成果物の作成を一連の作業で実施するとともに、モデルによる構造と要求の正確な定義を実現できるプロセスを構築できた。

ABSTRACT

It is necessary for automotive electrical and/or electronic systems to conform to the functional safety standard ISO 26262. ISO 26262 stipulates the implementation of Processes and the creation of Work Products in order to prove safety. However, ISO 26262 is based on a V-model development process, and the definition of structures and requirements at the design stage affects whether or not there is an increase in person-hours across the course of the development as a whole. The research discussed in this paper therefore focused on the system design stage, an upstream process, and formulated a system design process that conforms to ISO 26262 using model-based design as a means of controlling increases in person-hours. As a result, a process was formulated that makes it possible to implement processes conforming to ISO 26262 and create Work Products as a series of procedures, in addition to accurately defining structures and requirements by means of models.

1. はじめに

2011 年 11 月に機能安全規格 ISO26262 (Fig. 1) が正式に発効された⁽¹⁾。この規格は、車両に搭載される電気、電子システムが潜在的に持つリスクを許容できるレベルまで低減して安全を確保するためのものである。機能安全規格に準拠するためには、機能安全規格で定められている開発プロセスの実践、および安全を論理的に証明するための成果物の作成が必要になる。

しかしながら、機能安全規格に準拠した開発を実施する場合、その要件が多いため、規格適用前の開発に比べて工数増加することが懸念される。また、ISO26262 の基本は V 字開発プロセスであり、段階的に設計、および検証を実

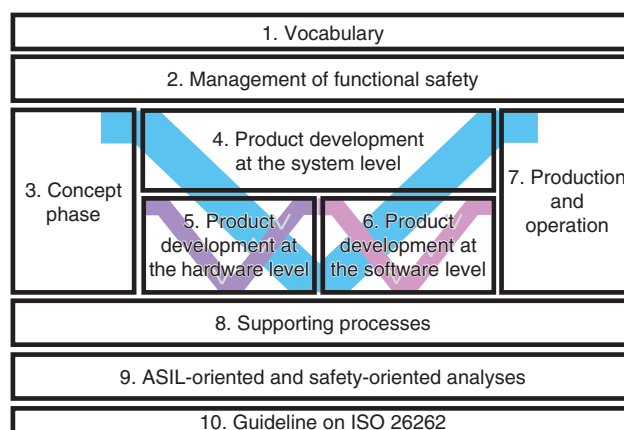


Fig. 1 Overview of ISO 26262

*四輪 R&D センター

施するウォータフォール型開発が基本となる⁽²⁾。一方で、Fig. 2に示すように、V字開発プロセスでは設計段階が上流になるほど検証は後工程になる。したがって、上流工程になるほど設計の精度が手戻りの大きさに強く影響し、最終的には工数増加につながる。以上の背景から、規格準拠のために予測される工数増加をいかに抑制しつつ開発するかが、今後の電気、電子システム開発の課題となる。

そこで、V字開発プロセスの上流工程にあたるシステム設計段階(ISO26262 Part 4に相当する部分)に着目し、前述の課題を解決する手段として、モデルベース設計を利用したISO26262対応システム設計プロセスを構築した。

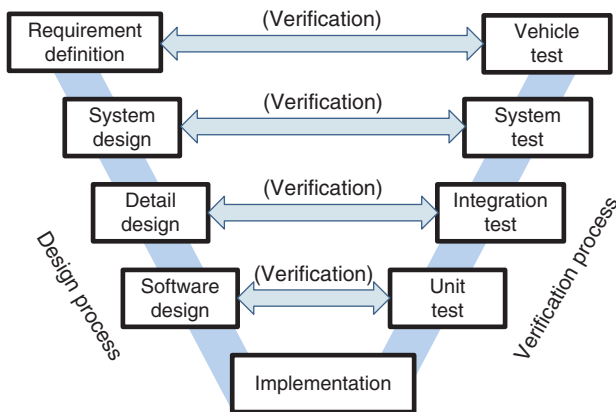


Fig. 2 General V-model development process

2. プロセス構築のねらい

ISO26262 Part 4の概要をFig. 3に示す。Part 4では、システムレベルにおけるV字開発プロセスが求められている。その中で、設計段階として該当する節は図中のPart 4-Clause 6、およびClause 7である。Part 4-Clause 6の目的は技術安全要求の仕様化であり、システムが潜在的に持つリスクを低減して安全を確保するための具体的手段を、システム特性に基づき定義することが求められている。また、Part 4-Clause 7の目的はシステム設計であり、Clause 6で定義した技術安全要求を実現するためのシステム構造を定義することが求められている。

