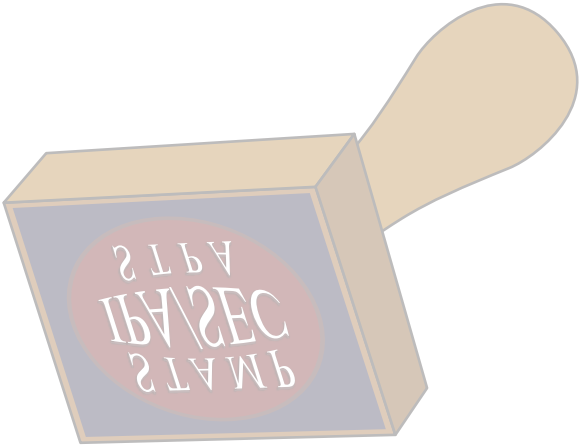
IPA Better Life with IT

はじめてのSTAMP/STPA（実践編）

～システム思考に基づく新しい安全性解析手法～

独立行政法人情報処理推進機構
Information-technology Promotion Agency, Japan（IPA）
技術本部 ソフトウェア高信頼化センター
Software Reliability Enhancement Center（SEC）
ソフトウェア高信頼化推進委員会
Software Reliability Enhancement Promotion Committee
システム安全性・信頼性分析手法 WG
System Safety & Reliability Analysis WG

Ver.1.0
2017年3月


STAMP
IPA/SEC
STPA

STAMP
IPA/SEC
STPA

はじめに

本書は、独立行政法人情報処理推進機構（IPA）技術本部ソフトウェア高信頼化センター（SEC）のシステム安全性・信頼性分析手法 WG（リスク評価手法チーム）の 2015 ～ 2016 年の活動成果をまとめた報告書である。2015 年に作成した「はじめての STAMP/STPA」[IPA2016] は、新しい安全解析手法として多くの産業界の方々に参考にされているが、さらなる産業界での活用を期待して、「実践編」という形で WG での活動成果をまとめた。

近年の我々の生活に満ち溢れている車や列車、航空機、ロボット、家電製品などの工学システムは、その内部にコンピューターと無線ネットワーク機能を持って、高度なソフトウェアによって制御されているが、これがますます複雑化・知能化しつつある。IoT（物のインターネット）と AI（人工知能）の時代といわれるゆえんである。さらには、既存の安全解析法や安全規格は、このような複雑システムの安全評価に対応できていないのも現実である。マサチューセッツ工科大学（MIT）の Nancy G Leveson 教授は、「旧来の安全分析はコンポーネント故障が事故を引き起こすという仮定に立ったものであり、コンポーネント間のコミュニケーション・ミスマッチが事故を引き起こすという近年の複雑工学システムの安全分析には適していない」と指摘し、新しい安全解析法として STAMP/STPA（System-Theoretic Accident Model and Processes/ System -Theoretic Process Analysis）という方法論を提唱している [Leveson2012]。しかしながら、日本の産業界の中にこのような新しい安全解析法が根付いているとは言えないのも現実である。その要因として、STAMP/STPA の基本的な考え方の理解が不足しているということと、現実の問題への具体的な応用方法が分からないという 2 点が指摘できる。

このような背景から、「はじめての STAMP/STPA」[IPA2016] では基本的な考え方と教科書に近い応用事例を中心に解説をしたが、「実践編」では、産業界でのニーズを考慮した多様な事例についての安全分析を試みた。ここで取り上げた事例は、いずれも、教科書で例示されているような標準的な制御構造とは異なっている。

列車の踏切の安全分析の事例では、フィードバック構造のない事例や、組織や人が絡んだ業務ワークフローを考慮した事例を取り上げた。さらには、ネット通販のようなエンタープライズ系システム解析事例なども取り上げた。これらの事例の制御構造図では、安全関連制御行動として誰が誰に指示をする、ないし、何を制御するという考え方そのものに複数の見方があり、その表現方法によってハザード誘発要因の発想の範囲が異なってくることもある。フィードバックがないことが安全設計の強みになっているという見方ができる事例もあるし、ブラントエンド（管理サイド）とシャープエンド（現場サイド）の責任の持ち方や対応の迅速さを評価できる事例もある。また、エンタープライズ系のように、安全制御行動ではなく、損失防止のための制御行動の欠陥を見つけるという使い方もある。

しかしながら、多様な応用の中で、STAMP/STPA の基本である「抽象化・階層化された制御構造図の表現の中で非安全制御行動とその誘発要因を見つける」という考え方は共通の思想として用いられている。複雑なシステムをトップダウン的に抽象化した機能表現で明示化することで、本当に大事な安全とは何か、本質的な損失とは何かを考えることができる。いわゆる「鳥の目・虫の目」の「鳥の目」であり、複雑な設計に埋もれ過ぎて俯瞰的な見方を忘れてしまいがちになる設計者への警告にもなる。

本 WG の活動がきっかけとなって、2016 年 12 月には、第一回 STAMP ワークショップ [IPA2016-2] が開催され多様な産業界からの参加を得た [IPA2016-3]。多くの参加者が、新しい時代の安全性をどうやって確保してゆくか悩んでいる現状を目の当たりにした。本「実践編」での事例は、このワークショップでの発表の中の一部であるが、この中から複雑システムの安全設計に役立つ情報を汲み取って頂ければ幸いである。

目次

図表目次

1. STAMP/STPA の活用方法

STPA の基本は、まず対象システム（要求仕様）を理解するといった準備（概念設計を含む）の後、回避すべきアクシデントとハザードを決めて STAMP によるモデリング（コントロールストラクチャーの構築）を行い、コントロールを行う側（コンポーネント）とその対象プロセス（コンポーネント）との間の相互作用において、安全制約が守られない状態に陥るシナリオを中心に解析する。（詳細は、「はじめての STAMP/STPA」[IPA2016] を参照）

図 1-1　STPA の実施手順

「はじめての STAMP/STPA」[IPA2016] では、図 1-1 で示したようにハザード誘発要因の特定までの手順を解説しているが、安全性解析の本来の目的は、ハザード誘発要因を排除したシステムの設計を行うことである。そこで、解析し発見されたハザード誘発要因に対し、そのハザード誘発要因を排除するための方策を検討し、安全制約としてシステムの設計にフィードバックする作業を Step2 の後に追加する。

安全制約には、

- システムの持つべき機能
- システム要素の設置方法
- システムの運用方法

も含まれる。

概念設計から実装設計完了時に至るそれぞれの段階で方策の実施の有無と具体的設計内容の有効性を確認することにより安全性を担保することができる。

ここで、要求仕様あるいは概念設計の安全性解析と実装設計は一旦終了するが、要求仕様、概念設計に新たな仕様、機能の追加・変更をすることにより元々の要求（目的）を高いレベルで充足することに気づくことがある。こうした場合には、再度新たな要求仕様、概念設計に対してハザード分析を行い新たなリスクの発生の有無を確認することになる。このサイクルを繰り返すことにより安全性を担保しながら、システムの価値を向上させることが可能になる（図 1-2）。

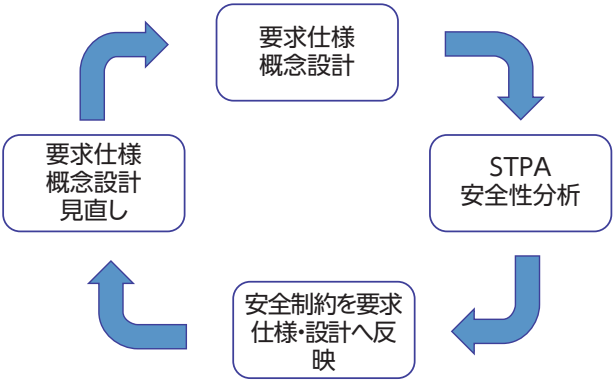


図 1-2　要求仕様改善サイクル

一方、実際のシステムは、人と組織と機械が複雑に絡み合っていることが多く、コンポーネントが階層的でなかったり、コントロールだけが存在してフィードバックが存在しなかったり、コントローラー（制御主体）と制御対象プロセスが一意に決めにくい（責任の所在が曖昧、分散している）場合もある。このような場合、コントロールの流れに沿ってそのままコントロールストラクチャー図にして解析してみる、あるいは制御主体と制御対象プロセスを入れ替えて解析してみるなどの方法も試してみることが肝要である。また、Step1 で UCA を抽出する際に、用意されている 4 つのガイドワードとハザード誘発要因（HCF）を特定するために提供されているガイドワードは、コントローラー、非制御対象プロセスの種類（機械・人・組織など）と当該システムドメインによっては適切さに欠ける場合もあるので、ガイドワードに捉われないよう考慮して STPA の活用を進めてほしい。

2.1. 機械と人間が連携する事例（鉄道踏切 “ とりこ検知 ”）

この章では、制御システムの解析事例として鉄道における踏切制御装置と連動して遮断中の踏切内に捉われた通行車・人を検出して踏切の安全を確保している “ とりこ検知 ” を取り上げる。

2.1.1. 対象システム（業務）の概要

“ とりこ ” とは、鉄道踏切において、遮断機が下りた状態で踏切内に人あるいは車が存在する状態のことを言う。“ とりこ検知 ” とは “ とりこ ” の有無を検知し、検出すると接近中の列車に伝えて衝突を回避するものである。

対象システム（業務）の登場人物は、

- 障害物検知装置
- 特殊信号発光機
- 通行車・人
- 列車と運転士
- 踏切制御装置

であり、それぞれの安全にかかわる役割は表 2.1-1 の通りである。

表 2.1-1　登場人物の役割

	登場人物	役割（安全関連責任）	備考
1	障害物検知装置	踏切遮断中に通行車・人があるか否か検知し、検知すると特殊信号発光機に点灯指示を出す。 “ とりこ ” 解消時に特殊信号発光機に消灯指示を出す。	
2	特殊信号発光機	障害物検知装置からの指示を受けて点灯・消灯する。	
3	通行車・人	踏切を通行する車・人。踏切遮断開始時に、踏切に進入してはならない。また踏切から退出しなければならない。退出できずに滞留すると “ とりこ ” という。	
4	運転士	特殊信号発光機の発光を確認（視認）するとブレーキをかけて列車を緊急停止させる。（“ とりこ ” との衝突回避）	目視
5	列車	運転士に制御されて踏切に向かって進行中の列車	
6	踏切制御装置	列車の接近をセンサーで検知して踏切を遮断するとともに障害物検知装置に動作開始を指示する。また列車通過完了をセンサーで検知して踏切を開通するとともに障害物検知装置に動作終了を指示する。	

また、対象システム（業務）のイメージを図 2.1-1 に示す。

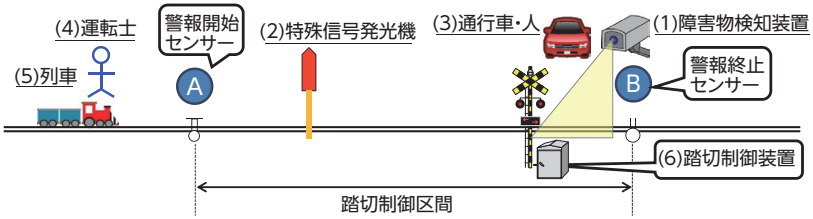


図 2.1-1　対象システム（業務）イメージ

以後、以下に示す手順に従って解析作業を進める。この手順には、Step2 で特定したハザード誘発要因を排除するための方策を検討し、安全制約としてシステムの設計にフィードバックする手順を追加している。

- ■ 事前作業　　前提条件の整理
- ■ 準備 1　　アクシデント、ハザード、安全制約の識別
- ■ 準備 2　　コントロールストラクチャーの構築
- ■ STPA Step1　　UCA（Unsafe Control Action：非安全制御動作）の識別
- ■ STPA Step2　　HCF（Hazard Causal factor：ハザード誘発要因）の特定
- ■ 対策の立案　　HCF を排除するための対策立案（設計上の安全制約）

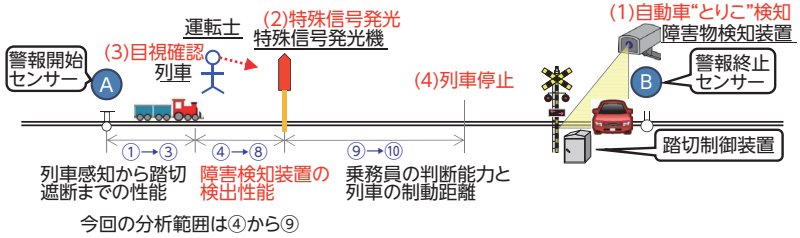
2.1.2. 事前作業

本業務の安全性を解析するに当たって前提条件を以下のように整理した。

1. 踏切遮断機・警報機は正常に機能するものとする
2. 分析範囲は、“とりこ”発生（踏切が遮断後）から列車停止までとし、列車停止後に乗客が線路に降りる、線路上を歩く、ことによるアクシデントは解析対象外とする
3. 障害検知装置はカメラと画像診断装置によるものとする

以下、解析の過程で追加した前提条件

1. 障害検知は、踏切が遮断後に作動する（障害物検知装置は障害物になることの予測機能を持たない）
2. “とりこ”状態が解消されると特殊信号発光機を消灯する
3. 特殊信号発光機、警報開始センサー、踏切の設置場所と各装置の性能の関係は図 2.1-2 の通りとする



障害検知システムの動作シーケンス(仕様)

① 開始センサーが列車到達を感知
　➢相互作用:センサー→制御装置
② 踏切制御装置が遮断開始
　➢相互作用:制御装置→遮断機
③ 踏切制御装置が遮断通知
　➢相互作用:制御装置→検知装置
④ 障害検知装置が検知開始
⑤ 障害検知/障害解消検知
⑥ 信号発光指示/消灯指示
　➢相互作用:検知装置→特殊信号発光機
⑦ 特殊信号発光機発光/消灯
⑧ 運転士が目視確認
　➢相互作用:特殊信号発光機→運転士
⑨ 停止判断、マニュアルブレーキ作動開始
　➢相互作用:運転士→列車
⑩ 列車停止

図 2.1-2　各装置間の連係動作

2.1.3. 準備 1：アクシデント、ハザード、安全制約の識別

分析対象システムのアクシデントを識別し、そのアクシデントを防止するためにシステムに装備されている安全機能を整理する。

アクシデントは次の 2 つが考えられる。

- 列車が " とりこ " 状態の車・人と衝突し、車の乗員・人、列車の乗員・乗客が死傷する
- 特殊信号が発光し続けて列車が走行できない

ハザードは " とりこ検知 " が適切に機能しない状態であり、安全制約はその裏返しであることから以下のように識別できる（表 2.1-2）。

表 2.1-2　アクシデント・ハザード・安全制約一覧

アクシデント（Loss）	ハザード（Hazard）	安全制約（Safety Constraints）
（A1）列車が " とりこ " 状態の車と衝突する。 ・通行中の人、車の運転手が死傷する ・列車の乗員、乗客が死傷する	（H1-1）" とりこ " 発生時に特殊信号発光機が発光しない	（SC1-1）" とりこ " 発生時に特殊信号発光機が発光すること
	（H1-2）" とりこ " 発生中に特殊信号発光機の発光が停止する	（SC1-2）" とりこ " 発生中は特殊信号発光機の発光が停止しないこと
	（H1-3）特殊信号発光機の発光を乗務員が目視確認できない	（SC1-3）特殊信号発光機の発光を乗務員が目視確認できること
（A2）特殊信号発光機が発光し続けて列車が走行できない	（H2-1）" とりこ " が発生していないのに特殊信号発光機が発光	（SC2-1）" とりこ " が発生していない時は特殊信号発光機は発光してはならない
	（H2-2）" とりこ " 対応処理完了後特殊信号発光機の発光停止できない	（SC2-2）対応処理完了後特殊信号発光機の発光停止できなければならない

今回は、アクシデント A2 は、人命・財産喪失という重大アクシデントではないため、解析対象をアクシデント A1 に絞った。

2.1.4. 準備 2：コントロールストラクチャーの構築

鉄道の主目的は、安全に列車を走行させることであり、踏切（とりこ検知を含む）は安全装置である。列車を走行させるために、運転士が列車の加減速・停止をコントロールし、踏切は列車の進入・進出により開閉・" とりこ検知 " を開始・終了するとともに " とりこ " を検知すると特殊信号発光機を発光させて運転士にフィードバックする。これらの制御関係を人（運転士）を中心にコントロールストラクチャーとして構築したものが図 2.1-3 である。

このシステムでは、踏切からのフィードバックが列車を経由しないで直接運転士に入っているのが特徴である。

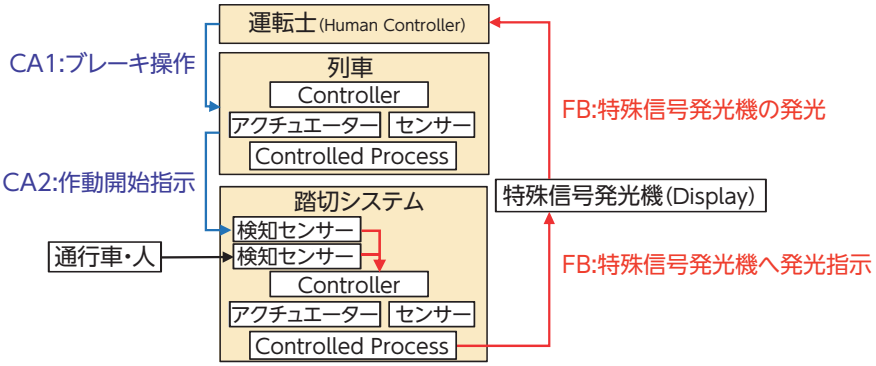


図 2.1-3　運転士を中心にしたコントロールストラクチャー

一方 “ とりこ検知 ” を中心に考えて踏切システムを展開（遮断機制御と障害物検知）したうえで、障害物検知装置を起点に要素間のコントロールの流れに沿って以下の図 2.1-4 のように構造を記述することもできる。この場合制御アクションに対するフィードバックはない。

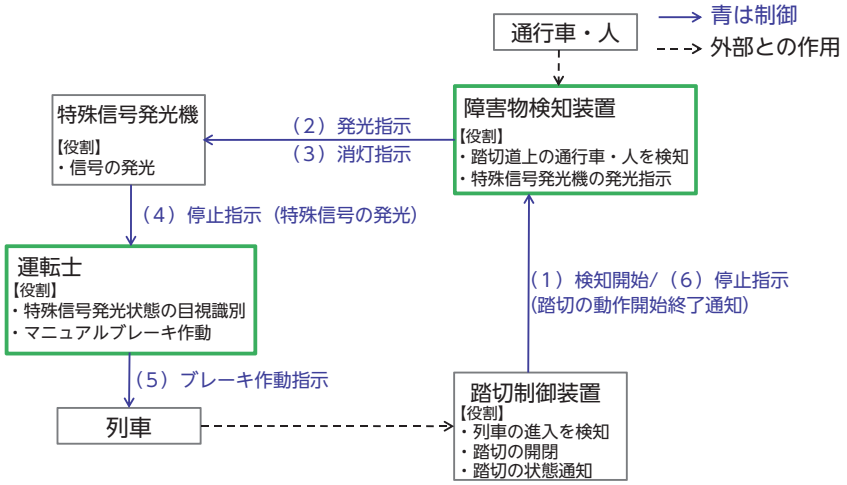


図 2.1-4　“ とりこ検知 ” の流れに沿ったコントロールストラクチャー

2.1.5. STPA Step1：UCA（Unsafe Control Action：非安全制御動作）の識別

人（運転士）を起点にシステムをトップダウンに構築したコントロールストラクチャー（図 2.1-3）と “ とりこ検知 ” 機能を中心にコンポーネントに沿って構築したコントロールストラクチャー（図 2.1-4）の 2 種類のコントロールストラクチャーがあるが、導出されるハザードシナリオはほぼ同じであるため、ここでは、コンポーネントに沿ったコントロールストラクチャーで説明する。

コントロールストラクチャー（図 2.1-4）のコントロールアクション（CA）全てに 4

つのガイドワードを適用して UCA を識別する（表 2.1-3　UCA 識別表）。

なお、黄色で示した UCA については、“ とりこ検知 ” とは関係なく通常の運転操作中に運転士が障害物を発見するとブレーキを動作させて停止すべきであるので今回の解析では、UCA とはしなかった。

ここでは、コントロールアクションがどのコントローラーからどの制御対象プロセスに出ているかが容易に判別できるように ”FROM”、”TO” の欄を設けるとともに、UCA にコントロールアクションの番号とどのガイドワードを適用したかがわかるように識別子を付けた。

UCAn-N/P/T/D

n：CA の番号

N：Not Providing　　P：Providing causes hazard　　T：Timing Too early/Too late

D：Duration Stop too soon/Applying too long

これで、HCF を導出する際に、都度コントロールストラクチャーや UCA 識別表に戻って確認する手間を省くことができる。

表 2.1-3　UCA 識別表

	コントロールアクション	Not Providing	Providing causes hazard	Too early / Too Late	Stop too soon / Applying too long
1	（踏切→検知装置）検知開始指示（踏切の動作開始通知）	（UCA1-N）検知開始指示が出ないので検知できないので発光せず　SC1-1 違反	踏切開状態で特殊信号発光機を発光する	（UCA1-T）Too Late で検知開始が遅れ、特殊信号発光機の発光が遅れるので検知できない時間がある SC1-1 違反 Too early で “ とりこ ” でない車を検知し発光指示する可能性あるがハザードにはならない	―
2	（検知装置→特殊信号発光機）発光指示	（UCA2-N）とりこがあっても発光せず列車を停止させない　SC1-1 違反	“ とりこ ” がないのに発光して列車を停止させる	（UCA2-T）Too late で発光開始が遅れ、列車が停止できない（ブレーキをかけるのが遅れる）　SC1-1 違反	―
3	（検知装置→特殊信号発光機）消灯指示	“ とりこ ” 解消しても特殊信号発光機消灯せず	（UCA3-P）“ とりこ ” 中に特殊信号発光機消灯 SC1-2 違反	（UCA3-T）Too early　同左	
4	（特殊信号発光機→乗務員）停止指示（特殊信号の発光）	（UCA4-N）“ とりこ ” があっても発光せず列車を停止しない　SC1-1 違反	“ とりこ ” がないのに発光し列車を停止させる	（UCA4-T）Too late で発光開始が遅れ、列車が停止できない（ブレーキをかけるのが遅い）　SC1-1 違反	―
5	（乗務士→列車）ブレーキ作動指示	（UCA5-N）運転士が特殊信号発光機の発光を認識できず列車を停止しない SC1-3 違反	“ とりこ ” がないのに停止する	**（UCA）Too late で列車停止が間に合わない（ブレーキをかけるのが遅い）→今回対象外とする**	**（UCA）Too soon で列車停止が間に合わない（ブレーキを途中で解除）→今回対象外とする**
6	（踏切→検知装置）検知停止指示（踏切の動作停止通知）	“ とりこ ” がないのに発光し列車を停止させる	（UCA6-P）列車が在線中に検知停止指示が出ると “ とりこ ” があっても発光せず列車を停止させない SC1-2 違反	（UCA6-T）Too early で “ とりこ ” があっても発光せず列車を停止させない　SC1-2 違反	―

2.1.6. STPA Step2：HCF（Hazard Causal Factor：ハザード誘発要因）の特定

ここでは、HCF 導出に当たり STPA の手順で示されているガイドワード（図 2.1-5）から該当するものを記載した上で 4 章に示した STPA 解析を実施する際のヒントワードの中から対応する人対人のヒントワードを利用してハザードシナリオを記入した。

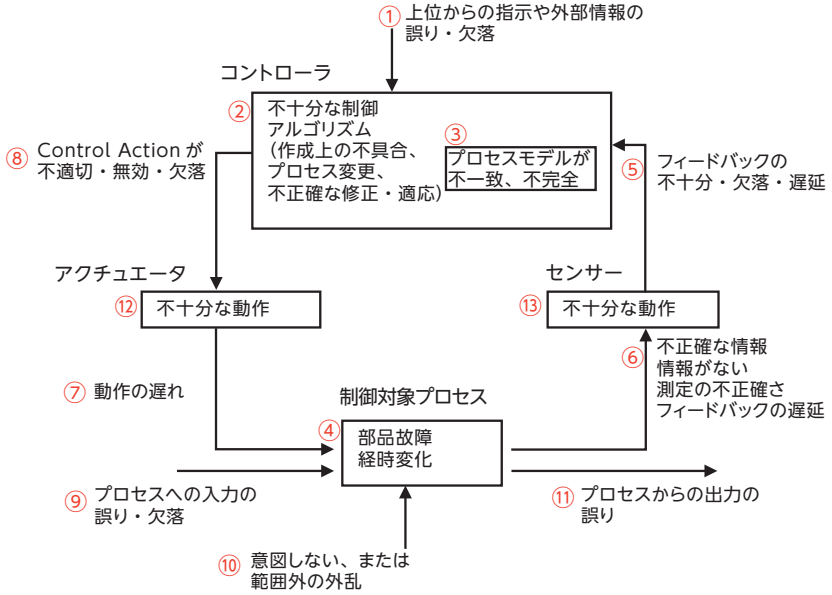


図 2.1-5　HCF 導出のためのガイドワード

以下、2.1.5 節で識別した UCA10 個についてハザードシナリオを示す。

（ｉ）ハザードシナリオにそれぞれ以下のような識別子を付けた。
HSn-N/R/T/D-m
n：CA の番号
N：Not Providing　　P：Providing causes hazard　　T：Timing Too early/Too late
D：Duration Stop too soon/Applying too long
m：UCA ごとに導出したハザードシナリオの連番
これにより、CA，UCA，HCF の間のトレーサビリティーを表すことができる。

（ii）UCA ごとにコントロールループ図を作成して STPA の HCF 導出のためのオリジナルの 13 個のガイドワードから該当するものをガイドワードの番号と合わせて記載し、そのガイドワードに対応するヒントワードを利用してハザードシナリオを、コントローラー、コントロールプロセスに記入した。

