システムズエンジニアリング手法の適用による機能安全活動の高度化および効率化

Sophistication and Efficiency of Functional Safety Activity by Applying Systems Engineering

金井信人　N. KANAI

In Functional Safety Activity based on ISO 26262, we have achieved sophistication and efficiency by applying Systems Engineering Method.
(1) Sophistication: Design information, which is becoming more and more complex as the number of functions increases, can be properly managed and visualized to improve explanatory capability.
(2) Efficiency: Document management can be centralized by using SysML tools. As a result, document creation time and verification time can be reduced. (= Reduction of development time)

Key Words: ISO 26262, functional safety, Systems Engineering, MBSE, SysML

1．はじめに

近年，電動パワーステアリング（Electric Power Steering：EPS）において，先進運転支援システム（Advanced Driver-Assistance Systems：ADAS）や自動運転（Autonomous Driving：AD）システム，ステアバイワイヤ（Steer By Wire：SBW）システムといったEPSに求められる機能の増加に伴い，設計の複雑さ，検証の難しさが従来のEPSに比べ指数関数的に増加しつつある．ISO26262に基づく機能安全活動においては安全設計および検証だけでなく，設計プロセス（設計コンセプトの妥当性，要件および検証結果のトレーサビリティ）も要求される．前述のような新規機能の開発においては，より高度な設計説明責任が求められ，かつコスト低減や開発期間短縮のため効率的な開発が求められている．一方，過去実績の積み重ねにて設計技術がある程度成熟した開発については，より設計プロセスに対する説明要求の比重が高まってきている．機能安全活動ではハードウェア，ソフトウェア問わず総合的な知識が必要とされるものの，過去実績から流用される機能については完成された設計仕様のみ転用される場合が多い．このため開発当時の知見・経験（機能または信号の目的・用途）が見えづらく，経験の乏しい担当者が活動を実施する場合，新規開発へ適用する際の影響分析において判断に迷うケースが発生し，開発効率を下げる要因となっている．

こうした将来的な開発要求や設計情報の円滑な共有を実現するためには，システムズエンジニアリングといわれる手法を取り入れる必要がある．本手法では，目的（要求）に合わせて機能をデザインし，かつ機能を実現する物理構成をデザインし，その構成要素間の関係性を明らかにして設計情報を構築していく[1)]．

そこで本報では，機能安全活動における設計情報の整理を実施することで「高度化」，さらにこれらの設計情報を管理するためSysML（Systems Modeling Language）ツールを活用したMBSE（Model Based Systems Engineering）として適用することで「効率化」を図った．

2．システムズエンジニアリング手法の活用と機能安全活動の「高度化」

本章では，具体的にどのようなシステムズエンジニアリング手法を用いて機能安全活動の「高度化」を図っていくか，「機能アーキテクチャーの構築」，「場合分け」，「物理アーキテクチャーの構築」といった三つの手法を用いて説明していく．

2.1　機能アーキテクチャーの構築

機能安全活動にて適用してきたEPSアーキテクチャーは，社内だけでなく顧客や仕入先とも共有するドキュメントであり，あえてより設計仕様に近い表現として機

能を実現する物理構成も混在させて定義していた．システムズエンジニアリング手法による設計情報整理のため，“システムの意図する振る舞いの連鎖”のみを示す機能表現のアーキテクチャー（EPS 機能アーキテクチャー）として構築し，EPS の各機能が果たすべき目的を定性的に表現した．**図 1** に EPS 機能アーキテクチャーの一例を示す．トルクセンサから入力される信号を EPS 内部の演算値へ変換するトルクセンサインターフェースのブロックをイメージし，「回転の力の程度を定量化する」機能と定義した．

