CB-002

STPA，FMEA を連携させた組込み制御ソフトウェア向けのハザード分析手法
A Hazard Analysis Method for Embedded Software Associating with UML, STPA, and FMEA

高橋 正和† 渡辺 喜道†
Masakazu Takahashi Yoshimichi Watanabe

1. はじめに

本論文では組込み制御ソフトウェア（EmBedded control Software: EBSW）を搭載した機器を使用した際、ハードウェアとソフトウェアの相互作用により発生するハザードの原因分析手法を提案する。特にハザードの原因となる EBSW の部位を明らかにする方法を提案する。

本論文で使用する用語について説明する。アクシデントとはシステムの損失につながる事象であり、損失とは使用者、環境、ミッション、当該システムへの悪影響を指す。ハザードとは、複数の（悪）条件が重なることでアクシデントが発生するようなシステムの状態である。アクシデントに至らない単純な故障はハザードとして扱わない。

近年、自動車、医療機器、航空宇宙機器等の工業製品は、高い機能と性能を実現するため、ハードウェアとソフトウェアを組合わせたシステムとして実現されており、機器と制御の構成が複雑になっている。そのため、開発時に想定していなかったアクシデントが使用時に発生している。このようなアクシデントはシステムの構成要素間の相互作用の結果としてハザードが発生し、その上でアクシデントを引き起こす（悪）条件が成立した結果として発生する。このようなアクシデント発生モデルを Systems Theoretic Accident Model and Process（STAMP）モデルと呼ぶ。さらに、STAMP モデルに基づいてアクシデントに至るハザードとそのシナリオを明らかにする安全性解析手法を System－Theoretic Process Analysis（STPA）と呼ぶ[1]。

提案手法の手順の概要を示す。①オブジェクト指向仕様記述言語（Unified Modeling Language: UML）で記述した EBSW の仕様とアクシデントの情報を入力として STPA を実施し、ハザードとハザードシナリオの一覧を出力する。②ハザードシナリオに相当するシーケンス図を作成し、ハザードの原因となるプログラム部位（Hazard Causal Factor: HCF）を明らかにする。③HCF に故障モード影響解析（Failure Mode and Effects Analysis: FMEA）を実施して HCF を発生させないための対策を行い、ハザードを回避する。これにより、安全な EBSW の実現に貢献する。なお、本論文ではテストで発見されるような単純なバグに起因するアクシデントについては、適切なテストで取り除かれているものとし、取り扱わない。

以降、本論文の構成について述べる。2 章では先行研究について述べる。3 章では提案手法の概要について述べる。4 章では提案手法の適用と評価について述べる。そして 5 章で今後の課題について述べる。

2. 関連研究

本章では先行研究および STAMP/STPA について述べる。

†山梨大学大学院 総合研究部 工学域 電気電子情報工学系
University of Yamanashi, Graduate School of Interdisciplinary Research Faculty of Engineering, Electrical and Electronic Information Engineering (Computer Science and Engineering)

2.1 関連研究

関連研究として様々な分野の安全な EBSW を開発するための規準、各種安全性解析手法について述べる。

はじめに様々な分野の安全な EBSW を開発するための規準について述べる。高い安全性を要求される工業製品のアクシデントは人命や環境に多大な影響を与える。そのため、工業製品の監督機関は、製造会社に対して開発規準に適合した EBSW 開発を要求している。そして EBSW 開発の中で安全性に関して十分な分析と対策を行うことを要求している。このような規準として、医療機器分野では JIS T2304[2]、IEC62304[3]、IEC82304-1[4]等、医薬品製造分野では Good Automated Manufacturing Practice[5]、自動車分野では ISO26262[6]、航空宇宙では DO-178C[7]、JAXA JMR-001[8]等が存在する。しかし、これらの規準には具体的な安全性解析の手順は明記されていないため、規準の解釈の誤りにより製造会社に追加作業を求めるケースも発生している。

次に各種の安全性解析手法について述べる。高橋らは、Failure Mode and Effects Analysis（FMEA）を用いて、医薬品製造装置の EBSW に生じる可能性のあるアクシデントを網羅的に明らかにして、その対策を決定する方法を提案した[9]。Weber らはアセンブリ言語で記述された航空機の EBSW に対して Fault Tree Analysis（FTA）を用いて故障原因を分析した[10]。Leveson らは、ソフトウェアの基本命令毎に故障の原因を明らかにする Fault Tree（FT） Template を準備し、それらを組み合わせることでソフトウェアの FT を作成することができることを示した[11]。高橋らは、故障が発生する過程を遡りながら FT template を結合して FT を作成するルールを作成し、機械的に FT を作成できることを示した[12]。Pai らは UML で記述された設計仕様から FT を作成してシステムの信頼性を求める方法を提案した[13]。しかし、これらの手法は特定のシステム構成要素の故障の原因を明らかにする手法であり、構成要素の相互作用に起因する複雑な故障を取り扱うことはできなかった。これに対して Leveson らはアクシデントの発生モデル STAMP を提唱し、それに基づいた安全性解析手法 STPA を提案した。これにより、システムの構成要素の間の相互作用で生じる複雑な故障（アクシデント）の原因を分析できるようにした。STAMP/STPA については 2.2 節で詳述する[14]。

2.2 STAMP と先行研究

はじめに STAMP モデルについて記述する。STAMP モデルではシステムはコントローラーと被コントロールプロセスから構成されていると考える。コントローラの内部にはコントローラーが想定している被コントロールプロセスのプロセスモデルが存在する。システムを動作させる場合、コントローラーはプロセスモデルの状態に基づいて、コントローラーから被コントロールプロセスへ制御指示（Control Action: CA）を与える。被コントロールプロセス

