STAMP ガイドブック

～システム思考による安全分析～

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

社会基盤センター
IT Knowledge Center on emerging tech trends

IoT システム安全性向上技術 WG
IoT System's Safety Enhancement Technique WG

Ver.1.0
2019 年 3 月

STAMP
IPA
STPA

はじめに

本書は、独立行政法人情報処理推進機構社会基盤センターの IoT システム安全性向上技術 WG における 2018 年度活動成果をまとめたものである。2015~2017 年度に作成した「はじめての STAMP/STPA」入門編、実践編、活用編は、これからのソフトウェア集約型の複雑システムに対応できる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きた。WG 活動の実践を通して学んできた手法 STAMP/STPA（Systems-Theoretic Accident Model and Processes/Systems-Theoretic Process Analysis）の理解は、独自に開発したツール STAMP Workbench の普及も含めて、十分に深まったといえよう。しかしながら、ビジネス分野でさらなる活用を目指すには、STAMP の枕詞である「システム理論」の意味をより深く理解し、目的に応じた柔軟な利用方法を考えることが大事である。そのために、今回、「STAMP ガイドブック～システム思考による安全分析～」という表題で WG の活動成果をまとめた。ここには、安全分析の背景にあるシステム思考の考え方に加えて、産業界にとって本来の有用性を感じられるいくつかの実践的な安全分析事例をまとめた。列車の保守作業管理システム、高齢者見守りサービスシステム、自動車の開発・販売・運用を含むライフサイクルの中に潜むリスク分析などである。さらに、事故原因究明の方法論である CAST（Causal Analysis using System Theory）の利用例もはじめて含めた。直接的な原因究明だけでなく、それを引き起こす背景要因や本質的な原因を探るための方法論として参考にしていただきたい。

周知のように、我々の日常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、近年、これがますます複雑化・知能化しつつある。IoT（Internet of Things）、AI（Artificial Intelligence）、SoS（System of Systems）というキーワードでこれらの製品のキー技術が表現される。一方で、既存の安全解析手法や安全規格は、このような「人と高度ソフトウェアを含み、しかも、互いにネットワークを介してつながるこれからの複雑システム」に対応できていないのも現実である。従来の集中統合型のシステムに比べて、多様な分野のベンダーが独立に開発したシステムの連携による分散処理型のシステムは、多様なサービスを提供し世の中を便利にすることが期待できるが、これは同時に、安全設計の想定漏れや安全責任のあいまいさ、セキュリティの脅威がセーフティ問題を誘発するといった問題を生み、事故が起こった後の後知恵による言い訳や責任の押し付け合いをも引き起こしてしまう。マサチューセッツ工科大学（MIT）の N. G. Leveson 教授の提唱する「旧来の安全解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ことが多い近年の複雑システムの安全解析には不十分である」という考え方をより深く理解し、これからのつながる社会での複雑システムのリスクや副作用を低減するために、STAMP をどのように使ってゆくかを考えるきっかけにしていただければ幸いである。

目次

図表目次

1. システム思考に基づく安全分析の本質

1.1. システム思考と STAMP/STPA

IoT システム安全性向上技術 WG では、2015 年以来、STAMP/STPA(Systems-Theoretic Accident Model and Processes/Systems-Theoretic Process Analysis)を中心に、複雑システム、特に、ソフトウェア集約型システムの安全分析や安全設計に関する調査と実問題への適用可能性の検討を行ってきた。その中での主要な議論の一つが、「複雑システムの安全分析における STAMP/STPA の利点は何か？」ということであった。提唱者の N. G. Leveson によると、「STAMP/STPA は従来の安全分析法に対するパラダイムシフトである」とか、「近年の複雑システムの事故はコンポーネント間のコミュニケーション・ミスマッチが引き起こしている」といった説明がなされているが、その本質的な意味は必ずしも正しく伝わっていないようにも思える。

従来のコンポーネント（ポンプや弁、電磁スイッチなど）の故障は、経年劣化により偶発的に起こされ、その故障モードも過去の経験からおおよそ明らかになっているのに対して、ソフトウェアの故障モードは多様であり、外的条件の偶発的な組み合わせにより発現することがほとんどである。このようなソフトウェア集約型の新しいシステムの安全分析を、従来法と同じ考え方だけで行うことには無理があるという主張は合理的でもある。そのため、N. G. Leveson は、ソフトウェア集約型の複雑システムの安全分析では、創発論（Emergence）に基づくシステミック・アプローチも必要であると主張している。信頼性工学における FTA や FMEA のような従来型の要素還元論に基づくシステマチック・アプローチへのアンチテーゼである。もちろん、STAMP/STPA が従来型の方法論に置き換わるということではなく、システミックとシステマチック両面のアプローチの組み合わせが重要であるということで、両者の優劣のみを比較することは誤解を生んでしまう。

これまで、STAMP/STPA では、非安全コントロールアクション（UCA）とそれを引き起こすハザードシナリオ・要因（HS、HCF）の妥当性が議論されることが多く、その結果得られる故障要因を従来法と比べて優劣が評価されてきた。しかしながら、「故障要因」だけに焦点を当てると STAMP/STPA の本質を見逃してしまいかねない。リストアップされた「故障要因」の評価には、必ず、後知恵というバイアスがかかり、どんな方法で考えても、最後には同じ結論に至るためである。しかしながら、STAMP/STPA の本質は、故障要因の抽出だけにあるのではなく、システミックな安全構造の把握にもある。そのためには、STAMP/STPA の略号にある「Systems-Theoretic」の意味をしっかりと理解し、その上で手法を使いこなしてゆくことが大事である。

本章では、この「Systems-Theoretic」の意味をもう一度考え直してみたい。本稿では、この「Systems-Theoretic」をシステム思考と表現することにするが、意味するところは同じである。この中で、特に焦点を当てるのは、複雑システムの安全は誰がどうやって確保するのかという方法論（安全設計の在り方）である[1]。

[1] 「システム思考」は、一般的には「問題となっている対象を、構造を持ったシステムとして捉え、問題解決を行おうとする考え方」といえる。システム全体の目的を明示化し、システムの構成要素の相互のつながりと関係づけてシステムの設計を行う。安全設計においては、システム全体の目的は事故を防ぐことであり、そのための防護策をシステムの構造の中に組み込むことが必要になる。

1.2. システム思考による安全分析とは

計算機技術の飛躍的な進歩によって、車の自動運転や多機能生活ロボットのような複雑でソフトウェア集約的（Software-intensive）な工学システムが日常生活に入り込んでいる。そこでは、より便利な機能を提供するために、人・システム外部環境・インターネットとコンピューターの間の複雑な相互作用が必然的に増えてくる。同時に、副作用として、この複雑な相互作用の欠陥や想定外の効果による事故も増えてくる。このような事故を事前に予測し、プロアクティブに安全設計に組み込むにはどうしたらよいであろうか？

このための方法論の一つとして、MIT の N. G. Leveson は、システム理論に基づく事故モデル（STAMP）とそれに基づくハザード分析法（STPA）を提唱している[Leveson2012]。従来の FTA（Fault Tree Analysis）、FMEA（Failure Mode and Effects Analysis）や HAZOP（Hazard and Operability Study）といったハザード分析法は、信頼性工学に基づくシステマチック・アプローチであり、対象システムをロジカルに分解し、それぞれの要素の故障要因を分析して、システムの事故を減らすことを目指すものであるが、これを補完する方法としてシステム全体をみたシステム思考のアプローチである STAMP/STPA の重要性を主張している訳である。著者はこのような経緯を図 1.2-1 のように可視化している。「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」という象徴的な表現で、従来の要素還元論的な故障分析法を補完する方法論として STAMP/STPA の必要性を説いている。言い方を変えると、ソフトウェア集約的なシステムでは、信頼性工学的方法論で全ての故障要因を拾い出してつぶすアプローチには限界があり、むしろ、故障の有無にかかわらずにクリティカルな危険状態を回避するという安全制御工学的方法論が必要ということである[兼本 2018]。

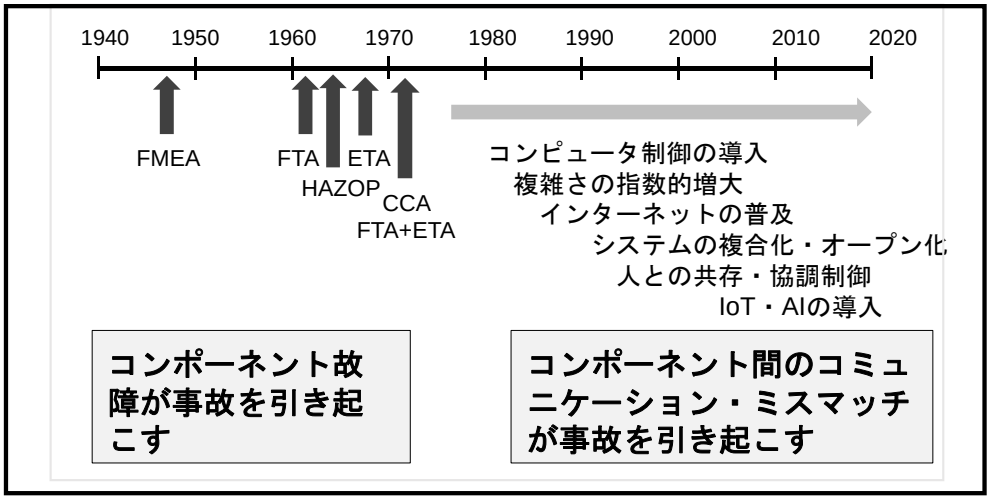


図 1.2-1　工学製品開発の進展と安全分析の時代変化

本章では、この STAMP/STPA の背景にあるシステム思考の考え方ならびに分析手順をいくつかの具体例を通して解説してゆくが、その前に、システム思考の大切さを示す二つの事例を示しておく。

ひとつはニューヨークの地下鉄の犯罪の低減事例である[割れ窓]。そこでは、的外れという批判には耳を貸さずに、まず地下鉄の落書き清掃作戦からはじめ、さらに、無賃乗車の撲滅にも取り

組んだ。その結果、それまでは見過ごされていた軽犯罪で逮捕された人の数は増えたものの、重罪事件は減っていった。この劇的な成功を支えたのは、犯罪学者のジョージ・ケリングが発案した「割れ窓理論」である。割れたまま放置された窓があっても誰も気にしなければ、まもなく他の窓も割れる。するとその無法状態の雰囲気が町中に伝わり、そこでは何でも許されるという信号を発しはじめ、より深刻な犯罪の呼び水になる。

もうひとつの安全分析の事例として 1986 年に起こったスペースシャトル・チャレンジャー号の打ち上げ直後の爆発事故を挙げる[失敗知識]。そこでは、ブースターロケットのジョイント部で使われていた O リングが、低温環境での使用で弾性が失われ、シール効果が不十分となって燃料が漏れ、これに炎がロケット下部から燃え移り爆発したと言われている。一見、単純な O リングの部品故障に思えるが、問題の根は深い。原因調査の中で、NASA および O リングを製作した会社が、低温における O リングの硬化の問題を予め知っていたこと、O リングの製作会社が当日の気象条件を見て、打ち上げを中止すべきとの意見を出していたにもかかわらず、打ち上げが強行されたことなどが判明した。ここでは、NASA 内での現場の技師と管理者との間の意志疎通が不十分であったことや、次年度予算を取るための圧力で、技術上のリスクを低く見積もりすぎていたという認知バイアスがあったことなどが指摘されている。これも、複数のコミュニケーションミスと機械の故障、人間や組織の判断ミスなどが複雑に絡まった結果発生した事故といえる。

これらの事例から、ソフトウェア集約型システムの安全分析に関して、以下のようなシステム思考の教訓が学べる。

(1) システム全体を俯瞰的に見て、システムの果たすべき本来の目的を理解し、さらには、各構成要素の複雑な相互作用を把握したうえで事故要因を考える。このためには、抽象化と階層化によるシステムの振る舞いの理解が必要である。
(2) 直接的な因果関係（相互作用）や、主要な根本原因だけでなく、間接的・非確定的でシステム全体に及ぶ相互作用の影響も共通原因として分析すべきである。
(3) これらの要因を事故後の後知恵による分析ではなく、プロアクティブに予測して安全設計に役立てる方法論に結びつけるべきである。
(4) システム全体の目的を達成するために、誰がどんな安全責任を持つかを明示化し、関係者の共通の理解を得ることも大事である。

1.3. STAMP/STPA の手順と背景にある考え方

ここでは、STAMP/STPA の手順を図 1.3-1 に沿って説明する[Leveson2018]。この手順そのものは簡単であるが、背景にある考え方を理解しておかないと期待する成果が得られないと考えられるので、少し詳しい説明を試みる。ここでは、単純な事例しか説明しないが、毎年開かれている MIT での STAMP ワークショップでは、ソフトウェア集約型の工学システムに加えて、工場や建設現場などでの作業安全、生産現場の効率向上、組織と人が絡んだ社会システムのコンプライアンス管理など、安全・品質・セキュリティ・生産性などが絡んだ多様な応用が議論されている点を付記しておく[Leveson2019]。

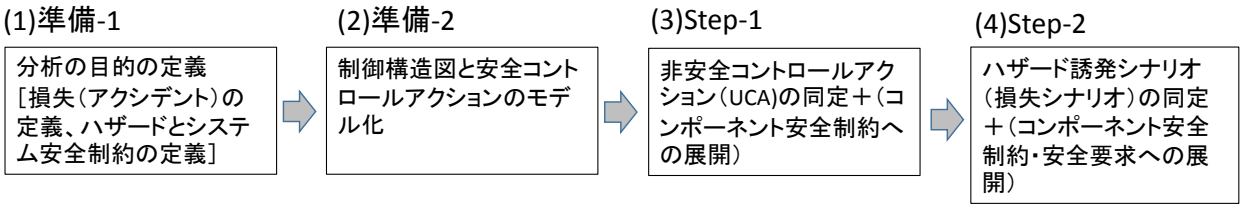


図 1.3-1　STPA の 4 段階の手順

1.3.1. 分析の目的の定義

図 1.3-1 の準備-1 の手順とその背景を説明する。

STPA の最初のステップは分析の目的として、下記の三つを定義する。

・損失（Loss）の識別（安全分野ではアクシデントと呼ぶ）

・システムレベルのハザードの識別

・システムレベルの安全制約の識別

ここで、損失は利害関係者にとって受け入れられない何らかの価値が喪失するという広い意味で定義され、損失するという事象をイベントの一種とする。人命の喪失や傷害、物的・経済的損害、環境の汚染、機密情報の喪失・漏洩などがこれにあたる。ハザードは、想定される環境条件のもとで、損失につながるようなシステムの状態である。システムレベルの安全制約は、ハザードを防ぐ（そして最終的に損失を防ぐ）ために満たすべきシステムの条件や動作のことである。

上記の中で特に難しいのはハザードの定義であるが、その際の注意点を以下にあげておく。

(1) 損失はイベント、ハザードは状態として定義する。他の規格でのハザードの定義である危険源と異なる定義である点に注意が必要である。

(2) システムの外部環境は制御不能なので、ハザードの対象外である。また、ハザードを予測する際には、外部環境は最悪の状態を想定して考える。

(3) ハザードは、その誘発要因（故障やエラーなど）と区別して扱うことが大事である。例えば、車を例にとると、ブレーキが間違って動作する状態というのは故障要因まで訴求しているので不適切である。意図しない加速や減速が起こる抽象的状態でハザードを定義することで、その後に続く分析ステップで、要因としてのブレーキ故障やアクセル故障、さらには、路面状態に起因する減速などまで発想を広げることが出来る。

(4) ハザードはシステム全体を見て定義し、その数は 10 以下が望ましい。これが多すぎるのは、システムの安全機能の理解の抽象度が足りない（具体的すぎる）ことを意味し、発想がシステム思考にならずに、具体的なモノ（要素）に囚われすぎた還元論的な発想になっていることを意味する。新しい製品の想定外のハザードをできるだけ少なくするためには、システム思考による全体を見た発想力が必要である。つまり、抽象度を高めてシステム全体の安全機能を理解した上で、ハザードを識別することが重要になる。

最後に、アクシデントとハザード、安全制約のいくつかの具体例を表 1.3-1 に示しておく。実際の分析にあたっての参考にされたい。

表 1.3-1　損失、ハザード、安全制約の例

システム	損失(アクシデント)	ハザード	安全制約
ACC(自動追従運転)	L1:2台の車の衝突	H1:前方ないし後方の車との不適切な車間距離	SC1:二つの車は最小の車間距離を越えてはならない
化学プラント	L1:有害物質による人命の損失または危害	H1:プラントからの有害物質の気中や地中への放出	SC1:有害物質はプラントから過失によって放出されてはならない
自動車のエアバッグ	A1:運転者の死傷	H1:衝突したのにエアバッグが開かない H2:通常走行時にエアバッグが開いてしまう H3:エアバックの異常な爆発（部品飛散）	SC1:衝突時にはエアバックが開く SC2:通常走行時にエアバッグは開かない SC3:エアバック開の際に部品を飛散させない

1.3.2. CS 図と安全コントロールアクションのモデル化

図 1.3-1 の準備-2 の手順とその背景を説明する。

図 1.3-2 に基本的な STAMP の CS 図（制御構造図）を示す。システム全体の安全制御構造を俯瞰的に理解し可視化するためには、この CS 図の各要素を、その機能に着目して抽象化・階層化してモデル化することが大事になる。その基本要素は、システムの目標（例えば、アクシデントの防止）を達成するためのコントローラーと、その指示によって動く被コントロールプロセスである。また、コントローラーには、制御対象（被コントロールプロセス）の挙動に対するプロセスモデルと、目標を達成するための動作指示（これをコントロールアクション（CA）と呼ぶ）を生成するためのコントロールアルゴリズムがある。さらに、コントローラーにはフィードバック（FB）と呼ばれる情報が入力され、CA の結果を確認したり、CA を生成するための情報として用いられる。もちろん、被コントロールプロセスからだけでなく、それ以外の外部からの入力情報もコントロールアルゴリズムの生成には使われる。

CS 図のプロセスモデル、ならびに、コントロールアルゴリズムは、コントローラーが人間の場合、それぞれ、メンタルモデルと操作手順書や意思決定ルールに相当すると考えるとわかり易い。

CS 図において、誰が誰をコントロールするかは、その権限とも重なっており重要である。その際の CA は、ゴールを達成するための目的を持った行動であり、責任を伴う役割がある。一方で、FB は特定の高レベルの目的を意識せずに提供する情報である。CA と FB の違いは自明ではないことがあるが、両者を取り違えても次のステップであるハザード誘発シナリオの結果に関しては重大な違いはない。なぜなら、ハザード誘発シナリオまで識別すれば、間違った CA ないし間違った FB として、同等のハザード誘発シナリオが導かれるためである。しかしながら、安全制御の要求仕様を考える際には違いが出てき得る。目的を持った CA と、高レベルの目的を意識しない FB では、その代替案を考える際に違いが出てくるためである。

CS 図は、システム全体の安全制御機能を理解するために用いられ、抽象化・階層化された機能に着目して、多様な利害関係者の中で共通の理解が得られるように作ることが大事である。このシステムの可視化による理解の共有は、ハザードの可能性をいろいろな視点で議論することで想定外のハザードを減らすために極めて重要なプロセスでもある。なお、この制御構造モデルはプログラムとして実行可能なものである必要はなく、その振る舞いの制約や要求仕様などが明確で

利害関係者の間で共有されていればよい。つまり、STPA は詳細な要求仕様や実行可能なモデルを作るために使われるべきであるので、実行可能で正確なモデルが柔軟な発想を妨げないようにしないといけない。

以下、STPA ハンドブック [1]の事例を少し変更して CS 図と CA のモデル化の手順を説明する。これは、航空機の離着陸時の自動ブレーキシステム（BSCU：Brake System Control Unit）の CS 図（図 1.3-3）である。アクシデントは衝突事故による人命の損傷であるが、その際のハザードは、「航空機が地上走行時に他の物体に近づき過ぎること」と定義される。このモデル化にあたっては、下記に示すように、関係する各コンポーネントの安全責任を明確化しておくことが大事になる。

(1) 車輪ブレーキ本体

R1：BSCU またはフライトクルーからの指示により車輪回転速度を下げる

(2) BSCU

R2：フライトクルーからの要求によりブレーキを作動させる

R3：車輪スリップの際に断続（パルス）ブレーキを作動させる

R4：着地ないし離陸中止時に自動ブレーキを作動させる

(3) フライトクルー

R5：いつブレーキを作動させるかの判断

R6：ブレーキ動作モードの設定（自動、手動）

R7：ブレーキの監視と BSCU 解除、故障時の手動ブレーキ操作

これらの責任を遂行するため、まず、BSCU 設定・解除・ブレーキ指示というフライトクルーから BSCU への CA が必要になる（R5、R6、R7）。BSCU から車輪ブレーキ本体への CA はブレーキをかける制御信号そのものである（R2、R3、R4）。ただし、この段階では、パルスブレーキや自動ブレーキは具体化されていない。最後に、車輪ブレーキ本体の責任として、フライトクルーないし BSCU からの指示に基づいて動作するという CA（手動ないし自動ブレーキ）が記載されている（R1）。

次の詳細化として、これらの CA を遂行するために必要な FB 情報が詳細化される(図 1.3-4)。フライトクルーが BSCU 設定・解除、ブレーキ指示を行うために必要な情報として、自動ブレーキモードと設定減速比があげられる。設定減速比は急停止を避けた自動ブレーキ操作に必要なものと考えられるが、フライトクルーの所掌責任（R6）としては明示化されていないので、本来は詳細化の時点でこの所掌責任も詳細化されるべきものである。また、BSCU からフライトクルーへの FB として、BSCU モードと BSCU 故障があげられているが、これも R5～R7 の遂行のために必要な情報となる。これに伴って、BSCU 電源 On/Off という CA が追加されている。抽象化すると BSCU 設定・解除という CA と同一視もできるが、BSCU 故障時の振る舞いも考えて別出しにしている。外部から BSCU への情報として、着地、離陸中止、慣性参照速度、車輪からの FB として車輪速度が追加されているが、これは、BSCU の責任 R3、R4 遂行のために必要になる。また、手動ブレーキの情報も BSCU の入力として追加されているが、これも、パルスブレーキ動作を手動ブレーキと並行して作動させることを想定すると必要な情報となる。

また、さらなる詳細化で明確化が必要なのはフライトクルーから BSCU へのブレーキ指示であ

る。ここでは、手動ブレーキを CA として定義したため、フライトクルーから BSCU を介したブレーキ指示（手動相当）を削除している。BSCU を介した手動ブレーキのみの設計としてしまうと、BSCU 故障によりブレーキ手段がなくなってしまうことが容易に想像できる。一方で、BSCU が自動モードの場合、手動ブレーキ情報とスリップ状況によりパルスブレーキ（人間の応答時間よりも早い短周期でのブレーキ指示と解除の繰り返し）を作動させるが、他に BSCU が手動ブレーキを強制解除する必要がある場合、これを機械優先の安全責任（安全制約）として明示化する必要がある。例えば、離陸決定速度（V1 速度）に達した後の手動ブレーキを BSCU が解除できるかどうかは、今回の安全責任の定義の中では明示化されていない。人・ソフトウェアが協調動作する今後のシステムでは、それぞれの責任分担を明示化して想定外のハザードを防ぐことがより重要になる。

以上のように、CS 図と各コンポーネントの所掌責任、目的達成のための CA、ならびに、それに必要な FB 情報をトップダウンで分析し可視化することで、システム全体の安全制御機構を把握することができ、コンポーネント間の優先度を含めた相互作用も明確にできる。これらは、後段のハザード誘発要因の分析の基本となるモデルとして重要になる。

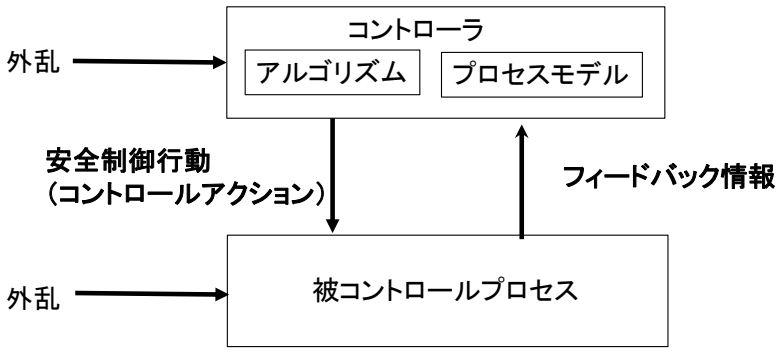


図 1.3-2　システム理論に基づく事故モデル STAMP

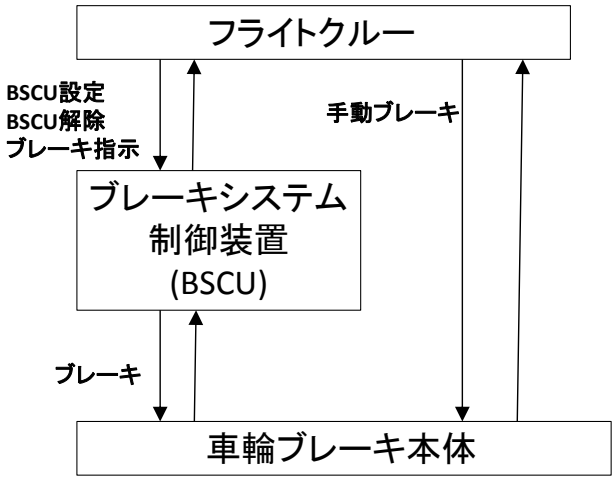


図 1.3-3　航空機の車輪自動ブレーキシステムの CS 図

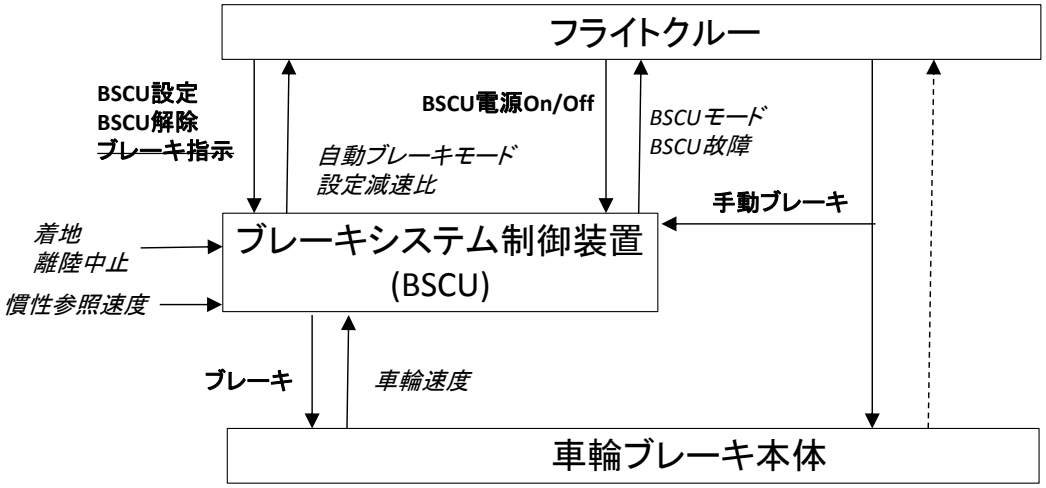


図 1.3-4　航空機の車輪自動ブレーキシステムの CS 図（フィードバック情報の具体化）

1.3.3. 非安全コントロールアクション（UCA）の識別

図 1.3-1 の Step-1 の手順とその背景を説明する。

UCA は、ある特定のコンテキストと最悪の環境下で、ハザードにつながる CA のことで、次の四つに分類される。

(1) CA を与えないことがハザードにつながる（N:Not Providing）

(2) CA を与えることがハザードにつながる（P:Providing causes hazards）

(3) 潜在的には安全な CA を与えるが、早過ぎる、遅過ぎる、または間違った順序である（T:Incorrect Timing/order causes hazards）

(4) CA があまりにも長く続いている、あるいは、あまりにも早く止まる(D:Incorrect Duration causes hazards)

この四つの UCA を分類したのが図 1.3-5 である。安全論証では、完全性（排他性や網羅性）が大事であるが、この UCA は分類方法としては互いに排他的で網羅性があることがわかる。ただし、この分類は産業分野によっては当てはまらないことがあるかもしれない。それぞれの分野でカスタマイズすることは可能だが、分類の排他性や網羅性に注意してカスタマイズすることが大事である。産業分野ごとの過去のトラブル経験による不安全行動などは、後段のハザード誘発シナリオで考え、この UCA はできるだけ一般的な分類とすることで、ハザード誘発要因の発想を妨げることがないようにすべきである。

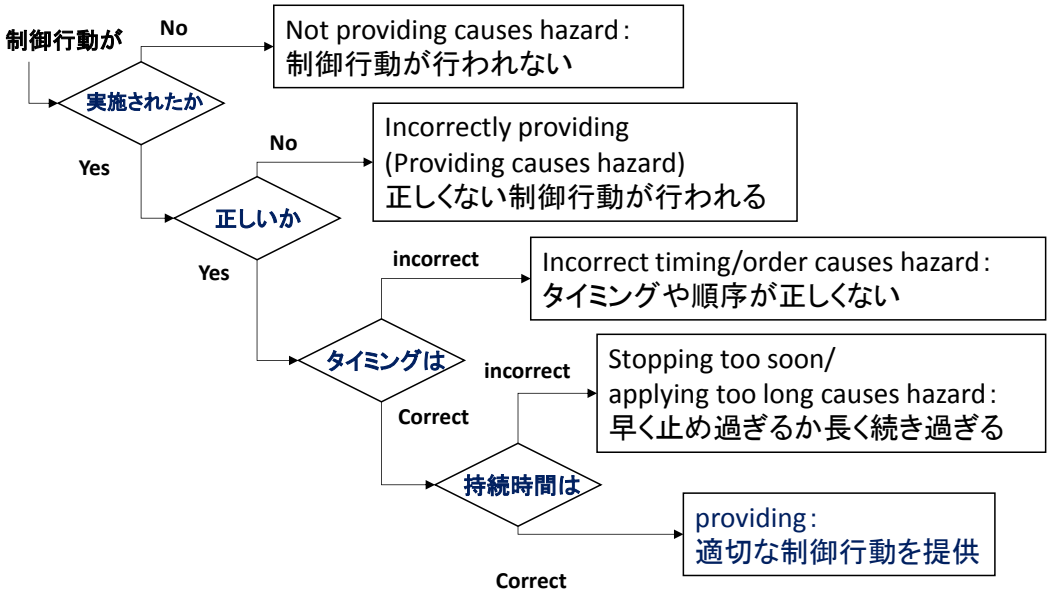


図 1.3-5　非安全コントロールアクション（UCA）の分類

1.3.4. ハザード誘発シナリオの識別

図 1.3-1 の Step-2 の手順とその背景を説明する。

STPA の最後のステップは、ハザード要因が UCA とハザードを通して損失につながるシナリオをハザード誘発シナリオとして記載することである。この損失シナリオは下記の視点、即ち、

- なぜ UCA が起こるのか
- なぜ CA が不適切に実行されたり、されなかったりしてハザードに至るのか

という二つの視点から考えることが大事である。図 1.3-6 にこのシナリオを考える際の大分類を示しておく。当初の N. G. Leveson の教科書[Leveson2012]では、ハザード誘発要因としてのキーワードがより細かく載っていたが、このキーワードが逆にチェックリストのように使われて柔軟な発想を絞ってしまうという欠点に気づき、その後は、シナリオという形で柔軟な発想をするように推奨されている[Leveson2018]。複数の要因が同時並行的に絡み合ってハザードに至るというシステミックな発想が大事であり、チェックリストのように要素分解してしまうと、従来の故障分析法である還元論的な発想に陥ってしまうという危険性を指摘している。

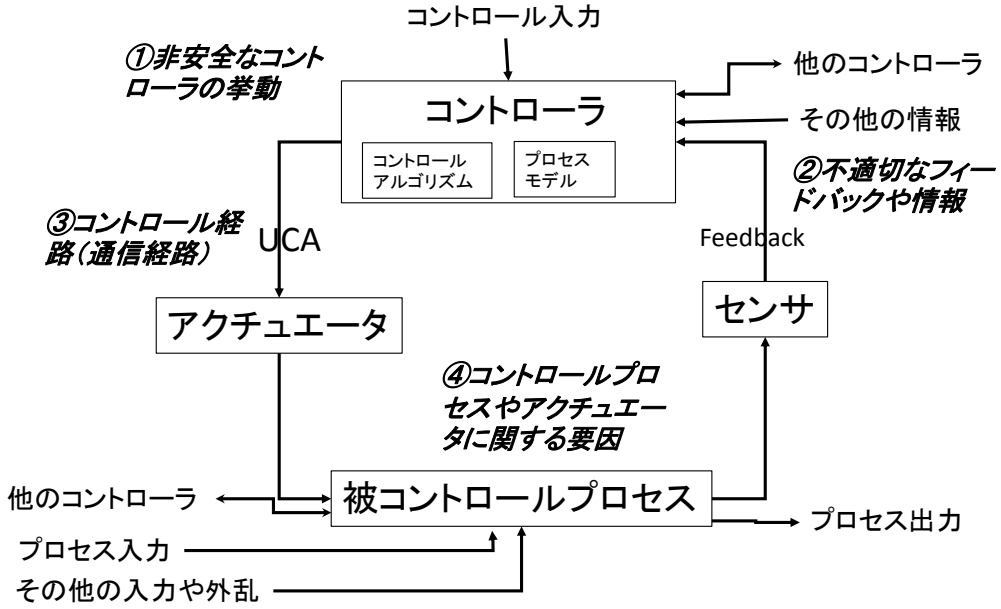


図 1.3-6　ハザード誘発シナリオの分類

1.4. まとめ

STAMP/STPA の実施手順とその背景にある考え方を述べた。より詳しい適用事例については過去の WG 報告を参考にされたい。

はじめての STAMP/STPA[IPA2016]