1章で述べたとおり、ISO26262に準拠するためには、規格で求められている開発プロセスの実践と、成果物の作成が必要である。したがって、工数増加を抑制しつつISO26262に準拠するためには、開発プロセスの実践と成果物の作成をいかにして連動させるかが重要になる。

また、システムレベルの設計工程は開発全体における上流工程であるため、ある程度必要な情報を抽象化して表現しなければならない。一方で、システム設計の後工程には、その設計を実現するためのハードウェア(HW)開発、およびソフトウェア(SW)開発を経て、システムレベルのテス

ト検証が控えている。以上から、抽象度が高い中で正確に要求や構造を決定することが、後工程からの手戻り等による工数増大を抑制するためにも重要となる。

以上の観点を踏まえて、プロセス構築のねらいを以下に定めた。

- (1) 設計プロセスを実践した結果、得られた情報を成果物に直接反映できる仕組みにする。
- (2) システムレベルにおいて、機能安全に関連する要求と構造が正確に定義できる仕組みにする。

3. ISO26262 対応のシステム設計プロセス

3.1. プロセス概要

ISO26262で要求されているV字開発プロセスを実践するために、モデルベース設計(Fig. 4)を応用した。モデルベース設計は、V字開発プロセスを効率良く実施するために有効な手段の一つである。ここで、モデルとは、システムの情報が抽象化され、整理された結果を表現するものである。システムを表現するモデルには、システムの構造、システムに求められる要求、およびシステムの振舞が含まれる。なお、モデルの表記方法としては、例えばシステムモデリング言語(SysML)がある⁽³⁾。よって、設計にモデルを利用するモデルベース設計では、設計の早い段階でも曖昧さを除いたシステムの定義が可能となるとともに、早い段階での検証もモデルを用いたシミュレーション等により可能となる。また、モデルを階層化することにより、段階的に要

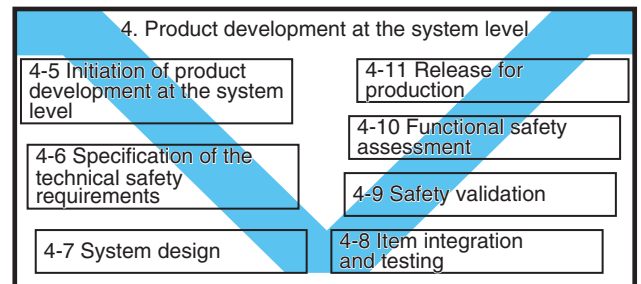


Fig. 3 ISO26262 Part 4

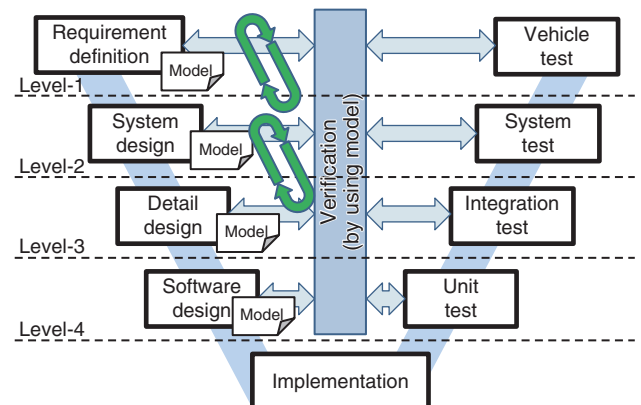


Fig. 4 V-model development process using model-based design

求や仕様を詳細化することができるために、前後工程との関連を明確に表現することができる。以上から、システム定義の曖昧さ、および抜け漏れを除くことで、手戻りの要因を排除し、工数の増大を抑制することが期待できる。

そこで、ISO26262 対応のシステム設計プロセスを構築するにあたり、システムをモデルで表現するために、Fig. 5 のようなシステムの構造と要求の階層を定義した。図中の SG は、危険事象を回避するためのトップレベルの安全目標を、FSR は、SG を達成するためのコンセプトレベルの機能安全要求を示す。また、図中の TSR1 は FSR の実現手段方式として定義する技術安全要求を、TSR2 は TSR1 の実現手段詳細 (HW, SW の識別、エレメントへの割り付け等) として定義する技術安全要求を示す。なお、SysML による階層の表現例は Fig. 6 のようになる。