では CA に基づいて状態が変化し、その結果をフィードバックデータ（Feedback Data: FBD）としてコントローラーに戻す。図 1 に STAMP モデルにおけるコントローラと被コントロールプロセスの関係を示す。プロセスモデルと被コントロールプロセスが一致している場合にはシステムは安全状態となり、それらが一致していない場合にはシステムは非安全状態となりハザードが生じる。そして、特定の条件が成立した場合にアクシデントになる。

次に STPA の手順について記述する。

はじめに分析対象システムのアクシデントとハザードを決定する。ハザードから安全制約（Safety Constraints：SC）を決定する。

次に制御構造図（Control Structure Diagram: CSD）を作成する。CSD とは安全制約の実現に必要となるコンポーネント（サブシステム、機器）とコンポーネントの間の相互作用（CA、FD）を定義したものである。図 2 に CSD の一例を示す。

三番目に非安全な CA（Unsafe CA: UCA）を抽出する。CSD 内に記載された CA の中で安全制約の実行に必要となるものを識別する。それらの CA に対して「ハザードにつながる UCA を識別するための 4 種類のガイドワード（与えられないとハザードとなる、与えられるとハザードとなる、早すぎ・遅すぎ・誤順序でハザードとなる、早すぎる停止・長すぎる適用でハザードとなる）」を適用して UCA を抽出する。

四番目に各 UCA についてどのような条件が成立した場合にハザードとなるかを明らかにする。CSD から個々の UCA に関係するコントローラーと被コントロールプロセスを 抽出して UCA に関係するコントロールループを抽出する。図 3 に「コントロールループ上で HCF となる可能性のある 11 種類の項目（ガイドワード）」を示す。実際のコントロールループ中の UCA に対してガイドワードを一つずつ当てはめ、ハザードが生じる可能性があるか検討する。ハザードとなる可能性がある場合は、その条件を明らかにする。これが HCF となる。さらに HCF が発生した後、HCF が発生してからハザードに至るまでのシナリオを作成する。これをハザードシナリオと呼ぶ。

そして最後にハザードシナリオを検討してハザードを発生させないための対策を立案する。

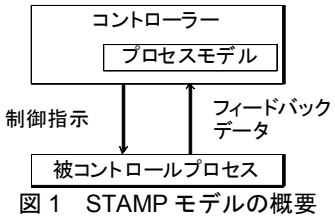

図 1　STAMP モデルの概要
（出典）[14]の pp.1 図 1.1-1

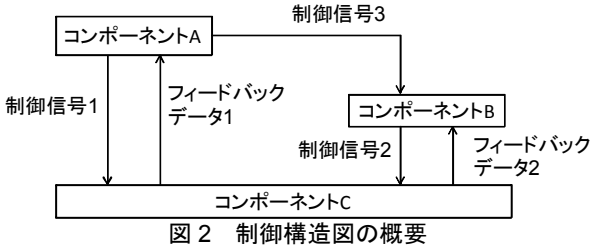

図 2　制御構造図の概要
（出典）[14]の pp.2 図 1.2-1

3. 提案手法の概要

本章では提案手法の概要を述べる。3.1 節で全体の概要を記述し、3.2 節では提案手法を構成する各作業について記述する。

3.1 提案手法の概要

提案手法の概要を図 4 に示す。提案手法は EBSW の要求定義と外部設計の完了後に適用可能となる（ユースケース図とクラス図の作成後）。提案手法は 4 つの作業で構成されている。はじめに「UML システム仕様の作成」では分析対象のシステムの要素、構成、制御に関する情報を記述する。次に「STPA を用いたハザードシナリオの作成」では分析対象システムのアクシデント、ハザード、制約条件、ハザードシナリオを決定する。三番目に「ハザードシナリオに対応するシーケンス図の作成と HCF のクラスへの割り当て」では EBSW のユースケース図とクラス図の情報をもとにハザードシナリオの内容に相当するシーケンス図を作成する。そして、EBSW 内のハザードの原因となる部位（HCF）を明らかにする。最後に「HCF 毎の FMEA の実施」では HCF に対して FMEA を実施してアクシデントの影響度を評価し、必要に応じて HCF を発生させないための（ハザードを発生させないための）対策を実施する。

3.2 提案手法を構成する各作業

本節では図 4 を構成する各作業の内容について説明する。

3.2.1 UML システム仕様の作成

「UML システム仕様の作成」では分析対象システムのユースケース図とクラス図を作成する。以降、これらを合わせて UML システム仕様と呼ぶ。ユースケース図では分析対象システムと ESBW と相互作用をするハードウェアを記述する。その際、ハードウェアはアクターで記述され、「ハザードシナリオのシミュレーションと HCF の EBSW への割り当て」でシーケンス図を作成する際に使用される。クラス図では EBSW のクラスとメソッドを記述する。

3.2.2 STPA を用いたハザードシナリオの作成

「STPA を用いたハザードシナリオの作成」では UML システム仕様とアクシデント、ハザード、安全制約から HCF を決定してハザードシナリオを作成する。

はじめに分析対象システムの使用方法を参考にして分析を行うアクシデントを決定する。そして、アクシデントを発生させるハザードと、それがアクシデントとなる条件を決定する。ハザードがアクシデントとなる条件をもとに安全制約を定義する。

次に UML システム仕様のユースケース図およびクラス図から CSD を作成する。CSD のコンポーネントはユースケース図のアクターおよびクラス図のクラスとする。コンポーネント間の CA は関連を有するクラス間でのメソッド呼び出しであり、CA の方向はクラス間の誘導可能性と同じ方向となる。コンポーネント間のデータは呼び出されたメソッドの戻り値となる。図 5 に UML システム仕様と CSD の対応関係の例を示す。

三番目に CSD 中の CA と「ハザードにつながる 4 種類のガイドワード」の全組合せの中から UCA を抽出する。表 1 に UCA 識別表を示す。表のセルの中には UCA と違反する安全条件を記述する。