はじめての STAMP/STPA（実践編）[IPA2017]

はじめての STAMP/STPA（活用編）[IPA2018]

STAMP/STPA はシステム思考によるハザード分析の考え方であるが、その背景には、最近の大きな事故事例の分析から反省される三つの問題点、想定外の事故、後知恵による解釈、そして確率論評価による認知バイアスがある。最初の 2 項目については、本章で述べたように、STPA による安全制御構造の可視化とピアレビューによるプロアクティブな安全設計を通してハザードを大幅に低減、緩和できることが期待できる。「想定外を想定して安全設計をしなさい」という要求仕様は一見矛盾するようであるが、それぞれの専門知識や過去の失敗経験によって想定できる範囲は異なるので、複数の利害関係者によるピアレビューと安全論証は、新しい製品ほど重要になる。事故が起こった後に初めて判明した知見を、事前にわかっていたように用いて責任追及をする、いわゆる後知恵（hindsight）による議論も事故原因究明でしばしば見られる。しかし、これも誰かに責任を押し付けることで、却って、将来の危険性を除去する検討が不十分になる。安全制御構造の欠陥として客観的な形で事故原因を可視化することで初めて将来の危険性を除去できると考えられる。

確率論による分析は、今の大規模システムでは広く用いられており、それが重要であるのは当然であるが、N. G. Leveson はその危険性も指摘している。ソフトウェアは確率論にのらない、未経験の製品は使用実績による故障確率に頼れない、人の行動は環境や仲間の行動にも大きく影響され確率論では推測できない、長期の使用の間に環境が変わったり交換部品の故障確率が変わっ

たりする、といった問題点を指摘している。STAMP/STPA では、最悪の環境下でのハザードシナリオを定性的に考えて、それを緩和する手段を考えるが、これは確率論ではできないことである。

システム理論に基づいて安全性を論証する STAMP/STPA という方法論を、N. G. Leveson の著作に沿って説明してきた。この中で述べてきた中で、もう一点強調すべきことは、安全責任の存在論である。IoT 時代の中で分散協調型のシステム開発が今後増えてくると考えられるが、その中で、どのコンポーネントがどこまでの安全責任を負うべきかがあいまいなままシステムが提供される可能性も増えてくるであろう。そのようなとき、本章で紹介した安全責任と、それを達成するためのコントロールアクションの関係を明示化する CS 図は、システムに関係するステークホルダー間での安全責任の共有化にも役立つといえる。IoT 時代に特に必要とされる方法論でもある。

今後の新しい製品開発に必須となるであろうプロアクティブな安全設計での利用を期待したい。

2. STAMP の効果的な活用事例と解説

本章では、STAMP の具体的な活用事例を紹介する。いずれも STAMP の実践的な応用例であるとともに、システムを俯瞰的に見て、構成要素の相互作用を把握した上で間接的な因果関係も含めて事故要因を考えるというシステム思考がうまく活かされた結果を示している。

最初の「列車警報システム」は、線路の保守作業を行う作業員に列車の接近を知らせるシステムを対象とした分析例である。鉄道会社で実際に使われているシステムをベースとして、GPS や携帯電話回線を利用した新しい形態のシステムを模索する際に STAMP による安全分析を適用した。自前の設備ではない GPS や通信システム等を連携させて利用するシステムであり、その構成要素の信頼性等に不確実性が伴うことが特徴となる。本例では、携帯端末等の信頼性を高めることは困難なため、システム全体を俯瞰して考えるシステム思考的なアプローチを用いた結果、人と機械の安全に関する役割分担を見直す発想が可能となり、システムの構造によって安全を高めるアイデアを得ることができた。STAMP/STPA は、一般的にはハザードシナリオを見つけるのに有効な手法と認識されているが、本例は、システムの構造に関する新しいアイデアを広げるためにも STAMP/STPA が利用できることを示す例といえる。

2 番目の「高齢者見守りサービス」は、ディジタル化された生活情報をネット経由で通信し、高齢者の見守りサービスを行う IoT 活用システムの分析例である。IoT システムは、従来実現困難だった新しい価値を生み出せる半面、ネットでつながるコンポーネントの思いもよらない相互作用が意図に反する結果を生じる可能性があり、STAMP による安全分析が適する領域の一つと考えられる。本例では、鍵の開閉センサーにより住人が意識せずとも在室/不在が警備会社に通知されるという機能に対して、想定していなかった相互作用が発現する可能性に着目してハザードの可能性を洗い出した。システム全体を俯瞰することにより、機能を個別に見て考えている限りは気づきにくい問題点を発見できたことは、システム思考の特徴が活かされた点といえる。

3 番目の「自動車製品ライフサイクル」は、自動運転機能を持つ自動車に対して、製造、運搬、販売、使用、保守、廃棄というライフサイクルを意識して STAMP/STPA 分析を適用した例である。JASPAR において 6 つの分析チームが、3 つの異なるライフサイクルステージのユースケースを対象として分析を行った試みについて紹介している。本例は、ライフサイクルという時間軸を意識的に俯瞰して考えた点が特徴的である。「自動車」というシステムに対して、時間的な変化を俯瞰的に考えた点は、システム思考が実践された例ということができる。また、STAMP/STPA の分析支援ツールである「STAMP Workbench」を用いるチームと用いないチームに分けて分析を行っており、ツール使用の有無で分析作業の効率に違いが出ることの確認もされている。

最後の「IT システム運用」は、人間系に対して、事故が起きた後の原因分析に STAMP/CAST の手法を適用した例である。人や組織間の指示系統をコンポーネント間の制御関係と見なして STAMP/CAST 分析を行っている。一見すると個人のミスが原因で起きたと考えられる事故が、組織全体をシステムと見て分析を行うと、組織間の構造的な問題との因果関係が見えてくることを示している。「コンポーネントの故障がなくても起き得る事故の要因を考える」という STAMP の思想が人間系のシステムにも適用できることを示した例である。

システム思考に基づく STAMP の特徴の理解を深める一助として、これら 4 事例について以下に詳しく紹介する。

2.1. 列車警報システム

2.1.1. はじめに

鉄道会社では、安全・安定輸送の確保はもちろんのこと、線路際で保守作業や工事を行う作業員の安全確保にも取り組んでいる。その一例として、JR 東日本で導入が進められている GPS を用いた列車警報システム（以降、「GPS 列警」と略記）がある。

本節では、STAMP/STPA 手法の適用事例として、GPS 列警の安全性分析を行うとともに、新しい作業安全の考え方である Safety x.0 との関係性についても述べ、将来の安全哲学でもある Safety 2.0 を見据えた GPS 列警の改善案の検討を行うこととする。

2.1.2. 列車警報システムの概要

鉄道において作業員の安全を確保する仕組みは、技術の進展とともにいくつかの形態を経て進化している。ここでは、JR 東日本における列車警報システム（以降、「列警システム」と略記）の変遷から、作業員安全確保の歩みを概観する。

① 初期のシステム：機械的なバックアップがなく、人間系に依存するもの（図 2.1-1）

列車見張員と作業員の体制により安全を確保する方法である。

列車見張員は指令に列車運行状況を確認するとともに、列車ダイヤ図を所持し、作業箇所への列車接近時刻を予測する。列車が接近した際は、列車見張員が作業員に待避を指示する。すなわち、列車見張員の注意力によって安全を確保する方法である。現在では、地方線区など、列車位置を機械的に把握できない線区で主に用いられている。

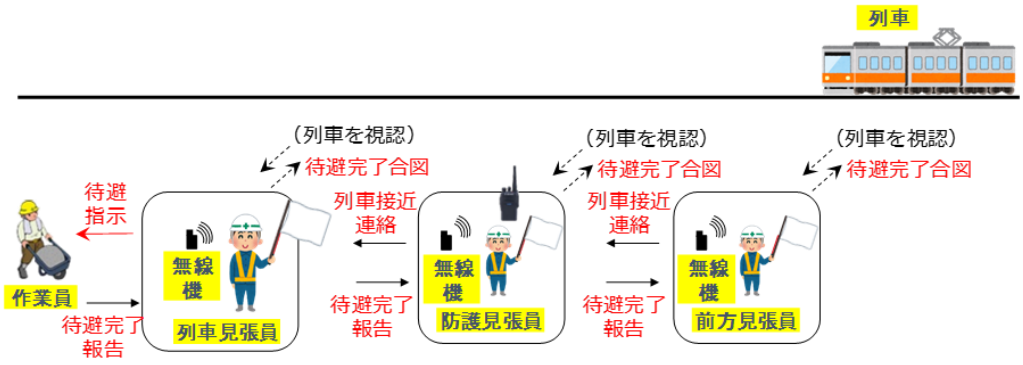


図 2.1-1　列車見張員による作業員安全確保

② 現行のシステム：列車位置検知に機械的なバックアップを用いるもの（図 2.1-2）

①の方式では、作業員の安全を人間系に依存している点が大きな課題である。

この課題を解決するために、列車位置検知を機械的に行う仕組みが導入されている。列車見張員と作業員は TC 列警（TC 型無線式列車接近警報装置）と呼ばれる受信機を所持する。走行中の列車の位置を、軌道回路と呼ばれる列車検知装置（おおよそ数百 m～km 単位の区間）で検知し、その位置情報を、現場に設置された無線基地局から TC 列警に電波で送信し、その情報を認めた列車見張員と作業員は線路外に待避する。

現在、首都圏を中心にした主要線区に用いられている。

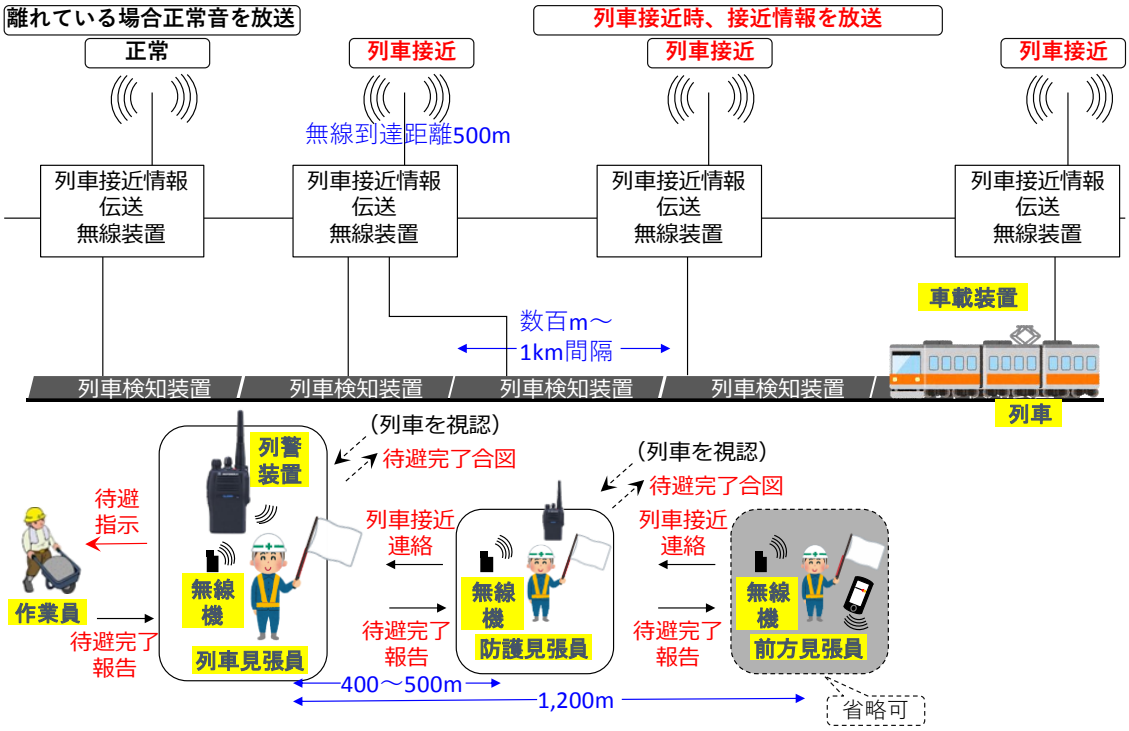


図 2.1-2　TC 列警による作業員安全確保

③ 今後のシステム：列車の位置把握と作業員の携帯端末に GPS を用いるもの（GPS 列警）

②の方式は、作業員の安全性向上に寄与しているが、ケーブルや無線などの設備が重厚であることや、列車検知（軌道回路）が設置されていない地方ローカル線などの区間は適用できない、列車が作業区間を抜けても軌道回路を抜けるまでは作業再開できず、場所によっては作業性が悪いといった課題がある。

この課題を解消するために、列車位置検知と作業員の端末に GPS を用いた「GPS 列警システム」を順次導入している。GPS 列警システムの構成を図 2.1-3 に示す。

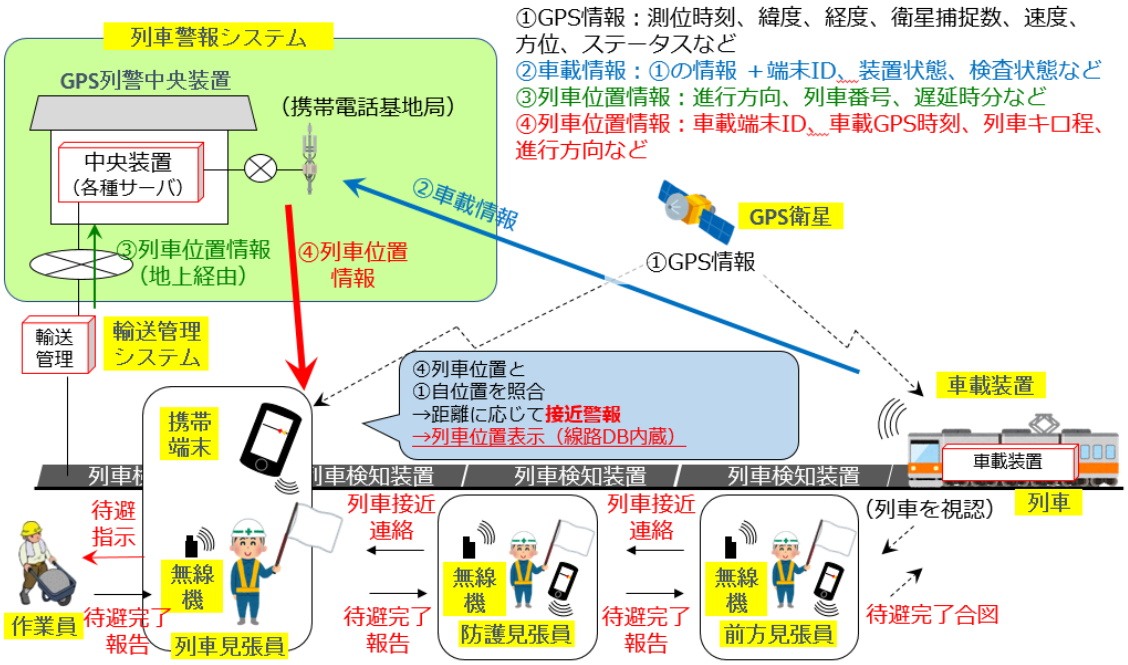


図 2.1-3　GPS 列警による作業員安全確保

GPS 列警の動作概要は以下のとおりである。

(1) 列車は、GPS により算出した位置情報を持っており、その位置情報は常時「列車警報システム」に伝送されている。

(2) 一方、列車見張員（前方見張員、防護見張員含む。以下同様）は「携帯端末」を所持しており、携帯端末も GPS により算出した自位置情報を得ている。

(3) 列車警報システムは、列車が在線中の線区にある携帯端末に対して、列車位置情報を送る。

(4) 携帯端末は、列車警報システムから受信した列車位置情報と、(2) で算出した自位置を比較し、一定距離内に接近したと判断した場合は「接近警報」を鳴らす。

(5) 列車見張員は、携帯端末の接近警報を認めたら、作業員に対して「待避指示」を出す。（※実際には列車見張員のほか、作業員も携帯端末を所持している。）

(6) 作業員は、列車見張員の指示を受けて線路外に待避する。

2.1.3. GPS 列警の安全性分析

この節では、GPS 列警システムの安全性分析を、従来から用いられている FTA 手法と STAMP/STPA を組み合わせて行い、システム改善案を検討する。

まず、GPS 列警を構成する各コンポーネントに対して FTA を用いて安全性分析を行った。その結果から、携帯端末、列車警報システム（中央装置）、および相互の伝送部分が GPS 列警のアキレス腱ともいえる部分であり、改善すべき課題があることが分かる。

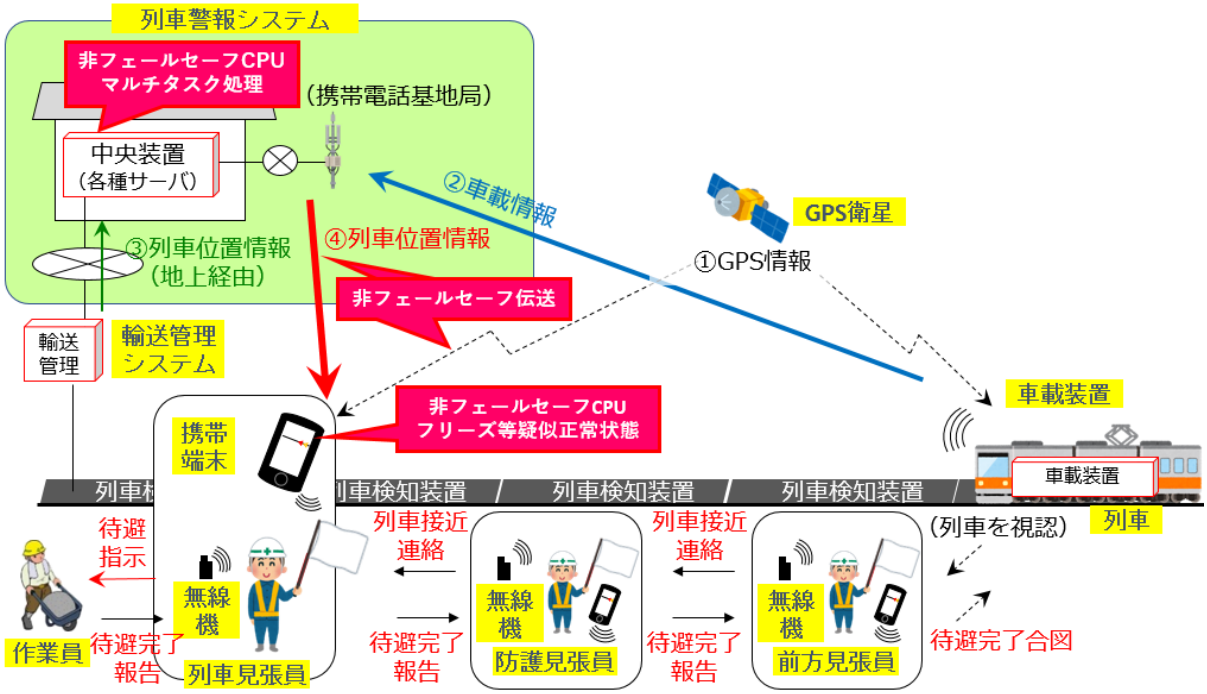


図 2.1-4　FTA 分析でわかった GPS 列警の課題

一方、GPS 列警のシステム全体を俯瞰的に分析し、制御構造上の課題を見出すために、STAMP/STPA を用いた分析を行った。

まず、GPS 列警におけるアクシデントは、以下の 2 つと定義した。

- 作業員や列車見張員が列車と接触し、負傷する。
- 必要な保守作業時間を確保できない。

次のステップとして、CS 図を以下のとおり構成した。CA は携帯端末から列車見張員への接近情報（接近警報・接近注意報）および故障情報と、列車見張員から作業員への待避指示、列車見張員から列車への停止／進行継続指示があり、システム内の情報の伝送は CA を作成するために必要な情報交換という位置づけである。

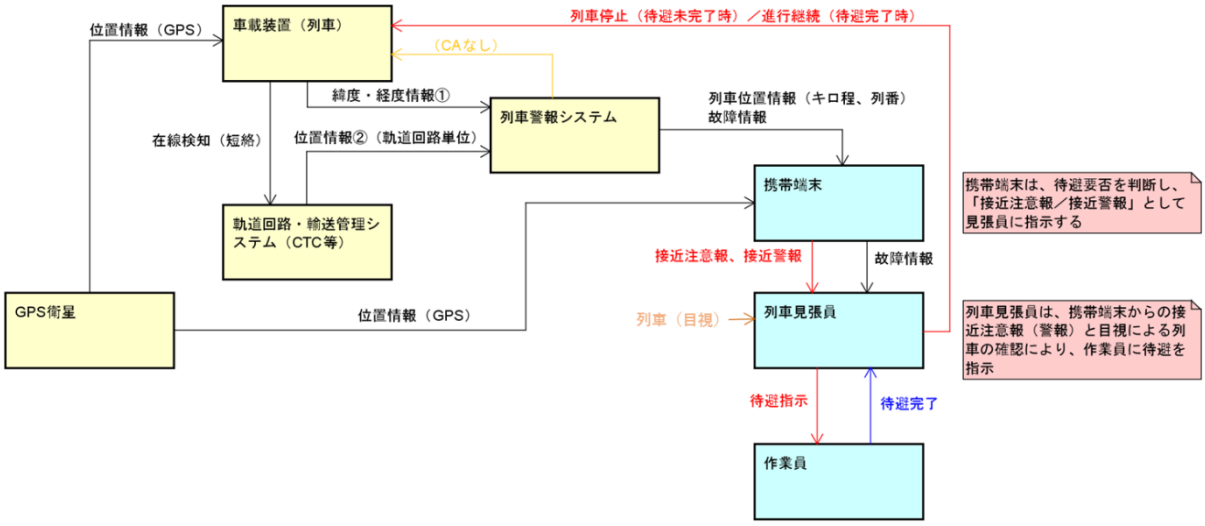


図 2.1-5　GPS 列警の CS 図

これらを踏まえ、UCA（Unsafe Control Action）の分析を行った。その結果を表 2.1-1 に示す。

表 2.1-1　GPS 列警の UCA 表

	CA	From	To	CA 提供条件	Not Providing	Providing causes hazard	Too early / Too late	Stop too soon/Applying too long
1	接近注意報、接近警報	携帯端末	列車見張員	列車が接近警報距離(1500m)以内に接近すると警報	(UCA1-N-1) 列車接近にも関わらず列車見張員が待避指示を出さない [SC1]	(UCA1-P-1) 列車接近していないにも関わらず、列車見張員が待避指示を出してしまう [SC4]	(UCA1-T-1) 列車接近にも関わらず列車見張員が待避指示を出さない [SC1]	(UCA1-D-1) 携帯端末の鳴動停止により、列車接近にも関わらず列車見張員が待避指示を出さない [SC1] (UCAa-D-2) 携帯端末が鳴りやまず、列車通過後も列車見張員が待避指示を継続する [SC-4]
2	待避指示	列車見張員	作業員	携帯端末の鳴動を認めたら待避指示を出す	(UCA2-N-1) 列車接近にも関わらず作業員が待避しない [SC2]	(UCA2-P-1) 列車接近していないにも関わらず、作業員が作業開始できない [SC4]	(UCA2-T-1) 列車接近していないにも関わらず、作業員が作業開始できない [SC4] (UCA2-T-2) 列車接近にもかかわらず作業員が待避しない [SC2][SC3]	(UCA2-D-1) 作業員が列車見張員の待避指示に気付かず、待避しない [SC2][SC3] (UCA2-D-2) 列車通過したにもかかわらず、作業員が作業開始できない [SC4]
3	故障情報	携帯端末	列車見張員	列車位置情報の不整合を検知したら故障情報を発する	(UCA3-N-1) 位置情報が不正確にもかかわらず、正確であると誤認して、列車の接近に気づかない [SC1][SC3]	(UCA3-P-1) 位置情報が正常にも関わらず、故障と誤認して、作業員が作業開始できない [SC4]	(UCA3-T-1) (Too early)正常にもかかわらず、故障と認識して、作業員が作業開始できない [SC4]	(UCA3-D-1) (Stop too soon) 位置情報が不正確にもかかわらず、正確であると誤認して、列車の接近に気づかない [SC1][SC3]

							(UCA3-T-2) (Too late)位置情報が不正確にもかかわらず、正確であると誤認して、列車の接近に気づかない [SC1][SC3]	(UCA3-D-2) (Apply too long) 位置情報が正確にもかかわらず、故障と誤認して、作業員が作業開始できない [SC4]
4	列車停止 ／進行継続	列車 見張員	列車	列車見張員が作業員の待避を確認したら進行継続指示、待避未確認ならば列車停止指示を出す	(UCA4-N-1) 待避未完了で列車接近中にもかかわらず列車停止指示を出さない [SC3] 待避完了にもかかわらず進行継続指示を出さず、列車を止めてしまう	(UCA4-P-1) 待避未完了にもかかわらず、待避完了と誤認して列車に進行継続指示を出す [SC3] 待避完了にもかかわらず、待避未完了と誤認して列車停止指示を出す	(UCA4-T-1) 待避未完了の状態で、列車停止指示を出すタイミングが遅れる [SC3] 待避完了にもかかわらず、進行継続指示を出すタイミングが遅れ、列車を止めてしまう	(UCA4-D-1) 列車が接近する前に一度出した停止指示を止めてしまい、列車が止まらない [SC3] 列車が接近する前に一度出した進行継続指示を止めてしまい、列車を止めてしまう

No.2 および No.4 は、今回の FTA 分析では出てこない項目であった。何故なら、今回の FTA ではシステム（機器）の正常動作（端末が正常に鳴動すること）に主眼を置いているためである。

これらの分析結果を STAMP の CS 図にマッピングすると、図 2.1-6 の通りとなる。赤色の項目が FTA、オレンジ色の項目が STAMP/STPA で抽出されたリスク要因である。

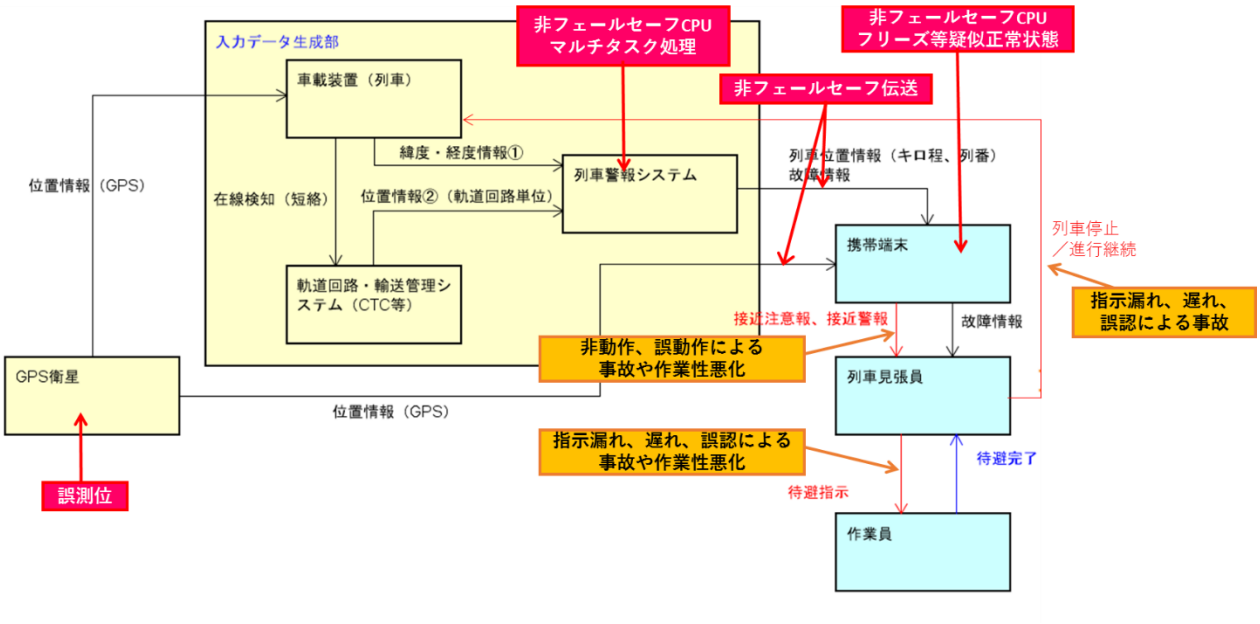


図 2.1-6　GPS 列警の FTA、STAMP/STPA 分析結果をマッピングした CS 図

GPS 列警の制御構造上の課題を見える化すると、安全責任を担う携帯端末に低信頼性要因が集中していることが見えてくる。

このように、FTA と STAMP/STPA を組合せて用いた結果、GPS 列警システムの構造上の課題が見えてきた。この活用例からわかるとおり、STAMP/STPA は、FTA など従来の安全性分析手法にとって代わるものではなく、それぞれの手法を相互補完するものであることに着目したい。

特に、人も絡んだ複数システムが複雑に絡み合う場合の相互作用については、STAMP/STPA の得意分野であり、スタティックでハードウェア構成に立脚した深い分析を行う FTA に対して、ダ

イナミックで自由度の高いハザードシナリオを生成できる STAMP/STPA の強みを生かした活用が有効と考えられる。

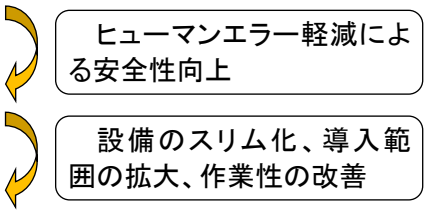
従来の列警システムでは、システムを構成するほぼ全てのコンポーネントが鉄道会社の自営設備であり、その重厚な設備によって高い安全性を確保することができた。一方、GPS 列警は位置認識のセンサーに GPS を用い、通信経路に通信キャリアが提供する公衆回線を用い、携帯端末に市販の端末を用いるといった典型的な IoT システムである。IoT システムでは従来と同じ考え方での安全性確保は難しい場合がある。例えば、鉄道会社が自前で強固な GPS サービスを構築するなど、技術的・コスト的に想定外である。そのため、システムを全体俯瞰して安全維持の仕組みを実用的なレベルで調整するアプローチが必須であり、かつ、有益でもあると考える。

2.1.4. Safety 2.0 を意識した GPS 列警の改善の試み

2.1.2 に示した作業員の安全確保の手段の「進化」は、近年産業安全の指標として提唱されている Safety x.0[中村 2017]と対応付けて考えると、表 2.1-2 のようになると考えられる。

表 2.1-2　列警システムと Safety x.0 の対応

作業員の安全確保	対応する Safety x.0
①　車見張員	Safety 0.0 : 列車位置はダイヤ使用。人間の注意力に依存。
②TC 列警システム	Safety 1.0 : 列車の位置把握を機械化（但し、精度は数百 m オーダー）
③GPS 列警システム	Safety 1.5 : GPS・通信回線により、列車と作業員の位置を高精度に把握
④協調安全型 GPS 列警システム（仮称）	Safety 2.0 : 列車と作業員が相互に情報交換を行い、協調して安全を確保



GPS 列警は、IoT 技術を活用して作業員や列車の位置情報を高精度化し、それを携帯電話回線で伝送する次世代の保安システムであるが、現在の仕様では協調安全の機能は有していないため、コンピューターを利用した安全制御である「Safety 1.5」に相当すると考えられる。ここでは、人と機械の協調によって安全を確保する「Safety 2.0」を目指した、より強固で柔軟な制御構造を持つ GPS 列警システムの改善仕様案について、CS 図をベースに検討してみたい。

Safety 2.0 は協調安全と呼ばれており、システムを構成するコンポーネントが相互に情報を交換することで、より安全でかつ高度な制御を可能とする安全レベルである。

GPS 列警の CS 図を見ると、列車警報システムから携帯端末、携帯端末から列車見張員・作業員の CA が一方通行で、FB を持っていないことに気付くであろう。このような FB を持たない「非クローズドループ」の構成は、列車見張員や作業員が待避完了したことがヒューマン依存の仕組みになっているため、ここにメスを入れることとする。

また、本システムでは作業員（携帯端末）の位置情報を GPS によって詳細に把握できるので、

これを使ってより安全性を向上させることを考える。

CS 図をベースに Safety 2.0 を意識したうえで改善案を検討し、以下の 2 形態を考えた。

①列車見張員からの待避完了情報により、列警システムが列車停止の要否を判断する方式

列車見張員は、作業員（と自分自身）の待避を確認したら、携帯端末の「待避完了」ボタンを押して、携帯端末に待避完了情報を与える。（列車見張員→携帯端末への FB）

その待避完了情報を列車警報システムに伝送し（携帯端末→列車警報システムへの FB）、もし待避完了情報があれば列車（車載装置）に進行継続指示を、なければ列車停止指示を出す。（列車警報システム→列車への CA）

