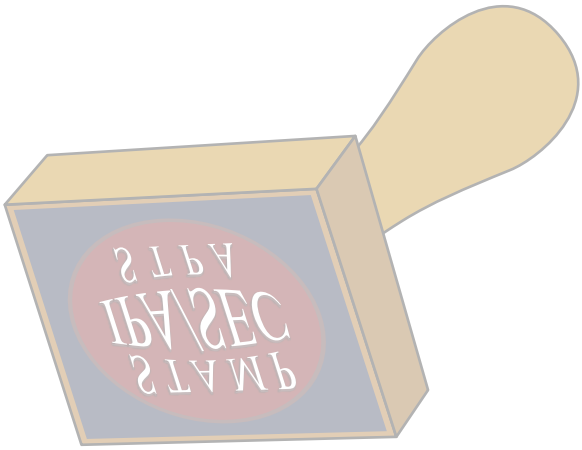
IPA Better Life with IT

はじめてのSTAMP/STPA（活用編）

～システム思考で考えるこれからの安全～

独立行政法人 情報処理推進機構
Information-technology Promotion Agency, Japan (IPA)
技術本部ソフトウェア高信頼化センター
Software Reliability Enhancement Center (SEC)
ソフトウェア高信頼化推進委員会
Software Reliability Enhancement Promotion Committee
IoT システム安全性向上技術 WG
IoT System's Safety Enhancement Technique WG

Ver.1.0
2018年3月


STPA
IPA/SEC
STAMP

STAMP
IPA/SEC
STPA

はじめに

本書は、独立行政法人情報処理推進機構技術本部ソフトウェア高信頼化センター（IPA/SEC）の IoT システム安全性向上技術 WG における 2017 年度活動成果をまとめたものである。2015~2016 年度に作成した「はじめての STAMP/STPA」入門編 [IPA2016]、ならびに、実践編 [IPA2017] は、これから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いるが、さらなる産業界への浸透を期待して、「活用編」という形で WG での活動成果をまとめた。

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（モノのインターネット）、AI（人工知能）、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んだ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。このような背景から、マサチューセッツ工科大学（MIT）の Nancy G Leveson 教授は、「旧来の安全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近年の複雑システムの安全解析には適していない」と指摘し、新しい安全解析手法として STAMP/STPA（Systems-Theoretic Accident Model and Processes：システム理論に基づく事故モデル/System Theoretic Process Analysis：システム理論に基づく安全解析手法）という方法論を提唱している。そしてその方法論は、米国において、ブラントや宇宙航空分野などで既に有用性が実証されている。しかしながら、日本の産業界でこのような新しい安全解析手法の理解が深まっているとは言えないのも現実である。今後ますます増加してゆく「人と高度ソフトウェアを含んだ複雑システム」の安全性を高めてゆくには、STAMP/STPA の使い方の理解だけではなく、その背景にある安全制御や安全論証の考え方のパラダイムシフトも理解しておく必要がある。これからの複雑システムでは、その構成要素の全ての故障を明示化し、低減するという従来の信頼性工学的手法だけでは限界があり、事故を回避するための制御行動の乱れを分析し、それを防ぐという安全制御工学的手法と組み合わせて安全性の向上を目指す必要がある。いわゆる「虫の目・鳥の目」の組み合わせである。

このような背景から、本書では、前述の初級編・実践編に加えて、活用編と称して、鉄道や車などの産業界での試行事例、人と機械の協調による安全制御の事例、セーフティとセキュリティの統合分析事例などをまとめた。また、これらの事例分析を支援するために IPA/SEC で本年度に開発した STAMP 支援ツールの紹介もしている。さらには、STAMP/STPA を越えて、将来の複雑システムの安全解析の在り方に関するビジョンの提言も行っている。

本 WG の活動がきっかけとなって、2016~2017 年に、2 回にわたる STAMP ワークショップが開催され、多様な産業界からの参加を得た。そこで、多くの参加者が新しい時代の安全性について悩み、また、STAMP への期待をしていることを目の当たりにした。本「活用編」を、今後の複雑システムの安全性向上に役立てて頂ければ幸いである。

目次

図表目次

1. 概　要

2012 年にマサチューセッツ工科大学（MIT）の Nancy G Leveson 教授がその著書 Engineering a Safer World［Leveson2012］の中で現代の複雑システムの新しい安全解析手法として STAMP/STPA を提唱し、現在もまだ手法の改善を An STPA Primer［Leveson2013］の改訂という形で継続している。米国では年々活用範囲が拡大してきており普及の段階から実用の段階に入ったと考えられる。一方日本の産業界では、まだこのような新しい安全解析手法が広く認知されていない状況であったため IPA では、WG（現システム安全性向上技術 WG 以下 " 本 WG"）を設立して STAMP の調査を米国にて実施し、STAMP 手法に関する調査報告書として 2015 年 8 月に公開した。この報告書は、医療機器や航空機などの STAMP による研究の実例において、アプローチや理論構成などを深く分析することで、その有効性や今後の事故モデルのあり方の検討に役立てることを目的としている。

しかし、手法の紹介だけでは、産業界に根付かせることは難しい。その要因として STAMP/STPA の基本的な考え方の理解が不足しているということと、現実の問題への具体的な応用方法が分からないという 2 点が指摘できる。そこで IPA では、2016 年 3 月に入門書として「はじめての STAMP/STPA」［IPA2016］を小冊子に纏めて配布した。小冊子では、我が国における STAMP/STPA のエキスパートによる解説をつけるなど、STAMP/STPA の初学者にとってわかり易い入門解説として利用されている。この中で、適用事例研究（教科書に近い応用事例）は、STAMP/STPA 初学者にとって、IPA でシステムの安全性について検討している本 WG における入門的適用経験を追体験して共有することにより、STAMP/STPA の理解と修得に有益なものになっている。

さらに、産業界において実適用を促進するために 2017 年 3 月に「はじめての STAMP/STPA（実践編）」［IPA2017］を同じく小冊子に纏めて提供した。この小冊子では、STAMP/STPA を実際のシステムに適用しようとした際に悩むであろうポイントにできるだけ解決のヒントになるよう産業界でのニーズを考慮した多様な事例についての安全解析を試みた。ここで取り上げた事例は、いずれも、教科書で例示されているような標準的な制御構造とは異なっており、定石通りに分析を進めるだけではこれらの事例の安全化を達成できないことを知ることになると思われる。こうした「定石通りではない分析」を容易に、かつ STPA の分析手順の中でハザード誘発要因（HCF:Hazard Causal Factor）を抽出する部分をよりクリエイティブに行うためのツールとして、ヒントワードの検討や米国で提案されているアーキテクチャ分析設計言語 AADL（Architecture Analysis and Design Language）との統合方法を提案している。

ここまで述べてきたように、新しい安全解析手法 STAMP/STPA 普及の先導役として、「紹介」、「入門」、「実践」と階段を登ってきた。ここで、もう一段の進化、即ち「理解する」から「やってみる」そして「当たり前にやる」ために、小冊子として纏めた。本書には、産業界での実施事例を中心に STAMP/STPA による定性的分析からシミュレーションによる定量的分析に繋げた事例、セキュリティ分析への応用事例を解説している。さらに、STAMP/STPA の先にある新しい安全理論としての Safety2.0［日経 BP2015］の実現に向けて FRAM［ホルナゲル 2013］をはじめとする安全解析手法についても解説している。

本書の内容は、目次に示す通りである。

2 章の活用事例では、4 種類の事例を取り上げて解説している。

STAMP によるクローズドループ型踏切制御システムの安全性評価の事例では、既存の鉄道踏切制御システムの課題を整理した上で、その解決方法として踏切制御装置と列車上装置が情報交換を行いながら制御を行う新たな制御式が有効であることを STAMP/STPA の解析によって示すとともに、得られた非安全なコントロールアクション（Unsafe Control Action）と

ハザード誘発要因を過去の踏切事故統計データと突き合わせることによりリスクに対する対策の効率性を評価している。

二つ目の適用事例である、二輪倒立ロボットの人・機械協調制御の事例では、コントローラーと機械間の自立制御指示と人とコントローラー間の移動指示のコンフリクトによる転倒分析、シミュレーションによる調和策の定量評価を二輪倒立ロボット制御に適用について解説している。STAMP/STPA 自身は定性評価のための分析手法であるが、分析結果とモデルベースの定量シミュレーションを組み合わせることにより定量評価まで可能にした事例である。

三つ目の適用事例である、一般社団法人 JASPAR（Japan Automotive Software Platform and Architecture）による自動車（ドライバーと車両に搭載される一つの制御システム）への STAMP/STPA 適用事例は、JASPAR 内で先駆的に活動しているメンバー（STAMP/STPA 活用ガイド開発チーム）が仮想的な EPB（Electric Parking Break）を取り上げて STAMP/STPA で分析できることの確認と安全要求導出の説明補強（安全性論証の強化）のツールとして使えることを確認したものである。さらに分析だけでなく安全要求仕様を抽出することができることも確認できた。また STAMP/STPA と自動車機能安全規格 ISO 26262 の安全解析との比較、STAMP/STPA 活用に際しての留意点を整理している。これは、自動車分野以外の産業分野における安全規格との整合を考える上で参考にしていただけるであろう。

次に、セキュリティ分析への応用の海外事例として、米国における広域送電網とローカル送電網の接続に関して安全性とセキュリティを統合して分析するための STPA 拡張手法である "STPA-SafeSec" を使って行った文献について解説している。機能を表す CSD（Control Structure Diagram）にさらに物理 CSD を追加することでセキュリティ上の脆弱性を明示化し安全性への影響を評価している。さらに、STPA Step 2 で HCF を特定する際に具体的なセキュリティ侵犯手段を想定し易くしている。基本は、システムの安全解析であるが、安全を阻害する要因（ハザード）が人為的に起こされる可能性について行う際に有用となる関連事項を併せて紹介している。

3 章では、STAMP ワークショップについて紹介している。

本 WG の活動をきっかけに 2016 年 12 月に第 1 回 STAMP ワークショップを開催したが、2017 年 11 月に第 2 回を東京で開催している。参加者は 4 か国 181 名、講演数も 28 件、ポスター発表 1 件、登録企業数は 110 社に達し、確実に普及が進みつつあることが伺える。また、MIT では、2017 年 3 月に第 6 回 STAMP ワークショップが開催され、参加者は 24 か国から 275 ～ 300 名（昨年度より 12%増）うち日本からは 15 名であった。講演は 20 程度の産業分野から 32 件、ポスター発表は 9 件で、日本からは講演 1 件とポスター展示 2 件であった。方や欧州でも第 5 回 ESW（European STAMP Workshop）［IPA2018-2］が 2017 年 9 月に Reykjavik で開催された。参加者は 16 か国から 65 名、日本からは 4 名であった。講演は、15 産業分野から 22 件、ポスター発表は 3 件、うち日本から 2 件の講演があった。特筆すべきは 3 大学で教育講座が開設されており産学が協調した取り組みがなされていることである。

4 章では、STAMP/STPA を適用する際のツール支援に着目し、STAMP/STPA 適用時の分析結果の様式や課題について述べ、合わせて IPA で提供する STAMP/STPA 支援ツールを紹介している。

STPA はいくつかの単純であるが手間のかかる作業を必要とする。例えばコントロールストラクチャー図の記述では、コンポーネント間の接続にコントロールアクションを対応させたり、コンポーネント内部にプロセスモデルを記述したりといった STAMP 特有のデータ構造を記述する必要がある。また、コントロールストラクチャー図中のコントールアクションと非安全なコントロールアクションを抽出する工程で分析するコントロールアクションの対応付けも必要である。図表の解釈やデータ連携を人間が適切に補うことで、汎用的な図表作成ツールを用いて STPA を実施することになる。IPA で開発している支援ツールの目的は、基本的な

清書機能、分析手順ガイド機能に加え、分析者が分析に注力できるよう支援する機能を提供することである。特に反復しながら分析を進める際に発生する手間（図表変更とその変更が引き起こす修正作業等）から分析者を解放することを目指している。

5 章では、今後の複雑システムの安全解析として、Safety2.0 について説明している。これまでの安全に関わる規格の流れ、複雑システムに対する現在の規格の限界、人と機械の協調制御、抽象化した " 本質制御 " に着目する等、将来の安全設計の在り方を説明している。" 本質制御 " とは、システムを構成する上で必須とされる要素が、相互に情報交換を行い、必要とされる機能を実現する形態と定義され、既存の制御装置の省略も可能とすることにより経済性を同時にシステムの信頼性 / 安全性 / 保全性を向上させるシステム設計概念である。

6 章では、自動運転をはじめ人工知能を搭載した不特定多数のエージェントが巨大な IoT システムを構成しつつ自由に行動する現実・仮想の空間を如何に安全なものしうるか、という課題について新しい安全理論である Safety2.0 を最上位の目標ととらえて今後の安全解析の在り方について述べている。ここで東京駅のコンコースがいかにして安全に保たれているかを機能共鳴解析手法（FRAM：Functional Resonance Analysis Method）を用いて制御の階層構造の特殊性から成功要因やリスク要因を導くことで事前には分からなかったハザードやハザード制御機能を識別できることを示している。

今後の人と高度なソフトウェアが一体となった複雑システムの安全確保には、実装に先立って十分な安全性分析と設計への反映が重要である。そのための手法として STAMP/STPA が有効である。活用に当たっては、システムの要求仕様、前提条件を教科書の基本に忠実に守って分析するだけでなく、システムの要求仕様、前提条件そのものの見直しまで踏み込んだ安全設計を目指していただくことにより真の安全性を確保されることを願っている。さらに次世代の人工知能を搭載したエージェントを含む複雑システムの安全確保のために、新しい安全理論である Safety2.0 の実現に向けて FRAM をはじめとする安全解析手法にも取り組んでいただけると幸いである。

2.1. STAMP によるクローズドループ型踏切制御システムの安全性評価

2.1.1. はじめに

踏切道改良促進法等の一部を改正する法律が平成 28 年 3 月 31 日に成立し、危険な踏切道や渋滞の原因となる踏切道について、国土交通大臣が指定を行い、道路管理者・鉄道事業者や地域の関係者が連携して、具体的な対策を検討する仕組みとなった。

平成 18 年以降、自動車等が関係した踏切事故の発生件数は減少傾向にある。しかし毎年 200 件以上が発生しており、そのほとんどは踏切警報機等の保安装置が整備されている第一種踏切道にて発生している。

この状況を踏まえると、踏切制御の安全性向上のために装置側での技術開発の余地があることを示しており、本質的な検討が望まれる。本節では、既存踏切制御システムの課題を明らかにし、その解決方法として踏切制御装置と車上装置が情報交換を行いながら制御を行うクローズドループ型が有効であることを STAMP に基づく解析によって示す。その際、定性的な安全性評価をする STAMP に対し、安全性の定量的評価ができるように方法論を拡張する。具体的には、過去の踏切事故統計データを STAMP の解析結果に適用し、安全性の定量的評価を行えるようにした。この拡張 STAMP を用い、既存踏切制御システムと比較しクローズドループ型踏切制御システムの有効性を評価する。

2.1.2. 比較対象システム

安全性評価の対象とする踏切制御システムの制御原理を次に紹介する。

（1）既存踏切制御システム［鉄道 2015］

制御が複雑な単線区間における踏切制御システムを図 2.1-1 に示す。踏切道の遠方に短小軌道回路を用いた上り列車用と下り列車用の始動点踏切制御子が、踏切道を挟んだ両側に配置される。この区間に列車が進入すると、警報を開始するが、列車の進行方向（上り / 下り）に応じて、踏切道を通過後に現れる反対側の踏切制御子に対しては、その機能をマスクする必要がある。終止点踏切制御子は、踏切道の鳴動を停止し遮断竿を上げるためのものである。このほかに、遮断後に取り残された自動車などの障害物を検知する装置や、障害物検知時に運転士に伝え、いち早く停止操作を促すための踏切支障報知装置等が設置されている。

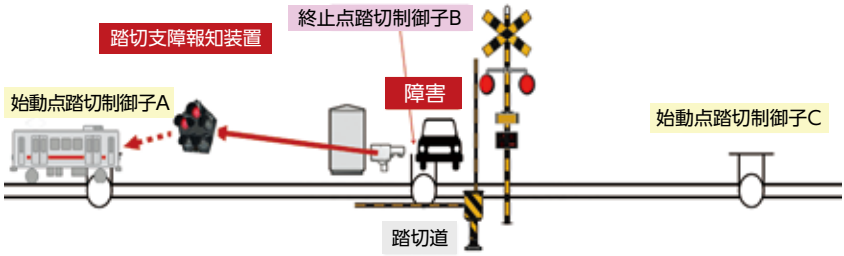


図 2.1-1　既存踏切制御システムの構成

このように、既存踏切制御システムは、地上の踏切制御装置で列車を検知し、処理を行い、

万一危険なときには踏切支障報知装置により乗務員にその旨を伝えるもので、地上装置で処理が完結している。万一、障害物を検知しても、踏切支障報知装置の情報を乗務員が気づかねば衝突事故を防止することはできない。また、列車速度にかかわらず、列車が一定地点に進入したときに制御を開始するため、鳴動開始から列車が到達するまでの時間に大きなばらつきを生じるといった課題もあった。

この弱点を克服し効率的な踏切制御を行うには、列車の位置だけでなく速度を含めた運転状況に応じて踏切道を制御することが有効で、車上と踏切制御部が情報交換を行いながら処理を行う、クローズドループ型の踏切制御方式が期待される。ここでは、クローズドループ型の踏切制御方式の一例として集中一括制御方式を紹介する。

（2）集中一括制御方式

集中一括制御方式は、DMV（Dual Mode Vehicle：道路とレールの両方を走行可能な新しい形態の交通機関）の開発時に提案された方式で、図 2.1-2 に示すように、センター処理装置での列車位置情報管理に基づき踏切制御装置を制御する。

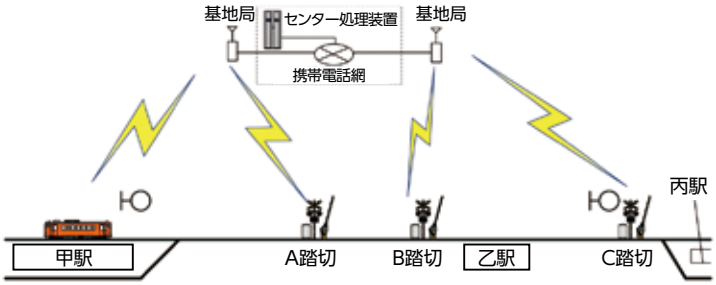


図 2.1-2　集中一括制御方式による踏切制御システム

集中一括制御方式では、例えば図 2.1-2 に示す甲駅を出発する列車が丙駅まで運行するにあたり、駅間に存在する踏切道（A，B，C 踏切道）の踏切制御装置に対し警報開始のタイミングを指示する。なお、警報開始時刻は踏切道を列車が遮断機の無遮断状態で通過することを防ぐために列車が線区で定められた最高速度で走行した条件で算出する。列車の出発許可は、全踏切制御装置からの受信応答が確認されることが条件となる。

個々の踏切制御装置は、伝達された警報開始タイミングになったら警報開始を行う。

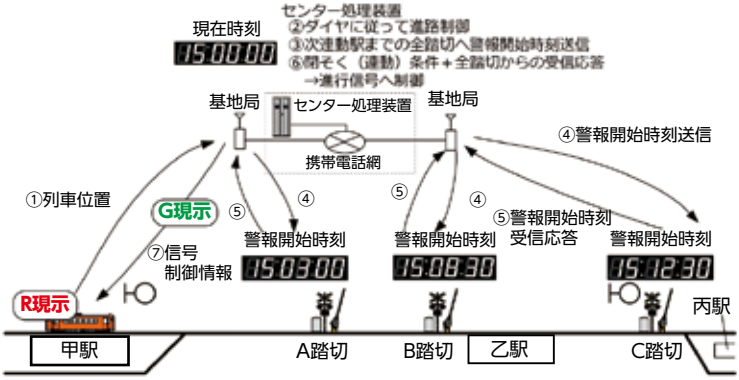


図 2.1-3　警報開始時刻の設定

列車の現在位置と走行速度など走行状況により警報開始時刻も変化し得る。センター処理装置は、現在列車位置、速度情報に基づき、各踏切道に対して警報開始時刻の更新情報を送信することで、常時最適な警報開始タイミングが与えられる。なお、踏切制御装置は、警報開始時刻のタイミングになると警報を開始し、一定時間後に踏切道を遮断、障害物が無いことを確認するとセンター処理装置に通過 OK を送信する。この情報を受けてセンター処理装置は列車に対し、踏切道を越えた走行可能地点を探索し車上装置に送信する。

なお、当該列車が踏切道を通過したことをセンター装置が知得すると、踏切制御装置に対して警報終止制御をする。

次に、このクローズドループ型踏切制御方式の安全解析及び評価について検討する。

2.1.3. STAMP によるシステム比較

2.1.3.1. 既存踏切制御システムに対する STAMP 解析

図 2.1-1 を用い、単線区間の点制御式の踏切システムについて説明する。列車検知に短小軌道回路式の踏切制御子を使用している、始動点の踏切制御子が列車の進入を検知すると、踏切道の鳴動を開始し、一定時間後に踏切遮断竿を降下させて踏切道を遮断する。列車が進入し、終止点の踏切制御子で列車を検知すると、鳴動を停止し遮断竿を上げる。なお、遮断完了後に障害物を検知したなら、特殊信号発光機により乗務員に告知して列車を停止させる。このシステムにおいては、乗務員が発光に気づくのが遅れ、衝突するといった事故がしばしば報告されている。このように既存踏切制御システムは、車上装置とは関係なく地上側のセンサーと踏切制御装置の間で制御が行なわれ、万一の際には乗務員の注意力に安全性を委ねているのが特徴である。

既存踏切制御システムについては、文献［IPA2016］に STAMP による解析事例があるが、解析は遮断までで終わっている。実際には、遮断完了時に障害物があるにもかかわらず、乗務員への伝達もしくは乗務員の認知が遅れ、事故につながるケースも多いため、本節ではこのケースも踏まえて解析を拡張した。

拡張分のケースについて登場人物を整理し、コントロールストラクチャーを図 2.1-4 のように定義した。

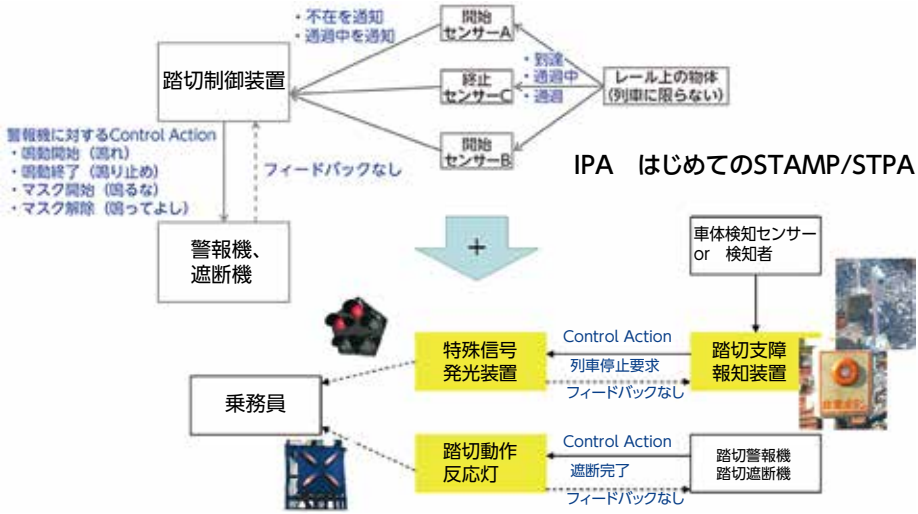


図 2.1-4　既存踏切制御システムのコントロールストラクチャー

この解析結果によると、文献［IPA2016］に抽出された 6 つの UCA（UCA1 ～ 6）のほかに追加分の 6 つの UCA（UCA7 ～ 12）で計 12 個の UCA が抽出された。抽出された UCA は表 2.1-1 の通りである。

表 2.1-1　抽出した UCA 一覧

UCA ID	UCA
UCA1	警報が鳴らずに列車が踏切道を通過する。（踏切が閉まらない）
UCA2	遮断終了する前に列車が踏切道に到達する。（閉まるのが遅く、間に合わない）
UCA3	列車が踏切道を通過完了する前に鳴動停止する。（閉めた後、開くのが早すぎる）
UCA4	列車が来ないのにマスク指示し、警報鳴動しない。反対側の開始センサーにもマスク指示し、警報鳴動しない。
UCA5	開始センサーへのマスク指示が遅れ、列車の当該センサー通過に間に合わないと、マスク指示が残り、対向列車が 2 本続いたときに警報鳴動しない。
UCA6:	列車が反対側の開始センサー通過後までマスク指示し続けると、対向列車が来ても鳴動しない。反対側センサーにマスク解除指示が出ず、対向列車が来ても鳴動しない。（マスク指示後に列車が引き返す場合を含む）
UCA7	障害物があるのに発光要求が出ず特殊信号発光機が点灯しない。
UCA8	遮断が完了していないのに遮断完了が出て踏切動作反応灯が点灯する。
UCA9	発光要求が遅れて特殊信号発光機の点灯が遅れる。
UCA10	発光信号認知からブレーキ操作までの時間が遅れる
UCA11	遮断完了後に、障害物を検知し発光信号機を制御したが、タイミングが遅れブレーキ制御による制動距離を確保できない。
UCA12	遮断完了している踏切道に列車が通過中にもかかわらず、障害物が進入した。

また、それぞれの UCA の原因となる Causal factor は文献［IPA2016］に挙げられた 17 個と追加分の 10 個（詳細は割愛）を合わせ全部で 27 個挙がった。なお、UCA11, UCA12 については、踏切道横断者側の要因となるためシステム要因となる Causal factor の抽出はしなかった。

この解析は、既存踏切制御システムへ行ったものであり、これらの事故回避の最終的な手段は乗務員に委ねられている。このことが、遮断完了時に障害物があるにもかかわらず、乗務員への伝達もしくは乗務員による認知が遅れ、事故につながるケースが多いことに繋がっていると考えられる。

2.1.3.2. クローズドループ型踏切制御システムの場合

（1）クローズドループ型踏切制御システムの評価結果

はじめに、コントロールストラクチャーは、登場人物を整理し図 2.1-5 のように定義した。

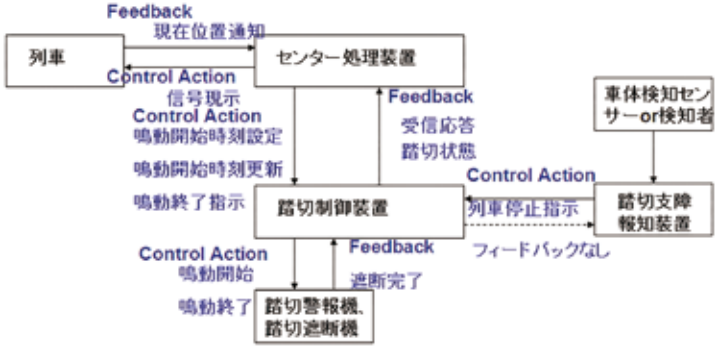


図 2.1-5　クローズドループ型踏切制御システム（集中一括制御方式）

道路と線路の交差部での自動車と列車の衝突事故を防護するためのシステムで、物理的に衝突が起きるが、回避するために必要となるのが本来考えるべき道路と線路の交差部における事故回避システムである。この場合制御装置は、現在の状態に応じて列車または、自動車に対して衝突を回避する制御を行なうものであるため、コントロールストラクチャーは図 2.1-6 をベースに考えればよい。

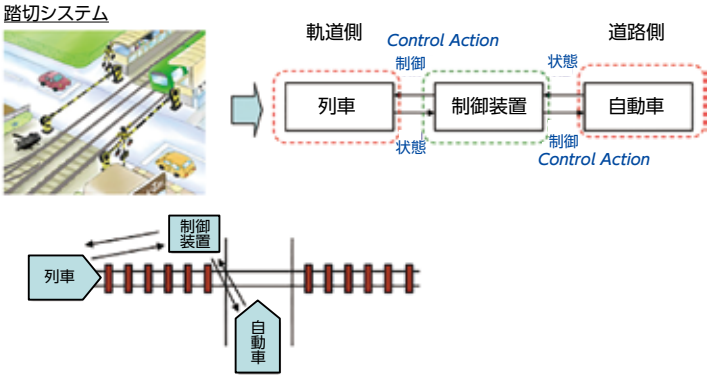


図 2.1-6　事故回避システムのコントロールストラクチャー

図 2.1-6 をベースに各踏切制御システムに対して定義したコントロールストラクチャー（図 2.1-5）を整理すると、踏切制御システムは、自動車に対して直接的に制御を行なわず踏切遮断機、踏切障害物検知装置に事故回避対策を委ねていることが、また図 2.1-4 を対象に整理すると既存踏切制御システムは列車に直接的に制御できないので、列車に対する制御に対しての事故回避対策は乗務員に委ねられることになることが分かる。

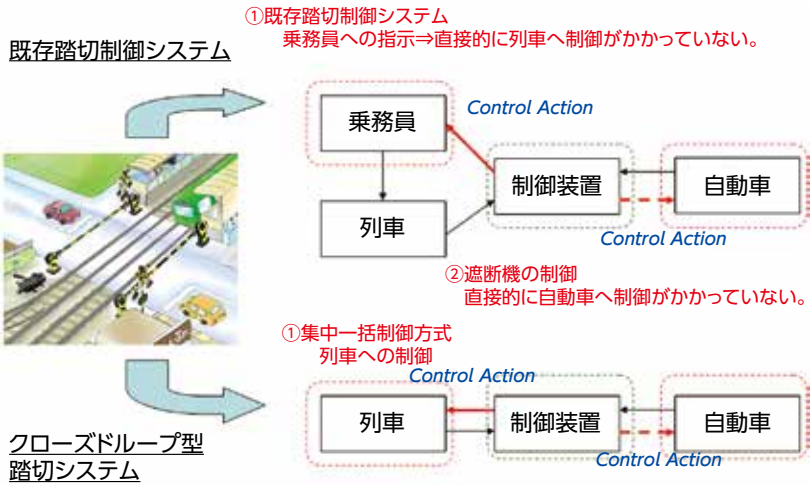


図 2.1-7　事故回避システムの分析結果

次にクローズドループ型踏切制御システムについては、7 つの UCA が抽出された。なお、抽出数には、既存踏切制御システムと同様に制御遅れとして踏切道横断者側の要因となる 2 つの UCA を含んでいる。既存踏切制御システムでは、始動点 2 箇所と終止点 1 箇所の制御子の列車検知条件をもとに制御を行なうため、単線区間での上り下り列車の区別を行ない反対側の始動点の検知を無効にする仕組み（マスク処理）が必須であった。これに対し、クローズドループ型踏切制御システムでは列車の動きに対応してセンター処理装置が制御を行なうため、このようなマスク処理は不要となる。したがって、このマスク処理アクションに対する UCA の抽出も不要となる。

最後に、11 個のガイドワードをもとに分析し既存踏切制御システムと同様に事故シナリオを導出すると UCA の Causal Factor は 19 個（詳細は割愛）となった。

分析した結果について、ここでは対策には言及しないが、既存踏切制御システムの場合は、各種センサー（制御子）の検知結果に基づき受動的に制御を行なっているため、UCA の数が多くなり予想される事故に至るシナリオが多岐に渡ることが分かる。

これに対して、クローズドループ型踏切制御システムの場合は踏切支障報知装置を除いては、制御結果を常に監視しセンター処理装置と車上装置間でクローズドループが確保される能動型の制御のため、予想される事故に至るシナリオは制御実態がある踏切制御装置、踏切警報機、踏切遮断機に限定される。

このため、事故シナリオは時刻管理などの制御アルゴリズムに限定され、それ以外の不具合は列車が踏切道の前で停車する安全側への遷移となる。また、制御アルゴリズムは FS-CPU による異常監視処理に基づくため、危険側故障確率は低く抑えられる。

クローズドループ型踏切制御システムでは、踏切制御装置は遮断完了後、現場状況（踏切

道内に障害物がないことなど）をセンター処理装置経由で車上装置に通知することにしている。車上装置は、安全が確認されない限り、踏切道を越えての走行を行なわない。安全性上不可欠なこの機能を前提にした上で、列車にブレーキがかからず、しかも、踏切の鳴動時分を確保する伝送タイミング等の最適化が求められる。

（2）STAMP による評価の有効性

運輸安全委員会が調査した踏切障害事故（2001 年 10 月から 2016 年 7 月公表分まで）としては、68 件の報告がある。このうち、障害物検知装置が設置されていた 17 件の要因別事故発生確率は以下の通りである。

・踏切道内停滞で特殊信号発光機動作しないもの：11%
・踏切道内停滞で特殊信号発光機動作したもの：23%
・直前横断（踏切道横断者側の問題）：47%
・側面衝突（踏切道横断者側の問題）：17%

この発生確率を抽出された UCA について発生確率を割り振り定量的な評価を行うと、実際には発生確率が低い事項と、注視すべき事項に UCA が分類できる。

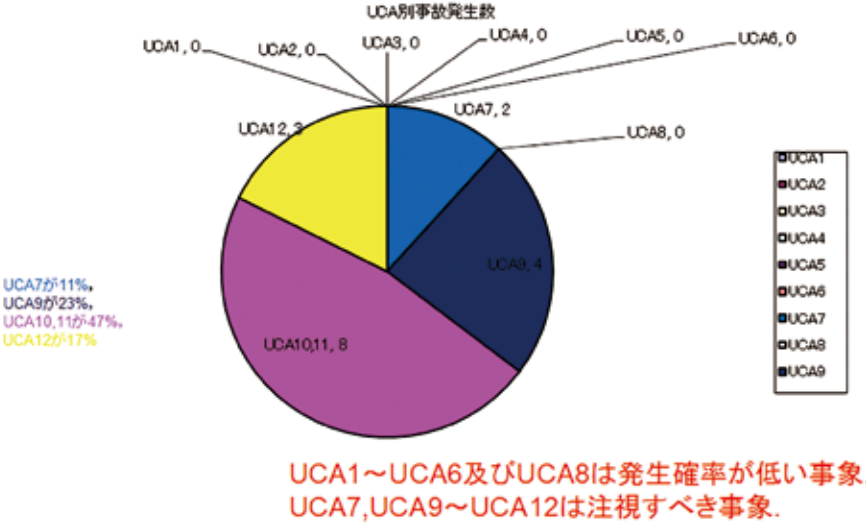


図 2.1-8　UCA と事故発生確率

具体的には、既存踏切制御システムにおける UCA1 ～ UCA6 及び UCA8 は、STAMP としては抽出されるものの、現実的には発生確率が低い事象である。一方、UCA7 が 11%、UCA9 が 23%、UCA10,11 が 47%、UCA12 が 17% であり、これらは注視すべき事象となる。このように、UCA 抽出の網羅性に優れるという STAMP の特長も、その生起確率データを基に UCA そのものを評価することにより、より現実的な安全解析手法とすることができると考える。

また、実際に事故に結びついた注視すべき UCA について分析すると、UCA7 については、踏切支障報知装置の検知性能によるものであるため、検知性能の向上がなされなければクローズドループ型踏切制御システムでも対処できない。これに対して、UCA9 については、乗務員に安全性を委ねる既存踏切制御システムでは限界があるものの、クローズドループ型踏切制御システムでは、踏切道の通過が許容されないため安全が確保できる。UCA10,11 についても、

UCA9 の場合と同様、クローズドループ型踏切制御システムでは、障害物なしが確認できなければ停止パターンが消去されず、踏切道を越えての走行ができないため、安全が確保できる。しかし、この場合でも遮断竿を折ってまで無謀進入する自動車に対しては防護できない。上述の踏切障害事故の直前横断（踏切の踏切警報機が鳴動し遮断かんが降下していたにもかかわらず、自動車が列車の通過直前に踏切に進入した場合）について分類した 8 件のうち、1 件（直前横断の 12.5%）は乗務員の認知が踏切道へ進入していることの認知なのでクローズドループ化で防護が出来る可能性がある。一方、残りの 7 件（直前横断の 87.5%）は、列車走行時速 70 ～ 100km/h で、踏切道からの距離が 190m 以内で障害物の進入を乗務員が認知している状況である。このタイミングで制御が行われるならクローズドループ型踏切制御システムでも回避は難しいと考えられる。この対策としては、遮断完了後には横断者が踏切道へ進入できない仕組み（ロシアの一部踏切に導入されているようなフラップ等のバリアで進入防止を図るなど）の構築や、ITS（高度道路交通システム）と連携し道路側からの進入者に対する制御の仕組みを取り入れることなどが必要となる。また、この対策を施せば、UCA12 に対しても事故防止が期待できる。