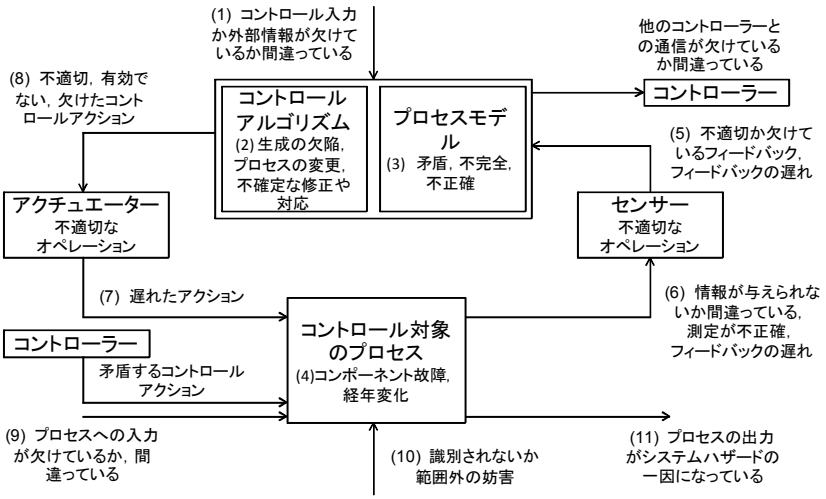

図 3　コントロールループ上で HCF となる可能性のある 11 種類の項目
（出典）[14]の pp.9 図 2.5-1

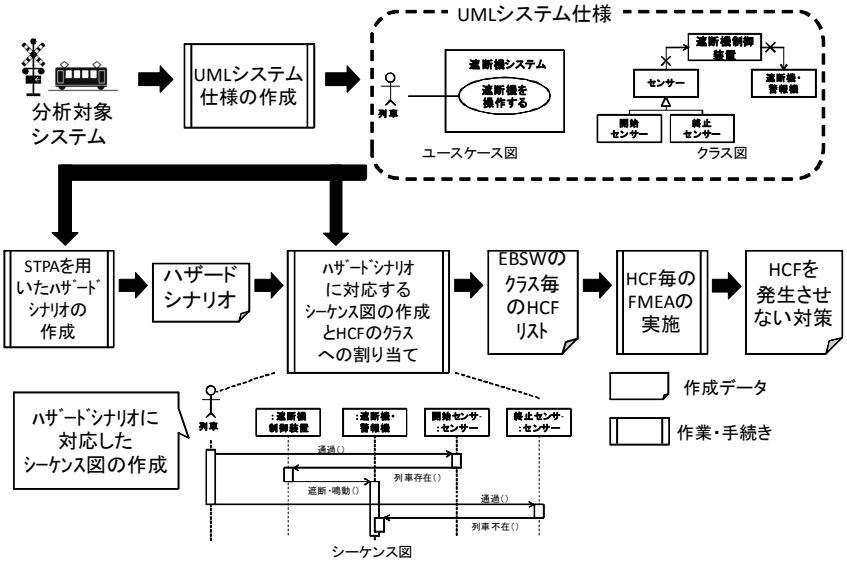

図 4　提案手法の概要

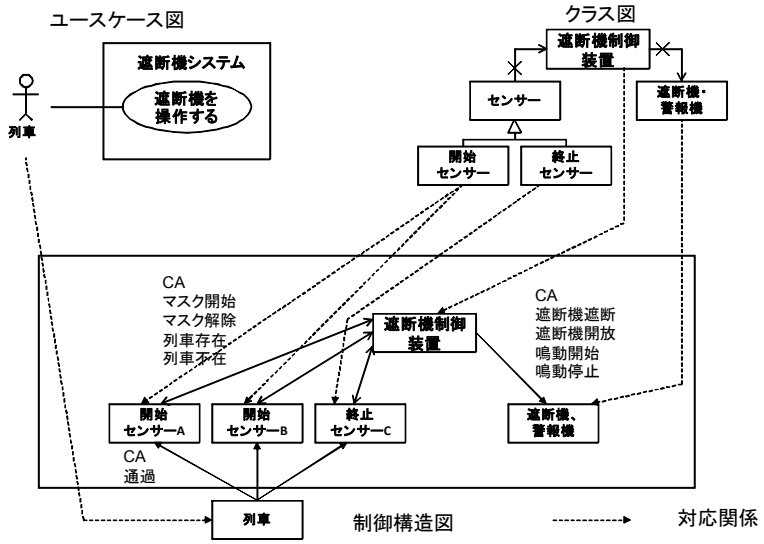

図 5　UML システム仕様と CSD の対応関係

表 1　UCA 識別表

	コントロールアクション	与えられないとハザード	与えられるとハザード	早すぎ、遅すぎ、誤順序でハザード	早すぎる停止、長すぎる適用でハザード
1	（コントロールアクション）	（条件）	（条件）	（条件）	（条件）
	….	….	….	….	….