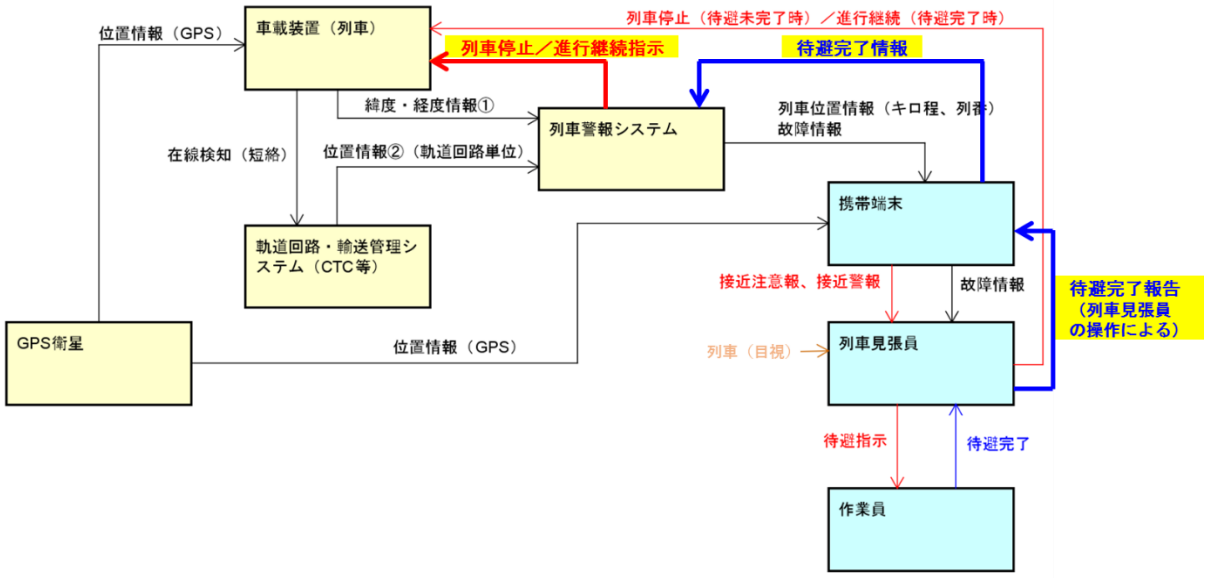


図 2.1-7　CS 図ベースで検討した GPS 列警の改善案①

この方式では列車警報システムから列車へ停止指示の CA が発生するため、安全性が向上する。しかし、待避完了情報を送る際に列車見張員の携帯端末操作を必要とするため、操作漏れや誤操作のリスクが存在する。そこで、本方式をさらに改良した次の方式を考案した。

②携帯端末からの位置情報により、列警システムが列車停止の要否を判断する方式

列車警報システムは、携帯端末に自身の位置（≒それを所持している作業員および列車見張員の位置）を把握させ、それが線路外であれば待避完了情報を列車警報システムに送らせる（携帯端末→列車警報システムへの FB）。なお、現地での待避完了状況の把握は GPS の他に画像を用いる方法など、いくつかの方法が考えられる。

以降は①と同様。

※②は、列車見張員のアクションがない分、①より高度になるが、以下の点に留意する必要がある。

- 携帯端末は作業員全員が持っている前提となる。
- 列車見張員からの指示ではなく個々の作業員が持っている携帯端末の位置情報がベー

スとなる。

- 線路内か、線路外かを識別するための精密な位置情報を列車警報システムが把握する仕組みが必要。

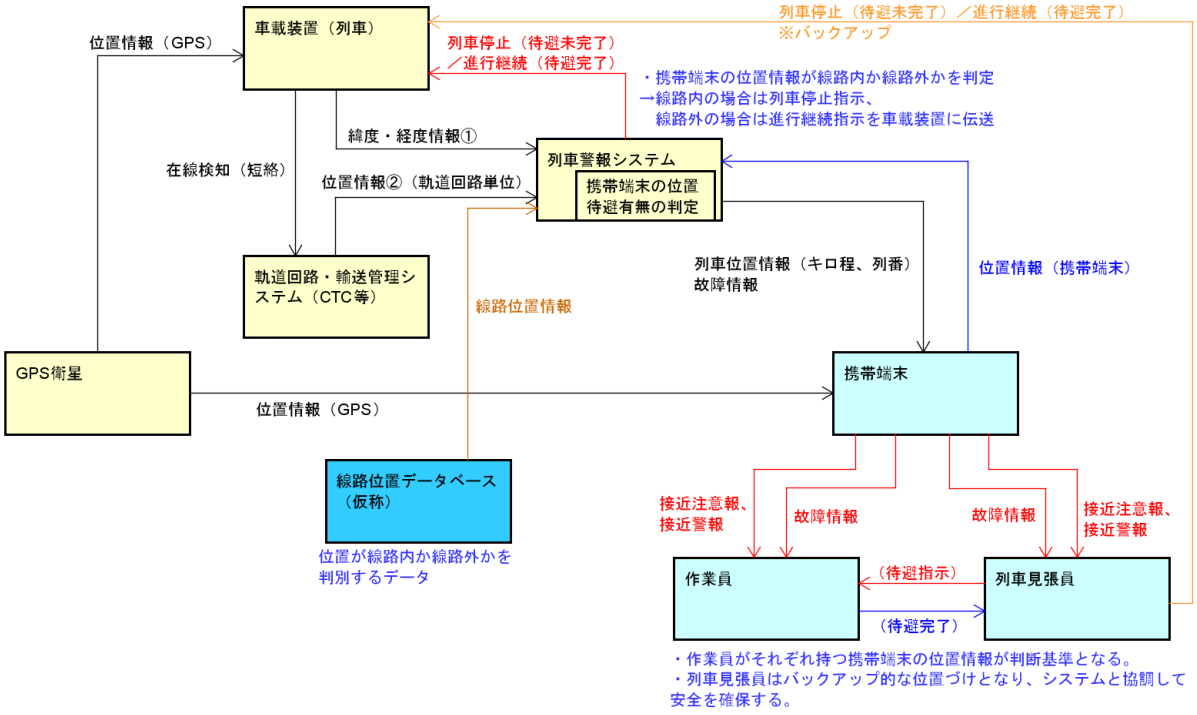


図 2.1-8　CS 図ベースで検討した GPS 列警の改善案②

これらを実現した新しい GPS 列警システムのイメージを図 2.1-9 図に示す。

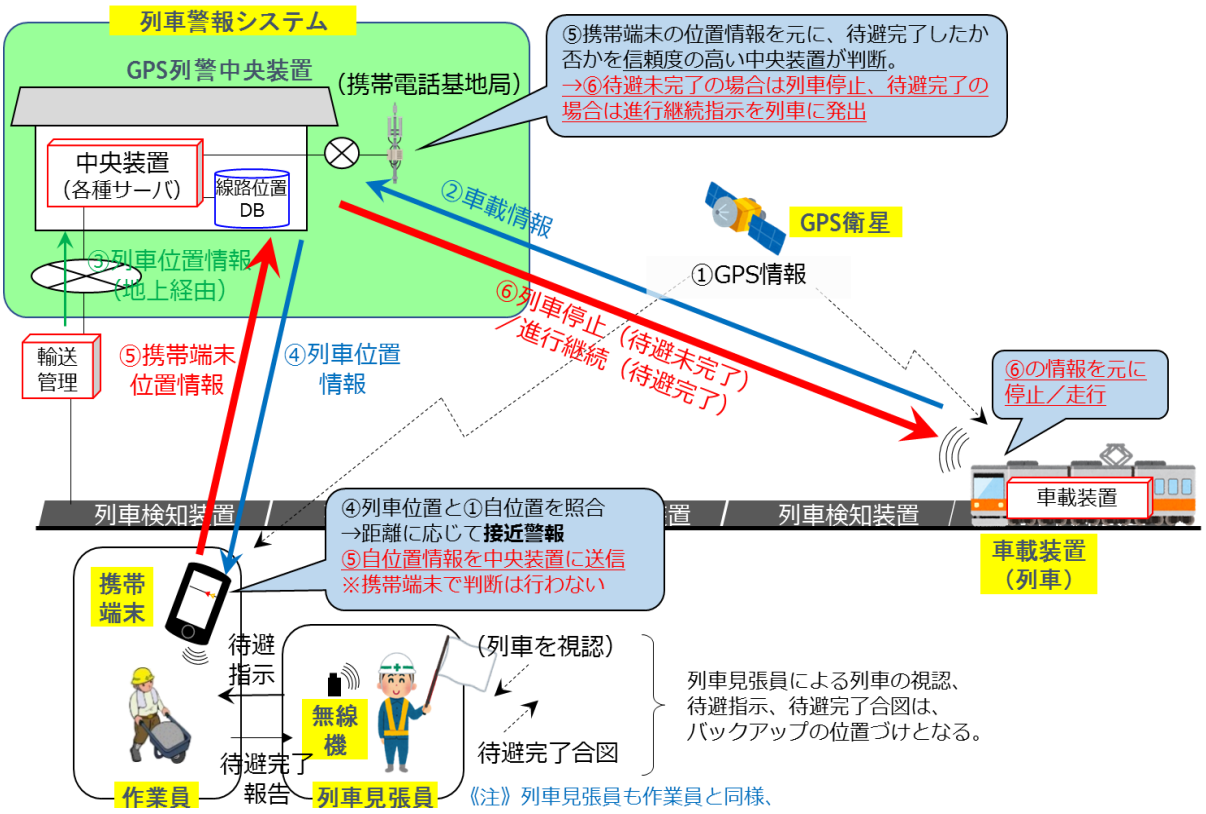


図 2.1-9　協調安全型 GPS 列警システムの例

これまで述べてきた GPS 列車警報システムの改善案の検討を通じて、以下の点について考察を深めることもできた。

■Safety 1.5→2.0 に伴うコンポーネントの責任所在の変化

CS 図をベースに本改善案を検討していく中で、システムを構成する各コンポーネントの安全担保の責務に変化が生じていることがわかった。

改善前の列車警報システムにおいては、CS 図（図 2.1-6）から分かるように信頼性に課題のある携帯端末が主に安全担保の責任を負っている（携帯端末が待避要否を判断して列車見張員に待避指示を出している）、と考えられる。改善案では、携帯端末は主要な安全責任を負わず、信頼性の高い列車警報システムが中心となって責任を担い、列車見張員と協調安全を図っている。

また、このような「責任分担の変化」を念頭に置くと、信頼性強化可能な列車警報システム（サーバー）が安全責任を担い、単なるデータ通信・表示装置の携帯端末は多重化（複数キャリアの端末を所持するなど）することで低信頼性要因を排除できることもわかる。

■CS 図ベースで検討することの有効性

本改善案を検討するにあたり、これまでの列警システムの概念を抜けない発想（特に、システムに詳しい開発者）では、TC 列警の発想から「音を鳴らす」手法に執着してしまい、本来の目的が「列車との衝突を防ぐ」ことを忘れてしまいがちであった。STAMP/STPA で CS 図を記載し、それをベースに検討することで、システム全体を俯瞰でき、本来の目的を意識したうえで、構造上不足している点を見つけやすくなった。また、画像による待避状況の把握など、これまでにない手段も含めてシステム構築のアイデアが広がる効果も見られた。

今回、STAMP/STPA 手法で特徴的ともいえる CS 図を中心に、FTA とも組み合わせることで、それぞれの方式を相互に補完した形で GPS 列警の改善案の検討を行った。STAMP/STPA の一つの活用手法として参考になれば幸いである。

2.1.5. まとめ

今回は、STAMP/STPA 手法を用いた安全解析手法を GPS 列警の改善に用いる中で、CS 図の改善による安全システム全体の見直し検討を実施した例を報告した。CS 図を作成し、安全検討を進めていく中で、安全責任を担保するコントローラー自体の見直しにより、新たな列警システムのアウトラインの作成を行うことができた。

一般的に STAMP/STPA 手法は、分析対象システムのハザードシナリオを見つけるのに有効であるが、今回はシステム構成の抜本的な見直しの検討に STAMP/STPA を適用することによりアイデアが広がり、STAMP/STPA の新たな活用方法を見出すことができた。特に、人も絡んだ複数システムが複雑に絡み合う場合の相互作用については、ダイナミックで自由度の高いハザードシナリオを生成できる STAMP/STPA の強みを生かす分析ができた。さらに、スタティックでハードウェア構成に立脚した深い分析を行う FTA と組み合わせることで、実用的な結論が得られた。

また、IoT 技術の進展に伴って、機械と人間の相互協調――Safety 2.0――を図ることにより、システムの役割分担が従来から変化し、システム全体としてさらに高度なレベルでの安全性を実現できることが、CS 図から見出すことができた。

実際には技術的、経済的な課題等、解決するべき内容は残っている構成であるが、列車接近警報装置のクローズドループ化や、携帯端末の位置情報などによる線路内外の識別など、安全システム構築に向けた自由な発想を導き出せたことは、STAMP/STPA の特徴であると考えられる。

2.2. 高齢者見守りサービス

2.2.1. はじめに

IoT 技術の進展と普及に伴って、その応用例としてよく取り上げられているのが、高齢者（独居）世帯の見守りサービスである。高齢者世帯の監視は、プライバシーに配慮した形で生活動作情報を得る必要があるが、多様なセンサーと通信手段を持つ IoT 機器がこれを可能にする時代になったということであろう。IoT を利用したサービスの価値は、独立のベンダーが相互に情報を利用しあってサービスを提供するため、多様で付加価値の高いサービスが提供できることにある。分散協調型のシステム開発といえる。集中統合型のシステム開発と異なり、サブシステム相互の情報利用に矛盾が出てきてサービスが安全を脅かすことになったり、何らかの被害が出た際の責任の所在があいまいになるといった問題点も指摘できる。

こうした今後の IoT サービスの不具合を少なくしたり、責任の所在を明確にするという観点から STAMP/STPA がどのように役立つかを、具体的事例をもとに議論してみたい。事例としたのは、高齢者の見守りサービスシステムであり、実際にシステムが提供され、さらには、不具合を起こしたものである。ただし、プライバシーに配慮して、サービスの提供形態を抽象化して説明することにする。

2.2.2. 高齢者見守りサービスシステムの構成

図 2.2-1 に今回想定した高齢者見守りサービスシステムの構成を示す。高齢者が住居に在室の場合、生活動作を、電気・ガス・水道メータの動きやトイレ・冷蔵庫などの動作音を用いて、健全な生活をしているかどうかをプライバシーに配慮した形で監視できる。このとき、一定時間（例えば 10 時間）連続で生活動作の兆候が検知できないとき、高齢者に何らかの異常があったとして、IoT 見守りシステムが警備サービスシステムに通報し、緊急の介護駆けつけを行う。この事例では、在室か不在かの判断は施錠センサーを用いて行い、その結果を遠隔地にある警備サービスシステムに表示し、そこで、在室で生活動作なしの警報が出た場合に、警備員が現地に駆けつけて介護を行うと仮定している。警備員呼び出しブザーも備わってはいるが、今回の分析では省略した。また、実際のサービスでは、地域の生活補助員の定期訪問などもあるので、「在室・不在・生活動作なし」といった表示は、各住居の扉に表示する場合もあるかもしれないが、ここでは、サービスシステムの本質的挙動のみに着目して、図のような抽象化を行った。

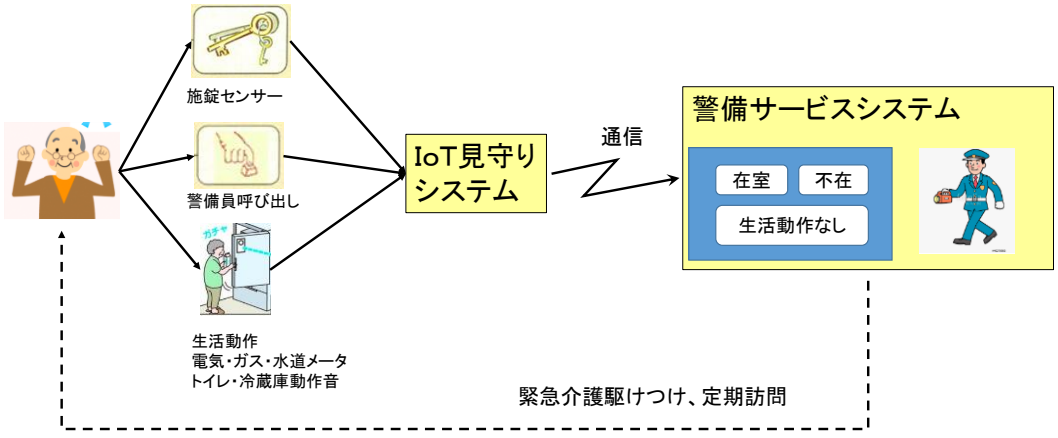


図 2.2-1　高齢者見守りサービスシステム

生活動作の検出には下記のような形態が考えられる。この検出アルゴリズムもシステムの大事な技術ではあるが、ここでの事例としては、このいずれか一つを採用するという仮定で分析を進める。

(1) 生活動作を状態として捉える場合

- 水道、ガスは、流量計があるとし、流量ゼロが 10 時間以上続くと「生活動作なし」とする。(水道出しっぱなし、ガスストーブつけっぱなしなどは、生活動作ありになるので注意が必要)

(2) 生活動作をイベントとして捉える場合

- 電気は、冷蔵庫など常時通電機器があるので、On/Off のイベントを検知して、イベントがない状態が 10 時間以上続くと「生活動作なし」とする
- ガスコンロやガス風呂なども、使用開始・終了をイベントとして捉えて生活動作ありと判断する
- トイレ使用、冷蔵庫開閉音などは、振動センサーでバースト音を捉えて生活動作とする

見守りサービスシステムへの要求仕様は、上記の生活動作の有無が検出可能という条件のもとで、高齢者が一人で在室の時に、何らかの異常で生活動作が連続して 10 時間以上ない時に介護に駆けつけるということである。在室・不在の判断は、ドアの施錠センサーを用いて行う。

このようなシステムを考えた際に、最初に以下の仕様を作成した。

① ドアを中から施錠で、監視開始（在室表示）
② ドアを外から施錠で、監視解除（不在表示）
③ 在室時に中から解錠しても、監視継続（在室表示）
④ 不在時に外から解錠しても、監視解除継続（不在表示）
⑤ 在室時に 10 時間以上生活動作がなければ、「生活動作なし」警告を表示し警備会社に通報
⑥ 不在時には「生活動作なし」警告は表示されない
⑦ 「生活動作なし」警告は、10 時間経過後でも、生活動作が検出されるか、または、高齢者が外出し、ドアを外から施錠するか、警備員が見守りシステムにアクセスして強制解除すれば、解除される

2.2.3. STPA による安全分析

以上の仕様に基づいたシステムのアクシデントとハザード、安全制約は下記のように定義できる。

アクシデント：高齢者が在宅時に倒れた際、介護に駆けつけることができない
ハザード　　：高齢者の生活動作が 10 時間以上ないにもかかわらず放置される
安全制約　　：高齢者の生活動作が 10 時間以上ない場合、必ず介護に駆けつける

以上の仕様に基づいて、作成した CS 図が、図 2.2-2 である。抽象化したコンポーネントを、「住居・単身高齢者」、「IoT 見守りシステム」、「警備サービスシステム・警備員」という三つとして考

える。この時の CA（コントロールアクション）は、上記①②に対応して、CA1:内鍵施錠による監視開始（在室表示）、CA2:外鍵施錠による監視解除（不在表示）となる。これは、「住居・単身高齢者」のコンポーネントの安全責任として R1、R2 を果たすための CA といえる。一方、IoT 見守りシステムの安全責任は、R5～R8 に記したように、施錠センサーによる在室・不在の判断、生活動作の監視（10 時間連続して生活動作がない場合の通報）、生活動作なしの警報解除、がある。これに対応した CA は、CA3:介護要請、CA4:警報解除である。この CA に必要な情報は、施錠センサーによる在室・不在の判断であり、生活動作情報フィードバックである。ここで生活動作情報を CA にせずにフィードバックとしたのは、介護要請という IoT 見守りシステムの通報行動が安全責任の本質であり、生活動作情報というのは、その安全責任を果たすための情報の一つであるという考え方によっている。生活動作情報を知らせることを安全責任としてしまうと、単身高齢者が異常情報を知らせる責任があることにもなってしまう。例えば、水道を閉め忘れたまま意識不明になったような場合を想定すると、生活情報は健全なままになって意識不明を見逃すことにつながるが、これは、単身高齢者自身のヒューマンエラーという責任になってしまうこともあり得る。このように、安全責任の存在と CA、フィードバックの意味を明確に定義し、ステークホルダーの共通の理解を得ておくことが大事であることを示している[2]。

最後に、警備サービスシステム・警備員コンポーネントの安全責任は、在室・不在・生活動作なし警告の表示と警備員の介護駆けつけであり、CA は、CA5 : 緊急時の介護となる。この図で可視化してみてわかることは、緊急時の介護では、施錠ドアを警備員が開ける必要があるということである。つまり、ドアのカギは、単身高齢者だけでなく、警備員も持っているということに気づく。自宅のカギを親族ではない第三者が持っているのは不安かもしれないが、警備会社がきちんと管理した状態で預けておけば問題は少ないといえる。このとき、警備員が解錠できるのであれば、介護訪問で住民が元気であることが確認でき帰社する際には、施錠もできるということに気づく。そうすると、図 2.2-3 に示すような CA、CA6 : 警備員による施錠、があることに気づく。この CA は、同時に、IoT 見守りシステムへの監視解除の指示にもなる。そこで、図 2.2-3 の安全責任 R4-1 として「介護後問題なければ施錠して帰社する」という項目を追加した。ただし、この安全責任は、同時に監視解除を意味することになり不適切な安全責任の定義になっていることが、後の分析で分かることになる。

もう一点、図 2.2-3 の CS 図を見て気がつくことは、監視開始、監視解除という住民から IoT 見守りシステムへの CA に対するフィードバックがないことである（図内には点線でこれを示した）。鍵の施錠・解錠操作で、意識せずに監視を開始したり解除したりすることは便利ではあるが、単身高齢者の自己責任として監視を依頼するという考え方をとると、監視状態を単身高齢者自身に知らせるフィードバック情報を提供したほうが良いことにもなる。このフィードバックは、室内で住民自身が確認できるようなランプやディスプレイのような表示装置で可能になる。

[2] 第 1 章で述べたとおり、コントロールアクションとフィードバックを取り違えても、分析されるハザード誘発シナリオの結果には重大な違いはない。

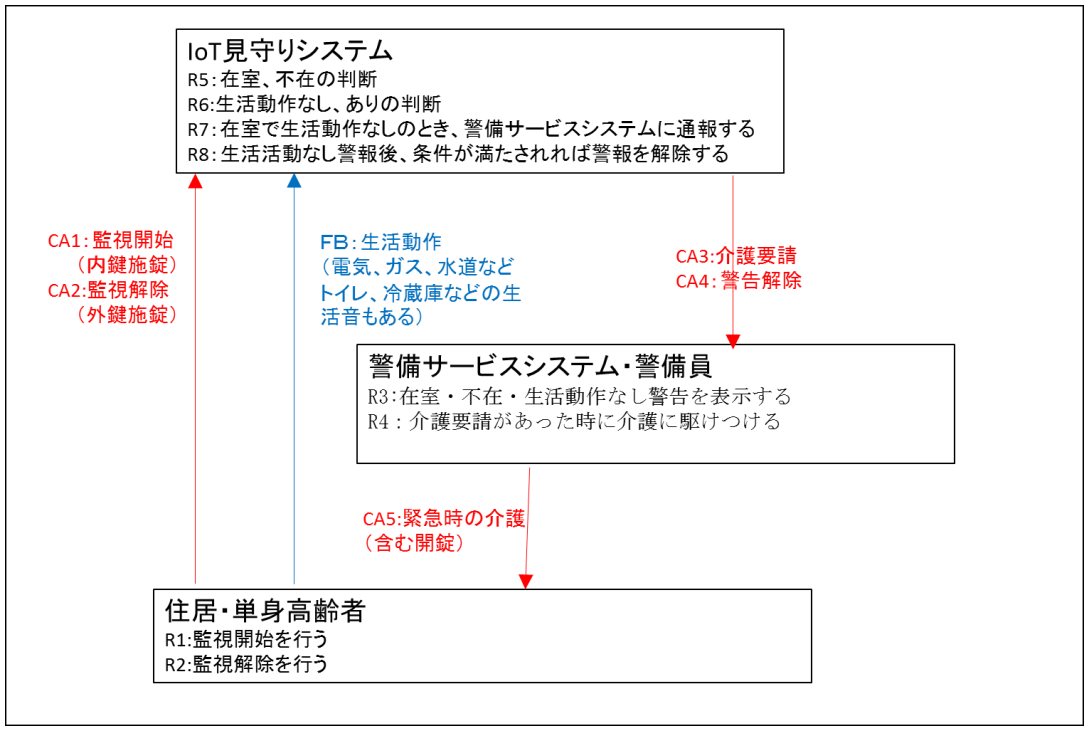


図 2.2-2　高齢者見守りシステムの安全 CS 図

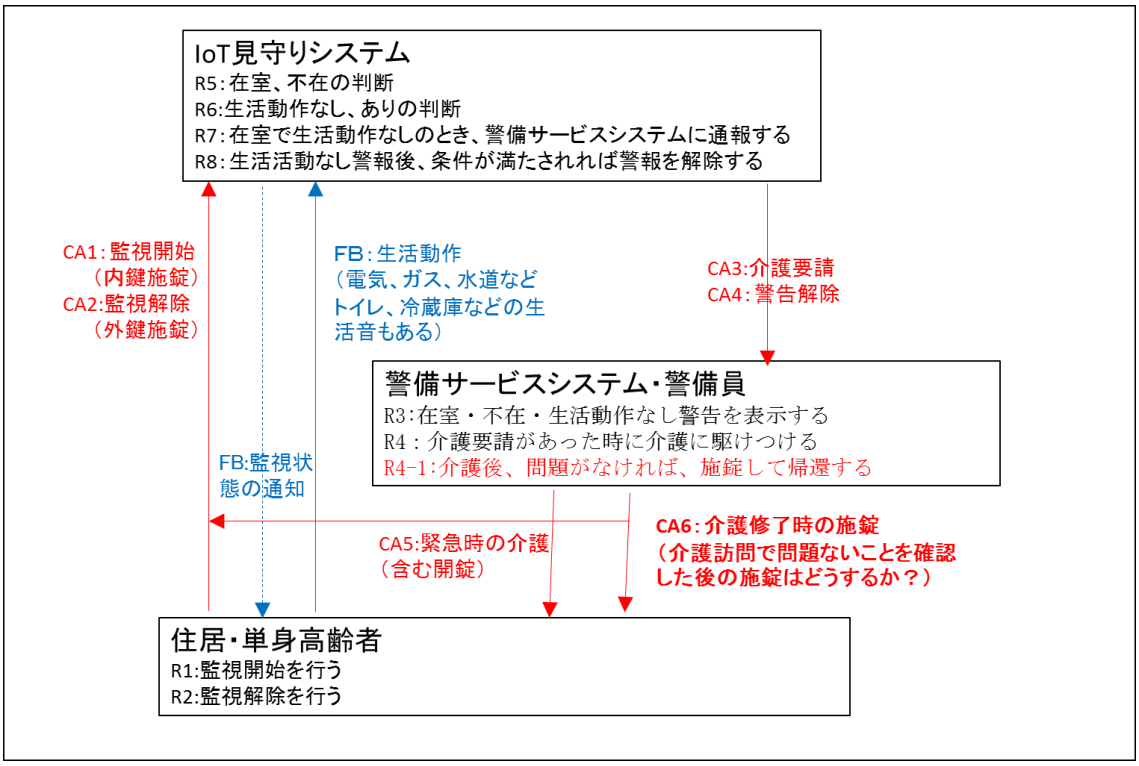


図 2.2-3　高齢者見守りシステムの安全 CS 図

こうして作成した安全 CS 図に基づいて、STPA 分析を行った結果を以下に示す。今回の記述では、STAMP の効果が現れている UCA までの分析を示しており、また、4 通りの UCA も、焦点となる Not Providing（NP）と Providing causes hazard（P）の二通りのみに絞っている。また、対策

（ア）～（オ）も併記したが、この詳細は後述する。

① CA1 監視開始（内鍵施錠）

NP：（内鍵施錠忘れ）在室でも監視しないのでハザード　　対策（ア）

P：（内鍵施錠）監視を開始するが、誤った生活動作（水道閉め忘れなど）で、実際は異常なのに、異常なしと誤解してハザード　対策（イ）

② CA2 監視解除（外鍵施錠）

NP：（外鍵施錠忘れ）　在室してないのに監視継続は、安全側

P：（外鍵施錠）　本人以外の人が施錠したとき、本人が中にいると、監視解除になるのでハザード　対策（ア）または（ウ）

③ CA3（警備会社への介護要請）

NP：（警備員へ確認を要請しない）　警備員が確認に行かずにハザード　対策（エ）および（オ）

P：（警備員確認要請）　警備員が確認に行くので安全側

④ CA4（生活動作なしの警報解除）

NP：解除なしは安全側

P：警備員が介護に出かける前に解除してしまうとハザード　対策（オ）

⑤ CA5（警備員の介護行動・含む外からの解錠）

NP：（警備員が介護に行かない）　確認に行かないのでハザード　対策（エ）および（オ）

P：（警備員が介護に行く）　安全側

⑥ CA6（警備員の外鍵施錠）

NP：介護に行って本人が元気であることを確認した後に帰る際、外鍵をかけない。その後、本人が内鍵を施錠しないと監視が始まらないのでハザード　対策（ア）または（ウ）

P：本人が元気な時、外鍵を閉めて帰ると監視解除（本人在室なのに不在と思ってしまう）でハザード　対策（ア）または（ウ）

以上の結果でハザードに至るシナリオを下線で示した。これを分類すると、

(1) 内鍵施錠忘れ、または、第三者による外鍵施錠により、在室なのに監視をしない

(2) 生活動作の誤検知（住民の誤動作も含む）により、在室で生活動作がないのに生活動作ありと誤判断してしまう。

(3) 警備員への介護要請の伝達ミス（システム故障、監視画面の見逃し、警備員間の伝達ミスなど）

という３通りに分かれる。これらのハザードシナリオを防ぐための対策案はいくつか考えられるが、その前に、安全責任の在り方を考えておく。まずは、単身高齢者自身が監視の開始と解除に関しては全責任を負うという考え方である。このとき、(1)の第三者による外鍵の施錠による監視解除は余計な動作になる。この場合、第三者（警備員）に対して、外鍵施錠を禁止するという手順を強いるということになる。そうすると、施錠センサー情報で監視の開始と解除を行うよりも、明示的に監視開始と解除のボタンを設けた方がよいことにもなる。前述のように、単身高齢者からの監視開始と解除という CA へのフィードバック情報として、自身が監視されていることを知ることが出来る何らかの表示装置を設置する案もある。

このような単身高齢者自身の自己責任に基づくシステム設計のほかに、高齢者を想定したサービスでは、IoT 見守りシステムを知能化して監視責任を機械側も分担するという考え方もある。下記の対策にも示したように、見守りシステムのアルゴリズムをより知能化して、住民が監視操作を忘れた場合に生活動作の有無から監視を自動再開するという考え方で、機械側の安全責任をより大きくするということになる。上記の(2)、(3)の不具合は、サービスシステム自身の技術的、組織的な対応をより改善するということであり、安全責任としては、サービス提供側が負うという考え方になる。

これらの安全責任とハザードシナリオを考慮すると、下記のような対策案が考えられる。各 UCA のハザードとの対応は既に示したとおりである。

(ア) 監視開始忘れ

- 不在時（監視解除時）でも、生活動作が一度でも検出されれば、監視を開始する。（不在時でも「生活動作なし」警告を出す。）

(イ) 誤った生活動作の検出（水道閉め忘れなど）

- 複数の生活動作検出センサーを設ける。

(ウ) 本人以外が外鍵を施錠して監視解除にしてしまう

- 対策（ア）で監視を自動再開。
- ドアの外に監視再開ボタンをつけて警備員の責任で施錠した場合でも監視再開できるようにする。
- 室内に監視状態表示盤を設け、単身高齢者自身で在室時に監視されていない場合は、これを再開することが出来るようにする。
- 警備員の手順書を整備し、警備員は外鍵を施錠せず、単身高齢者に内鍵を閉めるよう促す。

(エ) 通信ライン、システムの故障は定期検査で対応。警備サービスシステムの警報を見逃すヒューマンエラーは、わかり易い表示と警報音の併用でなくす。組織内での人的通報ミス、介護先の住所連絡ミスなどは、訓練で補う。

(オ) 警報解除は、現地の見守りシステムでしかできないとする。または、介護後の報告を持って、警報を解除する手順とする。

2.2.4. 安全責任と権限委譲・移譲

前節の STPA 分析の結果に基づいて、安全責任の問題を整理してみる。このために、図 2.2-3 で隠れているコンポーネントとして、サービス提供会社（IoT 見守りシステムならびに見守りサービス全体を提供する組織）と、サービスの費用負担も想定したサービス依頼者を追記した（図 2.2-4）。

この範囲で、システム全体の安全責任の主体として、次の 3 通りを考えてみる。

(1) 単身高齢者自身が、サービス依頼者であり、かつ、監視と解除に責任を持つ場合

この場合、単身高齢者自身がサービスシステムを購入し、その使用（監視と解除、介護）に責任を持つ。そのためには、監視状態にあるかどうかをフィードバック状態として知らせることが望ましい。また、「警備員が訪問した後に不要な施錠をしない」という警備会社への運用責任を委譲（Delegation）する方策も考えておかねばならないかもしれない。「委譲」としたの

は、間違って監視解除を警備員がしたとしても、その最終責任は単身高齢者にあるという考え方（自己責任）をとった場合である。しかしながら、見守られるべき対象者が一人住まいの高齢者であり、かつ、見守りサービスシステムの本来の社会的目的（孤独死の回避や地域での介護）を考えると、このような安全責任の考え方は割り切りすぎともいえる。

(2) 警備サービス会社（含む警備員）がサービス提供者となり、監視と解除、ならびに、緊急時の介護に責任を持つ場合

警備サービス会社自身が、サービス全体の責任を持つという考え方である。この場合、サービスの契約は、単身高齢者自身より、単身高齢者の保護者である別世帯の家族である場合が多いといえよう。その時、サービス失敗の責任をどこまで警備サービス会社が負うかは契約書として明確に決めておく必要がある。監視開始や監視解除には単身高齢者のミス、警備員のミスがありうるが、それぞれのミスの状況でサービス失敗の責任の取り方を決めておかないと混乱の元になりうる。生活動作の検出ミスも、IoT 見守りシステムの要求仕様の欠陥によることがあり得るが、これも、警備サービス会社と、IoT 見守りシステムの開発会社の間の瑕疵条項の契約として決めておくことが望ましい。要求仕様まで警備会社が出し、システム運用に関する責任を全て警備サービス会社に移譲（Transfer）されたとみなせる場合は、全ての責任を警備サービス会社が持つことになる。この立場を明確に意識すると、見守りサービス失敗を防ぐための対策としても、前項で分析したいくつかの対策案がとれる。これらは、サービス失敗後の対策として行うと大きなコストがかかってしまうが、設計時の要求仕様の段階で考慮しておけば、最小のコストでのサービスシステムの性能向上につなげることが出来る。これは、STAMP/STPA の安全設計の考え方の重要な部分である。