以下に、ISO26262 に準拠するためのシステム設計レベルでの構造と要求の設計プロセスを示す。

3.2. 技術安全要求の仕様化プロセス

技術安全要求は、コンセプトレベルの安全要求である機能安全要求の具体的実現手段として定義される。技術安全要求を詳細化することで、フェールセーフ等の具体的な仕様が定義される。ISO26262 では、システムの故障により危険事象に至ることが無いよう、技術安全要求を漏れなく定義することが規定されている。

そこで、Fig. 7 に技術安全要求を漏れなく、かつ的確に仕様化するためのプロセスを示す。なお、プロセスを説明するにあたり、前提条件として、Fig. 7(a) のようなコンセプトレベルの構造、および割り当てられた FSR が既に定義されていると仮定する。

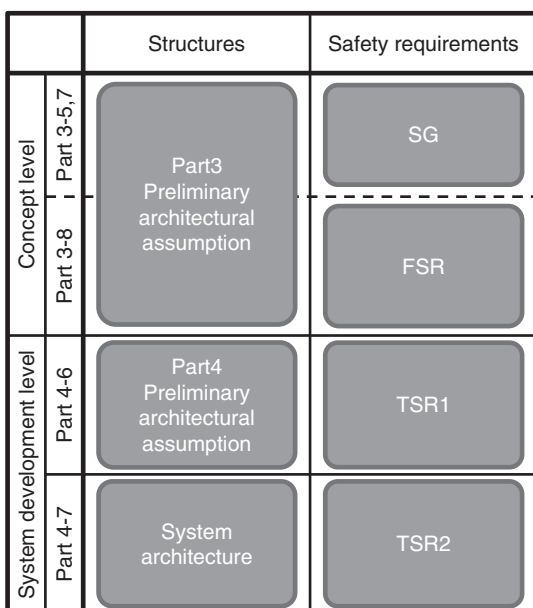


Fig. 5 Hierarchical definition of structures and requirements

3.2.1. 初期アーキテクチャの定義

上位要求 (コンセプトレベルの要求) の分析を実施し、コンセプトを実現するための手段、および構造をシステムレベルの初期アーキテクチャとして定義する [Fig. 7(b)].

3.2.2. 安全分析の実施

システムレベルの初期アーキテクチャに対して安全分析 (例えば、FTA⁽⁴⁾、FMEA⁽⁵⁾、HAZOP⁽⁶⁾ など) を実施し、安全目標を侵害する可能性のある初期アーキテクチャのエレメント、およびその故障モードを抽出する [Fig. 7(c)].

3.2.3. TSR1 の定義

3.2.2. 項で抽出した初期アーキテクチャエレメントの故障モードによる危険事象の発生を防止するための手段、および上位要求である FSR を実現するための手段を TSR1 として定義する [Fig. 7(d)].

3.3. システム設計プロセス

システム設計の目的は、技術安全要求を実現するための構造を決定することである。ISO26262 では、上記目的のために、技術安全要求を実現するエレメントの特定、技術安全要求の HW、または SW の役割の明確化、および HW