（出典）[14]の pp.8 表 2.4-1

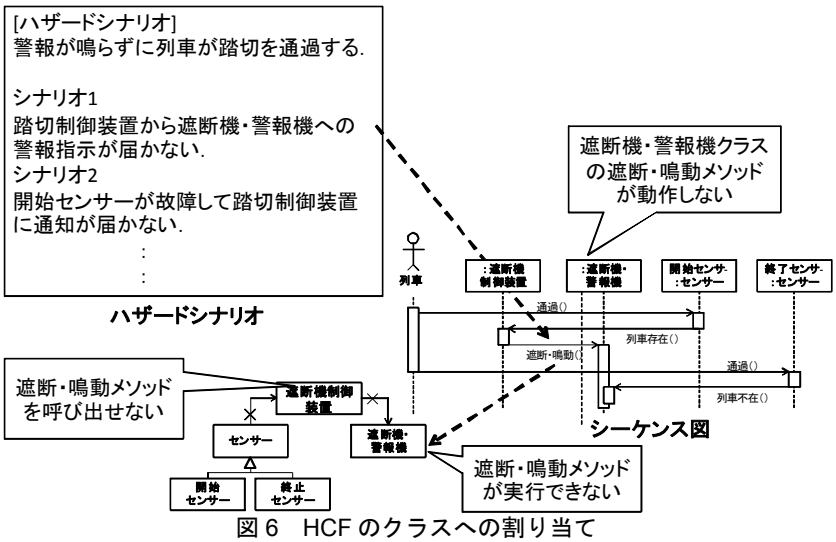

図 6　HCF のクラスへの割り当て

四番目に UCA と CSD からハザードを引き起こすコントロールループを識別し、コントロールループ上の UCA に HCF となる可能性のある 11 種類のガイドワードを一つずつ当てはめ、ハザードとなるか判断する。ハザードとなる場合は、その条件（HCF）を明らかにする。そしてハザードに至る過程をハザードシナリオとして明らかにする。

3.2.3 ハザードシナリオに対応するシーケンス図の作成と HCF のクラスへの割り当て

「ハザードシナリオに対応するシーケンス図の作成と HCF の EBSW 要素への割り当て」では UML システム仕様とハザードシナリオから、ハザードシナリオに対応するシーケンス図を作成する。シーケンス図のライフラインにはユースケース図のアクター（ハードウェア）と EBSW のクラスを使用する。ライフライン間で授受されるメッセージは EBSW のクラスのメソッドとなる。メッセージの向きクラス図の誘導可能性と同方向となる。シーケンス図の通りにメッセージの授受が行われた結果、ハザードが発生する。従って、シーケンス図の通りにメソッドが実行／非実行されることが HCF となり、それらの HCF はメッセージ受信側クラスのメソッドに割り当てられる。全てのハザードシナリオに対して HCF のクラスのメソッドへの割り当てを行うことでクラスごとの HCF（メソッド）が明らかになる。図 6 に HCF のクラスへの割り当ての一例を示す。

3.2.4 HCF 毎の FMEA の実施

「HCF 毎の FMEA の実施」では EBSW のクラスのメソッドに割り当てられた HCF に対して機能レベルの FMEA を行い、HCF が発生した時の EBSW に与える影響を評価し、影響が大きい場合には HCF の原因を明らかにして、リスクを低減するための対策を施す。

ここで、組込み制御ソフトウェアに対する機能レベルの FMEA について述べる[9]。EBSW の故障モードは、EBSW のメソッドが本来の機能を発揮できなくなることとなる。しかし、EBSW はソフトウェアであるので経年変化で機能を発揮できなくなる（機能の逸脱）ことはないので、本来の機能を発揮できなくなるのは、使用方法を誤ったり（起動条件の逸脱）、想定されてた範囲外の入力を与えたために結果が誤ったり（使用条件の逸脱）する場合である。これらを EBSW の故障モードとする。そこで既存の EBSW の FMEA の結果を分析て標準故障モードと標準故障対策を決定した。表 2 にそれらの一覧を示す。

EBSW 向けの FMEA は以下の手順で実施する。クラスの HCF となるメソッドに対して標準故障モードが適用可能であるか検討する。適用可能な標準故障モードがあった場合には、それに対応した標準故障対策の中から適用するものを選択して、メソッドにその対策を実施する。最後に当該メソッドの重大性、発生確率、発見確率を決定してリスク優先度を決定する。リスク優先度が許容可能な程度であれば安全対策は終了となる。なおリスク優先度の決定には図 7 に示すリスク評価行列を使用する。

4. 提案手法の適用と評価

提案手法の評価を行うために踏切遮断装置制御システムの安全性解析を実施した。4.1 節で適用事例の概要、4.2 節で適用結果と評価を述べる。

4.1 適用事例の概要

踏切遮断装置の安全性解析を実施する。この踏切遮断装置制御システムは IPA が STPA の分析事例として使用しているものと同様とする[14]。図 8 に踏切遮断装置制御システムの概要を示す。対象システムは踏切遮断装置制御システム、警報機・遮断機、センサー（2 個の警報開始センサーA と B、1 個の警報終止センサーC からなる。なお、これらのセンサーでは列車の方向は識別できない。）で構成されている。