以上、クローズドループ型踏切制御システムを導入するなら、17 件の事故のうち 7 件が救済可能となることが分かった。ただ、クローズドループ型踏切制御システムの場合には、メッセージのやり取りが成立して、順次処理フェーズが移行することと、処理の遅滞やメッセージの欠落がすべて「踏切道手前までの停止パターンが解除されない」という安全側の方向に作用する。従って、この前提下での論理部に起因する危険側事象は、条件が整わないにもかかわらず遮断 OK、障害物なしとしてパターン消去のメッセージを出すという論理部ソフトウェアの誤りによるものしか考えられない。この事象に対しては、実際には事前の入念な試験等で対処できるため、UCA1 ～ UCA6 と同様に UCA として抽出されても、無視できるものと考えられる。

2.1.4. おわりに

本節では、クローズドループ型踏切制御システムとして集中一括制御方式を紹介し、STAMP 評価手法を用いて既存の踏切制御システムとの比較を行なった。その結果、クローズドループ型踏切制御システムは、既存踏切制御システムの弱点を補い安全性向上に大きく貢献できることが明らかとなった。

なお、評価に際しては STAMP を用いたが、従来のソフトウェアの安全性評価等に用いられる FTA や FMEA 手法と比べて、より合理的な評価が可能であることが明らかとなった。FMEA によるソフトウェアの評価は、作業量が膨大になるにもかかわらず、ソフトウェアの故障がどのように処理に影響するかというシナリオの合理性についての懸念が払拭できないでいた。同様に、FTA はトップダウン的に致命的要因を抽出できるものの、それが、実際のソフトウェアモジュールのどのような故障によって発生するのかという問いには有効な答えが見出し得ない問題があった。STAMP はソフトウェアモジュールの故障時のインターフェースの挙動から解析できるため、より説得力のある解析ができるとの実感を持った。

また、ケースとして詳細に列挙し、項目として抽出した UCA に対し発生確率を過去の事故統計を根拠として評価した。その結果、UCA としては抽出されたが実際には発生確率が低いものと、現実に考え得る注視すべき UCA とを区別して評価することができ、STAMP 利用上の発展形が確認できた。

2.2. 二輪倒立ロボットの人・機械協調制御の事例

2.2.1. 本試行の目的

STAMP/STPA の適用事例として、LEGO 社 Mindstorms NXT を用いた二輪倒立ロボットの制御問題を取り上げる。本システムは IPA/SEC の WG で開発されたものである［IPA2016-2］。ロボットを目的地まで衝突を避けて移動させるために、操作員による遠隔制御と自立状態を確保するためのロボット本体のコントローラーに組み込まれたフィードバック制御は、相互に協調したり競合したりするため、STPA の有効性を検証する事例として有用と考えられる。また、両制御装置の役割を最適化してより安全なシステムにするには、STPA のような定性的ハザード評価だけでなく、モデルベースの定量シミュレーションによる評価も必要になるため、それらの組み合わせの有用性を示す事例にもなっている。

なお、本分析は、IPA の分析ツール STAMP Workbench を用いて行った。ここでは、ツールによる分析手順として図 2.2-1 のような形が示されているので、この流れに沿って分析を行った結果を説明する。

図 2.2-1　STAMP Workbench の標準分析手順

2.2.2. 分析対象の概要

STAMP/STPA を適用するにあたっては、対象システムの本質的な機能を理解しておくことが大事になる。図 2.2-2 は、対象とする二輪倒立ロボット（ロボット）の外観である。図のように二つの車輪を DC モータで制御することで、自立状態を保つとともに、前進・後進・旋回ができる。また、超音波センサーで障害物検知もできる。さらに、Wi-Fi 無線で遠隔の PC と通信ができ、本システムでは MATLAB/Simulink を用いて、自立・移動制御を行うアルゴリズムを実装した［Mathworks2016］。図 2.2-3 には、作成したシステム全体の外観を示すが、ここに示したように、遠隔操作はジョイスティックで行い、また、制御アルゴリズムと遠隔操作用のヒューマンインターフェースは、MATLAB/Simulink を用いて開発した。

制御アルゴリズムの詳細は・後述するが、基本的には、ジャイロで計測した車体角度とエンコーダで計測した車輪角度を用いた PID フィードバック制御で自立状態を保つ他、超音波センサーを用いた障害物検知、ジョイスティックのスティック角度と方向を利用した前進・後進・左右旋回の操作、Wi-Fi 無線通信による遠隔制御を行う。これらのアルゴリズムは、Simulink の基本ブロックの組み合わせで作成されるので、その内容がプログラミングの知識がなくとも容易に理解できる。システムの動作に際しては、パソコン上の Simulink で作成さ

れた実行コードがロボット側に搭載された CPU にも転送され、パソコンとロボットの CPU で同時に実行される。ここで理解しておく必要があるのは、パソコンとロボット間の通信速度の限界から、システム状態に応じて 100 ～ 1000msec の遅れが生じうること、ロボット側の CPU の性能から、実装アルゴリズムの複雑さに応じて制御サイクルが 10 ～ 50msec の範囲で変動することである。したがって、自立状態を保つためのフォードバック制御は、パソコン側から通信を介して行うのでは間に合わず、ロボット側の CPU で行う必要がある。一方で、前進・後進・旋回などの移動制御は、人がパソコン側から遠隔指示する必要があるが、そこには 100 ～ 1000msec 程度の遅れが発生しうることになる。また、自立制御には PID 制御を用いることから、車体角度と車輪角度の値とそれぞれの角速度も用いることになるため、計測値のノイズ対策も必要になる。PID 制御は線形モデルに基づいているため、車体角度が直立から大きく離れると非線形効果で直立近傍での復元力と異なる挙動となって PID 制御パラメータが適切でなくなって転倒しやすくなることも理解しておく必要がある。

これらの基本的な特性を前提条件として、STAMP Workbench ツールの「準備 1 －前提条件の整理」に入力したものを図 2.2-4 に示しておく。STAMP/STPA 適用に当たっては、分析チームの中でシステムに対する理解を共有しておくことが大事であるが、そのために、ここで紹介したような図表による説明だけでなく、チームで共有すべき最低限の知識を前提条件としてまとめておくことが大切である。

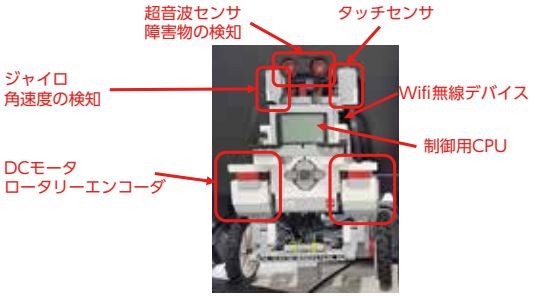
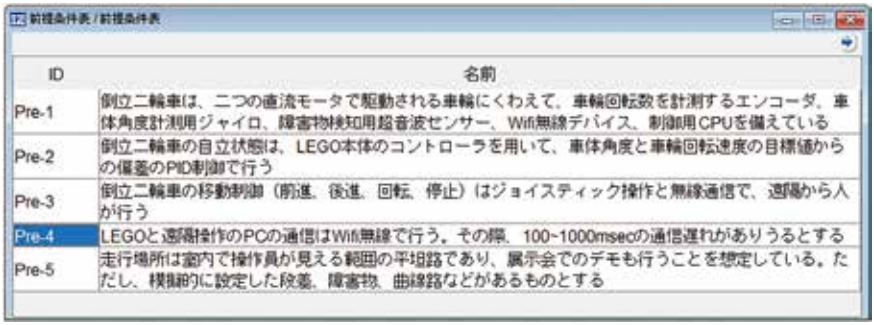


図 2.2-2　二輪倒立ロボットの外観と主要コンポーネント

図 2.2-3　二輪倒立ロボット、遠隔操作装置、遠隔制御（Simulink）の外観



ID	名前
Pre-1	倒立二輪車は、二つの直流モータで駆動される車輪にくわえて、車輪回転数を計測するエンコーダ、車体角度計測用ジャイロ、障害物検知用超音波センサー、Wifi無線デバイス、制御用CPUを備えている
Pre-2	倒立二輪車の自立状態は、LEGO本体のコントローラを用いて、車体角度と車輪回転速度の目標値からの偏差のPID制御で行う
Pre-3	倒立二輪車の移動制御（前進、後進、回転、停止）はジョイスティック操作と無線通信で、遠隔から人が行う
Pre-4	LEGOと遠隔操作のPCの通信はWifi無線で行う。その際、100~1000msecの通信遅れがありうるとする
Pre-5	走行場所は室内で操作員が見える範囲の平坦路であり、展示会でのデモも行うことを想定している。ただし、模擬的に設定した段差、障害物、曲線路などがあるものとする

図 2.2-4　分析対象の前提条件の一覧

2.2.3. STAMP/STPA　制御構造図と UCA 抽出（Step 0,1）

STAMP Workbench のガイドに沿って、システムのアクシデントとハザード、システム安全制約を定義したものを図 2.2-5 に示す。今回のシステムでのアクシデントとしては、二輪倒立ロボットであるため、転倒と衝突が主なものである。小型のロボットであることから「人への危害」のようなものは考慮から外した。これに伴うハザードは、「ロボットの重心の角度と角速度が制御可能領域を逸脱する」、「障害物を検知しても停止が間に合わない」という二つの状態である。また、安全制約は、その裏返しの表現になる。

次に、制御構造図作成のために、コンポーネント抽出表で必要なコンポーネントとそれぞれの役割、コントロールアクション（CA）とフィードバック（FB）を定義する。図 2.2-6 に示すように、（1）操作員・操作装置、（2）操作系コントローラー、（3）安全系コントローラー、（4）ロボット本体、（5）観客、（6）環境外乱という 6 種のコンポーネントによって抽象化して整理した。図 2.2-7 は、このコンポーネント抽出表から半自動で作成した制御構造図である。半自動と書いたのは、自動描画後に見やすさのためにレイアウトを手動で調整したためである。6 つのコンポーネントのうち、（5）（6）は、システムで制御できない外乱を示しているが、それぞれの外乱の影響は、便宜的（今回の試用版の制約）にフォードバック情報として扱った。観客は、今回のシステムが展示会でのデモで用いられることを想定したためであるし、環境外乱は、突風、段差、カーブなど二輪倒立ロボットの自立制御に影響する外部要因を代表する。また、（2）（3）の操作系と安全系のコントローラーは、ロボットに搭載した CPU で実行されるが、機能的な役割が異なるので、ここでは別のコンポーネントとした。操作系コントローラーは、人からの指示に基づいて前進、後進、旋回といった移動を制御し、安全系コントローラーは、PID 制御による自立機能と障害物の自動検知と衝突防止のための停止機能を担う。注意が必要な点は、人からの指示（例えば前進指示）と、自立制御指示（例えば車体が後ろに傾いた時後進指示を出して直立状態に戻す）が相互に矛盾したり、逆に、同期してより大きな支持になったりして車体を不安定な状態にする点である。これは、人と機械の CA の矛盾によるハザードの分析の必要性を示唆している。（1）の操作員・操作装置を一つに抽象化したのは、簡略化のためでもあるが、機械側のコントローラー（2）（3）との対比を明確にするのにも役立つ。

制御構造図に基づくハザード分析では、どのアクションを CA として定義するかである。ここでは、操作員・操作装置から操作系コントローラーに対する指示として、（1）前進・後進・停止指示、（2）旋回指示を、操作系コントローラーからロボットへの指示として、（3）前進・後進・停止指示信号、（4）旋回指示信号を、安全系コントローラーからロボットへの指示として、（5）PID 制御指示、（6）自動停止指示のそれぞれに注目して分析する。そのほかの FB と外乱

は図に記載した通りである。

この 6 種類の CA のハザードに至る可能性を分析した結果（UCA 表）を図 2.2-1 に示すが、24 通り（転倒 8 種、衝突 16 種）の UCA が抽出されている。STAMP Workbench の手順に沿って入力した表であるが、見やすさのため、データをエクセル表に一旦出力した後に整形したものを示している。UCA の具体的説明は、Step 2 の中で合わせて述べる。

アクシデントハザード安全制約表　　2017/12/30

アクシ...	アクシデント	ハザー...	ハザード	安全制...	安全制約
A1	LEGOの転倒	H1	ロボットの重心が制御可能域（車体角度と角速度）を逸脱する	SC1	ロボットの重心を常に制御可能域（車体角度と角速度）に抑える
A2	LEGOの衝突	H2	障害物を検知しても停止が間に合わない	SC2	衝突防止策が間に合う範囲で障害物を検知して減速する

図 2.2-5　アクシデント、ハザード、システム安全制約の定義

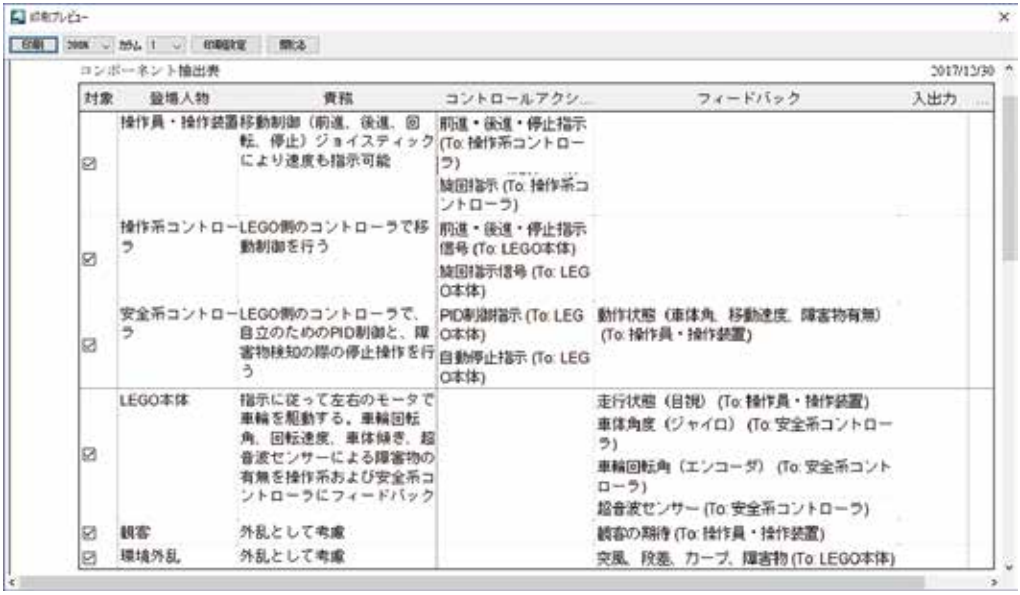


図 2.2-6　コンポーネント抽出表

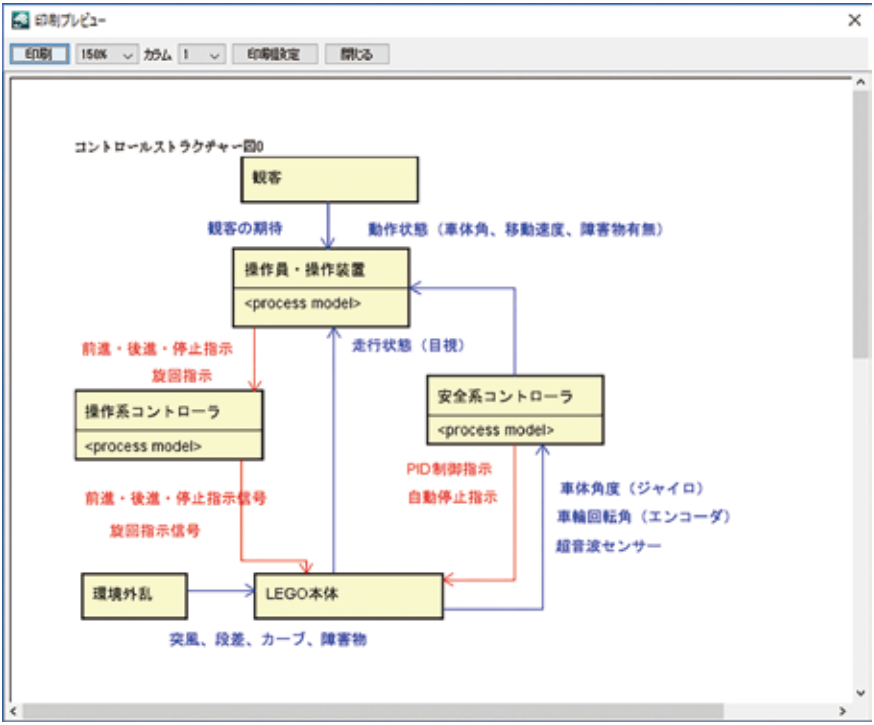

印刷プレビュー
印刷
150%
カラム
1
印刷設定
閉じる
コントロールストラクチャー図0
観客
観客の期待
動作状態（車体角、移動速度、障害物有無）
操作員・操作装置
<process model>
前進・後進・停止指示
旋回指示
走行状態（目視）
安全系コントローラ
<process model>
操作系コントローラ
<process model>
PID制御指示
自動停止指示
車体角度（ジャイロ）
車輪回転角（エンコーダ）
超音波センサー
前進・後進・停止指示信号
旋回指示信号
環境外乱
LEGO本体
突風、段差、カーブ、障害物

図 2.2-7　制御構造図

表 2.2-1　UCA 表

No	CA	From	To	CA 提供条件	Not Providing	Providing causes hazard	Too early/Too late	Stop too soon/ Applying too long
1	前進・後進・停止指示	操作員・操作装置	操作系コントローラ	人からの指示で遅れと、無線経由の遅れがありうる前進・後進・停止はレバー前後操作で制御	(UCA1-N-1) 障害物の回避のための後進・停止操作を出さずに衝突 [SC2]	(UCA1-P-1) 不適切な道路環境条件、不安定な LEGO 車体条件での前進・後進指示による転倒 [SC1] (UCA1-P-2) 障害物の前で前進指示を出して衝突 [SC2]	(UCA1-T-1) 障害物回避のための後進・停止指示が遅すぎで衝突 [SC2]	(UCA1-D-1) 長すぎる前進指示で障害物に衝突（UCA1-P2 と同じ） [SC2]
2	旋回指示	操作員・操作装置	操作系コントローラ	旋回はレバー左右操作で制御	(UCA2-N-1) 障害物回避のための旋回指示が出ないと衝突 [SC2]	(UCA2-P-1) 不適切な環境条件、不安定な LEGO 車体条件での旋回指示による転倒 [SC1] (UCA2-P-2) 障害物の前で間違った旋回をして衝突 [SC2]	(UCA2-T-1) 障害物回避のための旋回指示が遅すぎて衝突 [SC2]	(UCA2-D-1) 障害物回避のための旋回指示が短すぎて十分な回避が出来ずに衝突 [SC2]
3	前進・後進・停止指示信号	操作系コントローラ	LEGO 本体	人からの指示をそのまま伝えるだけとする	(UCA3-N-1) 操作員の回避操作を出さずに衝突 [SC2]	(UCA3-P-1) PID 制御指示との干渉（不安定な LEGO 車体条件）での指示）で転倒 [SC1] (UCA3-P-2) 障害物の前で操作員が出した前進指示をそのまま出して衝突 [SC2]	(UCA3-T-1) 操作員の障害物回避操作指示を出すのが遅れて衝突 [SC2]	該当なし
4	旋回指示信号	操作系コントローラ	LEGO 本体	同上	(UCA4-N-1) 操作員からの旋回指示を出さずに衝突 [SC2]	(UCA4-P-1) 不適切な環境条件、不安定な LEGO 車体条件での旋回指示による転倒 [SC1] (UCA4-P-2) 障害物の前で間違った旋回指示を出して衝突 [SC2]	(UCA4-T-1) 操作員の障害物回避のための旋回指示が遅れて衝突 [SC2]	該当なし
5	PID 制御指示	安全系コントローラ	LEGO 本体	PID 制御による自立のための前進・後進制御	(UCA5-N-1) 信号がなくなると転倒 [SC1]	(UCA5-P-1) 間違った信号を出すと転倒 [SC1]	(UCA5-T-1) 正しい制御信号のタイミング（早い・遅い）がずれると転倒 [SC1]	該当なし
6	自動停止指示	安全系コントローラ	LEGO 本体	障害物検知による自動停止	(UCA6-N-1) 自動停止指示を出さずに衝突 [SC2]	(UCA6-P-1) 不適切な道路環境で停止して転倒 [SC1]	(UCA6-T-1) 検知の遅れないし停止指示の遅れで衝突 [SC2]	該当なし

2.2.4. STPA ハザードシナリオの導出（Step 2）

6 通りの CA の中から、人から操作系コントローラーへの前進・後進・停止指示を例にとって UCA ならびにハザードシナリオの分析結果を図 2.2-8 にそって説明する。人からのこの CA

が出ないとき（Not Providing）は転倒には影響しないが、障害物への衝突を避けるための停止指示を忘れてしまうと衝突の可能性が出てくる。その要因（シナリオ）は一言で代表すると操作ミスであるが、その代表的なものが障害物の見逃しである。特に、展示会を想定して他に気を取られての見逃しを指摘している。もちろん、人の行動の 3 要素である認知、判断、行動のそれぞれにエラー要因がありうるが、ここでは、代表的なものしか指摘していない。実際には、図 2.2-8 のような UCA、ヒントワードなどを参考に複数のメンバーでブレインストーミングをすることで、もれなくシナリオを抽出することが大切である。今回は、人・操作装置を一体で扱ったため、ハードウェアの故障の代表として通信不良を挙げた。Providing causes hazard の UCA のハザード要因は少し複雑になる。一つは段差などの不適切な道路環境での操作ミスによる転倒である。また、急加速、急減速も転倒の可能性もあるし、さらに、加速の後の急な減速のような加速度の急な変化（躍度、Jerk）も転倒につながる可能性が大きい。二つ目のハザードシナリオは、障害物の前で間違って前進指示を出す操作ミスである。タイミングエラーに関わる UCA は衝突に関わるもので、操作の遅れと通信の遅れという人と機械に関わるハザードシナリオが考えられる。CA を与える時間の長さに関しても、前進指示を長く出しすぎて停止が間に合わないというシナリオが考えられる。以上は、人からの移動指示に関わる UCA のハザードシナリオである。ヒントワードを今回は入力していないが、これは、人と操作装置を一体で考えてしまったため、適切なヒントワードが試行版に入っていなかったためである。代わりに、HCF 欄にハザード要因を大くくりにしたキーワードを割り当てて、シナリオ欄に UCA に結びつくシナリオを記載した。この大くくりにしたキーワードは、後で、分析結果を整理する際に役立てることが出来る。

他の UCA についての説明は省略するが、今回分析した全ての UCA、ハザードシナリオならびに対策（コンポーネント安全制約）の一覧を表 2.2-2、表 2.2-3 に示す。それぞれ、転倒・衝突アクシデントにつながる UCA、ハザードシナリオ、対策をまとめてある。試行版では、このような表は一括では出せないが、Step 0、1、2 で得られた分析結果をエクセル表に出力して合成、ソーティングすることで容易に作成できる。ソーティングに際しては、アクシデント、対策対象コンポーネント、HCF キーワードを、第 1~3 キーとして指定して行った。本ツールの利点として、ハザードシナリオや対策が UCA を通してどのアクシデントにつながるかを容易に可視化できる点があげられるが、これは安全論証のトレーサビリティが保たれているということの論証にもなっている。この分析結果から得られる対策（コンポーネント安全制約）の要約を下記に示す。

1. 人と機械（コントローラー）の競合を避けるために、急激な人の操作指示を緩和するフィルタなどの機構をつける
2. 段差通過時などの大幅な車体の傾きに対応する新たな制御アルゴリズムで PID 制御を補完して転倒可能性を抑える（ロボットが不安定な時は操作系コントローラー入力を受け付けない、車体が大きく傾いた時、急速に傾いたときは、PID 制御指示信号にかかわらず強制的に車体を直立にもどすといったルールベース制御）
3. 人の遠隔操作訓練では、急激な加速度変化を避けるような操作習熟だけでなく、展示会を想定して説明しながらの操作のような訓練も行う
4. 制御に必要な計算速度を可能にする CPU、通信デバイスの確保
5. 自己診断機能の追加、運用ログの記録、事前検証、ハードウェア定期点検
6. セキュリティ対策（制御パラメータの改ざん）

この中で、（1）（2）は、線形制御である PID 制御を補うルールベース制御の必要性と、人と機械の指示の矛盾を軽減するアルゴリズムの必要性を示唆している。これらは、詳細設計での課題になるが、モデルベース設計と STPA の組み合わせの大切さを示す事例にもなるので、次節で説明を加えておく。

HCF表0 / HCF表

(UCA1-N-1) 障害物回避のための後進・停止操作を出さずに衝突

ヒントワードセット IPA - (人) 対 (機械)

ID	HCF	ヒントワード	シナリオ
HCF1-N-1-1	操作ミス	(1) Not Providing(操作忘れ)	操作員のミス（よそ見による障害物見逃しと警示音などで他に気を取られてのミス）
HCF1-N-1-2	通信不良		通信回線の不調、操作デバイス不調（電源消耗も含む）、通信デバイス節電モードによる不調

HCF表1 / HCF表

(UCA1-P-1) 不適切な道路環境条件、不安定なLEGO車体条件での前進・後進指示による転倒

ヒントワードセット IPA - (人) 対 (機械)

ID	HCF	ヒントワード	シナリオ
HCF1-P-1-1	操作ミス		道路の段差前後での前進または後進の指示（加速が足りないと段差を超えられないし、加速が強すぎると転倒）
HCF1-P-1-2	操作ミス		急加速による前進・後進指示による転倒 前進加速指示中に後進加速指示を出して転倒

HCF表2 / HCF表

(UCA1-P-2) 障害物の前で前進指示を出して衝突

ヒントワードセット IPA - (人) 対 (機械)

ID	HCF	ヒントワード	シナリオ
HCF1-P-2-1	操作ミス		障害物の前で間違って前進指示（操作ミス）

HCF表3 / HCF表

(UCA1-T-1) 障害物回避のための後進・停止指示が遅すぎで衝突

ヒントワードセット IPA - (人) 対 (機械)

ID	HCF	ヒントワード	シナリオ
HCF1-T-1-1	通信不良		通信遅れで後進指示が遅れて衝突
HCF1-T-1-2	操作ミス		操作員認知遅れで後進指示が遅れる

HCF表4 / HCF表

(UCA1-D-1) 長すぎる前進指示で障害物に衝突(UCA1-P2と同じ)

ヒントワードセット IPA - (人) 対 (機械)

ID	HCF	ヒントワード	シナリオ
HCF1-D-1-1	操作ミス		前進指示を長く出しすぎて停止が間に合わずに衝突

図 2.2-8　ハザードシナリオの分析結果（CA-1: 前進・後進・停止指示）

表 2.2-2　アクシデント、UCA、ハザードシナリオ、対策の一覧（転倒）

アクシデント	ハザード	安全制約	UCA	HCFID	HCF	HCF シナリオ	対策ID	対策	ID	対策対象コンポーネント
[A1]LEGOの転倒	H1	SC1	(UCA1-P-1) 不適切な道路環境条件、不安定な LEGO 車体条件での前進・後進指示による転倒 [SC1]	HCF1-P-1-1	操作ミス	道路の段差前後での前進または後進の指示（加速が足りないと段差を超えられないし、加速が強すぎると転倒）	M4	段差を超える環境での訓練	C1	操作員・操作装置
[A1]LEGOの転倒	H1	SC1	(UCA1-P-1) 不適切な道路環境条件、不安定な LEGO 車体条件での前進・後進指示による転倒 [SC1]	HCF1-P-1-2	操作ミス	急加速による前進・後進指示による転倒前進加速指示中に後進加速指示を出して転倒	M5	最大速度と最大加速度の制限をかける	C1	操作員・操作装置
[A1]LEGOの転倒	H1	SC1	(UCA1-P-1) 不適切な道路環境条件、不安定な LEGO 車体条件での前進・後進指示による転倒 [SC1]	HCF1-P-1-2	操作ミス	急加速による前進・後進指示による転倒前進加速指示中に後進加速指示を出して転倒	M6	加速度の変化率（Jerk）を制限する	C1	操作員・操作装置
[A1]LEGOの転倒	H1	SC1	(UCA2-P-1) 不適切な環境条件、不安定な LEGO 車体条件での旋回指示による転倒 [SC1]	HCF2-P-1-1	操作ミス	過大な速度で走行中に回転指示を出す	M12	速度に応じて回転曲率を制限する制御方式	C1	操作員・操作装置
[A1]LEGOの転倒	H1	SC1	(UCA4-P-1) 不適切な環境条件、不安定な LEGO 車体条件での旋回指示による転倒 [SC1]	HCF4-P-1-1	コントローラ故障	コントローラハードウェア故障	M23	コントローラの定期点検	C2	操作系コントローラ
[A1]LEGOの転倒	H1	SC1	(UCA3-P-1) PID 制御指示との干渉（不安定な LEGO 車体条件）での指示）で転倒 [SC1]	HCF3-P-1-1	CA 干渉	PID 指示で大きな補償信号が出たタイミング（段差、UT による自動停止信号など）で、前進・後進指示を出して転倒	M19	PID 出力時点で操作信号の制限をかける	C2 C3	安全系、操作系コントローラ
[A1]LEGOの転倒	H1	SC1	(UCA5-P-1) 間違った信号を出すと転倒 [SC1]	HCF5-P-1-1	セキュリティ対策不良	PID パラメータをハッカーが変えた	M30	途中でパラメータが変わったことを検知。代替機に代える	C2 C3	安全系、操作系コントローラ
[A1]LEGOの転倒	H1	SC1	(UCA4-P-1) 不適切な環境条件、不安定な LEGO 車体条件での旋回指示による転倒 [SC1]	HCF4-P-1-2	操作ミス	PID 指示で大きな補償信号が出たタイミング（段差、UT による自動停止信号など）で、旋回指示を出して転倒	M24	PID 出力時点で操作信号の制限をかける	C2 C3	安全系、操作系コントローラ
[A1]LEGOの転倒	H1	SC1	(UCA5-N-1) 信号がなくなると転倒 [SC1]	HCF5-N-1-3	通信不良	LEGO コントローラ出力信号線切断で PID 信号が出ない	M29	出力信号を診断する自己診断回路の追加	C2C3C4	LEGO 本体安全系、操作系コントローラ
[A1]LEGOの転倒	H1	SC1	(UCA5-N-1) 信号がなくなると転倒 [SC1]	HCF5-N-1-1	コントローラ故障	LEGO コントローラの故障で PID 信号が出ない	M27	故障の有無を検知する自己診断。故障があれば代替コントローラに切り替え、または、ゆっくりと停止させて倒す（緩和操作）	C3	安全系コントローラ
[A1]LEGOの転倒	H1	SC1	(UCA5-T-1) 正しい制御信号のタイミング（早い・遅い）がずれると転倒 [SC1]	HCF5-T-1-1	コントローラ性能不足	コントローラの制御周期が確保できずに、不安定になって転倒	M34	高性能 CPU の採用	C3	安全系コントローラ
[A1]LEGOの転倒	H1	SC1	(UCA6-P-1) 不適切な道路環境で停止して転倒 [SC1]	HCF6-P-1-1	ロジック仕様ミス	加速中に障害物を検知して緊急停止	M36	加速度の変化率（Jerk）を制限してゆっくりと停止（緩和停止）	C3	安全系コントローラ
[A1]LEGOの転倒	H1	SC1	(UCA6-P-1) 不適切な道路環境で停止して転倒 [SC1]	HCF6-P-1-2	ロジック仕様ミス	段差のあるところ（PID 信号が大きく振れているところ）での緊急停止で転倒	M37	PID 信号と同期させてゆっくりと停止する（知的制御のような新技術開発）	C3	安全系コントローラ
[A1]LEGOの転倒	H1	SC1	(UCA5-P-1) 間違った信号を出すと転倒 [SC1]	HCF5-P-1-4	想定外の外乱	段差や風、障害物などの突然の外乱	M33	運行条件に制約をつける。または、新しい制御方式の開発	C3	安全系コントローラ
[A1]LEGOの転倒	H1	SC1	(UCA5-N-1) 信号がなくなると転倒 [SC1]	HCF5-N-1-2	アクチュエータ故障	モータ故障、または、バッテリー切れによる不動作	M28	モータの定期点検。起動前モータ・バッテリー検査など	C4	LEGO 本体
[A1]LEGOの転倒	H1	SC1	(UCA5-P-1) 間違った信号を出すと転倒 [SC1]	HCF5-P-1-3	コンポーネント故障	車軸とモータのスリップ、車軸と本体固着、タイヤ劣化によるスリップ、タイヤの破損、過搭載（LEGO の特性変化）	M32	定期点検、または、適応制御による PID 最適化	C4	LEGO 本体
[A1]LEGOの転倒	H1	SC1	(UCA5-P-1) 間違った信号を出すと転倒 [SC1]	HCF5-P-1-2	センサー故障	ジャイロ、車輪エンコーダの信号喪失または計測遅れ、計測バイアス、ノイズ印加等の故障、ジャイロモード変化による信号不調	M31	入力信号の自己診断または多重化。故障時は緩和停止。ファームウェアの検証。	C4	LEGO 本体