(3) サービス提供会社が、全体のシステム並びにサービス運用に責任を持つ場合

IoT 見守りシステムを開発した会社が、サービス全体の責任を持つという考え方である。この場合、サービス失敗の責任はサービス提供会社自身がすべて持つことになる。ただし、日常のサービス運用では、警備サービス会社が主体となるため、どこまでの責任を委譲（Delegation）するかは契約で決めておく必要がある。この考え方についても、サービスシステムの運用前に明確化しておくことで、サービス失敗後のリコールのような余計なコストを防ぐ可能性を高めることが出来よう。

以上のように、3通りの運用形態に応じてサービス提供の異なる安全責任の考え方を例示した。これらは、あくまで例であり、どのような安全責任の分担にするかは、見守りサービスに関わるステークホルダーの間での合意として決めるべきものである。

STPA による CS 図は、サービス提供に関わる複数のステークホルダー間での安全責任の取り方（委譲と移譲）を明確化できる。これは、事前のシステム設計や、サービス失敗が仮に起こった場合での事後対応などに役立てることが期待できる。

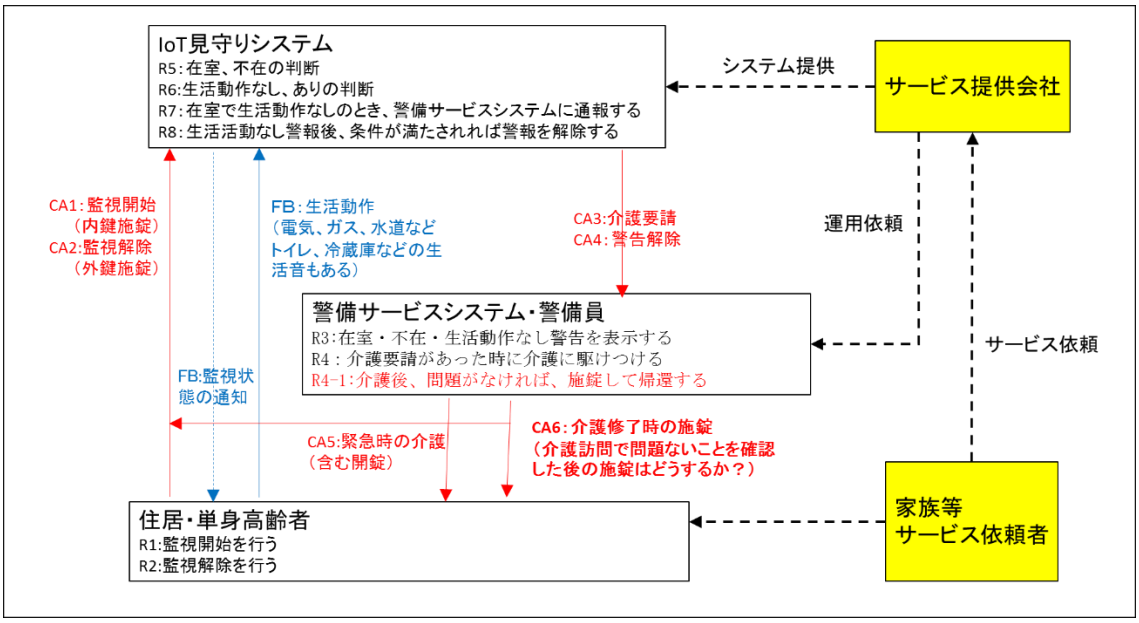


図 2.2-4　安全責任と権限委譲(Delegation)・移譲（Transfer）

2.2.5. まとめ

今回の事例は、事故原因の分析という面では簡単であり、すぐに要因に気づく人がいるかもしれない。しかしながら、安全責任の在り方という視点では、必ずしも唯一の答えがあるわけではなく、CS 図による安全責任の可視化を通して、いくつかの選択肢を示せた。STAMP/STPA の手順は、複数のステークホルダーの間で安全責任を含んだ設計思想やハザードシナリオをわかり易い形で共有化し、相互の視点で議論することで想定外の設計漏れや運用手順漏れを防ぐ方法といえる。今回の事例では、下記の具体的な観点から STAMP/STPA の有用性が示されたといえる。

(1) CS 図と CA・フィードバックの作成を通して、安全責任が明確になり、複数の立場の異なるステークホルダーで共有できる。今回いくつかの仕様漏れともいえる欠陥が UCA 分析で検出された。監視の開始と解除を施錠センサーで行ったため、本人のエラー、第三者のエラーを起こしやすくなったといえる。監視の開始と解除を完全な自己責任とする場合は、明確な監視開始・解除ボタンをつけるか、少なくとも、監視状態を住民自身が室内で知ることが出来るようにしておくという改善策がありうる。一方で、監視開始と解除を機械の判断に頼るのであれば、施錠センサーだけに頼るのではなく、機械側の在室判断のアルゴリズムを知能化したり、別のセンサーを併用して在室・不在の判断をするという改善策が出てくる。このような安全責任の明確化と、それらを設計仕様にどう組み込むかの判断は、複数のベンダーが関係する IoT システムでは特に重要になろう。

(2) 安全要求仕様の漏れのない作成に役立つ。CS 図とそれぞれの CA を明示化することで、第三者の外鍵施錠により監視解除されるという仕様欠陥に気づく。今回は、警備サービスシステムという遠隔監視盤での状態表示システムによる運用を仮定したが、これにより、警備会社側での監視負荷の増大や見逃しエラーの増大も想定される。これも、別のシステムを想定した分析と比較することで、より信頼性の高いシステムにすることも可能である。

(3) STPA の手順を全て踏襲しなくても役に立つ結果が導出できた。本例では、UCA を、「Not

Providing」と「Providing」だけで簡略化して評価し、UCA 以降のハザード誘発シナリオまでは評価しなかった。CS 図の各コンポーネントの安全責任と CA の整合性、フィードバックの必要性だけに注目し、UCA の分析結果を見ることで、システム全体の検討漏れ、改善策、異なる安全責任の考え方などを抽出できた。STPA の手順をそのまま踏襲するのでなく、それを簡略化することで、却ってシステム全体の安全機能を俯瞰化できる場合もある。

2.3. 自動車製品ライフサイクル

2.3.1. はじめに

本節では、「クルマと人」との相互作用に着目した自動運転システムの分析例を解説する。自動運転システムは、「走る」、「曲がる」、「止まる」といった主要機能を自動運転システムが制御するために、IT（Information Technology）や IoT（Internet of Things）によって「クルマとクルマ」、「クルマと社会インフラ」、「クルマと人」の相互作用により成り立つ大規模かつ複雑な自動運転システムである。このように複数のシステムやサービスがより複雑に絡み合う自動運転システムの普及が今後見込まれており、これまでに経験のない想定外の危険事象に遭遇する可能性が高まっている。このようなシステムにおける潜在的な危険事象を、いかに開発の早い段階で想定内と認知し、安全設計に取り込むことができるかが、自動車産業における重要な課題となっている。システムは構想から廃棄までの製品ライフサイクル全体を通して安全でなければならないが、これまでの STAMP/STPA の適用は、ライフサイクルの中でもエンドユーザが対象システムを利用する「使用ステージ」に対する分析が中心であった。しかし、使用ステージだけでなく、ライフサイクル全体を俯瞰的に捉え、開発の早い段階で STAMP/STPA を適用することによる効果が期待されている。本節では、分析対象を製品ライフサイクルに拡張し、使用ステージ以外のライフサイクルステージに跨るユースケースを設定して STAMP/STPA による安全分析を試行した事例を紹介する。また、IPA が提供する分析支援ツール「STAMP Workbench」の有効性についても紹介する。

2.3.2. 分析対象システムの概要

分析対象システムの概要は以下の通りである。

- 日本の高速道路または自動車専用道路で自動運転を可能とする、自動運転システム。自動運転中、ドライバーは運転以外のことをしてもよい。（＝SAE J3016（2018） Level3 Conditioned Driving Automation 相当）[SAE2018]
- システムは、車両を目的地に向けて自動操縦する“自動運転モード”、ドライバーの手動運転操作に従って操縦する“手動運転モード”を備えている。この 2 つのモードは、ドライバーが危険を感じたとき、または、システムが制御継続できないときなどに切り替えることができる。

2.3.3. STAMP/STPA による安全分析

本事例で想定したライフサイクルと STAMP/STPA による分析イメージを図 2.3-1 に示す。

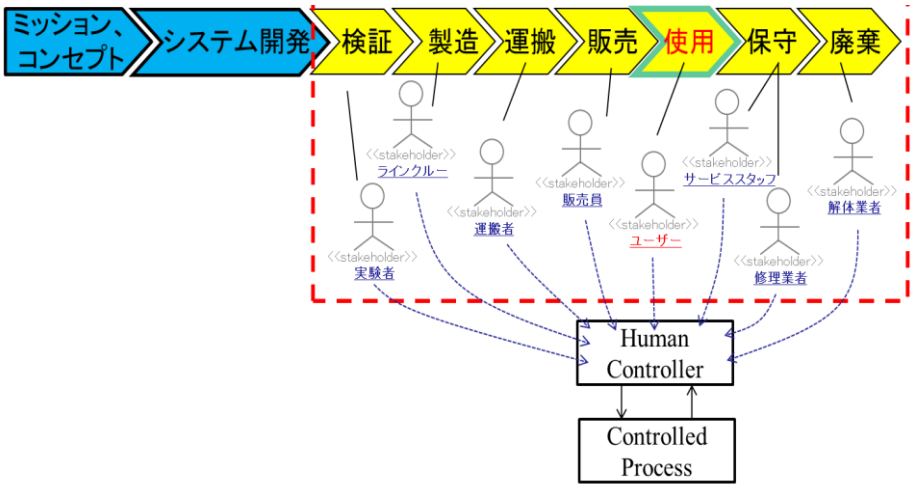


図 2.3-1　システムライフサイクルと想定ステークホルダーの関係

●ユースケースの設定

本事例では、分析作業をワークショップ形式で実施した。ワークショップでは、3 つのユースケースを想定し、ユースケース毎に 2 つのチームを編成し、計 6 チームで分析を試行した。各ユースケースに割り当てられた 2 つのチームの内、一方は分析支援ツール「STAMP Workbench」を使用し、もう一方は支援ツールを使わずに分析を試行した。

検討対象とした 3 つのユースケースを図 2.3-2 に示す。

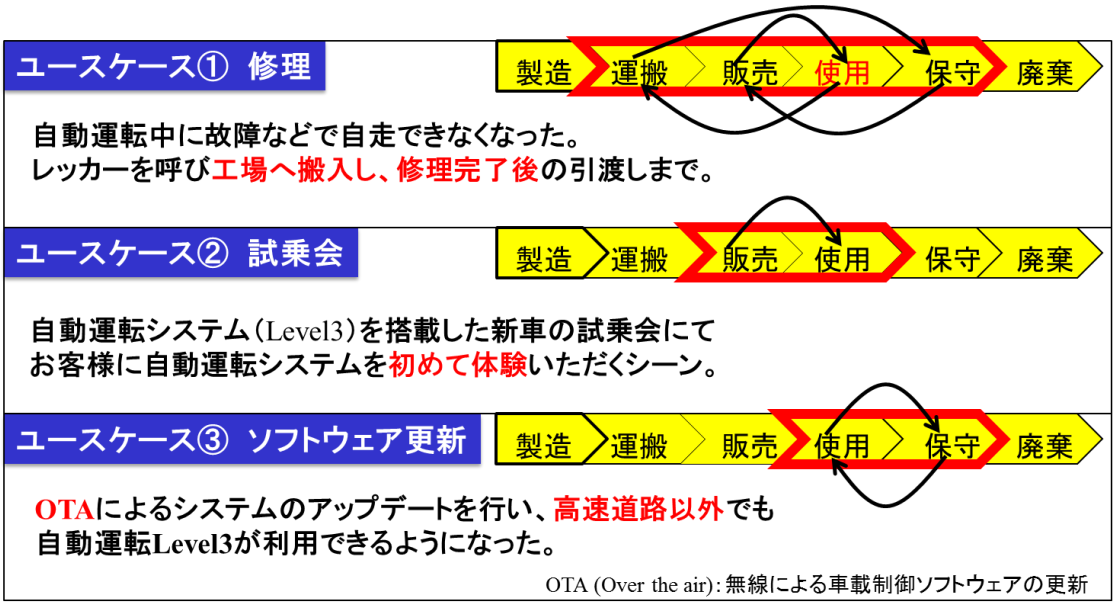


図 2.3-2　ユースケースの拡充

また、ユースケース拡充ワークショップの 6 つのチームとその事例の一覧表を以下に示す。

表 2.3-1 ユースケース拡充ワークショップの6つのチームとその事例の一覧表

ユースケース	チーム	ツール使用／未使用	事例名称
① 修理	A-1	未使用	ドライバーと修理関係者の相互作用に着目した STPA 試行
	A-2	使用	自動車保守フェーズへの STPA 試行
② 試乗会	B-1	未使用	試乗会での販売員とお客様の相互作用に着目した STPA 試行
	B-2	使用	損失（「売れない」）に注目した販売シーンにおける STPA 試行
③ ソフトウェア更新	C-1	未使用	OTA によるシステムのアップデートに関する STPA 適用事例
	C-2	使用	ドライバー/車両/OTA サーバ間の相互作用に着目した STPA 試行

本ワークショップにおける STPA による分析は、以下の手順で実施した。

(1) 登場人物、アクシデント（損失）／ハザード／安全制約の定義
(2) CS 図の作成
(3) UCA の識別
(4) 損失シナリオの特定（または HCF の特定）
(5) 対策の検討（必要に応じて CS 図の更新）

●分析

本書では、事例 B-2 の「損失（「売れない」）に注目した販売シーンにおける STPA 試行」について紹介する。

事例 B-2 は、ユースケース②の「自動運転システム（Level3）を搭載した新車の試乗会にて、お客様に自動運転システムを初めて体験いただくシーン」である。また、分析支援ツール「STAMP Workbench 1.0.1」を使用し、ツール画面のガイドに従い分析した。

【分析手順 1】登場人物、アクシデント（損失）／ハザード／安全制約の定義

分析の目的を明確にするために、取り扱う損失の観点と分析に必要な前提条件を仮定した。

- 取り扱う損失の観点
  - 安全に関連する損失
  - 販売機会の損失
- ユースケースに追加した条件
  - 場所：試乗のために準備した会場（公道ではない）

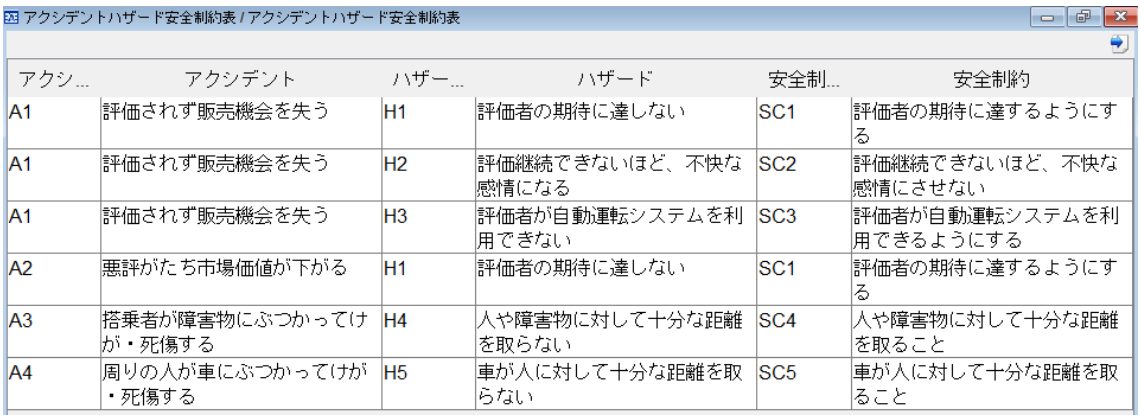
- 試乗プログラム：走行車線が 2 レーンある道路において、前方に障害物となる物体を設置し、前進している車両が自動的に障害物を避けるようレーンチェンジして走行し続ける

このように損失対象、分析対象のユースケースおよび条件を考慮し、本事例における登場人物を定義した。

- スタッフ：試乗会スタッフで、お客様に試乗内容および車両の機能について説明を実施する
- 評価者：試乗会に来たお客様で、試乗して車両を操作し、車両を評価する
- 車両：お客様の操作により動作し、また、自動運転機能により障害物を避けて走行する
- 環境：試乗会場の環境（障害物は、環境に含まれる）

また、取り扱う損失の観点を元に、考えられるアクシデント（損失）、ハザード、安全制約を以下のように導出した。

表 2.3-2　試乗会のアクシデント（損失）、ハザード、安全制約の識別
（STAMP Workbench ツール画面）


アクシデントハザード安全制約表 / アクシデントハザード安全制約表

アクシ...	アクシデント	ハザー...	ハザード	安全制...	安全制約
A1	評価されず販売機会を失う	H1	評価者の期待に達しない	SC1	評価者の期待に達するようにする
A1	評価されず販売機会を失う	H2	評価継続できないほど、不快な感情になる	SC2	評価継続できないほど、不快な感情にさせない
A1	評価されず販売機会を失う	H3	評価者が自動運転システムを利用できない	SC3	評価者が自動運転システムを利用できるようにする
A2	悪評がたち市場価値が下がる	H1	評価者の期待に達しない	SC1	評価者の期待に達するようにする
A3	搭乗者が障害物にぶつかってけが・死傷する	H4	人や障害物に対して十分な距離を取らない	SC4	人や障害物に対して十分な距離を取ること
A4	周りの人が車にぶつかってけが・死傷する	H5	車が人に対して十分な距離を取らない	SC5	車が人に対して十分な距離を取ること

【分析手順 2】CS 図の作成

前述の登場人物の定義およびユースケース内容から、CS 図を以下のように作成した。

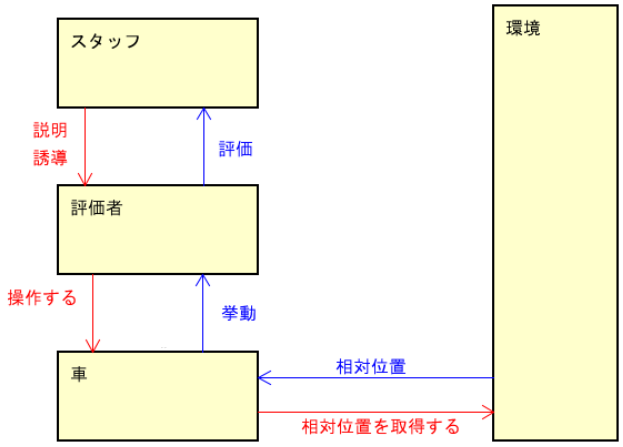


図 2.3-3　試乗会の CS 図（STAMP Workbench で作成）

【分析手順 3】 UCA の識別

UCA の識別結果を以下に示す。

表 2.3-3　試乗会の UCA 抽出結果（STAMP Workbench ツール画面）

UCA表0 / UCA表

No	CA	From	To	CA提...	Not Providing	Providing causes hazard	Too early / Too late	Stop too soon / Applying too long
1	説明	スタッフ	評価者		(UCA1-N-1) スタッフが, 説明をしないことで, 評価者が正しくシステムを作動できない [SC3] (UCA1-N-2) スタッフが, 説明をしないことで, 評価者が気分を害して帰ってしまう [SC2] (UCA1-N-3) スタッフが, 説明をしないことで, 評価者が正しく自動運転システムを操作できずに障害物に衝突してしまう [SC4][SC5]	(UCA1-P-1) スタッフが,説明を過剰に与えて, 評価者が飽きて/気分を害して帰ってしまう [SC1] (UCA1-P-2) スタッフが,正しくない説明を与えて, 評価者が自動運転システムを正しく操作できずに障害物に衝突してしまう [SC4][SC5]		
2	誘導	スタッフ	評価者					
3	操作する	評価者	車		評価者が, 自動運転システムを操作せず,自動運転システムの良さを知れない(が別に良い) 評価者が, 自動運転システムを操作しない (安全を脅かさないので問題ない)			
4	相対位置を取得する	車	環境		(UCA4-N-1) 車が,相対位置を取得しないことで,障害物を回避できずぶつかってしまう [SC4][SC5]	(UCA4-P-1) 車が,相対位置を過大に取得し,障害物にぶつかってしまう [SC4][SC5] (UCA4-P-2) 車が,相対位置を過少に取得し,早くレーンチェンジしてしまい, 評価者が不信感をもつ [SC1]	車が,相対位置を早く取得するだけでは、挙動には現れないので、問題ない (UCA4-T-1) 車が,相対位置を遅く取得し,レーンチェンジが間に合わずぶつかってしまう [SC4][SC5]	

【分析手順 4】損失シナリオの特定（または HCF の特定）

UCA として抽出された、(UCA1-N-1)「スタッフが、説明をしないことで、評価者が正しくシステムを作動できない」に対する HCF として、「自動運転車両の試乗対応マニュアルや教育体制が整備されていない」があげられる。この他に HCF となり得るケースがないかを確認するために、「なぜ評価者は正しくシステムを作動できないのか」という観点で、HCF 導出ヒントワードの一つである「プロセスモデルの矛盾、不完全、不正解」から、評価者のプロセスモデルを検討した。その結果が、以下である。

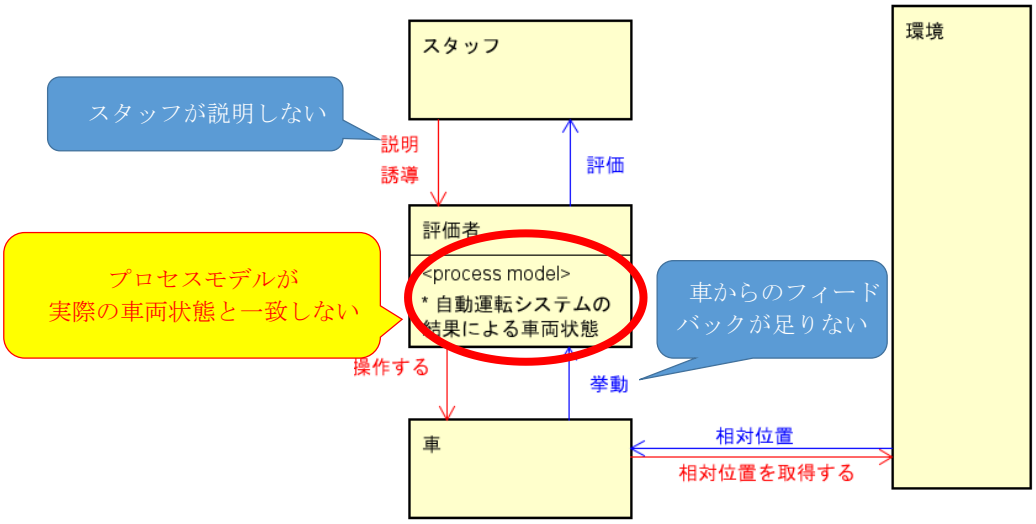


図 2.3-4　CS 図に評価者のプロセスモデルを追加

プロセスモデルを追加した CS 図から、評価者のプロセスモデル「車両の状態」が、実際の車両の状態と一致していない（評価者が自動運転モードなどの車両の状態を正しく認識できない）ことが、評価者が正しく操作できない原因の一つとして導出できる。これは、【分析手順 3】の UCA 抽出において、評価者が車に対して「操作する」という CA に関係する（表 2.3-3 で空白になっていた No. 3 の UCA を新たに抽出することができた、ということである）。このようにヒントワードやプロセスモデルなど CS 図を併用した分析により、初回分析時ではなかった新たな気づきを得られた事例でもある。

【分析手順 5】対策の検討

以上の分析から、(UCA1-N-1)「スタッフが、説明をしないことで、評価者が正しくシステムを作動できない」に対する対策として、以下が導出された。

対策 1 :「販売員はマニュアル、教育に従って、環境に応じて適切な運転操作アドバイスをドライバーに伝える」

対策 2 :「車両の状態を正しく認識できるように車両からのフィードバックを向上させる」

このように、STAMP/STPA の分析を通して、「売れない」というアクシデント（損失）に対する新たなハザードおよび安全制約が導出され、販売ステージにおける自動運転システムに対する新たな要求やフィードバック事項を獲得することができた。

2.3.4. 適用結果

前述の分析事例を含め、3 つのユースケース（修理・試乗会・ソフトウェア更新）について、各々、ツール使用/未使用の 2 チームで、計 6 チームで分析を試行し、いずれにおいても、分析に STAMP/STPA を活用できることが確認できた。以下に、各ユースケースで作成した CS 図を示す。

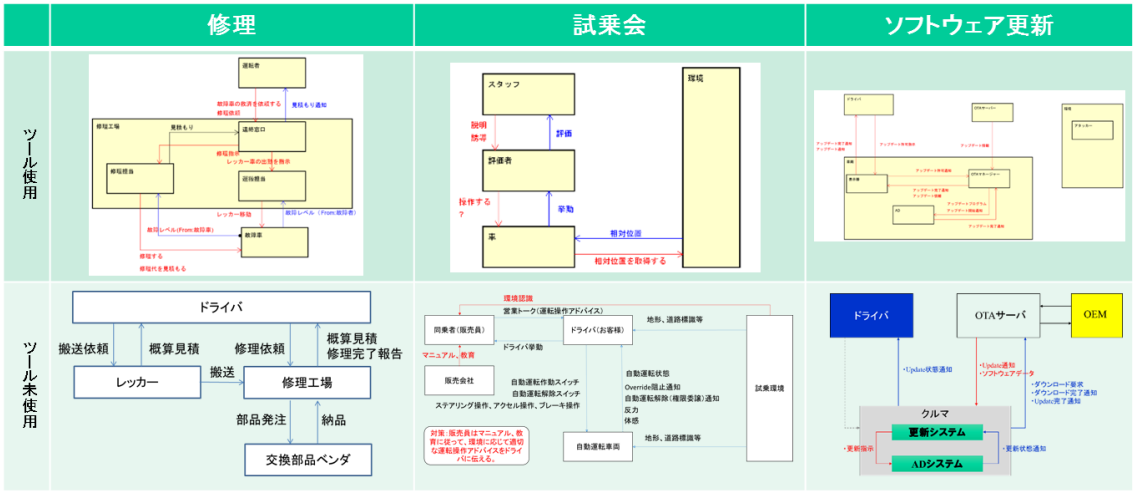


図 2.3-5　ユースケース拡充ワークショップの各チームで作成した CS 図一覧

また、その他の事例も含めた今回の試行による主な気づきを以下に示す。

【分析対象のライフサイクルステージ拡大】

- 「人⇔人」や「人⇔組織」などにおいても、コントロールする側、コントロールされる側との間に相互作用の関係性が存在し、STPA による可視化とそのリスク分析が可能であることが確認できた。
- 分析の視点が変化するため、アクシデント（損失）、ハザード、安全制約も変わる。（セールス失敗、車両の紛失、お金・・・）
- 登場人物の変化により、CS 図（およびインタラクション）も変わる。（人⇔機械、人⇔人、機械⇔機械、人⇔組織（企業））

このように、分析対象のライフサイクルステージを拡大することで、各ステージで異なるアクシデント（損失）が設定され、目的を達成するための新たな登場人物（コンポーネント）が明確になることで、また新たに相互作用も生まれる。その相互作用を持つコンポーネント間のリスク分析として STAMP/STPA を適用することで、より俯瞰的にリスク分析が実施できるようになり、要求・仕様の抜け漏れやシステム開発における検討漏れ防止に役立つ。

- 無線による車載制御ソフトウェアの更新事例では、安全だけでなくセキュリティに関係するリスクも抽出された。
- CS 図は、静的な構造を示すため、各コンポーネント間の時系列変化に対応しにくい。コントロールアルゴリズムやプロセスモデルなどの各コンポーネントが持つ振舞を含め、必要に応じてシーケンス図、アクティビティ図や状態遷移図などの動的ダイアグラムも

併用し分析に活用するとよい。

【分析支援ツール「STAMP Workbench」】

- 分析支援ツール「STAMP Workbench」を使用した場合は、サンプルファイルをテンプレートとして利用することなどにより、分析作業に集中することができ作業効率化が図れた。分析支援ツールを使用しなかったチームは、手書きしたものを電子化するなど重複した作業が多く、また修正による手戻りや分析内容の一貫性を保つための見直しなどに時間を要しており、分析支援ツールの使用/未使用により分析の進み具合にも差が生じていた。

- 試行を通して、分析支援ツール「STAMP Workbench」への新たな要望やフィードバックが抽出された。（コンポーネントの分割・集約機能のサポートなど）

2.3.5. まとめ

システムは、製品ライフサイクル全体を通してリスクを低減する必要がある。今回の試行においては、分析対象の範囲をライフサイクルの一部（使用ステージ）から拡大することで、さまざまなリスクを抽出することができた。これは、視点が変わることによりユースケースが明確化され、新たな登場人物（人・組織・環境など）とユースケースに着目したコンポーネント間の相互作用が新たに識別されたからである。このように、ライフサイクル全体を俯瞰し、コンテクストレベルでシステミックに分析しやすい点も STAMP/STPA の特徴の一つと考えられる。

また、分析支援ツール「STAMP Workbench」を用いることで、サンプルファイルをテンプレートとして利用するなどにより作業効率化が図れることを確認した。

2.4. IT システム運用（STAMP/CAST 分析例）

2.4.1. はじめに

システム理論に基づく原因分析手法である STAMP/CAST（Causal Analysis using System Theory）の事例として、実際に発生した IT システム運用における障害事故[IPA2017-2]を対象とした分析例を紹介する。STAMP はリスク分析の考え方を示したものであり、具体的な主な手法には、事前分析として設計時にリスク分析を行う STAMP/STPA（以後、STPA と略記）と、事後分析として事故原因分析を行う STAMP/CAST（以後、CAST と略記）[Leveson2012], [Leveson2017], [Nelson2008], [Leveson2017-2]がある。これまでの「はじめての STAMP」シリーズでは、STPA を中心に紹介してきており、CAST については特記してこなかった。CAST は、STAMP モデルに基づいているという観点から STPA と本質的な変わりはないが、Engineering a safer world[Leveson2012]に書かれている 9 つの手順は STPA と異なり、分かりにくいかもしれない。適用対象によっては、9 つの手順そのままでない使い方も必要となる。今回、IT システムの実際の事例を題材に CAST の適用を試みたので、実際の事例への応用の参考にしていただきたい。

CAST は、事後分析であることから、次のような特徴を持つ。

① 事故のおきた下位レベル（物理レベル）から上位レベル（システムレベル）へと分析対象を拡げていくこと

② 既に存在するシステムに対して、全体俯瞰のうえ改善勧告を出すこと

今回の事例は、CAST の適用が必須というものではないが、上述の CAST の特徴を理解する上で有用なため、CAST 分析の手順にそって、その特徴的な箇所に絞って紹介する。なお、本事例で分析対象とするシステムは「コンピューターシステムの運用」を含み、機器としてのコンピューターシステムだけでなく、人や組織も主な構成要素とする。

2.4.2. 分析対象とするシステムと事故の概要

(1) システム概要

本事例は、企業の内部で利用される「グループウェアサービス」を提供するシステムを分析対象とする。同サービスは、スケジュール管理やファイル共有、メール送受信等の機能を社員に提供するものである。図 2.4-1に示すように、統合アカウント管理とグループウェアとから構成されており、情シス部門がこれらの運用（機能を維持する作業）を行っている。グループウェアはユーザーのアカウント情報とメール情報のデータベースを持っている。このデータベースの変更・削除は運用作業者が統合アカウント管理の操作を通じて行い、グループウェアのアカウント情報を直接操作してはならないルールとなっている。

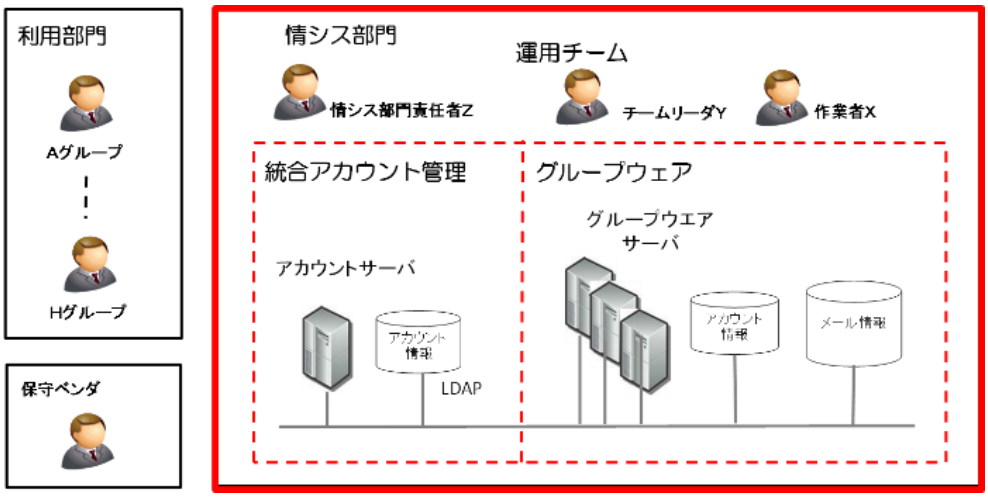


図 2.4-1　分析対象とするシステムの概要

（赤枠内、情シス部門内が対象システム範囲。利用部門、保守ベンダーは対象外）

(2)発生した事故：(損失につながるイベント)