踏切遮断装置制御システムに対する要求を以下に示す。

表 2　EBSW の標準故障モードと標準故障対策

故障モード	システムへの影響(故障)	対策の方針	標準故障対策
起動条件の不良	該当する作業ができない， システム不安全化	機能の起動条件の見直し	起動確認リスト作成，起動可能条件設定，起動不可能条件設定
		起動時の多重確認	起動確認リスト作成，起動ボタン複数化，起動確認回数複数化
		起動確認	起動状態確認データの追加，起動中表示の追加
		作業手順の徹底	機器操作標準作業手順書作成
終了条件の不良	該当する作業ができない， システム不安全化	機能の停止条件の見直し	停止確認リスト作成，停止不可能条件設定，停止可能条件設定
		停止時の多重確認	停止確認リスト作成，停止ボタンの複数化，停止確認回数複数化
		停止確認	停止状態確認データ追加，停止中表示追加
		作業手順の整備	機器操作標準作業手順書作成
		最優先での安全状態への移行	緊急停止機能追加
入力データの不良	不適切な処理が実行される	入力データの二重確認	複数人目視確認，多重入力
		入力データの形式と範囲確認	データ範囲確認，パリティ計算追加，フォーマット確認追加
		通信状態の確認	ネットワークトラブル標準作業手順書作成
出力データの不良	不適切な処理が実行される	出力データの二重確認	複数人目視確認，印刷確認
		出力データの形式と範囲確認	データ範囲確認，パリティ計算追加，フォーマット確認追加
		通信状態の確認	ネットワークトラブル標準作業手順書作成
アルゴリズム不良	該当する作業ができない， システム不安全化， システム不安定化， 不適切な処理が実行される	計算精度の上限確認（ｵﾊﾞｰﾌﾛｰ）	倍精度型変数の使用
		計算精度の加減確認（ｱﾝﾀﾞｰﾌﾛｰ）	倍精度型変数の使用
		ゼロ割の見直し	除算前の確認
		データ領域開放（メモリリーク）	プロセス終了時メモリ開放
		未定義アドレスの使用	使用可能なデータサイズの確認
		初期値設定確認	変数宣言時の初期化
プログラム破損	システム停止， システム不安定化	多重割り込み条件の見直し	状態とプログラムの組み合わせによる割り込みの禁止／許可
		データ領域の見直し	メモリ増設，ヒープ領域拡大
		再起動	再起動標準作業手順書の作成
		プログラムの再導入	プログラム導入標準作業手順書の作成
バックアップ不良	データ消失	データ保存領域の増設	保存データ取捨選択
		データ保存の多重化	ハードディスク多重化，書き込みデータ多重化
		バックアップ間隔の見直し	バックアップ間隔短縮，大容量ハードディスへの交換
セキュリティ不良	不適切な処理が実行される	利用者制限	パスワード設定，本人確認機能追加
		利用者制限の見直し	利用者の定期的な見直し
作業手順の誤り	該当する作業ができない， 不適切な処理が実行される	作業手順の徹底	機器操作標準作業手順書作成
ハードウェアの不具合	該当する作業ができない	ハードウェアの交換	不具合の判定，ハードウェアの修理・交換

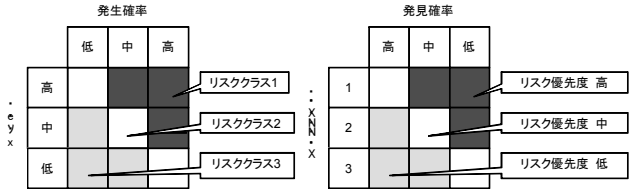

図 7　リスク評価行列

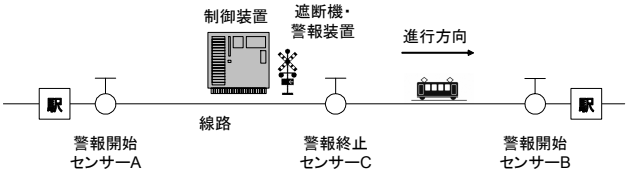

図 8　踏切遮断装置制御システムの概要
（出典）[14]の pp.11 図 3.1-1

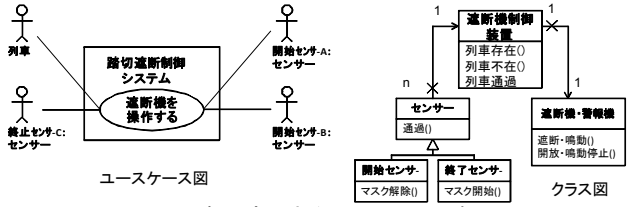

図 9　踏切遮断制御システムの概要

(1) 警報開始センサーＡまたはＢで列車を検知すると一定時間後に踏切鳴動を終止させる。
(2) 警報終止センサーC で列車を検知すると一定時間後に踏切鳴動を終止させる。
(3) A から B に向かう時、B をマスクする（列車を検出しないようにする）。
(4) B から C に向かう時、A をマスクする（列車を検出しないようにする）。

踏切の開閉動作や警報機の鳴動の制御を行う EBSW については記述されていないので、著者らが EBSW の構成を仮定した。図 9 にシステムの概要を示す。ユースケース図は列車アクターおよび各センサーアクターが踏切遮断装置制御システムを利用する形態となる。踏切遮断制御システムのクラス図は踏切遮断制御、センサー、遮断機・警報機の 3 個のクラスで構成されており、センサークラスは警報開始センサーと警報終止センサーのサブクラスを持つ。踏切遮断制御クラスはセンサーからの情報を基に遮断機と警報機の CA を決定する。センサークラスは列車を検出すると FBD として出力する。そして、遮断機・警報機クラスは CA を受け取り、夫々、遮断機と警報機を動作させる。

4.2 適用事例の概要

はじめに図 9 に示すテム仕様を作成する。

UML システム仕様を入力として STPA を実施する。以下の二番目から五番目までの作業は 3.2.2 で述べた手順と同様である。

二番目にアクシデント、ハザード、安全制約を識別する。今回はアクシデント「列車と人・車が踏切内で衝突する（A1)」、ハザード「列車が在線中に踏切が閉まらない（警報が鳴らない）（H1)」、安全制約「列車が在線中は踏切が閉まらなければならない（SC1）」について安全性分析を実施する。

三番目に CSD を作成する。踏切遮断機制御システムのコンポーネントは踏切制御装置、警報機・遮断機、警報開始センサーＡとＢ、警報終止センサーＣ及び列車となる。これらのコンポーネント間の制御、フィードバック、入出力情報を記入する。図 10 に CSD を示す。

四番目に UCA の抽出を行う。図 10 の CSD 上の制御について UCA を識別するガイドワードを当てはめ UCA を明らかにする。表 3 に UCA の抽出結果（抜粋）を示す。以降、「警報が鳴らず列車が踏切を通過する（踏切が閉まらない）（UCA1）、（SC1)違反」について分析を行う。