表 2.2-3　アクシデント、UCA、ハザードシナリオ、対策の一覧（衝突）

アクシデント	ハザード	安全制約	UCA	HCFID	HCF	HCF シナリオ	対策ID	対策	ID	対策対象コンポーネント
[A2]LEGOの衝突	H2	SC2	(UCA1-N-1) 障害物の回避のための後進・停止操作を出さずに衝突 [SC2]	HCF1-N-1-1	操作ミス	操作員のミス（よそ見による障害物見逃しと展示会などで他に気を取られてのミス）	M1	操作訓練とくに展示会などを想定した訓練	C1	操作員・操作装置
[A2]LEGOの衝突	H2	SC2	(UCA1-P-2) 障害物の前で前進指示を出して衝突 [SC2]	HCF1-P-2-1	操作ミス	障害物の前で間違って前進指示（操作ミス）	M7	操作訓練	C1	操作員・操作装置
[A2]LEGOの衝突	H2	SC2	(UCA1-T-1) 障害物回避のための後進・停止指示が遅すぎで衝突 [SC2]	HCF1-T-1-2	操作ミス	操作員認知遅れで後進指示が遅れる	M9	訓練特に展示会向け訓練	C1	操作員・操作装置
[A2]LEGOの衝突	H2	SC2	(UCA1-D-1) 長すぎる前進指示で障害物に衝突（UCA1-P2と同じ） [SC2]	HCF1-D-1-1	操作ミス	前進指示を長く出しすぎて停止が間に合わずに衝突	M10	訓練特に展示会向け訓練	C1	操作員・操作装置
[A2]LEGOの衝突	H2	SC2	(UCA2-P-2) 障害物の前で間違った旋回をして衝突 [SC2]	HCF2-P-2-1	操作ミス	操作ミスで障害物の前で間違った旋回操作をする	M13	操作訓練	C1	操作員・操作装置
[A2]LEGOの衝突	H2	SC2	(UCA2-D-1) 障害物回避のための旋回指示が短すぎて十分な回避が出来ずに衝突 [SC2]	HCF2-D-1-1	操作ミス	旋回操作の時間が短すぎて障害物を回避できずに衝突	M16	操作訓練	C1	操作員・操作装置
[A2]LEGOの衝突	H2	SC2	(UCA2-T-1) 障害物回避のための旋回指示が遅すぎて衝突 [SC2]	HCF2-T-1-1	操作ミス	障害物の認知が遅れて旋回操作が遅れるシナリオ	M14	操作訓練	C1	操作員・操作装置
[A2]LEGOの衝突	H2	SC2	(UCA1-N-1) 障害物の回避のための後進・停止操作を出さずに衝突 [SC2]	HCF1-N-1-2	通信不良	通信回線の不調、操作デバイス不調（電源消耗も含む）、通信デバイス節電モードによる不調	M3	操作前のバッテリー確認、事前動作確認	C1	操作員・操作装置
[A2]LEGOの衝突	H2	SC2	(UCA2-T-1) 障害物回避のための旋回指示が遅すぎて衝突 [SC2]	HCF2-T-1-2	通信不良	通信遅れで旋回操作が遅れる	M15	Xms 以下の通信速度を確保する通信機器の採用	C1	安全系コントローラ操作員・操作装置
[A2]LEGOの衝突	H2	SC2	(UCA1-T-1) 障害物回避のための後進・停止指示が遅すぎで衝突 [SC2]	HCF1-T-1-1	通信不良	通信遅れで後進指示が遅れて衝突	M8	Xms 以下の通信速度を確保する通信機器の採用	C1 C3	安全系コントローラ操作員・操作装置
[A2]LEGOの衝突	H2	SC2	(UCA3-N-1) 操作員の回避操作を出さずに衝突 [SC2]	HCF3-N-1-1	コントローラ故障	コントローラ故障（含むロジックのバグ）	M17	事前点検、ソフト検証	C2 C3	安全系、操作系コントローラ
[A2]LEGOの衝突	H2	SC2	(UCA4-P-2) 障害物の前で間違った旋回指示を出して衝突 [SC2]	HCF4-P-2-1	コントローラ故障	コントローラハードウェア故障	M25	コントローラの定期点検	C2 C3	安全系、操作系コントローラ
[A2]LEGOの衝突	H2	SC2	(UCA4-T-1) 操作員の障害物回避のための旋回指示が遅れて衝突 [SC2]	HCF4-T-1-1	コントローラ性能不足	コントローラの計算周期の遅れ	M26	高性能 CPU にする	C2 C3	安全系、操作系コントローラ
[A2]LEGOの衝突	H2	SC2	(UCA3-P-2) 障害物の前で操作員が出した前進指示をそのまま出して衝突 [SC2]	HCF3-P-2-1	ロジック仕様ミス	不適切な操作員からの指示をコントローラがそのまま実行	M20	コントローラ側で不適切な前進指示を防ぐ緩和機構をもたせる	C2 C3	安全系、操作系コントローラ
[A2]LEGOの衝突	H2	SC2	(UCA4-N-1) 操作員からの旋回指示を出さずに衝突 [SC2]	HCF4-N-1-1	ロジック仕様ミス	操作員の指示をコントローラ側で制限をかけて信号を出さない	M22	UT による障害物検知と停止、コントローラが人の指示を変えた際に、ログを残して、後日の状況把握に役立てる	C2 C3	安全系、操作系コントローラ
[A2]LEGOの衝突	H2	SC2	(UCA3-T-1) 操作員の障害物回避操作指示を出すのが遅れて衝突 [SC2]	HCF3-T-1-1	コントローラ性能不足	コントローラの計算周期の遅れ	M21	高性能 CPU にする	C3	安全系コントローラ
[A2]LEGOの衝突	H2	SC2	(UCA6-N-1) 自動停止指示を出さずに衝突 [SC2]	HCF6-N-1-1	センサー故障	UT の故障、設定ミス、アルゴリズム不具合	M35	事前の検証、センサー多重化	C3	安全系コントローラ
[A2]LEGOの衝突	H2	SC2	(UCA3-N-1) 操作員の回避操作を出さずに衝突 [SC2]	HCF3-N-1-2	アクチュエータ故障	アクチュエータ故障（モータ故障、接続不良）	M18	事前点検	C4	LEGO 本体
[A2]LEGOの衝突	H2	SC2	(UCA6-T-1) 検知の遅れないし停止指示の遅れで衝突 [SC2]	HCF6-T-1-1	センサー故障	UT 反応時間が劣化して遅れが生じる	M38	定期点検	C4	LEGO 本体
[A2]LEGOの衝突	H2	SC2	(UCA2-N-1) 障害物回避のための旋回指示が出ないと衝突 [SC2]	HCF2-N-1-1	操作ミス	障害物に気がつかずに旋回を忘れて衝突シナリオ	M11	UT による障害物検知と停止、操作訓練とくに展示会などを想定した訓練		操作員・操作装置

2.2.5. モデルベースの定量分析

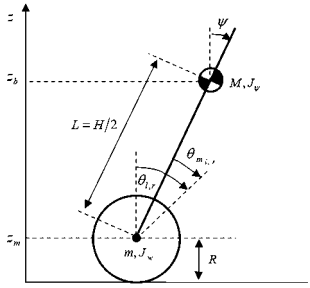
2.2.5.1. モデルならびに評価法の説明

前節でのハザード対策のうち、PID 制御を補うルールベース制御と、人と機械の指示の矛盾を軽減するアルゴリズムについては、定量評価が必要である。このためモデルベース設計手法を用いた評価事例を説明する［Sunouchi2017］。

ロボットの動特性モデルは、図 2.2-9 に示すように、車体角ψと車輪回転角θ、ならびに、それらを制御する力として車輪を回転させるモータの直流電圧ｖｒ、ｖｌ、さらに、路面からの力としてランダムに与えられる外乱ηからなる動特性方程式で記述できる。モデルの詳細式とロボットに相当するパラメータは文献にゆずる［Mathworks2016］、［Sunouchi2017］。モータの直流電圧は PID 制御器からの出力と人からの操作指示から構成される。この PID 制御系の構成は図 2.2-10 に示す。車輪角（すなわち水平面の移動距離）の目標値は、人からの指示で与えられ、そこに、PID フィードバック信号が重畳する。PID 制御は、車輪回転角と車体傾斜角を用いるが、車体傾斜角の積分ゲインはゼロに固定している。これは、傾斜角が直立状態からバイアスしたまま（傾いたまま）でも構わないことを意味している。一方で、車輪角の積分ゲインはゼロでない一定の数値を指定するので、目標値（すなわち、目標の移動距離）に達するまで制御が行われることになる。また、PID 制御器の指示は、外乱ηの時間変化に応じてランダムに変動するため、人からの操作指示と同期したりしなかったりという複雑なモータ電圧変動になる。したがって、この PID 指示値と人の操作指示値の干渉でロボットが転倒するかどうかは統計的に評価する必要がある。実機でこのような統計的評価を行うことは簡単ではないが、モデルベースのシミュレーションでは、乱数系列を変えて何回も異なる外乱のもとで転倒可能性を評価することが出来るというメリットがある。

なお、実機の PID パラメータの最適設定も課題であるが、今回の事例では、下記のような手順で実機のパラメータ設定を行っている［Mathworks2016］、［Sunouchi2017］。

- 前述のモデルを線形化し最適レギュレータ理論に基づいて PID パラメータを設定
- 前述の非線形モデルを用いたシミュレーションを繰り返し、外乱に対してロバストな PID パラメータ空間を絞り込む
- モデルベースの最適値を参考に、実機でのパラメータサーベイを行って最適値を決める。実機ではステップ応答を行うのが難しいため、一定時間の運転の間の車輪角と車体角の変動値（標準偏差）を計測し、車輪角の変動がある値（20 度）以下で車体角の変動が最小という基準で最適値を選んだ



$$[(2m+M)R^2+2J_w+2n^2J_m]\ddot{\theta} + (MLRcos\psi 2n^2J_m)\ddot{\psi} - MLR\dot{\psi}sin\psi = F_\theta$$

$$F_\theta = \alpha(v_l+v_r) - 2(\beta+f_w)\dot{\theta} + 2\beta\dot{\psi} + \eta$$

図 2.2-9　二輪倒立ロボットの動特性モデル

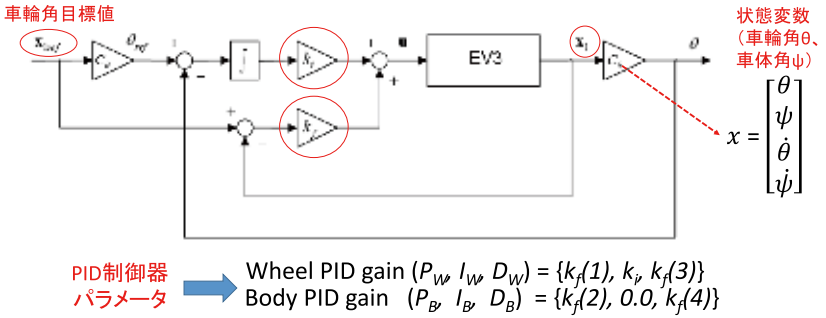
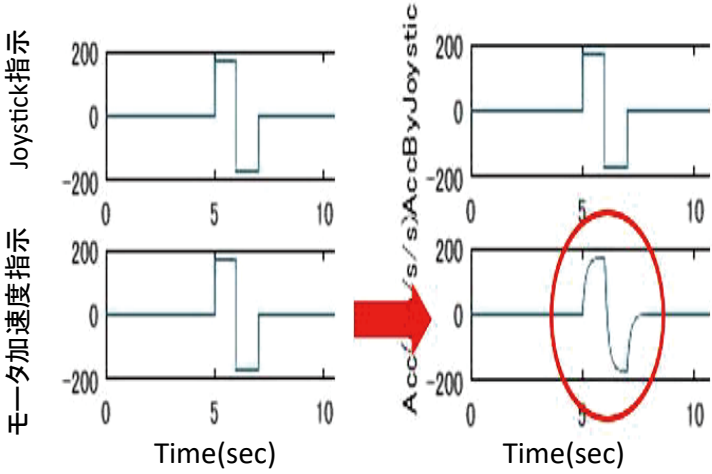


図 2.2-10　PID 制御系の構成

2.2.5.2. 人の加減速指示速度（Jerk）の制限［Sunouchi2017］

人の操作の中で前進加速から後進加速への指示の変更が一番急激な変化と考えられるので、その際に、ロボットの転倒の可能性が大きくなる。図 2.2-11 は、加速度の時間変化（上段）と、それにフィルタをかけて滑らかな変化にした際の時間変化（右側下段）の事例である。この急激な変化と PID フィードバック指示の重畳で転倒する可能性があるが、これは、先に述べたようにランダムに変動するフィードバック信号の状態に依存する。そこで、シミュレーションでランダムな時系列外乱を 100 ケース発生させ、その中で何回転倒するかをフィルタの時定数を変えて評価した。図 2.2-12 はその結果である。時定数 0.01 秒の場合、99% 転倒するのに対して、0.2 秒まで時定数を大きくすると転倒がなくなる。ただ、時定数を大きくすることは停止操作も含めて遠隔操作の応答が遅くなるので、衝突回避が出来なくなるという欠点も出てくる。そのため、12% の転倒を許容するとして、0.15 秒の時定数を最終的には採用した。



（左：制限なし、右：0.15 秒 1 次遅れフィルタ）

図 2.2-11　加減速変化速度（Jerk）の制限

TCofAcc (sec)	0.20	0.175	0.15	0.125	0.1	0.75	0.05	0.025	0.01
Overturning rate (%)	0	3	12	31	57	85	95	98	99

図 2.2-12　Jerk フィルタ時定数と転倒率（%）の関係

2.2.5.3. ルールベース制御による補助［Sunouchi2017］

PID 制御は線形モデルを仮定しているが、車体が大きく傾くと sin（ψ）に比例して力が働くため、線形領域で設定した PID パラメータでは復元制御ができなくなる。そこで、下記のようなルールベース制御の追加を考えた。つまり、車体が大きく、かつ、速く傾いた場合、無条件で復元のための制御信号（pidMax）を出すということである。

- If　車体角＜−α　and　角速度＜−β、Then　pid　＝　＋pidMax
- If　車体角＞α　and　角速度＞β、Then　pid　＝　−pidMax

前節と同様に、100 通りのランダムな外乱系列を与え、さらに、転倒しやすくするために、t=5 秒で加速、t=6 秒で減速操作を出し、30 秒間で転倒するか否かを評価する。その結果の一つで、復元のための制御信号 pidMax の値を 20 ～ 60 まで変更した際の転倒率を図 2.2-13 に示す。pidMax を大きくすると転倒率が 0%から 91%まで増える。このとき、同時に、上記のルールが作動した回数（nff、30 秒間での作動率）を評価したものを下段に示した。復元のための pidMax が小さすぎると、nff が 32.9%と高頻度で作動するため転倒はしなくても車体の揺れが不安定になってしまう。そのため、7%の転倒率、8.29%の作動率で、pidMax=40 を最終的には採用した。また、角度しきい値 α =40 度、角速度しきい値 β =200 度 / 秒を用いている。詳細は文献［Sunouchi2017］を参照されたい。こうして設定したパラメータによるルールベース制御の適用結果を図 2.2-14 に示すが、ルールを用いない場合の転倒（上段）、ルールによる転倒防止（中段）、ルールの作動頻度（下段）のそれぞれの時間変化を示した。効率的にルールが作動して転倒を防止していることがわかる。

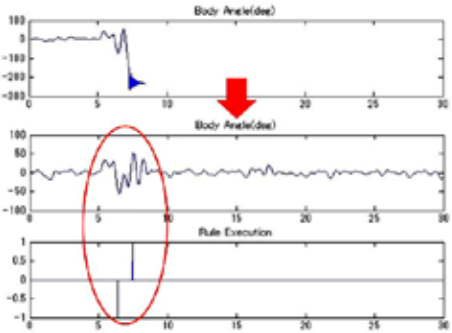
Initial value

pidMax	20	30	40	50	60
Overturning raio (%)	0	3	7	12	91
nff	32.90	25.49	8.29	5.5	8.22

High nff　　High Overturning ratio

（転倒率（%）とルール適用率（%））

図 2.2-13　ルールベース制御のパラメータサーベイ



（上：ルールベース制御なし、中：ルールベース制御適用時、下：ルール適用インデクス）

図 2.2-14　シミュレーション事例

2.2.6. 考察

二輪倒立ロボットの人・機械協調制御という問題に対して、STAMP/STPA によるハザード分析を行った事例を紹介した。STAMP Workbench を使うことで、分析結果がトレーサビリティを持って整理できた。STPA の標準手順に沿って、Step 0 でアクシデント、ハザード、安全制約、ならびに、システムの制御構造図を定義し、Step 1 で非安全制御行動（UCA）を分析した。Step 2 のハザード誘発要因・ハザードシナリオの分析については、簡単化のために抽象度を上げて記載した。そのため網羅性などの視点で不十分さを感じるかもしれない。これは、第三者レビューなどを通してブラッシュアップする必要があろう。ただし、本システムを展示会で使った実績からすると、主要なシナリオは抽出できていたと考えられる。

一方で、抽出された主要なシナリオの中で、人と機械の制御アクションの競合によるハザードについては、定性的な問題点を STPA で評価した後に、モデルベースシミュレーションで定量的な対策まで検討した。急激な人の操作の緩和策ならびに PID 制御を補完するルールベース制御の導入であり、これらは実際の試作版には Simulink のソフトウェアとして実装し、有効性を確認してある［Sunouchi2017］。

STAMP Workbench のツールとしての有用性も、本事例の適用で明らかにすることができた。制御構造図、アクシデント、ハザード、システム安全制約、UCA、ハザード要因（シナリオ）、コンポーネント安全制約という多段階の推論プロセスと、そこから得られる結果が整合性をもって管理できることが分かった。同時に、これらの結果をわかり易く整理したり、次の具体的な設計プロセスにつなぐためのツールとしての機能は、さらなる事例検討の結果をフィードバックしてブラッシュアップしてゆく必要もある。

もう一つの大事な知見として、制御構造図の可視化表現から見える課題がある。ここでは、自立制御のための制御ループ（10-50msec の応答）と、遠隔移動制御のための制御ループ（100-1000msec）の二つの構造があることがわかる。これらの異なる制御ループは相互に協調したり競合したりしてシステム全体を制御するが、それぞれの応答時定数の違いは制御構造図の中で明確に表現できてはいない。これらの相互作用に起因するハザードは定性的には STPA で分析できているが、応答時定数の違いに起因するハザードの定量的分析は、モデルベースシミュレーションに頼っている。このような複数の制御ループの協調と競合システム、さらには、各制御ループの応答時定数の違いなどを含めた現実的なシステムのハザード分析に STPA を適用する場合の手順については、さらなる分析事例を通してノウハウを積み上げてゆくことが望まれる。

2.3. 安全性論証に使う STAMP ／ STPA ～自動車編～

本節および付録 A）は、一般社団法人 JASPAR（Japan Automotive Software Platform and Architecture、以降 JASPAR）機能安全 WG STAMP/STPA チームが、JASPAR 会員向けに 2017 年 10 月 13 日発行した『安全性論証に使う STAMP ／ STPA ～自動車編～ Ver1.0』から内容を抜粋し、編集を加えたものである。

2.3.1. はじめに

●背景

クルマの運動性能を支える「走る」、「曲がる」、「止まる」といった主要機能の運転支援がグローバルマーケットで実用化され、自動運転や先進運転支援システム ADAS（Advanced Driver Assistance Systems）を視野に入れた取り組みが本格化するなど、クルマを取り巻く社会環境は劇的な変化を遂げようとしている。自動車産業の永い歴史の中で、これまでクルマは単独の移動手段として進化してきたが、自動運転は、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」がつながることで、複数のシステムやサービスが連携する大規模化・複雑化したものとなっている。また、これらの普及・展開にともなって、クルマはこれまでに経験したことのない想定外の危険事象に遭遇する可能性がある。このようなシステムにおける潜在的な危険事象は社会に与えるインパクトが大きく、想定外の危険事象をいかに想定内に取り込んで事前に対処するかが、自動車産業における重要な課題となっている。

このような大規模化・複雑化したシステムの安全解析手法の一つとして、すでに宇宙航空、鉄道など高い安全性が求められる分野で活用されている STAMP/STPA がある。STAMP/STPA は、システムを構成する各コンポーネント間のインタラクションに着目した事故モデルによって安全解析を行うといった特徴を持っており、これは自動車業界において古くから用いられてきた FTA（Fault Tree Analysis）や FMEA（Failure Mode and Effects Analysis）といったコンポーネント単体の機能不全や故障モードに着目した分析とは異なるアプローチといえる。したがって、FMEA や FTA 等の従来手法と STAMP/STPA を相互補完させることによって安全解析の強化を図り、想定外の危険事象を想定内に取り込む可能性を高めることで、危険事象への備えに厚みを持たせることが期待できる。

●目的と期待

一般社団法人 JASPAR（Japan Automotive Software Platform and Architecture）機能安全 WG は機能安全設計における安全性論証を効果的かつ効率的にすることを目的に、現場設計者が使える安全性論証ガイドの策定に取り組んでいる。同 WG では、その取り組みの一環として、2016 年度に具体的な事例「電動パーキングブレーキ（EPB：Electronic Parking Brake）」の安全設計を取り上げ、STAMP/STPA の適用可能性を検討した。

安全性論証において、自動車業界で古くから用いられてきた演繹型分析手法として FTA があり、帰納型分析手法として FMEA がある。いずれもシステムのコンポーネント（構成要素）単体の機能不全に着目している分析手法であるが、STAMP/STPA はシステムのコンポーネント間のインタラクションに着目している。そのため、STAMP/STPA の適用は、従来の手法では抽出できない危険事象や故障の抽出が期待できると考えられる。

この EPB 事例によって、ISO 26262 で用いられる従来の安全解析との違いを整理しつつ、自動車分野でも STAMP/STPA が適用できることを示すと同時に、現場設計者が STAMP/STPA を実践するときに役立つガイドを示す。これによって、STAMP/STPA による安全解析を、組織間にまたがったアイテムやシステムにおける効果的かつ効率的な安全性論証に役立てよう

とするものである。たとえば、自動運転や ADAS の設計現場で、従来とは観点の異なるアプローチを実践するにあたり、本事例を参考とすることで、効果的かつ効率的なクルマの安全設計につながることが期待できる。

IPA では JASPAR 機能安全 WG と共同で、自動車産業はもちろん、他産業でも参考となるようにするよう 2016 年度 JASPAR 機能安全 WG の検討事例の編集等を行ってきた。本節はその結果をまとめたものであるが、国内ではあまり見られない STAMP/STPA の自動車産業における事例が、他の製品分野において適用を進める上での参考や動機づけとなることが期待できる。

2.3.2. 国際安全規格及び従来の安全分析手法と STAMP/STPA

2.3.2.1. 国際安全規格と STAMP/STPA

安全性に関わる国際標準として、ISO や IEC が様々な安全規格を規定している。たとえば、IEC 61508 は電気・電子・プログラマブル電子の機能安全規格であり、ISO 26262 は本事例で取り扱っている自動車の機能安全規格である。安全性論証とは、これらの規格が定める対象範囲において、安全性を設計文書等のエビデンスを用いて論理的に説明できること（説明責任が果たせること）、として用いられている用語である。ISO 26262 に準拠するためには、安全分析において FTA や FMEA などの従来手法を用いることが多く、これらを用いて安全性論証を行ってきた。これらの分析では、HAZOP（Hazard and Operability Study）のガイドワードや、失敗事例やフィールドデータを含む既存の知識もベースとなっている。

しかし、現行の国際安全規格がスコープとしていない自動運転等の新しい分野では、安全性論証が従来の手法だけでは足りない可能性がある。STAMP/STPA を用いることは、現行の国際安全規格に準拠することとは直結しないが、自動車やそのシステムそのものの安全性論証を強化する一部として扱うことが可能となるかもしれない。

2.3.2.2. ISO 26262 における安全分析と STAMP/STPA との比較

●本項の概要

先に述べたとおり、従来の安全分析手法は、失敗事例やフィールドデータを含む既存の知識で補いながら、定められた範囲や条件下において網羅的な分析が可能である。たとえば、ハードウェアコンポーネントの故障に着目した FMEA では、そのコンポーネントから生じる機能不全については網羅的に分析することができる。しかし、システムが複雑化し、これまで経験したことのない製品やサービスでは、特に相互作用を含めた事象を網羅的に分析することが困難な場合があり、STAMP/STPA の適用が有効になりうる。

そこで、ここでは次の 3 つについて整理する。

・従来の安全分析手法と STAMP/STPA との比較
・従来手法と ISO 26262 の安全分析との比較
・ISO 26262 の安全分析と STAMP/STPA との比較

●従来の安全分析手法と STAMP/STPA との比較

システムにおけるアクシデントの原因を、機器の故障や人間のオペレーションミスに置く、従来のアクシデントモデルでは、システムのアクシデントの可能性が潜在している状態（すなわちハザード）とそれを引き起こす要因を事前に分析するための安全分析手法として、FTA や FMEA、HAZOP などの手法が用いられてきた（実際にはこれら以外にも様々な分析手法が知られているが、自動車業界においてはこの 3 つがよく用いられている）。

この従来の安全分析手法と STAMP/STPA との比較は参考文献［IPA2016］や［IPA2018］に

譲り、ここでは割愛する。これら参考文献では、技術的観点で比較したものとなっているが、適した開発フェーズや実施者に問われるスキルといったところにも違いがある。

●従来手法と ISO 26262 の安全分析との比較

本事例は自動車の機能安全規格の対象システムであるため、ISO 26262 を取り上げる。本項の説明は、用語の違いはあるものの、IEC 61508 等の国際安全規格でもほぼ同様に考えることができる。

① ISO 26262 の安全分析の特徴

ISO 26262 の安全分析の特徴を次の A）と B）に記載する。

A）階層的（段階的）な安全要求の導出

ハザード分析とリスクアセスメントの実施、その後の安全目標の導出

⇒　機能安全要求と機能安全コンセプト

⇒　技術安全要求と技術安全コンセプト

⇒　ハードウェア安全要求、ソフトウェア安全要求

このように安全要求を階層的に導出する。機能安全コンセプトは、自動車やそのシステムのアーキテクチャレベルで検討する。

なお、技術安全要求と技術安全コンセプトはシステムレベルで導出されるもので、これ以降が実装に依存した要求として扱われる。

B）階層的（段階的）な安全分析の実施

システム設計段階

⇒　ハードウェア設計段階、ソフトウェア設計段階

このように階層的に分析する。たとえば、システム FMEA の後にハードウェア FMEA、ソフトウェア FMEA を実施する。システム設計段階の安全分析は、システムのアーキテクチャレベルで行う。

②従来手法と ISO 26262 の安全分析との比較

従来の安全分析手法と ISO 26262 の安全分析との比較を表 2.3-1 に示す。

表 2.3-1　従来手法と ISO 26262 の安全分析との比較

		ISO 26262 の安全分析	従来の安全分析手法
安全要求の導出と配置		安全要求を導出し、エレメントに配置（エレメント毎に集計）	（特に規定なし）
安全分析	階層	原則 2 階層（システム⇒ハードウェア、ソフトウェア）	一般に 1 階層（D － FMEA）
	アーキテクチャ図	有 （機能安全コンセプトではアーキテクチャ図に安全要求を配置するのが一般的）	無 （ハードウェア回路図での検討が一般的）
	故障モード	システム FMEA ではエレメント間や外部とのインターフェース上の障害に着目。HAZOP のガイドワード等を利用	ハードウェア故障に着目。ON/OFF 固着故障等がメイン

● ISO 26262 の安全分析と STAMP/STPA との比較

ISO 26262 に準拠する上では、従来の安全分析手法以外に、直ちに STAMP/STPA を含めた新しい安全分析手法がなければならないわけではない。ただし、安全性論証を強化し、自動運転等の新しい分野では従来手法だけでは不足する可能性もある。

STAMP/STPA は、人や組織、複雑なシステムまたはソフトウェアの相互作用に関わる不具合に伴うハザードを分析することから、より上流の開発フェーズに適しているといえる。その意味で、ハザード分析とリスクアセスメント、安全要求の導出と配置、そしてシステム設計までの実施がより適していると考えられる。

ISO 26262 における安全要求の導出と STAMP/STPA との比較を表 2.3-2 に、ISO 26262 におけるシステム FTA またはシステム FMEA と STAMP/STPA との比較を表 2.3-2 に記載する。

表 2.3-2 と表 2.3-3 でまとめた比較結果から、従来手法でコンポーネント単位の機能不全に着目した分析を行い、併せて STAMP/STPA を用いることで、相互作用に関わる故障モードを系統的に導出・分析できることが分かる。

表 2.3-2　ISO 26262 における安全要求の導出と STAMP/STPA との比較

	ISO 26262 における安全要求の導出	STAMP/STPA
段階（フェーズ）	コンセプト	（明確な規定はない） コンセプトやシステム（もしくはコンセプトよりさらに上位）を想定
構成図（エレメント間や外部とのインタラクションの記載を含む）	アーキテクチャ図にインタラクションも記載されるのが一般的（ただし、インタラクションの記載に関して規格での明確な規定はない）	コントロールストラクチャーにインタラクションも記載 （インタラクションを記載するよう明確に規定されている）
手順	1）ハザード分析とリスクアセスメント及び安全目標の導出 2）機能安全要求の導出 3）機能安全コンセプト（配置） 4）技術安全要求の導出 5）技術安全コンセプト（配置）	Step 0：（準備 1）アクシデント、ハザード、安全制約の識別 Step 0：（準備 2）コントロールストラクチャーの構築 Step 1：非安全なコントロールアクション（UCA）の抽出 Step 2：ハザード要因（HCF）の特定

表 2.3-3　ISO 26262 における FTA、FMEA と STAMP/STPA との比較

	ISO 26262 におけるシステム FTA またはシステム FMEA	STAMP/STPA
段階（フェーズ）	システム	（明確な規定はない） コンセプトやシステム（もしくはコンセプトよりさらに上位）を想定
障害の箇所	エレメント間や外部とのインターフェース上に現れる障害（エレメント間や外部とのインタラクション）に着目	（同左）
障害の要因	各エレメントのランダムハードウェア故障やシステマチック故障に着目	左記に加え、複数のエレメント間や、人間（ドライバー）、車両等との相互作用に関わるハザード要因にも着目
故障モードの分析段階	故障モードは一般的に段階化せずに分析	故障モードを 2 段階で分析 Step 1：非安全なコントロールアクション（UCA）の抽出 検討対象のコントローラー間のコントロールアクションに着目して抽出する。 Step 2：ハザード要因（HCF）の特定　コントロールストラクチャーの各コンポーネント間のインタラクションに着目して網羅的に分析する手法が推奨されている。
故障モードのガイドワード、ヒントワード	規格での明確な規定はないが、一般的に HAZOP ガイドワードや、失敗事例やフィールドデータを含む既存の知識を利用	独自のガイドワード、ヒントワードを定義 UCA については、4 つのガイドワードが定義されている。（「与えられない」、「ハザードを誘発する不正内容が与えられる」、「早すぎ、遅すぎ」、など） また、HCF については、相互作用としてフィードバックへの影響も明記したヒントワードが提案されている。（例：「不適切なフィードバック、あるいはフィードバックの喪失」等）

2.3.3. JASPAR STAMP/STPA 事例

2.3.3.1. 概要

自動運転など大規模なシステムへの機能安全適用に備え、安全分析やそのレビューを補強することを目的に STAMP/STPA 事例を展開した。まず、STPA 手法に慣れることを目的に、十分実績のあるシステムである電動パーキングブレーキを題材に選んだ。なお、本節で取り扱う EPB はどの自動車会社の車両にも該当しない仮想システムとしている点に注意されたい。また、本節では、事例分析の抜粋を記載しており、詳細は付録を参照されたい。

JASPAR は、機能安全開発における説明補強を期待して STPA に取り組んでおり、主に下表に述べる点で工夫を施している

表 2.3-4　JASPAR の STPA 工夫点

STPA Step	目的	工夫点
Step 0	コントロールストラクチャー記述の標準化	人、システム、車両、環境の 4 つを起点に分析対象を捉えるテンプレートや凡例を標準化した
	第 3 者への分析対象の説明性向上	コントロールストラクチャーを起点とした STPA ではなく、なぜそのコントロールストラクチャーとしたのか、作成過程の説明を追加した
Step 1	分析結果の説明性と理解容易性向上	分析結果をコントロールストラクチャー上に記述しアクシデントに至る影響を記述し、第 3 者への説明や理解を促進する手がかりとした

2.3.3.2. 電動パーキングブレーキの説明

電動パーキングブレーキの説明の予備説明として、まず関連する自動車用語について説明する。これらは、分析用に必要というわけではなく自動車用語についての読者への説明としている。

●オートマチックトランスミッション車両の駐停車操作の説明

分析対象車両は、オートマチックトランスミッション（自動変速機）（以降 AT）を搭載した車両とする。一般的に AT 車を駐停車[1]する際は、以下の操作が求められる。

1. シフトポジションを P（パーキング）に入れる
2. パーキングブレーキをかける[2]
3. 車両の電源を OFF にする

（以降、ドアを開けて外に出る、施錠する）

1　道路交通法第 2 条参照

2　寒冷時に電動パーキングブレーキをかけると、パーキングブレーキが凍結し、解除できなくなるおそれがあるが、本分析では寒冷時を想定せず、パーキングブレーキをかけることとする

●パーキングブレーキの説明
　パーキングブレーキは、車両の駐停車状態の維持することに利用する。パーキングブレーキをかけるとタイヤがロックされ、駐停車状態が可能となる。パーキングブレーキを戻すとタイヤのロックが解除され、駐停車状態から走行に移ることができる。パーキングブレーキの例として、レバー操作によるパーキングブレーキがあげられる。レバーを引き上げるとパーキングブレーキがかかり、レバーを下げるとパーキングブレーキが解除される。

●電動パーキングブレーキの説明
　EPB はパーキングブレーキを一部電動化したものである。本事例では、以下の操作によって動作する EPB を定義する

・EPB の作動方法
1. フットブレーキを踏み、車を完全停止させる
2. EPB をかける
3. EPB が作動していることを、メータ内の EPB ランプが点灯していることで確認する

・EPB の解除方法
1. フットブレーキを踏み、車の完全停止を維持する
2. EPB を解除（手段は述べない）
3. EPB が解除していることを、メータ内の EPB ランプが消灯していることで確認する

2.3.3.3. アクシデント、ハザード、安全制約の識別
　機能安全上取り扱っている範囲をアクシデントとした。アクシデント、ハザードの記載内容については、MIT STAMP Workshop での自動車事例を調査し、なるべく定量化につなげられ、新規システムを対象にできるよう上位概念で記載した。

表 2.3-5　アクシデント、ハザード、安全制約

アクシデント		ハザード（車両レベル、システムレベル）				安全制約	
NO	内容	NO	内容	NO	内容	NO	内容
A1	自車が他車／車以外（固定物）と衝突	H1	安全な相対的距離が確保できない	H1-1	自車が意図せず動き出す（駐停車時）	SC1	自車が意図せず動き出さないこと（駐停車時）
				H1-2	自車が意図せず急減速する（走行時、走行開始時）	SC2	自車が意図せず急減速しないこと（走行時、走行開始時）

2.3.3.4. Step 0: コントロールストラクチャーの作成

コントロールストラクチャーは分析対象を把握した結果と捉えることができる。分析対象によっては、コントロールストラクチャーの作成過程を示す必要がある。その手段として参考文献（はじめての STAMP/STPA）では、SysML などが紹介されている。これらのダイアグラムは必ずしも必要でない場合もあるし、表や図（ポンチ絵）で示した方が適切な場合もあると考える。今回は、コントロールストラクチャーのコンポーネントを抽出するためポンチ絵を描くことから STEP0 を着手した。まず、人、システム、車両、環境の４つを起点に考える。

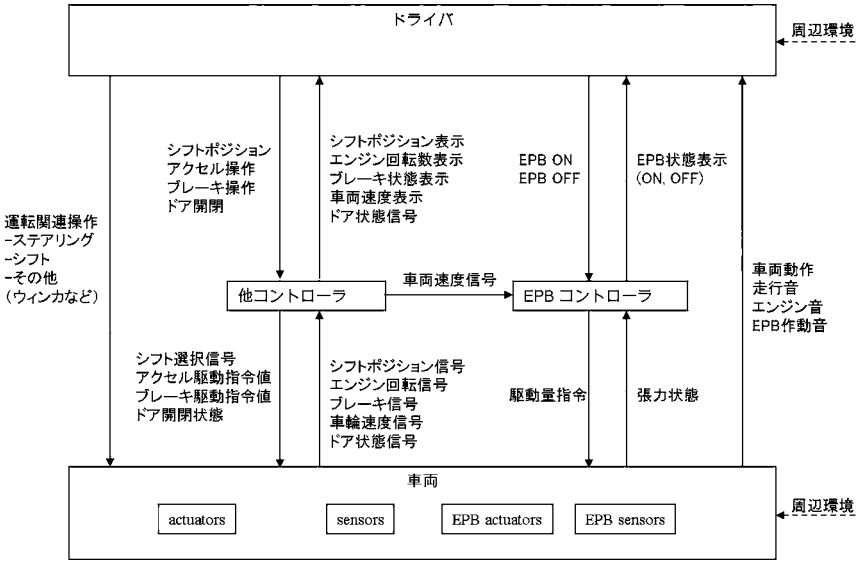
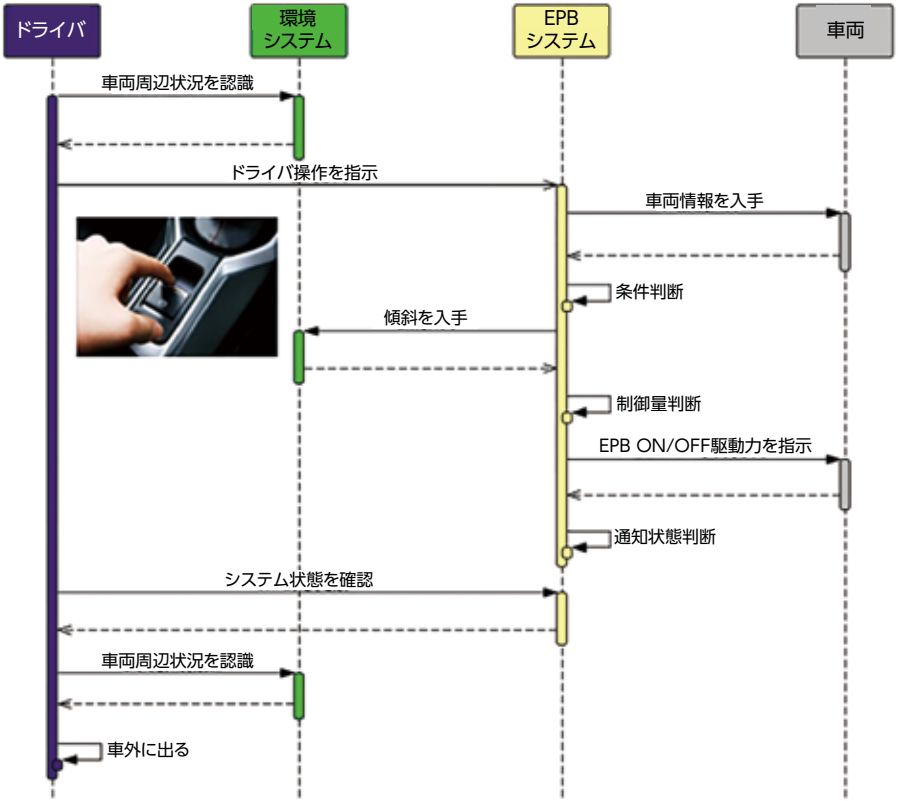