（a）UCA1-N に至るハザードシナリオ

（UCA1-N）：検知開始指示が出ないので検知できない（SC1-1 違反）

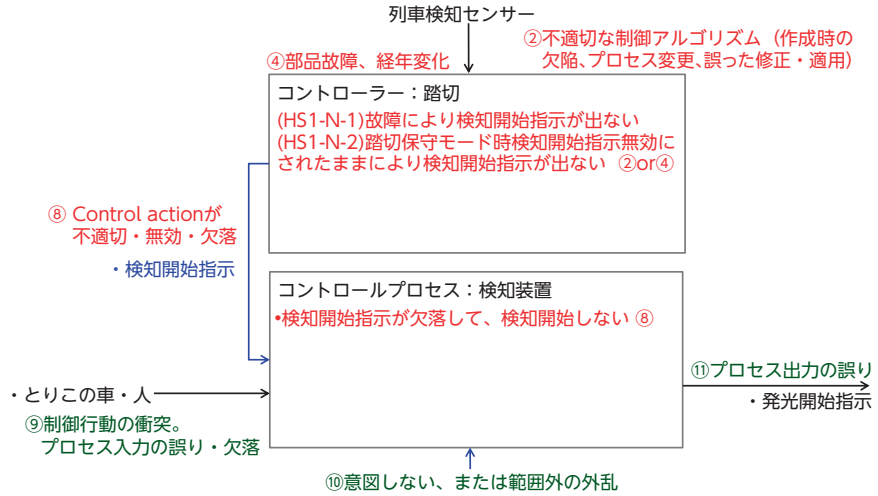


図 2.1-6　UCA1-N のコントロールループ図

（b）UCA1-T に至るハザードシナリオ

（UCA1-T）：Too Late で検知開始が遅れ、特殊信号発光機の発光が遅れるので検知できない時間がある（SC1-1 違反）

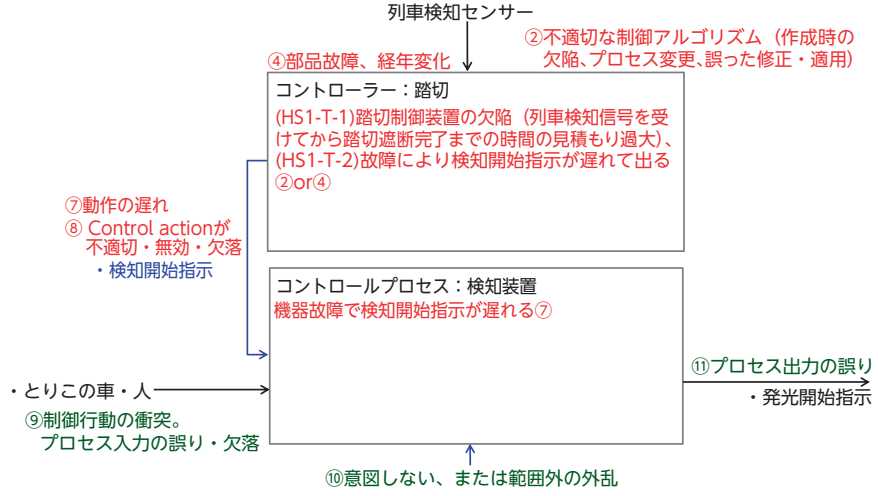


図 2.1-7　UCA1-T のコントロールループ図

（c）UCA2-N に至るハザードシナリオ

（UCA2-N）：“とりこ”があっても特殊信号発光機を発光せず列車が停止しない（SC1-2 違反）

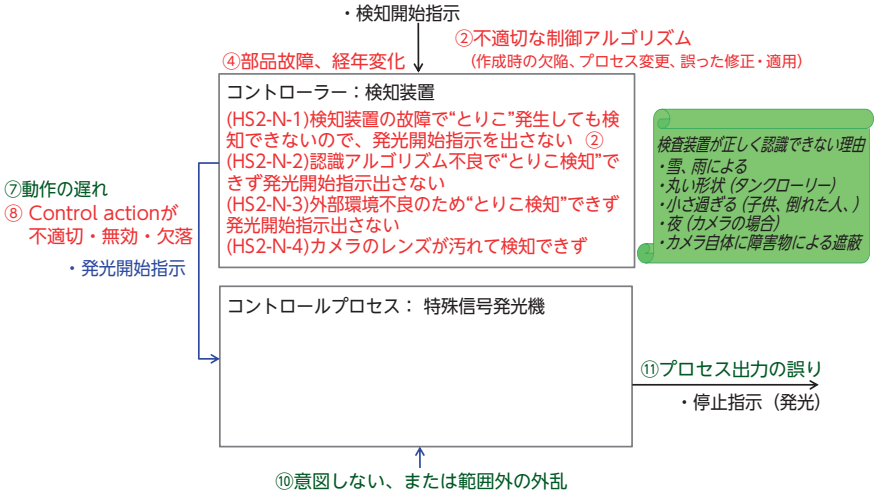


図 2.1-8　UCA2-N のコントロールループ図

（d）UCA2-T に至るハザードシナリオ

（UCA2-T）：Too late で特殊信号発光機の発光開始が遅れ、列車が停止できない（ブレーキをかけるのが遅れる）（SC1-1 違反）

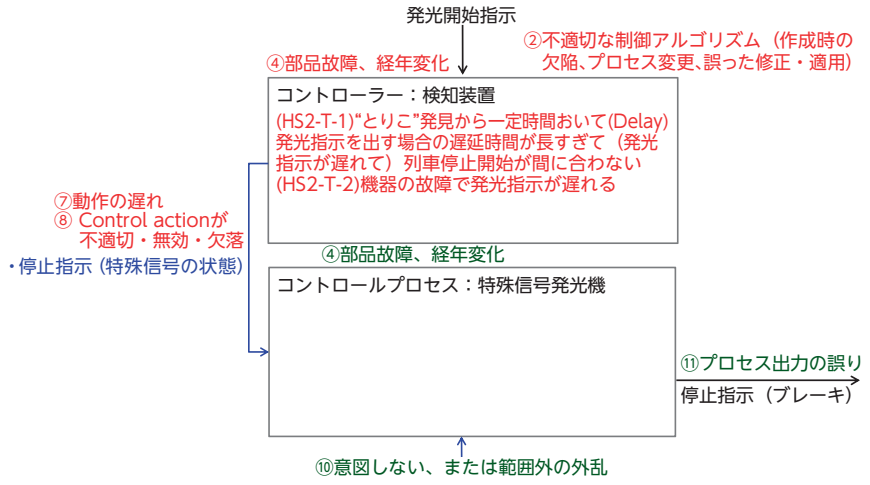


図 2.1-9　UCA2-T のコントロールループ図

(e) UCA3-P に至るハザードシナリオ

(UCA3-P):“とりこ”中に特殊信号発光機を消灯(SC1-2 違反)

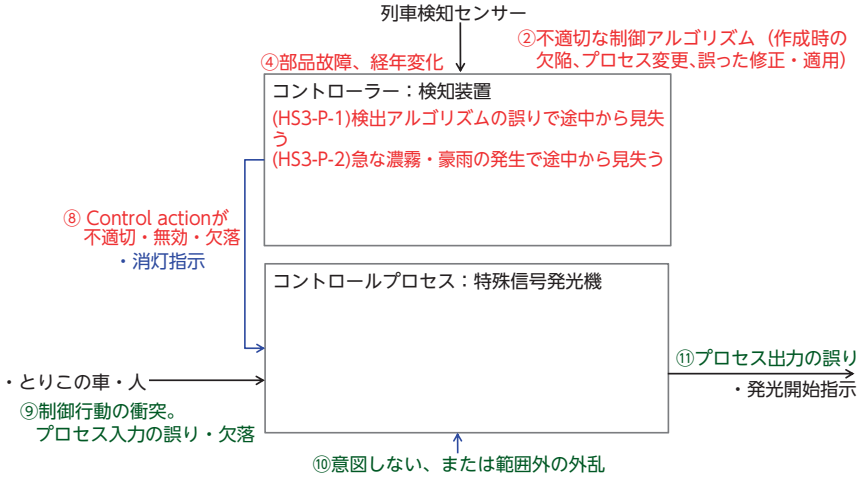


図 2.1-10　UCA3-P のコントロールループ図

(f) UCA4-N に至るハザードシナリオ

(UCA4-N):“とりこ”があっても特殊信号発光機が発光せず列車が停止しない(SC1-1 違反)

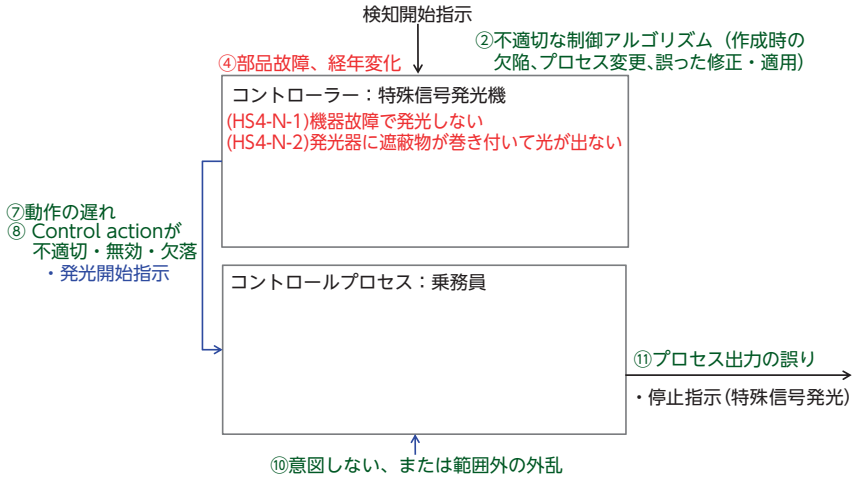


図 2.1-11　UCA4-N のコントロールループ図

（g）UCA4-T に至るハザードシナリオ

（UCA4-T）：Too late で特殊信号発光機の発光開始が遅れ、列車が停止できない（ブレーキをかけるのが遅い）（SC1-1 違反）

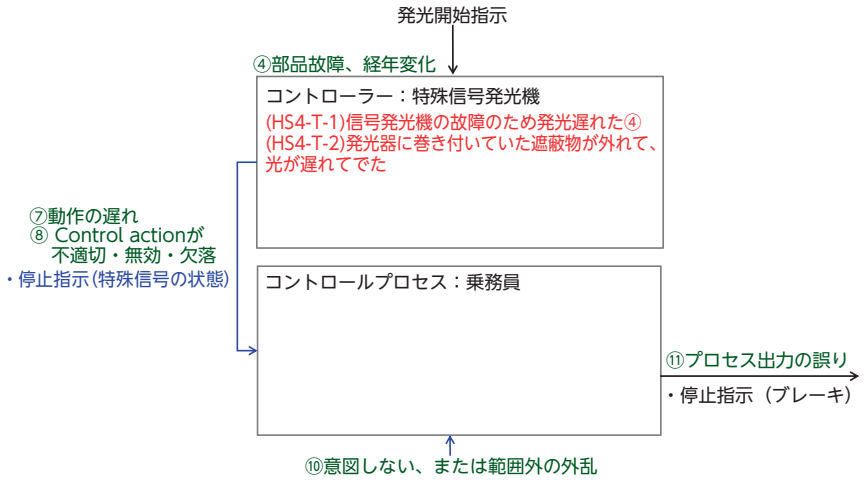


図 2.1-12　UCA4-T のコントロールループ図

（h）UCA5-N に至るハザードシナリオ

（UCA5-N）：運転士が特殊信号発光機の発光を認識できず列車を停止しない（SC1-3 違反）

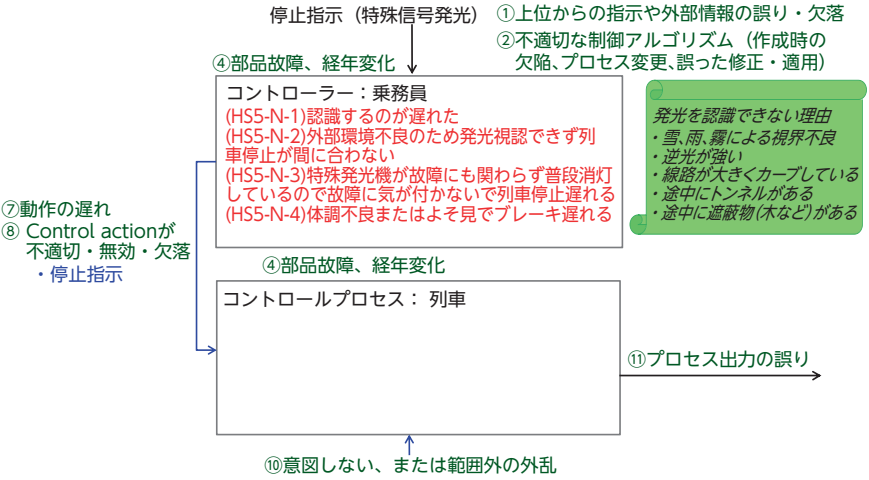


図 2.1-13　UCA5-N のコントロールループ図

(i) UCA6-P に至るハザードシナリオ

（UCA6-P）：列車が在線中に検知停止指示が出ると、“とりこ”があっても特殊信号発光機が発光せず列車を停止させない（SC1-2 違反）

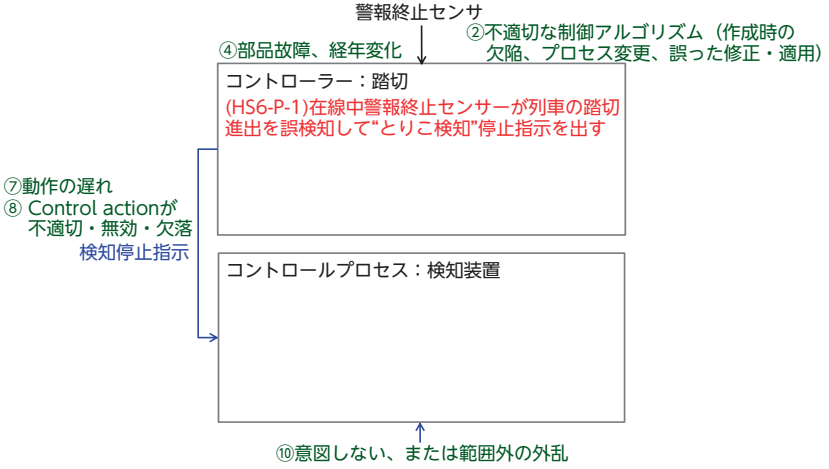


図 2.1-14　UCA6-P のコントロールループ図

(j) UCA6-T に至るハザードシナリオ

（UCA6-T）：Too early で“とりこ”があっても特殊信号発光機が発光せず列車を停止させない（SC1-2 違反）

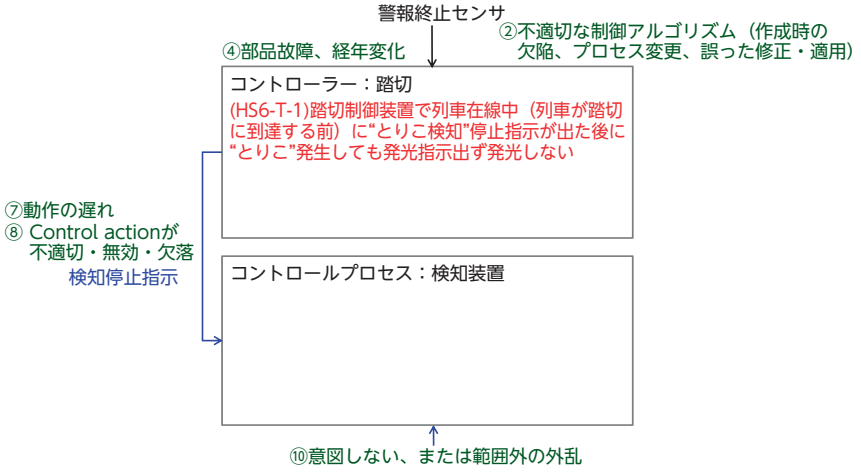


図 2.1-15　UCA6-T のコントロールループ図

2.1.7. 対策の立案

2.1.6 で導出した HCF の中から機能要件、非機能要件（性能）、運用要件に関わるものを設計制約として捉えて主要な対策の例を示した。

（a）機能（アルゴリズム）に関わる対策

（ｉ）制御機能に関する HCF

HCF：（HS1-T-1）検知開始指示は踏切装置が遮断完了してから出すと、とりこ検知して特殊信号発光機が発光しても列車停止まで間に合わない

対策：踏切が遮断完了までにかかる時間を δ、更にとりこ確定待ち時間を α、列車検知センサーから踏切までの距離を L、列車の最高許容速度を V_h、列車停止距離を Π（V）とすると、次の条件を満たさなければならない。

$$V_h \cdot (\delta + \alpha) + \Pi(V_h) < L$$

従って、踏切が遮断完了までにかかる時間、“とりこ”確定待ち時間、列車検知センサーから踏切までの距離、列車の最高許容速度の上限は上記制約式を満たすように決めなければならない。

（ii）認識機能に関する HCF

HCF：（HS2-N-2）認識アルゴリズム不良でとりこ検知できず発光開始指示出さない

対策：車を認識する場合、車の方向（前方 / 後方 / 斜め）、形状（乗用車、トラック、バス、コンテナ、自転車、バイク）、大きさを、車以外を認識する場合、種類（人、人以外の動物）、状態（起立、移動、転倒）、数などのバリエーションを考慮していなければならない。また、逆光、発光（とりこからの）など光に関する環境条件も考慮する必要がある。さらに、カメラの 2 台化（別角度からの像を合わせて検知）或いはカメラ以外の認識手段も考慮する必要がある。

（iii）検知機能に関する HCF

HCF：（HS2-T-2）とりこ状態確定判断するため一定時間待つ場合、発光開始指示が遅れ、列車停止間に合わない可能性がある。

対策：（ｉ）に含まれる。

（b）性能に関わる対策

HCF：（HS2-N-3）外部環境不良のためとりこ検知できず発光開始指示出さない

・雪、雨、霧による　・反射光強すぎ　・夜（カメラの場合）

対策：カメラを入力に使用する場合、入力光の量で制約が出るため感度、フィルタ、赤外線対応等も考慮する必要がある。評価用反射板を設置して環境不良をチェックすることも考慮する。

HCF：（HS4-N-2）発光器の発光部分に遮蔽物が巻き付いて光が出ず列車停止間に合わない

対策：風で飛ばされてくる可能性のあるもの（凧、ビラ、旗など）とその材質（紙、ビニール、布など）が巻き付きにくい形状にする。また、巻き付いて視界が遮られたことを判別して通知する機能の付加も考慮する必要がある。

（c）運用規約

HCF：（HS1-N-2）踏切制御装置保守のため検知機能無効状態のまま保守作業終了すると検知機能開始指示出ず、検知開始できない

対策：保守作業手順と規定、作業終了時点検方法を明確に規定する。また、保守作業完了後の運用開始スウィッチを踏切制御装置と連動する形で検知装置に設けることも考慮する。

（d）その他

HCF：（HS2-N-1）検知装置の故障で“とりこ”発生しても検知できないので、発光開始指示を出さない。

対策：検知装置から踏切制御装置へフィードバック機能を追加し検知装置からフィードバックがない場合に踏切制御装置から特殊信号発光機に発光指示を出すように機能追加する。

HCF：（HS4-T-2）特殊信号発光機の発光部分に巻き付いた遮蔽物が外れて光が遅れて出たが、列車停止間に合わない。

対策：前記（b）（HS4-N-2）と合わせて検討する

HCF：（HS2-N-2）外部環境不良のため発光視認できず列車停止が間に合わない。

- *雪、雨、霧による視界不良*
- *逆光強すぎ*
- *カーブがきつくて見えない*
- *途中の遮蔽物（木など）で見えない*
- *途中にトンネル*

対策 1：遮蔽物等に関しては定期的に保守（確認）することを運用管理作業に入れる。それ以外は設置場所の選定と通常の点検に含める。

対策 2：踏切を詳細化する前の 3 階層のコントロールストラクチャー（図 2.1-3）から考えると、踏切からのフィードバックが列車を経由しないで運転手に直接入っている。踏切からのフィードバックを列車に入れるようにすることで自動的に列車を停止させることが可能になる（図 2.1-16）。ただし、具体的な実現手段に相当のコストが必要になることは考慮しなければならない（すでに一部区間では実装されている）。この対策は図 2.1-4 の“とりこ検知”の流れに沿ったコントロールストラクチャーからは見つけるのが難しい。このことからモデルの抽象度によって考えられる対策の範囲に影響があることがわかる。

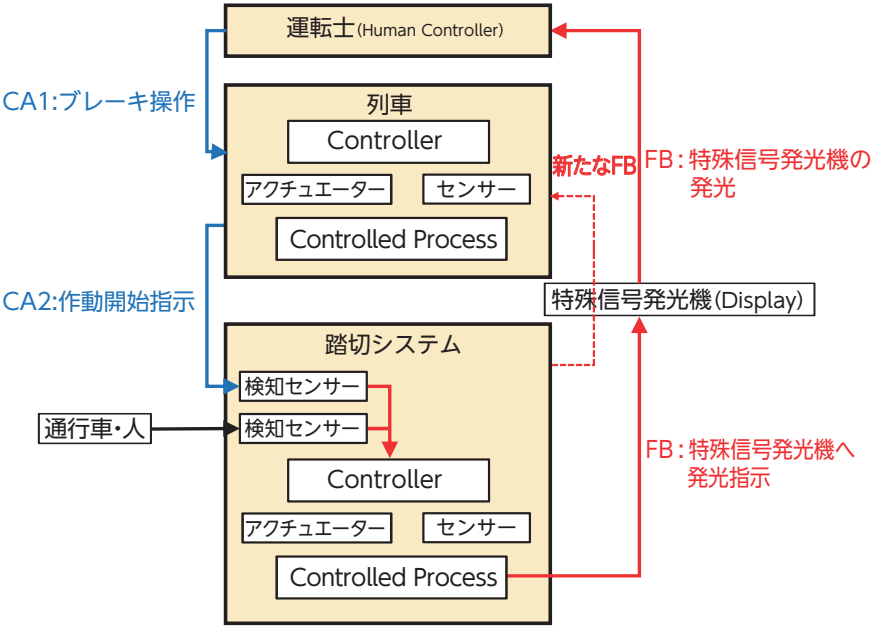


図 2.1-16　新たな FB を追加したコントロールストラクチャー

2.2. 組織と人間による業務の事例（鉄道踏切制御装置工事）

この章では、解析事例として鉄道における踏切制御装置の保守工事における安全対策のリスクを解析した。

2.2.1. 対象システム（業務）の概要

対象システム（業務）の登場人物は、

- 設計部門
- 施工部門（施工管理者、作業員、見張員）
- 指令部門（輸送指令、運転士、列車）
- 通行車・人
- 踏切制御装置

であり、それぞれの役割は表 2.2-1 の通りである。

表 2.2-1　登場人物の役割

	登場人物	役割（安全関連責任）	備考
1	設計部門	工事対象の踏切の制御論理、論理を実装する方法を設計し、施工部門に工事を指示する。	
2	施工部門	指令部門の了解のもと設計部門の指示内容に従って補修工事（結果確認を含む）を安全に実施する。	
2-1	施工管理者	万が一の列車進入に備えて見張り員を配置し、見張り開始終了を指示する。作業員に工事開始終了を指示する。	
2-2	見張り員	工事中列車進入の有無を監視し、発見時に作業員に待避指示し、作業員からの待避完了確認を待つ。待避完了を受け取ると列車を通過させ、確認がない場合列車を停止させる。	
2-3	作業員	踏切の補修工事を実施。列車進入時見張り員の指示に従って速やかに待避、待避完了後見張り員に待避完了確認を返す。	
3	指令部門	工事計画（工事許可申請）に従って当該工事区間の列車の運行を停止する。工事終了報告を受けると当該工事区間の列車の運行を再開する。	
3-1	輸送指令	工事計画（工事許可申請）に従って当該工事区間の列車の運転士に運行を停止させる。 工事終了報告を受けると当該工事区間の列車の運転士に運行を再開させる。	
3-2	運転士	輸送指令からの指示に従って担当列車を停止あるいは走行させる。 工事区間では、見張り員の指示に従って列車を停止あるいは走行させる。	
3-2	列車	運転士の指示に従って走行あるいは停止する。	

また、対象システム（業務）のイメージを図 2.2-1 に示す。

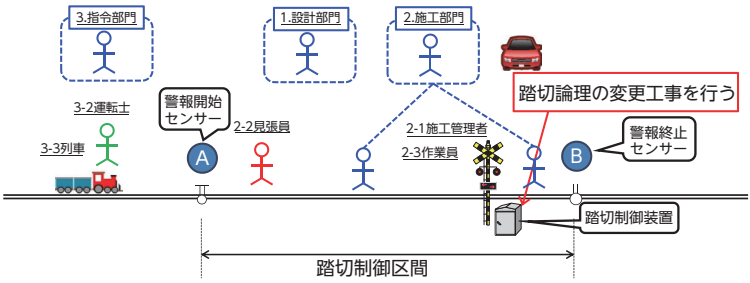


図 2.2-1　対象システム（業務）イメージ

以後、以下に示す手順に従って解析作業を進める。

■ 事前作業　　前提条件の整理
■ 準備 1　　アクシデント、ハザード、安全制約の識別
■ 準備 2　　コントロールストラクチャーの構築
■ STPA Step1　UCA（Unsafe Control Action：非安全制御動作）の識別
■ STPA Step2　HCF（Hazard Causal factor：ハザード誘発要因）の特定
■ 対策立案　　HCF を排除するための対策立案（設計上の安全制約）

2.2.2. 事前作業

本業務の安全性を解析するに当たって前提条件を以下のように整理した。

1. 工事開始に先立って、輸送指令（安全機能）に踏切制御区間の走行禁止指示を出してもらい、工事開始許可を受ける。
2. 工事開始前に手動操作により踏切制御装置の動作を停止するので、踏切は遮断したままになっている。
3. 踏切遮断している間、人・車が踏切内に進入することはない
4. 緊急車両は要請があれば通す。
5. 工事開始してから、工事終了し、最初の列車通過を確認するまでの時間を分析対象とする。
6. 工事には列車見張員（安全機能）が常駐する。
7. 工事作業者は、見張員→施工管理者からの待避指示に従って、自身と工事機材・車両が列車に衝突しない位置に待避する