表 3　UCA 抽出結果

	コントロールアクション	与えられないとハザード	与えられるとハザード	早すぎ、遅すぎ、誤順序でハザード	早すぎる停止、長すぎる適用でハザード
1	鳴動開始指示	(UCA1)警報が鳴らずに列車が踏み切りを通過する(踏切が閉まらない) (SC1)違反	列車が来ないのに警報が鳴る	(UCA2)警報が鳴動する前に列車が踏み切りに到達する(閉まるのが遅く間に合わない)	開始指示が継続するので、列車通過後に鳴動停止指示を出しても鳴動し続ける
2	鳴動停止指示	列車が通過後も警報が鳴りっぱなし	(UCA3)列車が通過中に鳴動停止する (SC2)違反	(UCA3)列車が通過完了する前に鳴動停止する(閉めた後、開くのが早すぎる) (SC2)違反	(UCA1)列車通過後も鳴動指示が続き、次の列車が来ても鳴動しない(開始指示と競合) (SC1)違反
3	マスク開始指示	A、Cを通過した列車がBに到達したときに再鳴動する	(UCA4)列車が来ないのにマスク指示し、警報鳴動しない (UCA4)反対側の開始センサーにマスク指示し、警報鳴動しない (SC1)違反	(UCA5)終止センサーへのマスク指示が遅れ、列車の当該センサー通過に間に合わないと、マスク指示が残り、対向列車が2本続いたときに警報鳴動しない	(UAC6)列車が反対側の開始センサー通過後までマスク指示し続けると、対向列車が来ても鳴動しない (SC1)違反
4	マスク解除指示	(UCA6)反対側の開始センサーにマスク解除指示が出ず、対向列車が来ても鳴動しない(マスク指示後に列車が引き返す場合を含む) (SC1)違反	警報が再鳴動する	列車がBを通過完了前に出ると再鳴動する	解除を後続列車によるマスク開始指示と競合すると、マスクされずに再鳴動する可能性がある

（出典）[14] pp.20 表 4.4-2

表 4　UCA とガイドワードから導かれたハザードシナリオ

	①上位からの指示や外部情報の誤り・欠落	②CAが不適切・無効・欠落	③動作の遅れ	④プロセス入力の誤り・欠落	⑤意図しない、または範囲外の外乱	⑥不十分な制御・アルゴリズム
(UCA1)警報が鳴らずに列車が踏切を通過(踏切が閉まらず)		・踏切通過後に引き返す列車向け制御が不適切 ・鳴動停止継続により次の鳴動指示と競合		センサーAが故障してAから踏切制御装置への通知が欠落		
(UCA2)鳴動前に列車が踏み切りに到達(閉まるのが遅い)			・警報機の動作遅れ			・制御装置の動作遅れ
(UCA3)列車が踏切を通過する前に鳴動停止(開くのが早い)					列車がAを通過後、踏切に到達する前に、Cが外乱により短絡する	
(UCA4)不正なマスク開始指示が出て、列車がきても警報鳴動しない		・踏切制御装置の状態管理が不適切				・踏切制御装置の状態管理が不適切
(UCA5)マスク解除指示漏れで、列車が来ても鳴動せず			・超高速列車に対応できずにマスク解除の指示遅れ	・レール上の物体による外乱		・制御装置の処理に問題があり、マスク解除の指示遅れ
(UCA6)マスク開始指示遅れ	・誤った外部入力(外乱)でマスク解除漏れ	・制御装置への処理遅れでマスク解除漏れ	・状態制御誤りでマスク解除漏れ	・不正な外部入力によりマスク解除漏れ		・非定常運行への対応遅れでマスク解除漏れ

（出典）[14]の pp.23 図 4.5-2

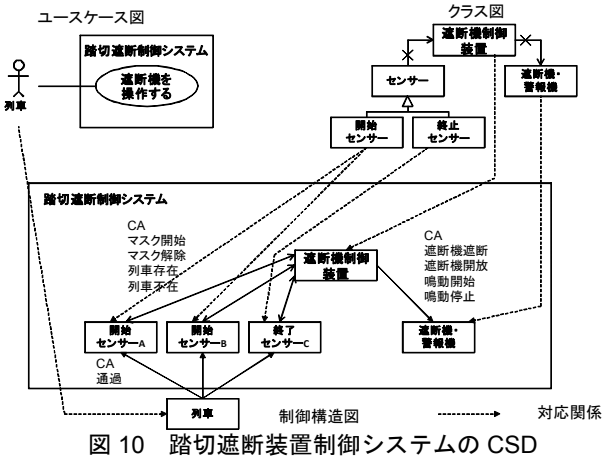

図 10　踏切遮断装置制御システムの CSD

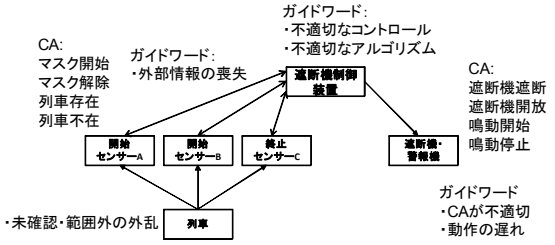

図 11　CSD への HCF を特定するガイドワードの割り当て

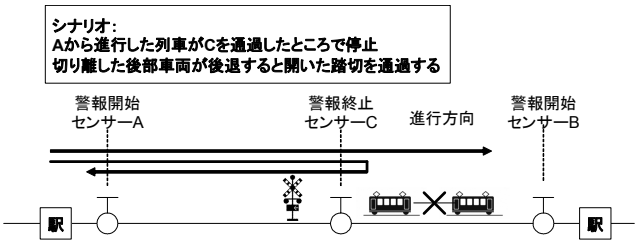

図 12　ハザードシナリオの一例
(出典) [14]の pp.24　図 4.5-2

五番目に UCA がハザードとなるか（安全制約違反するか）を検討する。CSD の各制御に HCF を特定する 11 種類のガイドワードを一つずつ当てはめ、ハザードとなるかを検討する。図 11 に CSD への HCF を特定するガイドワードの割り当てた結果を示す。その結果、踏切遮断機制御システムには 6 種類のガイドワードが該当することが分かった。ここで全ての UCA に 6 種類のガイドワードを当てはめて、ハザードとなるかどかを検討する。表 4 に検討結果を示す。UCA1 についてはガイドワード②「CA が不適切・無効・欠落」がある場合とガイドワード④「プロセスへの入力の誤り・欠落」がある場合にハザードになることが分かった。

