EAST-ADL 拡張によるセーフティクリティカルな機能、および ISO 26262 作業成果物のモデルベース開発 [1]

Bülent Sari[1], Hans-Christian Reuss[2]

[1] *Electronics Powertrain Technology, ZF Friedrichshafen AG, Friedrichshafen, Germany*

[2] *Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren Stuttgart (FKFS), Stuttgart, Germany*

キーワード：モデル駆動型アプローチ、セーフティ、ISO26262、EAST-ADL

概要

自動車に組み込まれる安全関連の E/E システムのレベルは高まり続けていて、安全性がますます重要になってきています。パワートレインの電動化によって車両システムの機能性が向上し、自動運転によって将来さらに向上することで、システム、ソフトウェア、そして安全アーキテクチャの設計が複雑になります。ISO 26262 は、複雑さを軽減し、さまざまな安全活動のトレーサビリティを承認することを目的としています。本論文では、アーキテクチャ記述言語 EAST-ADL（Electronics Architecture and Software Technology - Architecture Description Language）を用い、関連する規格 ISO 26262 に沿って、システム、ソフトウェア、そして安全アーキテクチャのモデルベース開発に関するアプローチを示します。特に、ハザード分析やリスクアセスメント、機能的・技術的な安全コンセプトの開発、そして安全分析の実施など、主な安全関連の活動がどのようにモデルベースで実行でき、それらの活動をシステムやソフトウェア開発とどのように関連付けられるかについて簡単に論じます。現時点での最新アプローチも示し、提案するアプローチと比較します。

1. はじめに

本論文は、鉄道・自動車・航空・船舶の電気技術に関する国際会議（ESARS-ITEC 2016）で発表された研究[1]を発展させたものです。今日、高級車両には複雑なアルゴリズムを計算できる 100 個もの処理ユニット（ECU）が搭載されています。高級車の組込みソフトウェアには 1 億行にもなるソースコードが含まれています。対照的に新しい「ボーイング 787 ドリームライナー」では、オンボードシステム全体で約 650 万行のコードが必要です[2]。

この比較は、今日の車両におけるソフトウェアがどれだけ複雑であるかを示しています。ソフトウェアの複雑性と規模は拡大し続けます[2]。ソフトウェアの規模が大きくなる主な理由は、自動車の電化と、すでに利用可能となっている高度な運転支援システムです。車両の総生産コストに対する電気・電子部品コストの比率は、2020 年に 35%、2030 年には 50%まで上昇し得ると考えられています[3]。電化の増加にともなってセーフティクリティカルなシステムの割合も増加しています。「運転中の駆動車軸の意図しないブロッキング」や「パワーステアリングが間違った方向に作用する」、「運転中のエアバッグの誤動作」などの異常は、生命にかかわる傷害につながる可能性のある例です[4]。

---

[1] 本論文は、Model-based Development of Safety-critical Functions and ISO 26262 Work Products using modified EAST-ADL (*Advances in Science, Technology and Engineering Systems Journal Vol. 2, No. 3, 1252-1259 (2017)* https://astesj.com/v02/i03/p158/ の日本語訳です)

既存でも新規でも車両システムの機能を向上させると、E/E システムの割合が大きくなり、システムやソフトウェア、そして安全アーキテクチャの複雑性が高まります。異なる分野のエンジニアが関与する必要があります。システムアーキテクチャとソフトウェアアーキテクチャは、開発のどの段階においてもさまざまな要件を満たす必要があります。この時点で、安全性および非安全性関連機能の増加について、業界がソフトウェアの品質を犠牲にすることなく、どのように対応しているかという疑問が生じます。

エンジニアがこれらの課題に対処するアプローチの一つは、モデル駆動型のシステム・ソフトウェア・安全開発を使用するものです。本研究では、セーフティクリティカルな機能のモデルベース開発と ISO 作業成果物のモデルベース開発のためのアプローチを示します。このアプローチの基礎として EAST-ADL を使用します。ただし、ISO 26262 ではセーフティクリティカルなシステムの開発による一貫性とトレーサビリティが必要になるため、システムアーキテクチャと安全アーキテクチャが統合されるように EAST-ADL を修正する必要があります。しかし、現在の EAST-ADL の仕様ではシステムアーキテクチャと安全アーキテクチャが異なるモデルで分離されていてシステムアーキテクチャモデルと安全モデルの間に直接的な関係がないため、これらの要件を実装するには不十分です。システムアーキテクチャは EAST-ADL の抽象化レベル内で開発され、安全モデルは信頼性モデル内で実現されます。ハザードとリスク、安全目標、機能的・技術的な安全要件、そして安全機能の間の依存関係を示すことは大きな課題であり、また ISO 26262（Part 4 - 図 2、Part 10 - 図 8 および図 9）の要件でもあります。安全性評価によって開発者は、どの安全目標がどの安全機能によって実装されているか示すことができ、安全機能と安全目標が同じ ASIL（Automotive Safety Integrity Level：安全性要求レベル）を持っていることを証明する必要があります。そのため、一貫性チェックや安全機能のトレーサビリティを提供し、安全コンセプトなどの ISO26262 作業成果物を自動生成するために、EAST-ADL を修正してこれらのモデルを統合することが不可欠でした。この拡張機能により、安全目標と安全機能の関係を示すことができ、どの機能がどの安全目標に対して実装されているかの証明が容易になります。したがって、安全活動の完全性を証明する方が簡単です。

この論文の主な寄与の二つ目は、プロジェクトの開発段階初期における安全要件のシミュレーション内で、ISO 26262-4:2018 の 7.4.8 節で要求される技術的安全コンセプトと機能安全コンセプトを検証し、妥当性を確認することです。したがって、安全コンセプトを早期に改善し、開発初期段階で体系的なエラーを見つけることも可能です。完全なシステムをモデリングすることは大規模なプロジェクトであり、ドメイン固有の知識が必要です。さらにアーキテクチャはいくつかのツールを使用して記述されているため、さまざまなツールに関する知識が必要です。現行のアプローチは、ADL[6, 7]を使用することで、ドメイン固有言語に基づいて開発されるアプローチ内で置き換えることができます。これにより、拡張と改善の範囲内で、システムモデル内ですべての情報が生成され、完全なシステム、安全性、そしてソフトウェアアーキテクチャを記述し、システムと安全性アーキテクチャ間の一貫性とトレーサビリティを提供し、そして開発初期段階で安全コンセプトを検証できるようになります。

2 章で、この目的のために EAST-ADL をどのように修正するかを示します。

2. アプローチの説明

開発したアプローチで、システム、ソフトウェア、および安全性の開発がどのように組み合わされて、一貫性のあるアーキテクチャによってシステムの複雑性が軽減されるかを図 1 に示します。