図 2.3-1　コンポーネント抽出のためのポンチ絵

図 2.3-1 は、EPB に関係したコンポーネントと相互作用について知っていることを図示したポンチ絵である。これを事前のアーキテクチャ想定として、表 2.3-5 に示した二つの安全制約に対応する機能（駐停車時と走行時）ごとにコントールストラクチャを作成する。なお、本節では駐停車時を想定した安全制約 SC1 関連にみに着目したコントロールストラクチャーとその分析結果の概要を紹介する。SC2 関連の分析結果を含めた詳細内容は付録を参照されたい。

この機能の想定は、「ドライバーが自車を傾斜地にフットブレーキで停車させ、シフトポジションを P にセットし、その後パーキングブレーキを効かせ、車両電源を OFF し、ドアをあけて車外に出る。」ことである。安全設計上厳しいシーンを抽出すべく、シフトポジションが本来の P ではなく、P 以外にセットされていることを前提とし、分析を行う。

まず、人、システム、車両、環境の 4 つをコンポーネントとして、その相互作用を、図 2.3-2 に示すようなシーケンス図を作成することで特定した。

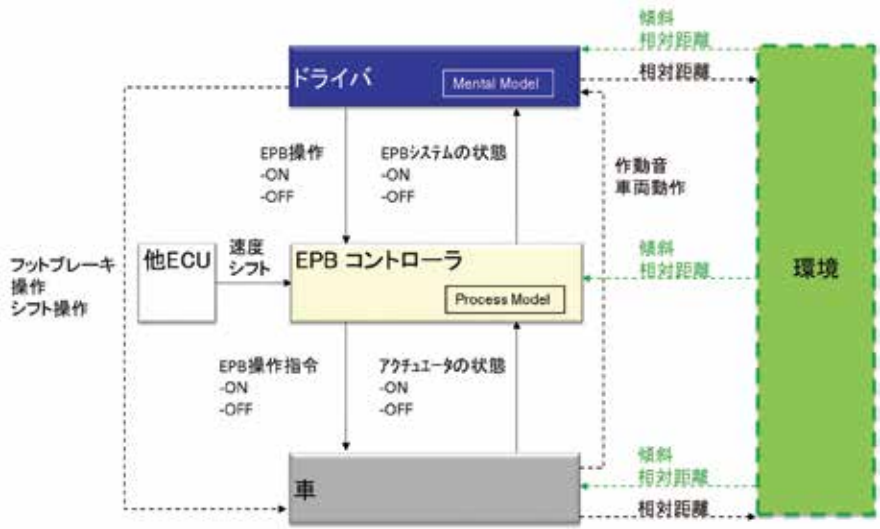


（前提：シフトポジションが本来期待される P ではなく、N にセットされている）

図 2.3-2　SC1 関連の相互作用特定

図 2.3-2 から得られた相互作用をもとに作成したコントロールストラクチャーを図 2.3-3 に示す。アクシデントに至るシナリオを記述するため、ドライバーと環境の間に相対距離と記

載し、ドライバーが車両を降りて車両から離れていくことを記載した。ここでのコントロールストラクチャーは、JASPAR 内で作成したテンプレートに基づいて表現している。



（前提：シフトポジションが本来期待される P ではなく、N にセットされている）

図 2.3-3　SC1 関連のコントロールストラクチャー

2.3.3.5. Step 1: UCA の抽出

図 2.3-3 に記載されているコントロールアクションに対して、4 つのカテゴリをあてはめ UCA を特定した。その結果を表 2.3-6 に示すが、SC1 自車が意図せず動き出さないこと（駐停車時）を侵害する UCA のみ UCA の ID を付与し、侵害しないものは「-」を付与した。この UCA を誘発するハザード要因を Step 2 で特定しているが、その詳細については付録を参照されたい。

表 2.3-6　SC1 侵害の UCA の抽出

コントロールアクション	コントロールアクションを与えないとハザード（Not providing causes hazard）	コントロールアクションを与えるとハザード（Providing causes hazard）	コントロールアクションを与えるのが早すぎ / 遅すぎでハザード（Incorrect Timing/ Order）	コントロールアクションを与えるのが長すぎ / 短すぎでハザード（Stopped Too Soon / Applied too long）
CA1-1 パーキングブレーキをかける操作をする（EPB ON）	UCA1-1_NP1 ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキをかける操作をしない	- ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキをかける操作をする（正常）	- （ドライバーが車を降りてからはパーキングブレーキの操作ができない）	UCA1-1_D1 ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキが効く前に、パーキングブレーキをかける操作を停止する
CA1-2 パーキングブレーキを解除する操作をする（EPB OFF）	- ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキを解除する操作をしない（正常）	UCA1-2_P1 ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキを解除する操作をする	- （ドライバーが車を降りてからはパーキングブレーキの操作ができない）	- ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキが解除する前に、パーキングブレーキを解除する操作を停止する
CA2-1 パーキングブレーキをかける（EPB ON）	UCA2-1_NP1 EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキをかけない	- EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキをかける（正常）	UCA2-1_T1 EPB コントローラーは、傾斜地上の車両からドライバーが降りた後に、パーキングブレーキをかける	UCA2-1_D1 EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキが効く前に、パーキングブレーキをかけるのをやめる
CA2-2 パーキングブレーキを解除する（EPB OFF）	- EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキを解除しない（正常）	UCA2-2_P1 EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキを解除する	UCA2-2_T1 EPB コントローラーは、傾斜地上の車両からドライバーが降りた後に、パーキングブレーキを解除する	- EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキが解除する前に、パーキングブレーキを解除するのをやめる

＊前提：シフトポジションが本来期待される P ではなく、N にセットされている
＊ UCA ID は元の CA と下記カテゴリと対応させ、追跡および遡及が容易にできるよう工夫した。

2.3.4. コントロールストラクチャーテンプレート提案

●テンプレートの狙い

コントロールストラクチャー作成時の悩みは、コンポーネントの抽象度があげられる。考え方の手がかりとして、自動車における STPA の階層的コントロールストラクチャーテンプレートを提案する。

自動車開発の特徴は、複数の組織 / 会社が協調して開発することである。ステークホルダが多岐にわたるため、下記 2 レベルとし相互関係性を持たせた

1. 車両レベルのコントロールストラクチャー
   人／システム／環境／車両をコンポーネントして考える抽象度とした
2. システムレベルのコントロールストラクチャー
   車両レベルの車載システムコントローラーを展開した抽象度とした。自社に関係する部分／しない部分が分かるようコンポーネントの凡例を分けた。

これによって、組織間の成果物の説明の強化が期待できる。例えば、車両レベルのコントロールストラクチャーを用いて自動車会社が分析し、システム会社が一段抽象度を下げたシステムレベルのコントロールストラクチャーを用いて分析する使い方が想定できる。上位の変更が下位へ、下位の変更が上位へと伝えやすくすることは、昨今開発のスピードが目まぐるしい自動運転の開発をサポートすることを期待する。

●凡例

表 2.3-7　コントロールストラクチャー凡例

NO	凡例	説明	役割	事例
1		人間 コンポーネント	コントローラ	ドライバ、乗員、ディーラメカニック
2		車載システム コンポーネント	コントローラ	システムコントローラ、ＥＣＵ、センサ、スイッチ
3		コントロールドプロセス コンポーネント	コントロールドプロセス	車両、機械部品、タイヤ、ブレーキパッド、ギヤボックス
4		環境 （コンポーネントではない）	コンポーネントへの インプットor アウトプット	環境要因（道路、歩行者、天候、後部座席の乗員）
5		相互作用	コントロールアクションor フィードバック	相互作用
6		関連するコンポーネント	コンポーネントへの インプットor アウトプット	ブラックボックスとして扱う （他社製システムなど）
7		コンポーネントからの 入力や出力	コンポーネントへの インプットor アウトプット	想定する相互作用（他社製システムが関連するインタラクションなど）
8		相互作用や入出力の内容	相互作用や入出力の内容	ドライバ操作ON/OFF

2.3.4.1. 車両レベルコントロールストラクチャー

人／システム／環境／車両の 4 つを起点とした抽象度で分析対象を捉える。

車両におけるコントローラーとコントロールドプロセスの関係をストラクチャとして表現した。エアバッグなど特殊なシステムを除いて、このテンプレートで表現することが可能である。図 2.3-4 において、ドライバーとシステム間はコントローラー、コントロールドプロセスの関係が成り立ち、システムと車両間においてもコントローラー、コントロールドプロセスの関係が成り立つ。環境は、ドライバーやシステムの入力、車両の出力と関係している。

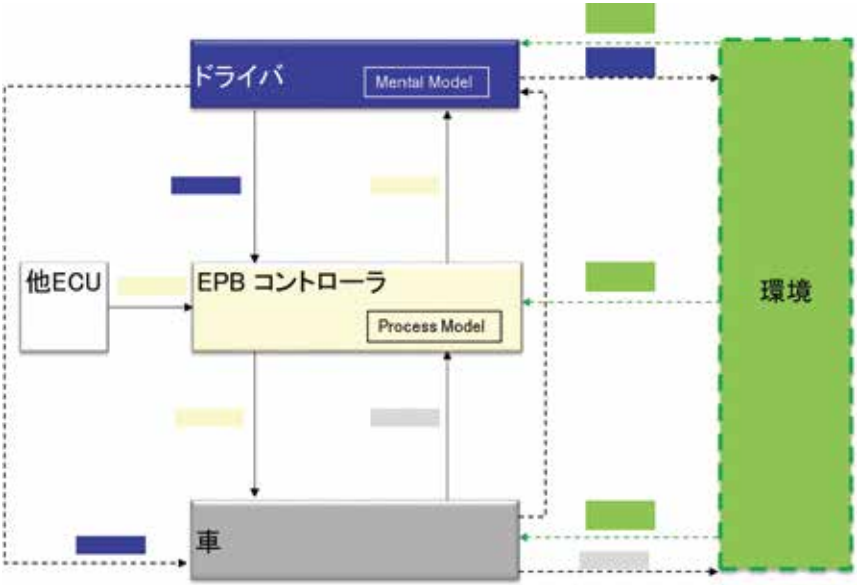


図 2.3-4　車両レベルコントロールストラクチャー

2.3.4.2. システムレベルコントロールストラクチャー

車両レベルの車載システムコントローラーを展開したレベルとした。当該システムに直接関係無いコンポーネントであってもインタラクションを記述することが必要であるため、システムコンポーネントの色（凡例参照）を変えることで表現した。環境は、ドライバーやシステムの入力、車両の出力と関係している。当該システムが環境に無関係の場合は、環境コンポーネントは記載せずともよい。

（ア）自社製品が環境と無関係な場合は、環境の記載は不要。

（イ）他社製 ECU とのインタラクションがある場合、他 ECU というコンポーネントを使用する。

（ウ）自社製品と関係のある HMI までは実線のインタラクションで、その先の HMI と人とのインタラクションは点線とした。

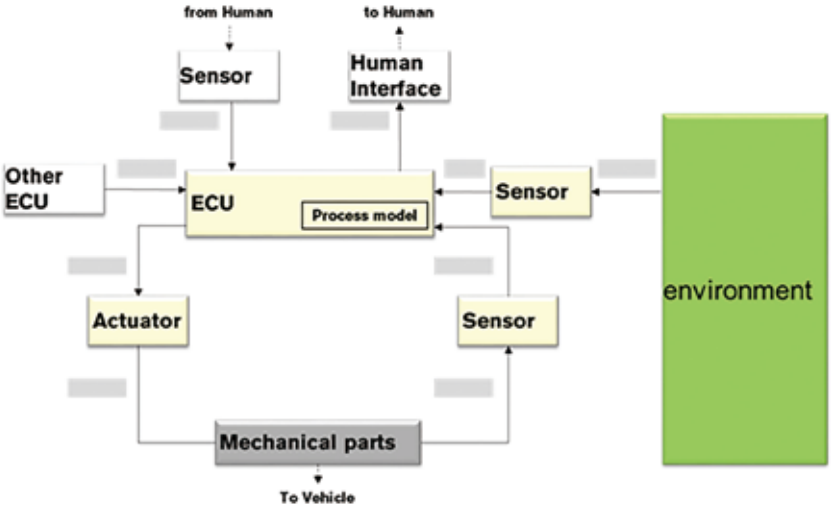


図 2.3-5　システムレベルコントロールストラクチャー

2.3.5. おわりに

今回の分析事例である EPB は良く知られたシステムであり、STPA の特徴が出し切れる題材として必ずしもふさわしくないかもしれない。STPA の特徴のひとつは、新規性の高いシステムに対して適用し、分析対象の設計情報があまりない状況でも分析できる点であるが、EPB システムはこれには相当していないことによる。しかしながら、EPB システムは、人間の運転操作とそれを受けたコンピューターの機械への指示、人間へのフィードバック表示を含んだ典型的な車載システムであり、自動運転における分析時に、法令 / 技術指針などの相互作用の影響を分析できる期待がある。また Hierarchical control structure の他方の Operation の中でも、車両が運転されるシーンに限定した分析を取り扱った。運転シーン以外のフェーズにも STPA を適用することでメリットが得られると期待し、今後の取り組みを検討したい。

アクシデントについては、機能安全上取り扱っている範囲とした。本来の STAMP/STPA に比べ範囲は狭く、この点においても STPA の特徴が見えるまでにはいたらなかった。しかし、今回の分析事例で、機能安全の補強として、下記 3 点の工夫を加え、安全分析範囲拡大と妥

当性説明を向上した STAMP/STPA の事例展開ができた。

1）安全分析および安全要求導出の説明補強
- ●完成されたコントロールストラクチャー起点の STPA ではなく、分析対象の特徴を捉えコントロールストラクチャーにいたる過程を示すことで、コントロールストラクチャーの妥当性示せた。
- ●車両 / システム / 人 / 環境を起点とするテンプレートで、分析の抽象度の手がかりを示せた。
- ●故障だけでなく非故障を取り扱うことで安全分析範囲を拡大できた。

2）設計仕様を抽出
- ●分析だけでなく安全要求仕様を抽出することができた

以上から、設計現場で使える、設計現場の人にとって使える事例を作成することができ、自動車への STPA 適用への可能を示せたと言える。

2.4. セキュリティへの応用の海外事例

本節では、STAMP 海外事例として STPA-SafeSec を紹介する。STPA-SafeSec は、安全性と脆弱性を統合して分析するための STPA 拡張である。また、とくに脆弱性分析を STPA ベースで実施する際に有用となる関連事項を併せて紹介する。

2.4.1. はじめに

STAMP はシステム理論に基づく事故モデルであり、STPA は STAMP に基づくハザード分析（安全分析、安全解析）手法である［Leveson2012］［IPA2016］。STPA は既存の安全分析手法で分析が困難であった複雑な対象に対し有効であると言われている［Leveson2012］。走行中の自動車のエンターテインメント系から走行系への乗っ取り、コンピューター・ワームによる遠心分離機の破壊といった、セキュリティ侵害が安全性を脅かす事例がある。これらの事例は、STAMP/STPA の用語を用いれば、セキュリティにかかわるハザード誘発要因（Hazard Causal Factor、HCF）が最終的に安全制約を破るという事例である。このような事例を分析するためには、安全分析とセキュリティ分析を統合した STPA の拡張が必要となる。

前述の事例は、STPA の Step 0 準備 1 において安全性にかかわるアクシデント、ハザード、安全制約を識別し、Step 2 の HCF 特定のヒントとしてセキュリティにかかわるヒントを導入するだけで、分析できそうに思える。しかし、セキュリティ分析にはシステムの詳細情報が必要なことが多く、STPA で使用するコントロールストラクチャー図（Control Structure Diagram、以下 CSD）がセキュリティ分析に十分であるかといった検討は必要であろう。このような背景の下に、STPA の拡張として、STPA-Sec［Young2013］や STPA-SafeSec［Friedberg2013］が提唱されてきた。

本節では、具体的な事例（マイクログリッドにおける広域電力網と局所電力網の接続（併入）を STPA-SafeSec で分析）を用いて STPA-SafeSec の手順が説明されている論文を紹介する。はじめに、STPA-SafeSec の特徴と手順を紹介し、次に STPA-SafeSec の適用事例を紹介する。更に、とくに脆弱性分析を STPA の拡張で分析する際に有用な事項を紹介する。

なお本節では、STPA-SafeSec で用いられている用語を、標準的 STAMP/STPA の用語に著者の解釈で置き換えている。

2.4.2. STPA-SafeSec の概説

本節では、文献［Young2013］で提案されている STPA-SafeSec を紹介する。STPA は安全性分析を目的としているが、STPA-SafeSec は、安全制約と脆弱性を統合して分析できる STPA-SafeSec の拡張であり、文献［Young2013］では、STPA-SafeSec の詳細な手順が提案されている。本節ではスペースの関係から、STPA-SafeSec の手順詳細を割愛し、標準的 STPA の手順［IPA2016］に合わせて解説する。

2.4.2.1. STPA-SafeSec の特徴

STPA-SafeSec では以下の上 2 つが貢献として挙げられている。また本節では、1 つ目の貢献からの派生効果であるが 3 つ目も貢献として挙げる：

1. 機能 CSD と物理 CSD を持つ
2. Step 2 で使用する HCF ヒントのセキュリティ拡張
3. 安全性・セキュリティ対策の統合

これらの特徴について、それぞれ述べる。

機能レイヤー（Control Layer）の CSD（以下、機能 CSD）内のコントロールループごと

に構築する物理レイヤー（Component Layer）の CSD（以下、物理 CSD）を用いることで、Step 2 でセキュリティ侵害の経路が特定しやすくなる。例えば、上位の機能 CSD の時刻同期機能は、下位の物理 CSD では GPS に詳細化されたとする。この詳細化により、GPS の既存の脆弱性を HCF として利用できる。また機能 CSD と物理 CSD 間のコンポーネントを対応付けることで、機能 CSD で特定した UCA から、物理 CSD において識別するその UCA へ至る HCF とハザードシナリオとの対応が容易になる。更に、物理 CSD を考えることで、既存の脆弱性分析を活用するには、抽象化した機能レベルの分析では限界があり、この事例のような物理 CSD の導入が必須となる。

標準的 STPA では安全性を分析するために、アクシデント、ハザード、安全制約を識別し、最終的に HCF とハザードシナリオを特定する。HCF を特定する際には、コントロールループ中で安全性にかかわる HCF ヒントとして、コンポーネント故障、ヒューマンエラー、コミュニケーションエラー、ソフトウェア不具合、要求仕様不具合などを用いることが一般的である。STPA-SafeSec では、これら従来のヒントにセキュリティにかかわる HCF ヒントを追加し、セキュリティにかかわる誘発要因としてなりすましなどのセキュリティにかかわる誘発要因を特定できるようにしている。

なお、STPA-SafeSec で採用されているセキュリティにかかわる HCF ヒント以外では、例えば、セキュリティにかかわるヒントとして STRIDE［MS2018］の利用が考えられる。

STPA-SafeSec の Step 2 では、抽象的ハザードシナリオから具体的ハザードシナリオを導出し、安全制約やセキュリティ制約を特定している。この導出方法により、ハザードシナリオ（安全・セキュリティ制約）たちは木構造となる。この木構造を分析することで、安全制約とセキュリティ制約間の関係が明らかになり、これらを統合できる。例えば、機能 CSD であるフィードバックが間違っていることが UCA の指示につながり、ひいてはハザードを引き起こすこというシナリオ 1 が策定できたとする。更に、この機能 CSD を詳細化した物理 CSD により、サイバー攻撃によりそのデータが改ざんされ、同じ UCA へ至るというシナリオ 1.1 を特定できたとする。このとき、安全の観点から導入されたデータチェック機構は改ざん検出にも利用できるため、シナリオ 1 の安全対策がシナリオ 1.1 のセキュリティ対策を兼ねることになる。

2.4.2.2. STPA-SafeSec の手順

STPA-SafeSec は詳細な手順に分割されている（[Friedberg2013]）。本節では、標準的 STPA のプロセスである［IPA2016］で解説されている手順にまとめて STPA-SafeSec を解説する。

Step 0　準備 1（STPA-SafeSec Ⅱ ～ Ⅳ）：対象とするシステムのロス（アクシデント）、ハザードを定義し、各ハザードに対する安全制約とセキュリティ制約を識別する。セキュリティ制約といったセキュリティにかかわる事項を対象とすること以外は、標準的 Step 0 準備 1 と同じである。

Step 0　準備 2（STPA-SafeSec Ⅴ）：上記制約の実現に必要な、機能コンポーネントとそれらの相互作用（コントロールアクションとフィードバック）を分析して機能 CSD を構築する。機能 CSD は標準的 Step 0 準備 2 で構築する CSD に対応する。Step 1（STPA-SafeSec Ⅵ ～ Ⅸ）：機能 CSD 内の各コントロールループに対し、トーマス博士が提案する拡張 Step 1［IPA2016-3］により、UCA（原文 Hazardous Control Action）を抽出する。拡張 Step 1 は、コントローラーの入力の組み合わせに対し網羅的に UCA か否かを判定するため、自動化に適しているといった特徴を持つ。

なお STPA-SafeSec の他手順との関連から、STPA-SafeSec Step 1 は標準的 Step 1 でも良いと考えられる。他方、STPA-SafeSec Step 1 で用いるガイドワードは標準的 STPA の 4 つのガイドワードと同じである。

Step 2 は STPA-SafeSec の特徴的な概念・手順を多く含むため、［Mathworks2016］に従い、以下の 3 つの手順に分割して解説する。

Step 2a （STPA-SafeSec X、XI）：機能 CSD の各コントロールループに対し、物理 CSD を構築する。物理 CSD は機能 CSD をアーキテクチャレベルへ詳細化した記述である。このとき、これらの構成要素間を対応付ける。また、抽象的ハザードシナリオ（原文 Safety Related Flaws、System Flaws）を策定する。この抽象的ハザードシナリオは、標準的 STPA［Leveson2012］の安全にかかわる HCF ヒントを参考に特定される。

抽象レベルでのシナリオとして、ハザードシナリオのみを扱うのは、標準的 STPA が扱うシナリオに加え、セキュリティ侵害が安全性を脅かすシナリオを扱うことを目的としているためと考えられる。

Step 2b （STPA-SafeSec XII）：Step 0 準備 1 で識別済みのハザード及びリスト 1、2 のセキュリティ制約を基に機能 CSD のコンポーネントへ抽象的安全・セキュリティ制約を課し、機能 CSD と物理 CSD の対応に基づき、抽象的安全・セキュリティ制約を物理 CSD の要素に割り振る。

物理 CSD のコンポーネントに課される安全制約は、Step 1 準備 1 で識別した安全制約であり、標準的 STPA の安全制約である。他方、セキュリティ制約は STPA-SafeSec Step 2b で登場する。後の事例において、物理 CSD のコンポーネントに既知の脆弱性としてスプーフィング（spoof）とジャミング（jam）が知られている場合に、このコンポーネントへセキュリティ制約（CSTR-A1、CSTR-A2）を課すという利用法からは、リスト 1、2 の内容はセキュリティにかかわる HCF ヒントであるとも言える。

リスト 1：完全性に対する汎用的脅威

- CSTR-I1 コマンド・インジェクション（Command injection）
- CSTR-I2 コマンド欠落（Command drop）
- CSTR-I3 コマンド操作（Command manipulation）
- CSTR-I4 コマンド遅延（Command delay）
- CSTR-I5 観測値インジェクション（Measurement injection）
- CSTR-I6 観測値欠落（Measurement drop）
- CSTR-I7 観測値操作（Measurement manipulation）
- CSTR-I8 観測値遅延（Measurement delay）

リスト 2：可用性に対する汎用的脅威

- CSTR-A1 通信遅延（Communication delay）
- CSTR-A2 通信欠落（Communication dropped）
- CSTR-A3 ノード過負荷（遅延）（Node overloaded（delay））
- CSTR-A4 ノード過負荷（欠落）（Node overloaded（drop））

Step 2c（STPA-SafeSec XIII）：Step 2a で識別した抽象的ハザードシナリオを機能 CSD に対するトップレベルのハザードシナリオとし、それを物理 CSD へ詳細化していく。このとき、詳細化関係があるため、ハザードシナリオたちは木構造となる。

このように、抽象的ハザードシナリオを具体的ハザードシナリオへ詳細化するアプローチは、[IPA2016-3]3.4 Identifying causal factor scenarios でも紹介されている。しかし［IPA2016-3］

では、機能 CSD のみを用いて詳細化しているのに対し、STPA-SafeSec では 2 つの CSD を用いて詳細化している点が異なる。

2.4.3. STPA-SafeSec による分析事例

本節では、［Friedberg2013］の 4、5 節にある事例を解説する。文献［Friedberg2013］は事例としてマイクログリッドを用いており、とくに広域電力網と局所電力網の接続（併入）におけるハザード分析を実施している。事例対象の簡単な解説は、本節の Step 0 準備 2 と Step 1 にある。また詳細な解説は、［Friedberg2013］と［Friedberg2015］を参照いただきたい。

2.4.3.1. Step 0 準備 1

STPA-SafeSec Step 0 準備 1 では、安全に関する事柄に加えセキュリティに関する事柄を考える以外は、標準的 STPA と同じである。この事例では、次のロスを識別している：

- L1：人間への危害
- L2：電力機器の損傷
- L3：ユーザの電器機器の損傷
- L4：停電

続いて、次のハザードを識別している（カッコ内は関連するロスを表す）：

- H1：非同期での系統併入（L1、L2、L3、L4）
- H2：電力機器の運転制限外での運用（L1、L2、L3、L4）
- H3：電力品質指標の逸脱
  - H3.1 電圧（L1、L3）
  - H3.2 周波数（L3、L4）
- H4：同期制御の不調（L4）
- H5：地域の電力需要への対応不可（L4）

更に、システムに対する高抽象度の安全制約を、ハザードの否定形を取ることで識別している。このとき、制約は安全制約（CSTR-Sn）、可用性制約（CSTR-An）、完全性制約（CSTR-In）のようにどのような属性に対する制約かを分けて番号付けしている。なおこの事例では、安全制約 CSTR-S1 から CSTR-S5（H1 から H5 の否定形）しか登場しないが、一般には可用性制約と完全性制約も扱う。

2.4.3.2. Step 0 準備 2

STPA-SafeSec の Step 0 準備 2 では機能 CSD を構築する。この機能 CSD が標準的 STPA の CSD に相当する。機能 CSD におけるコンポーネント（原文 Node）は Nn で、接続は Cn で番号付けされる。なお Step 2a で、この機能 CSD を詳細化した物理 CSD を構築し、2 つの CSD のコンポーネントを対応付ける。従って、対応が分かりやすい番号付けが望ましい。

この事例では、とくに速度制御器が制御するコントロールループ図に着目し、機能 CSD（図 2.4-1）を構築している。

図 2.4-1 について解説する。速度制御器（N1）、ローカル PMU（N4、ローカルマイクログリッドにある電圧位相計測装置（Phasor Measurement Unit））、ホスト PMU（N5）、速度制御器とローカル・ホスト PMU 間の接続（C4 と C5）により接続されている。各 PMU からは、電圧（Xm）、周波数（ω）、位相（φ）が周期的に送られてくる。速度制御器は、同期が取れているかを確認し、

サーキットブレーカー（N6）へ再開が安全か否かを送信する（この事例では、サーキットブレーカーは自動ではなく、操作員が相当する操作を実行すると仮定しているため、開閉命令ではなく、安全か非安全かの情報が送信されている）。また速度制御器は、原動機制御器（N2）を経由して、ジェネレーター（N3）へ運転設定値（set point、C1）を設定する。

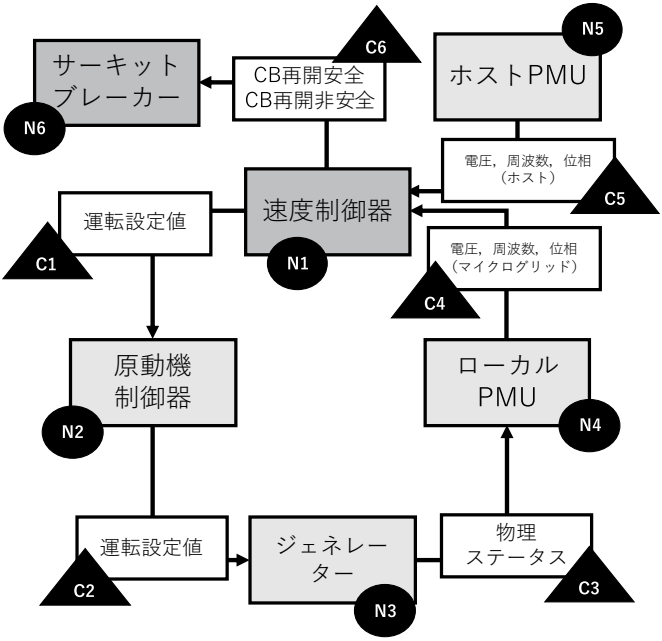


図 2.4-1　機能コントロールストラクチャー図

2.4.3.3. Step 1

STPA-SafeSec の Step 1 では、機能 CSD からトーマス博士が提案した拡張 Step 1［6］を用いて、UCA を識別する。ここで、コントローラーである速度制御器（N1）が参照する変数は、Δ Xm（t（）電圧差：ホストとローカルの電圧差、値は制限内、制限外）、Δ ω（t）（周波数差：ホストとローカルの周波数差、値は制限内、制限外）、Δ φ（t）（位相差：ホストとローカルの位相差、値は制限内、制限外）、Stc（b　サーキットブレーカーの状態、値は開、閉）の 4 変数であり、速度制御器は、Csp（原動機制御器（N2）への指示、値は運転範囲内、運転範囲外）、Ccb（サーキットブレーカー（N6）への STPA-SafeSec Step 2a ではじめに、機能 CSD を物理 CSD へ詳連絡、値は CB 再閉安全、CB 再閉非安全）の 2 つのコントールアクションを指示する。

STPA-SafeSec は拡張 Step 1 を採用しているため、Step 1 分析結果の記述形式が、標準的な記述形式［Leveson2012］［IPA2016］と異なる。この事例の UCA1 は、速度制御器のコントールアクション Ccb ＝ CB 再開安全とハザードへ至る条件Δ Xm（t）＝制限外の組み合わせに対し、ガイドワード Providing（Anytime）、Too early、Too late のときにハザード（H1、H3）へ至ると記述されている（すなわち UCA1 には、Too early と Too late が一つのガイドワード

であるとすれば、2 つの UCA がまとめられている）。

速度制御器からのコントールアクションに対する Step 1 の結果は以下の通りである：

- UCA1：ブレーカーが解放状態のとき、電圧差が制限外であるにもかかわらず、サーキットブレーカーへ CB 再開安全を Providing,Too early,Too late で指示（H1、H3）
- UCA2:ブレーカーが解放状態のとき、周波数差が制限外であるにもかかわらず、サーキットブレーカーへ CB 再開安全を Providing,Too early,Too late で指示（H1、H3）
- UCA3：ブレーカーが解放状態のとき、位相差が制限外であるにもかかわらず、サーキットブレーカーへ CB 再開安全を Providing,Too early,Too late で指示（H1、H3）
- UCA4：運転範囲外の設定値を原動機制御器に指示（H2）
- UCA5：ブレーカーが解放状態のとき、運転範囲内の設定値を原動機制御器に Too late、Not で指示（つまり、設定値の更新が行われない）（H3、H4、H5）

2.4.3.4. Step 2a：物理 CSD の構築

STPA-SafeSec Step 2a でははじめに、機能 CSD を物理 CSD へ詳細化する。物理 CSD は機能 CSD をアーキテクチャレベルで実現した記述である。図 2.4-2 は図 2.4-1 の機能 CSD を基に作成した物理 CSD である。

元の事例では、物理 CSD の要素（node）は Nn の形で、接続は Cn の形で表現される。また両 CSD の要素間には対応が付けられる。本節では、機能 CSD と物理 CSD 間の対応の理解性向上のために、機能 CSD 内の Nm と対応する物理 CSD 内のコンポーネントは Nm-n と表記する。なお、機能 CSD と物理 CSD の要素は多対多対応のため、Nm-n と Nm′-n′ が同じ要素を表すことがある点には注意が必要である。

機能 CSD と物理 CSD の対応の一部を示す。N1（速度制御器）は、N1-1（速度制御器 CPU）、N1-2（アナログ・デジタル変換器）と N1-3（N1-1 と N1-2 間の USB 接続）により構成される。また C5（ホスト電圧）は、C5-3（ホスト電圧）、C5-2（ファイアウォール）、C5-1（スイッチ）、C5-4（ホスト電圧、ローカル電圧）により構成される。

機能 CSD と物理 CSD の対応付け後に、続いて、安全にかかわる抽象的ハザードシナリオ（この事例では System Flaw）を特定している（STPA-SafeSec X）。抽象的ハザードシナリオは、標準的 STPA［Leveson2012］の安全にかかわる HCF ヒントを参考に、特定される。この事例では、以下の 6 つの抽象的ハザードシナリオを特定している。F1：速度制御器は電圧が制限内と誤認識、F2：速度制御器は周波数が制限内と誤認識、F3：速度制御器は位相角が制限内と誤認識、F4：原動機制御器は運転範囲外の設定値を受け取る、F5：速度制御器は設定値変更が要求されていないと誤認識、F6：遮断機制御器は「CB 再開安全」という誤情報を受信。

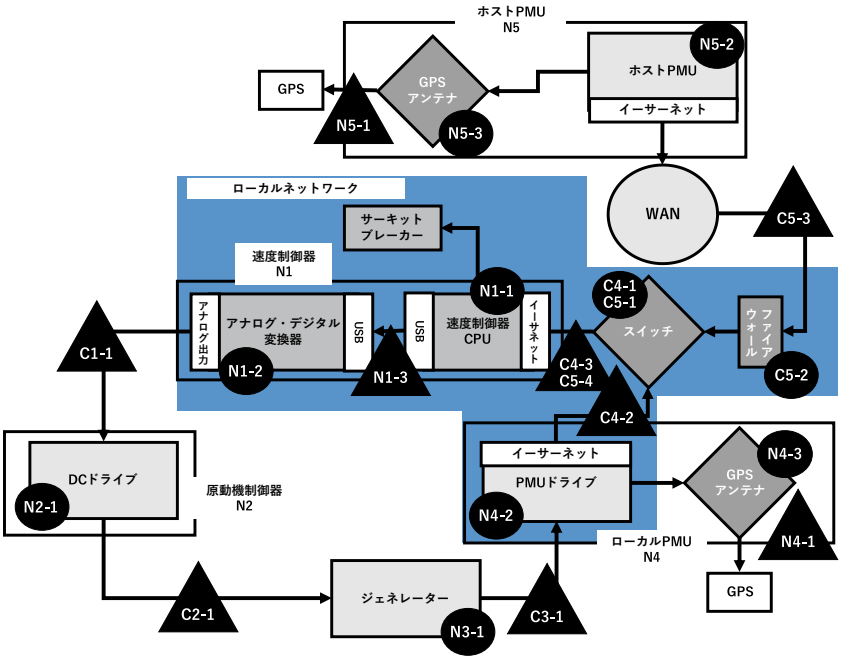


図 2.4-2　物理コントロールストラクチャー図

2.4.3.5. Step 2b：制約の詳細化

標準的 STPA では、このステップでハザードシナリオを導出する。他方 STPA-SafeSec では、ハザードシナリオ導出の前に、機能 CSD の要素へ安全・セキュリティ制約を課し、機能 CSD と物理 CSD の対応に基づき、安全・セキュリティ制約を物理 CSD の要素に割り振っている。この安全・セキュリティ制約を破る要因が HCF となる。

物理 CSD の要素に対し、安全・セキュリティ制約を課す利点としては、例えば以下の利点が挙げられている。物理 CSD のコンポーネントとして GPS が使用されていることが決まれば、GPS の既知の脆弱性としてスプーフィング（spoof）とジャミング（jam）が知られているため、これらに対するセキュリティ制約（CSTR-A-1、CSTR-A-2）を要素 N4-3、N4-1 に課す必要があることが分かる。しかし、機能 CSD では GPS が使用されるか否かは決定されていないため、これらの制約を課すべきか否かは決定できない。

STPA-SafeSec Step 2b では、はじめに、安全・セキュリティ制約と機能 CSD の要素を対応付ける。このとき、安全制約として識別済ハザードを用い、セキュリティ制約としてリスト 1、2 の制約を用いる。正確にはハザードやリスト中記述の否定形が制約である。