尚、今回の安全性解析の目的は「工事が安全に遂行できるようにすること」とした。

2.2.3. 準備 1：アクシデント、ハザード、安全制約の識別

分析対象システムのアクシデントを識別し、そのアクシデントを防止するためにシステムに装備されている安全機能を整理する。

アクシデントは次の 3 つが考えられる。

- 工事作業者・車両・機材が列車と衝突する
- 緊急車両（消防車、救急車）が踏切を渡れず手遅れになる
- 通行者・車と工事関係者・車が衝突する

このアクシデントを防止し、工事を安全に遂行するために

- 輸送指令が工事区間の列車の通行を停止
- 見張員は、列車が来てしまった際の列車の発見と待避のための連絡
- 見張員は、緊急車両が来たときの対処のための連絡
- 施工管理者は輸送指令と見張員とコミュニケーションを取り作業員に指示

5 者がそれぞれミッション（安全機能）を持っている。

ハザードはこれらのミッション（安全機能）が適切に遂行されない状態であり、安全制約はその裏返しであることから以下のように識別できる（表 2.2 2）。

表 2.2-2　アクシデント・ハザード・安全制約一覧

アクシデント（Loss）	ハザード（Hazard）	安全制約（Safety Constraints）
（A1）工事作業者・車両・機材が列車と衝突する	（H1-1）工事中に列車が踏切制御区間に進入する	（SC1-1）工事中列車を踏切制御区間に進入させてはならない
	（H1-2）工事中に列車が踏切制御区間に進入した時に工事が中断（待避）されない	（SC1-2）工事中に列車進入したときは工事を中断（待避）させなければならない
	（H1-3）工事中に列車が踏切制御区間に進入した時に見張り員の指示に従って停止しない	（SC1-3）工事中に列車が踏切制御区間に進入してしまった時は見張り員の指示に従わなければならない
（A2）緊急車両（消防車､救急車）が踏切を渡れず手遅れになる	（H2-1）工事中、緊急車両が来ても踏切が遮断している	（SC2-1）工事中でも緊急車両が来たら踏切を開放しなければならない
（A3）通行者・車と工事関係者・車が衝突する	（H3-1）工事中、踏切が遮断していない	（SC3-1）工事中は踏切を遮断していなければならない

尚、今回は、赤線で囲った部分を解析対象とした。

2.2.4. 準備 2：コントロールストラクチャーの構築

鉄道の本来業務は列車を円滑かつ安全に運行させることであり、円滑に運行する責任を持っているのが指令部門であり、安全を確保する責任を持っているのが設計部門である。従って踏切工事は、施工部門が両部門からの指示（許可）を受けて実施される。これをコントロールストラクチャーに表現したものが、図 2.2-2 部門間のコントロールストラクチャーである。

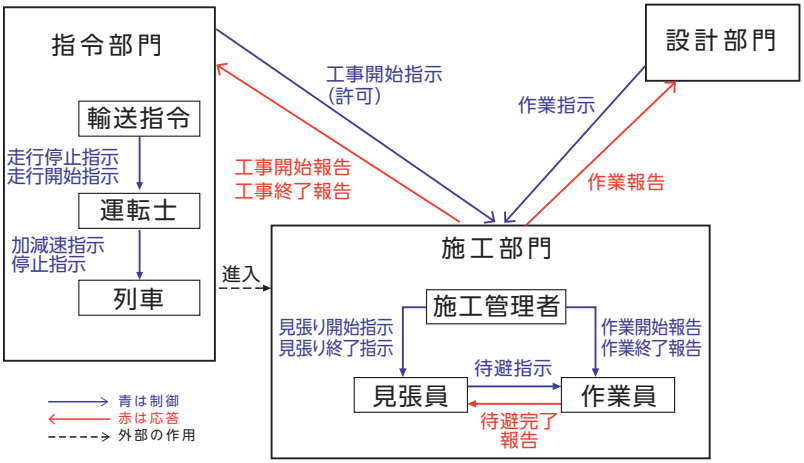


図 2.2-2　部門間のコントロールストラクチャー

一方、踏切工事の安全の責任を持つのは施工部門であり、工事中に列車が進入したとしても自らの安全を確保するための機能を有している。そこで施工部門を中心に制御アクションを正確に表現したものが、図 2.2-3 業務のコントロールストラクチャーである。ここでは、詳細に調査した結果、さらに見張員から運転士に対する走行許可・停止指示を加えている。また、見張員の列車発見が見張員の制御アクショントリガーになることから外部からの作用としている。

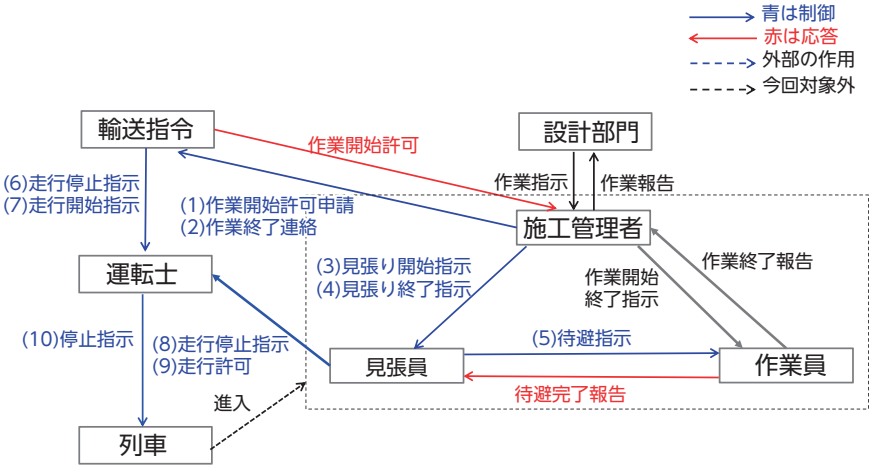


図 2.2-3　業務のコントロールストラクチャー

2.2.5. STPA Step1：UCA（Unsafe Control Action：非安全制御動作）の識別

ここでは施工部門を中心に作成したコントロールストラクチャー 2 のコントロールアクション（CA）全てに 4 つのガイドワードを適用して UCA を識別した（表 2.2-3 UCA 識別表）。

ここでは、コントロールアクションがどのコントローラーからどの制御対象プロセスに出ているかが容易に判別できるように "FROM"、"TO" の欄を設けるとともに、UCA にコントロールアクションの番号とどのガイドワードを適用したかがわかるように識別子を付けた。

UCAn-N/P/T/D

n：CA の番号

N：Not Providing　　P：Providing causes hazard　　T：Timing Too early/Too late

D：Duration Stop too soon/Applying too long

これで、HCF を導出する際に、都度コントロールストラクチャーや UCA 識別表に戻って確認する手間を省ける。

表 2.2-3　UCA 識別表

	コントロールアクション	FROM	TO	Not Providing	Incorrectly Providing	Too early / Too Late	Stop too soon / Applying too long
1	作業開始許可申請	施工管理者	指令部門	(UCA1-N) 作業開始連絡ないと運転停止指示が出ないので列車が進入する (SC1-1 違反)	列車の運転が停止されるだけなのでハザードにならない	(UCA1-T) Too late だと列車への運転停止指示が遅れるので列車が進入する (SC1-1 違反)	—
2	作業終了連絡	施工管理者	指令部門	終了連絡されなければ運転開始指示がでないのでハザードにならない	(UCA2-P) 終了連絡が作業中に出ると運転開始指示がでるのでハザード (SC1-1 違反)	(UCA2-T) Too early だと運転開始指示が早く出るのでハザード (SC1-1 違反)	—
3	見張り開始指示	施工管理者	見張員	(UCA3-N) 見張りをしていない時に列車が侵入するとハザード (SC1-2 違反)	工事していない時に見張りしてもハザードにならない	(UCA3-T) Too late だと列車進入するとハザード (SC1-2 違反)	—
4	見張り終了指示	施工管理者	見張員	工事していない時に見張りしてもハザードにならない	(UCA4-P) 工事中見張りを停止すると列車が侵入するとハザード (SC1-2 違反)	(UCA4-T) Too early だと列車進入するとハザード (SC1-2 違反)	—
5	待避指示	見張員	作業員	(UCA5-N) 列車が侵入しても待避指示されずハザード (SC1-2 違反)	工事が中断するだけ	(UCA5-T) Too late だと待避間に合わずハザード (SC1-2 違反)	—
6	走行停止指示	指令部門	運転士	(UCA6-N) 見張りをしていない時に列車が侵入するとハザード (SC1-1 違反)	列車が停止するだけなのでハザードにならない	(UCA6-T) Too late だと運転停止指示が遅れるのでハザード (SC1-1 違反)	—
7	走行開始指示	指令部門	運転士	列車が停止するだけなのでハザードにならない	(UCA7-P) 見張り員いない時に列車が侵入するとハザード(SC1-1 違反)	(UCA7-T) Too early だと列車への運転開始指示が早く出るので列車が進入する (SC1-1 違反)	—
8	走行停止指示	見張員	運転士	(UCA8-N) 列車が踏切に進入するのでハザード (SC1-3 違反)	列車が停止するだけなのでハザードにならない	(UCA8-T) Too late だと列車進入するとハザード (SC1-3 違反)	—
9	走行許可	見張員	運転士	列車が停止するだけなのでハザードにならない	(UCA9-P) 列車が踏切に進入するのでハザード (SC1-3 違反)	(UCA9-T) Too early だと列車進入するとハザード (SC1-3 違反)	—
10	停止指示	運転士	列車	(UCA10-N) 列車が停止せずに踏切に進入するのでハザード (SC1-3 違反)	列車が停止するだけなのでハザードにならない	(UCA10-T) Too late だと列車進入するとハザード (SC1-3 違反)	—

2.2.6. STPA Step2：HCF（Hazard Causal Factor：誘発要因）の特定

ここでは、HCF 導出に当たり STPA の手順で示されているガイドワード（図 2.2-4）から該当するものを記載した上で 4 章に示した STPA 解析を実施する際のヒントワードの中から対応する人対人のヒントワードを利用してハザードシナリオを記入した。

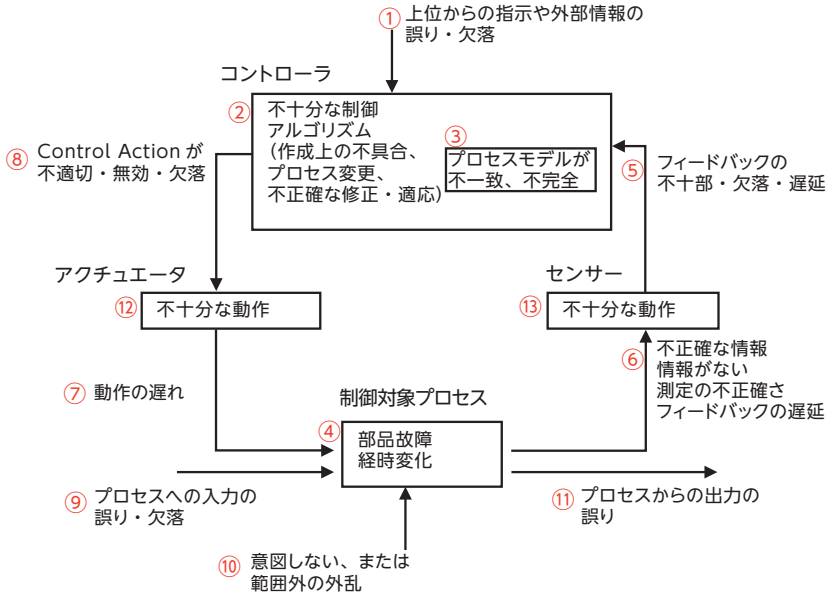


図 2.2-4　HCF 導出のためのガイドワード

以下、2.2.5 項で識別した UCA16 個についてハザードシナリオを示す。

（ｉ）ハザードシナリオにそれぞれ以下のような識別子を付けた。
HSn-N/R/T/D-m
n：CA の番号
N：Not Providing　P：Providing causes hazard　T：Timing　Too early/Too late
D：Duration　Stop too soon/Applying too long
m：UCA ごとに導出したハザードシナリオの連番
これにより、CA，UCA，HCF の間のトレーサビリティーを表すことができる。

（ii）UCA ごとにコントロールループ図を作成して STPA の HCF 導出のためのオリジナルの 13 個のガイドワードから該当するものをガイドワードの番号と合わせて記載し、そのガイドワードに対応するヒントワードを利用してハザードシナリオを、コントローラー、コントロールプロセスに記入した。

（a）UCA1-N、UCA1-T に至るハザードシナリオ

（UCA1-N）：作業開始連絡ないと運転停止指示が出ないので列車が進入する（SC1-1 違反）

（UCA1-T）：作業開始連絡が Too late だと列車への運転停止指示が遅れるので列車が進入する（SC1-1 違反）

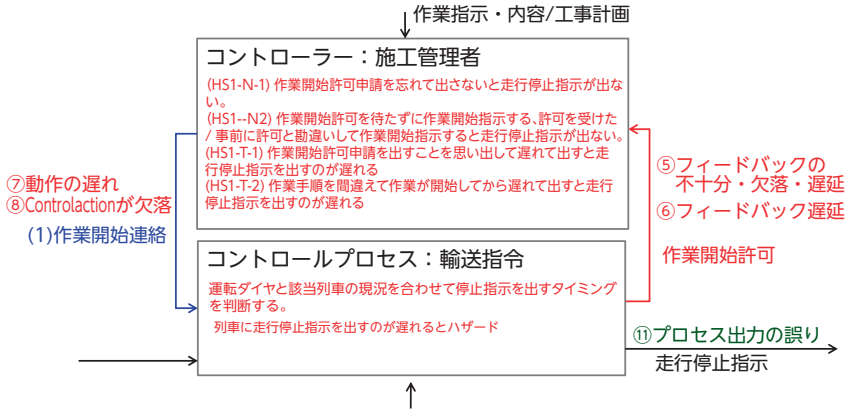


図 2.2-5　UCA1-N/UCA1-T のコントロールループ図

（b）（UCA2-P）、（UCA2-T）に至るハザードシナリオ

（UCA2-P）：終了連絡が作業中に出ると運転開始指示がでるのでハザード（SC1-1 違反）

（UCA2-T）：終了連絡が Too early だと運転開始指示が早く出るのでハザード（SC1-1 違反）

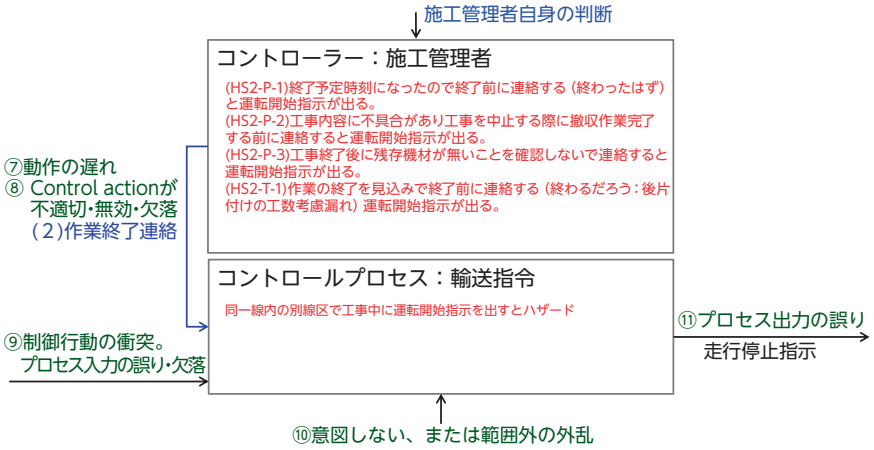


図 2.2-6　UCA2-P/UCA2-T のコントロールループ図

（c）（UCA3-N）、（UCA3-T）に至るハザードシナリオ

（UCA3-N）：見張り開始指示が出なくて見張りをしていない時に列車が進入するとハザード（SC1-2 違反）

（UCA3-T）：見張り開始指示が Too late だと列車進入するとハザード（SC1-2 違反）

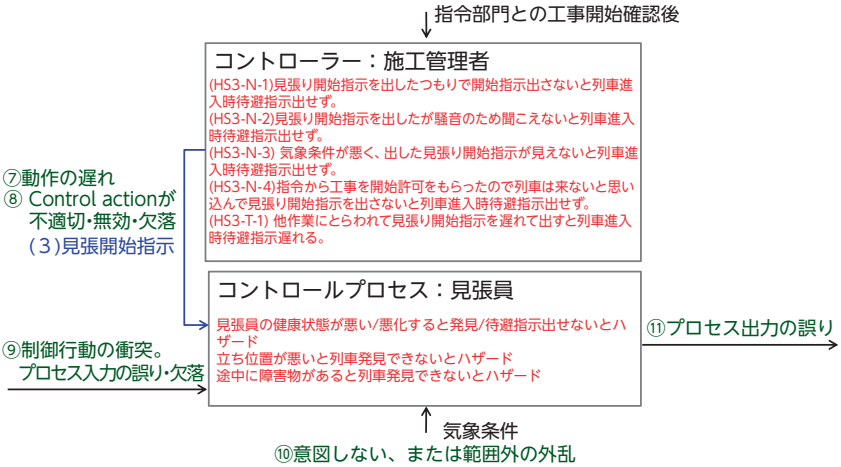


図 2.2-7　UCA3-N/UCA3-T のコントロールループ図

黄色で示したフィードバックは、施工管理者に対して作業者に向けた避難指示を要求することになるので、コントロールアクションとして見直した。

（d）（UCA4-P）、（UCA4-T）に至るハザードシナリオ

（UCA4-P）：見張終了指示が工事中に出ると列車が進入するとハザード（SC1-2 違反）

（UCA4-T）：見張終了指示が Too early だと列車進入するとハザード（SC1-2 違反）

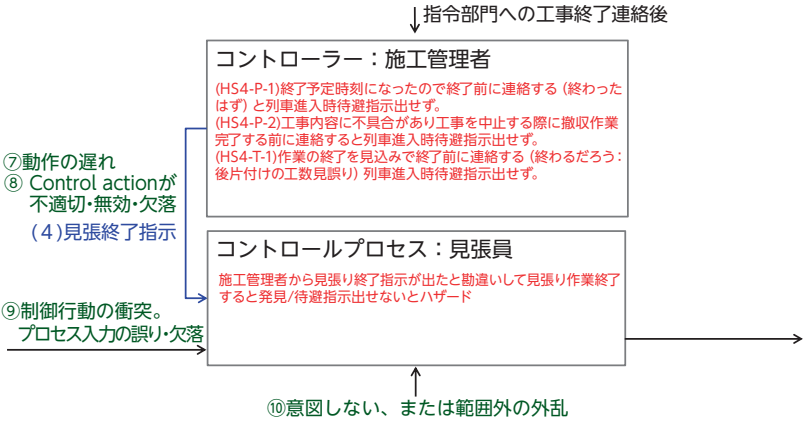


図 2.2-8　UCA4-P/UCA4-T のコントロールループ図

（e）（UCA5-N）、（UCA5-T）に至るハザードシナリオ

（UCA5-N）：列車が進入しても報告されずハザード（SC1-2 違反）
（UCA5-T）：Too late だと待避間に合わずハザード（SC1-2 違反）

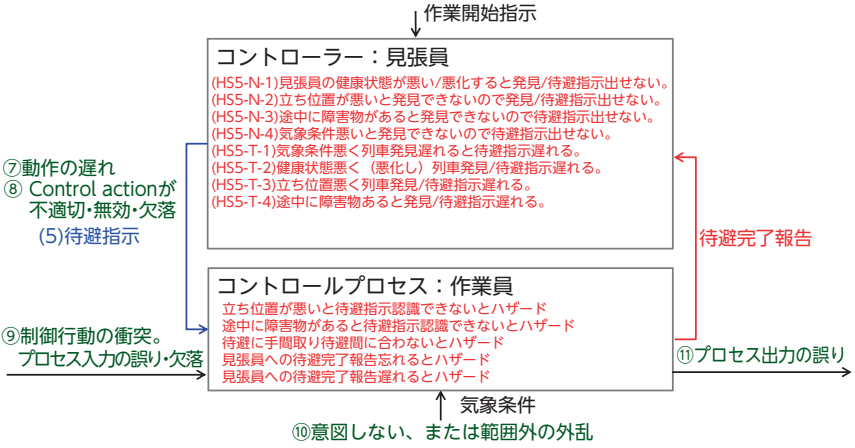


図 2.2-9　UCA5-N/UCA5-T のコントロールループ図

（f）（UCA6-N）、（UCA6-T）に至るハザードシナリオ

（UCA6-N）：走行停止指示（輸送指令より）が出ないと、見張りをしていない時に列車が進入しハザード（SC1-1 違反）
（UCA6-T）：走行停止指示（輸送指令より）が Too late だと、列車への運転停止指示が遅れるので列車が進入しハザード（**SC1-1 違反**）

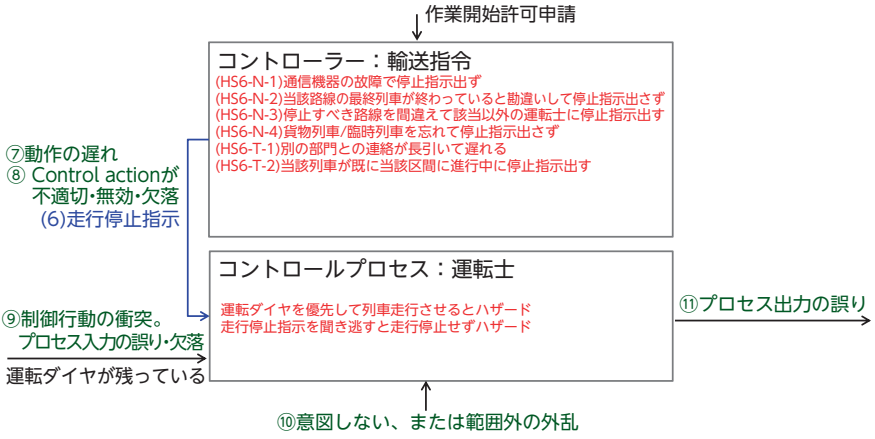


図 2.2-10　UCA6-N/UCA6-T のコントロールループ図

（g）（UCA7-P）、（UCA7-T）に至るハザードシナリオ

（UCA7-P）：走行開始指示が間違って出て工事中見張りをしていない時に列車が進入しハザード（SC1-1 違反）

（UCA7-T）：走行開始指示が Too early だと運転開始指示が早く出るのでハザード（SC1-1 違反）

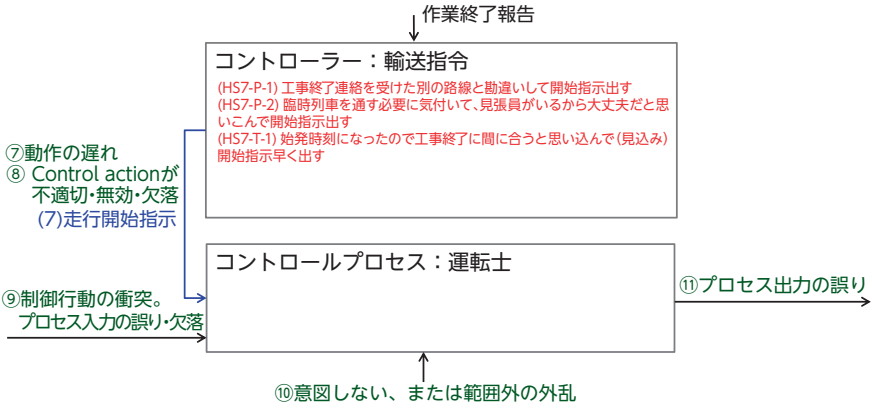


図 2.2-11　UCA7-P/UCA7-T のコントロールループ図

（h）（UCA8-N）、（UCA8-T）に至るハザードシナリオ

（UCA8-N）：走行停止指示（見張員より）が出ないと列車が踏切に進入するのでハザード（SC1-3 違反）

（UCA8-T）：走行停止指示（見張員より）が Too late だと列車への停止指示が遅れ列車が踏切に進入するのでハザード（SC1-3 違反）

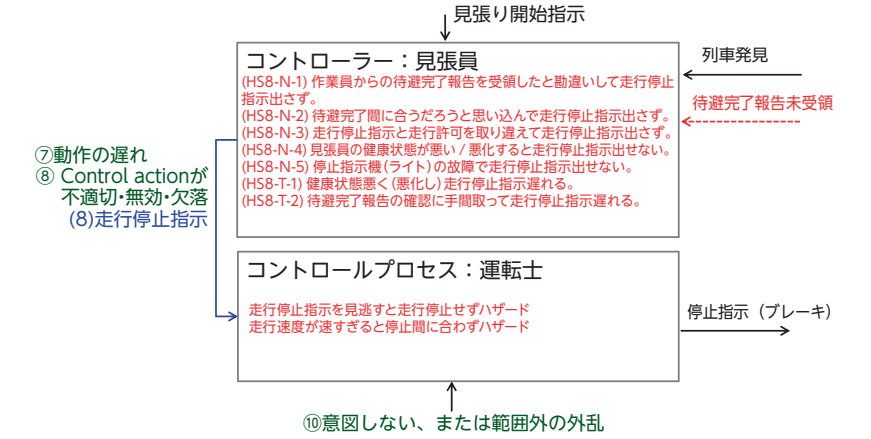


図 2.2-12　UCA8-N/UCA8-T のコントロールループ図

(i)（UCA9-P)、(UCA9-T）に至るハザードシナリオ

（UCA9-P）：走行許可（見張員より）を出すと列車が踏切に進入するのでハザード（SC1-3 違反）

（UCA9-T）：走行許可（見張員より）が Too early だと待避完了前に列車が踏切に進入するのでハザード（SC1-3 違反）

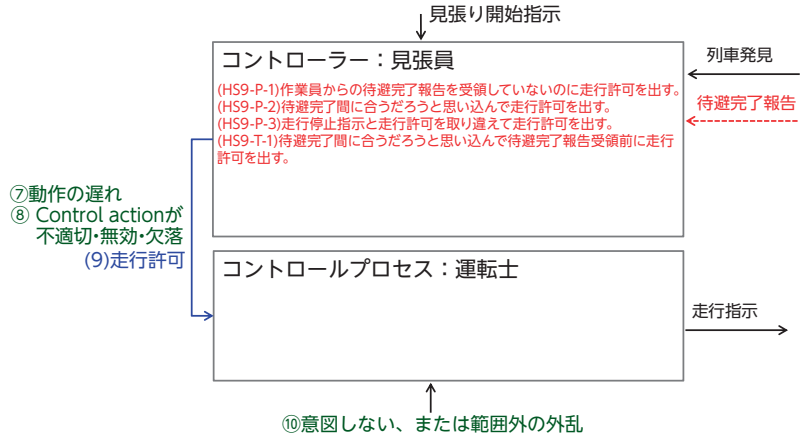


図 2.2-13　UCA9-P/UCA9-T のコントロールループ図

(j)（UCA10-N)、(UCA10-T）に至るハザードシナリオ

（UCA10-N）：運転士が停止指示（ブレーキ）を出さないと列車が踏切に進入するのでハザード（SC1-3 違反）

（UCA10-T）：停止指示（ブレーキ）が Too late だと列車が踏切前で停止できないで進入するのでハザード（SC1-3 違反）

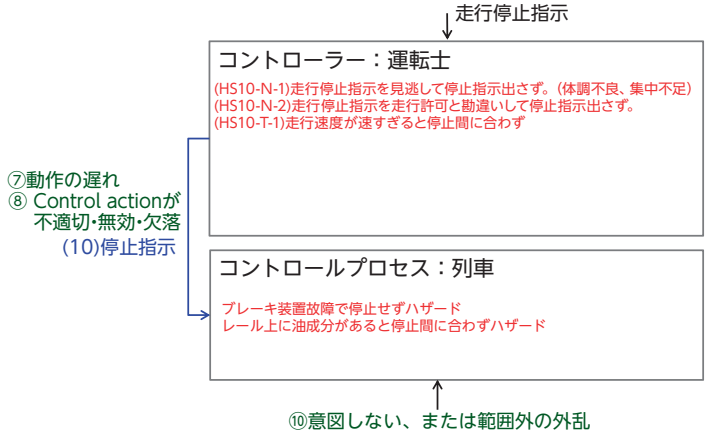


図 2.2-14　UCA10-N/UCA10-T のコントロールループ図

2.2.7. まとめ

今回の解析で多かった HCF は、「思い込み、失念、見込み」であった。中でも「見込み」による早すぎる指示・報告が最も多かった。これは、背景に焦りを誘発する原因がある場合によく見られる現象であることから、工事に許されている時間が短い、夜間等人間の注意力が低下しやすい状況での作業スケジュールへの影響も考えられる。もしそうであるとした場合は、スケジューリング基準の見直しも必要になるかもしれない。