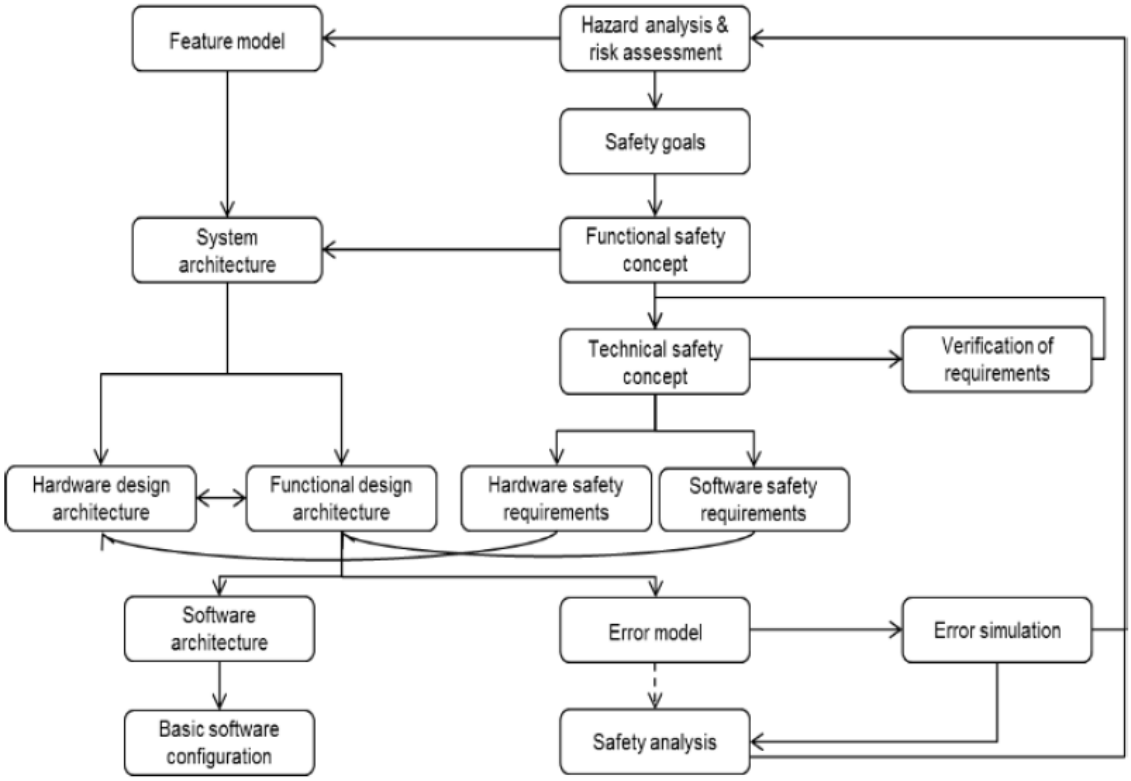

図1　システム、安全性、ソフトウェア開発のためのモデル駆動型アプローチ

開発した手法は、図 2 に示す 5 つの主要なパートで構成されています。

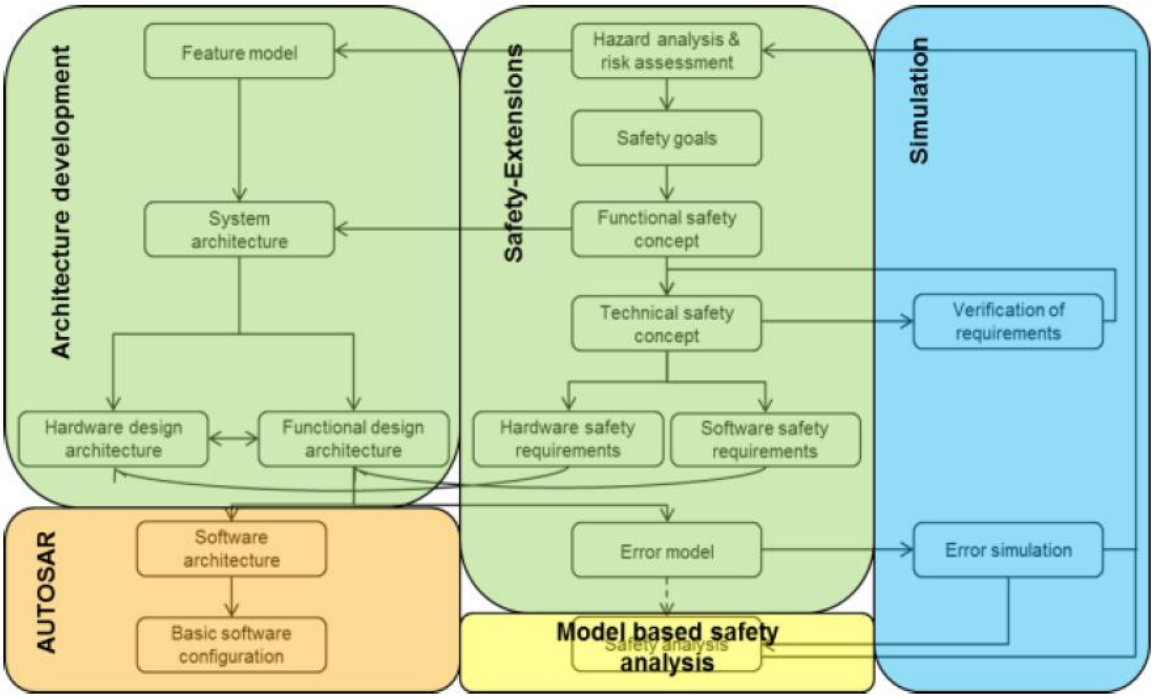

図2　アプローチの主要部分

最初のパートは、アーキテクチャ開発[7]です。このパートでは、フィーチャモデル、システムアーキテクチャ、機能設計アーキテクチャ（functional design architecture：FDA）、そしてハードウェア設計アーキテクチャ（hardware design architecture：HDA）の作成について説明します。FDA モデルの機能を HDA モデルの対応するハードウェア要素に割り当てることもできます。このパートは、システムアーキテクチャと安全アーキテクチャを結合するために追加された安全属性内で拡張されており、トレーサビリティと一貫性を証明するためにシステムアーキテクチャと安全アーキテクチャの関係を実現することができます。詳細は 2.1 節で説明します。

二つ目のパートは安全拡張[8, 9]です。この部分では、アーキテクチャ開発の拡張としてモデルベースでの ISO 26262 作業成果物の作成を扱います。まず、ハザード分析とリスクアセスメントを実施します。次に、ハザード分析とリスクアセスメントから安全目標を導き出します。安全目標を達成するために、機能安全コンセプトと技術安全コンセプトを以下のステップで作成します。このパートは、安全モデル、システムモデル、そして要件モデル間の関係を実現するために、追加の安全属性で拡張されています。安全モデルは、ハザード分析とリスクアセスメント（Hazard Analysis and Risk Assessment：HARA）や機能的・技術的な安全性コンセプトなどの安全作業成果物をモデルベースで実現します。この拡張により、開発されたスクリプトを使用して、モデルからこれらの安全作業成果物を自動的に生成できるようになりました。詳細については 2.2 節で説明します。

フィーチャモデルには、システムの非セーフティクリティカルとセーフティクリティカル両方の属性を含めることができます。ハザード分析とリスクアセスメントの後、システムのセーフティクリティカルな側面はフィーチャモデルの一部であるとみなされます。

システムアーキテクチャでは、システムの安全目標と対応する安全機能が考慮されます。機能およびハードウェア設計アーキテクチャでは、セーフティクリティカルな機能が詳細に説明されます。

三つ目のパートは AUTOSAR [10]です。これは、ソフトウェアアーキテクチャと基本的なソフトウェア構成に関するものです。ソフトウェアアーキテクチャは、必要なソフトウェアのフィーチャを含む FDA モデルから作成できます。

四つ目のパートはモデルベースの安全分析です。このステップでは、故障モデルの故障ツリー（FTA）が ADL のエラーモデルや外部ツール[11][12]を使用したシミュレーションモデルから自動的に生成されます。

最後のパートはシミュレーションと検証です。このパートは本研究の中で開発されたものです。このステップではエラーシミュレーションと要件の検証が実行され、開発初期段階でこれが可能になります。一方で、安全要件はシミュレーション環境を使用して検証されます。他方、エラーシミュレーションを用いて原因のシステムへの影響を判断することができます。したがって、ハザード分析とリスクアセスメントから定義されたシステムの上位事象を承認することができます。詳細については 2.4 節で説明します。

開発したアプローチにより、個々のステップの一貫性とトレーサビリティが実現できます。また、この手法により、ソフトウェアアーキテクチャからフィーチャモデルまで、そして安全分析からハザード

分析とリスクアセスメントまでの効率的なトレースが可能になります。

以下の節では、主要部分の詳細について説明します。

2.1. アーキテクチャ開発とAUTOSAR

アーキテクチャ開発は UML、SysML、または EAST-ADL で実現できます。私たちのタスクに対しては、アーキテクチャ記述言語 EAST-ADL（Electronics Architecture and Software Technology - Architecture Description Language）を精査します。

記述言語 EAST-ADL は ADL を代表するものであり、研究プロジェクト EAST-EEA（Electronic Architecture and Software Technology - Embedded Electronic Architecture）で最初に仕様が定められ、他の研究プロジェクト ATESST1（Advancing Traffic Efficiency and Safety through Software Technology）、ATESST2[7]や MAENAD（Model-based Analysis & Engineering of Novel Architectures for Dependable Electric Vehicles）[8]、SAFE(Safe Automotive Software Architecture)[9]、そして Synligare[21]において、EAST-ADL2 として発展したものです。さらに、この言語は AUTOSAR[10]がソフトウェアアーキテクチャの詳細な説明と実装に使用できるよう AUTOSAR に適合されました。これを用いて車両内の電子システムを記述し、車両エレクトロニクスの開発を容易にできます[5]。したがって、EAST-ADL はドメイン固有言語です。

EAST-ADL の目的は、エンジニアが標準化された形式で容易に車両の電子システムを表現し、説明できるようにすることです[5]。EAST-ADL は分析・設計目的だけでなく、機能要件や安全作業成果物のモデリングのために、さまざまな活動の中で使用できます[13]。

EAST-ADL のメタモデルは、4 つの異なる抽象化レベルで構成されており、それぞれが特定の役割を果たします。各レベルは「車両開発の異なる段階」[14]を考慮し、完全なシステムを表して EEA 全体に対する異なる視点を提供します[5]。