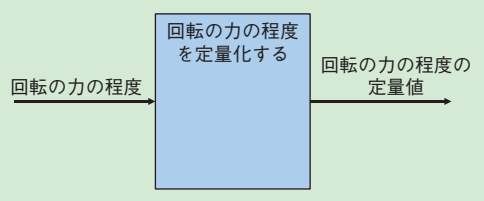

図 1　EPS 機能アーキテクチャーの一例
An example of EPS functional architecture

2.2　機能アーキテクチャーの場合分け

機能表現で作成したアーキテクチャーを EPS の動作モードごとに場合分けを実施した．結果として同じ物理構成を用いて実現する機能であっても EPS の動作モードが異なれば求められる要求も異なる場合が生じるためである．**図 2** は EPS の動作モードを“運転者の操舵をアシストする場合（Driver 操舵モード）”と ADAS システムのように“車両が判断して EPS を操舵する場合（Vehicle 操舵モード）”に場合分けし，それぞれにおける機能を定義したものである．Driver 操舵モードの場合では，「基本の支援力を演算するための回転の力の程度を定量化する（アシスト量を算出するための入力値）」とするのに対し，Vehicle 操舵モードの場合では，「運転者の操舵を判断するための回転の力の程度を定量化する（操舵判定のための入力値）」と定義される．同じ「回転の力の程度を定量化する」機能であっても，場合によって用途が異なっていることが分かる．

動作モードから機能の場合分けを実施することで EPS の境界線を明示することができ，機能要求をより明確（どんな機能だけでなく，どんな場合に必要か）に表現することができた．

2.3　物理アーキテクチャーの構築

機能アーキテクチャー同様の手順で，“システムの機能的な連鎖”のみを示す物理表現のアーキテクチャー（EPS 物理アーキテクチャー）を構築し，EPS の各機能が実現すべき設計情報を定量的に表現した．**図 3** では**図 2** の機能を実現手段として定義した例である．本例では場合分けされた Driver 操舵モード，Vehicle 操舵モ

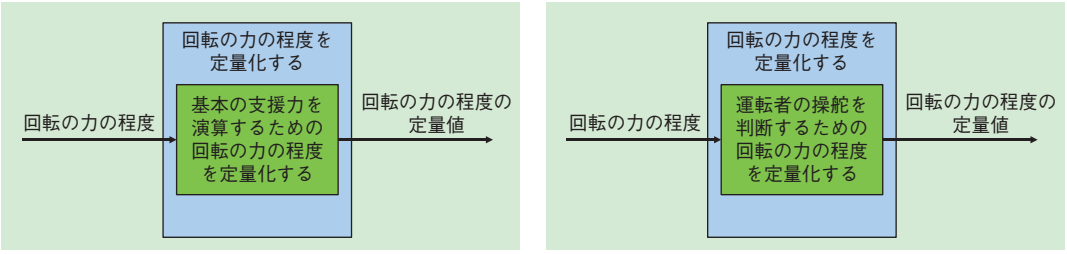

図 2　動作モードによる機能定義の違い（左：Driver 操舵モード，右：Vehicle 操舵モード）
Difference in functional definition depending on Operation mode (Left: Steer by Driver Mode, Right: Steer by Vehicle Mode)

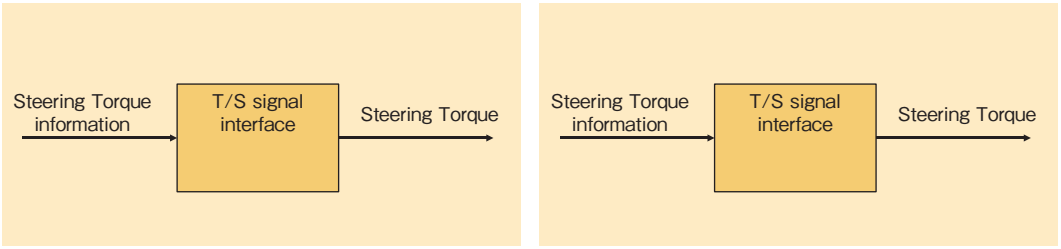

図 3　動作モードによる物理定義（左：Driver 操舵モード，右：Vehicle 操舵モード）
Physical definition depending on Operation mode (Left: Steer by Driver Mode, Right: Steer by Vehicle Mode)

ードにおける「回転の力の程度を定量化する」機能の実現手段をアーキテクチャーにて定義しているが，用途が異なる機能であるものの，いずれも「T/S signal interface」にて共通の物理で実現すること（設計仕様）を表現している．