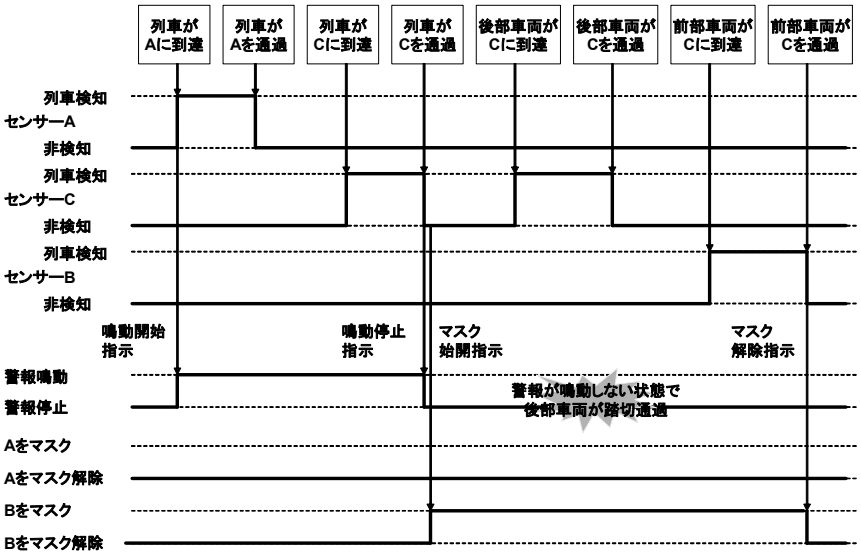

図 13　ハザードシナリオに基づいた列車の動作
（出典）[14]の pp.24 図 4.5-2

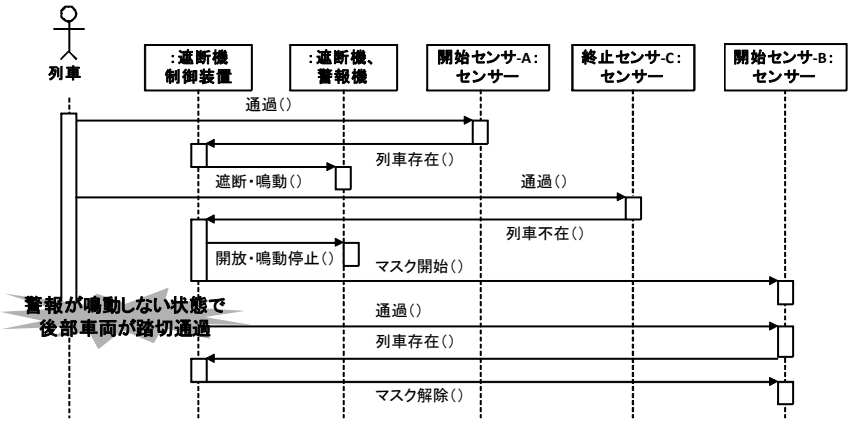

図 14　ハザードシナリオに基づいたシーケンス図

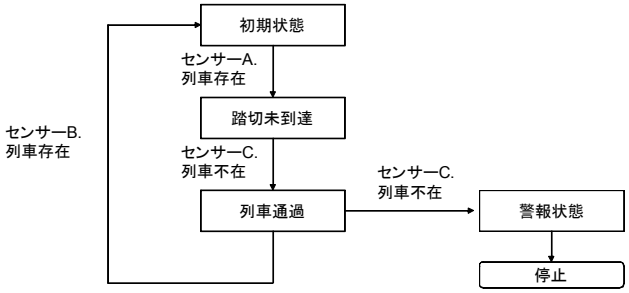

図 15　遮断機制御クラスの状態マシン図

具体的にはガイドワード②については「踏切通過後に引き返す列車向け制御が不適切でハザードになる」、「鳴動停止継続により次の鳴動指示と競合してハザードになる」となり、ガイドワード④については「センサーA が故障して A から踏切制御装置への通知が欠落してハザードになる」となる。以降ではガイドワード②の「踏切通過後に引き返す列車向け制御が不適切でハザードになる」場合に対応したハザードシナリオを作成する。図 12 に「踏切通過後に引き返す列車向け制御が不適切でハザードになる」場合のハザードシナリオを示す。

六番目にハザードシナリオに対応するシーケンス図の作成と HCF のクラスへの割り当てを行う。この作業は 3.2.3 節で述べた手順と同様である。ここでは EBSW に関わるハザードシナリオ「A から来た列車が C を通過した後、A 方向に引き返す」についてシーケンス図を作成する。図 13 にハザードシナリオの詳細を示す。そして、図 14 にその時のシーケンス図を示す。図 13 では列車は警報終止センサーC を通過した後、警報開始センサーB のマスクを開始し、その後、列車の一部が反転して警報終止センサーC を通過する。この時、遮断機は開放、警報機は鳴動停止の状態であるため、新たに遮断機装置制御システムが遮断機・警報機に開放・鳴動を行っても遮断機・警報機は動作しないので、遮断機が開放、警報機が鳴動停止の状態の踏切に列車が進入することになり、ハザードとなる。ここで、警報機・遮断機クラス、警報開始センサークラス、警報終止センサークラスが、単純に機器に対して入出力インタフェースを通じて制御指示を出力するだけであると仮定し、ハードウェア故障も無いと仮定する。するとハザードシナリオの割り当ては遮断機制御クラスにのみに行うことになる。