この 4 つのレベル「車両レベル、分析レベル、設計レベル、実装レベル」について、次の図 3 で詳しく説明します。

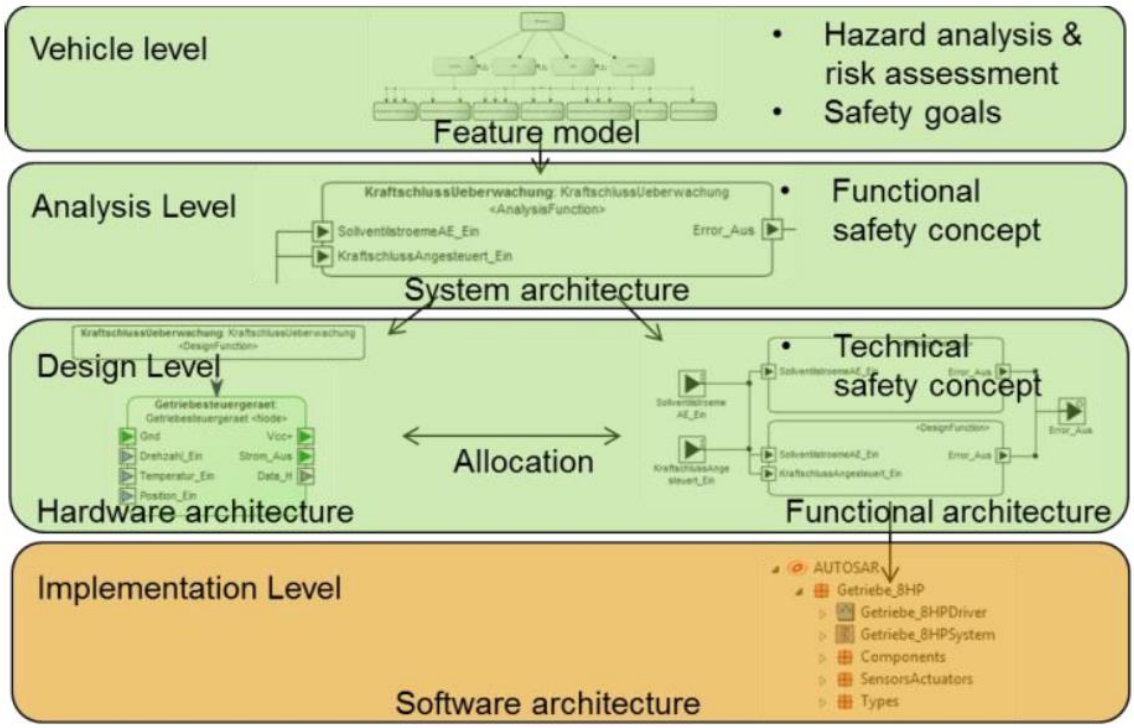

図3　EAST-ADLのレベルと本アプローチでの安全活動

前述したとおり、これらの抽象化レベルのライブラリ要素は図 4 に示すよう、さらに拡張されました。EAST-ADL 抽象化レイヤーのライブラリ要素には、対応する安全目標、安全要件、および安全モデルとシステムモデルの関係を有効にする ASIL 分類に関する追加情報が含まれています。さらに「verified」属性は、その機能が以前の開発フェーズで既に検証済みかどうかを示します。この情報は、安全事例の生成に非常に役立ち、ISO 26262-4:2018 の 7.4.8.1 節からも必要です。

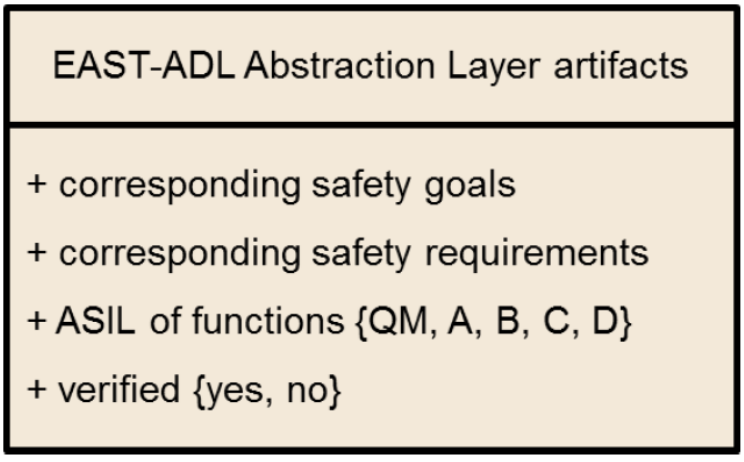

図4　EAST-ADL抽象化レイヤーの拡張

車両レベルでは、車両やシステム特性が多くの機能とともに記述されます。このステップでは「what」という質問はアーキテクチャ開発で実現する必要がありますが、「how」という質問は、そうではありません。リスクをできるだけ早く検出して除去するには、同じレベルでハザード分析とリスクアセスメントを並行して実行することが理にかなっています[15]。

分析レベルは、機能分析アーキテクチャ（functional analysis architecture：FAA）でフィーチャの実現を関数の形で記述します。ここでは、関数上、および抽象センサーとアクチュエータ上でフィーチャのトップダウンな分割を見つけます[16]。

設計レベルは FDA と HDA で構成されています。FDA は FAA の機能から成り、さらに詳細に分解することができます。加えて、割り当てグラフがモデル化され、FDA の機能ブロックが HDA のハードウェアコンポーネントに割り当てられます[18]。

実装レベルでは、AUTOSAR で FDA のソフトウェアアーキテクチャを 1 対 1 で記述します[17]。

2.2. 安全拡張

EAST-ADL は、安全拡張されたディペンダビリティパッケージを提供しており、図 6 に示すようにハザード分析やリスクアセスメント、安全目標などの安全開発プロセスの作業成果物を作成できます。

ディペンダビリティモデルはさらに、開発者が系統的なエラーとその伝播をモデル化して故障に結びつけるだけでなく、安全要件を作成し、安全分析を実行することをサポートするように設計されています。

前述のように本研究では、ディペンダビリティモデルのライブラリ要素を図 5 に示すようにさらに拡張しています。

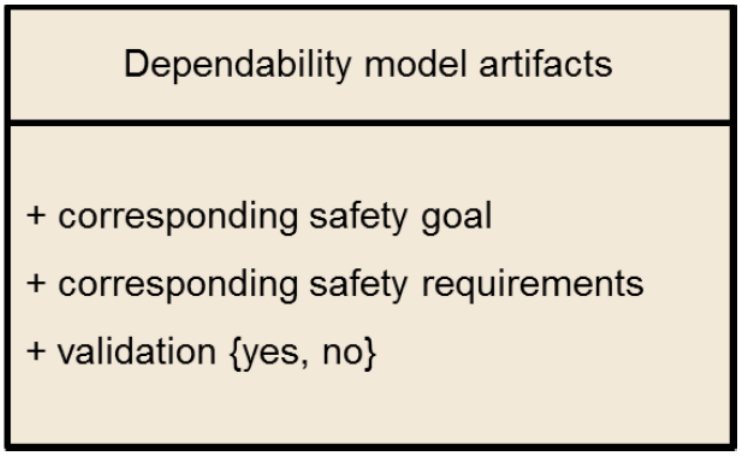

図5　EAST-ADLディペンダビリティモデルの拡張

本研究で開発された拡張機能により、開発された自動化スクリプトを使用して安全コンセプトの生成が可能になり、安全コンセプトは変更された要件モデルとディペンダビリティモデルから自動的に作成されます。