表 2.2-4　特定された“人に関する HCF”一覧

Not Providing（指示が出ない）	件数
（HC1）指示が必要と思っていない	1
（HC2）指示を知っていたが忘れる	1
（HC3）指示したつもり	1
（HC4）指示・フィードバックを見逃して操作をしない	1
Providing causes hazard（間違った指示、遅れた・早すぎる指示）	
（HC5）指示内容を間違える	0
（HC6）思い出す・手間取って遅れる	6
（HC7）指示内容を勘違い（取り違える）	5
（HC8）違う相手に指示を出す	1
（HC9）フィードバックを誤解して間違った操作を行う	3
（HC10）確認せずに見込みで指示を出す	12
オミッションエラー	
（HP1）指示が来たが受け取らない	2
（HP2）指示が来たがスキル不足で実施できない	0
（HP3）実行結果のフィードバックを忘れる	1
コミッションエラー	
（HP4）指示を誤解して実行する	2
（HP5）指示どおりの実行ができないまたは遅れる（不適切な環境、スキル不足、健康状態不良）	8
（HP6）思い出す・手間取って遅れる	1

3. STPA 活用事例解説　エンタープライズ系システム解析事例

本章では、エンタープライズ系システムに対して STAMP/STPA 分析を試行した例を示す。分析対象例題には、読者の多くが利用したことがあり馴染みが深いと考え、ネット通販システムを取り上げた。

3.1. 本試行の目的

STAMP では、アクシデントとは「損失」につながるような意図せざる事象のこととされる。「損失」の定義には人命の損失や怪我だけでなく、ミッションの未達や経済的な損失、財産や環境のダメージなども含まれており、広範囲の問題に適用できるように意図されている [Leveson2013]。

エンタープライズ系システムは、銀行や証券会社などの金融システム、各種手続きのための行政情報システム、電話・メール・放送等の情報通信システム、道路・鉄道・航空等の交通管理システムなど国民の安全な生活を支えるインフラであり、サービスが停止した場合の影響は非常に大きい [IPA2016]。したがって、サービスの停止をアクシデントとした STAMP/STPA 分析によってその危険性を下げるように備えることが可能であれば大きな恩恵が得られる。

一方で、制御系システムに対する STAMP/STPA 分析の事例は数多く見られるが、エンタープライズ系システムを対象とした分析は参考にできる事例が少ない。そこで、以下を明らかにする目的で、本章に示す例題を用いた STAMP/STPA 分析を行った。

- エンタープライズ系システムに対して STAMP/STPA 分析の手法は適用可能か
- STAMP/STPA 分析をエンタープライズ系システムに対して適用する場合に特有の注意点やコツはあるか

3.2. 分析対象の例題「ネット通販システム」

利用者からの注文を電子的なネットワーク通信によって受け、注文された商品を利用者に配送する一般的なネット通販業務を想定する。この業務を遂行するために必要な機能を表 3.2-1 に示すように定義した。なお、試行の例題として簡単化するために、以下のように業務内容を限定した。

- 代金の決済などの金銭授受に関する業務は含めない
- 商品の入荷に関する業務は含めない
- 利用者が注文した後のキャンセルや変更は無いものとする

表 3.2-1　分析対象とするネット通販業務の仕様

機能	業務内容
販売管理機能	●利用者からの注文を受けると、在庫管理機能に引当てを指示する。 ●在庫管理機能による引当て可否の応答が「引当て可」であれば受注ステータスを「確定」とし、「引当て不可」であれば受注ステータスを「不可」とする。 ●受注の確定／不可を利用者に通知する。 ●受注ステータスが確定となった場合は、出荷機能に対して出荷指示を発行する。
在庫管理機能	●商品の在庫データ（総在庫数、引当済み数）を管理する。 ●引当て指示を受けると、在庫データを確認し、引当て可能な在庫数（総在庫数－引当済み数）が注文数以上であれば「引当て可」を、そうでなければ「引当て不可」を応答する。 ●在庫データを更新する（引当てた数量を引当済み数に加える）。
出荷機能	●出荷指示を受けると、指示された商品をピックアップし、配送機能に引き渡す（出荷する）。 ●出荷した後、在庫データを更新する（出荷した数量を総在庫数と引当済み数から減ずる）。
配送機能	●出荷機能から商品を受け取り、指示された宛先に配送する。

表 3.2-1 に示した業務の流れを、エンタープライズ系システムの仕様記述でよく使われる UML アクティビティ図を用いて示す（図 3.2-1）。この図では、アクティビティ図の記法に則った表現に加え、データの CRUD（Create（生成）、Read（参照）、Update（更新）、Delete（削除））に関する情報も表記している。これは、エンタープライズ系システムはデータの入力・加工・参照・出力などの処理を行うシステムであり、データの依存関係を示すことによって各機能や処理の相互関係を分かりやすくすることを意図したものである。

なお、利用者は複数存在し、複数の注文は並行に処理されることを想定している。

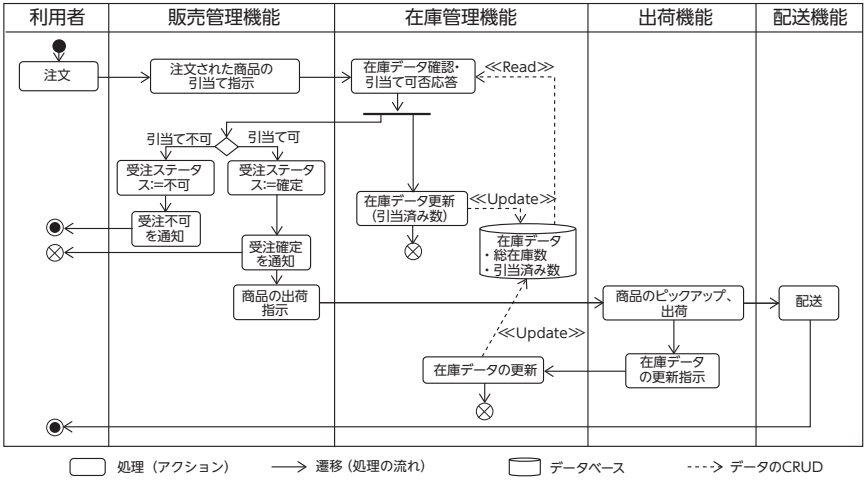


図 3.2-1　ネット通販業務の流れを表すアクティビティ図

3.2.1. 試行における前提

3.2.1.1. 電子システム化（自動化）に関する前提

表 3.2-1 および図 3.2-1 に示したネット通販業務の各機能が自動処理されるか人手によって処理されるかは、ハザードの可能性や原因に関わるため、明確になっている必要がある。今回の分析では、電子システム化（自動化）の範囲について以下の前提を置いた。

- 販売管理機能と在庫管理機能は電子システム化され、自動処理される
- 出荷機能と配送機能は、人手によって処理される

3.2.1.2. 分析対象範囲

STAMP/STPA 分析の対象は、表 3.2 1 に示した販売管理、在庫管理、出荷、配送の各機能とし、利用者の振舞いは分析対象外とした。

3.3. STAMP/STPA 分析

STAMP/STPA の提唱者である Nancy Leveson 教授らによるチュートリアル “An STPA Primer”[Leveson2013] および IPA より公開されているガイド「はじめての STAMP/STPA」[IPA2016] に示される手順に沿って分析を行った。また、第 1 章に示したとおり、安全性解析の目的はハザード誘発要因を排除したシステム設計を行うことであるため、本試行でもハザード誘発要因を排除するための方策の検討まで行った。以下に、詳細な作業内容と結果を示す。

3.3.1. Step0 準備 1 アクシデント、ハザード、安全制約の識別

3.1 節に示したように、期待されるサービスが遂行されないことをアクシデントとみなして分析を行った。今回の試行では、「注文した商品が利用者に配送されない」ことをアクシデントとした。

ハザードは、最悪の条件と重なることでアクシデントにつながるようなシステムの状態のことであり [IPA2016]、今回の分析では「受注ステータスが確定と決定された注文に対して、商品が出荷されていない状態」をハザードとした。

安全制約は、ハザード状態を起こさないことであり、「受注ステータスが確定と決定された注文に対して、速やかに商品が出荷されなければならない」となる。

以上を表 3.3 1 にまとめる。

表 3.3-1　アクシデント、ハザード、安全制約の識別

アクシデント	ハザード	安全制約
注文した商品が、利用者に配送されない	受注ステータスが確定と決定された注文に対して、商品が出荷されていない状態	受注ステータスが確定と決定された注文に対して、速やかに、商品が出荷されなければならない

3.3.2. Step0 準備 2 コントロールストラクチャーの構築

「モデル化」とは、物事の本質を深く理解することを目的として、興味のある要素を残して対象を大幅に簡略化することである。モデル化の巧拙が対象システムの理解のしかたに大きく影響するため、STAMP/STPA で使用するモデルであるコントロールストラクチャーの構築は非常に重要となる。

モデル化のスキルは属人性が高く、ノウハウを明文化することも困難であるが、今回の試行では図 3.2-1 に示すアクティビティ図に基づいてコントロールストラクチャーを作成する手順を見出すことを試みた。

今回試みたのは、以下の手順に基づいてコントロールストラクチャーの構成要素である「コンポーネント」「コントロールアクション」「フィードバックデータ」を浮き彫りにし、それらを用いてコントロールストラクチャーを作成することである。

（1）アクティビティ図から、安全制約に関係する「処理」を抽出する。
（2）抽出された処理のうち、その処理から発する矢印（遷移）が機能間をまたぐものについて「コントロールアクションを発行する処理」か「フィードバックデータを発行する処理」のいずれかに該当するか否かを判定する。
（3）コントロールアクションに該当する矢印（遷移）の矢元と矢先の機能を「コンポーネント」とする。
（4）上記によって得たコンポーネント、コントロールアクション、フィードバックデータを用いてコントロールストラクチャーを構築する。

この手順を今回の例題に適用した様子を以下に示す。

図 3.3-1 は、安全制約に関係する「処理」を抽出した結果である（安全制約に関係する処理を青色で示している）。安全制約は「受注ステータスが確定と決定された注文に対して、速やかに、商品が出荷されなければならない」であるので、受注ステータスの決定と商品の出荷に関わる処理を選択した。

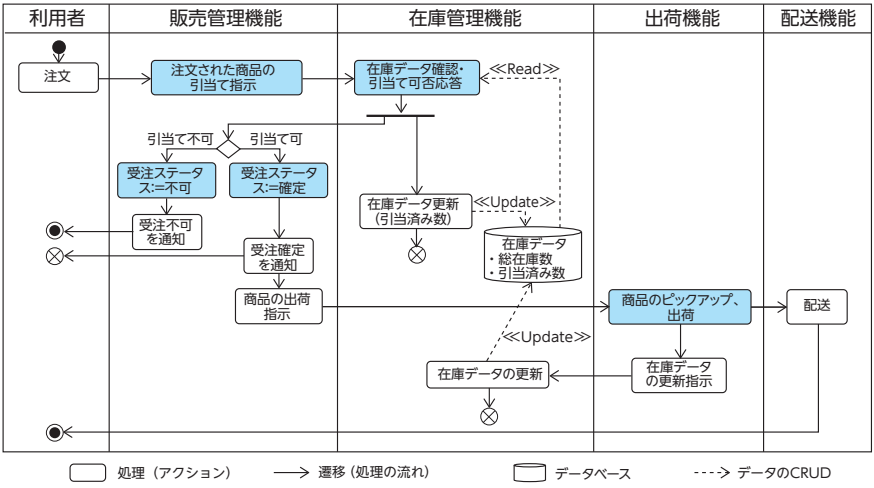


図 3.3-1　安全制約に関係する処理を抽出した結果

アクティビティ図において、機能間をまたぐ矢印は、機能間の何らかの相互関係を表している可能性がある。例えば、「注文された商品の引当て指示」処理から「在庫データ確認・引当て可否応答」処理に向かう矢印は、「販売管理機能」から「在庫管理機能」に対して商品の引当てを指示しているコントロールアクションとみなせる。また、「在庫データ確認・引当て可否応答」処理から「販売管理機能」に向かう矢印は、引当て指示に対する結果のフィードバックである。

このような考えに基づき、図 3.3-1 の青色で示した処理から発する矢印のうち、機能間をまたがるものについて、コントロールアクションまたはフィードバックデータに相当するかどうかを判断した。その結果を図 3.3-2 に示す。赤色の矢印はコントロールアクションに相当し、青色の矢印はフィードバックデータに相当することを表している。

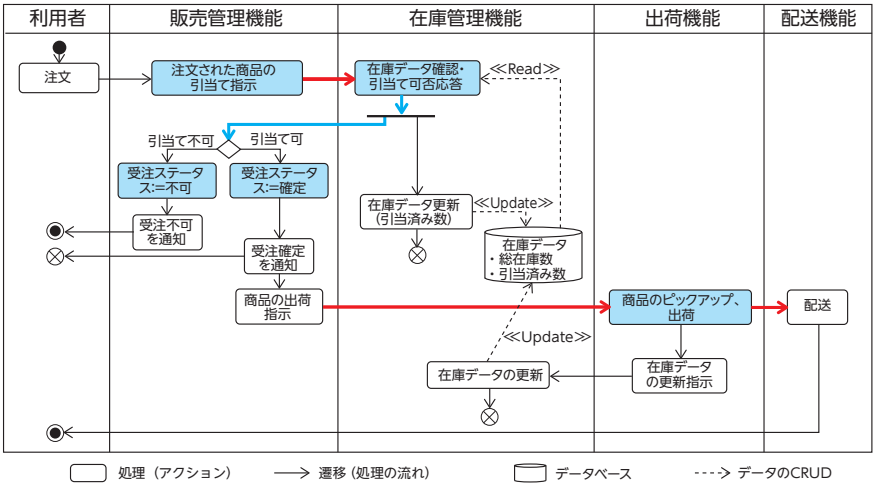


図 3.3-2　コントロールアクションとフィードバックデータの識別

コントロールアクションに相当する矢印は、矢元の機能がコントローラーを表し、矢先の機能が被コントロールプロセスを表すと考えられる。図 3.3-2 に示したコントロールアクションに相当する矢印から、「販売管理機能」、「在庫管理機能」、「出荷機能」、「配送機能」をコンポーネントとした。

ここまでの作業により、コンポーネント、コントロールアクション、フィードバックデータのそれぞれに相当するものが得られる。これらを用いて、図 3.3-3 に示すようなコントロールストラクチャーを構築した。また、各コントロールアクションの発行に関わる情報をプロセスモデル変数としてコントロールストラクチャー上に示した。

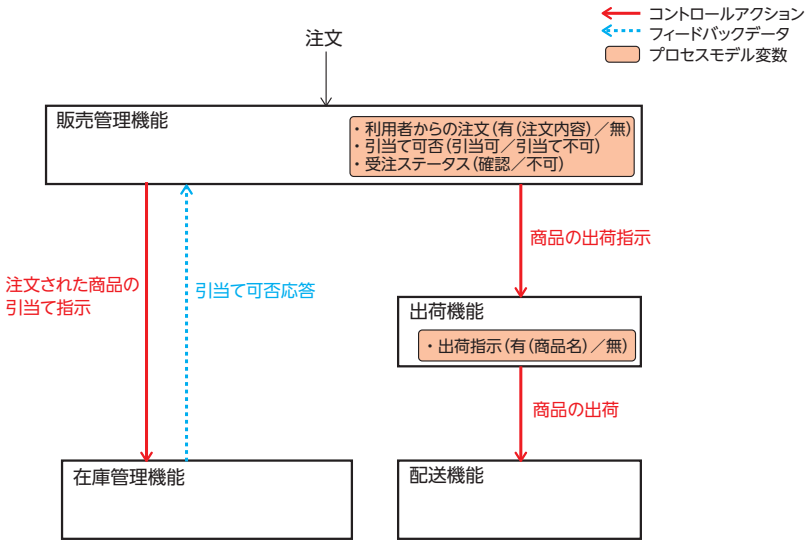


図 3.3-3　構築したコントロールストラクチャー

3.3.3. Step1 UCA（Unsafe Control Action）の抽出

各コントロールアクションについて、[IPA2016] に示される 4 つのガイドワードをヒントにして非安全なコントロールアクション（UCA）を抽出した。その結果を表 3.3-2 に示す。

3.3.1 項で識別した安全制約「受注ステータスが確定と決定された注文に対して、速やかに、商品が出荷されなければならない」に反する可能性がある状況として、UCA1-N から UCA3-T まで 7 つの UCA を抽出した。

なお、[IPA2016] に示されている通り、UCA の抽出は人の発想を活用して行う。したがって、表 3.3-2 に示したものはあくまでも一例であることに注意されたい。例えば、UCA2-P は、「商品の出荷指示」が発行されてハザードになる場合を考えた一例である。どのような場合にハザードにつながるかは多くの可能性がある。UCA2-P は、「何らかの理由で出荷可能な商品が足りないときに『商品の出荷指示』が出てしまう」状況を考えた。「出荷可能な商品の数」は、システムの挙動に影響を与える情報（環境の状態を表す変数）であり、UCA2-P はこのような変数に係わる状況を考えて抽出したものである。

なお、ガイドワードのうち、"Stopping too soon/Applying too long" に該当する UCA は抽出されなかった。これは、適用される時間の長さによって効果が変わるようなコントロールアクションがなかったためである。エンタープライズ系システムでは、離散的なアクションが多いため、"Stopping too soon/Applying too long" に該当する UCA はほとんどないと考えられる。

表 3.3-2　UCA の抽出結果

コントロールアクション		ガイドワード			
		Not providing causes hazard	Providing causes hazard	Too early, Too late, Wrong order causes hazard	Stopping too soon/ Applying too long
1	注文された商品の引当て指示	（UCA1-N）注文を受けた後、引当て指示が発行されない。その状況で受注ステータスが確定となり、出荷指示が発行されると、同じ商品の別注文により商品が不足した場合に出荷ができない。	注文されていない状況で引当て指示が発行される。その結果、引当て可能在庫数が実際より少なくなり、販売機会を失う。 →安全制約違反には該当しない。	注文を受けた後、引当て指示の発行が遅い。 →受注ステータスの確定が遅れるが、安全制約違反には該当しない。	–
2	商品の出荷指示	（UCA2-N）受注ステータスが確定と決定された注文に対して出荷指示が発行されない。そのため、出荷が行われない。	（UCA2-P）実際には出荷可能な商品が注文数より少ない状況で受注ステータスが確定と決定され、出荷指示が発行される。商品の現物が不足し、出荷ができない。	（UCA2-T）受注ステータスが確定と決定された後、出荷指示の発行が遅れ、速やかに出荷されない。	–
3	商品の出荷	（UCA3-N）受注ステータスが確定と決定された注文に対して出荷指示が発行されたが、出荷が行われない。	（UCA3-P）出荷指示とは異なる商品が出荷される。その商品が、他の注文に対して引当て済みの商品である場合、現物が不足し、その注文に対する出荷ができない。	（UCA3-T）受注ステータスが確定と決定された注文に対して出荷指示が発行された後、出荷がすぐに行われない。	–

3.3.4. Step2 HCF（Hazard Causal Factor）の特定

Step2 では、Step1 で抽出された UCA がどのような原因によって起こり得るかを考える。

UCA によっては、現実性の高い発生要因が存在しない場合もあり得る。例えば、UCA1-N は「注文を受けた後、引当て指示が発行されない」という状況を指しているが、販売管理機能が自動化されている前提からは、このような状況は考えにくい。自動処理するプログラムにエラーがあれば UCA1-N は起こり得るが、個々のコンポーネントのエラーにアクシデントの原因を求めることは STAMP/STPA の趣旨に沿わない。したがって、今回の分析では UCA1-N を HCF 分析の対象外とした。同様の理由で、UCA2-N と UCA2-T も HCF 分析の対象外とした。

その他の UCA に対する HCF 分析の結果を以下に説明する。HCF 分析では [IPA2016] に示されるガイドワードをヒントとして利用した。

3.3.4.1. UCA2-P の HCF 分析

UCA2-P「実際には出荷可能な商品が注文数より少ない状況で受注ステータスが確定と決定され、出荷指示が発行される」は、コントローラーである「販売管理機能」と被コントロールプロセスである「出荷機能」の間の非安全なコントロールアクションである。この 2 つのコンポーネントを図 3.3-4 に示す。

表 3.2-1 の仕様を参照すると、販売管理機能では以下のようなルールで受注ステータスが決定され、出荷指示が発行される。

- 在庫管理による引当て可否の応答が「引当て可」であれば受注ステータスを「確定」とし、「引当て不可」であれば受注ステータスを「不可」とする。
- 受注ステータスが確定となった場合は、出荷機能に対して出荷指示を発行する。

在庫管理による引当て可否の応答は、販売管理機能に対する入力情報である。ガイドワード「(1) コントロール入力や外部情報の誤りや喪失」に相当する UCA の誘発要因として、以下が考えられる。

「実際には出荷可能な商品が注文数より少ない状況で、引当て可否応答が『引当て可』となるため、受注ステータスが『確定』となり、出荷指示が発行される。」

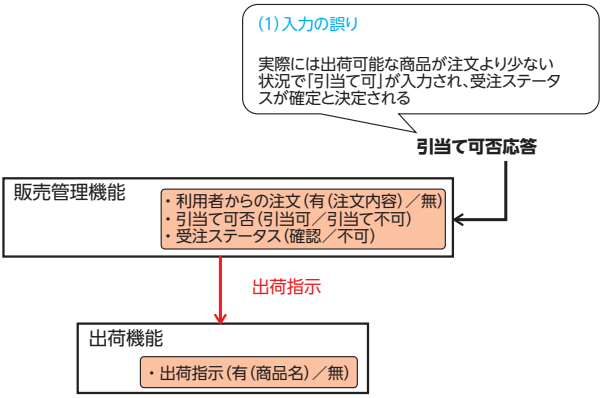


図 3.3-4　UCA3 の HCF 分析

さらに、「実際には出荷可能な商品が注文数より少ない状況で引当て可否応答が『引当て可』となる」ことの要因を分析する。引当て可否応答は在庫データに依存して行われるため、引当て可否応答が不正になる原因として在庫データが不正になる場合を考え、在庫データの読み・書きに着眼する。

図 3.3-2 のアクティビティ図のなかで、在庫管理機能から販売管理機能に対して引当て可否応答を行う箇所に注目する（図 3.3-5）。「引当て可否応答」では、在庫データが参照され、総在庫数と引当済み数のデータに依存して引き当ての可否が判断される。引当て可否を応答した後、在庫データの引当済み数が更新される（引当てた数量が加算される）。

ここで在庫データの更新・参照の同期性について考えると、今回分析対象とするシステムでは複数の注文が並列に処理されることを想定しているため、更新と参照の順番が一定とは限らない。例えば、ある注文（注文 A）に対して「引当て可」と応答した後、在庫データの更新が完了する前に他の注文（注文 B）に対する「引当て指示」により在庫データが参照される可能性がある。その場合に、注文 A に対する引当てを考慮に入れれば実際には注文 B に対する引当て可能な（出荷可能な）商品が不足しているにもかかわらず、注文 B に対して「引当て可」と応答されることが起こり得る。