グループウェアサービスへの誤操作により、多数のユーザーのデータ（送受信メール、スケジュール、アドレス帳等）が消失してしまい、復旧するのに 2 日間を費やした。

事故の経緯：

- 運用作業者が、アカウントサーバーの統合アカウント管理ツールを使って、新規ユーザー（50名分）のユーザー登録作業を実施したところ、ユーザーの設定が誤っていることに気づいた（図 2.4-2 ①）。
- 再登録するために統合アカウント管理で登録したユーザーを削除しようとしたが、手順書にその手順が明確化されておらず、また運用の訓練を十分にうけていなかったため削除できなかった。そこで先輩がやっているのをみたことがあるため、直接グループウェアサーバー上で登録したユーザー（50 名分）を削除しようとした（図 2.4-2 ②）。ところが、誤って“全ユーザー削除”を実行してしまった（図 2.4-2 ③）。

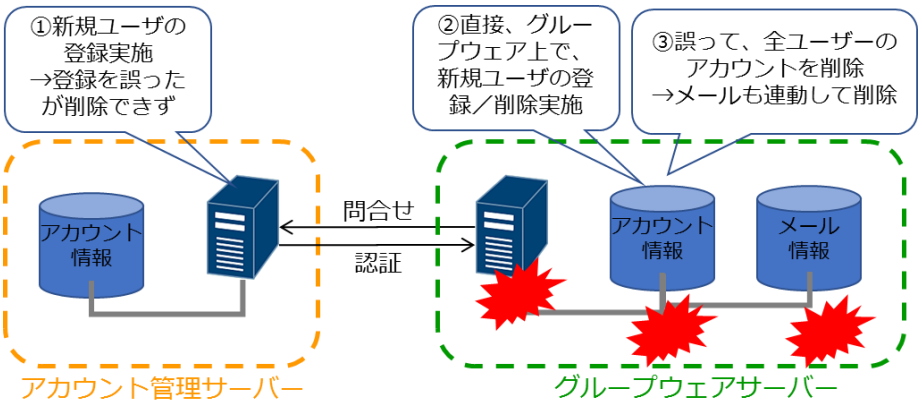


図 2.4-2　発生した事故

2.4.3. CAST による IT システム運用事故の分析

(1) 一般的な CAST 分析

本事例への分析を行う前に、一般的な CAST 分析について、説明する。

Engineering a safer world[Leveson2012]によると、CAST の分析は、後述の CAST 1 から CAST 9 までの手順になる。この分析手順は必ずしも一つが完了してから次のステップへというように逐次に実施されることを意味するものではなく、適宜統合的に実施可能である。

【一般的な CAST 分析手順】を図 2.4-3 に、**【コンポーネントごとの記述事項】**を図 2.4-4 に示す。CAST 分析手順の STPA との大きな違いは、事故のおきた下位レベル（物理レベル）から上位レベル（システムレベル）へと分析対象を拡げ、コントロールストラクチャーの構築をしていく点である。

【一般的な CAST 分析手順】

CAST 1. 損失に関連するシステムとハザードを明らかにする

CAST 2. ハザードに関連したシステムの安全制約やシステム要求を明らかにする

CAST 3. ハザードを制御し安全制約を課すよう整備されている安全コントロールストラクチャーを記述する。*1

CAST 4. 損失につながる近接したイベントを決定する

CAST 5. 損失を物理レベルで分析する *2

CAST 6. 安全コントロールストラクチャーの上位レベルに移り、如何にして、そして何故、より上位のレベルが現在のレベルにおける不適切な制御を許したかもしくは寄与したかを決定する

CAST 7. 損失に関与した共同作業、コミュニケーションの寄与者すべてを調査する

CAST 8. 損失に関連するシステムと安全コントロールストラクチャーの時間経過による動的な特性や変化、および安全コントロールストラクチャーの長期間での弱化を正確に定める

CAST 9. 改善勧告を出す

*1) これはコントロールとフィードバックの実行と同様に各コンポーネントの構造上の責任と権限を含む。このステップは以降のステップと並行して実施できる。

*2) このステップは以下を含む。

・発生した事象に対する次のものの寄与を識別 : 物理的、運用的な操作、物理的な障害、機能が損なわれた相互作用、コミュニケーション、共同作業の欠陥、処理されなかった外乱

・損失を防止するさいに何故、物理的なコントロールが効果的でなかったかを定義すること

図 2.4-3 一般的な CAST 分析手順

【コンポーネントごとの記述事項】

安全要求と制約

発生した非安全なコントロールアクション(UCA)： [コントロール]

意思決定がされた状況（コンテキスト)特定：[前提]

・責任と権限

・環境や行為形成の要素

・機能が損なわれた相互作用、故障、ミスコントロールアクションをひきおこす欠陥のある決定

プロセスモデル（メンタルモデル）の不備：[欠陥のあるコントロールアクションと機能が損なわれた相互作用の理由**]** *1

・制御アルゴリズムの欠陥

*1) コントローラーには制御するコンポーネントが認識するシステムや外部環境の状態を表すプロセスモデルが含まれており、特に人間が行うプロセスモデルはメンタルモデルと呼ばれている。

図 2.4-4　コンポーネントごとの記述事項

(2) CAST による IT システム運用事故分析

CAST は上記 9 つの手順であるが、分かりやすく今回の事例に適用するため、5 つのステップに要約して説明する。

【ステップ１（CAST 1 から CAST4)】

本ステップでは、ハザード、安全制約、想定していた CS 図の明確化という STAMP に共通の手順（**CAST 1-3）**を実施し、さらに発生した事故の経緯（CAST 4）を明らかにしている。システムレベルにおけるアクシデント、損失に関連するハザード、システムの安全制約は以下の通りであり、当初の情シス部門や運用チームが想定していた CS 図を図 2.4-5 に示す。損失につながる近接したイベントは発生した事故として前項に既述したとおりである。ちなみに、現場の運用作業者レベルではグループウェアサーバーへの直接アクセスとデータ変更が状況に応じて使われており、この CS 図では欠けていることにも留意が必要である。この点は、次の分析ステップで詳述する。

アクシデント：（全ユーザー情報が消失し）業務が停滞したこと
損失に関連するハザード：グループウェアサーバーのユーザー情報が欠如した状態
システムの安全制約：グループウェアサーバーのユーザー情報が欠如しないこと

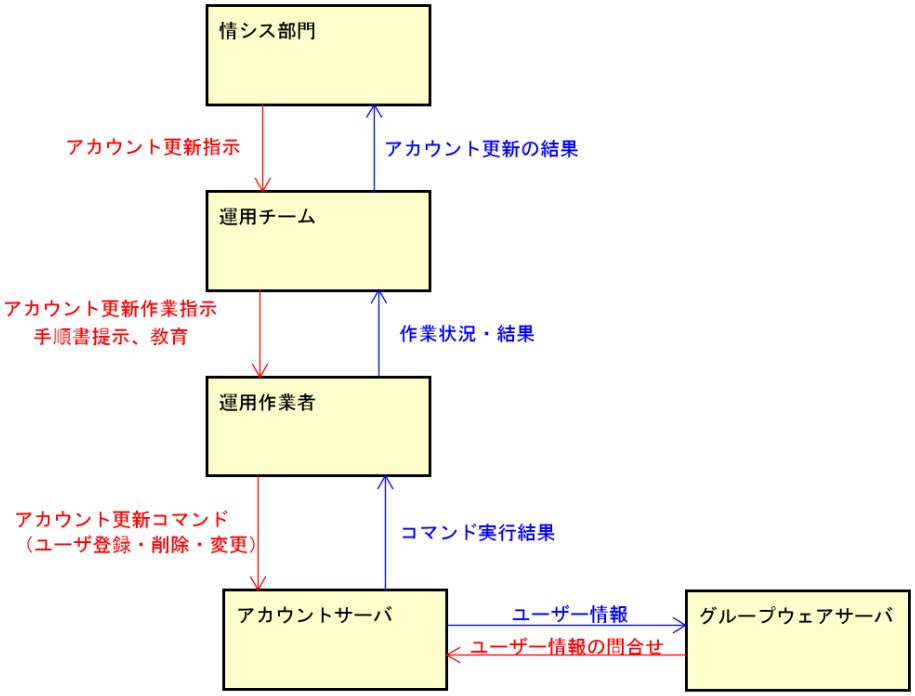


図 2.4-5　想定していた CS 図

【ステップ 2（CAST 5)】

CAST は事後分析であることから、事故が直接的に起こった箇所を特定して、そこから分析を開始することが可能であり、順次分析対象をシステム全体に拡げていくことが特徴である。

そこでまず「**ハザードに直接関わるコンポーネントは何か？**」から考える。ユーザー情報を消失したのはグループウェアサーバーであるため、コンポーネントは**グループウェアサーバー**となる。「**そのコンポーネントに非安全なコントロールアクション（UCA）はあったか？**」と考えると全てのユーザー情報を削除するコマンドが入力されていたことがわかる。本事例では、コンポーネントに影響した人や組織など他のコンポーネントにひそむ根本原因に着目しているため、グループウェアサーバーの分析は故障の有無と UCA の識別のみとしている。

次に、グループウェアサーバーと運用作業者の制御構造をコントロールストラクチャーとして定義する（図 2.4-6)。本事例では運用作業者がグループウェアサーバーに全削除コマンドを実行したことに着目して、これを分析対象に絞り、コンポーネントの記述事項を記載した。図 2.4-6 では運用作業者からインタビューや分析した結果をベースに、メンタルモデルの不備や意思決定がされた状況特定を行っている。STPA とやや異なるように見えるのは、このコンポーネントの UCA 分析の中で同時に UCA を引き起こした背景要因が分析されていることであるが、これは STPA のハザード誘発シナリオと本質的には同じ手順である。

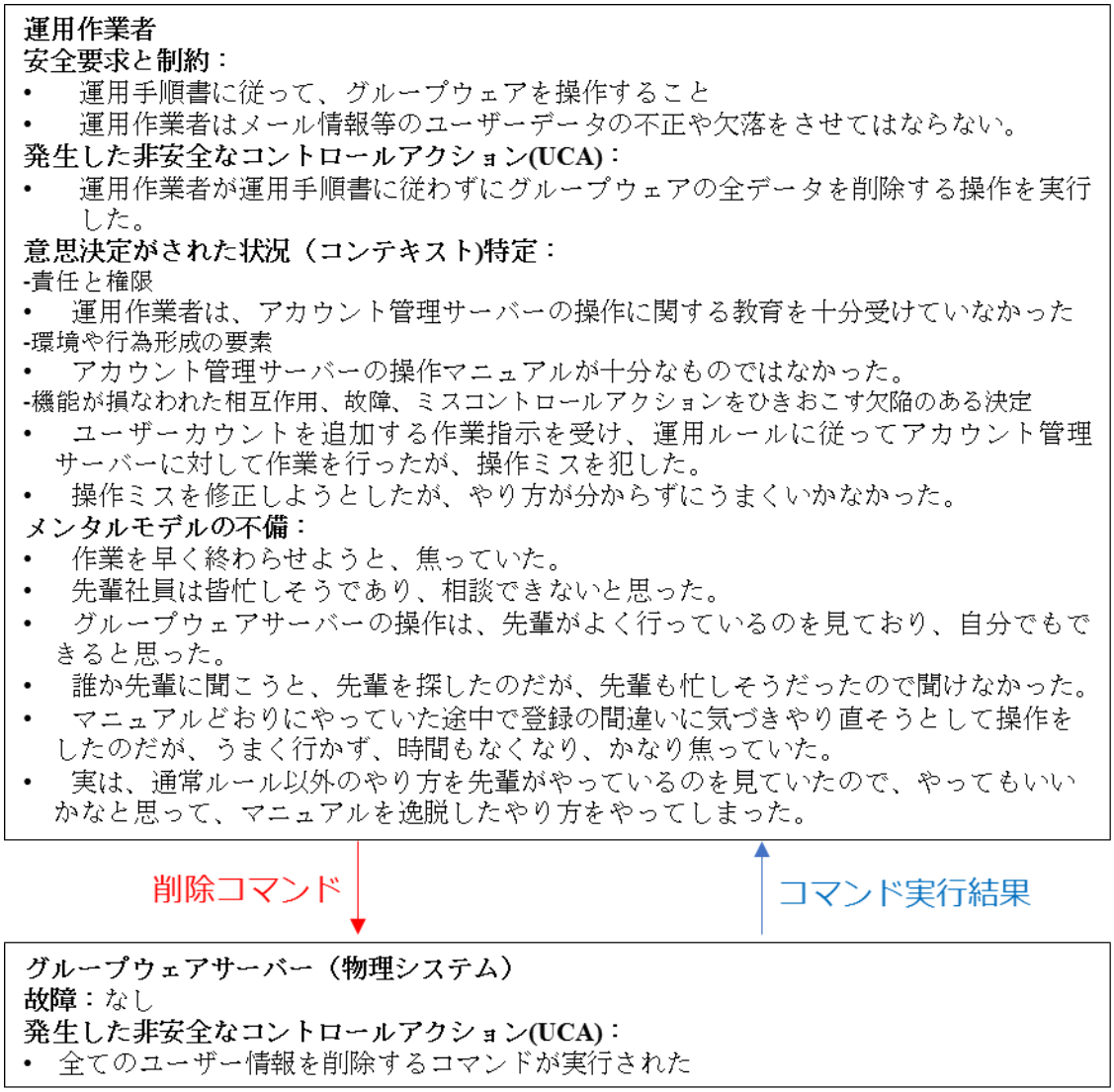


図 2.4-6　事故時のグループウェアサーバーと運用作業者の CS 図とコンポーネント詳細

【ステップ 3（CAST 6)】

ここで分析を止めると、「運用作業者の操作ミス」を事故原因とすることになり、運用作業者を悪者にして事故分析は終わってしまう。重要なのは、運用作業者が何故、このような UCA をおこしたのか、その根本原因を探ることである。そこで UCA の原因となるコンポーネントの相互作用を説明できるように、事故がおこった直接原因になる物理コンポーネントとそのコントローラー（図 2.4-6）から周辺のコンポーネント（図 2.4-7）に分析対象を拡大する。

例えば、運用作業者に作業指示をだした上位コンポーネントである運用チームについては、以下のように記述していく。運用チームの安全要求と制約の 1 つは「運用作業者から作業報告、相談、質問を受け、運用作業者が適切な行動をとれるように指導する。」である。これに対して、「運用作業者への作業報告、相談、質問を受け、運用作業者が適切な行動をとれるような指導を行わなかった。」という UCA が発生した。この意思決定がされた状況（コンテキスト）特定は「この時は、運用作業が目白押しだったため、運用チームの他のメンバーは忙しかった。」であった。UCA をおこしたメンタルモデルの不備は「運用作業者から相談・質問がないため、問題なく作業が進

捗していると思った。」になる。

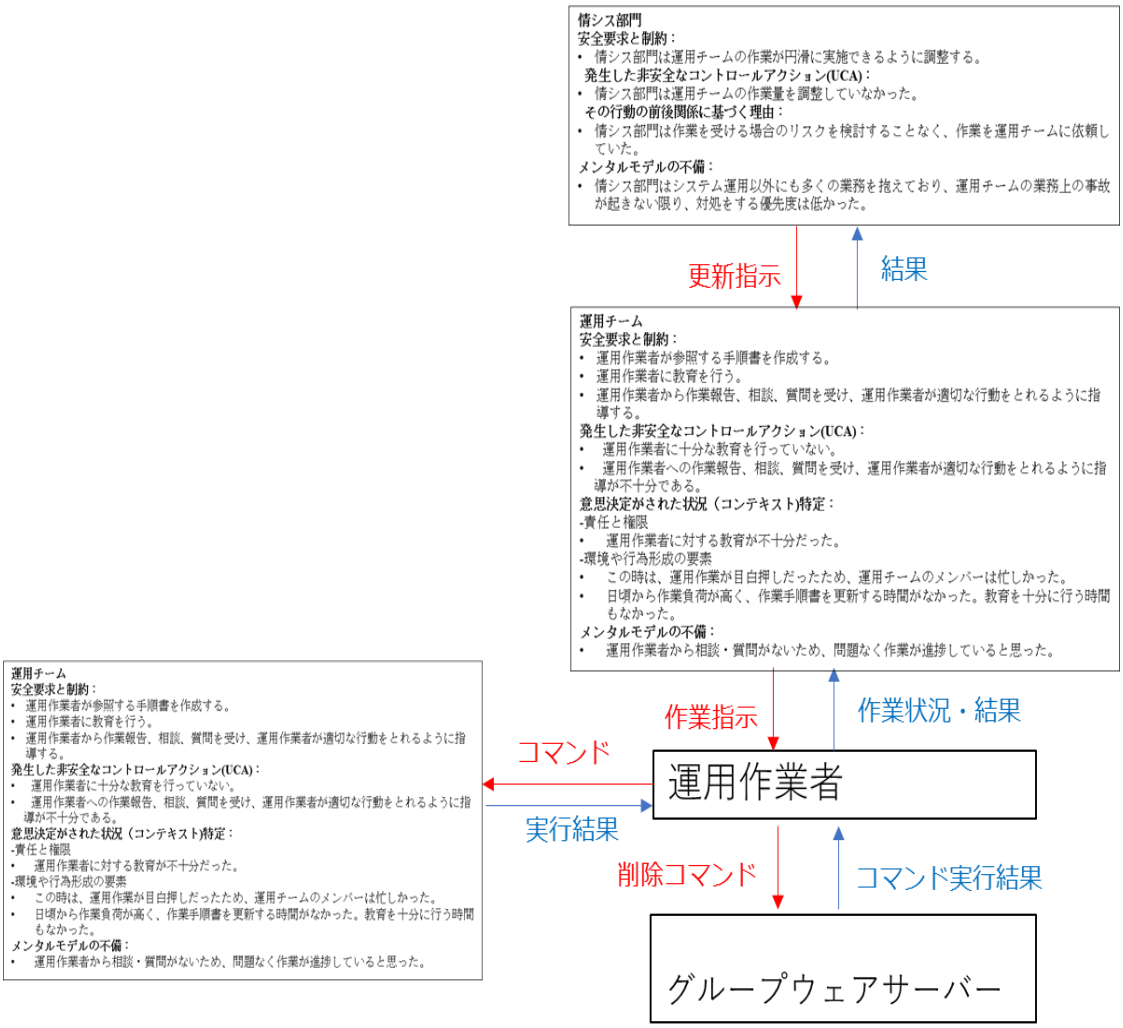


図 2.4-7　事故時のシステム全体の CS 図とそのコンポーネント詳細

【ステップ 4（CAST 7-9)】

ここでは、既に存在するシステムに対して、CAST7 と 8 の全体俯瞰による統合的確認を行った上で CAST9 の改善勧告を出す。

まず、損失を招いた全体的な調整ややり取りを相互の作用として、全体を俯瞰して確認をする。具体的には各コンポーネント（登場人物）にどのような責務があり、この責務に応じて、どのようなコントロールアクションがだされ、フィードバックが出されたのかについて、確認した。さらにアルゴリズムとプロセスモデルを吟味の上、各コンポーネント間（運用作業者や管理者等）のコミュニケーションの不備をリストアップした（表 2.4-1）。これは、各コンポ―ネントの①安全要求と制約、②UCA、③意思決定された状況、④メンタルモデルの不備に対して背景要因を全体俯瞰の上での統合的確認である。

表 2.4-1　全体俯瞰による確認

コンポーネント	コントロールアクション	アルゴリズム・プロセスモデル	不備な点
運用作業者	削除コマンド	・グループウェアサーバーの操作は、先輩がよく行っているのを見ており、自分でもできると思った。	理解不十分な状態で操作をし、全ユーザデータを削除した
運用作業者	削除コマンド	・操作ミスを修正しようとしたが、やり方が分からずにうまくいかなかった。	操作ミスを防ぎ、正しいやり方に誘導する仕組みがなく、教育も不十分
運用作業者	削除コマンド	・作業を早く終わらせようと、焦っていた。	十分な確認をする時間的、精神的余裕がなく、誤操作をおこした
運用作業者	削除コマンド	・先輩社員は皆忙しそうであり、相談できないと思った。	熟練者が忙しすぎて不明点を確認できない
アカウントサーバ	削除コマンド	・運用作業者はユーザー登録作業や修正作業方法を十分には理解していなかった。	過去にあった障害から、類推した対応を実施することができない
運用チーム	作業指示	・教育を十分に行う時間もなかった。	当該運用作業者は新人で熟練していなかった
運用チーム	作業指示	・日頃から作業負荷が高く、作業手順書を更新する時間がなかった。	提供している手順書に不備がある
運用チーム	作業指示	・運用作業者から相談・質問がないため、問題なく作業が進捗していると思った。	実施前に作業を確認する仕組みがない
運用チーム	作業指示	・運用作業者から相談・質問がないため、問題なく作業が進捗していると思った。	正規の手順を外れた作業を抑止する仕組みがない
情シス部門	更新指示	・情シス部門はシステム運用以外にも多くの業務を抱えており、運用チームの業務上の事故が起きない限り、対処をする優先度は低かった。	作業指示をするときに、指示を受けた側にどのような課題が生じるのかを把握していない

さらに時間経過による安全コントロールストラクチャーの弱点を明確化した。その理由は、当初の段階では安全を保つ仕組みが成り立っていても、時間が経つにつれて構成するコンポーネント自体やその相互作用に変化が生じ、安全制御が成り立たなくなることが多いからである。

本事例の場合、運用作業者は届け出による上位の許可がなければ、グループウェアサーバーにアクセスはできないことになっていた。しかし、運用チームの先輩たちが、日常的にグループウェアサーバーにアクセスしていたのを見ており、十分な知識もないままにアクセスし、事故を起こした。本来の想定モデルでは、運用作業者からグループウェアサーバーへのコントロールアクションは想定されていなかった。当初より運用の作業量が増えたことという経時的な変化の中で、運用作業者が許可がなくても実施できる状況を生んでいた。

さらに時間経過による安全コントロールストラクチャーの弱点を、当初想定していた CS 図（図 2.4-5）と事故発生時の CS 図（図 2.4-8）の違いとして明らかにすることができる。この 2 つの CS 図は管理サイドの想定と、現場の運用作業者の行動の違いを示すものになる。

この 2 つを比べると、実施の事故でおきたことを示す CS 図には、運用作業者がグループウェアサーバーへ削除コマンドを送信するという UCA が存在する。この UCA は当初想定していた CS 図には存在せず、管理サイドでは想定されていなかったことがわかる。このような両者の想定と

行動の違いが事故の背景にあることを 2 つの CS 図で明確化することができた。

このようなシステムレベルの図示によるわかりやすい可視化が STAMP によるモデリングの効果の 1 つである。

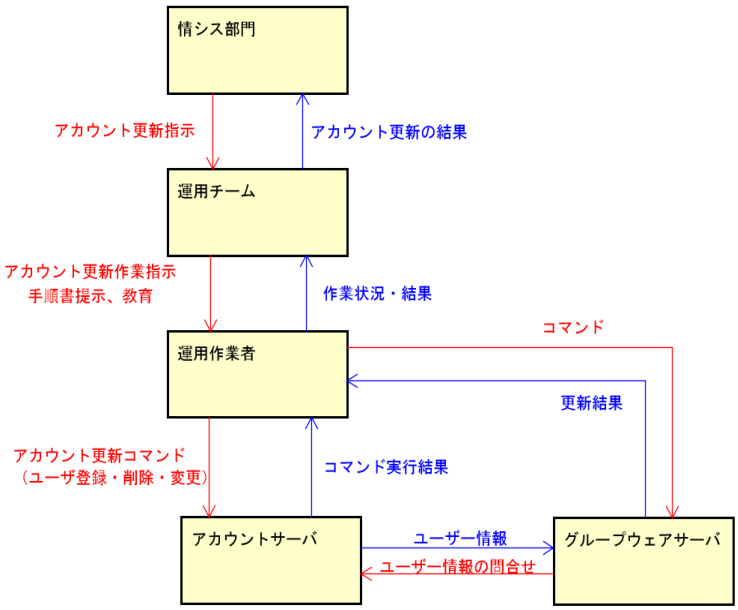


図 2.4-8 実際の事故でおきたこと

これらの不備な点と時間経過による安全コントロールストラクチャーの弱点を解決するため、表 2.4-2 の改善勧告が提示できる。

表 2.4-2 改善勧告

改善勧告
・削除ボタンを実行しても、上司の承認が実行されないと削除されないようにグループウェアサーバーのソフトウェアの変更をおこなう ・熟練した作業者にのみコマンド操作の許可をだす
・削除コマンドなど重大な影響を与えるコマンドの送信前に確認メッセージを出す ・運用作業者の知識レベルを向上させる教育を実施する
・本番環境での作業には、事前にテスト環境で実施して問題がないことを確認してからの実施とする ・作業ミスを防ぐため、運用作業は２名体制（実施者、確認者）で行う
・担当者の忙しさを緩和するように、情シス部門からの指示がきても運用チームで調整を図る
・インシデント管理（障害記録の作成）を実施し、チーム内で共有する
・運用作業者の知識レベルを向上させる教育を実施する
・新規ユーザ登録に失敗した場合の対処手順を作業者が理解するようにアカウントサーバー上の登録方法をわかりやすく記載し、手順書にまとめる
・運用チームは「作業実施前に手順書の確認をチーム内でレビューする」ルールを設定する
・運用チームは「手順書と違う状態が発生した時点で作業を止め、上位者へ報告する」ルールを設定する
・担当者の忙しさを緩和するように、情シス部門からの指示がきても運用チームで調整を図る ・情シス部門は、作業を受ける場合は、作業を実施した場合のリスクを分析して判断基準を作成し、その基準をクリアした上で作業実施を決定する

2.4.4. まとめ

①事故のおきた下位レベル（物理レベル）から上位レベル（システムレベル）へと分析対象を拡げていくこと、②既に存在するシステムに対して、全体俯瞰のうえ改善勧告を出すことというCAST の特徴を本事例の分析を通じて、確認できた。

①については、**【ステップ 2（CAST 5)】**で事故の発生したグループウェアサーバーに直接作用した運用作業者の分析を行ってから、**【ステップ 3（CAST 6)】**で運用チーム、情シス部門などの人や組織に分析対象を拡げ、事故の原因分析を実施した。事故発生時の CS 図（図 2.4-8）に基づいた UCA 分析ならびにその UCA の背後要因の指摘は、背景要因、組織要因までをわかり易く可視化でき、人と組織を含めたシステムの安全において将来の改善に役立つ。このステップは、STPA の CS 図と UCA 分析、ハザード誘発シナリオ分析を統合したものになっている。

②については、**【ステップ 3（CAST 7-9)】**で運用作業者、運用チーム、情シス部門などの各コンポーネントを統合的に捉え、不備な点を洗い出すことで全体俯瞰している。

さらに想定していた CS 図と実際に事故発生時の CS 図を比較することで、安全コントロールストラクチャーの弱化を明確化できることを考察した。

このようなシステム思考の特徴をもつ CAST は、事故が発生した箇所だけでなく、周辺のコンポーネントに拡げて原因分析していくことで、システム全体を俯瞰して、隠れた事故の本質的原因や背景要因を探る事故分析手法として有効だと想定される。

3. STAMP/STPA 演習教材

3.1. はじめに

(1) 教材の位置づけ

本章では、STAMP/STPA の理解を深めるための演習用教材を紹介する。初級編と中級編の演習教材は本書の付録としてダウンロードが可能である。各教材の特徴を理解していただき、自組織内の教育や対外的なセミナー等に活用していただくことを想定している。

各教材は、「システム思考による安全分析」の体験ができるように構成されている。すなわち、システム全体の制御構造を俯瞰する CS 図で共通認識を持ち、分析チームのメンバーが持つ様々な背景知識や想像力と、強制発想法に基づく議論によって幅広くハザードシナリオを識別する流れを、体験的に理解できることを目的としている。また、初級編と中級編は、STAMP/STPA 分析を支援するツール STAMP Workbench を用いた演習であり、演習によって STAMP Workbench の基本操作を感覚的に理解することができる。

(2) 教材で用いるツール（STAMP Workbench）

STAMP Workbench は、IPA からオープンソースソフトウェアとして公開しているツールである。STAMP/STPA の手順を誘導する仕組みを備えているため初心者でも迷わず分析を進められる点や、分析の各ステップで必要となる図や表が連動しているため、ステップ間の転記ミスや修正漏れを防ぎ、手間を省力化できる点が特徴である。

(3) 教材一覧

本章で紹介する教材の一覧を表 3.1-1 に示す。

表 3.1-1　本章で紹介する演習教材

初級編		
	受講対象	STAMP 初心者の技術者
	演習の狙い	STPA 手順概要とツール基本操作の修得
	所要時間	90 分
中級編		
	受講対象	STAMP 実適用を検討する技術者
	演習の狙い	具体的なシステムへの適用例を用いて分析の勘所を修得
	所要時間	120～150 分

なお、初級編と中級編のほかに、導入編に相当する演習教材を JASPAR[3]が作成しており、今後

[3] 一般社団法人 JASPAR（Japan Automotive Software Platform and Architecture）。高度化・複雑化する車載電子制御システムのソフトウェアやネットワークの標準化及び共通利用による、開発の効率化と高信頼性確保を目指す業界団体。

JASPAR から公開される予定である。本章ではこの JASPAR の教材についても紹介する。

導入編		
	受講対象	STAMP について知りたい技術者、管理者
	演習の狙い	STAMP とは何かを知る（啓発を含む）
	所要時間	60～120 分

所要時間にはおおよその時間を記した。演習実施の目的や受講者のレベル、受講者の人数、講師補助の人数などに応じて、時間をかけてグループ演習に重きをおく、逆に受講者のスキルに応じて解説を簡略化する、など上記所要時間に拘らずに活用して欲しい。また、教材内容に自組織の開発対象ドメインの話題を含める／置き換えるなどのカスタマイズを行っても良い。

3.2. 演習教材　初級編

■演習の目的と受講対象者

FTA,FMEA,HAZOP など他手法による安全分析に関する知識、経験の有無に関わらず、STAMP 分析（STAMP/STPA）にはじめて取り組もうとする初心者の方、あるいは、座学で STAMP についての知識はあるが、実際に自分が手を動かして STAMP 分析をした経験のない方を対象とする。システムの相互作用に着目した STAMP 分析の手順概要を、実際に自分の手を動かして体験することにより理解してもらうことを目的とする。併せて、STAMP 分析を実施する際、効率的に分析作業を実施するために有効なツールである STAMP Workbench の基本的な操作方法を習得してもらうことも目的とする。

■題材の概要

宇宙の遠い星から地球に来たウルトラマンが、地球の科学特捜隊に協力して、地球の平和を守るために、人類を脅かす怪獣や宇宙人と戦う、という設定が本教材の題材である。

安全分析における分析対象範囲・分析対象システムは、分析の目的やステークホルダーとの合意によって決まるものである。システム俯瞰と言っても、その対象システムとはどこまでなのか、システム境界はどこなのかが重要であり、本題材はそれを考える際のヒントを与えている。本演習の Step-0 では、まず分析の目的を明確にして、図 3.2-1 のように CS 図を描きながら分析範囲（分析対象システムのシステム境界）を明確化している。敢えてシステム境界が自明でない題材を用いて、システム境界を決めていく過程を体験してもらう。

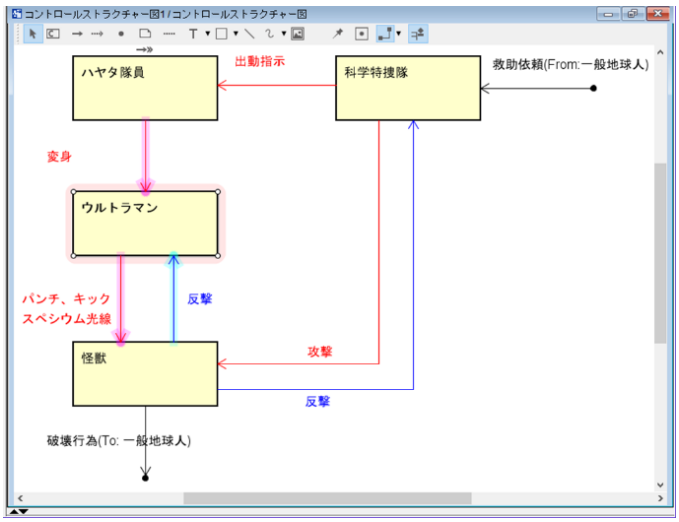


図 3.2-1　システムの安全構造を表す CS 図

■演習によって得られる効果

座学で得た知識を、演習で得た自分自身の経験と結びつけ、実践的なスキルへと昇華することができる。

また、STAMP Workbench が STAMP/STPA の手順を誘導するようになっていることを知り、目的に合ったツール活用の有用性を理解できる。

■本教材の特徴

STAMP 分析の具体的な手順を理解するとともに、分析を効率的に行うために有効なツール操作に慣れることができる。

本演習では、STAMP の本質を理解するところまでは狙っていないが、短時間の演習で分析手順を正しく理解することができる。STAMP の本質を理解して、真に有効な分析を目指すには、

本演習を実施した後、別途中級者向け演習を実施することを推奨する。

本演習の題材は、STAMP 分析の手順の理解に専念できる題材を選んだ。本演習の題材を既存手法で分析しようとしたら、「コンポーネントの故障？」とか「故障確率？」など、どうやって分析して良いか戸惑うことであろう。一方、STAMP では自然に分析ができることから、システムを俯瞰し、システムにおける相互作用に着目して分析を行う STAMP の特徴を感じることができる。

STAMP は“強制発想手法”とも言われ、分析を進める過程で新たな気付きを得られる機会が多い。本演習では、分析途中で新たな気付きを得て、前のステップに立ち戻り、分析結果の質を高めていくプロセスも経験できる。

■演習の進め方と注意点

本教材は、短時間で演習を実施できるようにしている。受講者が分析の経緯や結果の全てを入力していたのでは、分析例にならってツールに入力するだけで多くの時間がかかってしまい、受講者に考える時間を与えられなくなってしまいかねない。そこで、本教材では、各 Step で考える対象を絞り込み、入力時間よりも考える時間を多く与えられるように工夫した。具体的には、各 Step までの分析例を入力済みのプロジェクトファイルを用意している。それらのプロジェクトファイルは対象 Step までの全てを入力しているのではなく、部分的に未入力としている。受講者は、その未入力部分に絞り込んで実際の分析を行うことになる。