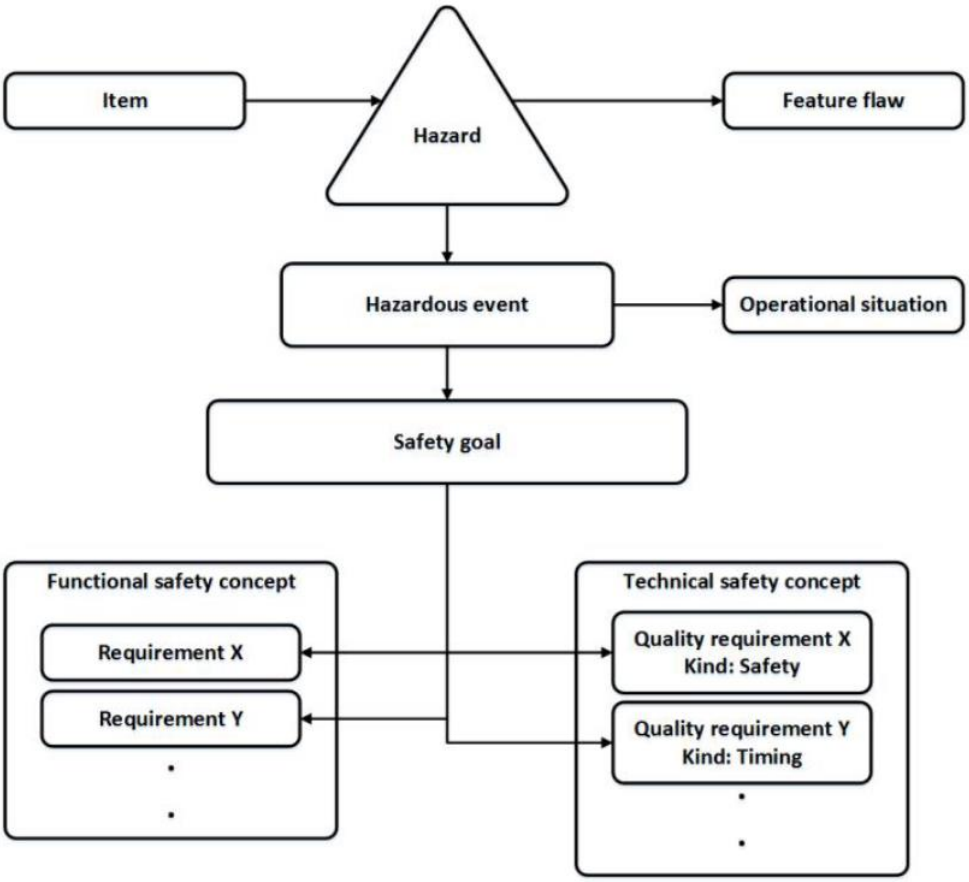

図6　ディペンダビリティモデル

システムの開発中に要件を定義する必要があり、これは、図 5 に示すように、個別の要件図でモデル化できます。要件モデルには、非セーフティクリティカル要件とセーフティクリティカル要件の両方が含まれる場合があります。したがって、システムアーキテクチャレベルでの要件モデリングから始めることをお勧めします。

EAST-ADL の要件モデルは、EAST-ADL[5]のメタモデルに適合した SysML（Systems Modeling Language）に基づいています。DOORS（Dynamic Object Oriented Requirements System）でも可能ですが、ユーザー定義の機能を要件に追加することができます。変更のトレーサビリティを向上させるために、ステータスや作成者、責任者などの情報が追加されます。要件モデルは ReqIF（Requirements Interchange Format）形式でエクスポートでき、他の要件ツールとの交換を容易に行うことができます。前述のように、要件モデルは図 7、8 のとおり ASIL 分類、対応する安全目標や分解適合性などの安全関連機能を備えて、さらに拡張されています。

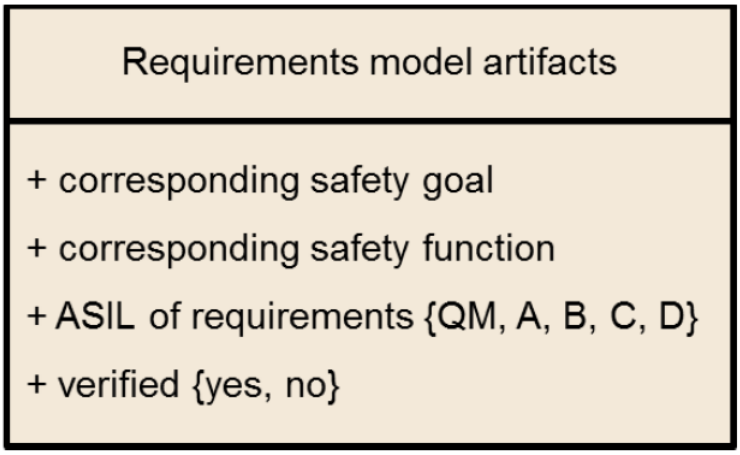

図7　EAST-ADL要件モデルの拡張

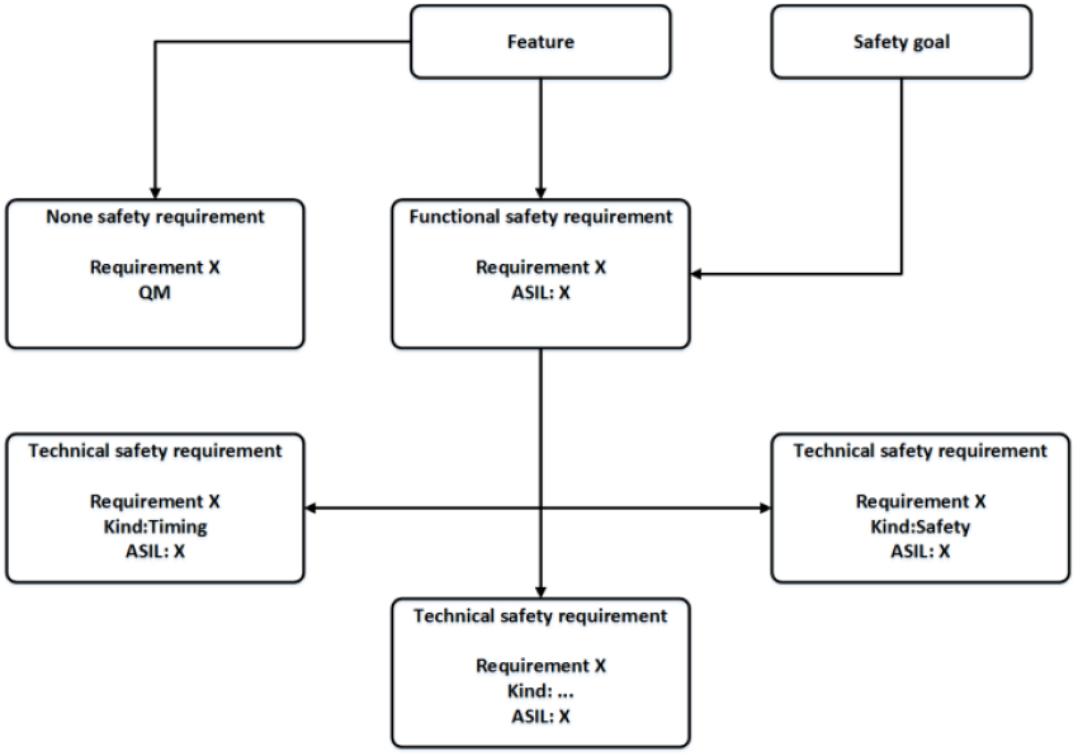

図8　要件モデル

本研究で行った拡張により、図 9 に示すように、開発された自動化スクリプトによる安全コンセプトの生成が可能になります。

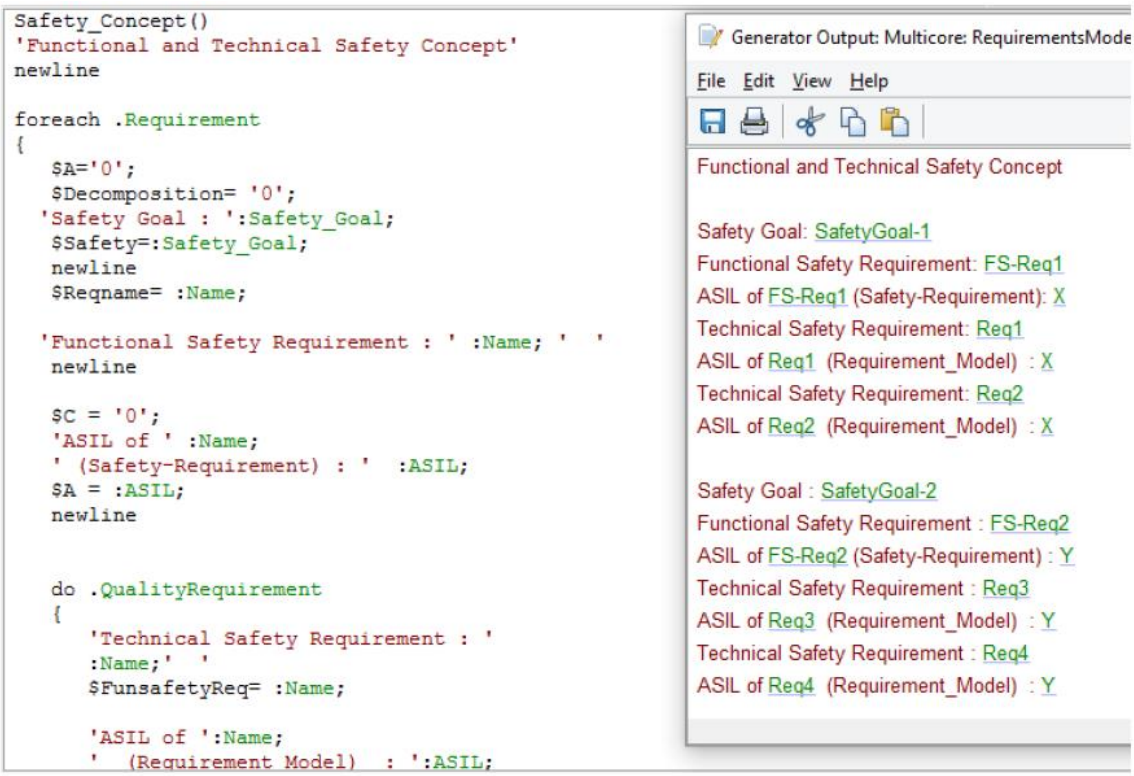

図9　安全コンセプトスクリプトと生成されたドキュメント

機能設計アーキテクチャモデル（FDA）からエラーモデルを自動的に生成することができます。その後、個々のサブシステムに対するエラーの説明とエラーロジックがモデル化されます。