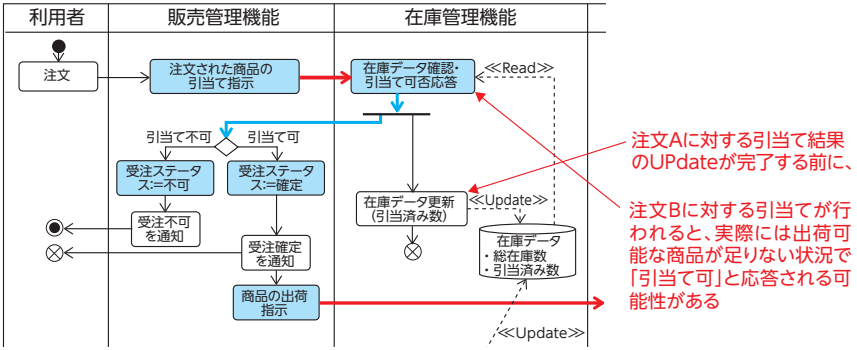


図 3.3-5　在庫が無い状況で引当て可が通知される要因の分析

以上の分析から、ハザードシナリオとして以下が考えられる。

UCA2-P に至るハザードシナリオ　HS2-P-1

「在庫データの確認・引当て可否応答」処理が、在庫データが最新状態に更新される前に実行されるため、実際には引当て可能な商品が足りない状況にもかかわらずそれが認識されず、「引当て可」が販売管理機能に通知され、出荷指示が発行される。

安全要件としては、

「商品の引当てを行う前に、在庫データの更新が完了し最新の状態になっていること」

を導くことができる。

この安全要件を満たすための対策としては、例えば、「在庫データ確認・引当て可否応答」処理において、在庫データの参照と更新を不可分な処理として同時に行うようにすることが考えられる（図 3.3-6）。

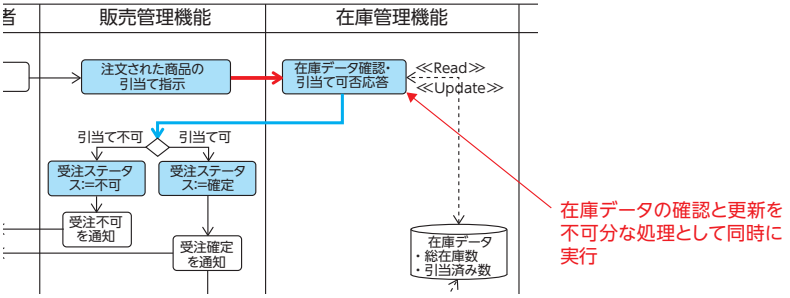


図 3.3-6　ハザードシナリオ HS2-P-1 に対する対策の例

3.3.4.2. UCA3-N の HCF 分析

UCA3-N「受注ステータスが確定と決定された注文に対して出荷指示が発行されたが、出荷が行われない」は、コントローラーである「出荷機能」と被コントロールプロセスである「配送機能」の間の非安全なコントロールアクションである。この 2 つのコンポーネントを図 3.3-7 に示す。

表 3.2-1 の仕様によると、出荷機能では以下のようなルールで配送指示が発行される。

- 出荷指示を受けると、受け取った出荷指示情報を参照して指示された商品をピックアップし、配送機能に引き渡す（出荷する）。

3.2.1.1 節「電子システム化（自動化）に関する前提」に示したとおり、出荷機能は人手による処理であり、出荷指示が見落とされることが考えられる。ガイドワード「(3) 不整合、不完全、または不正確なプロセスモデル」に相当するハザードシナリオとして以下が考えられる。

・UCA3-N に至るハザードシナリオ HS3-N-1

販売管理機能から出荷機能に対して商品の出荷指示が発行されたが、出荷担当者が出荷指示を認識せず、出荷が行われない

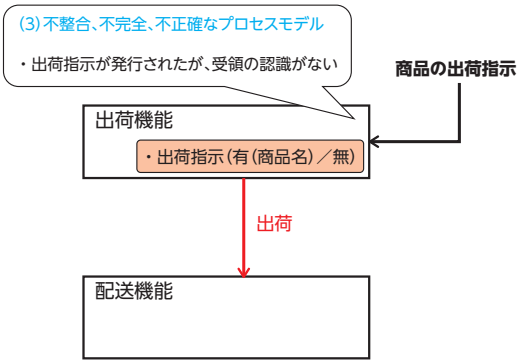


図 3.3-7　UCA3-N の HCF 分析（1）

HS3-N-1 から、安全要件として、

「商品の出荷指示が確実に認識されること」

を導くことができる。

この安全要件を満たすための対策としては、たとえば、図 3.3-8 に示すように、出荷機能が出荷指示を受領したことを販売管理機能に対して通知するフィードバックを追加することが考えられる。販売管理機能は、出荷機能からのフィードバックがない場合に出荷指示を再発行することなどにより、出荷指示の伝達の確実性を高めることが可能となる。

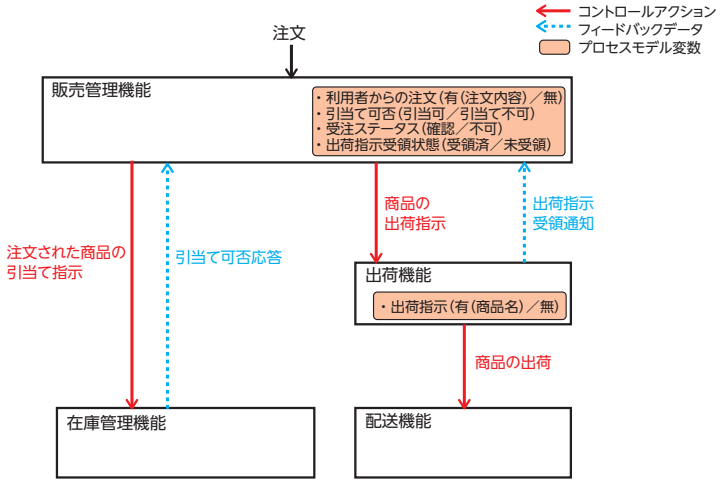


図 3.3-8　ハザードシナリオ HS3-N-1 に対する対策の例

また、ガイドワード「(4) コンポーネントの不具合、経年による変化」に相当するUCA の誘発要因として、配送機能が商品を受け取れない状況にある場合が考えられる。配送機能が商品を受け取れない要因としては、負荷が配送能力を超過し、新たな配送を受け付けられない場合などが考えられる（図 3.3-9）。

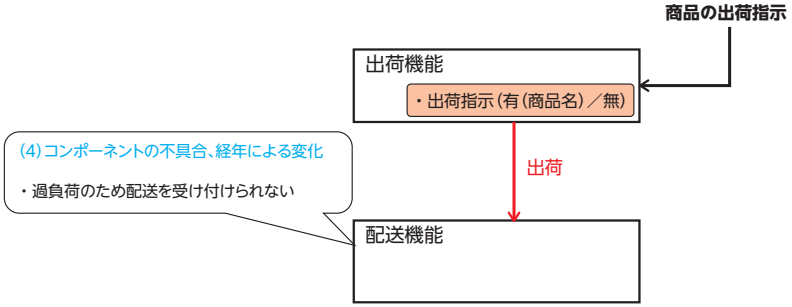


図 3.3-9　UCA3-N の HCF 分析（2）

したがって、UCA3-N に至る別のハザードシナリオとして、以下を考えることができる。

UCA3-N に至るハザードシナリオ HS3-N-2

配送機能の配送能力が不足し、出荷を受けられないため、出荷機能から出荷できない

安全要件としては、

「受注の最大量に応じた十分な配送能力が配備されていること」

を導くことができる。
この安全要件を満たすための対策としては、過去の実績などから最大の受注量を見積もり、それに応じた配送能力を持つ配送業者に委託することが考えられる。

3.3.4.3. UCA3-P の HCF 分析

UCA3-P「出荷指示とは異なる商品が出荷される」は、UCA3-N と同様に出荷機能のコントロールアクション「商品の出荷」に関わる UCA である。人手による処理であり、HS3-N-1 と同様のハザードシナリオが考えられる。

UCA3-P に至るハザードシナリオ HS3-P-1

出荷担当者が出荷指示の内容を誤認識し、出荷指示とは異なる商品が出荷される

安全要件は、

「商品の出荷指示が正しく認識されること」

が考えられる。この安全要件を満たすための対策としては、HS3-N-1 の対策と同様に、図 3.3-8 に示すようなフィードバックを追加し、出荷指示の認識を確実にすることが考えられる。

3.3.4.4. UCA3-T の HCF 分析

UCA3-T「受注ステータスが確定と決定された注文に対して出荷指示が発行された後、出荷がすぐに行われない」は、UCA3-N と同様に出荷機能のコントロールアクション「商品の出荷」に関わる UCA である。人手による処理であり、UCA3-N の HCF と同様に、出荷担当者による出荷指示の認識の遅れや、出荷先の配送機能が商品を受け付けられない状況が考えられる。

UCA3-T に至るハザードシナリオ HS3-T-1

販売管理機能から出荷機能に対して商品の出荷指示が発行されたが、出荷担当者による出荷指示の認識が遅れ、出荷がすぐに行われない

UCA3-T に至るハザードシナリオ HS3-T-2

商品の出荷指示が発行されたが、出荷先の配送機能の配送能力が不足し、配送を受け付けられない状態が続くため、出荷がすぐにできない

それぞれについての安全要件と対策は、HS3-N-1, HS3-N-2 と同様になる。

3.4. 試行結果と考察

本試行では、ネット通販システムを例題として、「注文した商品が利用者に配達されない」ことをアクシデントとした STAMP/STPA 分析を行った。その結果、表 3.4-1 に示すように UCA、HCF、安全要件とその対策を導くことができた。

表 3.4-1　STAMP/STPA 分析で得られた結果

UCA	HCF	安全要件	対策例
UCA2-P： 実際には出荷可能な商品が注文数より少ない状況で受注ステータスが確定と決定され、出荷指示が発行される。商品の現物が不足し、出荷ができない。	HS2-P-1： 「在庫データの確認・引当て可否応答」処理が、在庫データが最新状態に更新される前に実行されるため、実際には引当て可能な商品が足りない状況にもかかわらず「引当て可」が販売管理機能に通知され、出荷指示が発行される。	SC2-P-1： 商品の引当てを行う前に、在庫データの更新が完了し最新の状態になっていること。	「在庫データ確認・引当て可否応答」処理において、在庫データの参照と更新を不可分な処理として同時に行うように変更する。
UCA3-N： 受注ステータスが確定と決定された注文に対して出荷指示が発行されたが、出荷が行われない。	HS3-N-1： 販売管理機能から出荷機能に対して出荷指示が発行されたが、出荷担当者が出荷指示を認識せず、出荷が行われない	SC3-N-1： 商品の出荷指示が確実に認識されること。	出荷機能が出荷指示を受領したことを販売管理機能に対して通知するフィードバックを追加する
	HS3-N-2： 商品の出荷指示が発行されたが、配送機能の配送能力が不足し、出荷を受けられない	SC3-N-2： 配送機能は、受注の最大量に応じた十分な配送能力を持つこと。	最大の受注量を見積もり、それに応じた配送能力を持つ配送業者に業務委託する。
UCA3-P： 出荷指示とは異なる商品が出荷される。その商品が、他の注文に対して引当て済みの商品である場合、現物が不足し、その注文に対する出荷ができない。	HS3-P-1： 出荷担当者が出荷指示の内容を誤認識し、出荷指示とは異なる商品が出荷される。	SC3-P-1： 商品の出荷指示が正しく認識されること。	出荷機能が出荷指示を認識した内容を販売管理機能に対して通知するフィードバックを追加する
UCA3-T： 受注ステータスが確定と決定された注文に対して出荷指示が発行された後、出荷がすぐに行われない。	HS3-T-1： 販売管理機能から出荷機能に対して商品の出荷指示が発行されたが、出荷担当者による出荷指示の認識が遅れ、出荷がすぐに行われない。	SC3-T-1： 商品の出荷指示が速やかに認識されること。	出荷機能が出荷指示を受領したことを販売管理機能に対して通知するフィードバックを追加する。
	HS3-T-2： 商品の出荷指示が発行されたが、配送機能の配送能力が不足し、配送を受け付けられない状態が続くため、出荷がすぐにできない。	SC3-T-2： 配送機能は、受注の最大量に応じた十分な配送能力を持つこと。	最大の受注量を見積もり、それに応じた配送能力を持つ配送業者に業務委託する。

本試行を実施した動機は次の 2 点を明らかにすることであった。

- エンタープライズ系システムに対して STAMP/STPA 分析の手法は適用可能か
- エンタープライズ系システムに対して STAMP/STPA 分析を行う場合に特有の注意点やコツはあるか

これらの観点で本試行の結果を以下のように考察する。

1 点目に関しては、ネット通販システムがサービスを遂行できないことをアクシデントとして、STAMP/STPA の手法に則った分析を行った結果、仕様の不備を発見でき、必要な要件を導き出すことができた。エンタープライズ系に対しても STAMP/STPA 分析は適用可能であり、有益な結果が得られる場合があるといえるであろう。

2 点目に関しては、二つの発見があった。一つは、コントロールストラクチャーの構築にアクティビティ図の情報を利用するアイディアを得たことである。このアイディアに沿った方法で実際にコントロールストラクチャーを構築することができた。もう一つは、UCA2-P の HCF 分析において、データの更新・参照に関するアクションの依存関係に着眼した要因分析が有効であったことである。とくに後者は、データ処理を中心とするエンタープライズ系システムに特徴的な結果であったと考えられる。

今回の試行では STPA 分析の結果をもとにした安全対策の検討まで行った。例えば、UCA2-P の HCF 分析の結果をもとに、図 3.3-6 に示すように元々別であったアクションを 1 つのアクションに統合する改善案が得られた。これは、表面的に見ればアクティビティ図で書かれた仕様のレビューと修正である。しかし、個人の経験やセンスに大きく依存してレビューを行うのではなく、アクティビティ図のどの部分をどのような意識を持って見ればよいかを根拠を伴って系統的に導き出せる点が STPA を利用する価値といえるであろう。

今回の分析の結果として得られた安全要件は、常識の部類に属する内容であるかもしれない。しかし、システム開発には必ずしも経験の豊富な人材が参加するとは限らず、また、要求仕様が複雑化する傾向にあることを考えると、系統立った手法で要件を導き出せることは有益である。STAMP/STPA の手法がエンタープライズ系システムに適用可能であり、安全要件を導き出せることを確認できた点で今回の試行は有意義であったといえる。より複雑な仕様のシステムに対して分析を行えば、より価値の高い安全要件が得られる見通しが立った。

3.5. 分析結果の更なる活用の可能性

今回の試行は、前節に示したような分析結果をもって終了したが、安全要件への対策としてコントロールアクションの追加を考えた結果、さらなる分析に発展する可能性が得られた。WG の活動期間では、さらなる分析を詳細に行うことはできなかったが、発展の可能性について本節で述べておく。

安全要件 SC3-N-2「受注の最大量に応じた十分な配送能力が配備されていること」に対しては、3.3.4.2 に示した対策例のほかに、次のような対策も考えられる。

【SC3-N-2 への対策例】

- 利用者による「注文」アクションの前に、商品の予約行為である「カートに入れる」アクションを追加する。
- 「カートに入れる」アクションで指定された商品の情報から、発生する注文量の事前予測を行い、配送能力の増減を動的に制御するアクションを追加する[1]。

「カートに入れる」アクションを追加した業務フロー図（アクティビティ図）を図 3.5-1 に示す。在庫データの「予約済み数」を参照し、発生する注文量の事前予測に応じて配送能力を制御するアクションは、図 3.5-2 に示すように、非同期のアクションとして追加される。図 3.5-3 は、これらのアクティビティ図から、3.3.2 節に示した手順によって構築されるコントロールストラクチャーの一例である。

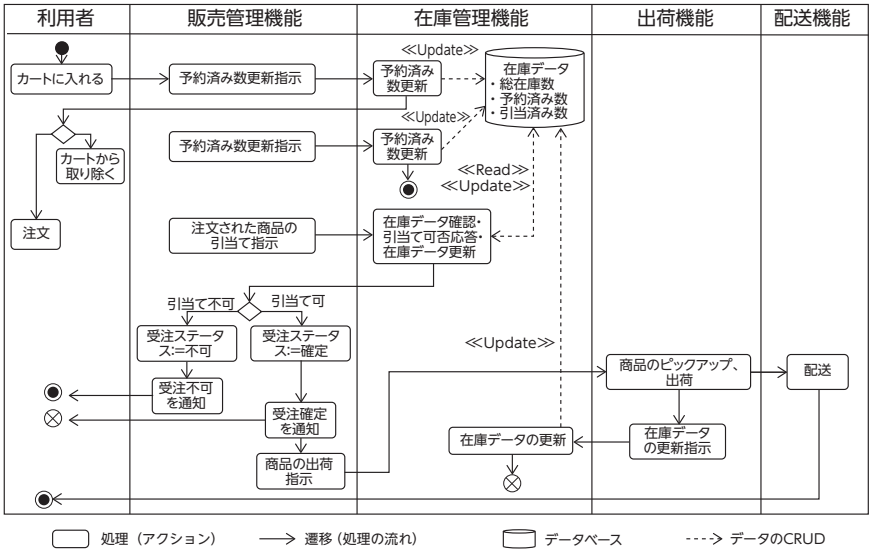


図 3.5-1 「カートに入れる」を追加したアクティビティ図

[1] 配送能力の増減は、例えば配送業者への臨時委託の増減で行うことが考えられる。

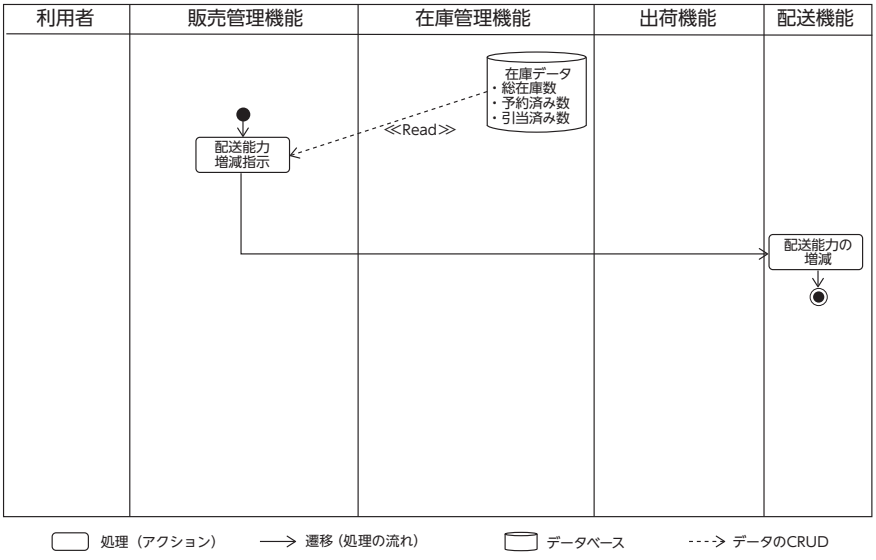


図 3.5-2　配送能力の制御を追加したアクティビティ図

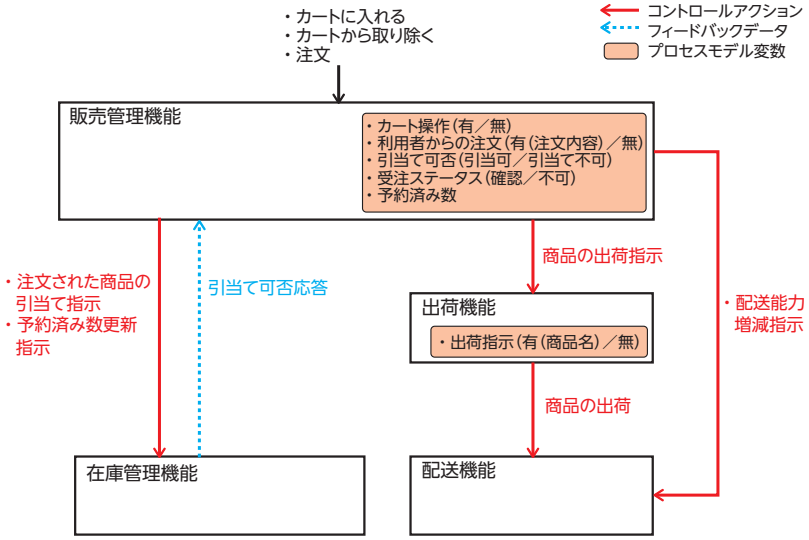


図 3.5-3　図 3.5-1 と図 3.5-2 から構築されるコントロールストラクチャーの例

このように対策を考えた結果、追加された「カートに入れる」アクションがきっかけとなり、以下に示すような新たな分析に発展する可能性が生じる。

今回例題としたネット通販システムでは、注文された商品の引当てができなかった場合、受注不可となり販売機会を逸する。しかし、上記の対策により「注文量の事前予測」が可能となることから、「在庫量を最適に保つ制御」の発想を得た。すなわち、在庫不足が発生しないように商品の入荷を行うアクションの追加である。図 3.5-4 は、「商品の入荷指示」を追加したアクティビティ図であり[2]、図 3.5-5 は、図 3.5-1、図 3.5-2、図 3.5-4 の 3 つのアクティビティ図で表される仕様から、3.3.2 節に示した手順によって構築されるコントロールストラクチャーの一例である。「商品の入荷指示」は、例えば、引当て可能な在庫量と予約済み数の差分がある閾値を超えた場合に発行されるものとして定義することが考えられる。

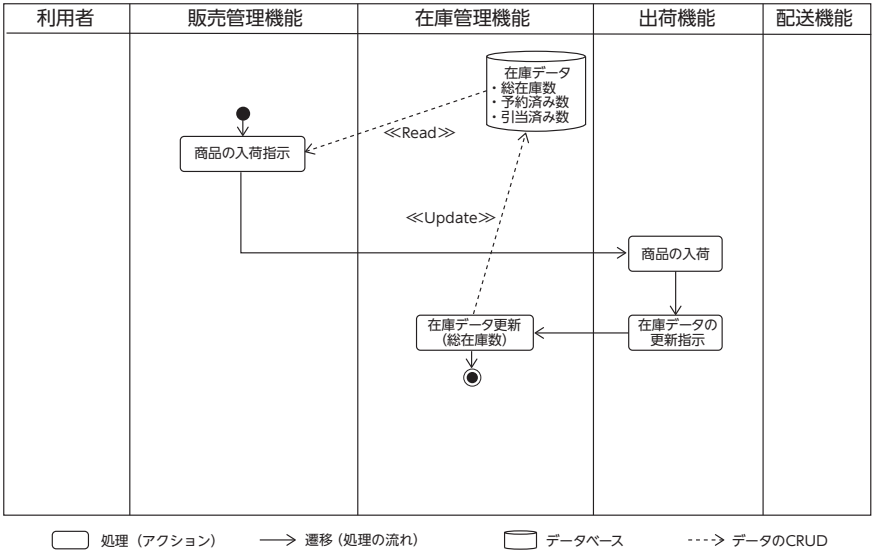


図 3.5-4　商品の入荷を行うアクションを追加したアクティビティ図

[2] 出荷機能は、入荷も行うように拡張したため、「入出荷機能」と改めた。

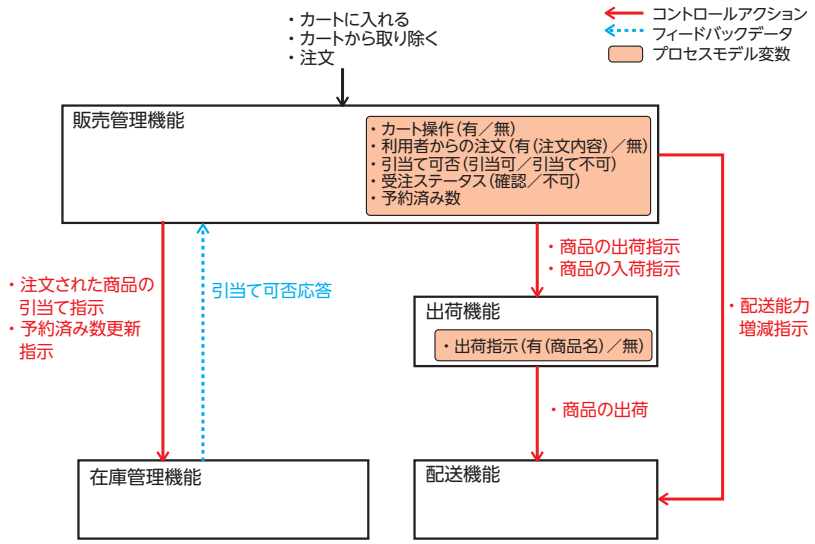


図 3.5-5「商品の入荷指示」を追加したコントロールストラクチャーの例

このように、在庫量を最適に保つ制御を持つシステムに対して、「販売機会の喪失」をアクシデントとして STPA 分析を行えば、販売機会の喪失を防ぐための要件を得ることができるはずである。すなわち、売り上げを高めるようにシステムを改善・進化させていける可能性がある。詳細な実証確認は未実施であるが、STAMP/STPA を単に安全分析のツールとしてだけではなく、システムを進化させるツールとしても利用できる可能性が確認できた。

4. STPA 解析を実施する際のヒントワード

4.1. ハザード誘発要因（HCF：Hazard Causal Factor）特定の考え方

STPA の特長の一つは、複数の機器や人・組織が、相互に作用する複雑なシステムにおいて、システム全体の振る舞いを確認しながらポイントを絞って解析可能にし、相互作用のハザード誘発要因を識別できることである。