本活動においては，既開発 EPS を題材とし，構築済みの物理アーキテクチャーに対して機能アーキテクチャーを再定義する流れで進めたが，本来のシステムズエンジニアリングとしては目的となる機能アーキテクチャーを作成し，それを実現するための物理アーキテクチャーを構築していく必要があると考えられる．

2.4　システムズエンジニアリング手法を用いた「高度化」の利点との課題

EPS アーキテクチャーを機能と物理に分けて再構築することで要求される機能（目的）とそれに対応する物理（実現手段）を明確にすることができた．流用開発においては共有されるべき設計情報が見えるようになり，新規開発においては機能（目的）を明確にすることで，選択した物理（実現手段）が適切であるかを相互確認しながら開発を進めることができると考えられる．変更点が発生した場合にも，その変更が物理要因（実現手段の変更）であるのか，機能要因（要求される機能の変更）であるのか，また影響範囲はどこまでか（どの動作モードに対して影響が発生するか）といった情報の抽出をアーキテクチャー上で網羅的に表現することができ，影響分析の抜け漏れが発生しにくい設計情報を構築することができた．ただし，設計情報が整理された一方で管理すべき情報量は増加した．今後も要求される機能の増加や EPS に求められる動作モードの増加に伴い，設計情報はより複雑に，またその検証の難しさも増加していくものと想定される．これらの設計情報を手作業で関連付けし管理していくことは困難であるため，SysML ツールを活用して一つの情報構造体へとモデルベース化し，MBSE として開発プロセスへ適用し「効率化」を図った．

3．MBSE 化による「効率化」

本章では、SysML ツールを活用して，整理された設計情報を一つの情報構造体としてツール内で MBSE 化し、「効率化」を達成していく流れを説明する．

3.1　SysML ツール内における情報構造体の構築

機能／物理それぞれの設計情報を，SysML ツール内にて情報構造体として構築した．

①対象システムのコンテキスト（対象システムを取り巻く全ての環境やシステム）を階層構造として定義

②各アーキテクチャーをシステム系統図として作成（各ブロック，信号の設計情報をパラメータとして定義）

これにより機能／物理表現の設計情報として，EPS における外部との関係性（①），内部（②）の構成を表現することができた．

③機能／物理表現の関係をアロケート

機能／物理表現の情報を関連付けることで関係性が明確になり，ツール内で一つの情報構造体として構築することができた．これによりどの機能がどの物理（実現手段）にて構成されているかをツール内で把握することができるため，機能側の変更あるいは物理側の変更のいずれかが発生した場合においても，影響が及ぶ範囲を抜け漏れなく管理することができるようになった．**図４**は SysML ツール内で構築された情報構造のイメージを示す．

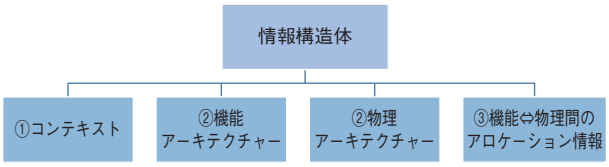

図４　SysML ツールにおける情報構造体のイメージ
Example information structure in the SysML tool

3.2　設計情報の可視化，アウトプット情報の抽出

実開発プロセスに適用できるよう，ツール機能を活用し情報構造体を正式ドキュメントの形式で出力できる仕組みを構築した．従来のプロセスでは各ドキュメントをそれぞれの形式で手作業にて作成・修正していたが集約された情報構造体から抽出することでドキュメント作成時間の短縮に限らず，ドキュメントの修正漏れや設計情報の不整合をツールで確認できるようになった．またさらに仕組みを構築することで影響分析時における対象箇所のみの抽出など，開発プロセスの各場面で求められる情報を任意の形式で可視化して出力することが可能なため，設計レビュー時など，設計情報の説明性の向上にも寄与することができた．**図５**に SysML ツールを適用した場合の活動プロセスの変化を示す．