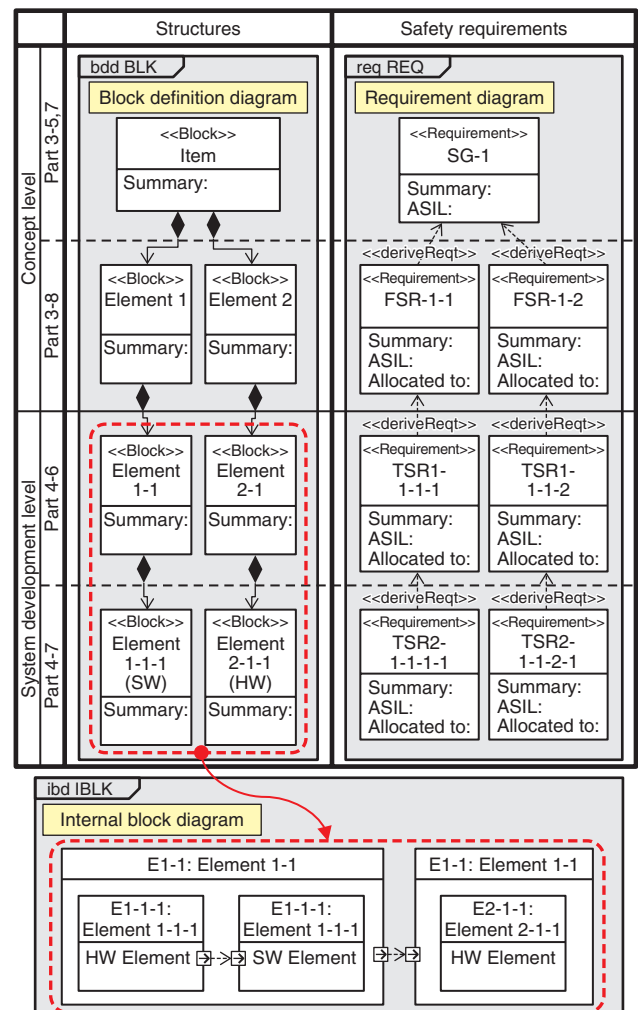


Fig. 6 Structure and requirement definition using SysML

エレメント、SW エレメント間のインタフェースの特定が求められている。

そこで、3.2. 節で定義した TSR1 に対し、上記項目を含めた構造を決定するためのプロセスを示す (Fig. 8)。

3.3.1. システムアーキテクチャの定義

3.2.1. 項で定義したシステム要求と初期アーキテクチャに対し、HW、SW の役割を特定できる段階まで詳細化されたアーキテクチャをシステムアーキテクチャとして定義する [Fig. 8(a)]。

3.3.2. 安全分析の実施

3.3.1. 項で定義したシステムアーキテクチャの安全分析を実施し、安全目標を侵害する可能性のあるシステムアーキテクチャのエレメント、およびその故障モードを抽出する。ただし、3.2.2. 項の安全分析の結果、安全目標を侵害しないエレメントであることが識別されている場合、それから更新されたエレメントに対する安全分析は実施する必要はない [Fig. 8(b)]。