次に STPA-SafeSec Step 2b では、Step 2c で策定するハザードシナリオの理解容易性を高めるため、機能 CSD の制約を物理 CSD へ詳細化する。この事例では、幾つかのデバイスに対して制約の詳細化が示されているが、本節では速度制御器（N1）に対する制約の詳細化のみを紹介する。

はじめに、ハザードより安全制約を導出し、リスト 1、2 よりセキュリティ制約を導出する。速度制御器に対しては、H1（非同期での系統併入）、H2（電力機器の運転制限外での運用）、H3（電

力品質指標の逸脱）と H5（地域の電力需要への対応不可）が課される。次に、これらの安全制約とセキュリティ制約を、速度制御器の構成要素である、速度制御器 CPU N1-1、アナログ・デジタル変換器 N1-2 と USB 接続 N1-3 に割り当てる。ここでは、H1、H3 と H5 は速度制御器 CPU に、H2 はアナログ・デジタル変換機に割り当てられる。

2.4.3.6. Step 2c ハザードシナリオ策定

トップレベルのハザードシナリオからハザードシナリオを詳細化していく。トップレベルのハザードシナリオ（シナリオ 1）は Step 2a で識別した抽象的ハザードシナリオであり、それに関連する UCA（Hazardous Control Action）と機能 CSD のコンポーネント及び物理 CSD のコンポーネントから構成される。次に、トップレベルのハザードシナリオに含まれる各層の要素に着目し、ハザードシナリオを詳細化していく。このとき、詳細化されたハザードシナリオの HCF に対応する制約を合わせて記述する。下位シナリオ（シナリオ 1.1 以降）は、機能 CSD のコンポーネント、物理 CSD のコンポーネントに加え、安全制約、セキュリティ制約から構成される。

トップレベルのハザードシナリオから詳細化して策定されたハザードシナリオたちは、木構造を成す。木構造の上位ノードはより抽象的ハザードシナリオが対応する。すなわち、あるノードの子ノードには当該ノードに割り当てられたシナリオのサブシナリオが付される。このような木構造にすることで、あるハザードシナリオへの対応策は、木構造におけるそのノードの下位ノードの対応策になる。

本節では、一部のハザードシナリオのみ紹介する。

シナリオ 1：速度制御器は、（ローカル・ホスト間の）電圧差が制限値内と誤認識する。「ハザード：H1（非同期での系統併入）、H3（電力品質指標の逸脱）、抽象的ハザードシナリオ：F1（電圧差が制限内と誤認識）、UCA：UCA1（ブレーカーが解放状態のとき、電圧差が制限外であるにもかかわらず、サーキットブレーカーへ CB 再開安全を Providing,Too early,Too late で指示（H1、H3））、機能 CSD 関連コンポーネント：N1、N4、C4、N5、C5、物理 CSD 関連コンポーネント：N1-1、N4-2、C4-2、C5-1、C5-2、N5-2、C5-3、C5-4）」

シナリオ 1.1：速度制御器（N1）は、正しいフィードバックを間違って認識する。「機能 CSD 関連コンポーネント：N1（速度制御器）、物理 CSD 関連コンポーネント：N1-1（速度制御器 CPU）、安全制約：デバイス（速度制御器 CPU）とアルゴリズムの信頼性、アルゴリズムの正しさ、セキュリティ制約：CSTR-I5 Measurement injection（フィードバック（以下 FB）信号へのインジェクション攻撃）、CSTR-I7 Measurement manipulation（FB 信号の操作）」

シナリオ 1.2：速度制御器は、ホスト PMU からの間違った信号を受け取るが、それを正しいと認識する。「機能 CSD 関連コンポーネント：N1、N5、C5、物理 CSD 関連コンポーネント：N1-1、C5-1、C5-2、N5-2、C5-3、C5-4（注意：C4-3 と同じ対象を指す）、安全制約：N5 の信頼性、セキュリティ制約：CSTR-I5（FB 信号へのインジェクション攻撃）、CSTR-I7（FB 信号の不正操作）」

シナリオ 1.2.1：ホスト PMU が間違った FB 信号を送る。「機能 CSD 関連コンポーネント：N5、物理 CSD 関連コンポーネント：N5-2、安全制約：N5-2 の信頼性、セキュリティ制約：CSTR-I5（FB 信号へのインジェクション攻撃）、CSTR-I7（FB 信号の不正操作）、N5-2 への脆弱性攻撃成功（Successful exploit）」

シナリオ 1.2.2：リモート PMU からの正しい FB 信号が、ホスト電圧（C5）で改ざんされる、またはインジェクション攻撃される。N1-3 の通信は正常であるとする。「機能 CSD 関連コンポーネント：C5、物理 CSD 関連コンポーネント：C5-3、N5-1、N5-2、安全制約：なし、セキュリティ制約：CSTR-I5（FB 信号へのインジェクション攻撃）、CSTR-I7（FB 信号の不

正操作）」

シナリオ 1.2.3：リモート PMU からの正しい FB 信号が、ホスト電圧（C5）で改ざんされる、またはインジェクション攻撃される。N1-3 の通信は異常だが受け入れられるとする。「機能 CSD 関連コンポーネント：N1、C5、物理 CSD 関連コンポーネント：N1-1、C5-3、N5-1、N5-2、安全制約：なし、セキュリティ制約：CSTR-I5（FB 信号へのインジェクション攻撃）、CSTR-I7（FB 信号の不正操作）」

2.4.4. 今後の課題

本節では、STPA をベースに脆弱性分析を行う際の課題として、Step 2 で用いる HCF 導出のヒントに関する課題を述べる。また STPA では分析時に妥当な仮定を置かずに分析を実施すると、分析対象が肥大化したり、分析者により分析結果が大きく異なったりといった状況に陥りがちである。そこで STPA 一般の課題として、分析時の仮定について述べる。

2.4.4.1. セキュリティにかかわる HCF ヒントに関する課題

STPA-SafeSec では、標準的 STPA Step 2 で利用される HCF のヒントに加え、セキュリティにかかわる HCF を導出するために、リスト 1、2 にあるヒントを利用する。他方、STAMP Workbench や SafetyHAT［SafetyHAT2018］といった STAMP/STPA 支援ツールでは、HCF ヒントを分析対象領域に依存して適切に変更でき、更に分析者が独自に編集できる。例えば、［Leveson2013］にある HCF ヒントは機械のコントローラーを想定しており、機械のコントローラーに対しては適切なヒントであるが、人間のコントローラーに対しては異なるヒントのほうが HCF を導出しやすいであろう。従って、セキュリティにかかわる HCF ヒントも、適宜修正・変更することで、HCF を導出しやすくなると考えられる。

STPA-SafeSec で採用されているセキュリティにかかわる HCF ヒント以外にも、例えば、セキュリティにかかわるヒントとして STRIDE［MS2018］の利用も考えられる。

2.4.4.2. 分析時の仮定に関する課題

STAMP/STPA で解析を行うときに一般に難しい点は、どの抽象度とどの仮定のもとでコントールストラクチャや HCF の設定を行うかであろう。モデル化を行う際にはある程度のドメイン知識を暗黙裏に仮定する。この仮定の妥当性は解析とモデル化を繰り返すことにより補強するのが現在の標準的な手順である。この際に known-unknowns や unknown-knowns などの仮定の境界上の事項［Sebastian2014］を意識することが強く望まれる。

STPA-SafeSec では物理 CSD の導入により unknown-knowns の気づきに貢献している。例えば、「物理 CSD のコンポーネントとして GPS が使用されていることが決まれば、GPS の既知の脆弱性としてスプーフィング（spoof）とジャミング（jam）が知られている」というのは「GPS の既知の脆弱性としてスプーフィング（spoof）とジャミング（jam）が知られている」というドメイン知識を解析者の unknown-knowns から known-knows への変換に寄与しているとみなすことができる。

一方、known-unknowns については次のような対策が取れる。known-unknowns については典型的には定性的要因が分かっているが、定量的な値が不明であるという特徴を持つことが多い。その場合は値に関する変数を不定値とみなしたり、あるいは統計的量として捉えることによりモデル化できることがある。その場合はそれぞれに適した数理モデルや解析手法の活用が可能となる。

2.4.5. まとめ

本節では STAMP 海外事例として STPA-SafeSec を紹介した。セキュリティ侵害が安全性を脅かすといった事象を分析するためには、安全性とセキュリティを統合して分析できることが肝要である。

STPA-SafeSec は安全性とセキュリティを統合して分析するために、標準的 STPA の CSD 階層や HCF と同等ではあるが、機能 CSD とデバイスに着目した物理 CSD を階層的に用いることで安全に関わるセキュリティの脆弱性を可視化し、さらに、STPA Step 2 で使用する HCF ヒントをセキュリティ拡張したという特徴を持つものである。トップダウンに分析し「セキュリティ侵害が安全性を脅かす事象」につながるセキュリティ（ハザード）シナリオを特定することができる。紹介した適用事例で分かるように、安全性とセキュリティに係るハザードシナリオを木構造として扱うことで、例えば、上位の安全性対策が下位のセキュリティ対策を兼ねるといった状況を見つけやすくしている。

なお、セキュリティにかかわる HCF ヒントはリスト 1、2 のヒント以外にも考えられ、既存のセキュリティ分析手法を STPA-SafeSec の手順に組み込んで、分析結果を充実させるといった点にも改善余地はあると考えられる。

STPA の基本思想は、従来の信頼性工学的方法である「システム要素の故障を全て分析しそれを最小化する」ことに対して、安全制御行動の網羅的な分析によって「システム要素間の相互作用に起因する事故を防ぐ」という安全制御工学的手法を用いるというパラダイムシフトである。セキュリティに関しても、システムへの侵入・改ざんを全て分析して防ぐという従来の考え方から、侵入・改ざんも相互作用に影響を及ぼす要因とし、非安全・非セキュアな制御行動がシステムの事故を引き起こすことがないような対策（セキュリティ・安全制約）を網羅的に統合して分析する方法論が必要である。

安全確保のため緊急避難口を設けたことで犯罪者の侵入経路を増やしてしまうというように、安全性とセキュリティが競合するケースもあり、安全性対策とセキュリティ対策を統合して考察できる STPA 手法が今後ますます重要になる。

3. 第 2 回 STAMP ワークショップ in Japan について

3.1. 開催概要

1 年前に福岡市で開催された第 1 回に引き続いて、第 2 回 STAMP ワークショップ in Japan は、場所を東京都にある慶応義塾大学三田キャンパスに移し、2017 年 11 月 27 日から 3 日間にわたって開催された。

4 カ国から延べ 181 名（前回は 117 名）の参加者が集まり、初日に米国 MIT からの基調講演 / チュートリアル、欧州 STAMP ワークショップ（ESW）からの招待講演が順に行われ、その後、一般講演として産業界から 13 件、学術界から 11 件、合計 24 件（前回は 16 件）の発表が行われた。あわせて、ポスター展示が 2 件（前回は 0 件）あった。また、IPA の STAMP 支援ツール「STAMP Workbench」が紹介され、期間中、デモ展示された。

概略日程は次のとおり：
2017 年 11 月 27 日（月）
9:35-9:45　実行委員長挨拶
9:45-11:45　基調講演 / チュートリアル
12:45-14:15　基調講演 / チュートリアル
14:30-15:30　招待講演
15:40-17:00　Overseas and Tools Session（3 件）
17:00-18:00　挨拶、ツールのデモ展示
2017 年 11 月 28 日（火）
9:00-10:50　ショートセッション（4 件）、標準セッション（1 件）
11:00-12:30　標準セッション（3 件）
13:30-16:00　標準セッション（5 件）
16:10-17:40　標準セッション（3 件）
2017 年 11 月 29 日（水）
9:00-11:30　標準セッション（5 件）
11:30-12:00　クロージング

なお、STAMP ワークショップに関する Web サイトを IPA/SEC が運営しており、その URL は次のとおり：
（日本語）https://www.ipa.go.jp/sec/our_activities/stamp.html
（英語）https://www.ipa.go.jp/english/sec/complex_systems/stamp_workshop.html
今回も、MIT と欧州のワークショップと連携がとられて開催されており、海外活動の詳細については、SEC Journal 52 号［IPA2018-2］を参照されたい。

3.2. 発表概要

今回の講演資料は、そのほとんどが IPA/SEC Web サイトに掲載されている（https://www.ipa.go.jp/sec/events/20171127.html）。この節では、概要のみを記載する。

（1）チュートリアル

前回同様、MIT の Dr. John Thomas が担当し、3 件のチュートリアルを行った。その概要は次のとおり：

T1：STAMP and STPA Introduction
多くの事例を用いて、STAMP の必要性と特徴を説明し、STPA 分析の概要を紹介。

T2：STPA Exercise
米国 DoD（Department of Defense）Access control barrier の題材を用いて、STPA 手順を演習を交えて順に詳細解説。

T3：Advanced STPA Topics
前回同様に、APA（Automated Parking Assist：自動駐車支援機能）に対する STPA 分析結果を解説。

（2）招待講演

European STAMP Workshop Board から Dr. Nektarios Karanikas（アムステルダム大学）が来日し、次の招待講演を行った :

G1：Situations of STAMP in Europe
欧州における STAMP の実情に関して、コミュニティ、ワークショップ、推進委員会、教育と研究などについて紹介。

（3）産業界からの一般講演

I1：STAMP/STPA の自動車向けの活用ガイド -JASPAR 機能安全 WG 活動成果より -（JASPAR）
仮想的な電動パーキングブレーキシステムに STPA を適用し、ISO 26262 との差分分析を行い、開発現場向けの活用ガイドを作成した。

I2：システムモデルを用いた STAMP/STPA 試行の事例紹介（日立産業制御ソリューションズ）
仮想的なドライバー異常時安全停車システムを題材として、開発初期に作成される、抽象度の高い SysML モデルを活用して STPA 分析を実施した。

I3：国際安全規格における STAMP/STPA 適用可能性の考察（東芝）
STPA が国際安全規格において、従来の安全分析手法との違いを踏まえて、どの開発工程に適するかを、規格、従来手法、STPA 適用事例などを調べて考察した。

I4：自動運転系の安全・セキュリティ解析のための自動化ヒューマンファクタに基づく STPA ガイドワードの提案（日立製作所）
自動化システムの監視制御系において、ヒューマンファクタに基づくハザード誘発要因を識別するためのガイドワードを提案し、自動運転系などに適用を試み、その有用性を確認した。

I5：STAMP による閉電路制御式踏切制御システムの安全性評価（京三製作所）
従来方式と新方式による踏切制御システムに STAMP/STPA を適用し、STAMP による安全性評価の有効性を確認した。

I6：STAMP/STPA の鉄道信号システムへの応用と拡張（東日本旅客鉄道）
STAMP/STPA を信頼性についても拡張し、鉄道信号システムにおけるこれまでの事故や不具合などを評価した。

I7：STAMP/STPA を用いた踏切障害物検知システムの安全性分析（東日本旅客鉄道）
踏切障害物検知システムに STAMP/STPA を適用し、ハザード誘発要因を識別すると ともに、設計上の安全制約を抽出した。

I8：STAMP/STPA による踏切制御システムの安全性要求分析（東日本旅客鉄道）
新たに試作した構内踏切制御論理に STAMP/STPA を適用して安全解析を実施し、安全要求事項の抽出を試みた。

I9：意図・要求記述レベルの STAMP/STPA 手法（JASA）
仮想的な電動アシスト自転車開発を題材とし、開発に関する意図と要求の記述をもとに STPA 分析を行い、その結果、意図の実現に適する推奨策を導出した。

I10：Extending STPA をベースとしたプロセスモデル抽出の工夫（日本ユニシス）
Extending STPA において 6W3H の視点からコンテキストを抽出する工夫を提起し、自動運転制御システムを題材にそれを試行した。

I11：STAMP/STPA を用いた Cyber-Physical Systems の検証（日本ユニシス）
複雑になるコントロールストラクチャーを整理する方法と、そのコントロールループの安全な状態をモデル検査により検証する手法を提起し、例題として Vehicle-to-Device（V2D）システムを利用した車の交通制御システムに適用した。

I12:STAMP/STPA を用いたリスクマネジメントフレームワークの提案（電通国際情報サービス）
既存のリスク抽出・管理方法に STAMP/STPA を用いることで、メカ観点でのリスクだけでなく制御観点でのハザードを共に抽出・管理できることがわかり、技術的なリスク管理だけでなく、開発日程面でのリスク管理もできるフレームワークを構築した。

I13：STAMP を STAMP してみた！（オムロンオートモーティブエレクトロニクス）
観光地や駅などにおかれている記念スタンプを題材に、STPA の基本手順を辿り、課題や利点などを考察した。

（4）学術界からの一般講演

A1：Integration of Security into CAST（Zurich University of Applied Sciences）
セキュリティ侵害を分析する手法を CAST プロセスに統合させ、代表的なセキュリティ事故に適用してその有用性を確認した。

A2：STAMP ベース・ハザード分析ツールの紹介（仙台高等専門学校）
STAMP/STPA 分析を支援するツールとしては、専門ツールや流用できる関連ツール等があり、それらの特徴を調べた。

A3：IPA が提供する STAMP 支援ツール i-STAMP（開発コード）（IPA/SEC）
STAMP/STPA 分析作業を支援して、思考に専念できるように、IPA はツールを年度末リリースを目標に開発している。

A4：プロジェクト管理における動機付けに着目した STAMP/STPA の適用（長崎県立大学）
プロジェクト管理の制御構造を STAMP によってモデル化し、STPA 分析によって制御構造

の適正化や運用指針の策定を行うアプローチを考察した。

A5：Freedom from interference に着目した STAMP/STPA の適用（長崎県立大学）
AADL（Architecture Analysis & Design Language）を使用するシステムアーキテクチャの記述や分析に STAMP/STPA を適用した。

A6：STAMP/STPA 事例の振り返りと GSN を用いた STPA プロセスの説明支援（日本大学）
STAMP/STPA 手順を行う上での課題及び工夫を抽出し、実施する際に重要なポイントをまとめ、GSN（Goal Structuring Notation）により表記した。

A7：STAMP/STPA を用いた自動運転システムのリスク分析 - 高速道路での合流 -（愛知工業大学）
自動運転システムと運転者の連携に着目して、特に危険が多いと考えられる自動運転中の高速道路の合流に STAMP/STPA を適用した。

A8：STAMP/STPA による多目的バッチプラントのリスク解析（名古屋工業大学）
マルチパーパス / マルチバッチといったフレキシブルな運用を行う化学プラントに対して、原料や洗浄液のコンタミネーションなどに注目して STAMP/STPA を適用した。

A9：電動アシスト自転車を対象としたハザード分析 /STAMP・STPA と数値シミュレーションの特徴比較（会津大学）
電動アシスト自転車という人間・機械の協調制御システムを題材に、STAMP/STPA によってどのようなハザード要因が分析可能かを検討し、SimuLink に代表される数値分析によるハザード分析との比較により、STAMP/STPA の有用性を示した。

A10：コントロールストラクチャーの状態遷移仕様とガイドワードを用いたシミュレーションによる STAMP/STPA の非安全コントロールアクションの識別方式の提案（大阪工業大学）
コントローラー及び被制御プロセスの仕様を状態遷移仕様で表し、ハザードとなる状態への到達可能性をガイドワードに則ってシミュレートすることで、UCA を半自動的に識別する方式を考察した。

A11：IoT/ 深層学習利用における STAMP と HAZOP についての研究（名古屋市工業研究所）
STAMP に注目して、HAZOP や FRAM 等の他手法との相違を整理した。

3.3. 傾向と期待

一般講演の傾向を明らかにするために、発表内容に応じて次の 4 種別で分類してみた :

試行事例：STAMP の適用可能性を評価するために、試験的に適用を試みた事例の紹介
活用事例：現在使用している、開発している、又は研究している製品やシステムなどに STAMP を適用した結果の紹介
手法解説：仮想的な題材をもとに、STAMP を適用する手順、適用するときの工夫や考慮事項などを解説するもの
手法改善：標準的な STAMP 関連手法の拡張や詳細化を研究し、その改善を提案するもの

分類の結果を表 3.3-1　一般講演の分類に示す。この表から次に列記する傾向が読取れる：

- 学術界からの発表は、手法解説に偏っているが、産業界からはどの種別にもほぼ同じ件数の発表がある。
- 試行事例の件数は産業界と学術界でほぼ同じである。産業界は、開発済みの制御システムを題材としているが、学術界は、制御システム以外にも目を向けている。
- 活用事例と手法改善は、産業界から多く発表されているが、開発中又は今後のシステムに適用してみようという産業界の意欲が感じられる。
- 産業界から手法改善が多かったのは、適用プロセスの標準化を目指して、手法の定型化を求めていることによると考えられる。

表 3.3-1　一般講演の分類

	産業界からの一般講演	学術界からの一般講演
試行事例	I5　I6　I7	A4　A5　A7　A8
活用事例	I8　I9　I12	
手法解説	I1　I3　I13	A1　A2　A3　A6　A9　A10　A11
手法改善	I2　I4　I10　I11	

今後、試験的な試行事例から一歩進んで、STAMP 適用時の課題の解決策、関連手法の改良、深化等を提案する発表が増え、実際の開発案件に活用されていくことを期待してやまない。

4. 安全性モデリングと STAMP/STPA、その最新ツール紹介

システム開発は、近年急速に高機能化、複雑化している。更に、IoT、つながる社会の到来によって「相互作用」にも目を向けなければ、ますます安全性の確保が難しくなってきている状況にある。MBSE（Model Based Systems Engineering）が注目されているのも、そういった俯瞰的な視点が必要となっていることの表れである。本章では、初めにこの中でのモデルの意味を考察する。次に、現在ハザード分析手法として脚光を浴びている
STAMP/STPA0 と支援ツールについて紹介する。

4.1. モデルとは何か

モデルを定義することは難しいが、筆者なりに定義をすると、「現実世界の対象物を、ある目的で捨象し、その目的下で扱いやすくした抽象物。」ということになる。現実世界は多くの場合複雑である。人間が複雑な問題を扱う場合、着目する「目的」に応じて、不要な情報を意図的に捨て去って（捨象）、本質情報を単純化して表出させる（抽象）することで、その目的に人間の知的活動の焦点を絞ることができるようになる。ここで「抽象」と「捨象」はちょうど作用と反作用のような関係になっている。モデル化に使う言語（表現形式）としては、数式、プログラミング言語、ブロック図、図面、UML/SysML のようなある程度フォーマルなモデリング言語がある。あるいは、物理的な整形物や木型（モックアップ）などもモデルに含まれるだろう。

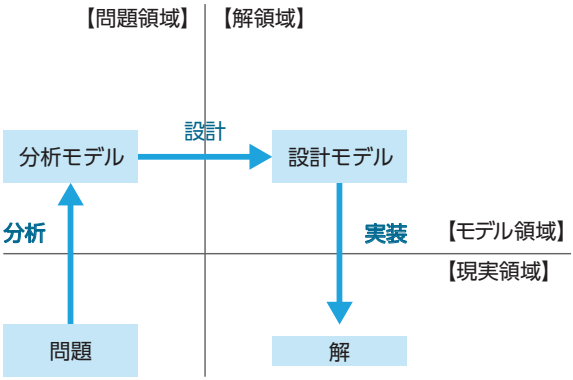


図 4.1-1　モデルの役割

図 4.1-1 はモデルの役割を描いたものである。問題領域を解領域に実装することを最終成果とする場合、現実の複雑なものを、いったんモデル空間に「抽象化」（そのためにほかを捨象）する。この行為を分析と呼ぶ。抽象・捨象（何を拾って何を捨てるか）は目的によって異なるが、モデルを使うことによって注目する問題がより鮮明になり、設計を導くための導線になる。そして、それをモデル領域でいったん解決したものを設計モデルと言い、それを実装したものが実際の解、すなわちシステムとなる。

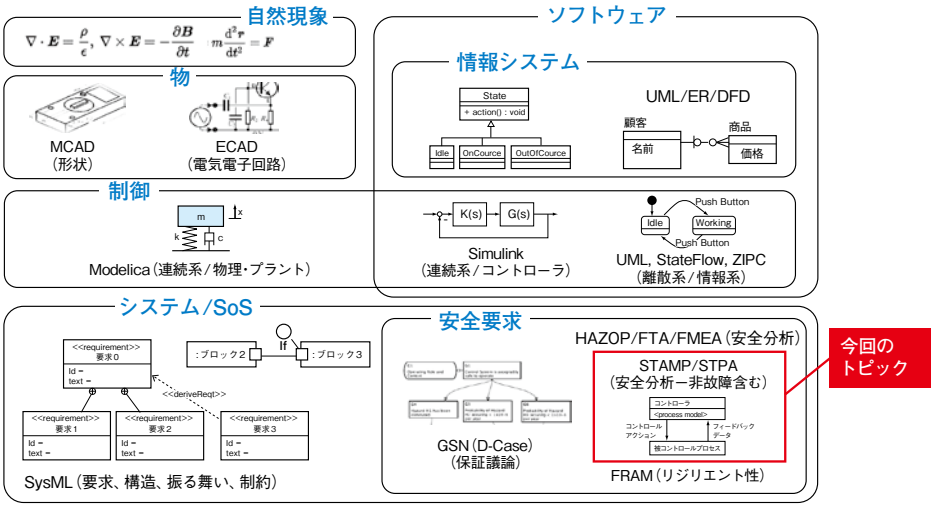


図 4.1-2　システムのモデリング言語

4.1.1. 様々なモデリング言語

どのような目的で何を抽象・捨象の対象にするのか、という観点から、モデリング言語は多岐にわたる。とくに専門分野の特化から、現在のシステム開発ではモデルの役割は非常に広くなってきている。以下は、システムを分析・設計する上でよく使われるモデルをカテゴリ化したものである。

自然現象の記述に数学を利用し、解析的・代数的に現象を記述したものが最初のモデリングであろう。ここでの目的は物理量とその予測である。更に、自然現象を人間の役に立つように応用した「エンジニアリング」分野で、とくにコンピューターを利用したモデリングが始まった。機械や電気的な「モノ」をコンピューター上に実現しようとした CAD の歴史は古く、実物でなくコンピューター上で構造物を表現して破壊実験をしたり、シミュレーションをしたり、意匠の確認ができるようになった。その後、ソフトウェア自身もプログラミング言語よりも高い抽象度で（例えば UML を使って）モデリングされるようになった。ソフトウェア自身のモデリングは、組込みシステムだけでなく企業の情報システム、データベースやワークフローとしても多く利用されている。

制御システムが各産業で重要になり、制御対象となるモノ（プラント）、制御ソフトウェア自身もコンピューター上でモデリングできるようになってきた。そして、エンジニアリング領域を横断的に、システム全体をモデリングしようという流れが生まれ、MBSE（Model Based Systems Engineering）が注目を集めている。例えば、そのモデリング言語の一つである SysML では、要求、振る舞い、構造、制約の視点から、システムを俯瞰的にモデリングできる。ここまで来ると、モデリングの目的はそのシステムのステークホルダごとに様々になる。構造的な視点、実現におけるコストの視点、要求の充足度の視点、などなど多岐にわたる。それぞれの視点ごとに最適な視点でモデルを提供する必要が出てきた。

その中でも、最近注目されているのが、安全解析の視点であり、中でもこれから開発が加速する、人とソフトウェアを内包した複雑工学システムの安全解析のためのモデリングツール STAMP/STPA である。

4.1.2. 安全解析のモデリング

従来から、安全解析手法として FTA、FMEA などが広く利用されている。FTA は「製品」（トップレベル）の起こり得る故障を想定し、「要素」にブレークダウンすることで原因を分析する。対して FMEA では、個々の「要素」（ボトムレベル）の故障モードから「製品」の故障を洗い出す。

これらの安全解析では、トップダウン、ボトムアップのアプローチの違いはあれど、最終的には故障が起こる潜在的な状況と個々の要素に注目し、故障の原因を分析してきた。しかし今回紹介する STAMP/STPA では、個々の要素が「非故障」であっても安全に影響を及ぼす「要素間」の関係性に注目して分析を行う。ここでのモデリングの目的は、満たすべき「安全制約」に関連する「コンポーネント」（要素）と、それらをつなぐ「相互作用」を取り出してそれらが形作る「構造」をあぶり出すことである。このように、現実を抽象・捨象して、システムの構成要素とその相互作用をモデリングし、安全解析を行っていくモデリング手法の一つとして、STAMP/STPA が注目されている。以下では更に詳しく解説をしていく。

4.2. STAMP/STPA と支援ツールの紹介

本節では、安全解析用モデルとして、従来のアクシデントモデルを拡張した新しいアクシデントモデルである STAMP と STAMP に基づくハザード分析手法 STPA［Leveson2012］に注目する。とくに、STAMP/STPA を適用する際のツール支援に着目し、STAMP/STPA 適用時の分析結果の様式や課題について述べ、その後、STAMP/STPA 支援ツールを紹介する。

STPA はシステマチックなハザード分析手法である。しかし、STPA は幾つかの単純であるが手間のかかる作業を必要とする。例えばコントロールストラクチャー図（Control Structure Diagram、以下 CSD）の記述では、コンポーネント間の接続にコントロールアクション（Control Action、以下 CA）を対応させたり、コンポーネント内部にプロセスモデルを記述したりといった STAMP 特有のデータ構造を記述する必要がある。

また、CSD 中の CA と Step 1 で分析する CA の対応付けも必要である。図表の解釈やデータ連携を人間が適切に補うことで、汎用的な図表作成ツールを用いて STPA を実施できる。しかし、専用ツールを用いて単純な作業を支援することで、分析者は本質的な分析に専念できるようになる。そこで、幾つかの専用ツールが公開されている。

4.2.1 項で STAMP/STPA を概説し、4.2.2 項で既存の STAMP/STPA 支援ツールを紹介する。更に、4.2.3 項で IPA/SEC で開発中の STAMP/STPA 支援ツールを紹介する。

4.2.1. STAMP/STPA 概説

本項では、文献［IPA2016］を基に STAMP/STPA の手順を簡単に解説する。とくに、各ステップでの出力（分析成果物）の典型的なデータ形式を文献［IPA2016］［Leveson2013］の例を基に紹介し、各ステップにおける課題について述べる。これにより、STAMP/STPA 支援ツールに求められる機能を洗い出す。

ドローイングツールを用いた CSD の記述、表計算ツールを用いた分析結果表の記述を念頭に各ステップの課題を洗い出す。どちらのツールも大変普及しているという利点があるが、反面 STPA 支援専用ツールではないためデータ連携が取れず、CSD や分析結果の修正時に生じる派生的な修正は人手により行う必要があるといった課題がある。

4.2.1.1. Step 0 準備 1

Step 0 準備 1 では、アクシデント、ハザード、安全制約を識別する。Step 0 準備 1 の入力は要求仕様書やドメイン専門家（の知識・知見）であり、出力はアクシデント、ハザード、

安全制約の一覧表（表 4.2-1）である［IPA2016］。

出力の一覧表では、アクシデント、ハザード安全制約間に関係があることを、同じ行に記載することで表している。一般にアクシデント、ハザード、安全制約間の関係は多対多であるため、複数個所に（例えば）同じアクシデントが現れることになる（表 4.2-1）。そのため人手により表を記述・管理する場合には、修正漏れに注意する必要がある。また STPA の後工程で、この一覧表中に記載したアクシデント、ハザードと安全制約を参照する場面が多々あるため、これらに番号を付けることは有効である。

表 4.2-1　アクシデント、ハザード、安全制約の一覧表

アクシデント ID	アクシデント（Loss）	ハザード ID	ハザード（Hazard）	安全制約 ID	安全制約（Safety Constraints）
A1	列車と人・車が踏切内で衝突する	H1	列車が在線中に踏切が閉まらない（警報が鳴らない）	SC1	列車が在線中は踏切が閉まらなければならない
A1	列車と人・車が踏切内で衝突する	H2	踏切遮断後、列車が在線中に踏切が開く（警報が鳴りやむ）	SC2	列車が在線中は踏切が開いてはならない
A2	踏切が開かず、交通が渋滞する	H3	列車が不在なのに踏切が閉まる（警報が鳴りだす）	SC3	列車が不在ならば踏切を閉じない
A2	踏切が開かず、交通が渋滞する	H4	列車が通過したのに踏切が開かない（警報が鳴りやまない）	SC4	列車が通過したら踏切を開ける

4.2.1.2. Step 0 準備 2

Step 0 準備 2 では、コントロールストラクチャーを構築する。Step 0 準備 2 の入力は要求仕様などであり、出力は CSD や表形式のコントロールストラクチャーである。図 4.2-1［IPA2016］は、責務とプロセスモデルを加えた CSD である。責務はコンポーネントに対する高抽象度の要求であり、プロセスモデルは CA 出力の判断に必要な変数とその値の組である。例えば、列車ドアコントローラーは、変数 " ドア位置 " が値 " 閉 " で、変数 " 列車位置 " が値 " プラットフォーム停車中 " のとき、ドア開命令を出す。

ドローイングツールで CSD を記述する場合、コンポーネントは単なる四角形である。従って、例えばコントローラーには責務とプロセスモデルを追加できるといった、コンポーネント種別による違いは人手により付ける必要がある。また、CSD 中の CA とプロセスモデルは次の Step 1 の出力である UCA 表中で参照される。このような連携は大変手間のかかる作業であり、人手で連携する場合には参照関係が壊れがちである。

分析対象は一般に大変複雑なので、CSD を抽象化することや、逆に CSD の一部を詳細化することは、分析対象のコントロールを理解するのに役立つ。他方、CSD 構築は適切な抽象度にするためのコンポーネント粒度統一作業を含むため、変更と修正を繰り返すことが多い。この修正作業により、ドローイングツールにより記述する場合には、高抽象度の CSD とその一部を詳細化した CSD の整合性を保つことは大変手間のかかる、かつ間違いやすい作業となる。

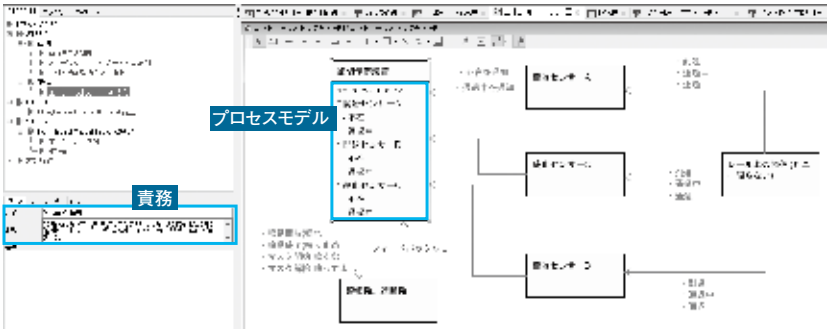


図 4.2-1　コントロールストラクチャー図

4.2.1.3. Step 1

Step 1 では、UCA（Unsafe Control Action、非安全制御動作）を抽出する。Step 1 の入力は UCA を導き出すための（STPA が提供する）4 つのガイドワード、Step 0 準備 1 出力のアクシデント、ハザード、安全制約の一覧表、Step 0 準備 2 出力の CSD であり、出力は UCA 一覧表である（表 4.2-2）。

UCA 一覧表の各行はそれぞれ CSD 中の CA に対応し、各列は 4 つのガイドワードに対応し、各セルには対応する CA とガイドワードを組み合わせた際に（もし至るならば）UCA へ至る条件と対応するハザードや安全制約が記載される。

表計算ツールを用いて UCA 一覧表を作成する場合には、CSD 中から CA を抜き出す際の漏れや、CSD 変更に伴う UCA 一覧表への影響反映忘れに注意する必要がある。また、UCA 一覧表中の UCA は Step 2 で参照するため、番号を付与すると参照が容易になる。

表 4.2-2　UCA 一覧表

No	コントロールアクション	Not Providing	Providing causes hazard	Too early / Too late	Stop too soon / Applying too long
1	鳴動開始	（UCA1-N-1） 警報が鳴らずに列車が踏切を通過する（踏切が閉まらない）［SC1］	列車が来ないのに警報が鳴る	（UCA1-T-1）警報鳴動する前に列車が踏切に到達する（閉まるのが遅く間に合わない）	開始指示が継続するので、列車通過後に鳴動停止指示が出ても鳴動し続ける
2	鳴動終了	列車が通過後も警報が鳴りやまない	（UCA2-P-1）列車が通過中に鳴動停止する［SC2］	（UCA2-T-1）列車が通過完了する前に鳴動停止する）SC2］	（UCA2-D-1） 列車通過後も鳴動停止指示が）SC1］
3	マスク開始	A, C を通過した列車が B に到達したときに再鳴動する	（UCA3-P-1） 列車が来ないのにマスク指示し、警報鳴動しない［SC1］（UCA3-P-2）反対側の開始センサーにマスク指示し、警報鳴動しない［SC1］	（UCA3-T-1）終止センサーへのマスク指示が遅れ、列車の当該センサー通過に間に合わないと、マスク指示が残り、対向列車が 2 本続いたときに警報鳴動しない［SC1］	（UCA3-D-1） 列車が反対側の開始センサー通過後までマスク指示し続けると、対向列車が来ても鳴動しない［SC1］
4	マスク解除	（UCA4-N-1） 反対側の開始センサーに除指示が出ず、対向列車が来ても鳴動しない（マスク指示後に列車が引き返す場合を含む）［SC1］	警報が再鳴動する	列車が B を通過完了前に出ると再鳴動する	解除が後続列車によるマスク開始指示と競合するとマスクされずに再鳴動する可能性がある