各 Step までの入力済みプロジェクトファイルを用いると、受講者全員が常に同じ進行状態で次の Step の演習を実施できる、というメリットがある。そして、講師にとっても全受講者に共通の解説を行えば良いというメリットがある。

図 3.2-2 は、プロジェクトファイルの一覧である。演習の各 Step に入るところで、対応するプロジェクトファイルを受講者全員に読み込んでもらう、という進め方を推奨する。

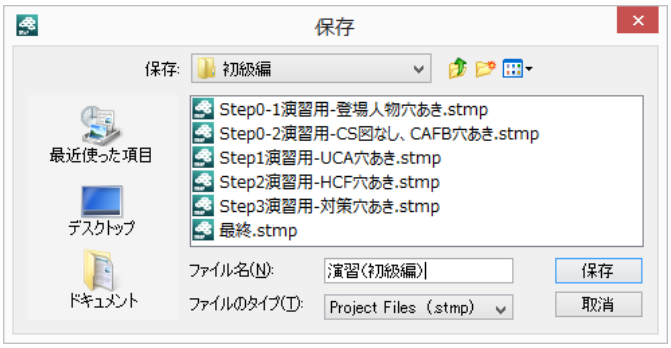


図 3.2-2　プロジェクトファイル一覧

本教材は短時間で実施できるように Step 毎の上記プロジェクトファイルを用意したが、時間をかけてでも一通り全部分析を経験してもらう方針であれば、利用するプロジェクトファイルを減らすなどしても良いし、プロジェクトファイル内の穴あき箇所を増やしても良い。

3.3. 演習教材　中級編

■演習の目的と受講対象者

STAMP 分析の手順を概要レベルでは理解しているが、実際の分析経験が少ない人を対象として、実践的な題材の分析を通じて、STAMP の本質（システム思考のアプローチとその狙い）の理解と、真に有効な分析実施ができるようになることを目的とする。

■題材の概要

施設等の自動車の入退場において、入場許可のある自動車のみを入場させるための「アクセス・コントロール・ゲート」の制御システムを題材とする。図 3.3-1 に示すように、自動車の通行を物理的にブロックする「ゲート」とその開閉を制御する「ゲートコントローラ」があり、ゲートコントローラは、自動車を検知するセンサーと、入場許可がある場合に押される「承認ボタン」の情報をもとにゲートを開閉するシステムである。

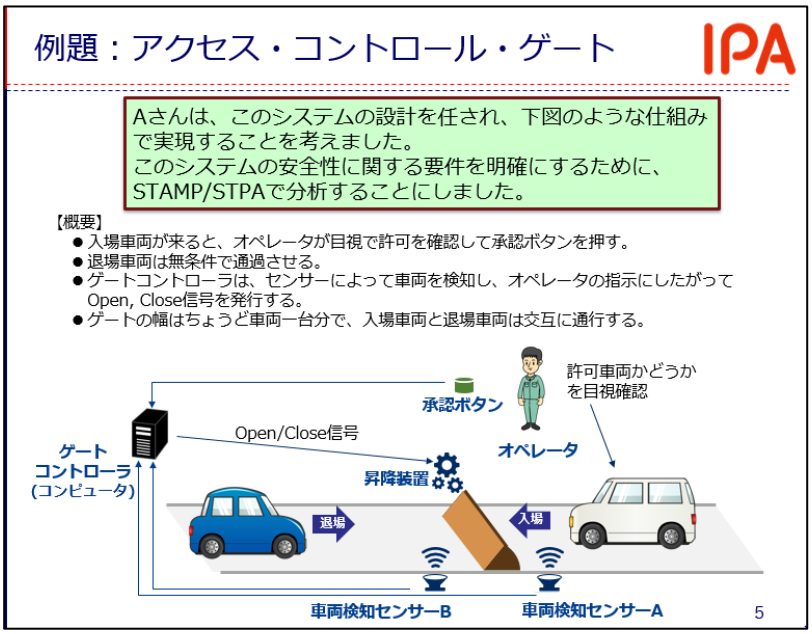


図 3.3-1　分析対象とするシステムの概要

■演習によって得られる効果

分析のステップごとに課題が設定されており、グループで課題について議論しながら分析を進める形式を想定している。多様な知識を持つ分析者が CS 図によって認識を共有し、その上で意見を出し合うことにより幅広いハザード要因を得るという STAMP 分析の効果を体験できる。

また、STAMP Workbench の操作方法について、マニュアルを参照しての理解ではなく、実際に分析を行う流れにそって理解を深めることができる。

■本教材の特徴

自動車の入退場ゲートという比較的多くの人が想像しやすいシステムを題材としている。システムの構成要素も、車両検知センサーや昇降装置など、常識的に理解しやすいものに限定しており、題材理解の難しさが演習の妨げにならないように工夫している。

また、入場と退場が非同期に起こり、その相互干渉が起こり得る設定にしている。これによ

り、シンプルなシステムでありながら考えるべき状況空間が比較的広くなるため、UCA の識別やその要因分析の際に、CS 図で全体俯瞰し、幅広く相互作用を考えるというシステム思考の特徴を体験しやすいことを意図している。

演習を進める上での工夫としては、STPA の各ステップの演習を行う前にそのステップの意味を説明するページがあり、参加者はそのステップの目的を明確にしながら分析を進められるようにしている。

■演習の進め方と注意点

本教材は、グループディスカッションの時間を含め、2 時間から 2 時間半程度で実施することを想定している。演習の流れは以下の通りである。

- 例題の説明
- STPA Step-0 – 準備 1（アクシデント、ハザード、安全制約の識別）
- STPA Step-0 – 準備 2（CS 図の作成）
- STPA Step-1 （UCA の識別）
- STPA Step-2 （UCA の発生要因の分析）

STPA の各ステップは、次のような順序で進めることを想定している。

- その分析ステップの目的や、必要となる概念等の説明
- 課題の説明
- グループ演習（グループでのディスカッション）
- グループ演習の結果の紹介とそれに対するコメント（Q&A）
- 回答例の説明
- STAMP Workbench への入力方法の説明
- STAMP Workbench への入力の実習

演習を進める上での注意点としては、以下のような点がある。

グループ演習の結果を紹介してもらうことによって、受講者が誤解している点が明らかになることが多いため、なるべく分析の各ステップで多くのグループから結果を紹介してもらい、それをもとに質疑、解説を行うとより深い理解を導くことができる。

また、ツールの入力操作には時間がかかる場合が多いため、あらかじめ用意したプロジェクトファイルを利用するとよい。本教材では、各ステップについて、そのステップを終了した状態の STAMP Workbench のプロジェクトファイルが用意されている。教材資料のなかで、プロジェクトファイルが用意されているページにファイル名が書かれている。入力作業が時間内に完了できない受講者に対しては、予め配布したプロジェクトファイルを読み込むことによって、次のステップに進むように促すとよい。

3.4. JASPAR が作成した演習教材　導入編

前述のとおり、導入編に相当する演習教材が JASPAR にて作成されている。ここでは、初級編、中級編と同様にその概要を示す。

■演習の目的と受講対象者

今後の製品開発におけるリスク分析に関心を持っている技術者や管理者を対象として、STAMP とは何かを知ってもらうことを目的とする。STAMP の基本的な考え方を解説するとともに主要な STAMP 分析の方法の概略を紹介する。

■演習題材の概要

プレゼンテーションにおいてノートパソコンが起動できないという状況を題材とする。

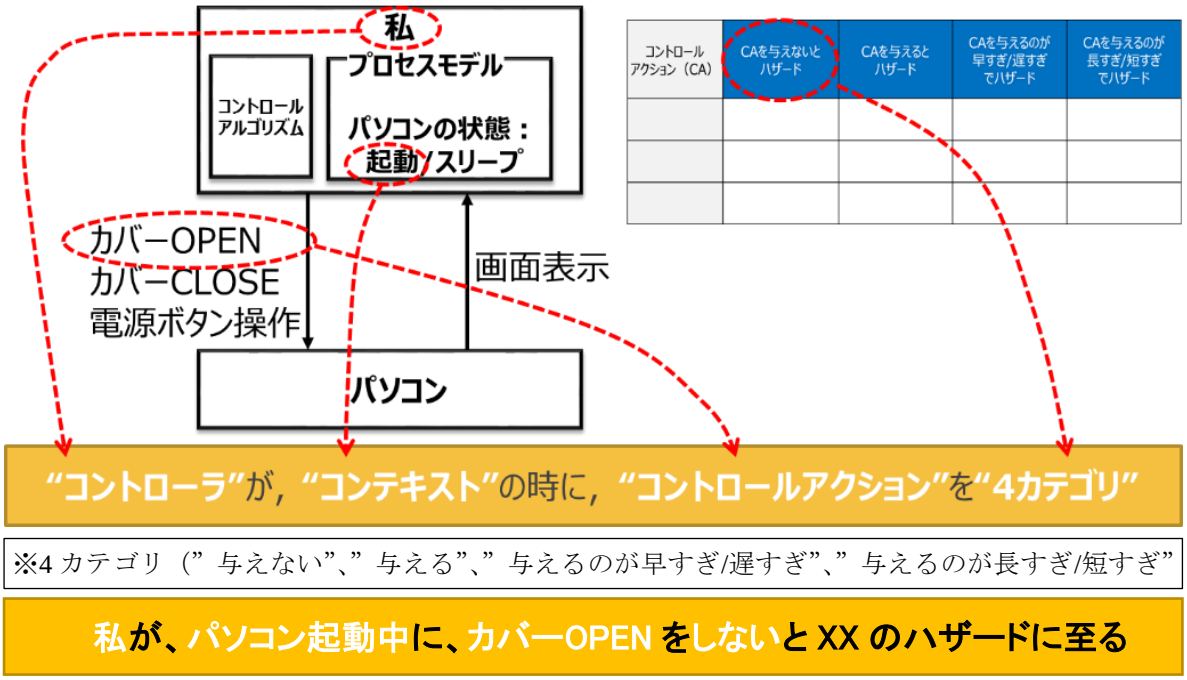


図 3.4-1　UCA 識別時の UCA 構文作成方法

■演習によって得られる効果

身近な事例“私とパソコン”による演習を通して STAMP に興味をもってもらうことを狙っている。

■本教材の特徴

STAMP とは何かを知ることに主眼を置いている。

■本教材について

本教材は、JASPAR が提供している教材の一つを掲載させていただいたものである。JASPAR の教材は、「知る」、「わかる」、「できる」の 3 部で構成され、それぞれ導入編、基礎編、応用編

となっている。本教材はそのうち導入編の教材である。

■演習の進め方と注意点

演習は、以下のような流れで進めることを想定している。

⑴ STAMP の概要説明

⑵ STPA 分析の体験

・1st step ～ 4th step のそれぞれのステップの概要説明と演習

なお、本教材では 4th Step の損失シナリオの識別までを対象としており、対策案の検討は対象外としている。

演習を進める上での注意点としては以下の点がある。

各 Step で分析中に前 Step の分析に追加/修正などがあることに気付く場合がある。その場合は、適切な Step へ戻る“イタレーション”による再分析を推奨すべきである。例えば、3rd Step の UCA 識別中に新たな“アクシデント”や既知のアクシデントに対する“ハザード”を識別される場合、1st Step に新たに識別した“アクシデント”と“ハザード”を更新し、2nd Step、3rd Step を再分析し更新する。

4. システム思考によるこれからの安全・レジリエントなセキュリティ

4.1. 概要

セキュリティにおける STAMP 関連の研究が盛んに行われるようになってきた。

一例を示すならば、2018 年 12 月に実施された STAMP ワークショップにおいては、一般講演の四分の一がセキュリティ関連のものであり、きわめて高い関心があることが分かる。しかし、これらの先進的な研究においても、大きな課題が識別されている。すなわち、「複数同時攻撃」や、「最もあり得なさそうな想定外の侵入」等、厄介な問題に対して、有効な対策は依然として困難という問題である。

一方、セキュリティに関する全く異なるアプローチが、レジリエンス・エンジニアリングの分野から示され、期待が高まっている。従来の手法が、主に強固な防御壁を構築することを主眼としてきたのに対して、レジリエンス・エンジニアリングは、より素早く変化を見抜き、自らも柔軟に変化し、経験から学習し、未来を予測するという、自然界の動物の有するレジリエンス能力を高めることによってセキュリティ能力を高められるという仮説に基づく考え方である。

本論では、まず、STAMP ワークショップにおいて発表された先進セキュリティ対策が有する課題を示した後、これらの課題に対して、レジリエンス・エンジニアリングがどのようなソリューションを提供しうるのかについて論じる。

4.2. セキュリティの解決困難な課題

ここでは、STAMP ワークショップにおいて発表されたセキュリティ関連の先進的な研究について概説する。それぞれのアプローチの共通の課題は、「想定外の脅威への対応」である。

4.2.1. 多重防護設計

近藤等は、「ICS におけるセーフティとセキュリティのための STAMP モデル適用」の発表の中で、多重防護がシステムに組み込まれていることを直接評価可能となる CSD(Control Structure Diagram)の表記法を提案した[近藤 2018]。これにより、同時多重に発生しうるサイバー攻撃に対しても、多重性を維持できる安全対策の設計が可能となる。STAMP/STPA を利用して、多重防護の有効性を分析することは有効な手段であり、ネットワーク構成情報と組み合わせることによって、脆弱な単一故障点などの識別に寄与することができる。

しかし、多重防護システムそのものは、複数同時攻撃によって同時に無効化される可能性を常にはらんでいる。つまり、常に想定外の複数同時攻撃の脅威にさらされ続けている。言い換えれば、既知の脅威には有効であるが、未知の脅威に対しては、常に想定外が発生しうる。

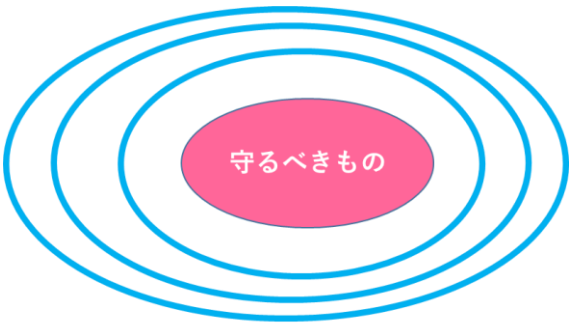


図 4.2-1　多重防護設計

4.2.2. 確率論的最適化設計

早川は、「STPA を用いたリスク分析・対策選定手法の提案と適用」の発表の中で、STPA によって導き出された複数のセキュリティ対策（安全制約）のうち、アクシデントの発生確率を元に最も有効な対策を選定する手法を提案した[早川 2018]。また、林等は、「STAMP/STPA を用いたリスクコミュニケーションツール」の発表の中で、リスクの準定量分析を行うツール MRC4IoT を紹介した[林 2018]。このツールによって、STAMP/STPA によって抽出された脅威を設計の最適化に利用する際に、膨大な脅威シナリオの中から、最も有効な設計を導き出すことが可能となる。

こうした確率論的最適化設計により、既知の脅威に対する最も有効な対策を効率的に実施することが可能となるため、システムの安全度は飛躍的に向上することが期待される。しかし、前項の多重防護設計と同様、悪意を持った攻撃者は、想定外の経路・方法からの攻撃を試みる可能性があり、確率論的には最も低い侵入経路から攻撃が加えられるリスクは無視できるわけではない。すなわち、ここでも、既知の脅威には有効であるが、未知の脅威に対しては、常に想定外が発生しうる。

「既知の脅威には有効であるが、未知の脅威に対しては、常に想定外が発生しうる」という問題こそが、防御壁構築型セキュリティを指向する際の問題となる。設計手法の工夫によって防御壁を強靭にしようとすれば、必ずコストと効率のトレードオフが必要となる。防御壁というもの自体は、システムの性能を高める機能ではなく、むしろコスト要因となるからである。トレードオフによって切り捨てられるセキュリティホール（図 4.2-2 の右側半分）こそが、悪意を持つ攻撃者の狙いどころとなりうる。

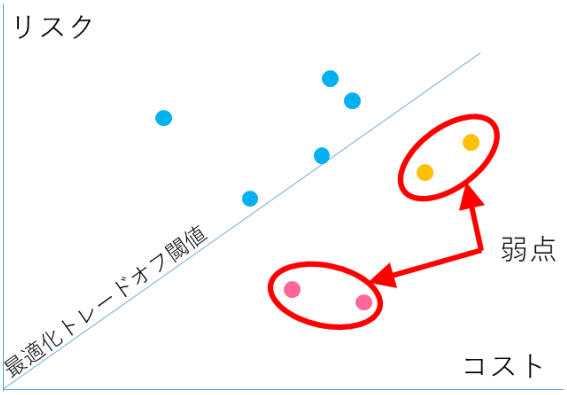


図 4.2-2　最適化設計と弱点

4.3. レジリエンス・エンジニアリング

強固な防護壁構築型セキュリティに対するアンチテーゼとして、レジリエンス・エンジニアリングは、ダイナミックに変容できる柔軟性こそがセキュリティを高めると主張した[Hollnagel2018]。

ホルナゲルは、Safety と Security の最大の相違点は、各々が扱わなければならない脅威の「種類」であるとする。すなわち、Safety が扱うのは、Regular Threat（部品の故障、制御の破綻など、予測可能な脅威）であるのに対して、Security が扱うのは Irregular Threat（想定外の経路からの侵入等、予測不可能な脅威）であるとしている。既知の脅威に対しては、強固な防御壁は有効であり、強固なシステムの構造によって、既知の脅威からシステムを保護することが可能であるが、未知の予測不可能な脅威に対して前もってシステムの防御を用意することは、「ほぼ不可能」としている。

ホルナゲルは、予測不可能な脅威に対抗する唯一の手段は、防御ではなく、よりアクティブな能力であると主張する。すなわち、レジリエンス・エンジニアリングで定義される、「レジリエン

スの指標となる 4 つの機能：監視する機能、反応する機能、学習する機能、予測する機能」であるとする。特筆すべきは、そこに「守る」という機能が入っていないことである。危険の予兆を監視し、それに迅速に反応できる能力、過去の傾向から学習し、そこから未知の脅威を予測する能力は、防御壁型の「守るセキュリティ」に対して、「攻めるセキュリティ」ということができる。原理的にどのような強靭な防御壁も完璧ではあり得ない。守ってばかりでは、セキュリティ脅威との戦いに勝利することは難しい。勝利を収めたいのであれば、守りと攻めのバランスを目指す必要がある。「監視機能」「反応機能」「学習機能」「予測機能」を、攻めるための「セキュリティ機能」と位置づけるのが、ホルナゲルの主張の骨子である。

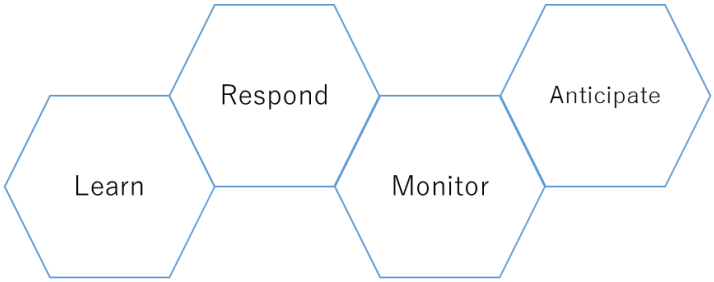


図 4.3-1　レジリエンス機能

では、実際に、セキュリティ分析・設計を如何に行えるであろうか？次項に、その具体的な方法を、レジリエンス・エンジニアリング解析手法である FRAM（Functional Resonance Analysis Method：機能共鳴分析手法）を使って示す。

4.4. FRAM によるセキュリティ分析・設計

FRAM は、分析対象システムの「機能」「やりたいこと」を一つの六角形で表し、その「機能」が実行されるために必要な各種条件（5 つの要素：トリガー、前提条件、時間制約、資源、制御パラメータ）と、出力情報を定義する。一つの機能から出力される情報は他の機能の入力となるが、それは、5 つの入力要素のうちのどれかになる（図 4.4-1）。分析対象システムの中の最も重要そうな機能に関して、6 つの要素を識別し、この接続先の機能に対しても、順次 6 つの要素を識別してゆくという作業を繰り返すと、分析対象の全貌を表現するネットワーク図が完成する。

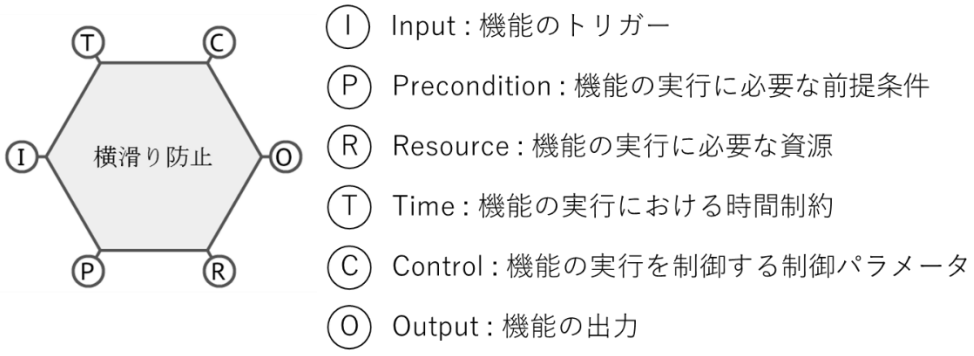


図 4.4-1　FRAM モデルの機能要素

図 4.4-2 は、近年の自動車に搭載されている横滑り防止機能(Electric Stability Control)の機能ネットワークである[ESC2019]。ESC は、想定外の車輪速度を検出すると、ホイールの横滑りが発生していると判断し、現在のステアリング角、アクセル開度、ブレーキ圧等の入力値を使って計算した最適制御を、ステアリング角、アクセル開度、ブレーキ圧等に対して行う。

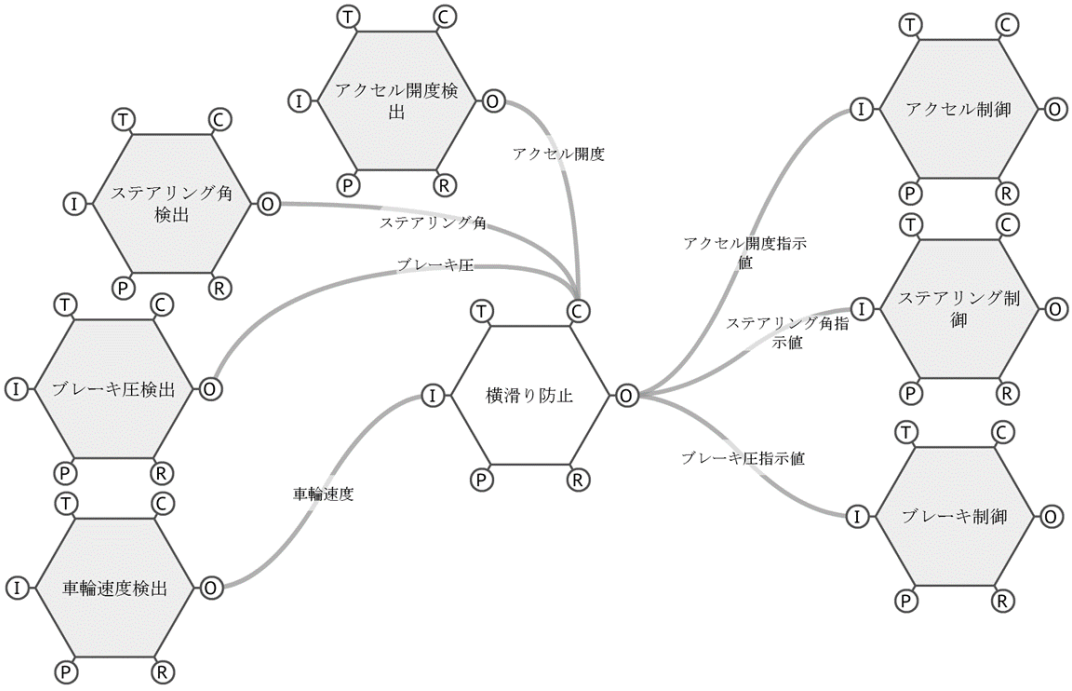


図 4.4-2　FRAM モデル

4.4.1. 問題点

図 4.4-2 のような単純な設計を行ったとすると、セキュリティ上の脆弱性が問題となる。図 4.4-2 のシステムはステアリング角、アクセル開度、ブレーキ圧といった、車両の動作を決定づける出力を行うため、もし、制御コンピューターが乗っ取られると、たちどころに車両の挙動を支配されてしまう可能性がある。従来のセキュリティ対策では、乗っ取りが発生しないよう、水際で食い止めるための防御壁を強固にすることが主要な目標であるが、この対策だけでは、一旦乗っ取られてしまうとシステムは極めて脆弱である。

4.4.2. レジリエントなセキュリティ対策

図 4.4-2 のようなシステムをレジリエンスにする場合は、システム全体を俯瞰し、レジリエントなシステムの保有するべき、以下の 4 つの機能を組み込む場所を設計する。

- 監視する機能（外敵の侵入に目を光らせる）
- 反応する機能（素早く安全化対策を実行する）
- 学習する機能（過去のリスクパターンを学ぶ）

- 予測する機能（将来のリスクを予測する）

これらの 4 つの機能は、動物が外敵から身を守るために本能的に備えている機能であり、強固な防御壁を作る能力と同様、生き残るためには必須の能力と言えるものである。特に、群れで移動する種族においては、固定の防御壁を設けることが困難であり、上記の 4 つの能力への依存度が高い。自動車もそれと同様、常に移動し続け、道路上、ネットワーク上の外敵に身を晒している環境に置かれており、これらの機能の有効性は高い。

これら 4 つの機能を横滑り防止機能に組み込むならば、それは、図 4.4-3 のような構成となる。

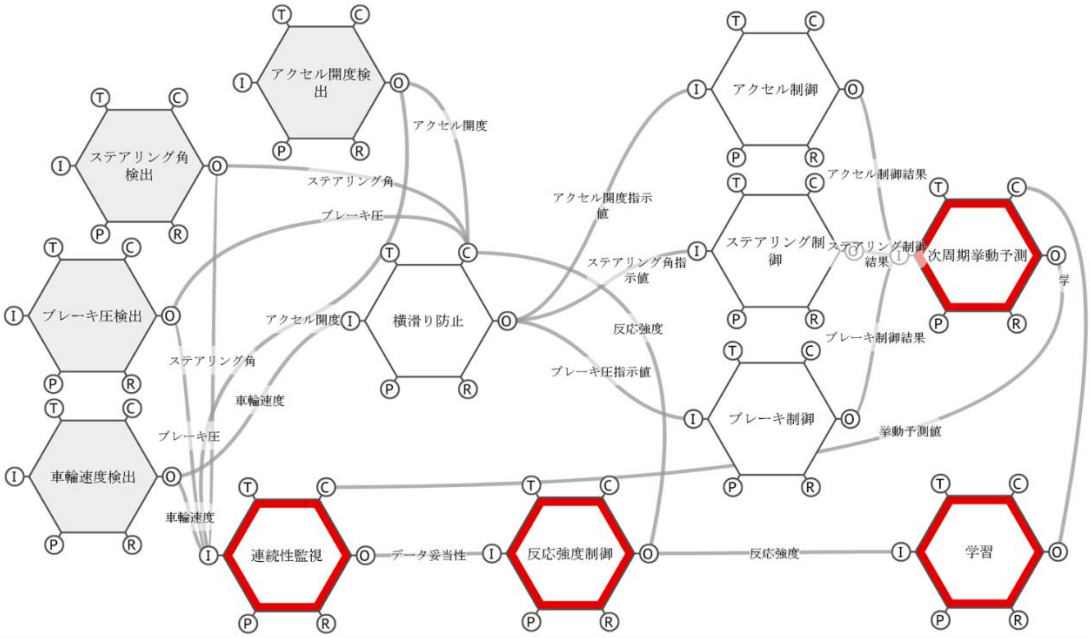


図 4.4-3　レジリエンス機能（赤枠）を付加

ここでは、4 つの機能に相当する、以下の機能が付加され、未知の外敵の侵入によるデータの書き換えが発生した場合でも、危険な反応をせず、被害を食い止めることが可能となる。

- 監視する機能（連続性監視）
- 反応する機能（反応強度制御）
- 学習する機能（学習）
- 予測する機能（次周期挙動予測）

FRAM による分析を行うことで、上記に示した 4 つのレジリエンス機能が存在するか否かが容易に識別可能となる。それらのいくつかが存在していない場合、そういった機能を追加する最適な箇所を検討することで、システムのパフォーマンスを最大限に高めつつ、強固なレジリエンスを獲得できるアーキテクチャの構築につなげることが可能となる。

4.4.3. レジリエント設計の特徴

図 4.4-3 の設計は、実は、現代の自動車設計としては特別なものではない。現代におけるインテリジェント制御を行っているシステムにおいては、データの連続性を監視し、過去のセンサー入

力のトレンドから、将来値を予測することは、一般的に行われている。しかし、これらのインテリジェント機能の多くは、商品性を高める目的で実装されているものであり、セキュリティ向上を主目的にしたものではない。ここで、もし、これらの機能の担う役割に、セキュリティ向上を追加し、外敵の侵入によるセンサー入力の書き換えや制御パラメータの書き換えが発生した場合に素早く安全化できる仕組みを強化した場合、商品性を向上することが、そのままセキュリティ向上に直結するという、好循環が生まれることになる。これこそが、レジリエント設計の最大の特徴ということができる。

4.4.4. レジリエント設計の実装例

一つの実装例として、鉄道における無線列車運行制御システム ATACS(Advanced Train Administration and Communications System)がある[ATACS2012]。我が国の鉄道の運行管理においては、長らく、線路上に設置されたセンサーによって列車の位置を検知し、踏切、信号等の制御を行って来た。しかし、この方式では、外部環境に影響を受けやすいセンサーに依存せざるを得ず、また、瞬間における在線・非在線情報しか持ちえないため、連続的な列車のトラッキングは困難であり、それを実現するためには極めて複雑な検知ロジックを要するため、運行の効率化に対する強いコスト要素となっていた。

従来の地上設備中心の複雑な制御論理の多用を要する方式から、無線を使った連続的な列車トラッキングシステムを実現することを目的に開発された ATACS は、上述のレジリエント設計の実装例となっている。無線通信を使い、列車と中央指令室が相互通信を行うことにより、より直接的にトラッキングを行うことができる一方、悪意を持つ侵入者が通信を改ざんした場合、致命的な事故に至る可能性があるため、ATACS においては、データの「合理性チェック」を実施し、データの ID や通信シーケンスを監視しつつ、それまでの列車運行トレンドから予測される列車位置情報と通信データ間の整合性をチェックする（監視・学習・予測機能）。また、「瞬時の安全回復機能」というセキュリティ機能が付加され、想定外の攻撃が発生した場合に、あらかじめ規定された安全行動を素早く実施できる仕組みを導入している（反応機能）。これらのセキュリティ機能群の付加により、列車間隔の短縮や地上設備の簡素化が実現するため、結果的に経済効果が高まることになるため、セキュリティと経済性が好循環を生む好例となっている。

4.5. まとめ

生物の進化の過程では、様々な動物がそれぞれの種族毎に際立った能力を発展させてきたのは、外敵から身を守り、生き延びるための必要条件であった。また、生物の歴史は、飢餓との戦いの歴史であり、常にコストパフォーマンスの最適化を成し得た種が生き残ってきた。動物の進化においては、最小エネルギーで生き延びられる能力（高い商品性）と、外敵から身を守る能力（高いセキュリティ性）は、本来一体のものであり、互いに競合する 2 つの側面ではない。一方、我々が従来設計してきた人工物の多くにおいては、商品性とセキュリティ性は、トレードオフ関係にあり、強固な防御壁は使い勝手やコストパフォーマンスを阻害するコスト要因としてとらえられてきた。セキュリティを「防御」という、パッシブなものとみなさず、「4 機能」というアクティブなものとみなすことで、このトレードオフのゼロサムゲームを乗り越えることができる。

外敵の侵入に対して守るばかりでは、どのような難攻不落の防御壁でもいつかは破られる。システム設計においても、守るだけのセキュリティでは、悪意を持った侵入者との戦いにおいて、常に後手を踏むことになり、永遠に勝利することはできない。我々は、図 4.5-1 に示したような攻めと守りのバランスを持つシステム作りを目指すべきではないだろうか。

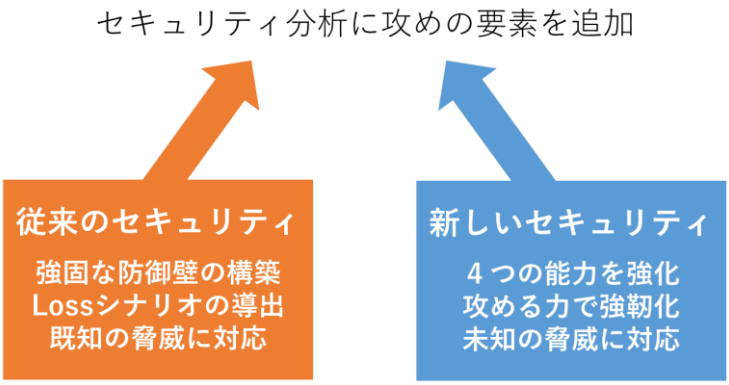


図 4.5-1　守りと攻めのバランス

5. おわりに

2015 年から IPA で活動してきた WG では、システム思考による安全分析の方法論 STAMP と STAMP に基づく手法 STPA の国内普及を目指してきた。本 WG が活動を開始した 2015 年当時、まだ国内で STAMP を知る人はほんの一握りであった。実は、本 WG のメンバーでさえも当初は STAMP 初心者が多かったというのが実情であるが、国内外の STAMP 関連論文・著書の調査、STAMP 提唱者である MIT の Nancy Leveson 教授や欧州の STAMP ワークショップ（ESW：European STAMP Workshop）ステアリングボードメンバーとの連携、多様な仮想システム／実システムの STAMP 分析試行と熱のこもったレビューなどの活動を行うと同時に活動から得られた知見をガイドブックとして都度公開してきた。これまでに公開してきた活動の成果物「はじめての STAMP/STPA」は既にシリーズ発行部数が 15,000 部、pdf のダウンロード数も 15,000 件に上り、日本における STAMP の教科書との評価も頂いている。

4 年間の活動を振り返ると、国内の STAMP 活用の盛り上がりについてしっかりとした手ごたえを感じる。STAMP ワークショップ in Japan は、2016 年に第 1 回を九州で開催し、その後は場所を東京に移して 2017 年に第 2 回、2018 年に第 3 回を開催してきた。参加人数は 2016 年が約 120 名、2017 年が約 180 名、第 3 回では約 280 名と右肩上がりの参加を得た。特に、第 3 回は 17 業種の企業・団体からの参加を得ており、まだ普及レベルには到達していないものの、産業界での STAMP 認知度の確かな高まりが感じられる。

本書では、システム思考に基づく安全分析の長所をうまく活用することによって効果を得られた事例を産業界から提供いただいて掲載している。

「2.1 列車警報システム」は、分析する場面に応じて FTA と STPA を使い分け、それぞれの長所をうまく活用することによって、安全分析にとどまらず次世代システムの更なる改善案を発想することに成功した例である。

「2.2 高齢者見守りサービス」は、分散協調型システム開発において曖昧になりがちな安全責任の所在についての問題点を見易くし、安全責任の所在とサービス運用形態を結びつけて考えるのに STAMP のモデルを活用している。

「2.3 自動車製品ライフサイクル」は、自動運転車の製品ライフサイクルという巨大システムを対象とした試行で、全体俯瞰することによって見えてくる様々なリスクを抽出できることを確認できた例である。

これらは具体的に STAMP の効果を体験できた例である。これらの例に限らず、STAMP を知った当初は「これからの複雑システムの安全分析は STAMP でなければならないのか？」と感じていた人も、システム思考に基づく STAMP の考え方の本質を理解し、STAMP の長所を知るにつれて、「既存手法か STAMP か」ではなく、分析対象や分析する場面に応じて、それぞれの長所を活かせる手法を活用することの有用性を感じるようになってきたのではなかろうか。

物理学が革命的発展を遂げた 20 世紀初頭の物理学の英雄時代に英雄の一人として登場し、量子

力学確立のみならず、その後の哲学・思想にも大きな影響を与えた天才物理学者ハイゼンベルクは海を眺め、波の美しさを見るときに、波を構成する素粒子同士の働きから考えることの無意味さを述べている（「Der Teil Und Das Ganze」、邦題「部分と全体」[Heisenberg1971]より）。場面によって、全体俯瞰的考え方が適している場合があり、逆に仰視する考え方が適している場合もあるなど、考え方の使い分けが必要ということである。21 世紀初頭の現代、IoT/AI 時代となり、複雑システムの安全分析においても同様に、場面に応じて考え方や手法を使いこなせることの重要性が増してきた。今後は、システム思考に基づく安全の考え方として STAMP だけではなく、レジリエンス・エンジニアリング（Safety-Ⅱ、そのモデリング手法 FRAM）や、Safety 2.0 なども有効になってくるであろう。これまでに経験したことが無いほど急激にシステムが大規模・複雑化しており、それに対応するため安全分析の考え方や手法は更に改善が続くと考えられる。今後の複雑システムの安全確保に向けて、それらの考え方や手法を、どのように使い分けてゆくかを考えるきっかけとして本書を活用いただければ幸いである。

参考文献

[1] [IPA2016]
IPA, “はじめての STAMP/STPA,” 2016. https://www.ipa.go.jp/files/000055009.pdf.