車両一般や、特に電気自動車は複雑なシステムで構成されているため、ハザードを防ぐためのあらゆる対策を決定することは非常に困難です。もちろん、OEM（Original Equipment Manufacturers）とそのシステムサプライヤーは、すべてのリスクを許容可能なレベルまで減らすことに責任をもっています。このことは、すべてのハザードとリスクに関連する安全目標について、安全コンセプト、安全要件、および安全対策を定義する必要があることを意味します。そのため ISO 26262 では、自動車産業界での故障モデルに基づいて安全分析を実施する必要があります。例えば、故障ツリー解析の助けにより特定のハザードのエラー原因を見つけることができます[19]。

EAST-ADL では、エラーモデルでエラーとその伝播をモデル化して故障動作を再現できます[20]。エラーモデルに基づいて、HiP-HOPS [11]や ALTARICA [12]などの追加ツールを使用し、モデルベースの安全分析を実行できます。

図 10 は、エラーモデルとエラーロジックの作成法を示しています。

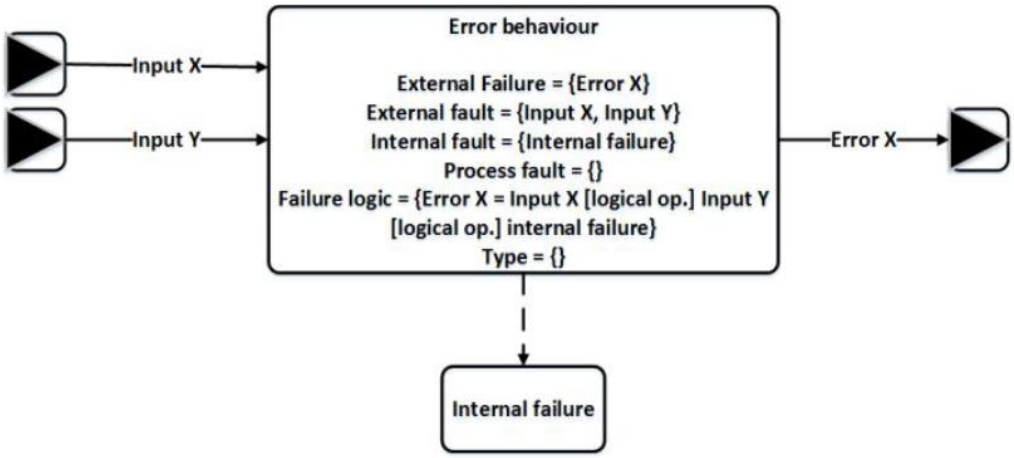

図10　エラーモデル

2.3. モデルベースの安全分析

HiP-HOPS[11]などの商用ツールを使用して安全分析を行うことが可能です。これらのツールではエラーモデルから自動的に故障ツリー解析（FTA）と故障モード影響解析（FMEA）を生成できます[11]。

安全分析により、適切な安全対策を講じて検出、回避すべきエラーの原因を見つけることができます。この目的のために、対策を導入するための安全対策と要件が定義されています。さらに、安全分析の結果に基づいて機能的・技術的な安全要件が定義または拡張できます。

2.4. シミュレーション

要件の検証やエラーシミュレーションには Simulink（MATLAB）、Dymola、CarMaker などのシミュレーション環境が使用できます。たとえば、電気自動車のパワートレインはシミュレーション環境の助けを借りてシミュレートでき、エラーモデルを設計・実装することができます。故障センサー値などのエラー原因を刺激信号として使用する方法により、車両の挙動をエラーモデルでシミュレートできます。シミュレーションは、要件の検証、開発の初期段階でのエラーの検出、そしてエラーのシステムへの影響の検出に使用されます。安全要件が意図したとおりに実装できないことが判明した場合、要件を再定義する必要があります。これを図 11 に示します。

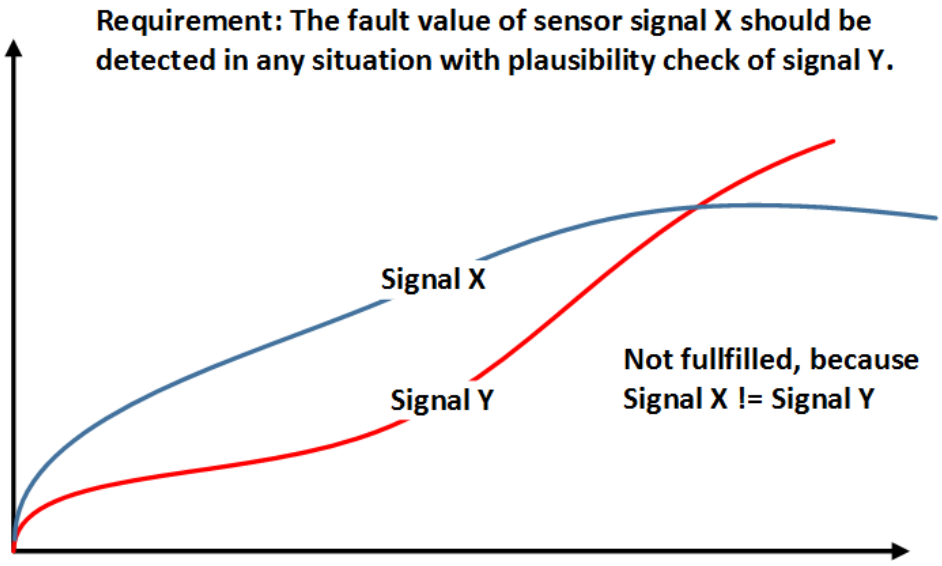

図11　検証とシミュレーション

図 12 は、定義されたハザードの評価手順を示しています。これらの故障のシステム反応を見つけるためにエラーの原因内でシミュレーションが実行されます。車両の挙動が定義されたハザードと同じである場合 HARA は承認され、それ以外の場合は、起こりうるハザードを検出し防止するために必要な安全対策を講じるために、ISO 26262-4:2018 の 7.4.8 節で要求されるように HARA をさらに拡張する必要があります。

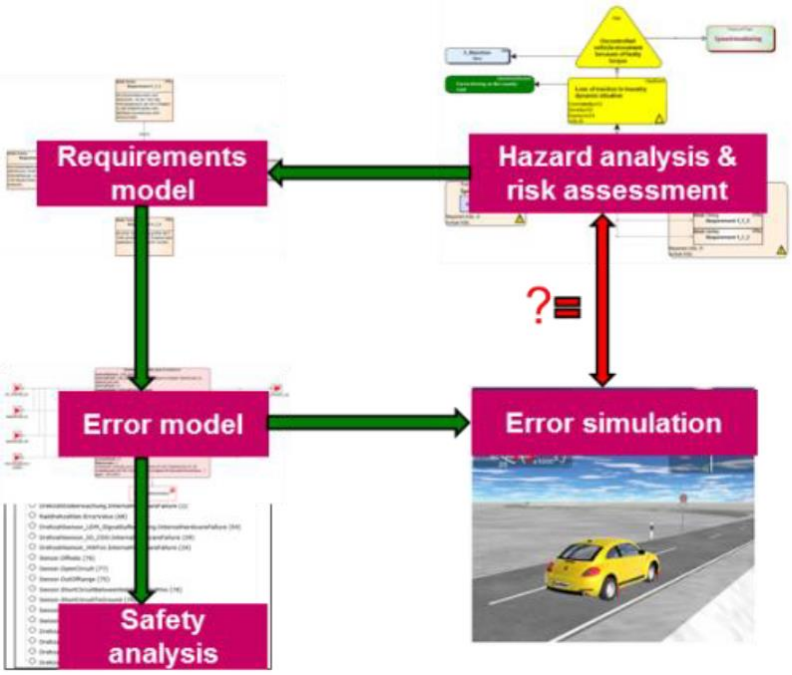

図12　エラーシミュレーション

3. ユースケース

開発したモデル駆動型アプローチの利点を示すために、セーフティクリティカルな機能の開発を示す

例を作成します。それはハザード分析とリスクアセスメントで始まり、安全目標および安全要件の検証で終わります。図 13 はシステムの典型的なディペンダビリティモデルを示しています。この具体的なケースでは、電気機械（E_Machine）はアイテムとみなされます。ハザードは「トルク障害による制御されない車両の動き」と定義されます。ハザード事象の ASIL は動作状況に基づいて分類されます。この場合、ハザード事象「限界での動的状況におけるトラクションの喪失」は、カーブを曲がることで危険な状況につながります。したがって、このハザード事象の ASIL は ASIL D（C3、S3、E4）に分類されます。次のステップでは、故障を検知して回避するために、「電気機械にはトルクがなく、車両は自由に回転している」という安全状態によって「限界での動的状況でのトルク障害なし」という安全目標が定義されます。この後、安全目標を達成するために機能安全コンセプトと技術安全コンセプトを作成します。機能的・技術的な安全要件は要件モデルからリンクされており、ディペンダビリティモデルで再指定してはなりません。