4.2.1.4. Step 2

Step 2 では、HCF（Hazard Causal Factor、ハザード誘発要因）を特定する。Step 2 の入力は、HCF 特定のためのヒント（抽象化されたハザード誘発要因の例）、Step 0 準備 2 で構築した CSD と Step 1 で作成した UCA 一覧表である。また Step 2 の出力は、ハザード要因の一覧表（表 4.2-3）とハザードシナリオである。表 4.2-3 のハザード要因の一覧表は、Step 1 で識別した UCA ごとに作成される。このハザード要因の一覧表の各行は一つの HCF に対応し、対応する HCF ヒントとハザードシナリオを記載する。更に各行には複数のハザードシナリオを記載できる。［IPA2016］では、表 4.2-4 の形式のハザード要因の一覧表を紹介している。表 4.2-4 の各行は一つの UCF であり、各列は HCF ヒントワードで、対応するセルに HCF またはシナリオを記載している。複数の UCA に対する表 4.2-3 のデータをまとめ、適切な形式に変換することで、表 4.2-4 が得られる。このような一覧表を用いることで、HCF の抜け、すなわち、その網羅性の確認に役立てることができる。

表 4.2-3　ハザード要因の一覧表

ID	HCF	ヒントワード	シナリオ
HCF1-N-1-1	踏切通過後に引き返す列車向け制御が不適切	（8）不適切、有効でない欠けたコントロールアクション	A から来た列車が C を通過した後、連結を切り離して、後部車両が A 方向に引き返す A から来た列車が C を通過した後、A 方向に引き返す
HCF1-N-1-2	鳴動停止継続により次の鳴動指示と競合	（8）不適切、有効でない欠けたコントロールアクション	A から来た列車が C を通過して B をマスクした後、B と C の中間で停止。救援列車が反対方向から侵入してセンサー B を通過。A 方向に進行する
HCF1-N-1-3	センサー A が故障して A から踏切制御装置への通知が欠落	（9）プロセスへの入力が欠けているか間違っている	センサー A が故障して A から踏切制御装置への通知が全く届かない センサー A が不電導物（葉っぱなど）に覆われて、車輪経由の検知電流が流れず、列車到達を検知できない

表 4.2-4　ハザード要因の一覧表［2］

	❶上位からの指示や外部情報の誤り・欠落	❷Control action が不適切・無効・欠落	❸動作の遅れ	❹プロセスへの入力の誤り・欠落	❺意図しない、または範囲外の外乱	❻不十分な制御・アルゴリズム
（UCA1）警報が鳴らずに列車が踏切を通過（踏切が閉まらない）		・踏切通過後に引き返す列車向け制御が不適切 ・鳴動停止継続により次の鳴動指示と競合		・センサー A が故障して A から踏切制御装置への通知が欠落		
（UCA2）鳴動前に列車が踏切に到達（閉まるのが遅い）			・センサー A が故障して A から踏切制御装置への通知が欠落			・センサー A が故障して A から踏切制御装置への通知が欠落
（UCA3）列車が踏切を通過する前に鳴動停止（開くのが早い）					・列車が A を通過後、踏切に到達する前に、C が外乱により短絡する	

	❶上位からの指示や外部情報の誤り・欠落	❷Controlaction が不適切・無効・欠落	❸動作の遅れ	❹プロセスへの入力の誤り・欠落	❺意図しない、または範囲外の外乱	❻不十分な制御・アルゴリズム
（UCA4） 不正なマスク開始指示が出て、列車が来ても警報鳴動しない		・踏切制御装置の状態管理が不適切				・踏切制御装置の状態管理が不適切
（UCA4） 不正なマスク開始指示が出て、列車が来ても警報鳴動しない			・超高速列車に対応できずにマスク解除の指示遅れ	・レール上の物体による外乱		・制御装置の処理に問題があり、マスク解除の指示遅れ
（UCA6） マスク開始指示漏れ	・誤った外部入力（外乱）でマスク解除漏れ	・制御装置の処理遅れでマスク解除漏れ	・状態制御誤りでマスク解除漏れ	・不正な外部入力によりマスク解除漏れ		・非定常運行への対策漏れでマスク解除漏れ

表 4.2-3、表 4.2-4 では、ハザード要因をハザード要因の一覧表で、ハザードシナリオを文章で記述している。これ以外にも、ハザード要因はリスト形式で、ハザードシナリオはアノテーション付きコントロールループ図（Control Loop Diagram、以下 CLD）で、それぞれ記述されることもある。

表計算ツールでハザード要因の一覧表を作成する場合には、UCA 一覧表中の UCA とハザード要因の一覧表の UCA の対応付けは人手で行う必要があり、漏れや修正による影響の反映といった点に注意する必要がある。また、ハザード要因の一覧表とシナリオを独立して記述する場合には、それらの対応付けも必要となる。

HCF は、コントローラーがソフトウェアか人間かといった条件や分析対象のドメインにより異なる。従って、HCF のヒントを適切に変更・提供することで、HCF 発想が容易になる。

STAMP/STPA 支援ツール SafetyHAT［SafetyHAT2018］ではドメインに特化したヒントを提供しており、ユーザによるヒントの編集も可能である。

4.2.2. STAMP/STPA 支援ツール

本項では、既存の STAMP/STPA 支援ツールを紹介する。また STAMP/STPA の専用支援ツールを紹介する前に、他ツールによる STAMP/STPA 支援について紹介する。文献［PSAS2018］や［Krauss2015］では STAMP ベースのツールとして、モデル検査連携を含む多彩な拡張機能を持つ XSTAMPP［XSTAMPP2018］、データベース連携やガイドワード編集可能な SafetyHAT［SafetyHAT2018］、トーマス博士が提案した拡張 STPA［Thomas2013］をサポートする an STPA tool［Thomas2015］、Enterprise Architect の拡張である SHARA［Krauss2015］などが紹介されている。

4.2.2.1. XSTAMPP

eXtensible STAMP Platform（XSTAMPP）は、A-STPA のスタンドアロン版の後継機以外にも、多くのプラグインを含み、基本的な STPA 以外にも多くの STMP ベース分析や開発連携を可能にしている。具体的には、XSTAMPP は、A-STPA 以外に、A-CAST（CAST 用）、XSTPA（トーマス博士の拡張 STPA 用）、STPASec（STPA for Security 用）、STPAPriv（STPA for Privacy 用）、STPA Verifier（モデル検査と STPA の連携）、STPA Safety-based Test Cases Generator といっ

たプラグインを含む。

XSTAMPP では、STAMP/STPA の構成要素（コンポーネント、CA など）が用意されており、各ステップ出力中に記述されるこれらの構成要素は関連付けられている。具体的には、Step 0 準備 2 において CSD を構築する際には、CSD の構成要素（コントローラー、アクチュエータなど）を選択肢から選び、CSD 中に追加できる。また、コンポーネント間の接続や接続に付随する CA やフィードバックも選択肢から選び、CSD 中に追加できる。追加された CA は、Step 1 の UCA 一覧表中に記載される CA と連動する。しかし、CSD のコンポーネント中にサブコンポーネントを記述するといった階層的記述は対応していない。

4.2.2.2. SafetyHAT

A Transportation System Safety Hazard Analysis Tool（SafetyHAT）［SafetyHAT2018］ は、その名が示す通り交通機関の安全解析に重点を置いたツールであり、交通機関に特化した 4 つの Step 1 ガイドワードと Step 2 の HCF のヒントを提供している。更に、コンポーネントタイプやガイドワードをユーザがカスタマイズ可能になっている。また、コントロールストラクチャーの記述は図形式ではなく、表形式を採用している。ただし、別途ユーザが作成した図形式のコントロールストラクチャーとのリンクは可能である。他方、データベースとの連携やエクセル形式で分析結果を出力することも可能である。

4.2.3. STAMP Workbench

本項では、IPA/SEC が開発している STAMP/STPA 支援ツール STAMP Workbench について紹介する。なお、STAMP Workbench は本章執筆時点では評価版であるため、公開版との差がある可能性があることには留意いただきたい。

STAMP Workbench の目的は、基本的な清書機能、分析手順ガイド機能に加え、分析者が分析に注力できるよう支援する機能を提供することである。例えば STAMP Workbench は、反復しながら分析を進める際に発生する手間（図表変更とその変更が引き起こす修正作業など）から分析者を解放することを目指している。STAMP Workbench はモデルベース開発で使用されるツールが持つ機能に加え、以下の 5 つの機能（図 4.2-2）を持つよう設計されている。

1）STAMP 図生成機能（STAMP で使用する図形及び STPA 基本手順支援機能で使用する図形を生成する機能）
2）STAMP 表生成機能（STAMP で一般的に使用する表及び STPA 基本手順支援機能で使用する表を生成する機能）
3）汎用形式データ出力機能（STAMP Workbench が保持する分析データを、汎用的な形式で出力する機能）
4）外部ツール連携 I/F（STAMP Workbench と他ツールの間で片方向あるいは双方向に入出力データを受け渡すためのインターフェース）
5）STPA 基本手順支援機能。

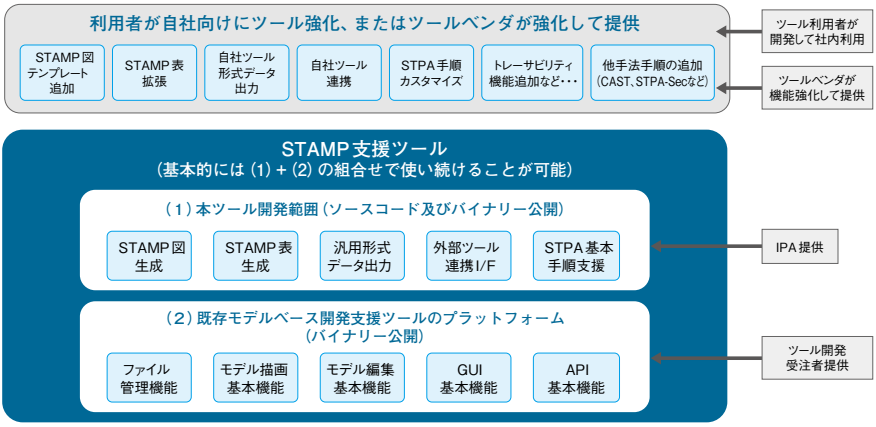


図 4.2-2　STAMP Workbench の構造

STAMP Workbench の特徴的機能を解説する。STAMP Workbench は、STPA の各 Step での標準的出力をサポートしている。実際 4.2.1 節において STPA の各 Step の出力を紹介したが、アクシデント、ハザード、安全制約の一覧表（表 4.2-1）、コントロールストラクチャー図（図 4.2-1）、UCA 一覧表（表 4.2-2）、ハザード要因の一覧表（表 4.2-3）は、STAMP Workbench を用いて作成している。これらの基本的機能に加え、表形式で整理したデータから CSD を生成する機能、Step 2 の HCF ヒントを選択・編集する機能、UCA や HCF などへ自動採番する機能を持つ。

CSD 生成機能について解説する。Step 0 準備 2 では、コントロールストラクチャーを要求仕様などの利用可能な資料から構築する、しかし一般的なモデル構築と同様に、コントロールストラクチャー構築は一般に困難であり、試行錯誤が要求されることが多い。そこで STAMP Workbench は、SafetyHAT と同様に、表により情報を整理し、整理された情報から CSD を生成する機能を持つ（図 4.2-4）。また、直接 CSD を記述できるようにも設計されている。

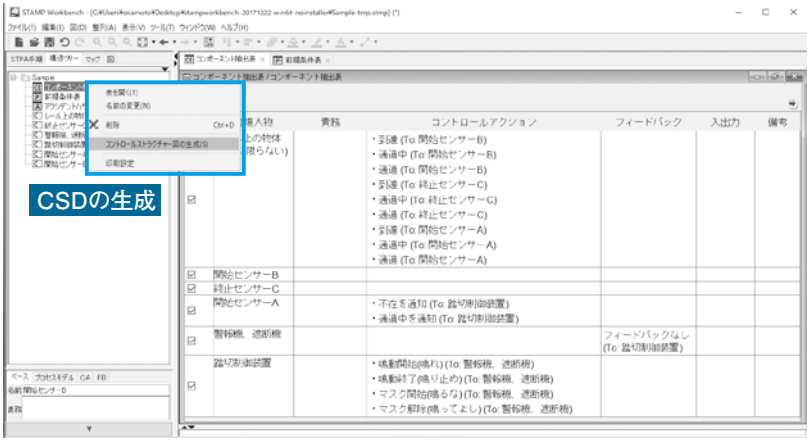


図 4.2-3　コントロールストラクチャー図生成機能

Step 2 の HCF ヒントの選択・編集機能について解説する。

Step 2 では、ハザード要因の一覧表とハザードシナリオを作成する。STAMP Workbench では、HCF のヒントはユーザが既存のヒントワードセットから選択可能であり、更に各ヒントワードセットはユーザが編集可能である（図 4.2-4）。この機能により、分析対象に適した HCF のヒントを与えられ、HCF を特定しやすくなる。

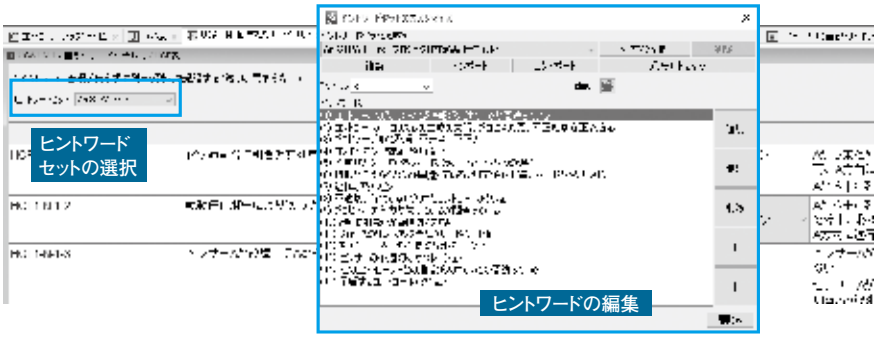


図 4.2 4　HCF ヒントの選択・編集機能

自動採番機能について解説する。STPA 各 Step の表中の分析結果（UCA、HCF など）に番号を付けることで、参照が容易になる。しかし、分析を行う中で過去の分析結果に追加・修正することもあり、それに伴い番号の変更を余儀なくされることがある。STAMP Workbench は、分析結果中のデータに自動的に番号付け（番号変更）を行う機能を持つ。

4.2.4. まとめ

本節では、STAMP/STPA 支援ツールを紹介した。4.2.1 項では、STAMP/STPA の手順、各 Step での出力形式や課題について解説し、STAMP/STPA 支援ツールに求められる機能を挙げた。4.2.2 項では、STAMP/STPA 支援ツールの紹介として、XSTAMPP と SafetyHAT を紹介した。4.2.3 項では、IPA/SEC で開発中の STAMP/STPA 支援ツール STAMP Workbench について、その目標と執筆時点での機能を紹介した。

IPA/SEC で開発中のツール STAMP Workbench は、IPA/SEC のワーキンググループでの事例検討の実績をもとに開発されている。例えば、データの修正による番号振り替えなどの作業を自動化して思考を妨げないこと、内部データの論理構造の一貫性を保ち、トレーサビリティを容易に確保できること、データのエクスポート機能を持たせユーザニーズに応じたカスタマイズが容易にできること、といった特徴を持たせている。今後、STAMP Workbench の現場での活用が期待される。

5. 複雑システムの安全性向上技術と Safety2.0

5.1. 複雑システムの課題と方策

ネットワーク技術の進歩により、それまで個別に構築されてきた個々のシステムが統合され、総合的なシステムとして発達してきた。複雑システムにおいては、全てのシステムを同時に構築することは困難なことが多いため、徐々にシステムの機能や、規模、対象エリアを拡張する方策がとられる。その際に、既存のシステムの動作を乱さない等の要件をシステム論として満足させる研究もアシュアランスシステムとしてなされている［松本 2003］。アシュアランスシステムは、システムが逐次サブシステムと結合され大規模化していっても、システムの動作環境が変動しても、一部に故障を引き起こすフォールトが存在しても、あるいは意図的なセキュリティ攻撃がなされても、障害を回避もしくは障害の範囲を限定し、期待されるサービスをタイムリーに保証するシステムである。

アシュアランスシステムとしては、これらの要件を満足させるための方法論が重要になるが、民生分野でのアシュアランスシステムの代表例である東京圏輸送管理システム ATOS（Autonomous decentralized Transport Operation control System）［伊藤 2011］では、自律分散の概念に依拠したシステム構成法が採用されている。

このように、複雑システムの安全向上技術としては、定まったシステムアーキテクチャの下でおこなわれる STAMP/STPA や FRAM といったシステム評価の方法論に加え、デザインフェーズにおけるアーキテクチャの選択やシステム構成におけるデザインコンセプトの確立も重要な意義を持つ。アシュアランス技術についてはすでに多くの研究が行われている［松本 2003］ので、その成果に委ねることとし、本章では、複雑化そのものを回避するシステムデザインの重要性と方法論について考察したい。

5.2. 本質制御と Safety2.0

5.2.1. 高機能化が複雑化を促進

複雑化に至った経緯は、システムの面的拡大によるものの他に機能性の向上を意図した技術開発の産物という側面もある。列車制御に関していえば、列車の進路を構成する転てつ機と安全な走行を伝達する信号機の制御は、当初は人間の梃子扱い（人力）でおこなわれていた。しかし、多くの人間の命を奪う重大事故が人間のミスや機械の故障により引き起こされたため、その防護策として、多くの列車制御システムが開発され、技術的にも事故の教訓を組み込み成熟していった。

今日では、単に安全性のみならず、乗心地への配慮や車両性能をそのまま発揮できる理想的システムがデジタル ATC として登場し、新幹線や都市圏の線区に導入されている。デジタル ATC は、レールを介して、現在走行している閉そくの番号と先行列車の在線位置（在線閉そくの番号）をデジタル電文として車上装置に伝達するシステムである。車上装置は、ATC デジタル電文を受信すると、現在の走行位置から先行列車が在線する閉そく境界までの距離を算出するとともに、その間の線路データを組み込んだ速度照査パターンを生成させて、速度を自動的にコントロールする（図 5.2-1 参照）。デジタル ATC の地上装置の構成を図 5.2-2 に示す。この地上装置で行っていることは、列車の在線位置を検知することと、各列車の在線位置から個々の閉そく区間に送信する電文の生成、そして、その電文を増幅しレールに送信することである。各電文の生成処理は信頼性 / 安全性に配慮された一台の処理装置で行っている。

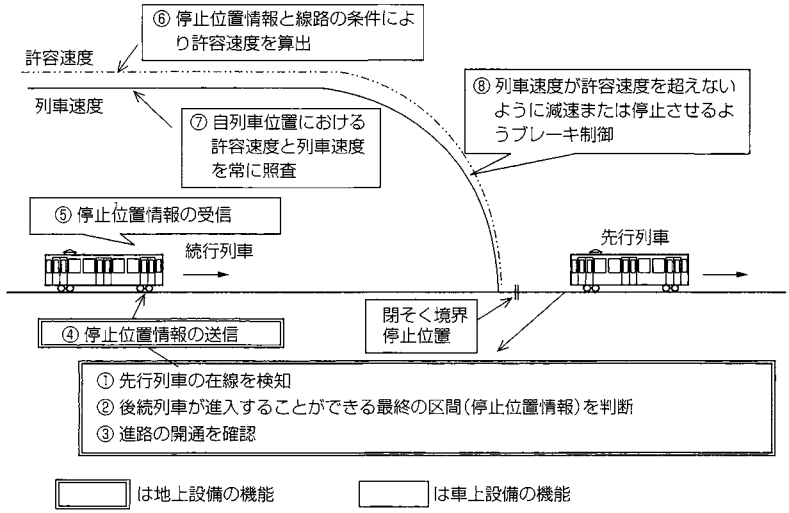
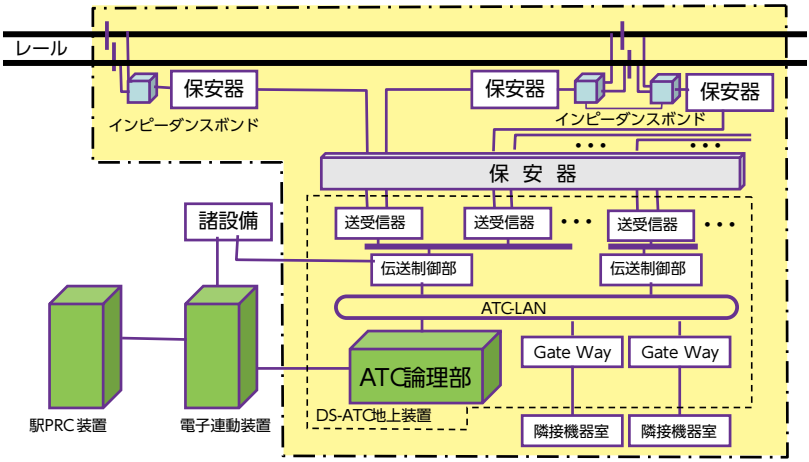


図 5.2-1　デジタル ATC のシステム概念［鉄道 2015］



デジタル伝送技術を応用した新幹線ATCシステム（DS-ATC）の開発　平成13年度信号セミナー予稿　参考

図 5.2-2　高度に発達した列車制御システム（デジタル ATC）の地上装置

5.2.2. 複雑化を回避して高機能化と安全を両立させる方策

一方、デジタル ATC の開発に並行して開発研究が行われていた、無線式列車制御システム CARAT（Computer And Radio Aided Train control system）［中村 1993］の成果は、JR 東日本に引き継がれ、ATACS（Advanced Train Administration and Communications System）［馬場 2012］として開花した。ATACS は、2011 年（平成 23 年）10 月に仙石線で実用化され、稼

働率 99.99999 という優れた成績を収めており、2017 年（平成 29 年）11 月には埼京線の池袋・大宮間にも導入された。CARAT の開発研究の意図は、複雑化する一方の列車制御システムの在り方に疑問を呈し、当時の情報技術、通信技術に依拠した列車制御をゼロベースで再構築することであった［中村 1993］。軌道回路や閉そく装置といった現場機器を用いずに、先端の機能を実現した ATACS の成功により、無線を含めた通信技術を用いて「関連する機器同士が相互に情報を交換して機能を実現するシステム形態」の利点がクローズアップされた。図 5.2-3 は、既存デジタル ATC と CARAT の安全性を FTA により比較したものであるが、現場装置を削減し、情報交換で安全を確保する CARAT の優位性が分かる。

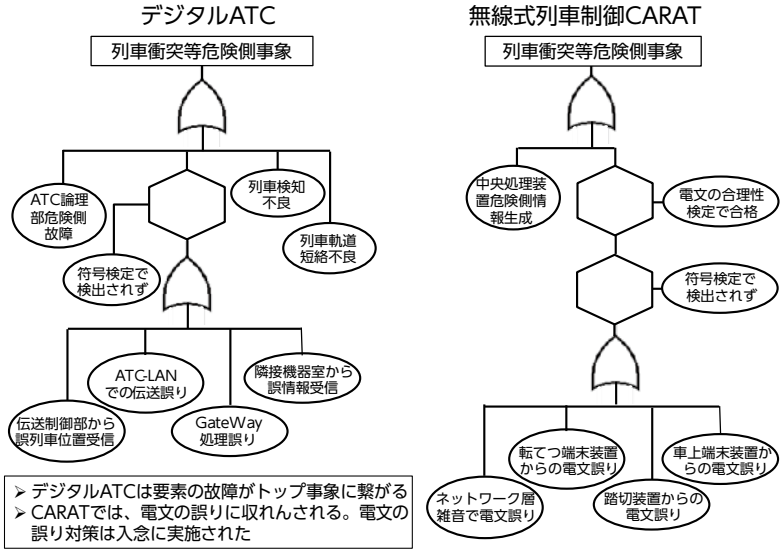


図 5.2-3　既存デジタル ATC と無線式列車制御システム CARAT の FTA 解析結果

5.2.3. 本質制御と IoT

システムを構成する上で必須とされる要素同士が、相互に情報交換を行い、必要とされる機能を実現する形態を「本質制御」と名付ける［中村 2016］。本質制御は、既存の制御装置の省略も可能とする経済性に富んだシステム設計概念であるが、同時にシステムの信頼性 / 安全性 / 保全性を向上させる方策でもある。

ATACS における地上側の設備構成を図 5.2-4 に示す。拠点装置は 2 又は 3 台の無線基地局を介してエリア内の列車と情報の交信を行う。列車が拠点装置の制御ゾーンを抜けるときには、それまでの拠点装置と次の拠点装置との間で情報交換を行い、列車の追跡と無線交信を要請するハンドオーバ処理を行う。この形態は、既存 ATC 処理装置が制御エリア内の軌道回路に ATC 信号を提供するものと変わりない。実は、それまでのアナログ ATC の制御概念を基に CARAT の仕様が定められたため、既存信号システムのアーキテクチャが残っているとも言える。

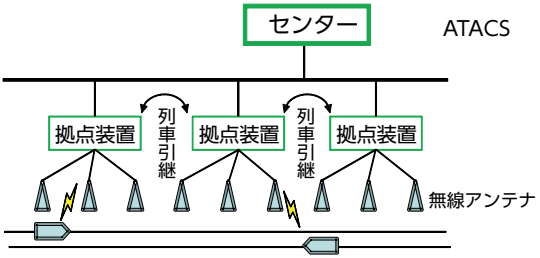


図 5.2-4　ATACS の地上設備構成

しかし、今日の無線技術では、ローミング技術によりデータは任意の箇所に伝達できるため、拠点装置を必ずしも沿線に配置する必要はない。究極の姿は、センターに配置した処理装置によって、全ての列車が追跡され、その情報を基に各列車に対し、走行可能な地点の情報が車上装置に伝達される形態である。

その形態の下では、列車の運行管理等を行うセンター装置、そして、車上装置、転てつ機、踏切装置が相互に情報交換を行う。その結果、これまで列車の安全な運転を支えていた連動装置や閉そく装置、ATS/ATC といった保安装置を現場に配置することなく、安全で高機能な列車制御が行える。まさに、本質制御の概念を列車制御システム上で展開したものであるが、本質制御は、複雑化するシステムの安全性向上策としても検討に値する。その応用事例として地方交通線向けに開発中の ATP 閉そくシステムを図 5.2-5 に示す。

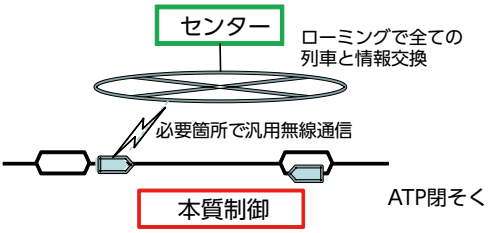


図 5.2-5　本質制御の概念に沿う ATP 閉そく（地方交通線向けに開発中）

今日、多くの産業分野や技術開発のフィールドで IoT なる用語が展開されている。あらゆるものがインターネットでつながるという形態である。しかし、つながった先に見えるものが、外出先からのクーラー等の家電装置の ON/OFF 制御であったり、これまで情報が取れなかったデバイスの情報を収集したりするだけに留まっていては、IoT を活かしたとは言えない。コンセプトが欲しい。本質制御は、IoT の環境を生かした新たなデザインコンセプトの提案でもある。

5.2.4. 本質制御と Safety2.0

一方、産業界の生産現場では、事故防止を目標とした新たな活動が、Safety2.0 のもとに展開されつつある。Safety2.0 は、IoT の利点を生かし、人とモノと環境が相互に情報交換を行い、

協調して安全を築こうとするもので、協調安全と呼ばれる日本発の新しい取り組みである［中村 2017］。

これまでの生産現場における安全は、リスク解析から始まり、リスクを許容可能な水準まで低減する方策の徹底を前提に行われていた。しかも、その前提は、危険領域からの人間の排除を主体に安全を確保するものであった。この方策は、フェールセーフに構築されるが生産性との兼ね合いから必ずしも生産現場では歓迎されず、皮肉を込めて「止める安全」と称されることもあった。Safety2.0 は、単に安全上の効果にとどまらず、これまでの「止める安全」から、「稼動を継続しつつ安全を確保する」形態に変化できるため、生産性向上にも寄与するなど、経営上の課題解決にも貢献し得る取り組みとして期待されている。

図 5.2-6 は、日経 BP 社のパンフレット［日経 BP2015］に掲載された Safety0.0 から Safety2.0 までの相違を説明した図である。

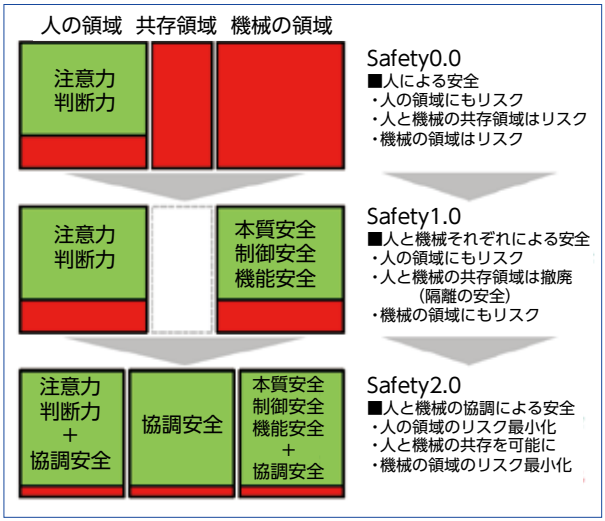


図 5.2-6　Safety0.0/1.0/2.0 の概念の比較

図 5.2-6 が示すように、Safety0.0 は、「安全第一」「指差喚呼（指さし確認）」など人間の注意力の覚醒に期待して、安全を確保しようとするものであった。一方、Safety1.0 においては、機械の稼働領域という危険源から人間を隔離することにより安全を確保しようとするもので、Safety0.0 に比して安全の領域が拡大した。しかし、隔離に要した共存領域に人がいれば生産は停止する。ただ、保守という非定常作業があり、完全に隔離することはできず、共存領域での災害が頻発していた。それに対し、Safety2.0 は共存領域での作業を前提に安全策を施すため、生産性と安全性両方の向上が期待できるのである。すなわち、要素同士が双方向で情報を交換しながら最適な安全を確保するという閉ループ制御の本質を IoT に依拠することで実現する。Safety2.0 の方向性でサブシステムを作れば、複雑システムの安全向上策に使える。

Safety2.0 の概念を社会に広め、海外にも展開しようとする活動が、一般社団法人セーフティグローバル推進機構（IGSAP : The Institute of Global Safety Promotion）のもとで開始されている［向殿 2017］。IGSAP は、2016 年（平成 28 年）7 月に設立されたもので、「ISO/IEC Guide 51 に規定する製品、プロセス、サービス、システムの安全性を対象としたリスクアセスメントをベースとする安全要員認証（以下安全要員認証）スキームを構築するとともに、

その運営、促進を通じて、安全技術の振興、産業安全の確保及び生産性向上を図り、もって我が国経済及び世界経済の発展に寄与することを目的」としている。

IGSAP の目的の実現には、企業を挙げての取り組みが必要となるが、「安全の確保がコストの問題ではなく企業に福音をもたらす投資である」という Safety2.0 の意義を、とりわけ経営層に正しく認識してもらうことを重視している。さらに、その具体化には、RISK 解析をベースに安全の勘所を押さえつつ Safety2.0 のアーキテクチャ構築ができる人材の育成も必要になる。一方、IoT を利用した Safety2.0 に依拠した新しいシステムが生み出されたなら、その概念や方法論の水平展開を図り広めるための工夫も必要になる。IGSAP は、その活動内容として、

①安全要員認証に関する規格作成及び要員認証制度の企画、提案活動
②安全要員認証に関する制度の開発、実施、運営及び普及活動
③安全要員認証に関する研究開発及び普及活動
④安全要員認証の利用企業振興のための諸活動
⑤安全要員認証制度の国際普及活動
⑥安全に関する講習等の教育活動
⑦安全に関する研究会、情報交換会等の啓発活動
⑧安全に関する表彰活動
⑨その他、本法人の目的を達成するために必要な事業及びこれに関連する事業

といった活動を掲げている。また、Safety2.0 を基本コンセプトとした活動も IGSAP を核として展開されつつある。

同時に製品認証についていえば、2017 年度（平成 29 年度）には，その具体化として，製品単独ではなく，人，モノ，環境などがつながった（Safety 2.0 の技術的定義に適合した）システムを対象に適合マークを付与し，Safety 2.0 の普及促進を図る活動も開始された。

企業トップを含めた関係者がシステムへの理解を深め、安全に対する意識をもって取り組む環境づくりは、設計段階、構築段階、運用段階、保全段階を含めた安全性向上に向けての良い環境を作り出すものとして期待される。

5.3. まとめ

ソフトウェアの安全性評価として故障に起因した障害の解析手法である FTA や FMEA の限界が認識されている。とくに本章で取り上げた複雑システムに対する FTA や FMEA の適用には、解析のための解析になりかねず実効性に対する懸念が払拭できない。この点で、STAMP/STPA や FRAM は、構成要素間のインターフェースに着目している点で、複雑システムに対しても合理的な評価が期待出来る。

とはいえ、アーキテクチャデザインの段階でできる限りシステムをシンプルにする努力は、STAMP/STPA や FRAM による解析の容易化や確実化にも有利に作用することはいうまでもない。Safety2.0 の究極の姿である本質制御のアーキテクチャ設計は、システムのシンプル化に寄与するため、安全解析を容易にする点でも有効な方法論でもある。

6. FRAM（機能共鳴分析手法）による IoT 特有の安全解析

6.1. はじめに

人工知能を搭載した不特定多数のエージェントが巨大な IoT［ITU2012］システムを構成しつつ自由に行動する現実・仮想の空間を如何に安全なものにしうるか、という課題は、特に一般道路上での自動運転等の分野で早急に解決されなければならない問題となってきている。一方、ソフトウェアにより高度に自動化されたシステムの安全化については、従来以下の戦略がもっぱら採用され、成果を上げてきた：

・想定されるハザードに対して、ハザードを発生させる原因を識別する
・ハザード発生原因毎に発生を防止するための安全制約を適用する
・ハザード発生原因毎に発生を検知し安全化するための制御を作りこむ

これらの方法は、制約、すなわちバリアを設けること、あるいは危険状態を検知して積極的に制御をかけることが有効であるという前提条件の下で成り立つ。

しかし、多数の自然知能（人）が自由に行動する現実空間は、必ずしも上記の方法で安全化されてきたわけではない。たとえば、本論で分析する東京駅のコンコースにおいては、人の行き来を制約するルールや危険を検知して人の流れを制御するような機構は設けられておらず、一見するところ、単なる通路でしかない。つまり、東京駅の安全化は、既存の安全制御のスキームとは異なるものによって達成されている可能性が高い。

また、近年注目されつつある Safety2.0［中村 2017-2］等の新しい安全理論の出現は、既存の安全化手法を超える必要性を示唆するものと考えることができる。

人工知能の問題を考察するにあたり、既存の安全制御の方法に縛られることなく、自然界に既に存在している例から将来の安全化策のヒントを得ることは、有効であると考えられる。

本論では、東京駅のコンコースがどのように安全に保たれているのかを分析し、次世代の人工知能安全につながる指針を得ることを目的に、機能共鳴分析手法（Functional Resonance Analysis Method: FRAM）［ホルナゲル 2013］を用いて、システムの成功要因・リスク要因を識別する。

6.2. 東京駅のコンコース

東京駅には、一日平均約 44 万人の人間が流出入する［JR 東日本 2018］。仮にメインコンコースに、全流入人口の半分の 20 万人が行き来すると考えると、朝 6 時から夜 12 時までの 18 時間、平均で毎時 1 万人強の流出入が繰り返されている、巨大かつ、極めて流動的なシステムと捉えることができる。