STPA では、ハザード誘発要因を特定する際の参考として安全制約を破られる原因の例が 13 個のガイドワードとして挙げられている（図 4.1-1）。

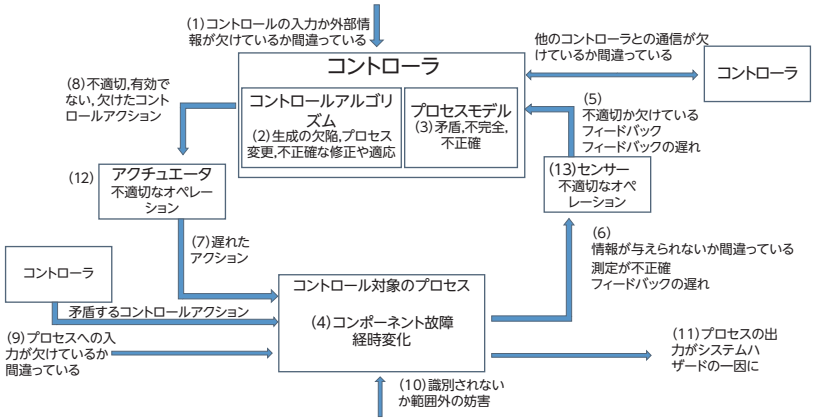


図 4.1-1　安全制約を破られる原因の例

ガイドワードは、想定外の HCF を見つけるための重要な働きをする。ドメインエキスパートではない人が分析できるようにするにはガイドワードが必要で、解析対象分野の共通基盤技術をプラスすることで解決できる可能性が期待できる。しかし、現状では、提示されているガイドワードが制御機械と稼働機械の組合せのみであり、現在、安全性解析を必要とするシステムの大きな部分を占めており、今後ますます重要性が大きくなっていくであろう人と組織と機械が協調して動作するシステムの安全性解析では、人と組織によって安全性が破られる原因の例が整理されていることが望まれる。

人と組織によって安全性が破られる原因の例を考えていくに際し、STPA が持っているガイドワードとの混乱を避けるため、以後の議論では " ヒントワード " と呼ぶことにする。

ヒントワードの要件として以下の 3 つが考えられる。

- 網羅性
  - ✓ UCA のガイドワードに対応
    （Not Providing/ Providing causes hazard , Providing too early , Providing too long）
- 現実的
  - ✓ 人間の生物的特性に沿っていること（視覚 / 聴覚 / 触覚）
  - ✓ 人間の生理的特性に沿っていること（集中力 / 記憶力 / 精神力 / 生理現象）
  - ✓ 善意 / 悪意 / 恐怖 / 圧力といった心理的条件を含む
  - ✓ 環境条件を含む

➢ 対策実現性
- ✓ 技術的実現性
- ✓ 業務ワークフローの実行可能性

これらの要件に添ってヒントワードを抽出し整理するために以下のアプローチをとった。
- ✓ ヒューマンファクターを参照
- ✓ 人間工学を参照
- ✓ 事故事例調査 / 収集と教訓化活動から得られたハザード誘発要因を抽出・整理
- ✓ コントロール側を Not Providing/ Providing causes hazard に、被コントロール側をコミッション / オミッションに対応づけて整理

ヒューマンファクターでは、日本ヒューマンファクター研究所から提案されている M-SHEL モデルを参照した（図 4.1-2）。

M-SHELモデル：(出典：日本ヒューマンファクター研究所)
機械やシステムを安全に、有効に機能させるために必要とされる人間の能力や限界、特性等のヒューマンファクターを表現するためのモデル。
中央の L：(当事者)
H：ハードウェア：（機械、機材、設備、等）
M：マネジメント：（コミットメント、体制、分担、リスク管理、等）
S：ソフトウェア（規定、規則、細則、要領、等）
E：環境（気温、湿度、換気、騒音、照明、空間、遠近、利便、安全文化、風土、慣習、等）
L：（相手、関係者、第三者）

図 4.1-2　M-SHEL モデル

このモデルは、STPA のコントロールループとの親和性が高い。双方の要素を次のように対応付けることができる。

STAMP/STPA	M-SHEL
コントローラー	（中央の）L
コントロール対象	（外側の）L,H
コントロールアルゴリズム	M
プロセスモデル	S
全体への影響	E

人間工学では、人の認知行動を以下のモデルでとらえることができる。このモデルを使って以下のようにハザードにつながる要因を考えることができる（図 4.1-3）。

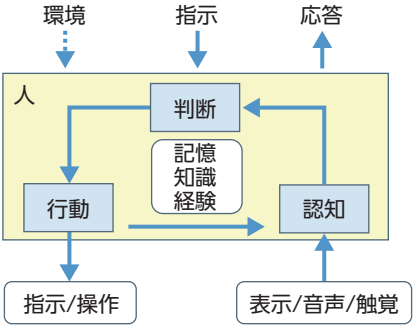


図 4.1-3　人の認知行動モデル

「認知」に対して考えられる要因
　表示など出力を欠落させる
　正しい / 正しくない出力を間違って認知する
「行動」に対して考えられる要因
　正しくない指示を出す
　指示を出さない
「判断」に対して考えられる要因
　指示 / 操作を忘れる、必要と思わない　意図的 / 無意識
　指示 / 操作を勘違い / 考え違いする
　指示済み / 思い込み
「環境」が人に対して与える要因
　誤った判断を引き起こす人的因子（コミュニケーションエラー）
　連絡不足 / 報告不足 / 確認不足 / 公開不足 / 隠蔽 / 複雑な手順
　誤った判断を引き起こす精神的因子
　プレッシャー / 規制強化 / 社会通念 / 事故感受性の不足
　視覚・聴覚の妨害 / 錯覚

　次に、ここで参照した内容から、IPA/SEC で進めている事故事例調査 / 収集と教訓化活動 [IPA2016-4] から得られたハザード誘発要因を抽出し、STPA のコントロールループにあわせてコントロール側、被コントロール側に分けた。
　さらにコントロール側は Not Providing/Incorrectly Providing（too early , too long を含む）に分け、被コントロール側はコミッション / オミッションに対応づけて整理した。

4.2. 人と組織に関するハザード誘発要因（HCF）の事例

4.1 節で述べた通りに STPA のコントロールループにあわせてコントロール側、被コントロール側に分け、さらにコントロール側を Not Providing/ Providing causes hazard（too early , too long を含む）に分け、被コントロール側をコミッション / オミッションに対応づけて整理した（図 4.2-1）。

人と組織と機械の組み合わせごとにヒントワードを図 4.2-2、図 4.2-3、図 4.2-4、図 4.2-5 の形式で整理した。ただし対称になっている組み合わせは省略した。

ヒントワードの組合せ

- （人）　対（人）　の HCF 導出のためのヒントワード
- （人）　対（機械）の HCF 導出のためのヒントワード
- （機械）対（人）　の HCF 導出のためのヒントワード　（省略）
- （組織）対（人）　の HCF 導出のためのヒントワード
- （人）　対（組織）の HCF 導出のためのヒントワード　（省略）
- （組織）対（組織）の HCF 導出のためのヒントワード

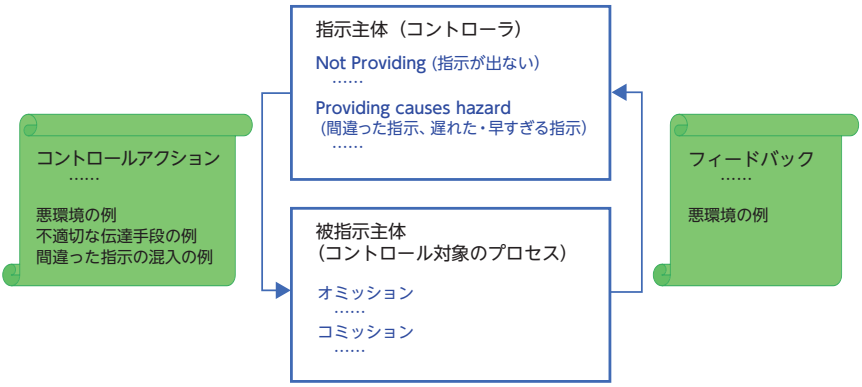


図 4.2-1　ヒントワードの表現形式

以下の図中で使う記号は次の通り。

A：アクション
F：フィードバック
HC：指示主体（人）に関するヒントワード
HP：被指示主体（人）に関するヒントワード
MP：被指示主体（機械）に関するヒントワード
AC：指示主体（組織）に関するヒントワード
AP：被指示主体（組織）に関するヒントワード

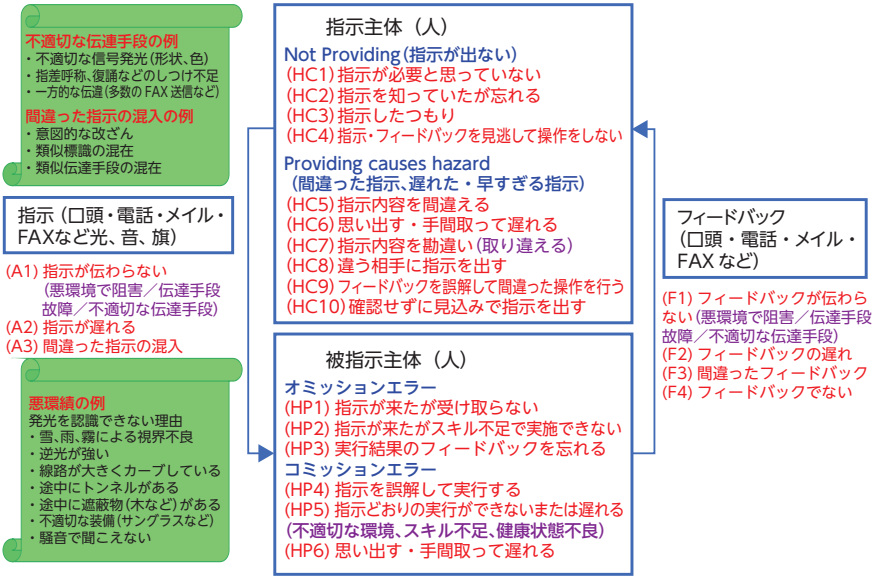


図 4.2-2 （人）対（人）の HCF 導出のためのヒントワード

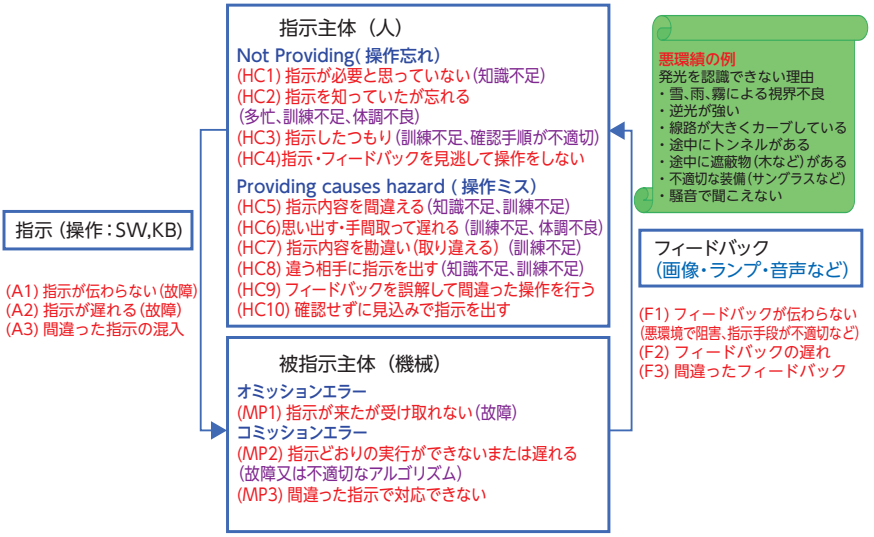


図 4.2-3 （人）対（機械）の HCF 導出のためのヒントワード

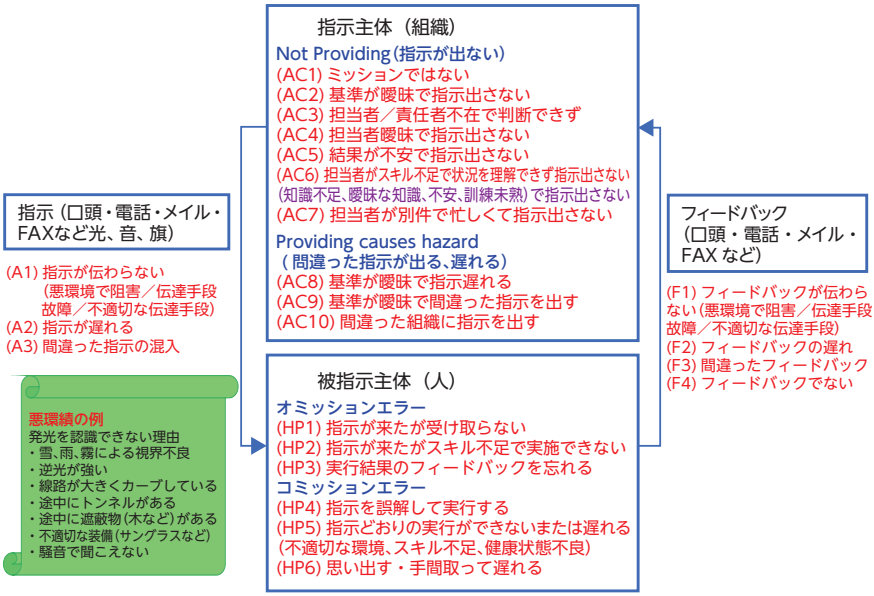


図 4.2-4 （組織）対（人）の HCF 導出のためのヒントワード

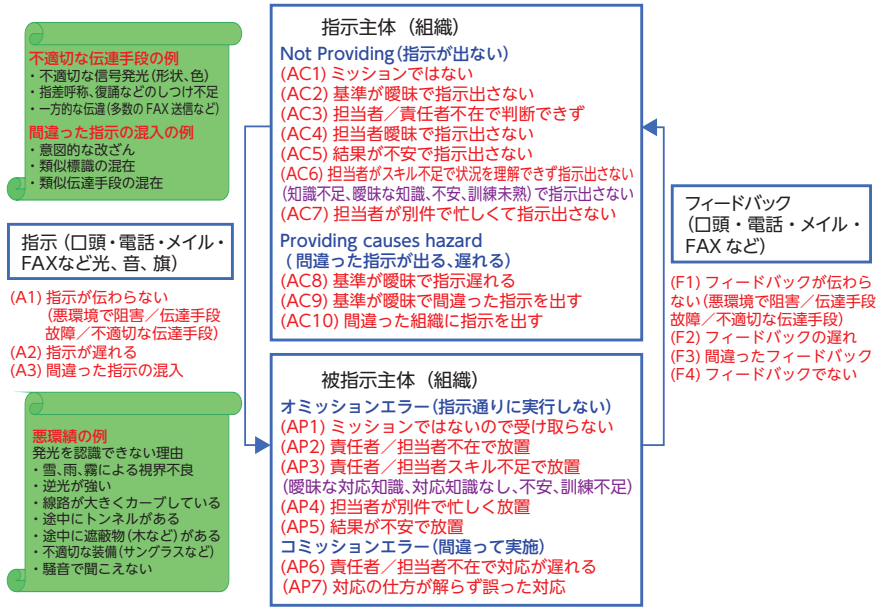


図 4.2-5 （組織）対（組織）の HCF 導出のためのヒントワード

5. STPA 支援手法

本章では、他手法を用いて STPA を支援する事例を紹介する。5.1 節では、アーキテクチャー分析設計言語 AADL（Architecture Analysis and Design Language）を用いたコントロールストラクチャー記述及び STPA 支援の事例を紹介する。5.2 節では、SAT/SMT ソルバーを用いた STPA 支援の事例として、Step1 分析結果の縮約と整合性検証を紹介する。

5.1. AADL による STAMP/STPA 支援事例

本節では、アーキテクチャー分析設計言語 AADL[[SAE2012],[Feiler2012]] を用いたコントロールストラクチャー記述及び STPA 支援の事例を紹介する。AADL を用いることで、分析設計モデル（AADL モデル）と安全解析モデル（STAMP）を統合できる。本節の構成は以下のとおりである。はじめに AADL と AADL のエラー記述用拡張である Error Model Annex について簡単に解説する。次にそれらを用いた STAMP コントロールストラクチャーの記述例と STPA 分析支援の例を紹介する。なお、本節の事例は文献 [Procter2015] を参考にした。

5.1.1. AADL の解説

AADL の特徴について述べる。AADL はアーキテクチャー分析設計言語であり、記述形式としてテキスト形式と図式の二種類の記述形式がある。また、コンポーネントベースで記述を行うため、STAMP のコントロールストラクチャーと相性が良い。AADL の主たる記述対象は組込みシステムであり、ソフトウェアはパッケージ、データ、サブプログラム、抽象構成要素等を用い、ランタイム・アーキテクチャーはプロセス、スレッド等を用い、ハードウェアはプロセッサ、メモリー、バス等を用いそれぞれ記述する。（図 5.1-1）また AADL は厳密な意味論を持つため、解釈に曖昧さが無い。

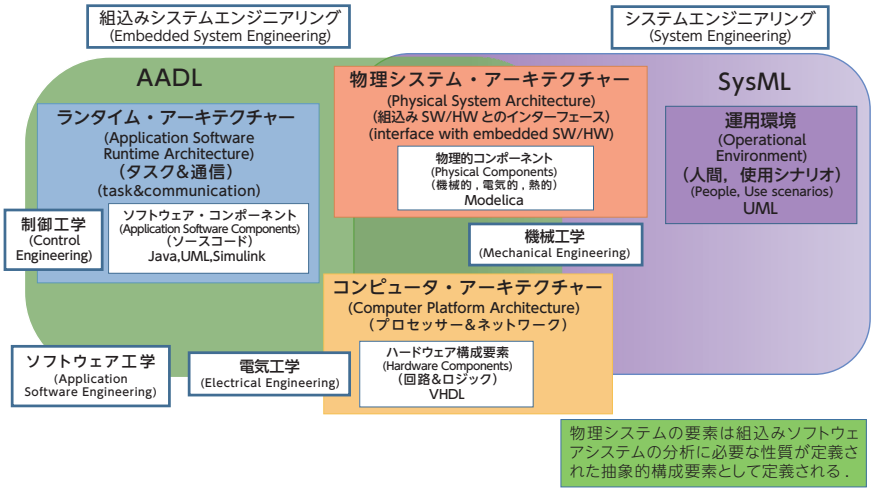


図 5.1-1　AADL の対象範囲 [Feiler2012]

AADL はコンポーネントベースの言語である。コンポーネントは物理的あるいは機能的なまとまりである。またコンポーネント間のデータやイベントのやり取りを AADL モデルに記述できる。さらに、コンポーネントは階層的に記述できるため、可読性を落とさずに複雑なシステムを表現できる。コンポーネントベースの記述は、STAMP のコントロールストラクチャーを記述するのに適している。図 5.1-3 に示すように、STAMP のコントロールストラクチャーの構成要素と AADL のモデル要素の間に対応を付けることができる。

AADL はアノテーションによる拡張をサポートしており、アノテーションを追加することで図 5.1-2 にある解析が可能となる。例例えば、アノテーションの一つである The Error Model Annex standard for AADL ver.2（EMV2）を用いてエラー情報を AADL モデル（分析設計モデル）に追記することで、分析設計モデルと安全解析モデルを統合できる。さらに EMV2 を用いたエラー情報を追加された AADL モデルに対し、FTA や FMEA 等の安全解析を実施できる。一般に、安全解析モデルは分析設計モデルやその他の資料を基に作成されることが多く、両モデルの整合性や保守性を維持するには注意が必要である。しかし、AADL では分析設計モデルに安全解析モデルを統合することで、両者の整合性や保守性を向上させている。

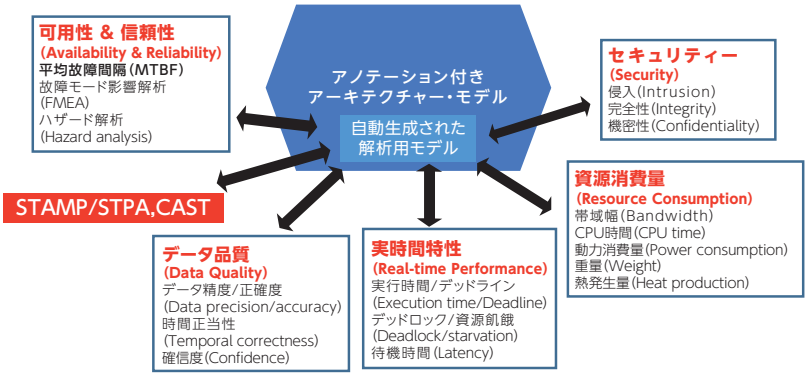


図 5.1-2　アーキテクチャーを中心としたモデルベースエンジニアリング [Feiler2012]

AADL の文法や記述例に関する教科書・辞書としては、文献 [Feiler2012] が挙げられる。AADL に関する情報は一通り書かれており、簡単な例に沿って文法が学習できるようになっている。また web 上には研究集会のスライドや AADL 記述の例が公開されている。

- https：//wiki.sei.cmu.edu/aadl/index.php/Main_Page
- http：//www.aadl.info/aadl/currentsite/

EMV2 を用いることで、AADL モデルにコンポーネント内のエラー動作（error behavior）や、コンポーネント内及びコンポーネント間のエラー伝搬（error propagation）に関する情報を付加できる [SAE2014]。EMV2 には、類型化されたエラーを階層的に整理したエラー・タイプ（error type）が用意されている。典型的なエラーはこのエラー・タイプに含まれており、このエラー・タイプを拡張して利用者独自のエラーを定義できる。

EMV2 を用いてエラー情報を追加した AADL モデルを用いることで、Fault Tree

Analysis（FTA）, Failure Model and Effect Analysis（FMEA）といった安全解析をOSATE2ツール [OSATE2016] を用いて実施できる。また必要な情報をAADLモデルへ追加することで、同じAADLモデルを用いて信頼性やセキュリティー等も解析できる（図 5.1-2）。

EMV2の文法や解析法に関する情報源としては、文献 [SAE2014] 以外にも文献 [Delange2014] や [Delange2014-2] がある。文献 [Delange2014] には、EMV2の文法や分析法がコンパクトにまとめられており、EMV2の概略を知るのに便利である。文献 [Delange2014-2] では、AADLを利用して ARP4761 Safety Assessment を実施するために、安全解析手法毎にAADLモデルの構築法と分析手法が提案されている。これらを参照することで、EMV2による記述がどのように定義されており、どのように利用できるかを知ることができる。

AADLはEMV2と組み合わせて使用することで、FTAやFMEA等の安全解析を可能にしている。他方、STAMPはシステム理論に基づくアクシデントモデル（安全解析モデル）であり、広範囲なシステムライフサイクルにおけるモデリングと安全解析や障害発生後の事後分析が可能である。そこでAADLとEMV2を用いたコントロールストラクチャー記述及びSTPA支援の事例を5.1.2で紹介する。

5.1.2. AADLを用いたSTAMPの事例紹介

本項ではAADL及びEMV2を用いたコントロールストラクチャー記述とSTPA支援の事例を紹介する。紹介する事例は、文献 [Procter2015] を基に詳細部分を一部独自に追加・修正した事例である。参考にした事例ではAADLを用いて医療機器に対しSTPAを実施している。具体的には、STAMPのコントロールストラクチャーをAADLおよびEMV2で記述し、STPAのStep1,2の分析結果からの文書自動生成や、人手と機械支援による相補的STPAを提案している。

STAMPのコントロールストラクチャーの構成要素とAADLのモデル要素の対応を示す。AADLはコンポーネントベースの言語であるため、コントロールストラクチャーの各要素（コントローラー、アクチュエーター、被コントロールプロセス、センサー）はコンポーネントとしてモデル化する。文献 [Procter2015] ではそれぞれの役割を考慮し、コントローラーはAADLのプロセス（AADL *Process*）として、アクチュエーターとセンサーはAADLのデバイス（AADL *Device*）として、そして被コントロールプロセスはAADLの抽象要素（AADL Abstract）としてそれぞれモデル化している。また、コントールストラクチャー中のこれらの要素は相互に接続されており、それらの接続はAADLの接続（AADL *Connection*）を使ってモデル化される（図 5.1-3）。

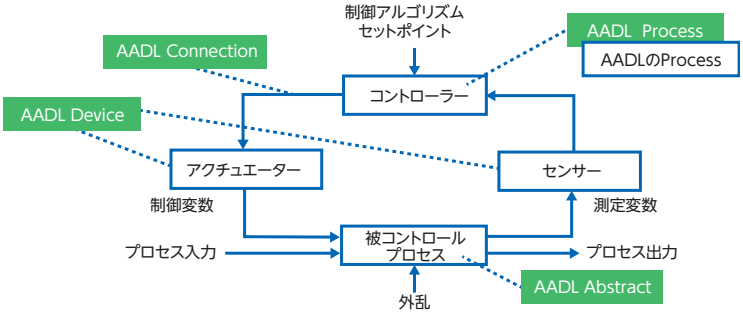


図 5.1-3　AADLを用いたコントロールストラクチャー [Procter2015]

STPA の Step1 では、４つのガイドワード（“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”）が用意されている。これら４つのガイドワードは EMV2 が提供するエラー・タイプや解析者が定義するエラー・タイプ集合（error-type set）として定義できる。（表 5.1 1）

表 5.1-1　EMV2 エラー・タイプと STPA ガイドワードの対応例

STAMP/STPA（ガイドワード）	EMV2（エラー・タイプ）	分析対象固有エラー・タイプ（EMV2 の拡張として定義）
与えられないとハザード	ServiceOmission, ItemOmission, …	ServiceOmission 等の拡張エラー・タイプとして定義
与えられるとハザード	ServiceComission, …	ServiceComission 等の拡張エラー・タイプとして定義
早すぎ、遅すぎ、誤順序でハザード	EarlyDelivery, Late Delivery, …	EarlyDelivery 等の拡張エラー・タイプとして定義
早すぎる停止、長すぎる適用でハザード	EarlyService Termination, LateServiceTermination, …	EarlyService Termination 等の拡張エラー・タイプとして定義

前述のコントロールストラクチャーの AADL による記述方針（図 5.1-3）と STPA Step1 の４つのガイドワードの記述方針（表 5.1-1）に基づき、STPA を実施する。