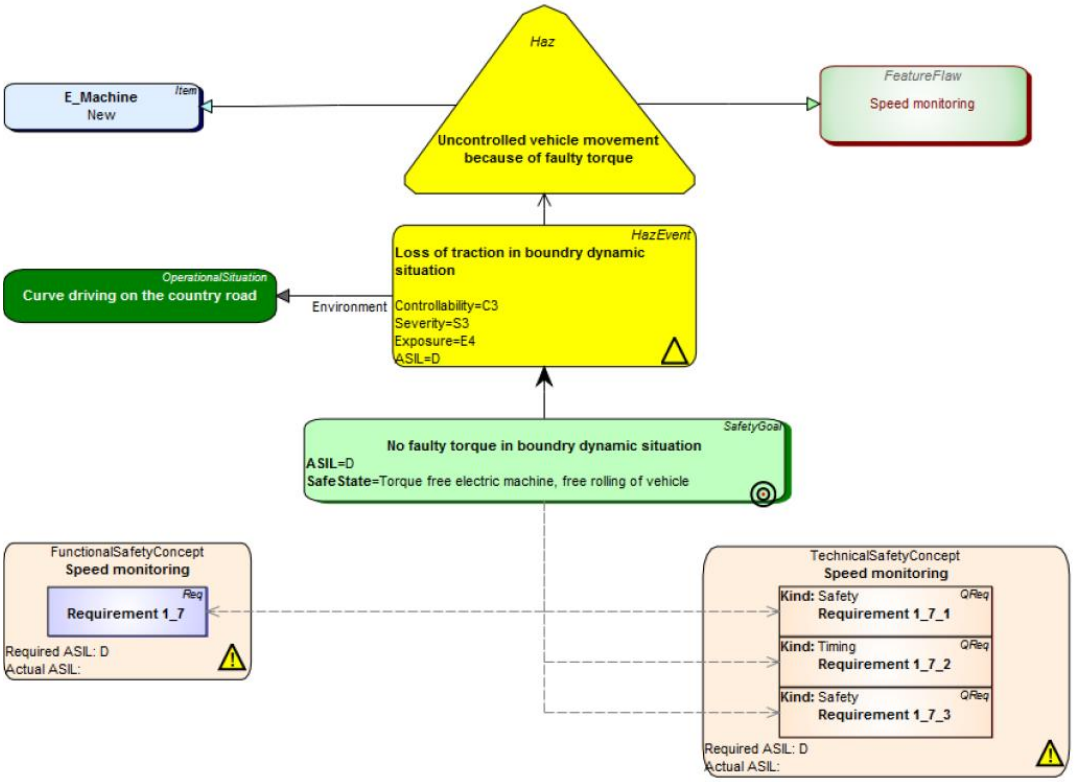

図13　ディペンダビリティモデル – パワートレインの例

機能的・技術的な安全要件は、要件モデル内の関連情報で指定されます。それらのアーキテクチャ要素への割り当ては、Satisfy 接続を使用するだけで実現されます。この場合、図 14 に示すように機能「Speed monitoring」(FAA)により安全目標が実現されます。

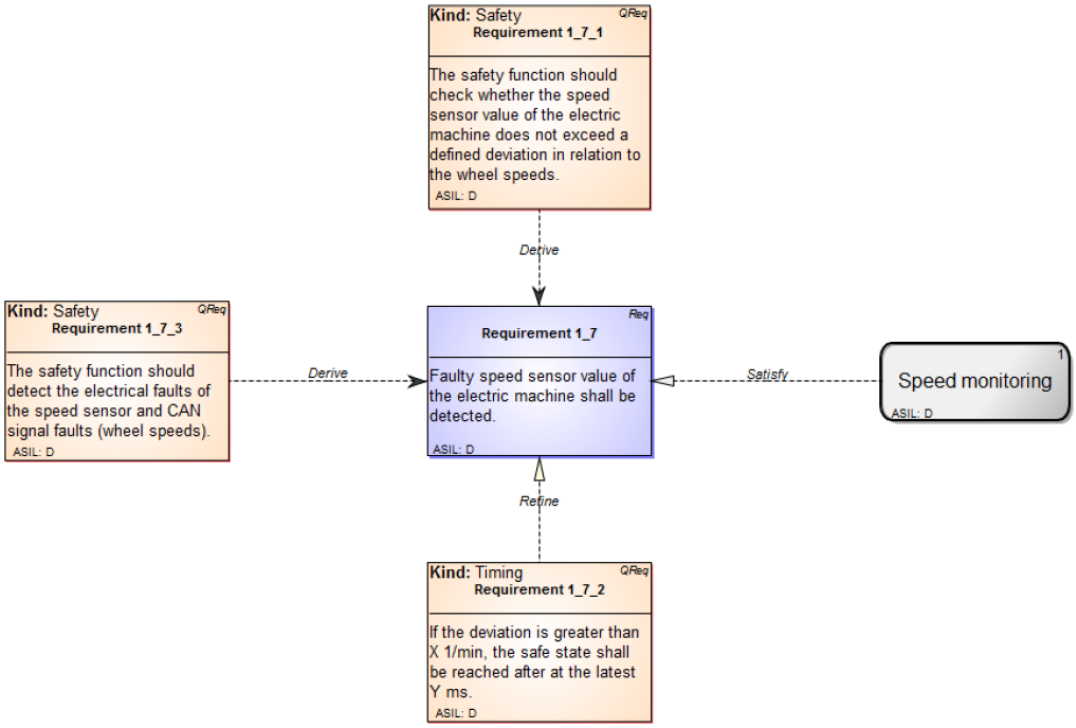

図14　要件モデル – パワートレインの例

図 15 に示す開発されたスクリプトを使用して、ディペンダビリティモデルと要件モデルから機能的・技術的な安全コンセプトを自動的に作成できます。

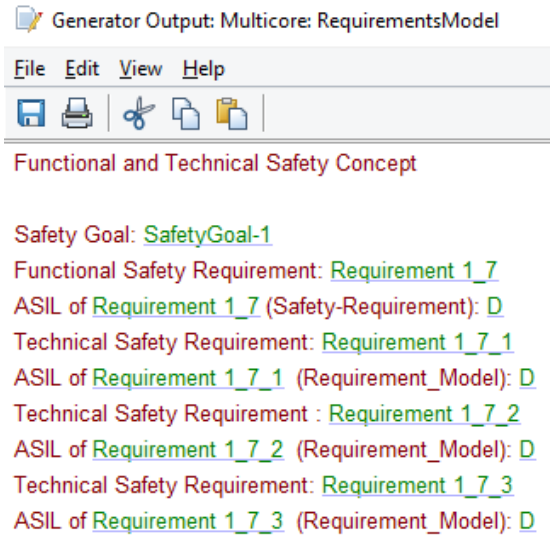

図15　機能的・技術的な安全コンセプト – パワートレインの例

図 16 に FDA モデルを示します。安全要件は、一貫性とトレーサビリティのチェックを可能にするために安全属性内で拡張された FDA モデルの下位機能内で実装されます。

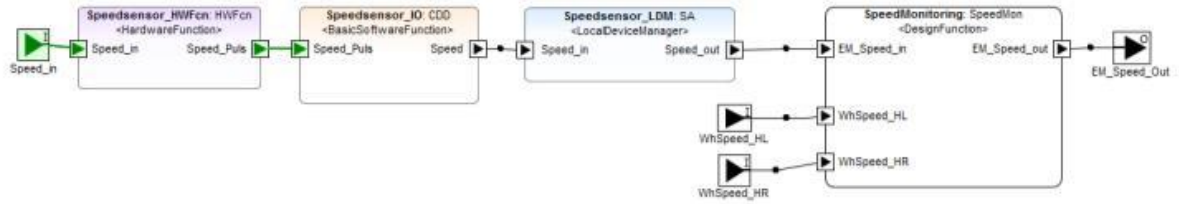

図16　機能設計モデル – パワートレインの例

ハザードのエラーモデルは FDA モデルから自動的に生成されます。その後、サブシステムのエラーロジックを、エラーの伝播を考慮して、考えられる障害原因とともに個別に説明します。エラーロジックには、図 17 に示すように、内部ハードウェア故障、内部ソフトウェア故障、および入力信号故障に関して、サブモジュールそれぞれで発生する可能性のあるすべての出力故障が含まれています。