このコンコースを観察すると、以下の特徴がある：

（1）固定のプレイヤーが固定の位置にいることが無い。すなわち、最も重要なプレイヤーである「人」は固定構造の中の固定コンポーネントとなっていない。

（2）コンコース全体の安全を管理している管理者、所謂 Safety Controller がおらず、かつ、安全を確保するための信号や交通整理の指示と言った安全制御アクションが存在していない。

（3）大規模公共施設のハザードとして、何を考えるべきか確定することが難しい。「衝突」のような即物的なハザードを立ててしまうと、ミクロな衝突防止のメカニズムに解析が限定される恐れがあり、自宅の廊下での衝突と本質的に変わらない解析になる可能性がある。ここでは大規模 IoT システム特有の問題点を抽出したい。

6.3. FRAM 分析

本論では、IoT システム固有の問題点を抽出するため、FRAM（Functional Resonance Analysis Method: 機能共鳴分析手法）を適用する。FRAM 分析は、上記のような、システムの全体構造が良くわかっていないカオス的なものの隠れた構造を可視化するのに役立つツールである。FRAM は、システムが持つ「機能」を取り出し、機能と機能がどのように結びついているかによってシステムの様相を明らかにする。このやり方が有効である理由は、「プレイヤー」は入れ替わっても、そこに存在する「機能」は変わらないという特性による［Deleuze1984］。言い換えると、コンコースを歩く人は常に変わっているが、皆が行おうとしている「目的地に向かって歩く」という機能は不動である。

FRAM を使ったモデリングは、取り付く島の無いカオス的な状況において、シンプルなアプローチを可能にしている。以下にそのアプローチを示す。

（1）まず、分析対象のシステムのうち、最も代表的、あるいはもっとも重要と思える機能を一つピックアップする。
（2）次にその機能が出力するものは何かを定義する。
（3）次に、その機能の実行条件を決める様々なパラメータ（動作トリガー、前提条件、資源、時間制約、制御パラメータ）が外部の機能から入力されているかを分析する。
（4）次に、その機能への入力を提供している他の機能についても、上記の（2）と（3）を行い、機能の相関関係を明らかにする。
（5）以上のように数珠つなぎに機能を追加してゆき、最後に、識別した機能が特に大きな変動要素もなく安定的に出力を繰り返しているものになるまで行う。このような機能をバックグラウンド機能と呼ぶ。バックグラウンド機能は、FRAM 分析の分析範囲の周縁である。

6.4. 東京駅コンコースの FRAM 分析

東京駅コンコースの安全を分析するうえで一番注目したい機能を一つピックアップする。ここでは、コンコースの最大の目的である、「歩行者の移動」、すなわち、「歩く」という機能を取り上げてみる。

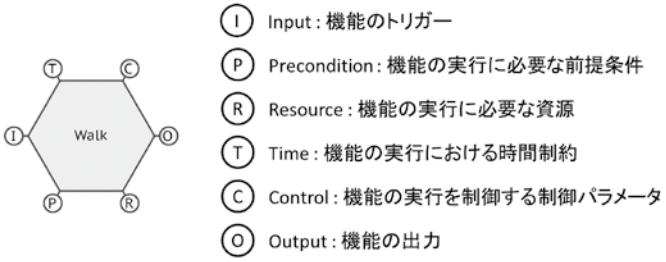


図 6.4-1　最初に注目する機能 "Walk"

まず、「東京駅コンコースを歩く」という機能を分析するときに 5 つの入力要素を考える：

（1）東京駅コンコースを歩くという動作のきっかけとなるトリガーは何か？
（2）東京駅コンコースを歩くという動作を行うための前提条件は何か？
（3）東京駅コンコースを歩くという動作を行うために必要となる資源は何か？
（4）東京駅コンコースを歩くという動作を行うための時間制約は何か？

（5）東京駅コンコースを歩くという動作を行うために使われる制御パラメータは何か？

次に、「東京駅コンコースを歩く」という機能の出力は何かを考える

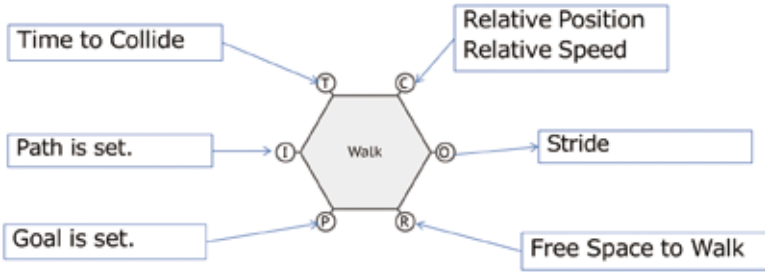


図 6.4-2　Walk 機能への入出力

歩くという機能の出力は当然、出される「一歩」である。

次に入力を考える。ここで注目するべき事実は、「歩く」や、「呼吸する」などのような、無意識で行われている機能においては、あらゆる要素が総動員されて動いていることが多いということである。長年の鍛錬や進化によって、徐々に獲得されてきた生命維持のための機能に特徴的な性質である。上図のように、それらの要素を識別したならば、次に、それらの要素を出力する機能は何かを考える。

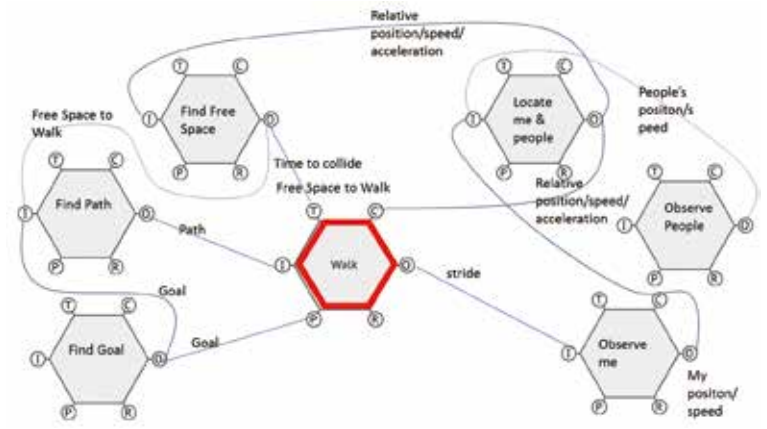


図 6.4-3　完成した FRAM モデル

さらに、上記のプロセスを繰り返し、「歩く」機能のネットワークがバックグラウンド機能に到達し、分析対象範囲をカバーし終えた時、FRAM モデルが完成する。

6.5. 作成した FRAM モデルの分析

作成した FRAM モデルは、FRAM Model Visualizer の機能を使って縦横無尽に構成を変えな

がらその特徴をとらえてゆくことができる。一見スパゲッティ状の複雑なネットワークであっても、機能間ネットワークである以上、純粋にランダムなものではなく、人工的なネットワークであれば、設計者の設計思想を反映した「設計アーキテクチャ」、自然物のネットワークであれば、長年の進化の過程で練り上げられてきた「創発的アーキテクチャ」を持っているはずである。そうした構造を探り出す作業は、FRAM 分析の中でももっとも建設的なものであり、過去に同様のシステムを設計、または分析した経験を持つアナリストにとっては、経験を生かした分析能力の発揮のしどころであり、そのような経験を持たないフレッシュなアナリストにとっては、自ら偉大なシステムアーキテクチャを創造、発見する喜びに満ちた作業である。アーキテクチャが見えにくい複雑なネットワークは、接続ラインの長さ、交差数等、ネットワーク構造を複雑にするパラメータを最小にする機能ブロックの配置を最適解とみなし、ツールを使ってそれを見つけ出してゆく。

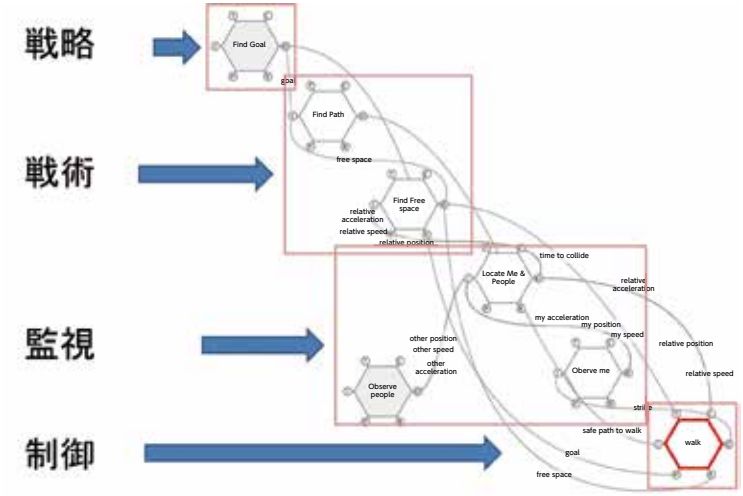


図 6.5-1　4 つのレイヤー構造

ここでは、コンコースにおける歩行の機能ネットワークを、上図のような構造と捉えた。ここから作成したモデルの評価を行ってゆくのであるが、FRAM を使った安全評価は、FRAM の工学的なバックボーンとなっているレジリエンス・エンジニアリングを抜きにしては語ることができない。レジリエンス・エンジニアリングとは、従来の安全工学と異なり、システムのリスクやハザードが失敗要因からではなく、成功要因からも発生すると考える［ホルナゲル 2013］。したがって、FRAM の安全評価作業においては、対象システムの成功要因、あるいは、優れた特質を分析することから始まる。東京駅コンコースのように、長年の運用を経て少しずつ練り上げられた人工物であれば、明治初期に設計された設計思想そのものではなく、運用実績をフィードバックして少しずつ改良を加えられた結果として現在の形となっており、かつ、時代の移り変わりや利用者の嗜好等を反映して、常に変化し続けているはずである。また、その結果作り上げられたコンコースのシステムは、現在、極めて安全で効率的なものになっており、成功要因の分析対象として優れた例題となっている。

上図に示したように、4 つのレイヤーに分割されているととらえると、各レイヤーは、以下のような内容を持っている：

（1）戦略層：歩くという行動の最終目的を提示する
（2）戦術層：どこへ向かって、どのようなコースで歩くべきかを提示する
（3）監視層：戦術を達成するために自分の動きと群衆の動きの相対関係を監視
（4）制御層：歩くという物理的な行動を行う

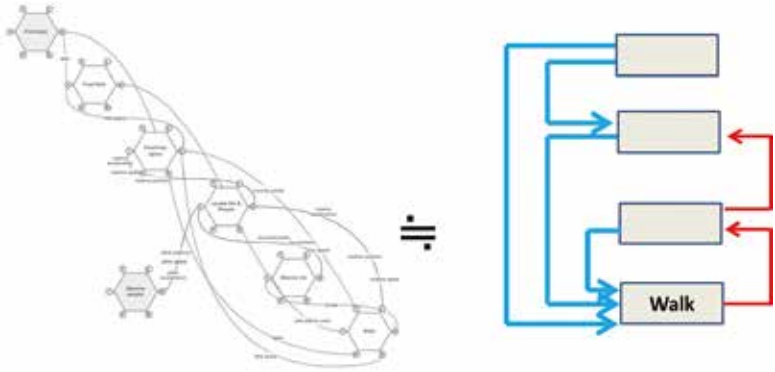


図 6.5-2　FRAM モデルを STAMP モデルに簡略化したもの

次に各層の間の関係を分析する。各機能からの出力は、一つ上の上位層に伝達されているものとそうでないものが混在している。特に特徴的なのが、最下層の制御層に多くの入力が集中しているという点である。この分析により、東京駅における歩行は、STAMP の解析に頻繁にみられる階層的フィードバックループ構造［Leveson2013］とは異なることがわかる。そこで、どれほど階層的フィードバックループ構造と異なっているのかを下図に示した。

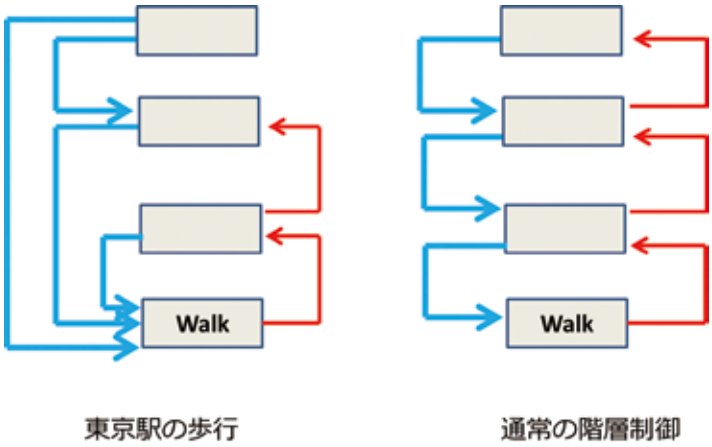


図 6.5-3　東京駅モデルと STAMP 的階層構造の違い

上図の左側が FRAM モデルの構造。右側が STAMP 的階層的フィードバック構造である。STAMP 的なフィードバックの階層構造はトップダウン型の制御ということができる。上位層

は常に下位層をコントロールするためにControl Actionを出している。下位層から上位層へは、そのControl Actionからのフィードバックが行われ、フィードバック制御ループ構造を作っている。

一方左側の構造は、トップダウン構造ではない。レイヤー構造らしく上位層へデータがリレーされてゆく構造を持ってはいるが、上位層各層からは、出力が最下層の制御層に集中している。すなわち、下から吸い上げてきた情報が、随時さまざまな形で制御に使われているという構造であり、制御の主体は最下層にあり、上位層はサーバーとしてサービスを提供しているだけである。これは、むしろボトムアップ構造ということができる。

また、興味深いのは、最上位層からは一方的に情報が出されるだけであるということである。ここだけオープンループになっており、従来型の制御構造とはますます異なっている。

これらの特徴、及び、長い歴史によって証明されてきた実績から、東京駅が何故安全に保たれているのかも判明する。つまり、東京駅は、トップダウンな交通整理や安全制御を誰も行っていないからこそ安全に保たれていると考えざるを得ない。さらに、目的地を与える最上位層は、とにかく安定的に目的地を提示しているだけで、人々の動きのフィードバックを受け付けていないからこそ安全が保たれていると考えられる。

6.6. 成功要因の分析

前述の2つの制御構造図の比較から、東京駅の以下の成功要因が見えてくる：

（1）一旦ゴールが設定されると、ボトムアップに練り上げられる戦術にしたがって皆が歩き続けることができる。
（2）ゴールは常に安定的に提供されている。
（3）戦術は、常に自動的に参加者全員によって環境変化に適応して更新し続けられる。戦術を決めるのは、ルールや交通整理によるトップダウンプロセスではなく、参加者によるボトムアッププロセスである。

つまり、東京駅が安全である理由は、歩行者全員の行動によって創発的に戦術が練り上げられるからである。言い換えると、カオスからの自己組織化がこのシステムの安全性を作り出している。ボトムアップに練り上げられる戦術というものは、極めて柔軟であり、環境変動に即座に対応する。たとえば、団体客がコンコースの真ん中に滞留しているときには、歩行者全員が無意識のうちに戦術を変更してコースを変えてゆく。誰も交通整理をしていないからこそ、このような安全化が可能となる。同様な研究事例として、クレイグ・レイノルズの「群れのルール」の研究がある。［Craig1987］

レイノルズはシミュレーションによって、鳥の群れは、以下の3つの要因だけで統一のとれた団体行動をとることができることを示した：

・分離（Separation）
　鳥オブジェクトが隣の鳥オブジェクトとぶつからないように距離をとる。
・整列（Alignment）
　鳥オブジェクトが隣の鳥オブジェクトと速度と方向を合わせる。
・結合（Cohesion）
　鳥オブジェクトが群れの中心方向へ向かうように方向を変える。

それぞれの鳥は、各々、隣の鳥との距離・速度を保ち、群れの中心方向を指向する。これらは、特に上位コントローラーが不在でも可能な制御である。つまり、東京駅と同様にボトムアッププロセスによって創発的に実現されている。東京駅は群れが同じ方向へ向かう行動とは異

なり、それぞれのオブジェクトは、それぞれの目的を持っているという点が異なる。目的が異なり、統一的な行動をとる必要は無くとも、それぞれがボトムアップに戦術（歩きかたの決定）を練り上げてゆくという点は同様である。

6.7. リスク要因の分析

前項で見てきた成功要因は同時にリスク要因にもなりうる。ボトムアップに戦術が決定されるということは、上位からの指示無しで行動し続けられる反面、一旦ゴールが提供されなくなると、各歩行者は当所もなく俳諧し始めることを意味する。もし東京駅でこのような状態が発生すると、新規流入者によってたちまちコンコースが飽和状態にまで混雑し、衝突事故が多発するなど、パニック状態に陥ることになる。これこそが東京駅のハザードである。ゴールが提供されるか否かは個々人からのフィードバックによらずに一方的に決まることであるため、この機能は、ひとえに駅の設備によって実現されている。コンコースにおけるゴールを提供する機能とは、すなわち行き先案内版の表示機能である。東京駅を観察すると明らかにわかるのは、コンコースのどこにいても、必ず案内板が見えるように、夥しい数が設置されていることである。「あらゆる場所から案内板が見えるようにする」ことこそが、東京駅のハザード制御となっている。このハザード制御は、長い東京駅の歴史の中で徐々に改良を受けて完成されてきた仕組みであり、歩行者が自己組織化によって安全になっているのと同様、創発的に練り上げられてきた工夫である。

東京駅のコンコースは、行き先案内板以外には、特に特徴的な安全フィーチャーを持っていない、単なる通路である。しかし、それゆえに、通行人の自由度が高く、最小面積に最大乗客数が出入りできるため、容易にパニックには至らず、結果極めて安全なシステムになっている。

6.8. まとめ

東京駅の安全解析を、FRAM によって行った。FRAM 分析によって明らかになった制御の階層構造の特殊性から成功要因やリスク要因を導き、解析開始前にははっきりとわからなかったハザードやハザード制御機能を識別することができた。この方法は、IoT システムや、プレイヤーがインテリジェントに振る舞う人工知能システムの安全解析手法として有効であると考えられ、Safety 2.0 を実現するためのひとつの方法論として大いに寄与することが期待される。

おわりに

2015~2017 年にわたって IPA/SEC で活動をしてきた WG では、新しい安全解析手法 STAMP/STPA 普及の先導役として、「紹介」、「入門」、「実践」と階段を登ってきた。ここで、もう一段の進化、すなわち、「理解する」から「やってみる」そして「当たり前にやる」ために、「はじめての STAMP/STPA（活用編）」を小冊子としてまとめた。さらに、STAMP/STPA の先にある複雑システムの新しい安全解析法についての展望もまとめた。

本書では、鉄道や車などの産業界での試行事例、人と機械の協調による安全制御の事例、セーフティとセキュリティの統合分析事例などをまとめている。また、これらの事例分析を支援するために IPA/SEC で本年度に開発した STAMP 支援ツールの紹介もした。さらには、STAMP/STPA を越えて、将来の複雑システムの安全解析の在り方に関するビジョンの提言も行っている。

STAMP/STPA の基本理念は、複雑システムではコンポーネントの故障ではなく、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主張である。人とソフトウェアが協調して安全を守るシステムでは、人の予想できない行動やソフトウェアの要求仕様そのものの欠陥がコミュニケーション・ミスマッチの原因となって事故を引き起こす。このために、人の操作手順書を整備しても、ソフトウェアの信頼性を完璧にしても、安全性を向上させることはできない。人やソフトウェアのエラーは無限のバリエーションを持ち、創発的でもある。これを如何に有限化し、排除するかの回答の一つが STPA　Step 1 の非安全制御行動の 4 つのガイドワードを用いた二分法である。STAMP の事故モデルは、事故を防止するための安全制御機能をシステムモデルの中で明示化するところに最大の特徴がある。そして、その機能が適切に働くかどうかを二分法で分類し、網羅的に対処法を考える手順を STPA が提供している。このモデルと手順が、これからの複雑システムの安全論証を行う一つの有力な手段になりうる。従来の構成要素の故障要因を全て明示化し、それを最小化するという信頼性工学的な手法で複雑システムの安全論証を行おうとすると、容易に組み合わせ爆発が予想され限界が出てくるが、これを回避するためにも、Nancy G Leveson 教授の提唱する安全解析のパラダイムシフトを理解し、応用してゆくことが重要になる。

今後の、人と高度なソフトウェアが一体となった複雑システムの安全確保には、実装に先立って十分な安全解析と設計への反映が重要である。そのための手法として STAMP/STPA が有効である。活用にあたっては、システムの要求仕様、前提条件を教科書の基本を忠実に守って分析するだけでなく、システムの要求仕様、前提条件そのものの見直しまで踏み込んだ安全設計を目指してゆくことで、真の安全性の確保が実現できると考える。産業界での活用のために本書を役立ててもらえば幸いである。

さらに、IoT・AI を含んだ SoS の時代には、統一的な安全制御行動を持たない大規模かつ自律的な分散システムや、事前に完全な検証のできない複雑で高度なソフトウェアも想定される。このような複雑システムの時代に対応できる概念として Safety2.0 という協調安全の概念も紹介している。ここでの協調安全は、人間と安全制御ソフトウェアの協調だけではなく、独立に作られたソフトウェア同士の協調も含むが、このような協調により、設計時に想定しなかったような環境で安全を確保したり、安全性と経済性の両方を高めるようなシステムを作ることが可能になる。このような協調安全のシステムを実現するには、その中のサブシステム間の安全制御に関わる協調構造を分析し改善するための分析手法が必要であるが、そのヒントとなる事例や分析法も本書の中で提供している。今後の複雑システムの時代の安全を考えるきっかけにしていただけると幸いである。

参考文献

［1］［IPA2016］
「はじめての STAMP/STPA ～システム思考に基づく新しい安全性解析手法～」, IPA, http://www.ipa.go.jp/sec/reports/20160428.html, 2016

［2］［IPA2016-2］
「大規模・複雑化した組み込みシステムのための障害診断手法 Ver. 2.0」, https://www.jpa.go.jp/sec/reports/20160331_4.html , 2016

［3］［IPA2016-3］
「大規模・複雑化した組み込みシステムのための障害診断手法」ソースコードの公開、https://www.ipa.go.jp/sec/reports/20170321_2.html ,2016

［4］［IPA2017］
「はじめての STAMP/STPA（実践編）～システム思考に基づく新しい安全性解析手法～」, IPA, https://www.ipa.go.jp/sec/publish/tn17-001.html , 2017

［5］［IPA2018］
兼本茂、これからの複雑システムの安全分析 STAMP/STPA、SEC Journal、Vol.13、No.4（52 号）、pp.19-22、2018

［6］［IPA2018-2］
十 山 圭 介、 三 原 幸 博、European STAMP Workshop 2017 参 加 報 告、SEC Journal、Vol.13、No.4（52 号）、pp.51-53、2018

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

［8］［Leveson2013］
“An STPA Primer v1”, Nancy Leveson, et al, 2013 p.13 http://sunnyday.mit.edu/STPA-Primer-v0.pdf p.13

［9］［PSAS2018］
“Partnership for a Systems Approach to Safety（PSAS）web pages,”https://psas.scripts.mit.edu/home/

［10］［ホルナゲル 2013］
エリック・ホルナゲル，社会技術システムの安全分析―FRAM ガイドブック , （2013），p.25-38，海文堂出版

［11］［Thomas2013］
John Thomas.（2013）. EXTENDING AND AUTOMATING A SYSTEMS-THEORETIC HAZARD ANALYSIS FOR REQUIREMENTS GENERATION AND ANALYSIS. Ph.D Thesis,MASSACHUSETTS INSTITUTE OF TECHNOLOGY.

［12］［Thomas2015］
John Thomas, Dajiang Suo,“A Tool-Based STPA Process”, 2015.

［13］［日経 BP2015］
日経 BP Safety 2.0 プロジェクト，Safety2.0 コンセプト編，日経 BP 社（2015.12）

［14］［鉄道 2015］
鉄道信号，日本鉄道電気技術協会，（2015.12）

［15］［Mathworks2016］
「NXTway-GS のモデルベース開発～ LEGO Mindstorms NXT を用いた二輪型倒立振子ロボットの制御～」, http://www.mathworks.com ,2016

［16］［Sunouchi2017］
Ryo Sunouchi, “Hazard Analysis and Safety Design of Human-Machine Cooperative Control System”, Graduation Thesis（The University of Aizu）, March 2017.
［17］［Young2013］
William Young, Nancy Leveson.（2013）. Systems Thinking for Safety and Security. In Proceedings of the 29th Annual Computer.
［18］［Friedberg2013］
Ivo Friedberg, McLaughlin,Paul Smith,David Laverty,Sakir SezerKieran.（2017）. STPA-SafeSec: Safety and security analysis for cyber- physical systems. Journal of Information Security and Applications.
［19］［MS2018］
マイクロソフト，モノのインターネットのセキュリティアーキテクチャ：https://docs.microsoft.com/ja-jp/azure/iot-hub/iot-hub-security-architecture
［20］［Friedberg2015］
Ivo Friedberg, DavidLaverty, Kieran Mac Laughlin,Paul Smith.（2015）. A Cyber-Physical Security Analysis of Synchronous-Islanded Microgrid Operation. Proceedings of 3rd International Symposium for ICS & SCADA Cyber Security Research.
［21］［Sebastian2014］
Sebastian ElbaumS. RosenblumDavid.（2014）. Known unknowns: testing in the presence of uncertainty. Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering.
［22］［SafetyHAT2018］
“SafetyHAT：A Transportation System Safety Hazard Analysis Tool,”
https://www.volpe.dot.gov/infrastructure-systems-and-technology/advanced-vehicle-technology/safetyhat-transportation-system
［23］［XSTAMPP2018］
Asim Abdulkhaleq,“XSTAMPP For Safety Engineering of Software Intensive Systems”, http://www.xstampp.de/
［24］［Krauss2015］
Sven Stefan Krauss, Martin Rejzek, Christian Hilbes,“Tool qualification considerations for tools supporting STPA”, Procedia Engineering, Volume 128, 2015, Pages 15-24.
［25］［松本 2003］
松本雅行，森欣司；自律分散型列車制御システムにおけるアシュアランス技術と評価法，電子情報通信学会論文誌，Vol.J86-D-1，No.1、pp.14-22、（2003.1）
［26］［伊藤 2011］
伊藤桂一，東京圏輸送管理システム（ATOS）の展開と更新、JR EAST Technical Review-No.36、pp.63-66（2011.Summer）
［27］［中村 1993］
中村英夫，CARAT に関連する研究成果を概観する，鉄道総研報告，Vol.7.No.5，pp.10-16，1993.5
［28］［馬場 2012］
馬場裕一，平塚敦，佐々木英二，山本修，宮本昌和，無線を用いた列車制御システム，日立評論，Vol.94，No.06，pp.54-57、（2012.6）

［29］［中村 2016］
中村英夫，未来の鉄道システム（ICT 技術が切りひらくしなやかで強靭な鉄道）、鉄道サイバネティクス 50 周年記念論文（2016.5）
［30］［中村 2017］
中村英夫，Safety2.0 の概念と鉄道における事例，第 8 回横幹連合カンファレンス、D-4-1、（2017.12）
［31］［向殿 2017］
向殿政男，セーフティグローバル推進機構（IGSAP）における安全への取組み、第 8 回横幹連合カンファレンス，D-4-2、（2017.12）
［32］［ITU2012］
TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU（2012）"Overview of the Internet of things" p.2
［33］［中村 2017-2］
中村英夫，IoT 時代の新しい安全「Safety 2.0」の全貌　ET2017 独立行政法人 情報処理推進機構　SEC 先端技術入門ゼミ，（2017）
［34］［JR 東日本 2018］
JR 東日本，各駅の乗車人員 " http://www.jreast.co.jp/passenger/
［35］［Deleuze1984］
Gilles Deleuze & Felix Guattari（1984）. "What Is Philosophy?" p.118, Verso Books ISBN-10: 0860916863
［36］［Craig1987］
Reynolds, Craig（1987）. "Flocks, herds and schools: A distributed behavioral model". SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques. Association for Computing Machinery: 25–34. doi:10.1145/37401.37406. ISBN 0-89

索引

付録

A）JASPAR STAMP/STPA 事例

A-1. 概要

自動運転など大規模なシステムへの機能安全適用に備え、安全分析やそのレビューを補強することを目的に STAMP/STPA 事例を展開した。まず、STPA 手法に慣れることを目的に、十分実績のあるシステムである電動パーキングブレーキ[3]（EPB：Electronic Parking Brake）を題材に選んだ。

JASPAR は、機能安全開発における説明補強を期待して STPA に取り組んでおり、主に下表に述べる点で工夫を施している

表 A-1-1　JASPAR の STPA 工夫点

STPA Step	目的	工夫点
Step 0	コントロールストラクチャー記述の標準化	人、システム、車両、環境の 4 つを起点に分析対象を捉えるテンプレートや凡例を標準化した
	第 3 者への分析対象の説明性向上	コントロールストラクチャーを起点とした STPA ではなく、なぜそのコントロールストラクチャーとしたのか、作成過程の説明を追加した
Step 1	分析結果の説明性と理解容易性向上	分析結果をコントロールストラクチャー上に記述しアクシデントに至る影響を記述し、第 3 者への説明や理解を促進する手がかりとした

3　本事例で取り扱う EPB はどの自動車会社の車両にも該当しない仮想システムである。

A-2. 電動パーキングブレーキの説明

電動パーキングブレーキの説明の予備説明として、まず関連する自動車用語について説明する。これらは、分析用に必要というわけではなく自動車用語についての読者への説明としている。

●オートマチックトランスミッション車両の駐停車操作の説明

分析対象車両は、オートマチックトランスミッション（自動変速機）（以降 AT）を搭載した車両とする。一般的に AT 車を駐停車[4]する際は、以下の操作が求められる。

1. シフトポジションを P（パーキング）に入れる
2. パーキングブレーキをかける[5]
3. 車両の電源を OFF にする

（以降、ドアを開けて外に出る、施錠する）

●パーキングブレーキの説明

パーキングブレーキは、車両の駐停車状態の維持することに利用する。パーキングブレーキをかけるとタイヤがロックされ、駐停車状態が可能となる。パーキングブレーキを戻すとタイヤのロックが解除され、駐停車状態から走行に移ることができる。パーキングブレーキの例として、レバー操作によるパーキングブレーキがあげられる。レバーを引き上げるとパーキングブレーキがかかり、レバーを下げるとパーキングブレーキが解除される。

●電動パーキングブレーキ（EPB）の説明

EPB はパーキングブレーキを一部電動化したものである。本事例では、以下の操作によって動作する EPB を定義する

・EPB の作動方法
  1. フットブレーキを踏み、車を完全停止させる
  2. EPB をかける
  3. EPB が作動していることを、メータ内の EPB ランプが点灯していることで確認する
・EPB の解除方法
  1. フットブレーキを踏み、車の完全停止を維持する
  2. EPB を解除（手段は述べない）
  3. EPB が解除していることを、メータ内の EPB ランプが消灯していることで確認する

4　道路交通法第 2 条参照

5　寒冷時に電動パーキングブレーキをかけると、パーキングブレーキが凍結し、解除できなくなるおそれがあるが、本分析では寒冷時を想定せず、パーキングブレーキをかけるとこととする

A-3. アクシデント、ハザード、安全制約の識別

機能安全上取り扱っている範囲をアクシデントとした。アクシデント、ハザードの記載内容については、MIT STAMP Workshop での自動車事例を調査し、なるべく定量化につなげられ、新規システムを対象にできるよう上位概念で記載した。

表 A-3-1　アクシデント、ハザード、安全制約

アクシデント		ハザード（車両レベル、システムレベル）				安全制約	
NO	内容	NO	内容	NO	内容	NO	内容
A1	自車が他車／車以外（固定物）と衝突	H1	安全な相対的距離が確保できない	H1-1	自車が意図せず動き出す（駐停車時）	SC1	自車が意図せず動き出さないこと（駐停車時）
				H1-2	自車が意図せず急減速する（走行時、走行開始時）	SC2	自車が意図せず急減速しないこと（走行時、走行開始時）

A-4. 分析対象の把握

分析内容によっては、Safety Control structure（参考文献 X 参照）を定義し、システム開発とシステム運用の両面から分析が必要な場合がある。それには、ライフサイクル全体でステークホルダを抽出し、要求や制約を考えることが必要な場合がある。例えば、EPB は、自動車の運転シーンだけでなく、自動車製造工場生産時、メカニックによるメンテナンス時にも機能を持っており、従来の機能安全開発では検討範囲に入れている。今回は、下表に示す EPB の運用フェーズの一部のみを対象とした。

表 A-4-1　安全制約と各 Step

Step	概要	安全制約	
		SC1	SC2
		自車が意図せず動き出さないこと（駐停車時）	自車が意図せず急減速しないこと（走行時、走行開始時）
Step 0	コントールストラクチャの作成		
Step 1	UCA の抽出		
Step 2	UCA の要因特定		

表 A-4-1 における安全制約を、機能安全開発における安全目標と同等と捉え、以下のように安全制約別にアーキテクチャを定義し、安全分析を行い、安全要求を導出する。

1. SC1 の分析：自車が意図せず動き出さないこと
2. SC2 の分析：自車が意図せず急減速しないこと

A-5. Step 0: コントロールストラクチャーの作成

コントロールストラクチャーは分析対象を把握した結果と捉えることができる。分析対象によっては、コントロールストラクチャーの作成過程を示す必要がある。その手段として参考文献（はじめての STAMP/STPA）では、SysML などが紹介されている。これらのダイアグラムは必ずしも必要でない場合もあるし、表や図（ポンチ絵）で示した方が適切な場合もあると考える。

今回は、コントロールストラクチャーのコンポーネントを抽出するためポンチ絵を描くことから Step 0 を着手した。まず、人、システム、車両、環境の 4 つを起点に考える。

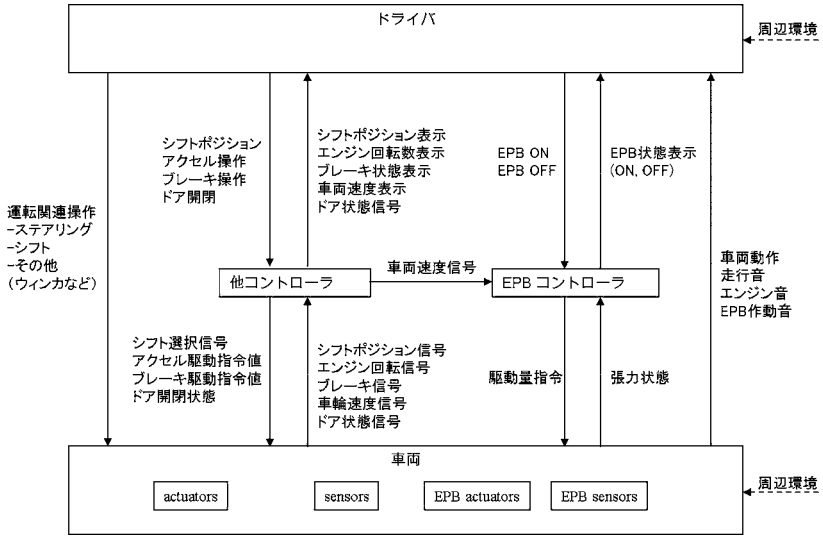


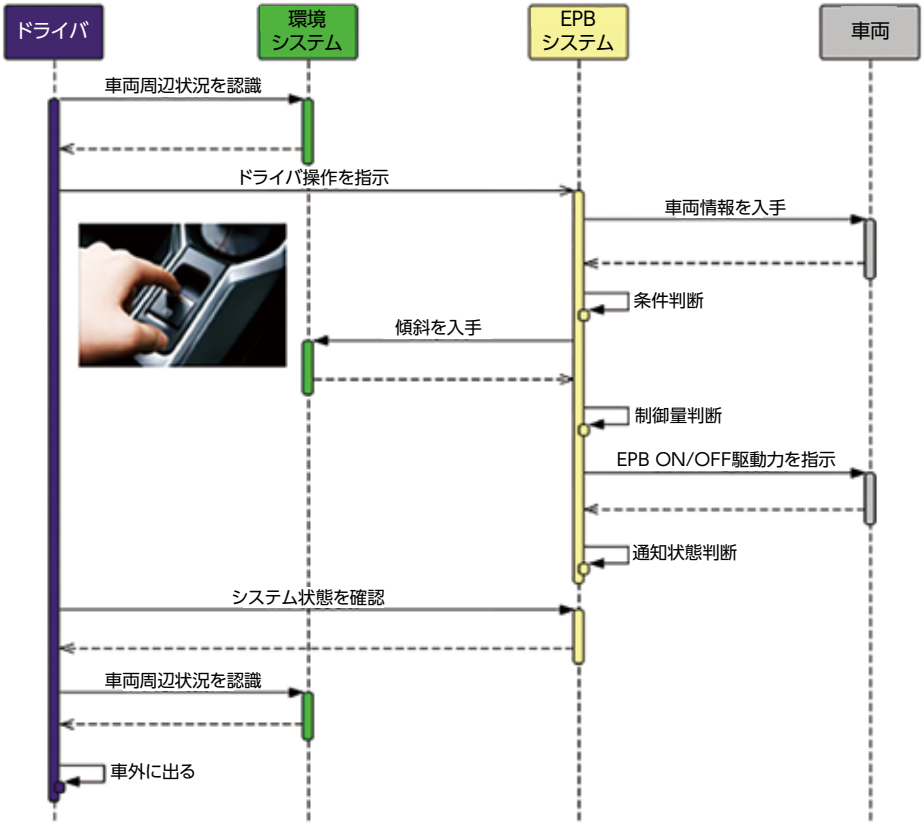
図 A-5-1　コンポ―ネント抽出のためのポンチ絵

図 A-5-1 は、EPB に関係したコンポーネントと相互作用について知っていることを図示したポンチ絵である。これを事前のアーキテクチャ想定として表 A-4-1 に示した機能ごとにコントールストラクチャを作成する。