[2] [IPA2017]
IPA, “はじめての STAMP（実践編）,” 2017. https://www.ipa.go.jp/files/000058231.pdf.

[3] [IPA2017-2]
IPA, “情報処理システム高信頼化教訓集 2017 年度版 PART I,” IPA, 2017.

[4] [IPA2018]
IPA, “はじめての STAMP（活用編）,” 2018. https://www.ipa.go.jp/files/000065199.pdf.

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

[6] [Leveson2017]
N. G. Leveson, “Analyzing Accidents and Incidents with CAST,”
http://psas.scripts.mit.edu/home/wp-content/uploads/2017/03/Nancy-Leveson_CAST-Tutorial-2017.pdf.

[7] [Leveson2017-2]
N. G. Leveson, “CAST Analysis of the Shell Moerdijk Accident”.

[8] [Leveson2018]
N. G. Leveson, “STPA Handbook,”
http://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf.

[9] [Leveson2019]
“Nancy Leveson's Home Page at MIT,” http://sunnyday.mit.edu/.

[10] [兼本 2018]
兼本茂, “これからの複雑システムの安全分析 STAMP/STPA,” SEC Journal, 第 52 号, pp. 19-22, 2018.

[11] [割れ窓]
“割れ窓理論,” https://ja.wikipedia.org/wiki/割れ窓理論.

[12] [失敗知識]
“失敗知識データベース,” http://www.shippai.org/fkd/cf/CA0000639.html.

[13] [中村 2017]
中村英夫, “IoT 時代の新しい安全「Safety 2.0」の全貌,” 2017.
https://www.ipa.go.jp/files/000062789.pdf.

[14] [SAE2018]
SAE International (Society of Automotive Engineers) J3016 Levels of Driving Automation,
https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic

[15] [Nelson2008]
P. S. Nelson, “A STAMP ANALYSIS OF THE LEX COMAIR 5191 ACCIDENT”.

[16] [近藤 2018]
近藤駿、佐藤草太、濵口孝司、橋本芳宏, “ICS におけるセーフティとセキュリティのための STAMP モデル適用,” STAMP ワークショップ, 2018.
[17] [早川 2018]
早川拓郎、金子朋子、佐々木良一, “STPA を用いたリスク分析・対策選定手法の提案と適用,” 第 3 回 STAMP ワークショップ, 2018.
[18] [林 2018]
林浩史、髙橋雄志、金子朋子、佐々木良一, “STAMP/STPA を用いたリスクコミュニケーションツール,” STAMP ワークショップ, 2018.
[19] [Hollnagel2018]
E. Hollnagel, “To feel secure or to be secure, that is the question,” 2018.
[20] [ESC2019]
“Wikipedia, "Electric Stability Control",”
https://en.wikipedia.org/wiki/Electronic_stability_control.
[21] [ATACS2012]
東日本旅客鉄道株式会社, “無線を用いた新しい列車制御システム ATACS の安全確保の考え方,” https://www.ntsel.go.jp/forum/2012files/1107_1350.pdf.
[22] [Heisenberg1971]
W. K. Heisenberg, Der Teil Und Das Ganze、邦題「部分と全体」, 和訳　みすず書房, 1971　和訳 1974.

索引

本書で用いる略語

CS 図	コントロールストラクチャー図
CA	コントロールアクション
FB	フィードバック
UCA	Unsafe Control Action （非安全なコントロールアクション）
HCF	Hazard Causal Factor （ハザード誘発要因）

編著者（敬称略、50 音順）

IoT システム安全性向上技術 WG（主査、以下 50 音順）

主査	兼本　茂	公立大学法人会津大学
	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
	金田　光範	地方独立行政法人東京都立産業技術研究センター
	北村　知	東日本旅客鉄道株式会社
	杉浦　弘人	東日本旅客鉄道株式会社
	永井　康彦	株式会社日立製作所
	中村　英夫	東京大学大学院
	中村　洋	株式会社ジェーエフピー
	野本　秀樹	有人宇宙システム株式会社
	橋本　岳男	株式会社日立産業制御ソリューションズ
	福島　祐子	日本ユニシス株式会社
	余宮　尚志	株式会社 東芝

IPA（50 音順）

石井　正悟
金子　朋子
向山　輝

付録　「IoT/AI 時代の安全を考える」

IoT システム安全性向上技術 WG では、第 9 回の会議において、有識者を招き、「IoT/AI 時代の安全を考える」というテーマで意見交換を行った。ここではその内容を紹介する。

第 9 回 IoT システム安全性向上技術 WG
開催日 ：　2019 年 2 月 20 日　10:30-12:30
開催場所：　IPA 13 階会議室 C（文京グリーンコート）
議論テーマ：　「IoT/AI 時代の安全を考える」
参加者　明治大学名誉教授　向殿政男　　東北大学名誉教授　北村正晴
早稲田大学教授　小松原明哲　　日本大学名誉教授　中村英夫
会津大学名誉教授　兼本茂
IPA 社会基盤センター長　片岡晃（ファシリテーター）

趣旨説明

（片岡）

IPA では、これまで 4 年間にわたって、複雑システムの安全性向上技術に関する WG 活動をしてきましたが、そのまとめも兼ねて今回の座談会を企画しました。近年、自動車、列車、ロボットなど多くの工学製品が複雑化・知能化していますが、その安全性をどうやって確保したらよいかという懸念が高まっており、IPA でも、システム理論による事故モデル STAMP の勉強会をきっかけにして、複雑システムの安全性向上のための議論を進めてきました。この座談会では、この安全性向上のために今後何をすべきかという点に焦点を絞ってご意見をいただきたいと考えています。IPA でなぜこういう安全を考えるのかは不思議に思われる方がおられるかもしれませんが、現在はソフトウェアで安全を制御するシステムが普通になってきており、IPA の大事な関心事に含まれます。

最初に、兼本先生から今回の座談会の趣旨を説明頂けませんでしょうか。

（兼本）

安全分野はとても広いのですが、従来議論されている機械安全や労働安全ではなく、「システム安全」という視点で本日の議論を進めて頂ければと思っています。近年の工学製品、特に、我々の日常生活に入り込んでいる製品は、ICT 技術の進展とともに急激に複雑化しており、それらの安全性はソフトウェアで保たれているといっても過言ではないと思います。IoT/AI 時代と言われていますが、そこでは、システムが相互にインターネットを介してつながることで高度な機能を発揮していますが、同時に、セキュリティ問題や安全責任の分散化といったリスクも存在します。さらには、知的なソフトウェアにより人間と機械が協調して安全を保ったり、究極的には、人間を全て置き換えた自動運転がなされたりするといった技術開発も急速に進展しています。

このような状況の下で、現状の安全設計法や安全規格の限界も見えており、「システム安全」という視点で、STAMP、Safety-I&II、Safety 2.0、レジリエンス工学など、いろいろな新しい安全工学

の考え方が、今回参加されている先生方からも提唱されています。工学的な方法論だけでなく、安全文化や法制度、工学倫理との関係も大事になってくるかもしれません。

今回、「IoT/AI 時代の安全を考える」という表題を掲げたのは、先生方の提唱されているさまざまの新しい「システム安全」に関する考え方を、それがどうして必要になってきたのか、また、それらをどのように社会実装してゆくのかといった視点から、まとめてみたいと考えた次第です。

結論ありきの提言ではなく、安全に関わる技術者がそれぞれの価値観で読み取って自らの将来の行動の参考にできるような、多様な提言をお願いしたいと考えております。

現状認識

（片岡）

趣旨説明ありがとうございました。議論のきっかけとして、現状の問題点の認識から議論を進めたいと思います。

（兼本）

趣旨で申し上げた中にも含まれていますが、現状の安全規格、特に、向殿先生がよくご存知のソフトウェアに関して大事な機能安全規格 IEC 61508 では、AI のような複雑な安全制御や、人間の複雑な行動は排除されています。その一方で、自動運転に代表されるような技術開発がどんどん進んでいます。AI 分野では、人間と機械の協調安全制御のつもりが、両者の競合により事故を引き起こしてしまう可能性があります。IoT 分野では、大会社の統合開発からベンチャーやオープンソースを用いた分散開発の時代に入っており、そこでの安全は保たれるのかという疑問が生じてきます。だれがどんな安全責任を持っているのかが必ずしも明確でないシステムが多いような気がします。法制度や工学倫理の問題も議論が必要になってくると思います。また、セーフティとセキュリティの統合開発についても、まだ、議論が始まったばかりの段階ではないでしょうか。

（向殿）

IoT だけでなく AI も含んだ ICT 全般のシステムの安全とみなして議論を進めさせていただきたいと思います。たとえば、セーフティとセキュリティの違いでは、何から何を守るかを明確にしておかないと議論が混乱します。セーフティでは、人の命を守るのが第一で財産はその次になるし、セキュリティでは、財産や情報を守ることが焦点になります。特にセキュリティでは悪意ある攻撃を前提にしないといけない。

機械安全の経験では、単体の機械の積み上げでシステム全体が出来てくるというボトムアップの発想で安全を考えるが、システム安全の世界では、全てが分かって積み上げてゆくというアプローチは難しいし、ハザードも全てを明確にして進めるというのも困難な、いわば、不安定な状況でシステムを設計してゆかなければいけない。そこに、さらに IoT のようなつながるシステムが出来てくると、想定外のハザードが起こるかもしれないが、それにどう対応するかが悩ましいところです。そこで今やっているのは、システムをブロックに分けてそこで何が起こるかを分析し、さらに、それをつなげたとき何が起こるかを考えるという階層的な発想で安全を考えている

のが現状です。安全の専門家が考えてきた「どこにハザードがあるか」という発想と、セキュリティの専門家が考えてきた情報漏洩や改ざんなどの発想が、どこでどうつながるかを考えることが大事になると思います。

安全における機械と人間の役割では、従来は、機械がまず安全を確保して、残ったリスクを人間が補い、さらに、事故が起こったら保険で補償するという考え方をとってきました。一方、今は、機械が知能化してきており、人間と機械が協調して安全を担保するという新しい時代に入っている。いままでの考え方は通用しない、面白い時代に入っている。そうすると、責任分担をどうするか、どうやって認証するかという新しい問題もでてきます。機械と人間のどちらに責任があるかの法制度も整っていないのが現状であり、法制度だけでなく、社会制度や、場合によっては工学倫理などの見直しも必要になってきます。

（北村）

向殿先生がうまく問題点を整理頂いたので、そこで述べられていない部分を指摘してみたいと思います。まず、つながるシステムと IoT では、この言葉を聞いて考える技術領域が人それぞれで違うだろうし、カテゴリによっても違うと思います。インフラ領域では、センサーを多数ばらまいて、その情報からいろいろなサービスを提供しようとする考え方があります。システムへの入力が多数化・多様化した際の安全問題も考える必要があるのではないでしょうか。

AI についてもいろいろな意見があります。「それ、AI でなんとかならないか」と言われることがよくあります。「Unknown Unknown」（知らないということに気づいていない）のようなものに対して、AI で何とかしてくださいという発想では何も進まないと思います。インテリジェンスの源泉はディープラーニングだけではない。AI には、Abduction（仮説推論）や Analogical reasoning（類推）などいろいろな技術があるので、そちらへの注目も必要だと思います。

セーフティとセキュリティは深刻です。いままで、悪意は考えないというのが安全の考え方だった。セキュリティは悪意の塊のような世界で、インターネットにつながっていないので安全と言ってすむ世界ではない点に注意が必要です。

自動運転みたいな話だけでなく、MaaS (Mobility as a service)のようにトランスポーテーションシステムの形を変えてしまう技術も問題になると思います。鉄道と自動車のリンクなどがあります。便利にはなるが、責任の所在が曖昧になる。限りない便益性を考える時代になっており、副作用は起きた後で考えようという風潮はあるが、これを批判するだけでは何も進まない。企業の中で便益性の追求を抑えるのは現実的ではない。従って、先端技術については、特定の企業が考えるのではなく、社会システムとして倫理学者や政治学者、法学者などを含めて考える必要がある。ただ、誰がこれらの様々な人の意見の間をつなぐのかも課題でもあり、技術屋の責任でもある。このつながりの不足は、社会の考える安全と技術者の考える安全に差を生じさせ、大きな事故の要因になることがあり得ると考えています。このつながりを機能させるには、技術者自身の説明能力というのが非常に大事になります。

（中村）

先に出た 3 つの課題であるつながるシステム（IoT）の問題、人間と機械の協調による安全制御

の問題、安全責任の分散や法制度の問題は、それぞれに解をあたえればいいかというと、そういうやり方ではうまくいかないところに難しさがあると思います。北村先生のおっしゃるように、サービスや生産性の追求がものつくりを生み、複雑システムを生み、さらには、IoT をベースにした新しいものつくりが始まりつつあります。そこでの問題点は、安全哲学や倫理まで思いが至らないところだと思います。これらの課題を統一的に捉える考え方が大事と思います。IoT でいえば、安全責任が分散してしまうという副作用ではなく、安全性を高める IoT の使い方もあるでしょう。複雑系の利便性を享受しながら、複雑系の持っている弱さをどうやって克服していくかを考えると、新しい複雑系の在り方が出てくるのではないでしょうか。安全面での脆弱さを認識しつつ安全性を向上させる方法論を多くの議論の中で見い出す必要があるでしょう。ただ、この議論が各ドメインでばらばらに行われてはならないと思います。

ものつくりの世界では、いいもの、便利なものを作れば、社会に広がってゆくと考えてきましたが、最近は、一定の基準・規範にのっとっていることが普及するための前提になっていて、利便性よりも規格を満たしていることが大事にされています。この風潮が逆に新しいものを生み出す障害になっているようにも思います。ものつくりの世界でもシステムを統一的に考え、今の作られ方やシステム論の課題を明らかにする必要があると思っています。

（小松原）

人間工学や人間生活工学を中心に研究しています。これまでの論点と外れるかもしれないが、人工物に関して安全の議論をするときに、安全という言葉をいきなり使わないほうがよいと思います。狭い見方かもしれませんが、人工物の設計では、その設計目的と機能があるはずで、その機能を安定的に提供できず、結果的にネガティブ効果がでてしまったことを事故とみなすのだと思います。裏返すと、機能が安定的に提供されている状態が安全といえると思います。ですので、目的とした機能を如何に安定的に提供できるかという視点で議論したほうがわかり易いのではないか。よく、安心安全といいますが、その二つの間に安定的な機能提供があって人々が信頼する。その状況が安心につながるのだと思います。安定的な機能が提供されないと信頼できないし、不安になってしまう。そうなると、安定を損なう事象を見つけることが大事になる。機能提供が停止してしまう、システムが暴走する、期待外れの結果になる、設計者の意図しない副作用が出る、などのいろいろなリスクがあると思います。

少し話がそれますが、ペットボトルが出てきたとき、その便利さのネガティブな副作用として、研究室では、従来あった協力してお茶の準備をしたり片づけをするといった文化がなくなってしまったことを経験しています。同じように、人工物は設計者が想定しなかった副作用をもたらすことがあり、それが安全を脅かすということもある。この副作用はすぐに出てくることもあるし、先々に出ることもある。だからといって、ペットボトル禁止といったことにすると更なる副作用が出るかもしれない。AI を止めるという語論も同様であろう。この副作用は自然に出てしまうこともあるし、悪意を持って出されてしまうこともあり、これがセキュリティ問題であると言えます。

IT や IoT は、目に見えない形で社会に浸透してくると思いますが、人間の行動支援として使わ

れても来る。これまでの人間行動ではできなかった新しい世界を築くことになるかもしれないが、そこには新たな問題が出てくることもあるのではないでしょうか。

安定的な機能提供を中心に、安全、安心、信頼を議論してゆけば、責任や法制度も議論しやすくなるのではないか。誰がどんな結果責任や設計責任をとればよいかが明確になってくる。

また、STAMP や FRAM、レジリエンス工学のような技術では、それぞれ研究者が研究の前提としているシステムは異なるので、どんなシステムを想定してそのツールやアイデアを出したかを理解することも大事になる。これらのツールは、どんな対象でもうまくゆくわけではなく、適材適所でうまくいったりいかなかったりするので、それぞれの限界を把握しておくことも大事になる。

（片岡）

いまのそれぞれの現状認識についてご意見、質問はありませんか？

（兼本）

機能を安定的に提供しなければいけないという小松原先生の話は、その通りだと思うのですが、ペットボトルの登場の例であったように、新しい便利な機能が提供されたとき、副作用として従来の文化が失われるといった事例は多くあると思います。では、その便利さと文化の喪失という二つの効果をどうやって世の中の人にわかってもらうか、または、どういった妥協点を見つけるのか、といったことはどのように対処したらよいのでしょうか？

（小松原）

新しい機能が出てきたとき、古い文化が失われるのはやむをえないのではないでしょうか。新しい機能に合った新しい文化を作ってゆくしかないのだと思います。携帯の普及で位置情報が把握されるというのも、副作用かもしれないが、新しい生活文化の事例といえそうですね。

（向殿）

科学技術の進歩は、いままでも同じことが起こっていると思います。

話は変わりますが、安定というのは社会では重要だと思います。ハザード（危害）は何かを考えるときに、安全屋は、人の命を第一に考えるし、車が止まった、商品が止まったなどの財産の問題もあります。どれを優先するかということ。それを整理しないと議論がかみ合わないことになる。

（北村）

何が安定かの議論を言い直すと、安定であるということは、生命の維持に寄与しているという捉え方もできるのではないでしょうか。

（向殿）

安定が止まってしまうときに便利さが損なわれる。その MAX が人の命といえます。

（北村）
逆に言うと、安定が保たれれば、人命も守られる。優先度の問題ではなく、どの切り口から課題に取り組んでいくかという視点が色々あるのではないでしょうか。

（兼本）
安定が失われたときにどう行動するかは技術者だけでは発想しづらい。例えば、防災対応時にエネルギーが切れたらどうなるかといった問題は、いろいろな立場の人と議論することでヒントが見えてくることがある。先ほどのペットボトルの存在で旧来の良い文化がなくなるといった気づきは、小松原先生のようにヒューマンファクターを研究されている方だから思いつくことかもしれない。ペットボトルを一生懸命開発してきた技術者にとって、その便利さに隠れた副作用まで思い至るのは難しいのではないか。先ほど北村先生の話にありましたが、異なるバックグラウンドを持った人どうしの議論が大事ということを示唆しているように思います。これは、想定外を想定するということとも関係するので、後でもう一度議論させていただきたい。後知恵で気づくというのは、事故発生の後の良くあることですが、これを事前に気づくにはどうしたらよいか、という問題とも関係しているように思います。

（小松原）
新しい技術が入った時に生活がどう変わるかを予測するという技術はないわけではないので、それを使うという方法もある。

（兼本）
話題は変りますが、中村先生のモノづくりの限界という話で、規格を守れば十分だと思ってしまって新しいものつくりの知恵が出てこないという話がありましたが、具体的な事例はあるのでしょうか？

（中村）
列車制御の分野では、それが問題になっています。新しい規格が出来ると、その認証をビジネスにする人がでてくる。ところが、認証をビジネスにする人は、必ずしもその対象分野の専門家ではないので、認証してもそれがいいものに結びつくわけではない。一方、製造側では、いいものを作るというよりも、認証者に理解してもらうことが主目的になり、閉塞感が出てきます。

(兼本)
それを打ち破るにはどうすればいいのでしょうか？

（中村）
ものつくりに自信のある日本の会社では ISO9000 の品質規格をとらなかったところもあるが、国際的なリコール問題が発生したとき、第三者認証をとっていなかったことが対応を困難にした

場合もあると伺っています。そのため、機能安全の規格である IEC 61508 や ISO26262 の認証には前向きに取り組むようになりました。ただ、それに従って作ればよいものができるとは全く考えていず、独自のものつくりの文化を醸成しているようです。

モノで競う、安全性で競う、考え方で競う、といった文化を国際規範として作っていかないといけないと思います。とても大きな壁ではありますが。

（兼本）

いま、JASA（組込み技術協会）で、安全設計の規格の解説書を執筆中などですが、規格を説明しようとするとどうしても表面的なルールの説明になって、そのルールの背景にある安全哲学のような考え方が伝わらないと感じています。

（中村）

規格の文章には、その背景の考え方までは書かれていない。日本は、過去に品質でもって世界を席巻したが、その時に欧米でとった戦略は、モノの実際の品質で勝負するのではなく、どういった作り方をしたかというプロセスで評価することでした。これにより最低限の品質は保たれるであろうという考え方です。その結果、製品品質における日本の優位性を打破しました。この成功体験が、その後の欧米の規格戦略に表れていると思っています。ただし、機械部品と異なって、ソフトウェアを対象にした場合、結果としての品質を評価するのは簡単ではないので、その製作プロセスを重視するという考え方をとる必然性はあったかと思います。その点でソフトウェアに対する機能安全規格の意義は認めています。

（北村）

それは大きな問題だと思っている。そういう規格で競争に勝とうというのは、欧米は得意です。一方、日本は匠の技でものつくりをやってきた。いいものを作っても、規格という大きな罠をかけられて競争に負けてしまうことがこれまで何度もあった。

最低限この基準をクリアしてくださいというのが規格であって、最高のエクセレンスを目指すのは別の話だと思う。エクセレンスを目指すアプローチと、最低点を目指すアプローチは区別して考えたほうがいい。

（向殿）

現実には、規格は社会的責任を果たすための便法にすぎない。「規格を満たせばいいんでしょ」という考え方でものつくりをすると本末転倒になってしまう。規格や法律は最低基準であって、それをはるかに越えた安全性や信頼性で競争しないといけない。最低基準を満たしているかどうかで競争するのはあまり面白くないということです。

（中村）

日本の規格はこういうものを作りなさいといった構造規格が多いが、欧米は、作り方を規定するプロセス規格、すなわち、求められる各プロセスの要件に対しどうやったか記録し、エビデン

スとして残しなさいというものが多い。ソフトウェアに対しては仕方がないが、それがハードウェアやシステムの作り方まで適用されるというのは問題だと思います。

(向殿)
日本の規格は仕様を決めるものが多いが、欧米は、こういうステップで作りなさい、エビデンスを残して説明責任をとれるようにしなさいというのが多い。

（兼本）
規格の説明として、なにかあったときの説明責任を果たせる、というのがあるが、現実はそうでもない。事故が起これば後知恵で責任を問われるし、PL 法（製造者責任）での責任もある。こういった問題をどうすれば乗り越えられるのでしょうか？

（向殿）
説明責任は、刑事と民事で違い、民事だととことん責任を追求されます。刑事では、基準があってそれを満たしていれば情状酌量の余地がある。逆に、書いてあることをやらないと罰則はでてくる。

（小松原）
プロセスの説明の責任、結果の責任と分けて考えてものつくりを進めてゆくことが大事でしょう。キリスト教国の考え方では、神様に全てを説明できる状況を作っておきなさいということですね。結果だけよければよいという考え方ではなく、作るプロセスまで説明できるようにしておかないといけない。

（中村）
ただし、いくらプロセスが良くても結果が悪ければ神様は怒るということに気を付けておかないといけませんね。

（向殿）
製品そのもの性能を結果として評価するというのは大事です。大学の学生でも、卒業したら一定の評価をするということではなく、本人そのものを評価しないといけない。

（中村）
機能安全規格 IEC 61508 の初版では、第三者認証だけでなく、自己認証であっても説明責任が満たされればよいということになっていたが、いまは、現実的には第三者認証が要求される時代になっている。

（兼本）
大学で安全を教えるときに規格の考え方は大事ですが、そこで、表面的なことだけ教えると、

よい学生が育たないと思うのですがどのようにしているのでしょうか？

（中村）

規格にはいい面と悪い面があり、上手に使う知恵を教える必要がある。また、開発、調達、国家戦略、ビジネス戦略など使う立場によって使い方を考えることも大事ということを含めて教えています。

（兼本）

いろいろ規格の話題が出たが、話題を戻します。北村先生のお話で、産業界では、便益を追求するのが本能であり、そのとき起こる想定外の副作用はあとで考えざるを得ない、といった話がありましたが、それを改善する手段はあるのでしょうか？

（北村）

分野にもよるが、原子力のように起きてから考えようというやり方は許されない分野もある。想定外という話があったが、これは想定外というより、想定除外というべき事例が多いのではないでしょうか。全く思いつかない事象はめったにないのではないかと思います。ただ、それを極端な事象なので「対策をしない」という判断をしたことを、エンジニアリングジャッジメントであると言ってしまうと、一般の人には分からなくなってしまう。もう少しわかり易く説明する可視化手段はあるかもしれない。しかしながら、その一方で、何らかの想定除外をしないと人工物は作れないというのも事実である。

（兼本）

車の自動運転の安全の議論でも、安全規格 ISO 26262 を守るだけでは十分でなく、SOTIF（Safety Of The Intended Functionality）という考え方で、他のドライバーの運転や天候など、システムのエラー以外の安全上のリスクを想定することも必要という議論が始められています。悪天候下での自動運転は当然想定しないといけないが、その限界をどこにして設計をするかは自明ではない。

（向殿）

設計するということは基準を作るということ、基準を作るというのは、それ以上は見るのを止めましょうということですね。

（北村）

設計基準を超えるものは必ずでてくるが、それは人間系がカバーするとか、住民に避難してもらうといったことが必要になる。その意味で、想定そのものは真剣にする必要があるが、設計に際しての想定除外という考え方は必要になる。

（兼本）

想定除外の事象というのは、あまり表面に出ないのではないでしょうか。つまり、想定除外し

たあとは、なかったものかのように扱われてしまう。

（北村）
安全対策を、ゼロにするか 100 にするかと考えてしまうとものを作れない。想定除外で安全とはいっても、対策はゼロにするのではなく、100 の代わりに、例えば 2 や 3 の対策はするという考え方はあってもいい。

（向殿）
安全技術者で絶対安全なんて考えている人はいない。ただ、社会から安全なんですねと念をおされると、それ以上の対策の議論が出来なくなり、2 や 3 の対策を準備しておけなくなる。いざというときのためにロボットを配置しましょうと言っても、それは事故が起こるということですかと思われ、結局は、配置できなくなることもある。

（北村）
現状ですでに安全と言って良いレベルにはあっても、もっと対策をすることは妥当な姿勢だという説明が必要でしょう。安全というのは、安全か危険かの 2 モード問題ではないという安全哲学の考え方の説明が必要だと思います。

（兼本）
そこに、レジリエンスという考え方が出てくるということですね。STAMP による安全分析で、確率論ではなくワーストケースで考えるという議論があります。もちろん、ワーストケースといっても、隕石が落ちるような極端なことは考えません。ただ、ワーストケースで事故の可能性を定性的に考えると、ゼロか 100 の安全対策ではなく、2 や 3 の安価で簡単な対策、場合によっては、人の能力に頼った対策のアイデアも出てくる可能性もあるのではないでしょうか。
話題が少し変わりますが、セーフティとセキュリティの問題はどうするか意見を聞きたいと思います。

（北村）
IPA では、セキュリティ問題をどう考えているのでしょうか？セキュリティ専門家と意見交換や現場見学をした折には、様々なシステムがひどい攻撃をされている実態を知ることができました。

（片岡）
IPA では、産業サイバーセキュリティセンターというのを作って、重要インフラ系に対するサイバー攻撃にどう対処するかを考えています。人材育成も扱っています。特に、欧米で起きているような悪意を持った攻撃によって発生する問題にどう対処するかを考えています。ただ、日本の場合、それほど悪意のある問題を経験してないので、サイバー攻撃に対する感度が高くなかったかもしれません。それでも、東京オリンピックに向けて多くの攻撃を受けているのですが、そ

れが表に出ていないのが実状です。先ほどの安全神話と同様に、攻撃を受けているということを表に出して対策の議論をしていかないといけないと思っています。その意識改革を、特に、経営者層にたいして啓発してゆき、対策をしてゆく必要があります。

（北村）

仮想通貨に関連していろいろ不適切なことが行われているというのは、重要インフラとは別次元の話なんでしょうが、実生活に大きな影響をもたらし得ると思います。もうひとつは、政府中枢の意識は良く分からないが、セキュリティに関する国家の戦略が無防備に近いように思える。本当のところ現状はどうなっているのか気になるところですね。

（兼本）

セーフティとセキュリティの議論も、実務をやっている人から聞くんですが、ソフトウェアでいうと、セーフティは分かりやすく作れ、セキュリティは分かりにくく作れ、というたがいに矛盾する規格がいっぱいあると、これをどうしましょうかというのがこれからの課題だということです。技術的には、いったんわかり易く作ったものをソフトウェアでシャフリングして分かりにくくして実装するといった方法があるかもしれませんが、まだ、具体的に決まったことは何もない状況だと思います。この話は一例ですが、安全工学者とセキュリティ工学者の間の意思疎通は、あるようでないのでは？

（向殿）

ないですね。この前議論をしたのですが、セーフティ屋さんとセキュリティ屋さんで相当意識が違うとういことが分かった。セーフティ屋は、「人が死ななければいい」と思っている。つながらないから関係ないと思っているところをセキュリティ屋さんは一生懸命やっている。セーフティ屋さんは、まさかそんなものがつながって悪意でもって人が死ぬなんてことはないと思っている。でもつながってみると、これは可能性があるなっていう話になって、議論し始めてみると、そもそも安全に対する意味が違っていて、何から守るかも違っている。セーフティ屋さんは、今まで、悪意のある攻撃が工場のラインにも入ってくるなんて思っていなかった。そういうことは想定しないでやっていた時代と、セキュリティ屋さんの発想は違っていて、これはどうやったら話が通じるかというと、用語の定義からやりましょう、という話になった。安全ではこうやっているけどセキュリティではこう考えている、それがついにつながってきたからお互いどこまで協力しましょうかと。どこでつながっているかを良く考えてみると、ハードウェアから見ると、コンピューターのソフトウェアというか機能安全のところ、そこにバグがあって何か入ってくると人の命に関わるような可能性がある。この機能安全のソフトウェアのところがキーポイントで、そこでセーフティとセキュリティがつながるのだろうか、といった話をしているところです。