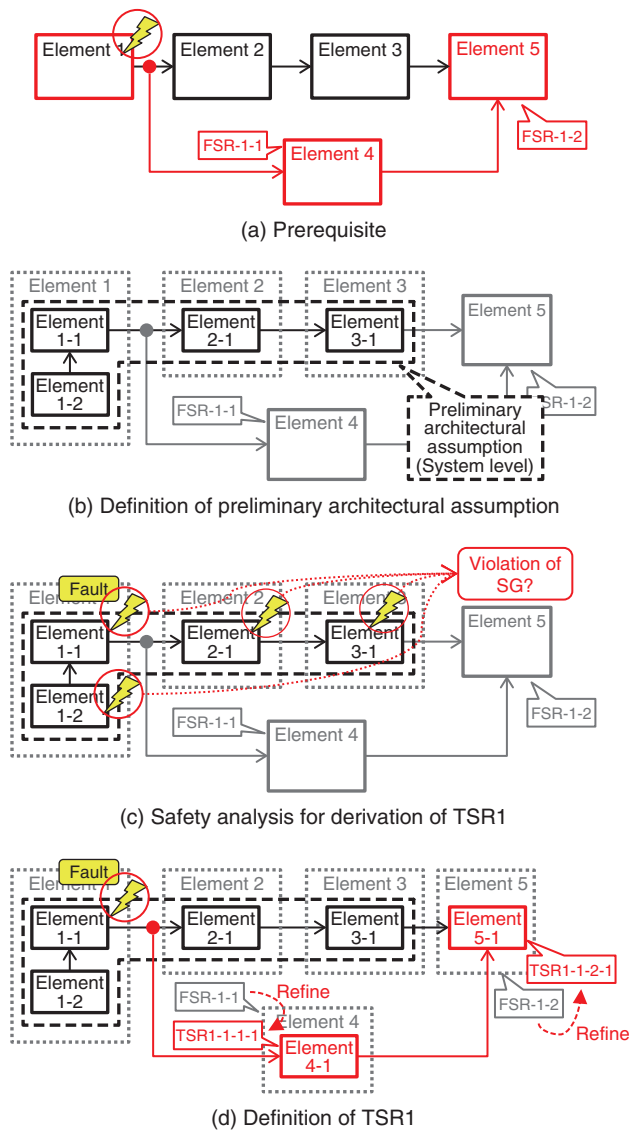


Fig. 7 Procedure for derivation of TSR1

3.3.3. TSR2 の定義

3.3.2. 項で抽出したシステムアーキテクチャエレメントの故障モードに対し、3.2.3. 項で定義した TSR1 を HW、

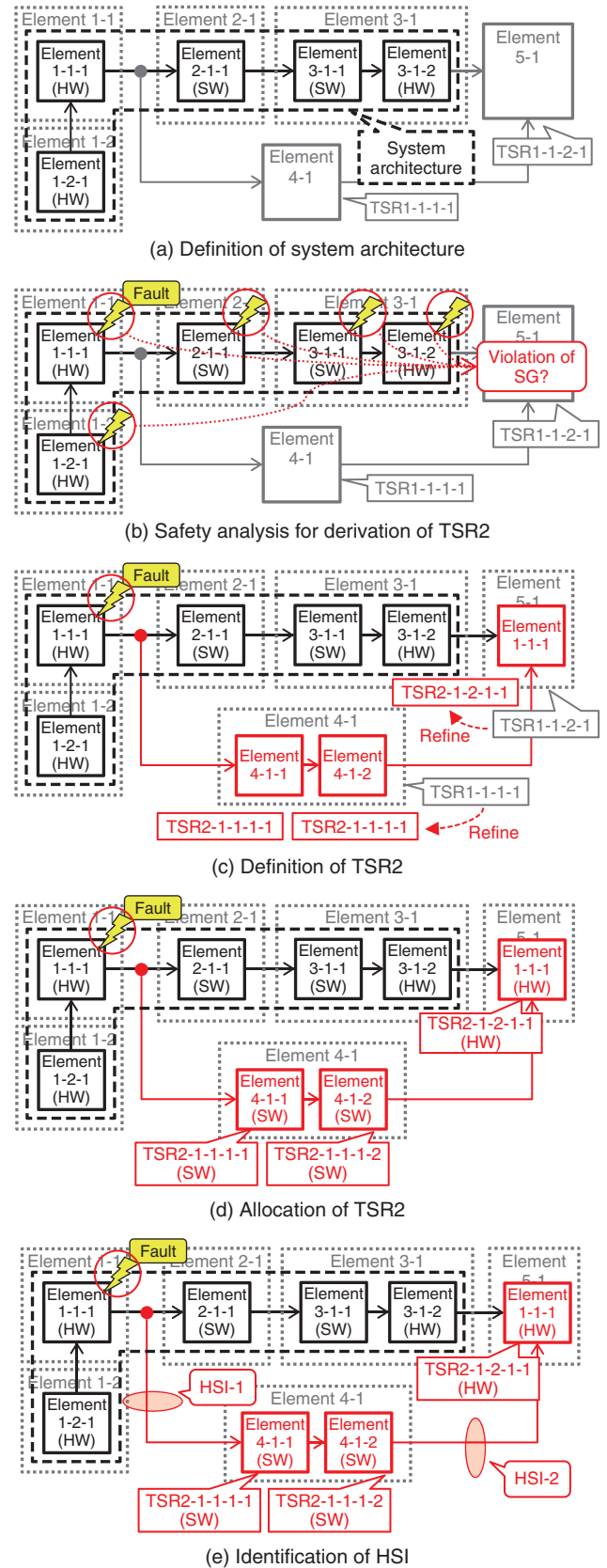


Fig. 8 Procedure for system design
(Including derivation of TSR2)

SW の役割を特定できる段階まで詳細化する。その結果を TSR2 として定義する [Fig. 8(c)]。

3.3.4. TSR2 の割り付けと HW, SW の役割の決定

3.3.3. 項で定義した TSR2 に対し、システムアーキテクチャエレメントへの割り付けをおこなう。このとき、TSR2 を実現するために、必要があればシステムアーキテクチャに新たなエレメントを追加する。また、HW, SW のどちらの役割とするかも併せて定義する [Fig. 8(d)]。

3.3.5. ハードウェアソフトウェアインタフェースの定義

3.3.4. 項で更新されたシステムアーキテクチャから、TSR2 に関わる HW エレメント, SW エレメントの特定を実施する。その上で、HW エレメントと SW エレメントが直接接続されている部分のインタフェース仕様をハードウェアソフトウェアインタフェース (HSI) として定義する [Fig. 8(e)]。

4. 成果物の作成

4.1. 成果物の概要

ISO26262 では、Part 4-Clause 6 の成果物として仕様化された技術安全要求, Part 4-Clause 7 の成果物として技術安全コンセプトとシステム設計仕様, および HSI 仕様が求められている。これらはいずれも規格準拠性を論理的に証明するために必要なドキュメントであり、そのためには ISO26262 で求められているプロセスを踏まえた上で作成する必要がある。