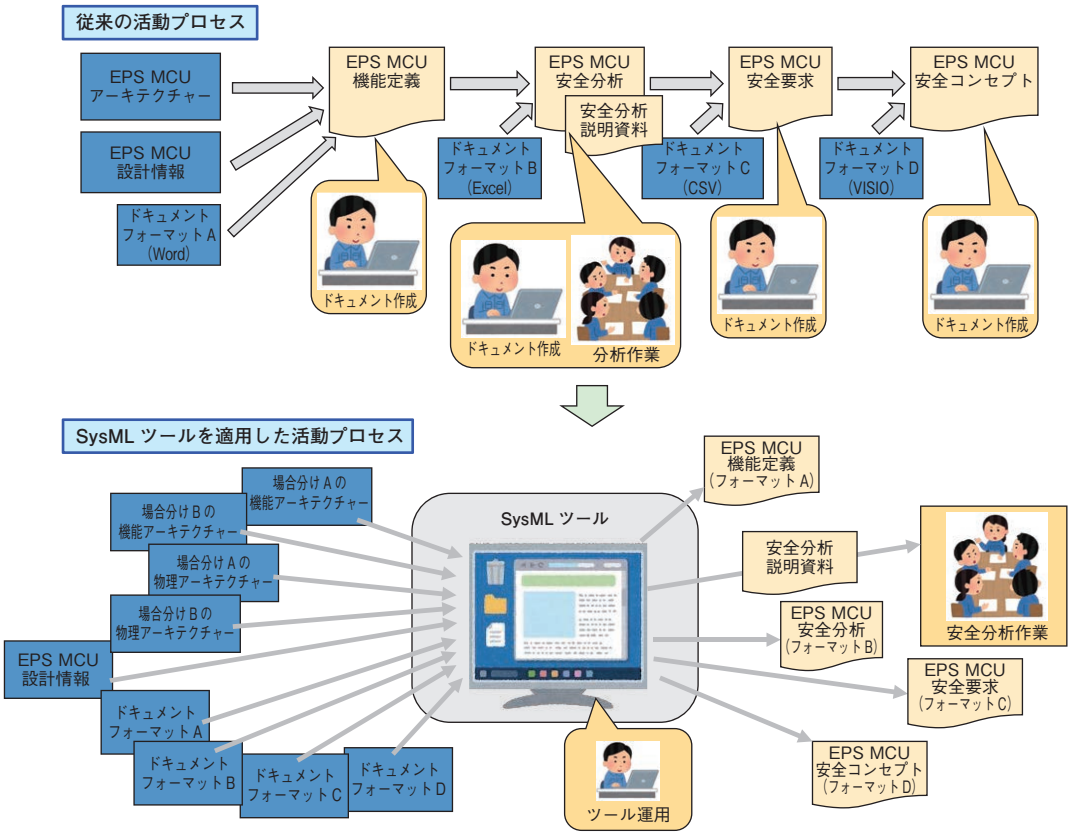

図5　SysML ツールを適用した場合の活動プロセスの変化
Changes in the activity process when the SysML tool is applied

4. おわりに

システムズエンジニアリング手法を用いて，EPS に求められる機能を振る舞いから定義し，設計情報と関連付けることにより「高度化」を行った．さらに，SysML ツールを活用することで増えた情報の管理および活用するための作業（影響範囲の抽出，説明資料，ドキュメント作成）といった情報の可視化を機械に実施させた．これにより人は考える作業に注力できるようになり，「効率化」を図ることができた．

本活動では機能安全活動における「機能定義」を一例として述べ，システムズエンジニアリング手法の有効性を示した．「安全分析」，「安全要求」，「安全コンセプト」といった後工程の活動に対しても，この考え方は適用できる．たとえば，ツール内で設計情報を関連付けしながら構築することで，変更が発生した場合に“上位→下位”，“下位→上位”の観点で影響分析を実施し，情報を管理することが可能となる．さらにこれらは機能安全活動に限らず設計活動全般においても適用できる考え方である．

ただし単純に SysML ツールを使ってドキュメントを出力する．ということではなく，設計情報をきちんと整理してこそ設計業務に対する良さ（説明性の向上）が出てくるので，真に活用していくにはシステムズエンジニアリング知識の習得，また SysML ツールを運用するリソースなど組織レベルでの底上げが必要になると考えられる．

参考文献

1）S. Friedenthal, A. Moore, R. Steiner：A Practical Guide to SysML Third edition, The MK/OMG Press(2014).

筆　者

金井信人*
N. KANAI

*　自動車事業本部　技術企画部