シーケンス図から、このクラスに割り振られるメソッドは列車存在、列車不在となる。これらについて、FMEA を実施する。列車存在メソッドは遮断機・警報機クラスの遮断・鳴動メソッドを起動するものである。このメソッドが

実行されても遮断機が閉まった状態となり、警報機が鳴動し続ける状態となるだけである。従って、遮断機が下りたままで踏切を渡ることができないという問題が発生するが、これがアクシデントに至る可能性は低いので、今回は対策を行わないこととする。一方、列車不在メソッドは、遮断機・警報機クラスの開放・鳴動停止メソッドを起動するものである。通常、列車不在メソッドは列車存在メソッドとペアで使用される。加えて、列車不在メソッドと列車存在メソッドは交互に実施されるべきである。表 2 の標準故障モードを適用すると機能の起動条件の不良が該当する。従って、標準対策として機能の起動条件設定と非起動条件設定の標準故障対策を施す。例えば、遮断機制御クラスに図 15 に示すような状態遷移を追加する。そして、列車通過待ち状態の時に列車不在メッセージを受けた場合には、安全監視担当者に連絡する（警報信号発行メソッドを追加する）、遮断機の遮断と警報機の鳴動を行う等の対策を行う。これらの対策によりハザードが発生する可能性を低減することができる。

他のハザードシナリオについても同様に適切な対策を施すことができることが確認できた。

提案手法を適用した結果、踏切遮断機制御制御システムのハザードを明らかにし、それを発生させない適切な対作を発見することができた。これにより、当該ハザードが発生するリスクを低減し、対象システムの安全性を高めることができた。一方、ハザードシナリオは多数考えられるため、効率的に対応策を検討する方法が必要であることが分かった。また、提案手法ではハザードシナリオ毎に対策を実施するので、対策の間に矛盾が生じる可能性があることが分かった。そのために対策間の矛盾を確認するための方法が必要になる。また、今回の事例では EBSW の設計変更によりハザードシナリオに対処した。しかし、実際には警報開始センサーA と B の間では列車の反転や切り離しを許可しないという標準作業手順を制定し、この問題を解決することも可能である。実際の対策を決定する場合には、安全性、費用、期間等を考慮したうえで、標準作業手順（ルール）の制定、ハードウェアの設計変更、ソフトウェアの設計変更のいずれの方法で対応するのが妥当であるかを検討し、効果的に安全な仕組みを構築する必要がある。

5. 提案手法の適用と評価

本論文では STPA、FMEA を組合わせたシステムの構成要素の相互作用により発生する複雑なハザードの原因を分析し、その対策を提案する手法を提案した。提案手法を適用することで、ハザードの原因を明らかにすることができるようになった。さらに、ハザードを発生させないための対策を立案できるようになった。しかし、分析と対策には時間を要することも分かった。特に、複雑なシステムのハザードを分析する場合は、ハードウェアとソフトウェアの数が多くなる、ハザードおよびハザードシナリオの種類が多くなる、分析結果に基づいた対策が矛盾する可能性がある等の問題が考えられる。今後は、制約条件や分析結果等を論理式で記述して論理演算を用いて機械的に取り扱うことで、効率的かつ適切な安全性解析を実現する仕組みの実現を検討する。さらに、提案手法を規模の大きいシステムのハザード分析に適用して評価するとともに、その結果を提案手法に反映させて、提案手法を改善していく。

謝辞

本研究はスズキ財団平成 29 年度科学技術研究助成「FMEA、FTA、HAZOP を連携させた組込み制御ソフトウェアの安全性解析手法」の支援を受けて実施いたしました。

参考文献

[1]N. Leveson, Engineering a Safer World, The MIT Press (2011).
[2]日本規格協会, JIS2304 医療機器ソフトウェアーソフトウェアライフサイクルプロセス (2017).
[3]International Electro technical Commission, ICE 62304 Medical Device Software (2006).
[4]International Electro technical Commission, ICE 82304-1 Health Software - - Part 1: General Requirements for Product Safety (2016).
[5]International Society for Pharmaceutical Engineering: GAMP5 A Risk-Based Approach to Compliant GxP Computerized Systems (2008).
[6]International Organization for Standardization, ISO26262 Road vehicles – Functional safety (2011).
[7]Radio Technical Commission for Aeronautics, DO-178C Software Considerations in Airborne Systems and Equipment Certification (2011).
[8]宇宙航空研究開発機構, JAXA JMR001 システム安全標準 (2008).
[9]高橋正和, 難波礼治, 福江義則, 医薬品製造に関わるコンピュータ化システム向けの FMEA を用いた運用リスクマネジメント手法の提案, 計測自動制御学会論文集, Vol.48, No.5, pp.285-294 (2012).
[10] W. Weber, Heidemarie Tondok, and Michael Bachmayer: Enhancing Software Safety by Fault Trees: Experiences from an Application to Flight Critical SW, Proc. of SAFECOMP2003, LNCS 2788, pp.289-302 (2003).
[11] N. G. Leveson and P. R, Harvey: Analyzing Software Safety, IEEE Transaction on Software Engineering, Vol. 9, No.5, pp.569-579 (1983)
[12] M. Takahashi, and Reiji Nanba, A Proposal of Fault Tree Analysis for Control Programs, Proc. of SICE Annual Conference 2014, pp.1719-1724 (2014).
[13] G. Pai and J. Dugan, Automatic Synthesis of Dynamic Fault Tree from UML System Model, Proc. of 13th International Symposium on Software Reliability Engineering, no page number, 2002.
[14] 情報処理推進機構, はじめての STAMP/STPA~システム思考に基づく新しい安全性解析手法~(2016).