一方、3 章では ISO26262 に準拠するためのプロセスを構築した。したがって、プロセスの実施の過程で取得する情報を成果物に反映すれば、成果物による規格準拠性の論理的な証明は十分可能である。

そこで、本節では 3 章のプロセスを踏まえた成果物の作成手順を示す。

4.1.1. 仕様化された技術安全要求

3.2.3. 項、および 3.3.3. 項により作成した TSR1, および TSR2 を、技術安全要求の成果物としてまとめる。この際に、階層設計により得られたトレーサビリティも含めて成果物に反映することで、追加の作業をおこなわなくても規格準拠性を証明できるドキュメントにすることができる。例えば、SysML の要求図を用いて作成すると、要求図そのものを成果物とすることができる。また、別の作成例として、Table 1 のように、FSR から TSR2 までを表にまとめて記載してもよい。

4.1.2. 技術安全コンセプトとシステム設計仕様

TSR2 のエレメントへの割り付け結果、および HW, SW の役割の選択結果を、技術安全コンセプトの成果物としてまとめる。例えば、Table 2 のように、TSR2 のエレメントへの割り当て先、および HW, SW の役割を明確に記載すれば良い。

Table 1 Example of work product of TSR

SG	FSR	TSR1	TSR2
SG-1	FSR-1-1	TSR1-1-1-1	TSR2-1-1-1-1
			TSR2-1-1-1-2
	FSR-1-2	TSR1-1-1-2	TSR2-1-1-2-1
			TSR2-1-1-2-2
...	...	...	...

Table 2 Example of work product of technical safety concept

TSR2	Allocated system element	HW or SW
TSR2-1-1-1-1	Element A	HW
TSR2-1-1-1-2	Element B	SW
TSR2-1-1-2-1	Element C	HW
...	...	...

Table 3 Example of work products of HSI

Description	HW			SW			Interface type	Related TSR2
	Symbol	Range	Unit	Symbol	Range	Type		
XXX1	xxx1h	0 to 5	V	xxx1s	0 to 255	Byte	A/D	TSR2-1...
...								

4.1.3. HSI 仕様

3.3.5. 項により特定した HW エレメントと SW エレメントの間のインタフェースを、HSI 仕様の成果物としてまとめる。例えば、Table 3 のようなドキュメントが HSI 仕様のドキュメントとなる。

5. 電動パワーステアリング制御システムへの適用事例

電動パワーステアリング (Electric Power Steering: EPS) システムはドライバの操舵負荷を軽減することを目的として、ドライバの操舵トルクに応じてアシスト量を決定し、出力するシステムである。

ここでは、3 章、および 4 章で構築したプロセスについて、EPS の代表例を用いてその適用事例を説明する。

5.1. SG と FSR

EPS システムの機能安全対応のシステム設計例を説明するにあたり、前提条件として必要な SG, FSR を含むコンセプト定義について、EPS システムの代表例を定義し

た. SG は車両が勝手に曲がる出力をさせないと定義した. また, FSR は操舵トルク検知機能の故障を検知したら出力を遮断すると定義した. 以上を踏まえたコンセプトのアーキテクチャを Fig. 9 のように定義した.

5.2. TSR1 の導出

EPS システムのシステムレベルの初期アーキテクチャを Fig. 10 に示す. トルクセンサにより操舵トルクを検知し, トルクセンサ値からアシスト量を決定する. また, アシスト量を出力するためにモータ制御を実施して, 実際のアシスト量をモータから出力する.

この初期アーキテクチャをもとに 5.1. 節で設定されている SG を侵害する可能性のある故障の要因を分析する. その結果, トルクセンサの出力が故障により規定範囲から外れた値が出力される. その場合, 意図しないアシスト量が算出, および出力され, SG を侵害して危険事象が発生する可能性があることが抽出された. この結果から, TSR1 は, トルクセンサ出力が規定範囲から一定時間以上外れたらリレー機能によってモータ駆動の出力を遮断すると定義した (Fig. 11).