Step0 エラー情報無しコントロールストラクチャー（CS（AADL））の記述：図 5.1-3 に示すように、アーキテクチャーレベルのコントロールストラクチャーは AADL を用いて記述できる。（ただし本節では、抽象レベルの STPA を実施するため、コントロールストラクチャーの構成要素の一部を AADL の抽象要素（AADL Abstract）としている。）この AADL モデル CS（AADL）は通常のコントロールストラクチャーと同等の情報を有しており、このコントロールストラクチャーを基に人手による STPA が実施可能である。（図 5.1-4）

この AADL モデルは分析設計モデルと安全解析モデルが統合されたモデルであるため、モデルの整合性やモデルの保守性の観点から有用である。文献 [Procter2015] では、Step1,2 の分析結果を CS（AADL）に追記し、合わせて必要なツールを開発することで、分析結果付きコントロールループをドキュメントとして自動生成している。

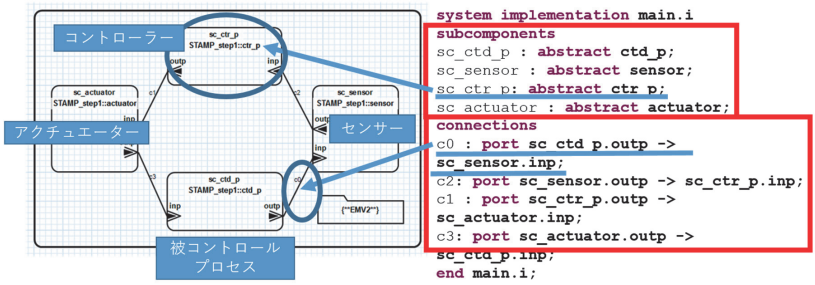


図 5.1-4　コントロールストラクチャー（エラー情報無し）

Step0 エラー情報付きコントロールストラクチャー（CS（AADL+EMV2））の記述：エラー情報無し AADL モデル CS（AADL）を用いた STPA Step1,2 では、一般的な Step1,2 同様に人手で分析を実施する。CS（AADL）に対し EMV2 を用いてエラー情報を追加することで、人手による分析を支援する方法について紹介する。

エラー伝搬に関する情報を CS（AADL）に追加することで、例えば「コントローラーからアクチュエーターに " 与えられないとハザード " が伝播する」とコントローラー側に記述されているのに、アクチュエーター側には対応する記述が無いといった記述漏れがツールにより指摘できる。またコンポーネント内やコンポーネント間のエラー伝搬情報を追記することで、Fault Impact Analysis によりエラーの影響範囲を調査できる [[Delange2014], [Delange2014-2]]。エラー伝搬に関する情報を CS（AADL）に追加した AADL モデル CS（AADL+EMV2）の例を図 5.1-5 に示す。なお、本項では、エラー伝搬は EMV2 のフロー（flows）だけを用いてモデル化した。フロー以外にも、コンポーネントのエラー状態遷移が記述できるエラー動作モデル（component error behavior）や、サブシステムのエラー状態をそのサブシステムに含まれるコンポーネントのエラー状態の合成として記述できるエラー合成（composite error behavior）を用いることで、エラーの伝搬をより詳細にモデル化できる。

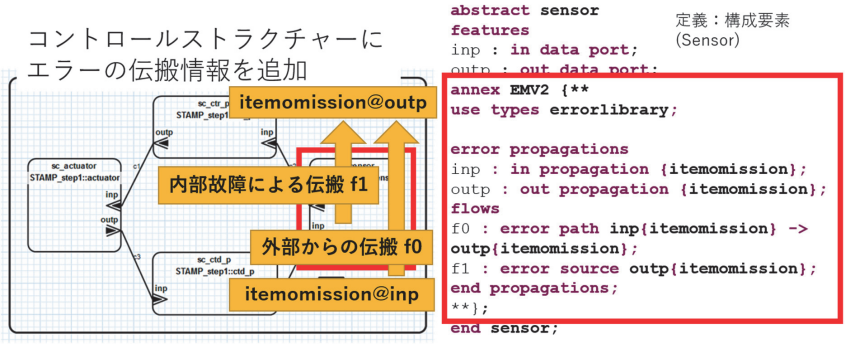


図 5.1-5　コントロールストラクチャー（エラー情報有り）

STPA Step1,2 支援：文献 [Procter2015] では、人手による STPA Step1,2 と CS（AADL+EMV2）に基づく分析支援を相補的に用いる分析方法を提案している。図 5.1 6 は [IPA2016] の事例を基に [Procter2015] で提案されている分析方法を Fault Impact Analysis で実現した図である。提案分析方法の概略は以下の通りである。初めに記述した CS（AADL+EMV2）に対して人手による Step1,2 を実施する。続いて STPA の結果判明した情報を CS（AADL+EMV2）へ追加し、AADL で利用可能な分析を利用して調査を行う。この調査の結果を基に、Step1,2 の結果へ追記・修正を行うというサイクルを繰り返す。このサイクルを繰り返すことで、STPA の分析結果を改善していく。

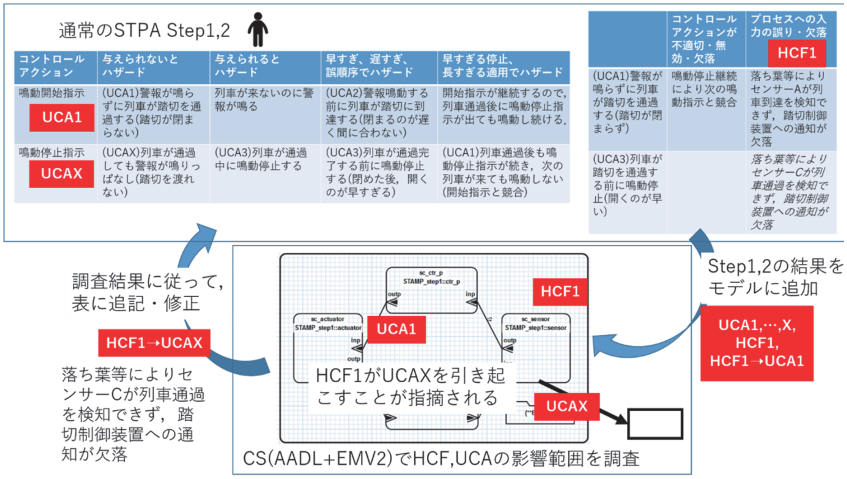


図 5.1-6　Fault Impact Analysis を用いた STPA 支援

人手による STPA Step1,2 以外の部分について、具体的な方法を解説する。

Step1,2 の結果判明した UCA が伝播してハザードへ至る様子は、エラー・タイプ（STPA Step1 のガイドワードに相当）が伝搬していく様子として捉えられる。そこで、EMV2 のエラー伝搬（flows）を用いて、UCA が伝搬する様子を CS（AADL+EMV2）に追記する。同じく Step1,2 の結果判明した HCF は UCA を引き起こす要因の一つであるので、HCF は EMV2 のエラー・ソース（error source）を用いてモデル化し、CS（AADL+EMV2）へ追記する。一般に STPA Step2 で指摘するのは UCA の複合的な要因であるが，単純なコンポーネント故障の場合には、上述のモデル化で要因をモデル化できる。

新たにエラー情報を追加された CS（AADL+EMV2）に対し Fault Impact Analysis を実施することで、追記した HCF 及び UCA による影響を指摘できる。特に OSATE2 ツールによる Fault Impact Analysis は網羅的に影響範囲を指摘できるため、人手による分析時に漏れたエラー伝搬の発見が期待できる。他方、Fault Impact Analysis は単一要因からの影響範囲を調査するのに適しているが、STPA の特徴である複合要因による相互作用による影響範囲を調査できない。したがって、ある UCA 候補がハザードへ至るか否かを自動的に判定することは、この CS（AADL+EMV2）に基づく Fault Impact Analysis だけでは実施できず、人手による分析が必要である。さらに、人手による気づきはモデルに記述されていない事柄に気づく有効な手段である。このように、機械による分析支援と人手による分析を組み合わせることは分析の質の向上に寄与する。

5.1.3. まとめ

本節では、アーキテクチャー分析設計言語 AADL を用いたコントロールストラクチャー記述及び STPA 支援の事例を紹介した。AADL を用いることで、分析設計モデル（AADL モデル）と安全解析モデル（STAMP）を統合できた。参考にした事例 [Procter2015] では、AADL 及び EMV2 を用いて STAMP のコントロールストラクチャー（CS（AADL+EMV2））を記述し、報告書作成等の支援を実施しており、CS（AADL+EMV2）を用いた STPA 支援も合わせて提案している。

AADL を用いて STAMP のコントロールストラクチャーを記述することで、分析設計モデルと安全解析モデルの統合できる。この統合により、分析設計モデルと安全解析モデルの整合性や両モデルの保守性向上が期待できる。また、分析設計モデルに必要な情報を追記することで、可用性、信頼性や資源消費量等の分析も実施できる。

文献 [Procter2015] で提案している STPA のプロセスでは、人手による AADL モデルに基づく人手による STPA と、AADL モデルに基づく安全解析支援を組み合わせている。この相補的プロセスを用いることで、AADL モデルに記述されていない（想定の少し外側の）事象を AADL モデルに取り込めることが期待できる。また、特に複雑化した AADL モデルに対して、人間の気づき漏れといった点を機械が指摘してくれることも期待できる。

5.2. SMT ソルバーによる支援事例

本節では、SAT/SMT ソルバーを用いた STPA 支援の事例として、STPA Step1 分析結果の縮約と整合性検証を紹介する。初めに近年様々な手法のバックエンドで用いられている SAT/SMT について解説し、合わせてそれらがどのように用いられるかを紹介する。次に SAT/SMT ソルバーによる STAMP/STPA 支援事例として、John Thomas 氏による形式化された Step1 分析結果の縮約による Step2 支援、および Step1 分析結果の整合性検証を紹介する。

SAT/SMT ソルバーは近年の高速化に伴い、有界モデル検査やテストケース生成といった処理のバックエンドで利用されている。また、本節で紹介するように、SAT/SMT ソルバーへの入力を工夫することで分析結果の抽象化や整合性検証にも活用できる。

5.2.1.SAT/SMT ソルバーの解説

SAT（Satisfiability）は命題論理式に対する充足可能性判定問題のクラスを表し、SAT ソルバーは SAT 問題を解くツールである [梅村 2010]。 SAT ソルバーの入力は命題論理式であり、入力の命題論理式が充足可能（satisfiable）であれば充足であることと解（入力された論理式に含まれる命題変数への真偽割当）を出力し、充足不能（unsatisfiable）であれば充足不能であることを出力する。ここで、充足可能性を判定したい命題論理式 F に含まれる命題変数に対する真偽割当で F を真にするものが存在するとき、F は充足可能であるといい、そうでないとき F は充足不能であるという。

SAT ソルバーの入出力の例を以下に示す：

- 入力：$p_1 \vee p_2$（p_1 OR p_2）　➔ 出力：充足可能：p_1= 真 , p_2= 偽
- 入力：$p_1 \wedge \neg p_1$（p_1¬ AND（NOT p_1））➔ 出力：充足不能

一つ目の例の入力 $p_1 \vee p_2$ に対しては、p_1= 偽 ,p_2= 真と p_1= 真 ,p_2= 真も $p_1 \vee p_2$ を真にする割当であるが、SAT ソルバーは $p_1 \vee p_2$ を真にする割当の中から一つだけ出力する。また出力が充足不能であるときにも、後述する MUS 列挙を用いることで、例えば FT 図から最小カットセットを求めることができる。

命題論理式の集合に対する充足可能性は、集合に含まれる論理式たちの論理積に対する充足可能性として定義される。今、命題論理式の集合 S が充足不能であるとする。このとき、S の部分集合 A が S の極小矛盾集合（Minimal Unsatisfiable Set，MUS）であるとは、A は充足不能でありかつ A の真部分集合は全て充足可能であることをいう [Biere2009]。 直感的には、MUS A は S が充足不能になる本質的な（必要最小限な）要因である。また、ある S に対して、一般に複数の MUS が存在する。

次の例を用いて具体的に MUS を求める：

論理式の集合 $\{p_1、\neg p_1、\neg p_1 \vee p_2、\neg p_2、\neg p_1 \vee p_3、\neg p_3\}$

上の集合に対する充足可能性は、以下の論理式 F の充足可能性と同等である：

$F=（p_1）\wedge（\neg p_1）\wedge（\neg p_1 \vee p_2）\wedge（\neg p_2）\wedge（\neg p_1 \vee p_3）\wedge（\neg p_3）$

このとき、F の MUS たちは以下の通りとなる：

$\{p_1, \neg p_1\}, \{p_1, \neg p_1 \vee p_2, \neg p_2\}, \{p_1, \neg p_1 \vee p_3, \neg p_3\}.$

一つ目の MUS に含まれる要素は二つであるため、この集合に真に含まれる部分集合で充足不能な集合は存在しない。二つ目と三つ目の MUS は三要素を含むため、三要素を全て考えると充足不能であるが、どの二要素の組み合わせも充足可能である。

SMT ソルバー利用者の観点から SMT について簡潔に解説する。SMT の詳細な解説は、文献 [岩沼 2010] などを参照されたい。SMT は Satisfiability Modulo Theories の略であり、SMT ソルバーは入力として一階述語論理式の一部が取れ、入力された論理式を真にする論理式中の変数記号や関数記号への解釈（例：変数記号 x を定数 1 であると解釈する）が存在すれば、充足可能であることと各解釈を出力し、そのような解釈が存在しなければ充足不能であることを出力する。例えば、一階述語論理式 "x+y=10 ∧ x>0 ∧ y<0" を入力すると、充足可能であることと x=11, y=-1 を出力する。SAT ソルバーと同様に、SMT ソルバーも解釈を 1 つ返すだけであり、一般には複数の解釈がありうる。

SMT に対しても SAT と同様に MUS を定義できる。例えば一階述語論理式の以下の集合

$$\{x > 2, x < 1, x < 0, (x + y > 0 \vee y < 0),\ (y >= 0 \vee x >= 0), (y < 0 \vee x < 0), (y > 0 \vee x < 0)\}$$

に対する MUS たちは、以下の三つの集合となる：

$\{x > 2, x < 0\}$, $\{x > 2, (y < 0 \vee x < 0), (y > 0 \vee x < 0)\}$, $\{x > 2, x < 1\}$.

5.2.2. 事例紹介：分析結果の縮約と整合性検証

本項では、文献 [Thomas2013] 内の形式化された STPA に対して提案されている STPA 分析結果の縮約および整合性検証を SMT ソルバーにより実現した事例を紹介する。初めに文献 [Thomas2013] で紹介されている STPA の形式化を解説する。次に、形式化された STPA の分析結果を縮約する方法として doesn't matter の除去について解説し、その縮約の SMT ソルバーによる実現事例を紹介する。最後に、形式化された分析結果の整合性検証について解説し、その整合性検証の SMT ソルバーによる実現事例を紹介する。

文献 [Thomas2013] における形式化

Thomas 氏は文献 [Thomas2013] において、STAMP のコントロールアクションと STPA Step1 のガイドワードを組み合わせた UCA 候補を四つ組（SC,T,CA,Co）として形式的に定義している。ここで、CA はコントロールアクション、SC は CA を出力するコントローラー、T は CA のタイプ { 与えられるとハザード（Provided）, 与えられないとハザード（NotProvided）} をそれぞれ表す。また Co は CA が T となったコンテキストを表す。さらに、Co はプロセス変数 p のその値 v の組 (p,v) の集まりとして定義される。運行中のバスの乗務員によるドア開閉制御を題材としたときの UCA 候補の例を以下に示す。

「バスが移動中（p_1,v_1）」かつ「乗客がドア付近にいる（p_2,v_2）」とき、
「乗務員（SC）」が「ドア開命令（CA）」を「与える（T）」

この UCA 候補では、SC は「乗務員」、T は「与える」、CA は「ドア開命令」であり、Co は「バスが移動中」かつ「乗客がドア付近にいる」である。さらに Co は、プロセス変数「バスの状態」と値「移動中」の組と「乗客不在検知センサー」と「検知」に分解される。

この形式化により、Step1 は各（SC,T,CA,Co）に対しハザード H へ至るか否かを判定することとなる。この判定を行うには、STPA Step1 は各（SC,T,CA,Co,H）に対し、ハザー

ドへ至る（Yes）あるいは至らない（No）を判定した表を定義すればよい。形式化された STPA の結果を表 5.2 1 に示す。なお簡略化のため、SC（乗務員）は省略した。また T は「与えられるとハザード」のみを考えることとし、T は第 4 列の項目名として記載した。

この表を定義することは、各（SC,T,CA,Co,H）を入力として {Yes,No} を出力する関数を定義することと同等である。（この関数を fstep1 と呼ぶ）なお、文献 [Thomas2013] では T を関数名に含めることで、「与えるとハザード」に対応する HP（H,SC,CA,Co）と「与えないとハザード」に対応する HNP（H,SC,CA,Co）の二つの関数（表）を用いているが、本節では T も含めた（SC,T,CA,Co,H）を引数とした。

表 5.2-1　H（SC,CA,T,Co,H）の表形式での表現（一部）

	コンテキスト		
コントロールアクション	バスの状態	乗客不在検知センサー	与えられるとハザード？
ドア開命令	移動中	検知	Yes
ドア開命令	移動中	未検知	Yes
ドア開命令	停留所停止中	検知	No
ドア開命令	停留所停止中	未検知	Yes

STPA Step1 分析結果の縮約

各プロセス変数の取りうる値が二つであったとしても、プロセス変数の個数が増えるにつれ、形式化された STPA Step1 の結果として得られる表（表 5.2 1）のサイズ（行数）は指数的に増大する。したがって、この表に含まれるすべての行に対し、STPA Step2 を実施することは大変コストがかかる作業となる。そこで、この表を（本質的に情報を失うことなく）縮約できれば、Step2 の作業量を低減できる。また、表を完成させる前に、部分的な表を縮約できれば、優先度の高い対象から先に Step2 を実施できるようになる。

このような表の縮約例として、文献 [Thomas2013] では、“doesn’t matter” セルの除去を提案している。例えば、各プロセス変数の取りうる値が二つであると、10 個のプロセス変数を持つことになる UCA 候補がハザードへ至るか否かを検討しているとき、実は 6 個のプロセス変数が特定の値であれば、他の変数の値がどのような値であってもハザードへ至ることが分析結果から分かったとする。このとき、この 6 個のプロセス変数の値こそがハザードへ至る本質的要因であり、残りの 4 個のプロセス変数は気にする必要が無い（doesn’t matter）といえる。すると、残りの 4 個のプロセス変数分の行をまとめることで、24 行が 1 行に縮約できる。また、6 個のプロセス変数が本質的であることが分かれば、分析者はそれらのプロセス変数に関する個所の分析に集中できる。

前述した表の縮約は SAT/SMT の MUS 列挙問題に帰着できる。またここで紹介する縮約法は、分析が未完了な表（あるいは表の一部分）に対しても適用可能である。未完了な表はいくつかの行で分析結果が不明であるが、本節で提案する MUS 列挙問題では、そのような行に対してはハザードへ至らないと仮定して縮約を実行する。言い換えれば、確実に縮約できる行たちのみを縮約する。提案する縮約法では、表の各行を一つ

の SMT ソルバーに対する制約条件として記述する。また変数 SC,T,CA,Co,H を用意し、それらが取りうる値を SMT ソルバーに対する制約条件として明示する。さらに、変数 SC,T,CA,Co に対してはハザード H へ至らないという条件を SMT ソルバーに対する制約条件 fstep1（SC,T,CA,Co,H）=no として記述する。

表にハザードへ至る結果を表す行が含まれている場合には、これらの制約条件は矛盾する。例えば、SC=sc1, T=t1, CA=ca1, Co=co1 のとき H=h1 へ至るという Step1 分析結果の行（fstep1（sc1,t1,ca1,co1,h1）=yes）だけが表に含まれる場合には、各変数の取りうる値を表す制約条件たち SC=sc1, T=t1, CA=ca1, Co=co1, H=h1 と制約条件 fstep1（SC,T,CA,Co,H）=no は充足不能である。この表にさらに制約条件 fstep1（sc1,t1,ca1,co2,h1）=yes も含まれているとすると、前出の充足不能となる制約条件の集合から Co=co1 を除いた集合だけで充足不能となる（ただし、変数 Co は co1 と co2 の二値であると仮定している）。このように、本節で定義した制約条件の集合の MUS は必ずハザードへ至るための必要最小限の変数とその値を提示するので、残りの変数は doesn't matter となる。

表 5.2-1 の縮約結果を以下に示す。表 5.2-1 に対しては、{ バスの状態＝移動中 }, { 乗客不在検知センサー＝未検知 } の二集合が MUS として列挙される。前者の場合，バスの状態＝移動中さえ決まっていれば、乗客不在検知センサーが検知でも未検知でもハザードへ至るため、乗客不在検知センサーは doesn't matter となり，表 5.2-1 の上 2 行を 1 行にまとめられる（表 5.2-2）。

表 5.2-2　MUS による分析結果の例

コントロールアクション	バスの状態	乗客不在検知センサー	与えられるとハザード?
ドア開命令	移動中	doesn't matter	Yes
ドア開命令	停留所停止中	検知	No
ドア開命令	停留所停止中	未検知	Yes

STPA Step1 分析結果の整合性検証

文献 [Thomas2013] では STPA Step1 の結果得られる表に対するある種の整合性検証を提案している。この検証は、同一のコンテキストの元で、与えても与えなくてもハザードへ至るコントロールアクションの有無の検証である。このようなコントロールアクションは指摘されたコンテキストの元で、与えた場合と与えない場合で同一のハザードを引き起こすときは設計上の不備があると考えられ、与えた場合と与えない場合で異なるハザードへ至るときは意味の無いコントロールアクションといえる。また前者の場合には、コントロールアクションを与えるか与えないかを判断するためのコンテキストが不十分であるために、ハザードを引き起こす可能性を排除しきれない可能性も考えられる。

文献 [Thomas2013] では、この検証で示されているコントロールアクションが存在しないことを、以下の論理式（*）が成立することとして形式化されている：

$$\forall H1 \in H, H2 \in H, SC \in SC, CA \in CA(SC), Co \in Co(SC)$$
$$HP(H1, SC, CA, Co) \Rightarrow \neg HNP(H2, SC, CA, Co) \quad (*)$$

上の論理式が不成立になる時には、ある CA,Co,H1,H2,SC が存在し、CA が同じ Co の元で、与えられるとハザードのときはハザード H1 へ至り、与えられないとハザードのときはハザード H2 へ至るときである。

論理式（*）が不成立となることは、以下の論理式が充足可能であることと同等である：

$$H（SC,Providing,CA,Co,H1）\wedge H（SC,NotProviding,CA,Co,H2）$$

すなわち、上の論理式の充足可能性を判定することで、H1=H2 のときは設計の不備を、H1 ≠ H2 のときは意味の無い CA を検証できる。しかし、未完成な表に対しては、この論理式を充足させる解は一般に大量に出力される。これは、H（SC,Providing,CA,Co,H1），H（SC,NotProviding,CA,Co,H2）が公理（必ず満たす必要がある制約条件）に含まれず、かつそれらの否定も公理に含まれない場合に、SAT ソルバーは充足可能であると出力するためである。

H（SC,Providing,CA,Co,H1）と H（SC,NotProviding,CA,Co,H2）の両方の式が公理に含まれる場合にのみ通知させるためには、以下の論理式が充足不能であることを確認すればよい：

$$\neg H（SC,Providing,CA,Co,H1）\vee \neg H（SC,NotProviding,CA,Co,H2）$$

上の論理式の充足不能性を判定することで、不完全な表に対しても本来知りたい結果のみが得られる。

5.2.3. まとめ

本節では、SAT/SMT ソルバーによる STPA の支援事例として、文献 [Thomas2013] で提案されている Step1 分析結果の縮約と整合性検証を紹介した。初めに SAT/SMT の基礎的な定義について解説した。次に STPA 支援事例として、文献 [Thomas2013] で提案された形式化された STPA、及び Step1 の分析結果である表に対する縮約と整合性検証を紹介し、その縮約と整合性検証の SMT による実現例を紹介した。

本節の入力となる UCA の表（表 5.2-1）の最後列の Yes, No は人手により識別される。今後の課題としては、前節で紹介した AADL による検証可能モデルと組み合わせることで、この Yes,No 識別を支援することが挙げられる。

6. 第 1 回 STAMP ワークショップ in Japan について

2016 年 12 月 5 日（月）、6 日（火）、7 日（水）の 3 日間、九州大学稲盛財団記念館、九州大学西新プラザにおいて第 1 回 STAMP ワークショップ in Japan を開催し [IPA2016-2]、一般講演が 16 件で、約 130 名という多くの講演者・参加者を得た（図 6.1-1）。ヨーロッパでの STAMP ワークショップが数十人の参加者から始まり、数年の継続開催を経て 100 人以上が参加するようになったことから、第 1 回 STAMP ワークショップ in Japan は予想を上回る盛況ぶりであった [IPA2016-3]。

今回のワークショップでは、一般講演者、参加者ともに多様な業界からの参加があり、日本における STAMP に対する関心・期待の急激な、そして大いなる盛り上がりが感じられた。

6.1. STAMP Workshop のプログラム

1 日目：MIT の Dr. John Thomas による STPA チュートリアル（初級）[Thomas2016]、STPA チュートリアル（中級）[Thomas2016-2]、STPA 事例研究 [Thomas2016-3] の講演が行われた。
続けて、4 件の招待講演が行われた。招待講演は STAMP 関連が 1 件、形式手法関連が 2 件、その他 1 件。

2 日目：招待講演では、STAMP と並び今後の安全性解析に有効性が期待され、また、STAMP との組み合わせも期待されるレジリエンスエンジニアリングについて 2 件の講演が行われ、その後、ショート講演セッション、一般講演セッション A、B、C で計 11 件の発表が行われた。

3 日目：一般講演セッション D、E で計 5 件の発表が行われ、最後のクロージングでは、次回以降の STAMP ワークショップ in Japan 開催について議論が行われた。