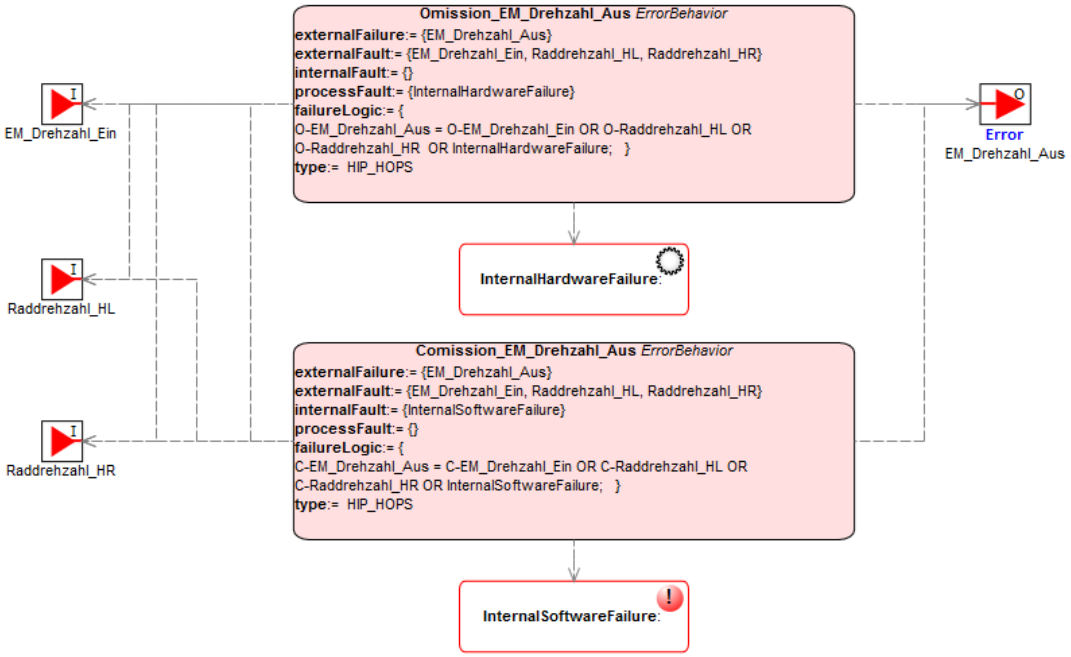

図17　エラーモデルのロジック – パワートレインの例

追加ツール HiP-HOPS を使用すると、既存のエラーモデルからモデルベースの安全分析を自動的に実行できます。このツールは、カットセット、故障ツリー解析（FTA）、故障モードと影響解析（FMEA）を生成できます。図 18 は Top Event「Faulty torque（トルク障害）」に対して生成された FTA を示しています。この上位事象（Top Event）につながるセンサー故障や内部ソフトウェア故障、内部ハードウェア故障などの原因がこの FTA に一覧表示されています。この場合定性的な安全分析が実現されますが、このツールでは定量分析を実行することもできます。

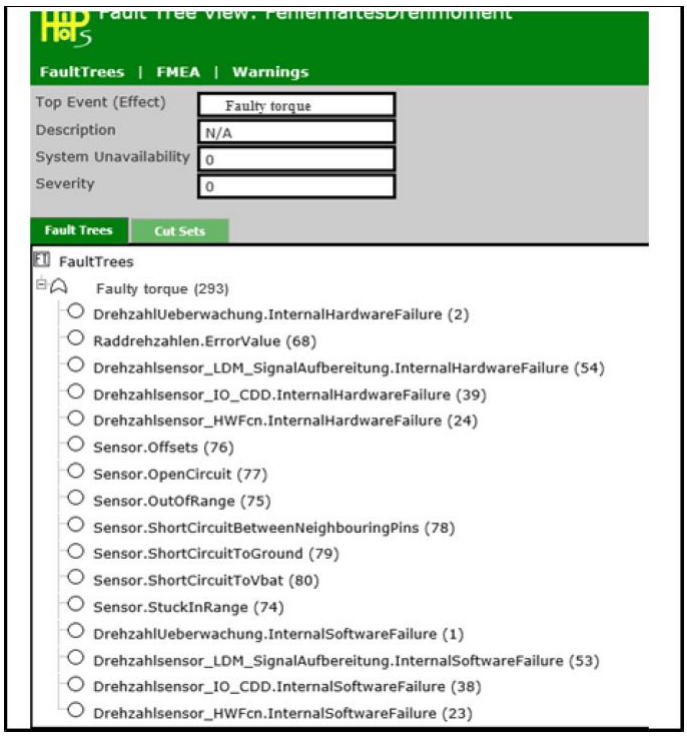

図18　HiP-HOPSによるモデルベースの安全分析 – パワートレインの例

最後に、HARA 内で以前に定義されたシステム効果（上位事象）を検証するために、エラーシミュレーションが実行されます。このため、シミュレーションは故障ツリー解析に関するエラー原因（基本事象)内で刺激されます。これらの基本事象は上位事象につながるはずです。シミュレーション後、結果が HARA の結果と比較されます。両者が同じ場合、定義されたハザード事象は正しいもので、そのように検証されます。ただし、車両の動作が定義された事象以外の場合は、HARA を拡張する必要があります。新たに検出されたシステムへの影響を回避するために、必要に応じて新しい安全目標を定義しなければなりません。この例では、具体的なハザード事象(E-Machine の速度センサーの故障によるトルク障害)はトラクションの喪失につながり、車両が不安定な状態に陥ります。限界での動的状況では、例えばカーブを曲がるときドライバーはもはや車両を制御できなくなります。これは致命的な事故につながるかもしれません。図 19 は、技術的な安全要件の検証結果とエラーシミュレーションに関する安全目標の妥当性確認の両方を示しています。

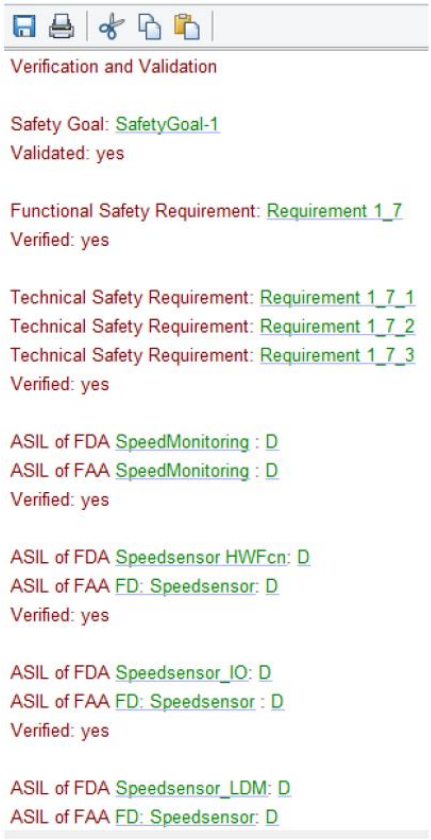

図19　検証と妥当性確認 – パワートレインの例

4. 結論

提案したアプローチは、開発者がモデルベースのシステム、ソフトウェア、および E/E システムの安全アーキテクチャを作成することをサポートします。現在、それぞれ固有の開発分野でさまざまなツールが使用されています。このアプローチの主要なコンポーネントであるアーキテクチャ記述言語 EAST-ADL は、システムと安全アーキテクチャの開発全体を EAST-ADL 内で実現できるように、それらのツールを置き換えます。提案アプローチには、プロジェクトの開発初期段階で機能安全コンセプトと技術安全コンセプトを検証でき、開発されたシミュレーション環境内のハザード分析とリスクアセスメントを妥当性確認できるという利点があります。

この手法の最大の利点は、拡張されたライブラリ要素と開発されたスクリプトの助けを借りて、さまざまな開発手順と安全作業成果物のトレーサビリティを証明することです。ユーザーがこの方法に慣れると、アーキテクチャ全体が理解しやすくなります。図 20 は、さまざまなレベル間の関係を強調し、異なるステップのトレーサビリティを示しています。

このアプローチにより、アイテムのシステムやソフトウェア、そして安全アーキテクチャを非常に迅速に理解できます。このアプローチを適用する場合、EAST-ADL の要件モデル、ディペンダビリティモデル、そしてライブラリ要素が安全機能（要件の ASIL 分類など）内で拡張されて、必要な属性を持つセーフティクリティカルな要件をモデル化でき、機能的・技術的な安全コンセプトを開発されたスクリプトで自動的に作成でき、システムと安全アーキテクチャを組み合わせることができます。したがってユ