5.3. TSR2 の導出

EPS システムのシステムアーキテクチャを Fig. 12 に示す. これは, 5.2. 節で作成した初期アーキテクチャの各エレメントに対し, HW, および SW の役割が識別できるま

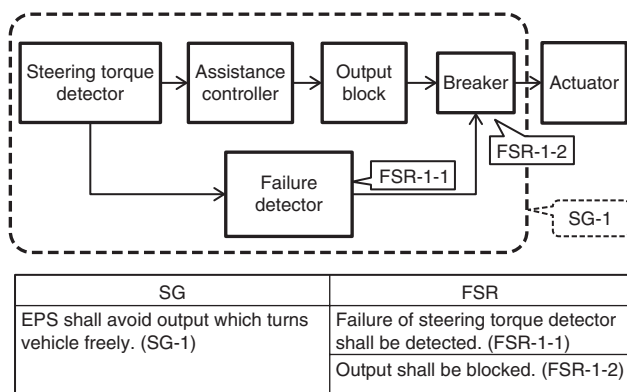


Fig. 9 Example of SG, FSR, and architecture at concept level

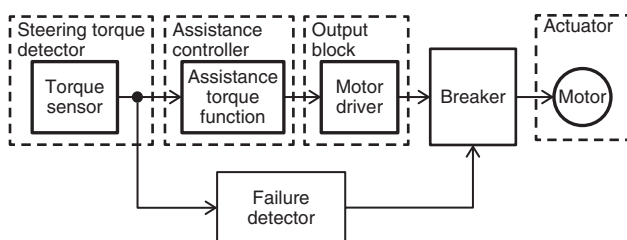


Fig. 10 Example of preliminary architectural assumption at system level

で詳細化された結果である.

ここで, 再度システムアーキテクチャレベルで要因の分析を実施した. その結果, HW エLEMENTのトルク信号入力部に故障が発生した場合に, 5.2. 節で抽出した事象が発生することを抽出した. この結果から, TSR2 は, トルク信号入力部の出力が規定範囲から一定時間以上外れたら, リレーによってモータ駆動部の出力を遮断すると定義した.

さらに, TSR2 を分割して, システムアーキテクチャエレメントへの割り当てを実施した. SW エLEMENTのトルクセンサ故障検知機能に割り当てる TSR2 は, トルク信号入力部の出力範囲が規定範囲から一定時間外れることを監視すると定義した. また, SW エLEMENTのリレー制御機能に割り当てる TSR2 はリレーオフの指示を出力すると定義した. さらに, HW エLEMENTのリレー部に割り当てる TSR2 はリレーオフによりモータ駆動部の出力を遮断すると定義した. これらの結果をもとに, Fig. 13 のようにシステムアーキテクチャを更新した.

5.4. HSI の導出

5.3. 節にて更新したシステムアーキテクチャより, トルク信号入力部とトルクセンサ故障検知機能の間, およびリレー制御機能とリレー部の間のインタフェースが

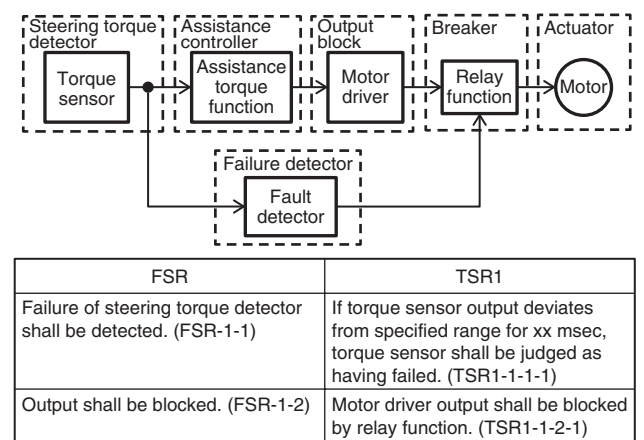


Fig. 11 Example of TSR1 and refined preliminary architectural assumption at system level

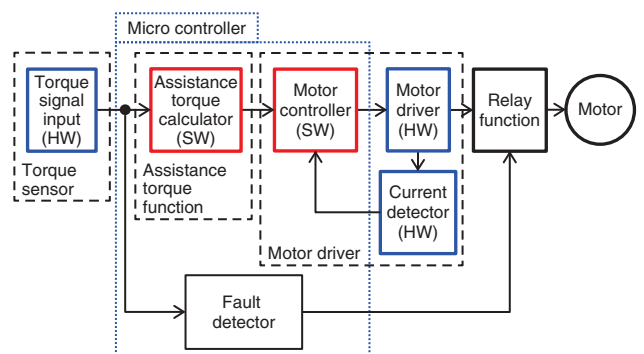


Fig. 12 Example of system architecture

TSR2 に関連するため、HSI 定義の対象部分として抽出した (Fig. 14)。

5.5. 開発プロセス適用による効果

構築した開発プロセスを EPS システムの設計に適用した結果、後工程からの手戻りの原因となる安全目標の侵害の潜在的な要因を排除することができた。例えば、システムの構造と要求が階層的に明確に定義されるため、要因を抽出し、排除するためのレビューやシミュレーション等による設計段階での検証精度が向上した。また、後工程の