A-5-1. SC1 関連のコントロールストラクチャー

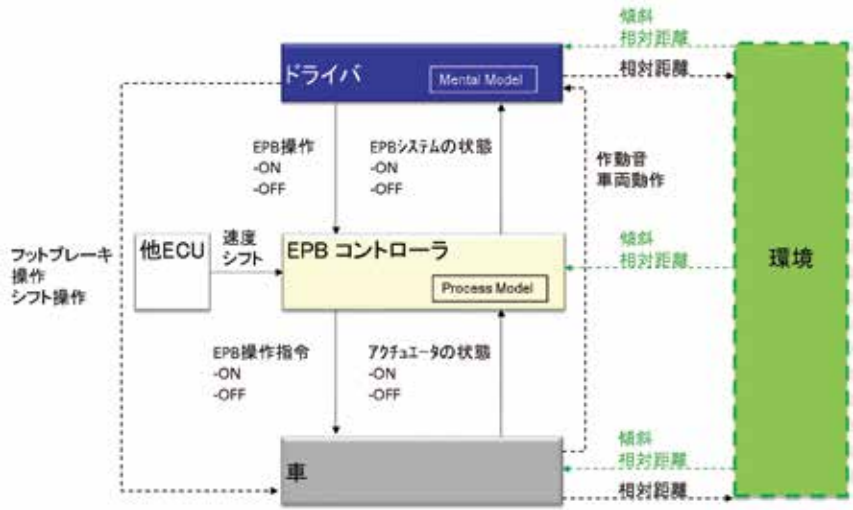
この機能の想定は、「ドライバーが自車を傾斜地にフットブレーキで停車させ、シフトポジションを P にセット。その後パーキングブレーキを効かせ、車両電源を OFF し、ドアをあけて車外に出る。」ことである。安全設計上厳しいシーンを抽出すべく、シフトポジションが本来の P ではなく、P 以外にセットされていることを前提とし、分析を行う。

まず、人、システム、車両、環境の 4 つをコンポーネントとして、その相互作用を特定すべくシーケンス図を作成することで特定した（図 A-5-2）。



（前提：シフトポジションが本来期待される P ではなく、N にセットされている）
図 A-5-2　SC1 関連の相互作用特定

図 A-5-2 から得られた相互作用をテンプレートにあてはめコントロールストラクチャー図 A-5-3 を得た。アクシデントに至るシナリオを記述するため、ドライバーと環境の間に相対距離と記載し、ドライバーが車両を降りて車両から離れていくことを記載した。凡例については、「表 2.3-7　コントロールストラクチャー凡例」を参照。



（前提：シフトポジションが本来期待される P ではなく、N にセットされている）

図 A-5-3　SC1 関連のコントロールストラクチャー

A-5-2. SC2 関連のコントロールストラクチャー

この機能の想定は、「ドライバーが駐停車している自車の電源を ON にし、フットブレーキを踏み、エンジンを始動し、パーキングブレーキが効いている状態を解除し、シフトポジションを P から D にセット。その後、フットブレーキから足を外して、アクセルペダルを徐々に踏み込み車両を走行状態にする。」ことである

A-5-1. と同様に、各コンポーネントにおける相互作用を特定すべくシーケンス図を作成することで特定した。

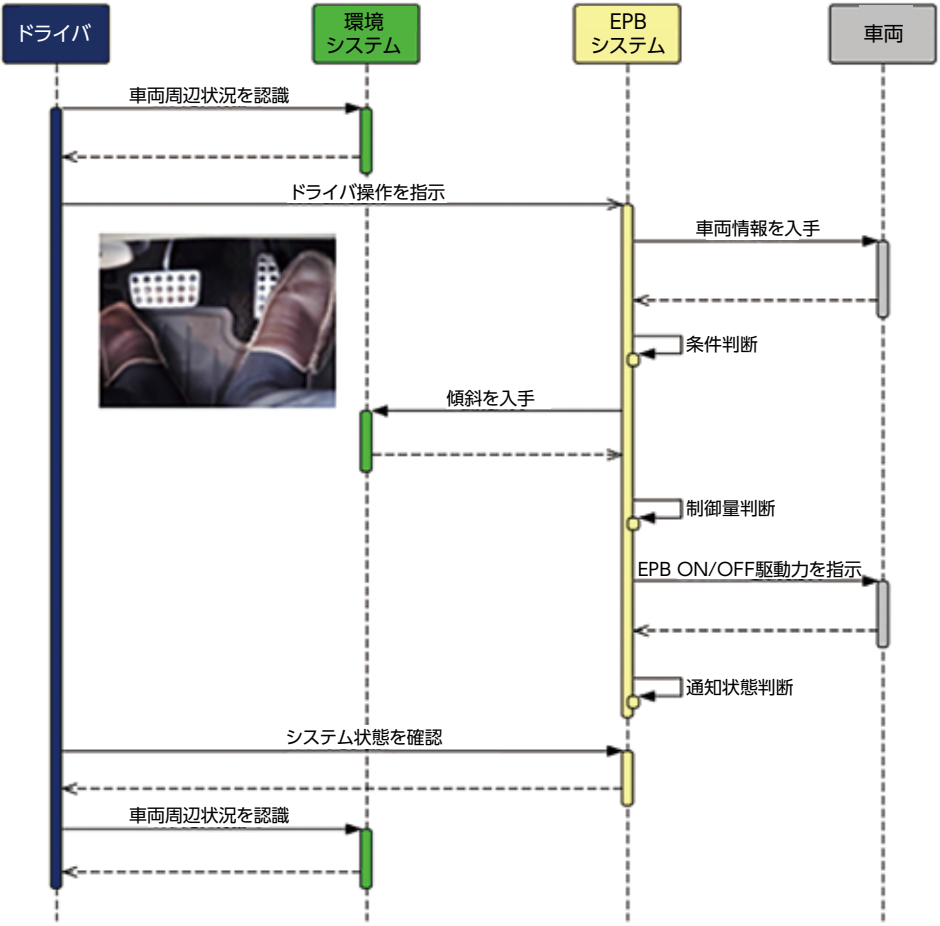


図 A-5-4　SC2 関連の相互作用特定

次に、A-5-1. と同様にして、図 A-5-4 から得られた相互作用をテンプレートにあてはめコントロールストラクチャー図 A-5-5 を得た。

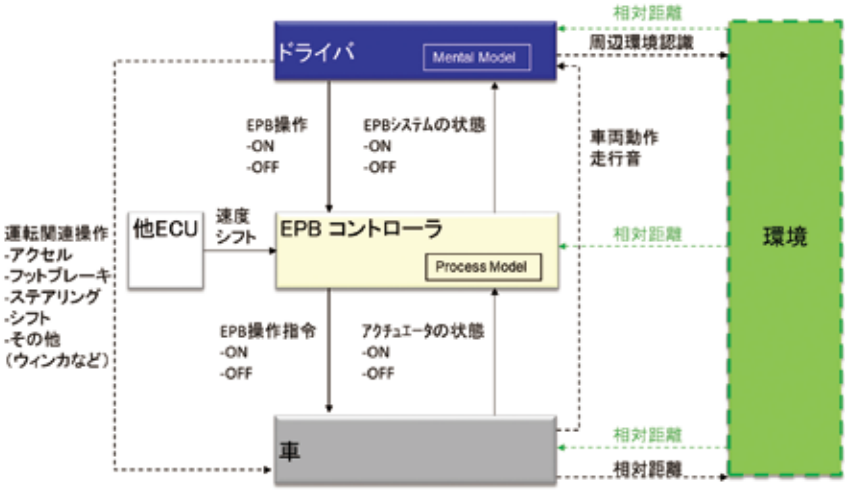


図 A-5-5　SC2 関連のコントロールストラクチャー

A-6. Step 1: UCA の抽出

図 A-5-3 と図 A-5-5 に記載されているコントロールアクションに対して、4 つのカテゴリをあてはめ UCA を特定した。

A-6-1. SC1 侵害の UCA 特定

SC1 自車が意図せず動き出さないこと（駐停車時）を侵害する UCA のみ UCA の ID を付与し、侵害しないものは「-」を付与した。

表 A-6-1　SC1 侵害の UCA の抽出

コントロールアクション	コントロールアクションを与えないとハザード（Not providing causes hazard）	コントロールアクションを与えるとハザード（Providing causes hazard）	コントロールアクションを与えるのが早すぎ / 遅すぎでハザード（Incorrect Timing/Order）	コントロールアクションを与えるのが長すぎ / 短すぎでハザード（Stopped Too Soon / Applied too long）
CA1-1 パーキングブレーキをかける操作をする （EPB ON）	UCA1-1_NP1 ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキをかける操作をしない	- ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキをかける操作をする（正常）	- （ドライバーが車を降りてからはパーキングブレーキの操作ができない）	UCA1-1_D1 ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキが効く前に、パーキングブレーキをかける操作を停止する
CA1-2 パーキングブレーキを解除する操作をする （EPB OFF）	- ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキを解除する操作をしない（正常）	UCA1-2_P1 ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキを解除する操作をする	- （ドライバーが車を降りてからはパーキングブレーキの操作ができない）	- ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキが解除する前に、パーキングブレーキを解除する操作を停止する
CA2-1 パーキングブレーキをかける （EPB ON）	UCA2-1_NP1 EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキをかけない	- EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキをかける（正常）	UCA2-1_T1 EPB コントローラーは、傾斜地上の車両からドライバーが降りた後に、パーキングブレーキをかける	UCA2-1_D1 EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキが効く前に、パーキングブレーキをかけるのをやめる
CA2-2 パーキングブレーキを解除する （EPB OFF）	- EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキを解除しない（正常）	UCA2-2_P1 EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキを解除する	UCA2-2_T1 EPB コントローラーは、傾斜地上の車両からドライバーが降りた後に、パーキングブレーキを解除する	- EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキが解除する前に、パーキングブレーキを解除するのをやめる

＊前提：シフトポジションが本来期待される P ではなく、N にセットされている
＊ UCA ID は元の CA と下記カテゴリと対応させ、追跡および遡及が容易にできるよう工夫した。

表 A-6-2　SC2 侵害の UCA の抽出

コントロールアクション	コントロールアクションを与えないとハザード（Not providing causes hazard）	コントロールアクションを与えるとハザード（Providing causes hazard）	コントロールアクションを与えるのが早すぎ / 遅すぎでハザード（Incorrect Timing/ Order）	コントロールアクションを与えるのが長すぎ / 短すぎでハザード（Stopped Too Soon / Applied too long）
CA3-1 パーキングブレーキをかける操作をする （EPB ON）	- ドライバーは、自車が走行中に、パーキングブレーキをかける操作をしない（正常）	UCA3-1_P1 ドライバーは、自車が走行中に、パーキングブレーキをかける操作をする	UCA3-1_T1 ドライバーは、自車が停車する前に、パーキングブレーキをかける操作をする	- ドライバーは、自車が走行中に、パーキングブレーキをかける操作をするが、パーキングブレーキがかかる前に操作をやめる
CA3-2 パーキングブレーキを解除する操作をする （EPB OFF）	- ドライバーは、自車が走行前に、パーキングブレーキを解除する操作をしない（走行できない、発熱などへの影響が考えられる）	- ドライバーは、自車が走行中に、パーキングブレーキを解除する	- ドライバーは、自車が走行開始した後に、パーキングブレーキを解除する	- ドライバーは、自車が走行開始時に、パーキングブレーキを解除する操作をするが、パーキングブレーキが解除する前に操作をやめる
CA4-1 パーキングブレーキをかける （EPB ON）	- EPB コントローラーは、自車が走行中に、パーキングブレーキをかけない（正常）	UCA4-1_P1 EPB コントローラーは、自車が走行中に、パーキングブレーキをかける	UCA4-1_T1 EPB コントローラーは、自車が停止する前にパーキングブレーキをかける	- EPB コントローラーは、車が走行中に、パーキングブレーキをかけるようとするが、パーキングブレーキがかかる前にやめる
CA4-2 パーキングブレーキを解除する （EPB OFF）	- EPB コントローラーは、自車が走行前に、パーキングブレーキを解除しない（走行できない、発熱などへの影響が考えられる）	- EPB コントローラーは、自車が走行中に、パーキングブレーキを解除する	- EPB コントローラーは、自車が走行開始した後に、パーキングブレーキを解除する	- EPB コントローラーは、車が走行開始時に、パーキングブレーキを解除しようとするが、パーキングブレーキが解除する前にやめる

＊ UCA ID は元の CA と下記カテゴリと対応させ、追跡および遡及が容易にできるよう工夫した。

A-7. Step 2 要因特定

次に、Step 1 で導出した安全制約が侵害される要因を特定する。

表 A-7-1　SC1 侵害の UCA に至る要因特定

ID_UCA	UCA	EPB コントローラー関連		ドライバー関連	
		HCF	シナリオ	HCF	シナリオ
UCA1-1_NP1	ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキをかける操作をしない	EPB コントローラーのパーキングブレーキ状態の誤表示とドライバーの不適切な操作	EPB コントローラーが、パーキングブレーキ OFF にもかかわらず ON を表示。ドライバーは、パーキングブレーキが効いていると思い込み、パーキングブレーキをかける操作をせずに降車する。	ドライバーの不適切な操作（思い込みや忘れ）	ドライバーは、パーキングブレーキが効いていると思い込み、パーキングブレーキをかける操作をせずに降車する。
UCA1-1_D1	ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキが効く前に、パーキングブレーキをかける操作を停止する	EPB コントローラーのパーキングブレーキ状態の誤表示とドライバーの不適切な操作	EPB コントローラーが、パーキングブレーキが効く前に ON を表示。ドライバーは、パーキングブレーキが効いたと思い、操作をやめる。	ドライバーの不適切な操作（操作不足、理解不足）	1）ドライバーは、パーキングブレーキが効いていると思いこみ操作を途中でやめる。 2）操作の仕方がうまく分からず、操作が不十分なまま降車する
UCA1-2_P1	ドライバーは、自車が傾斜地に駐停車中に、パーキングブレーキを解除する操作をする	なし	なし	ドライバーの不適切な操作	ドライバーは、パーキングブレーキをかける際に、間違えてパーキングブレーキを解除する操作を行い、降車する。
UCA2-1_NP1	EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキをかけない	1）EPB コントローラーの操作検出部の不良 2）EPB コントローラーの判断部の不良 3）EPB コントローラー出力部の不良	1）EPB コントローラーが、パーキングブレーキをかける操作を検出できず、パーキングブレーキをかけることができない 2）EPB コントローラーが、パーキングブレーキをかける判断をすべき時に、判断しない / できない 3）EPB コントローラーが、パーキングブレーキをかけるべきときに、かけない / かけることができない	ドライバーの不適切な操作（思い込みや忘れ）	ドライバーは、パーキングブレーキが効いていると思い込み、パーキングブレーキをかける操作をせずに降車する。

ID_UCA	UCA	EPB コントローラー関連		ドライバー関連	
		HCF	シナリオ	HCF	シナリオ
UCA2-1_T1	EPB コントローラーは、傾斜地上の車両からドライバーが降りた後に、パーキングブレーキをかける	1) EPB コントローラーの操作検出部の不良 2) EPB コントローラーの判断部の不良 3) EPB コントローラー出力部の不良	1) EPB コントローラーが、遅れてパーキングブレーキをかける操作を検出し、ドライバーが降りた後に、パーキングブレーキをかける 2) EPB コントローラーが、遅れてパーキングブレーキをかける判断をし、ドライバーが降りた後に、パーキングブレーキをかける 3) EPB コントローラーが、遅れてパーキングブレーキをかけ、ドライバーが降りた後に、パーキングブレーキをかける	なし	なし
UCA2-1_D1	EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキが効く前に、パーキングブレーキをかけるのをやめる	1) EPB コントローラーの操作検出部の不良 2) EPB コントローラーの判断部の不良 3) EPB コントローラー出力部の不良	1) EPB コントローラーが、パーキングブレーキをかける操作を検出できず、パーキングブレーキをかけることができない 2) EPB コントローラーが、パーキングブレーキをかけ続ける判断をすべき時に、途中でやめる 3) EPB コントローラーが、パーキングブレーキをかけるべきときに、かけるのをやめる	なし	なし

ID_UCA	UCA	EPB コントローラー関連		ドライバー関連	
		HCF	シナリオ	HCF	シナリオ
UCA2-2_P1	EPB コントローラーは、自車が傾斜地に駐停車中に、パーキングブレーキを解除する	1) EPB コントローラーの操作検出部の不良 2) EPB コントローラーの判断部の不良 3) EPB コントローラー出力部の不良	1) EPB コントローラーが、ドライバー操作していないにもかかわらず、パーキングブレーキを解除する操作を検出する 2) EPB コントローラーが、パーキングブレーキを解除すべきでない時に解除する判断をする 3) EPB コントローラーが、パーキングブレーキを解除すべき時に解除する	ドライバーの不適切な操作	ドライバーは、操作の仕方がうまく分からず、走行中にパーキングブレーキを解除する操作をする
UCA2-2_T1	EPB コントローラーは、傾斜地上の車両からドライバーが降りた後に、パーキングブレーキを解除する	1) EPB コントローラーの操作検出部の不良 2) EPB コントローラーの判断部の不良 3) EPB コントローラー出力部の不良	1) EPB コントローラーが、ドライバー操作していないにもかかわらず、パーキングブレーキを解除する操作を検出する 2) EPB コントローラーが、パーキングブレーキを解除すべきでない時に解除する判断をする 3) EPB コントローラーが、パーキングブレーキを解除すべき時に解除する	なし	なし

表 A-7-2　SC2 侵害の UCA に至る要因特定

ID_UCA	UCA	EPB コントローラー関連		ドライバー関連	
		HCF	シナリオ	HCF	シナリオ
UCA3-1_P1	ドライバーは、自車が走行中に、パーキングブレーキをかける操作をする	なし	なし	1）ドライバーの不適切な操作（理解不足） 2）緊急時における使用	1）ドライバーが、操作の仕方がうまく分からず、走行中にパーキングブレーキをかける操作をする 2）ドライバーが、緊急事態でフットブレーキの代わりにパーキングブレーキを使用する
UCA3-1_T1	ドライバーは、自車が停車する前に、パーキングブレーキをかける操作をする	なし	なし	1）ドライバーの不適切な操作（理解不足） 2）緊急時における使用	1）ドライバーが、操作の仕方がうまく分からず、走行中にパーキングブレーキをかける操作をする 2）ドライバーが、緊急事態でフットブレーキの代わりにパーキングブレーキを使用する
UCA4-1_P1	EPB コントローラーは、自車が走行中に、パーキングブレーキをかける	1）EPB コントローラーの操作検出部の不良 2）EPB コントローラーの判断部の不良 3）EPB コントローラー出力部の不良	1）EPB コントローラーが、ドライバーが操作していないにもかかわらず、パーキングブレーキをかける操作を検出する 2）EPB コントローラーが、パーキングブレーキをかけるべきでない時に、パーキングブレーキをかける判断をする 3）EPB コントローラーが、パーキングブレーキを解除継続すべき時にパーキングブレーキをかける	1）ドライバーの不適切な操作（理解不足） 2）緊急時における使用	1）ドライバーは、操作の仕方がうまく分からず、走行中にパーキングブレーキをかける操作をする 2）緊急事態でフットブレーキの代わりにパーキングブレーキを使用する

ID_UCA	UCA	EPB コントローラー関連		ドライバー関連	
		HCF	シナリオ	HCF	シナリオ
UCA4-1_T1	EPB コントローラーは、自車が停止する前にパーキングブレーキをかける	1）EPB コントローラーの操作検出部の不良 2）EPB コントローラーの判断部の不良 3）EPB コントローラー出力部の不良	1）EPB コントローラーが、ドライバーが操作していないにもかかわらず、パーキングブレーキをかける操作を検出する 2）EPB コントローラーが、パーキングブレーキをかけるべきでない時に、パーキングブレーキをかける判断をする 3）EPB コントローラーが、パーキングブレーキを解除継続すべき時にパーキングブレーキをかける	なし	なし

A-8. EPB 事例の考察

なお、JASPAR における STAMP/STPA の EPB への試行事例では、一部仕様変更された事例についても試行した。その事例とは、発進時にドライバーがアクセルペダルを踏むとパーキングブレーキが自動解除されるという事例である。この事例に対して、STAMP/STPA を試行すると、「走行開始時にブレーキが遅くに開放され、意図せぬ急加速発生（ドライバーがアクセルをさらに強く踏み込むなど）」という UCA を追加抽出できた。ドライバーとのインタラクションおよびタイミング不正に関する UCA を検討することで、新たなハザードに気付いて追加した。

B）JASPAR の EPB 事例分析における Q&A

本付録では JASPAR における EPB 事例分析中に生じた疑問について、その回答例とともに英語表記のアルファベット順で紹介する。

1. Accident（アクシデント）

Q1-1：Accident とはどのような事象が該当するのか？

A1-1：Accident とは人命、身体的傷害、財産などの損失につながる（危険）事象が該当する。

ISO 26262 では、人命、身体的傷害に関わる事象ごとに過酷度（S）、暴露確率（E）、制御可能性（C）の 3 つから評価し、ASIL（自動車安全度水準）として 4 つに分類する。STPA では、アクシデントに至る可能性のあるハザード状態、それを引き起こす非安全制御行動（UCA）、UCA を誘発する要因からなる一連のシナリオを Worst Case で考えて定性的に分析する。

2. Control Action（コントロールアクション）

Q2-1：Control Action の分析対象数が多くなる場合など、作業を合理化する方法はないか？

A2-1：すべての Control Action を分析する必要があるが、ソフトウェアの支援ツールを用いて作業は合理化できる可能性がある。

また、コンポーネントの相互作用で、安全制御に関連する Control Action とその他のフィードバック情報を区別し、安全関連 Control Action のみに注目して分析をすることで作業を合理化できる。さらに、コンポーネントの機能に着目した抽象化・階層化をすることで、安全制御に影響するシステム全体の制御構造図を簡略化・可視化でき、作業の合理化もできる。

3. Control Loop（コントロールループ）

Q3-1：Control Loop にはどのような抽出方法があるのか？

A3-1：1 つの案として N スイッチカバレッジ（状態遷移のカバレッジの考え方。N スイッチカバレッジは（N+1）回のイベントで遷移する経路を全て網羅するカバレッジのこと）のような形で、Control Structure から機械的に Control Action と Feedback をたどりながら Loop を抽出する方法がある。

Q3-2：Control Loop の網羅性はどのように説明すればよいか？

A3-2：すべてのコンポーネントを起点として、機械的に Control Loop を抽出することで網羅性を主張する方法がある。まず、安全論証したいアイテム内のすべての Control Loop を検証し、次にアイテム外のコンポーネントとのインタラクションを検証することで説明できる。ただし、一般的に網羅性で抜け漏れがないことを説明することは容易ではない。

4. Control Structure（コントロールストラクチャー）

Q4-1：STPA に取組むにあたり、SysML（Systems Modeling Language）や SCDL（Safety Concept Description Language）などの表記法で Control Structure の代用をすることは可能か？

A4-1：Control Structure を代用することは難しい。SysML の内部ブロック図や SCDL と比べ、Control Structure は Feedback Loop を明確に表現できる点に優位性がある。

Q4-2：Control Structure を描くにあたり、推奨するコンポーネント数は幾つぐらいか？

A4-2：コンポーネント数は 4 ～ 5 つ程度が好ましい。初めはできるだけシンプルな構成で分析することがポイントである。コンポーネント数が多くなる場合には、階層モデルによ

る分析を推奨する。階層モデルでは、幾つかのコンポーネントをまとめた上位層での分析をおこない、個々のコンポーネントをさらに下位層で分割して分析する。

5. Controller（コントローラー）

Q5-1：ドライバーと EPB システムの関係ではどちらが Controller となるのか？
A5-1：安全分析を行う設計思想によって異なる。ドライバー判断を優先する場合には、ドライバーが Controller で EPB システムが制御対象となる。また、EPB システムの判断を優先する場合には、EPB システムが Controller でドライバーはその入力となる。

6. Environment Component（環境コンポーネント）

Q6-1：環境コンポーネントとは何を指すのか？
A6-1：アイテムの意図した機能（主機能）が振る舞う際の環境要因（道路、歩行者、天候、後部座席の乗員など）を指す。

Q6-2：システムからドライバーへの表示に関する Feedback は、システムから直接ドライバーに返すのではなく、システム⇒環境⇒ドライバーという Loop で捉えてはどうか？
A6-2：ドライバーとシステムの間のインタラクションを把握しやすくするためには、システムから直接ドライバーに Feedback を返す Loop で検討する方がよい。

7. Feedback（フィードバック）

Q7-1：コンポーネントの Control Action は Unsafe Control Action の分析対象であるが、Feedback も分析対象となるのか？
A7-1：Feedback は Unsafe Control Action の分析対象外であるが、分析時に考慮する必要がある。

Q7-2：表示装置は Controller が人に伝える（つまり人への）Feedback として表現してもよいのでは？
A7-2：Unsafe Control Action 抽出段階における Control Structure においては、人への Feedback としてもよい。

8. HCF：Hazard Causal Factor（ハザード要因）

Q8-1：HCF 表の用途や注意点は何か？
A8-1：分析結果のまとめとして HCF 表を利用する。しかし、分析開始時に HCF 表を使うと、表のセルを埋めることが目的になってしまい、自由な発想に基づいた分析に制限をかけてしまうことがある。

Q8-2：HCF のヒントワード（Control Loop で安全制約を侵害する原因）のうち、「(10) 識別されないか、範囲外の妨害」の観点は、どのように抽出するとよいか？
A8-2：Control 対象のプロセス範囲外のモノから何らかの侵害があったという観点で捉え、抽出してみるとよい。

Q8-3：HCF のヒントワードを使って分析する嬉しさは何か？
A8-3：HCF のヒントワードを Control Structure と照らし合わせると、どの箇所をどの観点で分析しているかが分かりやすい。また、ヒントワードを使って分析しているため、抜け漏れが生じにくく、場当たり的な分析ではないことを示すことができる。しかし、分析対

象に応じてヒントワードが異なることが一般的なので、適切なヒントワードを選択する必要がある。

Q8-4：Loop のすべての競合／干渉を分析対象とすると膨大な量になり、集中力を持続しながらすべての分析をやり切ることは現実的に難しくはないか？
A8-4：分析対象が膨大となってしまうが、完成車メーカーとサプライヤーで分担するなど、負担の軽減方法を工夫するとよい。

Q8-5：HCF の分析対象を絞り込む方法はないか？
A8-5：インターフェースに着目して問題となりそうな箇所を絞り込むなどの方法がある。

9. Process Model（プロセスモデル）
Q9-1：Process Model には何を記述すればよいのか？
A9-1：Controller が制御対象をどのように見ているかの記述であり、その動作に影響があるパラメータや状態を記述する。

10. Sequence Diagram（シーケンス図）
Q10-1：シーケンス図は時間的な観点しか表現できないため、状態遷移などの検証も必要ではないか？
A10-1：シーケンス図は Control Action 抽出のために利用し、Control Structure にステートマシン図を併記しながら分析することも有効な方法である。例えば、Control Structure とステートマシン図を重ねて Unsafe Control Action を抽出し、それを UCA 表に整理するというアプローチもある。

11. STAMP/STPA（スタンプ／エステーピーエー）
Q11-1：STAMP/STPA による分析が難しいと感じたときはどのような工夫があるのか？
A11-1a：車速センサーや傾斜角センサーといった、細かい要素を扱うレベルから Control Structure による分析を行うのではなく、ドライバーモデルを扱う抽象的なレベルから分析を始めるとよい。
A11-1b：STPA で一連の分析を実施したのち、Control Structure に必要な安全機構を追加して、再度分析するとよい。

Q11-2：STAMP/STPA は実際の開発現場で実施可能か？また注意点は例えば何か？
A11-2：HAZOP と比べるとガイドワード数が少なく取り組みやすいことなど、実際に開発現場に導入する際には初めに STAMP/STPA のよさをうまく伝えるとよい。また、ハザード発生メカニズムの知識があれば、ある程度は一人でも分析できるが、複数人で取り組む方がよい。

Q11-3：STAMP/STPA を実施する嬉しさはどこにあるのか？
A11-3a：Unsafe Control Action は、Control Structure から抽出できること。
A11-3b：ドライバーモデルを分析範囲に含められること。（Control Structure でドライバーを登場させること。ドライバーモデルが存在しないと従来手法と同じ視点／結果になってしまう可能性がある。）
A11-3c：Control Structure において、どの Control Action が Unsafe Control Action かを明示できるため、第三者への説明性が向上すること。

A11-3d：機能安全で段階的に詳細化／具体化するように、STAMP/STPA においてもコンセプト層、システム層などの各階層において Control Structure を描くと、アーキテクチャの階層構造が提示できるため説明性が向上すること。

Q11-4：STAMP/STPA の分析結果は誰が実施しても同じになるのか？
A11-4：STAMP/STPA は STEP0（コントロールストラクチャ等）から分析結果が一意に決まるわけではなく、正解があるわけでもない。よって、同じ自動車の EPB 事例を分析しても（仮にアクシデント、ハザード、安全制約を同じだと仮定しても）異なる結果となる。

12. Support Tools（支援ツール）

Q12-1：ソフトウェアの支援ツールを使うメリットとは例えば何か？
A12-1：使用するソフトウェアツールによって様々なメリットがある。例えば、Control Structure などの図を描く際の手間が省ける。

13. UCA：Unsafe Control Action（非安全なコントロールアクション）

Q13-1：UCA のガイドワードのうち、Not Providing ／ Providing causes Hazard は、それぞれどのような意味をもつのか？
A13-1：表 A-8-1 を参考にするとよい。

表 A-8-1　Not Providing ／ Providing causes Hazard

ガイドワード	意図	詳細ガイドワード	事例
Not Providing	実施	No	制御行動が意図に反して行われなかった
Providing causes Hazard	量	More	制御行動が意図に反して多く行われた
		Less	制御行動が意図に反して少なく行われた
	質	As well as	意図に反して複数の制御行動が正しく実施された
		Part of	制御行動が意図に反して一部のみ行われた
	判断／選択	Reverse	制御行動が意図と異なる論理で行われた
		Other than	間違って選択された制御行動が正しく実施された

Q13-2：Providing causes hazard の Unsafe Control Action を検討する際、アルゴリズムを「Control Action の実行条件」と「ロジック」とに分けて検討する利点は何か？
A13-2：Providing causes hazard を考える際、ある実行条件が満たされないときにどうなるかを考え、実行条件（制御のトリガー／イベント）を受けて実行される処理をロジックとして記載できるようにした。
　これは、STAMP/STPA に慣れている場合は分析者の頭の中で同様な思考を行っているた

めに手間がかかるように見えるが、STAMP/STPA に慣れていない場合には分析漏れを防ぐことにつながり利点となる。

Q13-3：対策の導出は必要なのか？
A13-3：分析だけで終わるのではなく、処置が必要なものについては対策の導出までやりきることを推奨する。

Q13-4：必要なブレーキトルクに至るまでのトルク上昇勾配が、想定より緩やかあるいは急峻な場合（変化スピードの異常）は、どのガイドワードで導出するとよいのか？
A13-4：どのような場合も 4 つのガイドワードに当てはまる。この場合には、必要なブレーキトルクに至るまでの時間が Too late（遅すぎでハザード）、あるいは Too early（速すぎでハザード）ということで導出できる。また、Providing causes Hazard において、ブレーキトルク量が Less（少なすぎてハザード）あるいは More（多すぎてハザード）を用いることでも導出できる。

Q13-5：シーケンス図での検討には、幾多ものシーケンス記述が必要になるのではないか？
A13-5：そのような面はあるが、客観的に安全性を説明するためには重要となる基本シナリオをシーケンス図でおさえておくことには意味がある。

C）用語説明

Accident
　望ましくない事象、事故・損失。アクシデント。

Hazard
　アクシデントが潜在している具体的な状態。ハザード。

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

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

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

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

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

プロセスモデル
　コントローラーが想定する被コントロールプロセスの状態。

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

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

分析・解析

「分析」と「解析」の使い分けには例えば次のような定義もある。
分析：見えるものを分けるのが分析
解析：（安全性のように）見えないものを分けるのが解析
しかし、数値や数式は見えるが「数値解析」、ハザードは見えないが「ハザード分析」のように上記定義に当てはまらない一般的な慣用句が数多くある。つまり、「分析」と「解析」の使い分けに関して汎用的で適当な定義は無い。
そこで、本書では「分析」、「解析」のいずれを用いても誤解なく意味が通じるところでは「解析」を用い、「分析」が一般的なところでは「分析」を用いている。
また、同じ用語であっても分野によって用い方の慣習が異なる場合もある。そこで、本書の事例解説においてはそれぞれの分野の慣習にならうこととした。

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
	大原 衛	地方独立行政法人東京都立産業技術研究センター
	岡野 浩三	国立大学法人信州大学
	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
	杉浦 弘人	東日本旅客鉄道株式会社
	北村 知	東日本旅客鉄道株式会社
	日下部 茂	公立大学法人長崎県立大学
	小松原 明哲	早稲田大学理工学術院
	高橋 信	東北大学大学院
	中村 英夫	日本大学
	中村 洋	株式会社レンタコーチ
	野本 秀樹	有人宇宙システム株式会社
	向山 輝	日本電気株式会社
	余宮 尚志	株式会社 東芝

IPA/SEC（50 音順）

石井 正悟
金子 朋子
十山 圭介
松田 充弘
三縄 俊信
三原 幸博

協力者（50 音順）

岡田 学	一般社団法人 JASPAR　機能安全 WG
河野 文昭	一般社団法人 JASPAR　機能安全 WG
高田 哲也	株式会社京三製作所

はじめての STAMP/STPA（活用編）
～システム思考で考えるこれからの安全～

2018 年 5 月　第 1 刷発行

発行　独立行政法人 情報処理推進機構（IPA）

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

制作　株式会社サンワ　Printed in Japan

「はじめての STAMP/STPA」入門編

- STAMP を**理解する**ための STPA 手順解説書
- 教科書に近い分かり易い事例を用い、勘所を交えて STPA の手順を具体的に解説
- 2016 年 3 月公開

http://www.ipa.go.jp/sec/reports/20160428.html

「はじめての STAMP/STPA（実践編）」

- STAMP を**やってみる**ための STPA 事例解説書
- 教科書通りにはいかない産業界の事例を用い、STPA の効果的な活用方法を具体的に解説
- 2017 年 3 月公開

http://www.ipa.go.jp/sec/reports/20170324.html

「はじめての STAMP/STPA（活用編）」

- STAMP を**当たり前にやる**ための STPA 事例解説書
- 産業界での試行事例、人と機械の協調による安全制御の事例、セーフティとセキュリティの統合分析事例を解説
- 将来の複雑システムの安全解析の在り方に関するビジョンを提言
- 2018 年 3 月公開

http://www.ipa.go.jp/sec/reports/20180328_2.html

STAMP 支援ツール

STAMP Workbench

- 産業界の実システムで使える STAMP に特化したモデリングツール
- 単純作業を極力自動化し、分析者は試行に専念
- オープンソースソフトウェアとして無償で公開

https://www.ipa.go.jp/sec/tools/stamp_workbench.html

ISBN978-4-905318-61-3
C3055 ¥556E
SEC-TN18-003

定価：本体 556 円＋税

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

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