（兼本）

セーフティとセキュリティのつながる場所を見つけようとしているということですね。

（向殿）
はい。別々の用語を使って概念も違っているけど、いっしょに話が通じるようにしましょうという話と、どこが接点なのかという話があります。
セキュリティ屋さんにセーフティまでというのはかなり難しい話で、セーフティ屋さんはセキュリティのことはほとんど分からない。それをお互い理解しあいながら、フィジカルとサイバーが一緒になったときにどう安全を守るか、安定を守るかという議論ができるように、通じる話し方ができるようにしましょうよ、という話をしているところですね。

（片岡）
それはすごく大事な話で、さきほど申し上げた人材育成では 100 人くらい毎年来るのですが、70%が IT 屋（セキュリティ中心）で、30%が OT 屋（制御屋さん）で、最初はまったく会話が通じない。IT 屋さんには OT のこと、OT 屋さんには IT のことを勉強してもらうのですが、なかなか実感として自分の中に入ってこないようですね。いっしょにワークをしてもらうと分かってくるのかもしれませんが、さきほどの結び付くところ、接点を明確にしてあげるのが大事かもしれませんね。

（向殿）
最後に役割分担も明確になってきて、やっと法律などの責任問題にもつながるのだと思います。

（兼本）
鍵をかけておけばセキュリティ的に安全ですが、いざというときには助けに行けなくなるのでセーフティ的には問題だと、そういう話を両者の立場で話し合わないといけないということですね。

（中村）
実際には無線を使ったコントロールが多くて、そうすると鍵をかけようがないのですね。我々も無線を使った制御はやっていてセキュリティは重視しています。ただ、セーフティに関しては議論できるのですが、セキュリティに関してはオープンに議論できない。悪意のある人に悪用されるからですね。本当にクリティカルな制御の世界のセキュリティは、プロに別系で見てもらっているというのが実情です。従って、どういう風にやっているかということも含めてオープンにできないところがあります。

（兼本）
難しい課題ですね。セーフティの人がセキュリティを勉強するのが早いか、セキュリティの人がセーフティを勉強するのが早いのか、という議論をよくしますが、答えはないですね。セーフティの人がもうちょっとセキュリティを勉強した方が、致命的な事故を防ぐにはいいのではないか、という意見が強いですが、それ以上具体的になっていません。いずれにしてもこれからの課題ですね。

安全の哲学

（兼本）

次のテーマに移りたいと思います。

安全の哲学に関してご提言をいただきたいのですが、レジリエンス性の実現や Safety I & II というのが福島の事故以来かなり注目を浴びてきています。また、MIT のナンシー・レブソン教授の話を聞いたときに、創発（Emergence）という言葉と reductionism（還元論）という話が出てきて、STAMP の理論は創発論であるという話があります。システムを俯瞰して見る「システミック」と、論理的にものごとを分解してみるという「システマチック」という言葉も STAMP の本には出てくるのですが、このあたりがいろんな人に理解してもらうのが難しい。よくよく考えてみると、モノを作るときにはどうしても論理的に考えることが必要で、システミックであろうとシステマチックであろうと、上からだろうと下からだろうと論理性は必要なのだろうという気がします。また、システム思考と擦り合わせ技術という考え方もあります。日本は、擦り合わせ技術は得意だが、トップダウン型のシステム思考は苦手ではないかということです。米国から導入したプラントの事例では、システム思考という視点では実によく作られているが、いざ動かしてみると多くのトラブルが出てきて、日本型の擦り合わせ技術でトラブル原因を究明し改善してゆくということがしばしばありました。先ほどの規格の議論で、結果を重視するかプロセスを重視するかという話と似たところがあり、どちらかに偏りすぎると良いものはできないということではないかと思っています。このような視点からご意見をいただきたいのが一つです。

もう一点、想定外事故をどうやって減らすかということについてもご意見を伺いたい。さきほど「想定除外」という話はその通りだと思いますが、完全除外するのではなくて、レジリエンスを考えると最低限やることはいくらでもアイデアはあるのではないかなと思っています。STAMP のワーキングでは、制御構造図を使って安全を可視化して「どんな外乱があったらどう対処すればよいか」というのを、いろんな人が議論すると智恵が出てくる、というのを見てきました。実際の現場でも、専門家同士が専門的知識の中だけで議論しすぎて、それ以外の人の知恵が入りにくい、ということはあると思います。そういうところでの意見をお聞きしたい。

さらに一点、確率論については、ナンシー・レブソンの本では、意味がないと書かれている。事例としては、アメリカの原子力潜水艦を管理している組織では、確率論ではなくシステマチックな運用や保全をして事故は一度も起きていない、という話があります。人間やソフトウェアが絡むと確率論で議論できないので、ワーストケースで考えた方が良いのではないか、と考えている。

安全の哲学とはどうあるべきか、これからのシステム安全はどうあるべきか、ということについて、この機会にご意見を伺いたい。

（向殿）

すべてを想定してシステマチックにやろうというのは、設計では当然必要な概念で、それである程度合理的にみんなで評価できる、という風に思いますが、実はすべて想定はできないし、外乱やハザードは全部想定できないので、あり得ない話ですね。そうすると、作ったはいいけど、

実はそれは常に外部から脅威にさらされているし、中でいろいろ起きる、ということを考えて、想定外（想定除外）が起きた時にどうするかという対策を考えておくというのが一つのレジリエンスの方法で、必要な手を打っておくというのが大事であると思います。システマチックに作っておけばこれですべて終わりですよ、ということはまずあり得ないですね。

もうひとつは、人間が絡む場合は、人間には人間の良さがあって、いざという時には人間の能力を発揮できる。そういう意味では、常に、システマチックに作ったものに対して上からチェックをしながら変えていくとか運用していくとか、そういう創発的な発想が実は現実的には必要。創発だけでうまくいくかというとそうではなくて、システマチックに作っていく概念と、でも自分の限界をちゃんと知っていて現場では人間が人間の能力を発揮していい方向に持っていくというその両方が必要であろうと思います。いまやっている Safety 2.0 というのは、システマチックに機械安全を作っていったんだけど、実は人間のことはあまり考えてなかった。人間の能力はすごいものがあるが、すぐに間違えるので、「人間に任せるな」というのが一つの方針だったが、実は、いざというときは人間の素晴らしさ、発見能力を使わない手はないだろうと。安全の実現のために技術も使うけど、人間の能力も役に立つことがある、というのも踏まえていっしょに安全を実現していくという、従来の機械安全の設計から見ると、やっと機械側に対して、環境も人間も一緒になって総合的に安全を実現していきましょう、という協調安全という新しい安全の思想が可能になってきた。なぜ可能になったかというと、ICT の利用でデータ、情報を共有できるようになった。これがキーポイントで、これでやっと人間の良さを発揮できるし、機械側も人間の良さ・悪さを見ながら協調できるようになった。そういう意味では、新しい安全の哲学というか、協調安全みたいな話がそろそろ出てきてもいいのではないか、その中には、Safety II みたいな発想、いい方向にどんどんいきましょう、人間の知恵とか創発的な発想を活かそうという意味があって、その得た知恵はもう一回還元して、システマチックに作るという、そういう「行き来」が必要なのではないかという気がします。

それから、確率論というのは、ソフトウェアとか人間にはほとんど当てはまらないので、事故に遭った人にはイチゼロだけど、「何万人に使って事故は何万分の一だ」というときの話、その違いをちゃんと理解して確率論を使わないと間違った使い方になりますね。

（兼本）

いまの協調安全は、Safety 2.0 のなかにきちんと書かれているんですね？

（向殿）

Safety 2.0 というのは、どちらかというと協調安全という概念が先で、それを実現するために Safety 2.0 という ICT を使ったいろんな技術がありますよ、という発想です。協調安全のための技術的な側面のツールが Safety 2.0 になります。

（兼本）

機械側の状態とか挙動を人間が理解できるような形で作らないと、Safety 2.0 は成り立たないということですか？

（向殿）

見える化というのはすごく大事で、第三者が知恵をだすとかヒントを出すのは、見えなければできないわけです。ICT を使うと画像などで分かりやすく出すことができるので、やっと可能になってきたということだと思います。

（北村）

Safety-Ⅱは、セーフティと言わない方がよかったかもしれないという気がちょっとします。レジリエンス・エンジニアリングと Safety-Ⅱの話を、経営者、とくに、クリエイティブな考え方ができる人の前ですると、「これって安全の方法論と限定してしまうとちょっと違うように感じられます。うちの工場をマネージメントしていくための方法論のように聞こえます。」と言われることがあります。やっぱり、いいところは取って改良につなげていかなければいけないし、常に予見を行っていなければ、現状ルールだけでやっていたのでは、今の時代、ビジネスはすぐに行き詰ってしまうと思います。ビジネス分野でのマネージメントの責任は重く、ちょっと間違うと瞬時に事業そのものがなくなってしまう時代だと思います。Safety-Ⅱは、通常時と過渡時をシームレスにつないでマネージメントないし安定状態を保つ管理技術あるいはプラニング技術といえます。さらに、その中の対処技術のところがレジリエンスとよくいわれるものではないかと思っています。

それから、創発論というと、ときどき議論がこんがらがってしまうことがあります。Systemic Failure というのは、予測しなかった形で大きな惨事が起きることですね。これは Emergent（創発）といってよいと思いますが、この惨事への対策においても、Emergent する対策系があると思います。さきほど向殿先生がおっしゃったことに全く同感で、「設計は想定してやります、外れる部分は緩和系（mitigation）で対応します」ということでしょうが、でもそこまでも含めて広い意味での設計といえるのではないでしょうか。一方、人間の Emergent なレジリエント対応というのは、その範囲でもなお想定できなかったことが起きても、人間はちゃんと考えて、創発的に対策を考える、という段階が、次のレジリエンスで、モノで備えるレジリエンスと、それで間に合わなくなったときに人の創発能力で備えるレジリエンスとあります。それに期待するのは設計者の立ち位置としてはよろしくないかも知れませんが、そういう創発能力があるということは、認めてもいいと思います。緩和系は、モノで備えるのと人の創発能力という２つに分けて考えた方がよいと思います。

安全と安心についてですが、安心は結果としてもたらされるのかもしれないが、エンジニアがそれを追求することは不可能なことだと私は思っています。小松原先生がおっしゃるように、安定から、先行きそれが信頼と安心につながるというのは、その通りだろうと思います。ただ、技術屋が設計論の内側でそれを語るのは無理だろうと思っています。リスク心理学の木下富雄先生は、安全と安心という言葉を並列に使ったのは、総合科学技術会議の大きなミステイクであるとおっしゃっています。どうやってちゃんと作るかがオペレーショナルに分かっていないものを社会の目標とするといったことを、「安心・安全」という言葉で安易に使ってしまったことは反省すべきかもしれないということだと思います。これから、言葉についても技術屋は感度を持たなけ

ればいけないのではないかと思います。

確率論についてですが、機械系についてはいいかもしれないが、人間が入ってきて、最後はEmergent な安全対策みたいなものも視野に入れるなら、これは全然だめですね。確率論的安全評価をやって、そのおかげで、人間が気付かないことが見つかってよかった、という例を知りません。ハードウェア的な設計、デザイン A と B を比べるための確率論ならいいと思います。ですが、人間まで含めて、「このシナリオのときにどう対応できるか、この発電所の安全対策はこれでいいか」というような安全評価に使うのは、ちょっと違うのではないかと思います。

（兼本）

人間が Emergent な対応ができるというのは非常に大事だと思いますが、そういうエンジニアはどうやったら育てられるでしょうか？また、そういう方法論はあるのでしょうか？従来の、規格だけを教えるのではなくて、工学の原理から教えていくしかない、ということなのですかね。あるいは、さらに社会学まで教えなければいけないのか、といった点はどうでしょうか。

（北村）

社会学まで教えるのは、やりすぎではないかと思います。小松原先生と一緒に訳した本にあったのですが、レジリエンスのアイロニーという考え方があります。自動化が注目されていた時代には、自動化すると便利なのだが、重大な逆作用もある、これが自動化のアイロニーとして指摘されていた問題です。便利なブラックボックスに全部依存してしまうと、それが機能しているうちは幸せだが、それがなくなったときに、絶対大丈夫ですとは言えなくなります。レジリエンスにも類似した言い方ができます。レジリエンスを高めるために想定範囲を拡大し、対応して様々な非常用設備などを導入し訓練も高度化するとします。それ自体はいいことだと思うのですが、結局、増強されたハードに依存してしまって創発的能力を涵養する機会はさらに失われてします。これがレジリエンスのアイロニーです。

（兼本）

あるプラントでは、建設に携わったベテランがかなり残っていたので、Emergent な対応ができた、という話もあります。しかし、これが 10 年たってしまうとどうなるかは不安が残ります。

（中村）

ただ、実践している現場では、訓練と称して、想定外を含めていろんなことをやっていますね。それを恒常的にやっていくことによって、いざというときに創発的な対応を期待できるのではないかと思います。

（中村）

さきほど自動化のアイロニーの話がでましたが、この自動化では、単に人間の労働を軽減するだけでなくて、その目的の中に安全性の向上も必ず入っているはずです。自動化によって何が可能で、逆にどういうリスクが出てくるか、という分析と、そのリスクへの対応は必ずやります。

開発を振り返ると、その繰り返しだったように思います。

また、確率論の話がありましたが、その中で本質安全設計の議論が欠けているのが気になっています。本質安全設計が今はもう死語になってしまっているということです。ソフトウェアの SIL（安全度水準）で安全度を評価する時代になっていて、そこは、本質安全設計ではなくロジックの世界だけになっています。ソフトウェアが間違ってなければ良くて、それを説明するかが大事、という話になってしまっている。

我々の安全設計というのは、何かあったときには位置のエネルギーを使って、水が下に落ちて行くといった本質安全の原理を使うのが基本で、それが行き過ぎると困るからコントロールするという考え方です。そういう原理が使えないときに、初めて、確率論的に事故の可能性はどのくらいあるのかを考えなさい、とずっと教えられてきました。それを最初から確率論だけで考えてしまうのは、大事なところが喪失しているのではないかという気がします。

レジリエンスの話がありましたが、これは非常に大事な話で、FRAM にしても STAMP にしても、それをどう捉えるかというのが重要だと思います。FRAM は、複雑系を表現するにはいいけど、FRAM から、こういう風に設計しなさい、というのは出てこない。STAMP でも、創発でいろんな事故が起こる可能性があるというのを見出すのはいいのですが、それをどうやって防ぐ設計をするか、というのは出てこない。

今日、我々の周りには、いろいろなツールがあります。それをどうとらえて、総合的に何をやるのか、ということが問われているのではないかと思います。

向殿先生が協調安全の話をされましたが、非常に大事なことだと思っています。昔、装置が独立に動いていたときは、お互いが別々に動くと事故につながるということで、インターロックという概念が生まれました。最初は機械的なものでしたが、電気的なロジックになり、ソフトウェアに置き換わりました。電気的なロジックでは特別なセンサーを使ってフェールセーフにしていました。

では、いまは何ができるかというと、IoT に依拠した協調安全です。協調安全というのは、モノづくりの思想として、それぞれのコンポーネントがやろうとしていることを伝える。受け側は、それを判断する。その、情報のクローズドループによって安全を確保することが期待できるのではないかと考えています。

もうひとつ、いま勉強しているのは「本質制御」というものです。システムはどんどん複雑になっていきますが、本当に必要なものと情報は何なのか、を考えるものです。中間部分の安全を守るための装置が肥大化しつつありますが、それは必ずしもいらない。それぞれの構成要素が情報交換して、「私はこれをやりたい。やりました。」という情報を交換する。それをもとに、「そこまでやったのだったら、ここまで行ってよい。」という情報を出す。そういう関係にしていくと、複雑だった中間制御部を持たなくてもシンプルに機能が実現できる、ということが分かってきました。モノが少なくなるので安全性も信頼性も向上していく。センサーをたくさん使って制御する、という従来の考え方でなく、必須の構成要素レベルでお互いに情報交換しながら、やろうとしていることをみんなで達成していく、という方法論です。そうすると、IoT における安全責任の分散を危惧するだけではなく、むしろ、IoT をうまく使いこなすことによって複雑系の問題も解決できるのではないかと思います。まさに協調安全という新しいパラダイムの具現化、という気が

しています。協調安全の方法論に本質制御という概念もいれていくといいと思います。ちょうどいま、安全制御も面白い時代に来ているのかなという気がします。

（兼本）

まさに、それが Safety 2.0 の具体的な考え方ですね。ところで、本質制御については、抽象化モデルで考えるという理解でよいでしょうか？

（中村）

そうです。例えば、航空機の制御は、飛行機と管制のやりとりが本質で、どこにいるかというのが分かれば、そこから何 km 以内は他を入れないという制御ができます。本質制御という考え方で単純化して、そこからモノ作りをすることが大事になります。

（兼本）

その抽象化モデルでは、各要素は、システム全体の目的達成のために、自分が何をすべきかを明確に持っていないといけないと思います。

今年の IPA の WG で、独居高齢者の見守りシステムについて STAMP 分析をやりました。そこでは、独居高齢者が内側から鍵をかけると監視開始する。外出時に外鍵を閉めると監視を解除するというシステムです。監視時に生活動作が一定時間検知できない場合警備員が介護に駆けつけることになります。ところが、介護に駆けつけた警備員が、高齢者に問題がなく帰社する際に、外から鍵をかけたため、高齢者が中にいるのにもかかわらず監視が解除され、部屋の中で倒れてしまったのを見逃すという事故が起こりました。これを、STAMP の安全制御構造図で可視化し、各要素（高齢者自身、警備員、見守りシステム）の安全責任を明確に定義すると、事故の原因や見守りシステムの設計の欠陥が見えるようになります。これは、中村先生のおっしゃる本質制御と相通じるところがありますね。今後、安易な設計に起因したサービスの失敗の話はたくさんでてくるのではないでしょうか。

（中村）

お年寄りの発する情報を的確に把握し、見守りシステムのクローズドループに中にうまく組み込むことが大事ですね。今回の失敗事例では、その情報がクローズドループの中に中途半端にしか入っていなかったということでしょう。

（小松原）

今の見守りシステムの失敗事例では、新しいシステムでもあり、利用者の立場やセンサーの利用法を考えていなかったといったことがシステムの欠陥につながったと思います。設計者が、人間工学でいう人間中心設計プロセスを踏んでいないことが感じられます。

ところで、自動車は、長年の多くの経験を持つ分野だけに、安全の組み立てはよくできていると思います。まず、自動車が健全であること、次に道路状況に合わせて運転を調整する能力が必要で、その基本能力は運転者が持つ必要がある。カーナビなどの運転者を助ける行動支援技術も

ある。さらに、法体系や道路の整備も必要になります。そこまで出来て次に問題になるのは、相手車両との関係（相互のコミュニケーション）になります。相互のコミュニケーションのミスマッチは事故につながるので、各運転者が状況を共有する必要があります。ただ、そこに悪意ある車両がいることを前提にするとセキュリティに通じます。悪意ある運転は、文化や教育のような漢方薬的な処置で抑えることが必要。そして、それでも事故が起きた際を考えて、シートベルトを用意するといったことが必要になります。最後は保険でカバーすることになる。このように、安全の体系的な考え方はできているので、IoT のような新しい事例が出てきても、同じように使えるのではないか。自動車と同様に安全の思想の体系があって、それぞれの対象に応じて安全を守る方法、仕組みができてくる。

（兼本）

自動車の場合は、安全体系、レジリエンス体系が出来ているということですが、自動運転の時代になって、その体形が崩れることはありませんか？自動運転の、特にレベル 3 で、運転者と機械の安全責任の分担が確立されていないようにも思うのですが如何でしょうか？

（小松原）

おそらく体系は崩れないが、運用の基準が変わることはあるかもしれないと思います。導入の最初はうまくゆかないことがあっても、運用経験が増えるにしたがって合意形成がなされ、体系化され、安全基準や保険の調整がなされてゆくのではないでしょうか。

（兼本）

車の場合、多くの運用経験があるので保険会社が値づけできる。一方、大規模なプラントなどでは保険屋の値づけはできず国レベルで考えないといけないこともある。一方、IoT などでこれから出てくる新しいサービスでは、うまく値づけ（失敗したときの保証）ができるかどうかの懸念があると思います。難しい課題ですが、STAMP の安全制御構造図のような可視化手段も含めた分析法で値付けの合意が得られる可能性もあると思っています。ただし、今後の課題ではあります。

今後の安全のための提言

（片岡）

そろそろまとめに入りたいと思います。

今日の議論は、本質的な部分では、Safety 2.0 の協調安全の話に収斂しているように思えます。IoT や AI とかの広がりで対象とするところが広がってきた。そういう環境変化のもとで、どういう方向性で安全を考えたらいいかということだと思います。これが、協調安全的なことで解決するのか、当初兼本先生が言ったような、社会視点としての安全責任の分担や法制度まで考えてゆかないといけないのかを考えてみたい。自動車では安全体系がうまくできているということであれば、そういうものをベースとして考えていくという方向性もあるかもしれない。

（北村）

自動車の例は説得力がありますが、母集団が大きくて、ダメージを統計的にとらえられるのが前提だと思います。列車も同様。

しかし、ネットワーク系の事故や、電力ネットワークの大規模停電などはどうか。北海道のブラックアウトのような事故は、保険屋に頼んでも保証として受けてくれないと思います。

IoT のシステムでは、ネットワーク時代のセキュリティという問題で、まだ、入口に立っている段階だろうと思います。例えば、軍のセキュリティは、アメリカはだいぶ進んでいる。Google あたりも、人を集めてやっている。Google の強みはあらゆる産業分野から人が集まる。こういう分野のセキュリティとすみ分けて考えてゆく必要もあると思います。

（向殿）

自動車はもともと馬の代わりだった。馬は賢くて自分で判断して止まったりした。本来は、自動車も自分で安全を判断して止まったりすべきだったと思いますが、技術はそこまで追いついてなかった。いま、AI が出てきて、車が人間と協調安全するのは、やっとあるべき姿になってきたということだと思います。

（片岡）

自動車会社が悩んでいるのは、メーカーだけで責任をとれないことが自動運転で発生してくること。自動運転でたとえ高齢者の事故が減ったとしても、自動運転のミスで何人かは人が亡くなる可能性が大きい。そういう前提で社会合意ができるか。社会的受容性をどうやって高めてゆくかが課題になる。

（兼本）

車の安全管理の事例は、先進的事例として大変参考になるということだと思います。一方で、大規模なネットワーク災害は別に考えないといけない。

また、Safety-Ⅱの話では、マネージメントの問題として考えることが大事という話がありましたが、Safety 2.0 ではどのように考えておられるのでしょうか？

（向殿）

Safety 2.0 は、Safety-Ⅱに近いところもあり、人間の能力に期待するところが含まれています。

Safety 2.0 は、人と機械の協調から始めたが、マネージメントが入ってくると、Safety-Ⅱに近づくと考えています。マネージメントは最終的には安全にも影響する。Safety 1.0、2.0 と技術の観点から安全性向上を進めてきたが、ようやく、ICT 技術のおかげで人も取り込めるようになってきて、人と機械の協調安全ができるようになってきた。これには、マネージメントも含まれるので、Safety-Ⅱに近づいてきたということでしょう。

（兼本）

STAMP や FRAM といった方法論は、Safety 2.0 や Safety-Ⅱの中のツールとして、例えば見える

化の手法として使いうるのでしょうか？

（向殿）

ICT を使った見える化で人間の直観力に訴えるのは大事です。安全の予知や予想にもかなり使えるのではないかと思います。そこで STAMP との関連も出てくる。Safety 2.0 の基本は本質安全で、どうしようもなかったら止まれということですが、もったいないので、できるだけ動けとなり、そこに制御が出てくる。その中で安全を保つために、コンピューターと最後には人間を使いましょうということになる。

（北村）

ダメなら止まれが通用しないのが飛行機で、最後はパイロットに頼らざるを得なくなる。

（兼本）

先ほどの見える化の話で、Safety 2.0、Safety-Ⅱで、設計時点の見える化、運用時の見える化、ライフサイクルマネージメントで違いはあるのでしょうか？

（向殿）

設計でも、運用後の保全や予兆の捉え方を想定しておくのが本来の考え方ですね。運用時の安全も設計段階で考えておくのが本来の姿ではないかと思います。

（兼本）

STAMP では、STPA という設計時のハザード分析と CAST という運用後の事故分析法と二つのツールが用意されている。WG では事例が入手しにくいので CAST には踏み入っていない。でも、運用後のハザード分析には、運用環境も変わってくるので、設計時と違う視点も必要なのではないか。そのような考え方は Safety 2.0 ではどうなっているのでしょうか？

（向殿）

必ずモノは劣化するから、寿命をチェックする必要があるが、その際に見える化をどうやって行くのかが課題になるが、これも、設計時に見える化の方法を組み込んでゆくべきだと思います。

（北村）

Safety-Ⅱは、マネージメントだけでなく、安全も当然大事に考えますが、それ以外に、最近、Hollnagel 教授が言っているのは、オポチュニティ（好機）も考えるということです。これは安全とは少し違うわけですが、金融経済の不安定さの分析みたいなものもあって、人間の貪欲さをどう扱うかが問題になる。これも、Safety-Ⅱのテーマになりうるという思いはある。

（向殿）

マネージメントの本には、リスクとオポチュニティの両方を考えろと必ず書いてありますね。

（北村）
安全だけを考えてオポチュニティを考えないと、その組織は衰退するので、組織としては安全でなくなるということですね。

（片岡）
リスクを見ていくのも大事ですが、とくに IPA がやっているようなディジタルのところでは、どうやってオポチュニティを生み出していくか、ということも含めて考えていかなければいけないということだと思います。
最後に、将来方向で言い残したことがあればお教えください。

（小松原）
IoT になって誰も全体が分からないという状況が出てくる。全体が分からないときの安定、安全をどうするかは大きな課題になる。例えば、防犯カメラがネットにつながってしまうと何が起こるか、そのリスクに気がついた時には、手を打とうとしても手遅れといったことがある。これは、責任分担以前の問題でどうしたらよいのでしょうか？

（向殿）
社会的合意が大事なのではないでしょうか。情報開示して、「こういう時にはこうなるけどいいか？」と聞いて合意をしてゆくのが大事なのではないか。合意をとりながら少しずつ進むのがいいのではないか。新しい技術を早急に進めると何が起こるか分からない。

（北村）
ビジネスの立場で考えると、立ち止まれないこともある。自動運転車を作れと言われると、危ないから作れませんとは簡単には言えない。なので、組織としての対応が必要と思う。工学倫理という言葉だけで片付けるのは好きではないが。
便利さを追求するあまり、副作用への対策がおろそかになったという経験が多くあるわけなので、そろそろ、新しい技術への取り組みをするさい、便利さだけの追求ではなく、副作用を慎重に考える文化が出来てほしい。

（向殿）
便利さだけでなく、不便さを楽しむくらいの余裕があった方がいいのではないでしょうか。

（兼本）
便利さと安全のトレードオフの社会的合意は大事だと思いますが、北村先生のような体験が社会全体に行きわたっているわけではない。こいうった話題に対しては一般の人はマスコミを通してくらいしか知るすべがない。企業も本当に困った問題は外に出さないし、産業界の分野ごとに安全に関する考え方はかなり違うと思います。このような話題を学会レベルでも議論し、社会実

装する場はできないものでしょうか？

（小松原）

「科学技術社会論」というのがありますね。東大などの先生を中心に議論されています。科学技術のメリットを、科学者と市民の間で合意を図るにはどうしたらよいかという研究かと思います。科学者は技術のベネフィットを主張するが、市民は技術に関しては不信感しか持っていないので、合意までに何十年もかかってしまう。また完全な合意はほとんど不可能の気もします。

（兼本）

大規模な開発と、コンシューマ対応の個別の開発では違うかもしれませんね。社会技術は大事だけれども、一般市民と議論すると技術の詳細な話が出来なくなってしまう可能性もある。

（小松原）

科学技術を、自動運転のような何百万台もあるものに使うか、社会インフラのような大きく一つしかないような問題に使うかで違うかもしれませんね。

（北村）

科学技術社会論は大事ですが、その議論の中に、科学者と社会学者はいるが、技術者はいないという問題もあります。技術というのは理念だけで考えて行けるものではない。技術者は常に技術の長所と短所の折衷論を考えているわけだが、それを科学者や社会学者と議論できるコミュニケーションリテラシーをある程度はもたないといけない。

（兼本）

確かに、科学と社会は分かりやすいが、技術が議論に入ってこない。

（北村）

物理学者や社会学者は、批判的にものごとを論じる。それももちろん重要だが、それだけではモノは作れないという側面がある。モノを作るにはどこかで妥協しないといけない。

（片岡）

技術屋は、会社の制約に縛られること批判的になれない。技術者は自分の技術が生かされることに喜びを感じる。でも、技術屋は、社会とコミュケーションをしてくることが少なかったといえますね。でも、そういうコミュニケーションを活性化する場があってしかるべきかもしれませんね。そこで、技術のいいところも悪いところも社会にわかってもらうことが大事だと思います。

（北村）

技術者は自分たちの強いところをもっていないといけないが、社会との接点も持っていないといけないと思います。技術者は技術だけで成果を出してゆきたいのでしょうが、やはりコミュニ

ケーションも必要ということでしょう。

（中村）
技術者はそれぞれの領域では自信を持っているのでしょうが、異なる分野の技術屋どうしが議論することが少ない。

（向殿）
安全の世界では、自分の専門と違う分野を学ばない人が多い。それで、「日本安全学教育研究会」というのを作ったんですよ。そこでいろんな分野の安全を学ぶことが出来る。

（北村）
欧米の安全屋は、複数の分野の知識を持っていますね。

（兼本）
技術屋は結果で議論することが多い。「いいものができました。」で終わって、なんで良いものでできたかという説明がない。しかし、社会はプロセスを見ることが多いが、そこを技術者は語るスキルがない。

（北村）
そこは大事なところで、技術屋は結果で語れる。ウォークマンを作ったら売れるし、スマホを作ったら売れる。それで社会を変えることが出来る。だから、言葉を尽くして語る必要がないという歴史的事情があった。なので、言葉を尽くして説明するという訓練がない。

（兼本）
それで事故を起こしてしまうと社会からたたかれて、一挙にその技術が社会から消えてしまう。なので、先ほど出たように、コミュニケーションを尽くして理解をしてもらう努力を、技術者も入って行うことが大事ということですね。

（兼本）
おおよそ話は出たと思います。いろいろ大事な話がでましたが、今日の話をどういう形でまとめるかという点で意見を伺いたいと思います。今日の話を、そのままは難しいのですが、できるだけ生に近い形で議事録としてまとめ、良いところを読者に読み取ってもらうので良いかと思っています。
結論としては、Safety 2.0 や Safety-Ⅱは融合していく、という話は大事だと思います。また、科学技術社会論も大事で、安全学教育研究会で異分野の安全技術者の交流を通じて、その普及を図ってゆくというのも大事なまとめになると思います。

（中村）

ノーバート・ウィーナーがサイバネティクスを過去に提唱した。当時は、フィードバックループしかなかったわけですが、今は、IoT、IT でより高度な情報が得られ、コミュニケーションを高度にできる。それによりが、より信頼できるシステムが可能になるにすることが出来るが、これも大事なまとめになるのではないでしょうか。

（北村）

中村先生は情報の大事さを、小松原先生はコミュニケーションの大事さを訴えられましたが、私は意図の共有の大事さを強調したいと思います。意図が同じであれば、異なる手段でそれを達成しても、それを許容しようということです。安全については、「意図はこれなんだから、ベストはできなくてもネクストベストを許容する」という考え方が必要だと思います。こういった考え方で複雑システムをある程度整理できるのではないでしょうか。意図が同じであれば、広い意味での安定な状態を維持できるということで、レジリエンスの考え方にも通じます。

（北村）

従来は、PID のフィードバック制御だったが、IoT ではもっと高度なフィードバックが可能になるということですね。

（兼本）

社会システムを制御しようとすると、変なループで混乱に陥れる可能性もありませんか？

（中村）

社会システム全体を制御しようとするとそうかもしれないが、それを構成する個々のシステムの在り方についてシステム全体の意図に沿って制御すれば混乱をすることはないと思います。

（小松原）

少し話がそれますが、落語やコントがなぜ楽しいか。あれは、最初に、芸人と聴衆とで場（状況）を共有することが大事なのですね。最初の場を省いて落語を聞くと全く面白くなくなる。留学生が落語を楽しくないといっていたのですが、それは、状況を共有できない、落語の背景にある日本文化を知らないということによるようです。

意図を共有する、状況を共有する、というのは非常に重要ですが、複雑システムの安全制御でもこういった考え方は大事になるのだと思います。

（北村）

いま興味を持っているのは、クロスロードというゲームツールです。阪神大震災の経験をもとに作ったもので、どういう意思決定をしても困るという状況を提示して、判断を迫るというゲームです。ここでは、小松原先生の話で出た基本的な状況認識の共有、価値判断の共有が大事になります。判断できない状況でも判断しなければいけないというゲームですが、それで終わりではなく、そこからゲーム参加者の相互の議論を始めるというのが大事になります。設計で入れてお

くべき能力と、人間の対応能力に依存する部分を議論することつながります。技術者の柔軟な発想能力を育てるためにも大事なことではないか。クロスロードは簡単なゲームですが、やればやるほど面白いという人がいます。

今日出た STAMP、レジリエンス、Safety 2.0 といった技術ツールと、それらを使いこなす人間をどうやって育成するかという問題もパッケージとして考えてゆかないといけないのではないでしょうか。

（兼本）

安全の教育では、技術論だけではなく、価値判断を持った意思決定まで含めて教えないといけないということですね。

（向殿）

安全は、技術だけでなく、社会もかかわる。安全は、価値観が含まれるから難しいのですが、これから頑張ってほしいことです。

（北村）

今日出たように異分野融合は大事ですね。

（片岡）

今日の話では、そこに人間性もかかわる。文化や価値観に立ち戻って考えなければいけないということですね。

本日は安全に関する技術論、社会論、教育論など、いろいろな刺激を受けました。今後の活動に役立ててゆきたいと思います。どうもありがとうございました。

以上