HW 開発、および SW 開発のそれぞれの対象がシステムアーキテクチャにより明確になるため、情報伝達の抜け漏れによる要因を排除することができた。

6. まとめ

構築した開発プロセスにより

- (1) 設計の過程で取得した情報を成果物に直接反映することで、成果物の作成にかかる工数を抑制することができた。
- (2) システムの構造、および安全要求を階層的に定義することで、手戻りの原因となるシステムの安全目標の侵害要因を削減した結果、工数増加を抑制することができた。

参考文献

- (1) ISO26262: Road Vehicles – Functional Safety(2011)
- (2) 村上真, 野村和範, 久木隆, 笠原康彦: パワープラント分野におけるシステムモデリング技術, Honda R&D Technical Review, Vol.22, No.2, p.99-105
- (3) <http://www.omgsysml.org/>
- (4) IEC61025: Fault tree analysis (FTA)(2006)
- (5) IEC60812: Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA)(2006)
- (6) IEC61882: Hazard and operability studies (HAZOP studies) – Application guide(2001)

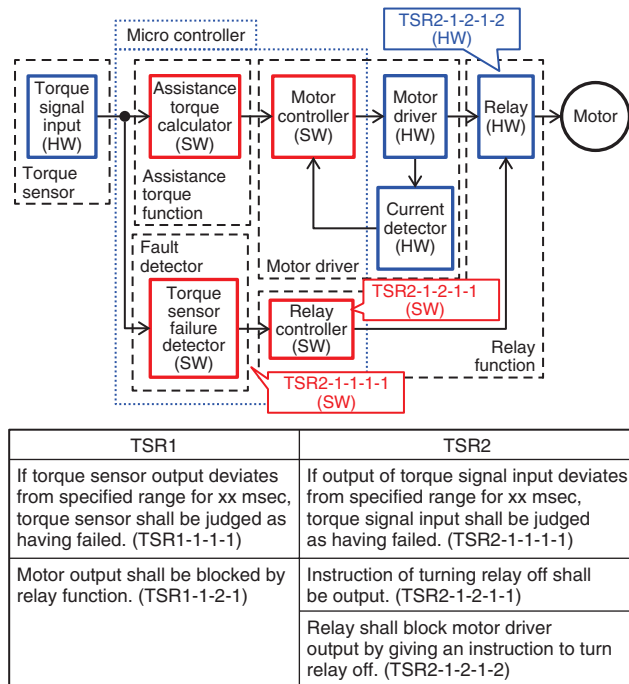


Fig. 13 Example of TSR2 and refined system architecture

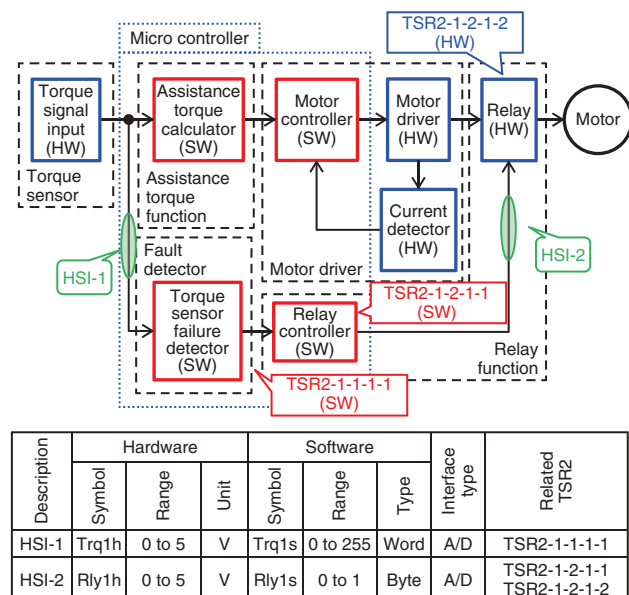


Fig. 14 Example of HSI

著者



相模 宏樹



広中 慎司



森 良司



向井 拓幸