一般講演は表 6.1-1 の通り、合計 16 件が発表された。

図 6.1-1 第 1 回 STAMP ワークショップ in Japan でのチュートリアル風景

表 6.1-1　第 1 回 STAMP Workshop in Japan の一般講演プログラム

STAMP ショート講演セッション **①農業用ハウス制御における STAMP モデリングの試行（長崎県立大学）** **②組織内ネットワークのセキュリティ被害軽減対策における STAMP モデリングの試行（長崎県立大学）**
STAMP 一般講演セッション A ③複雑システムの安全設計のための発想法（会津大学） ④ ET ロボコンにおける STAMP/STPA の試行およびウエブベース STPA ツールの設計と開発（日本大学） ⑤ ET ロボコン走行体システムへの STAMP/STPA 適用事例の紹介（仙台高等専門学校）
STAMP 一般講演セッション B ⑥水上パーソナルビークル MINAMO を題材にした安全性解析（首都大学東京） ⑦水上セグウェイみなもの STAMP/STPA 分析／運用組織の視点からのハザード分析（会津大学） ⑧社員証型センサーを用いた健康増進システムへの STAMP/STPA の適用検討（愛知工業大学）
STAMP 一般講演セッション C ⑨倒立二輪車の人間・機械協調制御システムの STAMP 解析（IPA/SEC） ⑩業務系システムへの STAMP/STPA 適用事例（日本電気株式会社） ⑪制御システム以外の組込みシステムのソフトウェア障害に対する STAMP 適用の試み（日本電気通信システム株式会社）
STAMP 一般講演セッション D ⑫駅構内論理装置の踏切制御機能仕様に対する STAMP/STPA 解析（JR 東日本） ⑬人と組織に関する HCF ヒントワード提案と事例適用（IPA/SEC） ⑭閉電路制御式踏切システムの安全性評価（株式会社 京三製作所）
STAMP 一般講演セッション E ⑮ UCA 抽出における Extending STPA の試行事例（日本ユニシス株式会社） ⑯ STAMP/STPA におけるモデル検査の利用（日本ユニシス株式会社）

6.2. チュートリアル

John Thomas 氏による初級者向けチュートリアルの STAMP/STPA Beginner Introduction [Thomas2016] では、まず、

- 火星探査機、ボーイング 787 バッテリー火災、テスラ自動車事故を例示し、機器故障がなくても事故につながっており、新たな視点での分析が必要である
- Ford のリコールなどを例示し、ほとんどのソフトウェア関連事故は要求仕様の問題に帰結することができる
- 中国航空機の着陸事故などを例示し、人との相互作用をシステム視点で捉えると、操作ミスは原因ではなく結果としての現象であることを紹介し、新しい安全性分析の考え方としての STAMP が概説された。その後、自動車の Shift by Wire（電動ギヤシフト装置）を題材として参加者自身が手を動かしてコントロールストラクチャー（図 6.2-1）を描き、手順に沿って STPA 分析の演習を行った。

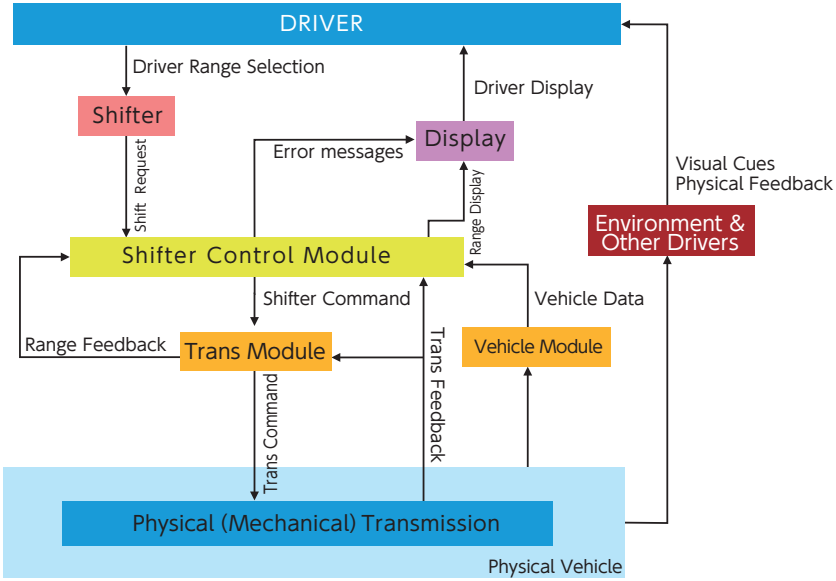


図 6.2-1　Shift by Wire のコントロールストラクチャー

- Step1 で UCA を抽出する際には、次の 4 つの構成要素で UCA 文言を作る
  [制御を実行するコンポーネント] ＋ [制御を与えるか否か] ＋ [制御行動] ＋ [実行条件]
- Step2 で HCF を導出する際には、まず（Step2A）何があったらアクシデントに至るかを考え、次に（Step2B）どういう要因で非安全制御行動になるのかを考える

ということがアドバイスされた。

中級者向けチュートリアルの STAMP/STPA Intermediate Tutorial[Thomas2016-2] では、化学プラントを題材として、Step0 から Step2 まですべてやりきるという演習を行った。ここでは、初級者向けのアドバイスに加え、安全制約導出までを行うこと、表にして UCA とハザード、安全制約のトレーサビリティーを確保することがアドバイスされた。

STPA 事例研究 [Thomas2016-3] の講演では、米国自動車メーカー（GM）と MIT が共同で行った自動車の APA（Automated Parking Assist：自動駐車支援機能）に対する STPA 分析結果が紹介された。その中では、APA の自動化レベルが上がると、実はリスク要因の数は増大するという驚くべき結果が示された（表 6.2-1）。それも、人の操作に関する UCA（Driver UCAs）、コンピューターの制御に関する UCA（APA Computer UCAs）のいずれもが増大するというものである。この分析結果は、自動車の自動運転に対する安全性分析には STAMP を使わなければならない、という警鐘にも成り得る。また、多くの STPA 分析者がどう表現すべきか悩んでいた " 人のモデル（Human Controller Model）" が、その作成手順の解説と共に具体的に示され（図 6.2-2）、とても得るものが多い講演であった。

表 6.2-1　自動駐車支援機能の自動化レベルと UCA の数

	Level 1 “Driver Assistance”	Level 2a “Partial Automation”	Level 2b “Partial Automation”	Level 3 “Conditional Automation”
Driver UCAs	35 in common		32 in common	
	42	41	38	44
		30 in common		
APA Computer UCAs	5 in common		28 in common	
	5	13	28	28
		13 in common		
Total	40 in common		60 in common	
	47	54	66	72
		43 in common		

NEW PROCESS

- Identify UCAs
- Identify Process Model variables
- Identify Process Model Flaws
- Identify flaws in Process Model Updates
- Identify unsafe decisions (Control Action Selections)

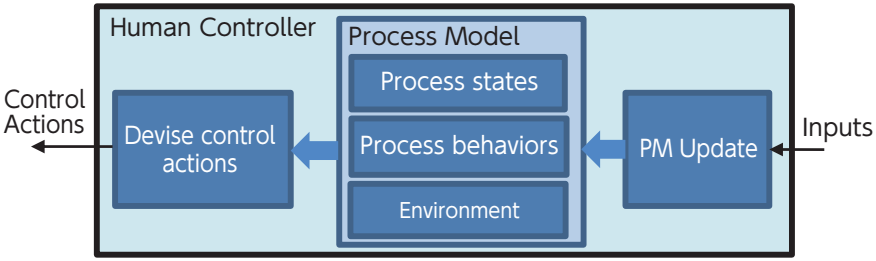


図 6.2-2　Human Controller Model 作成手順

STAMP ワークショップでは、MIT でのワークショップでも、ヨーロッパでのワークショップでも必ずチュートリアルが行われる。Thomas 氏も、必ずチュートリアルを実施すべきと強調していた。特に、STAMP は自分自身で実際に手を動かしてやってみないと理解できないものなので、演習付きの hands-on は必須であろう。

今後、STAMP ワークショップ in Japan を継続的に開催し、日本での STAMP 普及促進につなげたい。そして、そこでは海外を含めた豊富なチュートリアルが組み合わせて行われることを期待する。

John Thomas 氏のチュートリアル、基調講演をはじめ、本ワークショップでの講演資料を IPA/SEC の Web サイトで公開しているのでぜひ参照し、役立てて欲しい。

第 1 回 STAMP Workshop in Japan　IPA/SEC Web サイト

（日本語）http：//www.ipa.go.jp/sec/events/20161205.html

（英語）http：//www.ipa.go.jp/english/sec/complex_systems/stamp_workshop-1.html

講演資料のダウンロードも可能。

Column　STPA-Sec　セキュリティーへの STPA 適用

STAMP ／ STPA は安全性解析手法であり、セーフティーを中心に展開されてきたが、STPA の特徴はセキュリティー上のリスク分析にも適用可能である。そこで 2016 年 3 月 STAMP workshop でのチュートリアル資料（[Young2016]、[Young2016-2]）に基づいて、サイバーセキュリティーなどに特化した STPA-Sec（STPA for Security）について紹介する。

①既存のアプローチと比較した特徴

STPA は全体を俯瞰してトップダウンに分析する手法であり、STPA-Sec も同様である。従来のセキュリティーエンジニアリングが物理的な構造・機能（図 C-1 の青の楕円部分）にフォーカスしていたのに対して、STPA-Sec はより広範囲で概念的な機能・目的（図 C-1 の緑の楕円部分）をも対象とする。

また、STPA は手段（How）に着目した手法ではなく What に着目した手法である。STPA-Sec も同様に従来のセキュリティー要求分析手法であるアタックツリーやミスユースケースのようにどのような脅威があるのかを洗い出す手段（How）ではなく、攻撃から何を守るべきか（What）を明確にするアプローチである。

セキュリティー対策は現状、保守・運用段階での脆弱性対処が中心である。しかし、多様な機器・システムが複雑につながる IoT 時代には何がセキュリティーを確保する際の問題となるのかを事前に把握し対処できることがより重要になってきている。そのため STPA-Sec のアプローチが今後注目される。

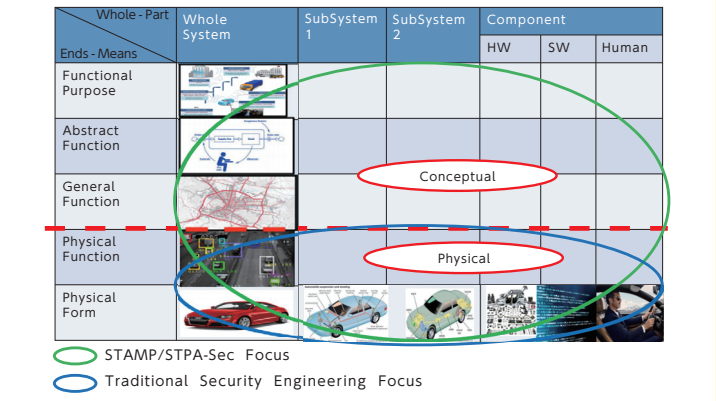


図 C-1 STPA-Sec のフォーカスエリア

② STPA-Sec 分析手順

STPA-Sec の分析手順では、STPA の分析手順に対し、非安全と対になるように、セキュアでない（Unsecure）の視点を追加している。つまり、安全でない状態を考えるのと同様に、セキュアでない状態を考慮することになる。また、非安全なコントロールの原因の特定に際し、セキュアでないコントロールアクションを導くシナリオを識別し、影響度を踏まえ、よりクリティカルなコントロー

ル戦略を選択する。

STPA と同様 STPA-Sec でも、Step2 における具体的で標準的な手法・手順はないが、「守るべき情報が、何を元に生成・変更されるのかを追跡する」という情報ライフサイクルのトレースは重要である。図 C-2 では、STPA に対する追加手順を赤字で示した。図 C-2 は STPA-sec の標準的な手順ではなく、分析手順の一例である。

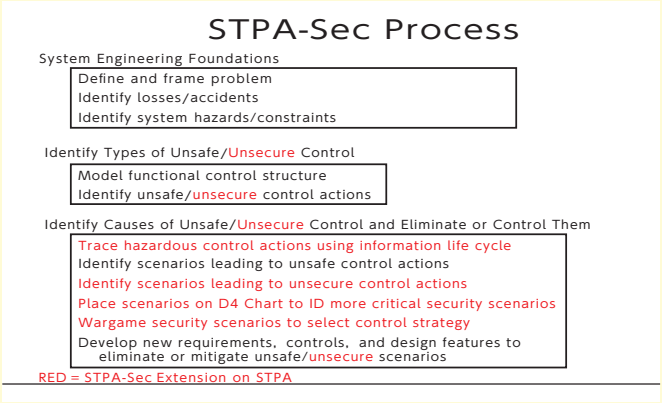


図 C-2 STPA-Sec の分析手順の例 [Young2016]

③ STPA と STPA-Sec との違いは？

STPA と STPA-Sec の分析手順は基本的には変わらない。ただし、セキュリティー上の脅威抽出に必要な分析の視点が追加される。要因の特定に関して図 C-3 に示すオレンジ色の部分が STPA-Sec での追加事項となる。コンポーネントに対する悪意ある、権限を持たない、部分的なインプットが脅威の原因に成り得るかを追加的に分析する。

STPA はセーフティーのリスクとしてハザード分析を行うが、STPA-Sec はセキュリティーのリスクとして脅威分析を行う。コントロールストラクチャーを共有することで、ハザード分析と脅威分析のゴールを統合的に考慮していくことが示唆されており、今後の普及が望まれる。

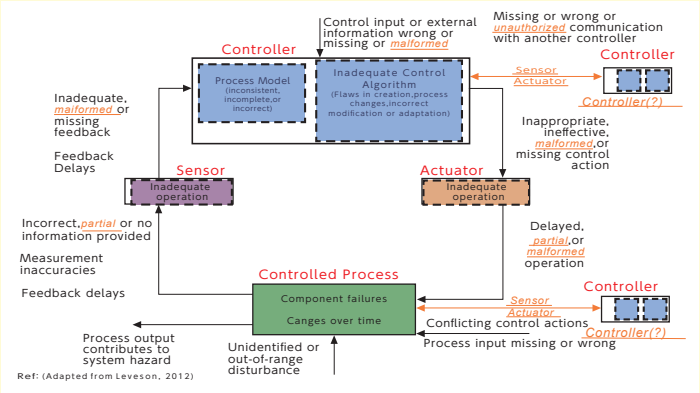


図 C-3 STPA-Sec のコントロールループ図における原因特定の例 [Young2016]

おわりに

STAMP 理論は、常に進化し続けている。本書は、「初めての STAMP/STPA」[IPA2016] の応用編である。ここに紹介するいくつかの事例は、どれも、Leveson 著 "Engineering a Safer World: Applying Systems Thinking to Safety"[Leveson2012] に例示されているような安全制御構造とは異なっている。

たとえば、第 2 章「制御システム解析事例」における安全制御構造にはフィードバックがほとんど存在していない。フィードバックが無ければ適切な制御が行われているかどうかをコントローラーが知ることはできないが、ここに挙げられた事例には、そのようなフィードバックが無いことがむしろ安全上の強みになっているものもある。また、第 3 章「エンタープライズ系システム解析事例」では、識別された 2 つのコントロールアクションが、別々のアクションのままでは安全ではなく、1 つの不可分なアクションにまとめられる必要があるという、興味深い対策が提示された。

2.1 節「制御システム解説事例」の事例として、踏切のとりこ検知システムを取り上げた。このシステムは、列車や踏切制御システムに対するフィードバックを持たないオープンループシステムである。オープンループ故の簡便さやローコスト性により、多くの踏切に対応でき、広く普及することになった。つまり、フィードバックを持たないことが安全の推進力となったとさえ言えるのである。同様のシステムとして、自動車の緊急ブレーキシステムがある。当初ドライバの意思を無視して自動停止するという挙動を問題視する意見もあったが、そのシンプルさゆえに、今や多くのドライバに受け入れられ、安全性の向上に大きく貢献している。

フィードバックを持たないことは一般的に安全の阻害要因であるが、その逆にもなり得る。この章における分析結果は、その両面を考えるきっかけになるであろう。

第 3 章「エンタープライズ系システム解析事例」は、さらに進んだ分析事例である。STAMP も含め、従来の安全解析手法は、全体から個別へと分解するプロセスを踏んで行われる。一旦分割された安全制御アクションを、一歩下がって結合し、この章の結論にみられるような安全対策を考案することは、「解析・分析」というよりも、「統合」に近い。STAMP の実施において、我々が陥りやすい間違いは、識別した制御構造をひたすら詳細化してゆくだけに終始してしまうことである。本書第 1 章の図 1-2 は、分析と統合がぐるぐると輪廻するプロセスモデルを示しているが、このような複合的な知的活動こそが、我々の目指す姿と言えるのである。

本書を読んだ読者は、おそらく、「立て板に水」のごとく、定石通りに分析を進めるだけではこれらの事例の安全化を達成できないことを知ることになると思われる。それこそが、執筆者一同のメッセージと言っても過言ではない。第 4 章で試みられる Hazard Causal Factor を導くためのヒントワードの検討や、第 5 章に示される AADL との統合は、こうした「定石通りではない分析」のためのツールである。これらのツールは、STPA の手順上、Step2 に相当する部分を、よりクリエイティブに行うために提案されている。実際の STAMP の実施現場では、このような工夫は必ず必要となる。実際、Leveson の書いた "An STPA primer"[Leveson2013] には、以下の記述がある。

"Step 2 requires the most thought and prior experience by the analyst and there is, so far, much less help provided compared to Step 1."

Leveson 自身が書いているように、アナリストの洞察力や多様な経験を要する Step2 の分析方法は、実は STPA 自身が提供するところは少ない（！）。そこは、我々アナリストが自ら工夫し、そして経験を積んでゆかなければならない。そのような工夫によってこそ、STAMP 理論は進化し続けることができる。本書に示したような応用問題が、共に新たな工夫を生み出してゆくきっかけになることを願うものである。

参考文献

［1］［IPA2016］
「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～」, IPA, http://www.ipa.go.jp/sec/reports/20160428.html, 2016

［2］［IPA2016-2］
第 1 回 STAMP ワークショップ in Japan、
（日本語）http://www.ipa.go.jp/sec/events/20161205.html、
（英語）http://www.ipa.go.jp/english/sec/complex_systems/stamp_workshop-1.html

［3］［IPA2016-3］
実施報告 第 1 回 STAMP ワークショップ in Japan、
（日本語）http://www.ipa.go.jp/files/000056629.pdf、
（英語）http://www.ipa.go.jp/files/000057872.pdf

［4］［IPA2016-4］
「情報処理システム高信頼化教訓集（組込みシステム編）」2015 年度版、IPA、http://www.ipa.go.jp/sec/reports/20160331_2.html

［5］［Leveson2012］
Nancy G. Leveson:Engineering a Safer World: Systems Thinking Applied to Safety（Engineering Systems）, The MIT Press, 2012.

［6］［Leveson2013］
“An STPA Primer v1”, Nancy Leveson, et al, 2013

［7］［Procter2015］
Using STPA to Support Risk Management for Interoperable Medical Systems, Sam Procter et al., 2015

［8］［SAE2012］
SAE International, AS5506 - Architecture Analysis and Design Language（AADL）, 2012.

［9］［SAE2014］
SAE International, AADL Error Model Annex,（Standards Document AS5506/1, 2006. in revision as Document AS5506/3 2014, 2014.

［10］［OSATE2016］
OSATE, Carnegie Mellon University, http://osate.org/index.html,2016

［11］［Feiler2012］
Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis & Design Language, Peter H. Feiler and David P. Gluch, Addison-Wesley Professional, 2012

［12］［Delange2014］
Architecture Fault Modeling with the AADL Error-Model Annex, Julien Delange and Peter Feiler, 2014

［13］［Delange2014-2］
AADL Fault Modeling and Analysis Within an ARP4761 Safety Assessment, Julien Delange, Peter Feiler, David P. Gluch and John Hudak, 2014

［14］［梅村 2010］
梅村 晃広 ,SAT ソルバ・SMT ソルバの技術と応用 , コンピュータソフトウェア , Vol.27, No.3（2010）, pp.24–35, 2013

［15］［岩沼 2010］
岩沼宏治、鍋島英知 : SMT : 個別理論を取り扱う SAT 技術、人工知能学会誌、25 巻 1 号、pp.86-95，2010
［16］［Biere2009］
Handbook of Satisifiability, Armin Biere, Marijin Heule, Hans van Maaren and Toby Walsh, IOS Press, 2009
［17］［Thomas2013］
EXTENDING AND AUTOMATING A SYSTEMS-THEORETIC HAZARD ANALYSIS FOR REQUIREMENTS GENERATION AND ANALYSIS, John Thomas, 2013
［18］［Thomas2016］
STAMP/STPA Beginner Introduction、http://www.ipa.go.jp/files/000056812.pdf
［19］［Thomas2016-2］
STAMP/STPA intermediate Tutorial、http://www.ipa.go.jp/files/000056813.pdf
［20］［Thomas2016-3］
STPA Applied to Automated Parking Assist、http://www.ipa.go.jp/files/000056811.pdf
［21］［Young2016］
William Young Jr, Security Tutorial Part 1　A Systems Approach to Security, 5th STAMP Workshop in BOSTON,
http://psas.scripts.mit.edu/home/wp-content/uploads/2016/01/SECURITY-Tutorial2016.pdf
［22］［Young2016-2］
William Young Jr, Understanding STPA-Sec Through a Simple Roller Coaster Example、
http://psas.scripts.mit.edu/home/wp-content/uploads/2016/01/24-BillYoung-W2016.pdf

索引

付録
A）用語説明

Accident
　望ましくない事象、事故・損失。アクシデント。

Hazard
　アクシデントが潜在している具体的な状態。ハザード。

UCA（Unsafe Control Action）
　非安全制御行動。事故・損失（アクシデント）につながる制御、制御行動、動作。

HCF（Hazard Causal Factor）
　ハザード誘発要因。危険な状態（ハザード）を引き起こす原因。

コントロールアクション（Control Action）
　コントローラーが被コントロールプロセスに対して行なう制御指示・制御動作・制御行動。

コントロールストラクチャー（Control Structure Diagram）
　制御構造図。システムにおいて、安全制約の実現に関係するコンポーネント、およびコンポーネント間の相互作用から成る構造図。

コントロールループ（Control Loop）
　コントローラー、被コントロールプロセス、コントロールアクション、フィードバックから成る循環関係。

プロセスモデル
　コントローラーが想定する被コントロールプロセスの状態。

4 種類のガイドワード
　非安全コントロールアクション（UCA）の分類であって、UCA を抽出する際のヒントとなる、次の 4 つの言葉を指す。
　✧（**与えられないとハザード：Not Providing**）安全のために必要とされるコントロールアクションが与えられないことがハザードにつながる。
　✧（**与えられるとハザード：Providing causes hazard**）ハザードにつながる非安全なコントロールアクションが与えられる。
　✧（**早過ぎ、遅過ぎ、誤順序でハザード：Too early/too late, wrong order causes hazard**）安全のためのものであり得るコントロールアクションが、早すぎて、遅すぎて、または順序通りに与えられないことでハザードにつながる。
　✧（**早過ぎる停止、長過ぎる適用でハザード：Stopping too soon/applying too long causes hazard**）（連続的、または非離散的なコントロールアクションにおいて）安全のためのコントロールアクションの停止が早すぎる、もしくは適用が長すぎることがハザードにつながる。

本書ではガイドワードという表現を用いているが、最近の論文ではガイドワードという表現を避け、タイプという表現を用いることもある。
上記 4 つの言葉は、対象ドメインや粒度によっては適切とは言えない場合もある。また、分類としても網羅性はあるものの排他性に欠けるという性質がある。
よって、ガイドワードという表現や、4 つの言葉の意味にあまり捉われないこと。

分析・解析

「分析」と「解析」の使い分けには例えば次のような定義もある。
分析：見えるものを分けるのが分析
解析：（安全性のように）見えないものを分けるのが解析
しかし、数値や数式は見えるが「数値解析」、ハザードは見えないが「ハザード分析」のように上記定義に当てはまらない一般的な慣用句が数多くある。つまり、「分析」と「解析」の使い分けに関して汎用的で適当な定義は無い。
そこで、本書では「分析」、「解析」のいずれを用いても誤解なく意味が通じるところでは「解析」を用い、「分析」が一般的なところでは「分析」を用いている。

編著者（敬称略）

システム安全性・信頼性分析手法 WG（主査、副主査、以下 50 音順）

主査	兼本 茂	公立大学法人会津大学
副主査	日下部 茂	公立大学法人長崎県立大学
	荒木 啓二郎	国立大学法人九州大学
	大原 衛	地方独立行政法人東京都立産業技術研究センター
	岡本 圭史	独立行政法人国立高等専門学校機構 仙台高等専門学校
	川野 卓	東日本旅客鉄道株式会社
	北村 知	東日本旅客鉄道株式会社
	中村 洋	株式会社レンタコーチ
	野本 秀樹	有人宇宙システム株式会社
	向山 輝	日本電気株式会社
オブザーバー		
	国藤 隆	東日本旅客鉄道株式会社
	星野 伸行	有人宇宙システム株式会社

IPA/SEC（50 音順）

石井　正悟
金子　朋子
十山　圭介
松田　充弘
三縄　俊信
三原　幸博
八嶋　俊介

はじめての STAMP/STPA（実践編）
～システム思考に基づく新しい安全性解析手法～

2017 年 5 月　第 1 刷発行

発行　独立行政法人 情報処理推進機構（IPA）

〒 113-6591 東京都文京区本駒込 2-28-8
文京グリーンコート　センターオフィス 16 階

制作　株式会社サンワ　Printed in Japan

ISBN978-4-905318-51-4
C3055 ¥370E

9784905318514

定価：本体 370 円＋税

1923055003705

IPA 独立行政法人情報処理推進機構
Information-technology Promotion Agency, Japan

技術本部 ソフトウェア高信頼化センター
Software Reliability Enhancement Center(SEC)