ーザーは、どの機能に対してどの要件が実装されているかをすばやく見つけることができます。

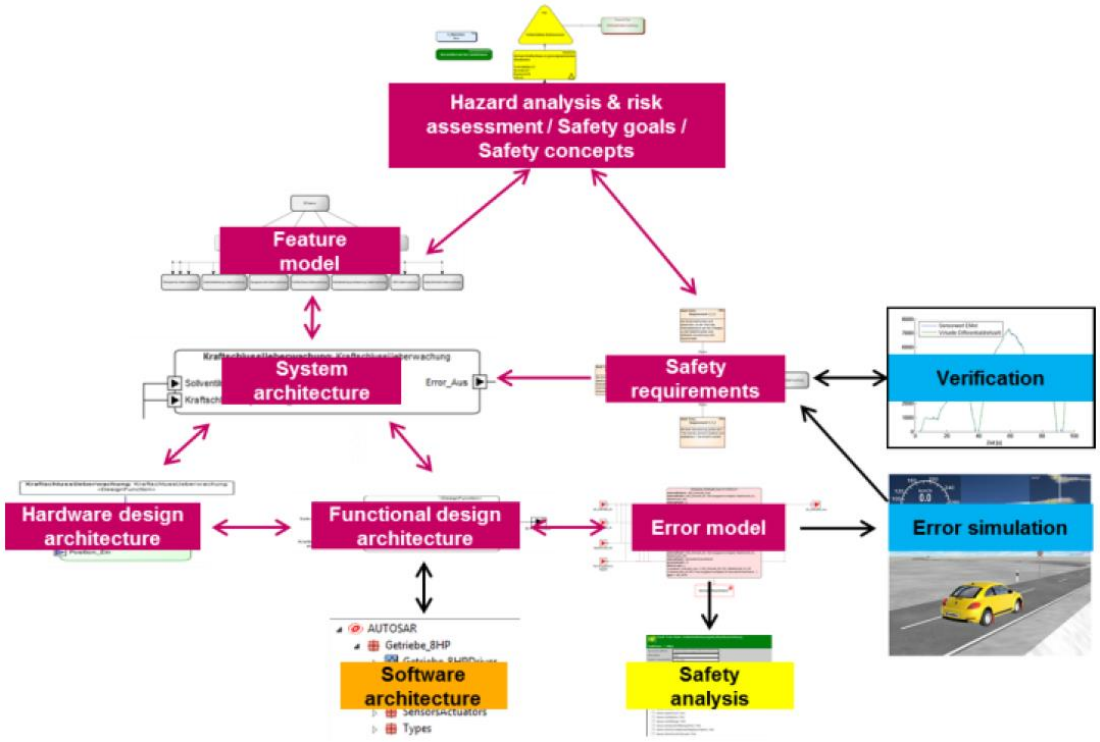

図20　開発手順のトレーサビリティ

このアプローチのもう一つの大きな利点は、EAST-ADL のレベルを ISO 26262 の作業成果物と要件にマッピングすることです。初期段階ですでに、EAST-ADL と ISO 26262 の関係は非常に重要です。EAST-ADL のディペンダビリティパッケージによって、アーキテクチャ開発ワークフローで ISO 26262 の安全性の側面を早期に検討できます。したがって、安全目標を発見するためにハザード分析とリスクアセスメントが行われます。

ディペンダビリティモデルを使用すると、定義された安全目標を達成するために必要な要件をモデル化できます。その後、機能的・技術的な安全コンセプトを作成できます。したがってユーザーは、どの安全コンセプトがどの安全目標に向けて策定されているのか、概要を把握します。EAST-ADL は、FDA モデルからエラーモデルを自動的に生成する可能性を提供します。このアプローチと外部分析ツールの助けにより、エラーモデルから安全分析を自動的に実行できます。

体系的なエラーは要件の検証とエラーシミュレーションによって検出でき、開発初期段階で安全対策を指定することも可能です。

最後に、提案したアプローチの利点は次のように要約できます。

- アーキテクチャ記述言語での安全関連機能のモデリング
- 自動車組込みシステムの効率的で一貫したモデルベース開発の達成
- ハザード分析・リスクアセスメントから安全要件に至るISO 26262作業成果物の開発スクリプトによるモデルベースでの自動作成

- トレーサビリティを達成し、ASIL を考慮した安全目標と安全機能の関係を示すためのシステムと安全開発の組み合わせ
- プロジェクトの開発初期段階での機能安全コンセプトと技術安全コンセプトの検証と妥当性確認
- 体系的なエラーとエラーのシステムへの影響の早期検出

参考文献

[1] B. Sari, H.C. Reuss, "A model-driven approach for the development of safety-critical functions using modified Architecture Description Language (ADL), *Electrical Systems for Aircraft, Railway, Ship propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC 2016)*.

[2] R. N. Charette,: This Car Runs on Code. Hyperlink: http://www.real-programmer.com/interesting_things/IEEE%20SpectrumThisCarRunsOnCode.pdf, February 2009. Accessed January 30, 2017.

[3] PWC DEUTSCHLAND: Autoindustrie treibt Chipnachfrage an. Hyperlink: http://www.pwc.de/de/automobilindustrie/autoindustrie-treibt-chipnachfrage-an.html. Accessed January 30, 2017.

[4] AK-L_Orientation-list-V1.2_2010-11-25_DE (AA-I3/AK 16 - Functional Safety).

[5] H. Blom, H. Lönn, F. Hagl, Y. Papadopoulos et al.: EAST-ADL – An Architecture Description Language for Automotive Software-Intensive Systems – White Paper Version 2.1.12. Hyperlink: http://www.maenad.eu/public/conceptpresentations/EAST-ADL_WhitePaper_M2.1.12.pdf. Accessed January 30, 2017.

[6] P. Cuenot, D. Chen, S. Gerard, H. Lönn et al.: Managing Complexity of Automotive Electronics Using the EAST-ADL. In: Engineering Complex Computer Systems, 2007. 12th IEEE International Conference on, 2007.

[7] ATESST2: ATESST2. Hyperlink: http://www.atesst.org. Accessed January 30, 2017.

[8] MAENAD: MAENAD. Hyperlink: http://www.maenad.eu/. Accessed January 30, 2017.

[9] SAFE: SAFE. Hyperlink: http://www.safe-project.eu/. Accessed January 30, 2017.

[10] AUTOSAR: AUTOSAR. Hyperlink: http://www.autosar.org/. Accessed January 30, 2017.

[11] HIP-HOPS: HiP-HOPS, Automated Fault Tree, FMEA and Optimisation Tool. Hyperlink: http://hip-hops.eu/. Accessed January 30, 2017.

[12] ALTARICA: ALTARICA. Hyperlink: http://openaltarica.fr/. Accessed January 30, 2017.

[13] ATESST2: EAST-ADL Overview. Hyperlink: http://www.atesst.org/home/liblocal/docs/ConceptPresentations/01_EAST-ADL_OverviewandStructure.pdf, 2010. Accessed January 30, 2017.

[14] M. Hillenbrand, „Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik / Elektronik Architekturen von Fahrzeugen“, Ph. D. Thesis, KIT Karlsruhe, 2011.

[15] ATESST2: EAST-ADL2 Vehicle Level. Hyperlink: http://www.atesst.org/home/liblocal/docs/ConceptPresentations/02_EAST-ADL_Vehicle_Level.pdf, 2010. Accessed January 30, 2017.

[16] ATESST2: EAST-ADL Analysis Level. Hyperlink: http://www.atesst.org/home/liblocal/docs/ConceptPresentations/03_EAST-ADL_Analysis_Level.pdf, 2010. Accessed January 30, 2017.

[17] ATESST2: EAST-ADL Overview Implementation Level. Hyperlink: http://www.atesst.org/home/liblocal/docs/ConceptPresentations/05_EAST-ADL_Implementation_Level.pdf, 2010. Accessed January 30, 2017.

[18] ATESST2: EAST-ADL Overview Design Level. Hyperlink:

http://www.atesst.org/home/liblocal/docs/ConceptPresentations/04_EAST-ADL_Design_Level.pdf, 2009. Accessed January 30, 2017.

[19] ISO 26262 “Functional safety of road vehicles”, 2011.

[20] EAST-ADL ASSOCIATION: EAST-ADL – Domain Model Specification – Version V2.1.12. Hyperlink: http://east-adl.info/Specification/V2.1.12/EAST-ADL-Specification_V2.1.12.pdf, 2013. Accessed January 30, 2017.

[21] Synligare: Synligare (literally “more visible” in Swedish). Hyperlink: http://synligare.eu/HomePage.html. Accessed January 30, 2017.