『IoT時代』の
セーフティ＆セキュリティ

2021年1月22日

国立情報学研究所
金子朋子

NII
National Institute of Informatics

目次

自己紹介

Ⅰ. IoTとセキュリティ・バイ・デザイン

Ⅱ. IoT高頼化機能要件

Ⅲ. 安全性理論：

　STAMPとレジリエンス・エンジニアリング

Ⅳ.セーフティとセキュリティ統合リスク分析

自己紹介

金子朋子　博士（情報学）

・公認情報セキュリティ監査人（CAIS）
・国立情報学研究所特任研究員（NTTデータより出向）
・日科技連SQiP研究会セーフティ&セキュリティ分科会主査
・情報セキュリティ大学院大学　客員研究員
・東京電機大学サイバーセキュリティ研究所研究員
・電子情報通信学会知能ソフトウェア工学研究会　幹事
・日本セキュリティマネジメント学会ITリスク学研究会幹事
・機械学習工学研究会機械学習システム
　セーフティ・セキュリティWG幹事

Eric Hollnagel 教授と共に

Nancy　Leveson教授と共に

研究テーマ：セーフティ＆セキュリティ
　　セキュア開発方法論
　　機械学習システムの安全性

- NTTデータで長年SEとして勤務
- 2017-2019年(独)情報処理推進機構研究員

https://researchmap.jp/knktmk/(研究業績一覧)

セーフティ・セキュリティとの関わり

1990年：社会問題化したパチンコプリペイドカード事件の開発チームに
→日本で最初の大規模なセキュリティ攻撃→何の対応もできない苦い経験

2003-2007年：社内ボランティアでテレワーク制度化→情報漏洩対策に試行錯誤、リモート技術調査とアセスメント・実証実験の実施、「テレワーク推進賞」を受賞

2008-2014年：情報セキュリティ大学院大学初の女性博士号取得者に

2016年−2019年：「つながる世界の開発指針」等の執筆や世界のセーフティの2大巨匠のシステム理論、レジリエンスエンジニアリングの普及展開に従事

2017年−セーフティ&セキュリティ分科会主査として技術者と研究活動

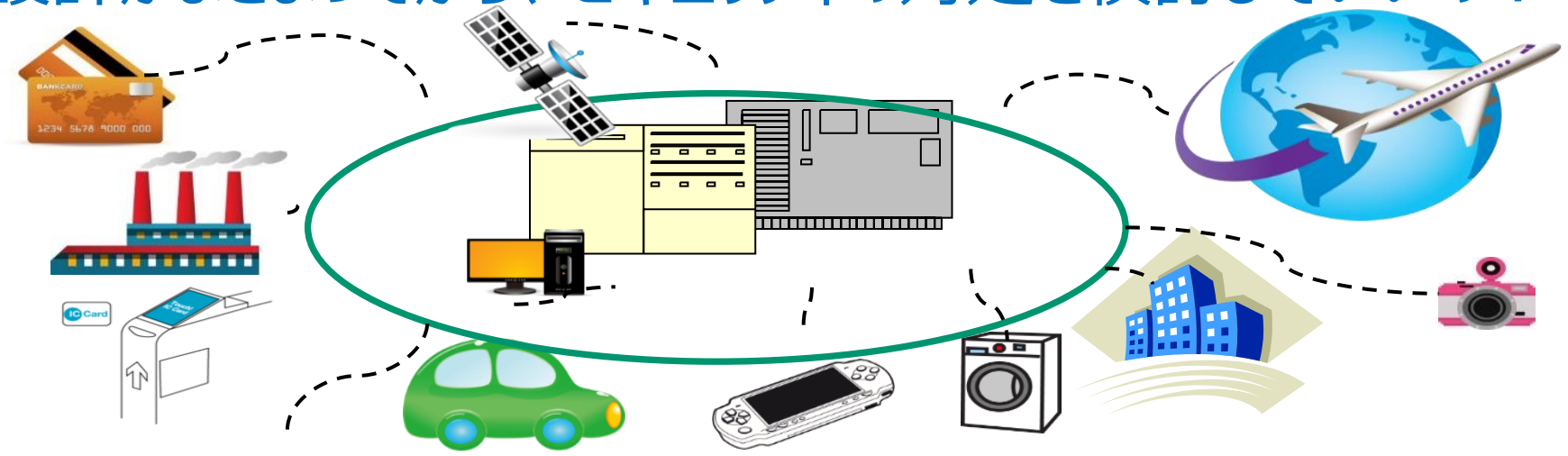
2019年−国立情報学研究所で機械学習システムのセーフティ・セキュリティを研究
＊「Safety＆Security～IT博士と学ぼう！デジタル社会の歩き方」月間誌「潮」 新年号より連載開始
*来春、日科技連出版社より、セーフティ＆セキュリティ入門書　発刊予定

Ⅰ　IoTとセキュリティ・バイ・デザイン

IoTの特徴と開発の課題

IoT：モノのインターネット（Internet of Things）
IoTの特徴：多様な機器・システムがつながること
IoT開発→より複雑なセキュリティ要件をもつことになる。
セキュリティが不安。

設計がまとまってから、セキュリティの対処を検討していいの？



IoTセキュリティの現状

IoTシステムへの脅威事例は日増しに増加

- HDDレコーダーの踏み台化は情報家電に対する初期の攻撃事例（2004年）
- 心臓ペースメーカの不正操作（2013年）
- Blackhat2015でCharlie Miller氏とChris Valasek氏がジープのチェロスキーの遠隔操作法を発表攻撃者は数マイル離れた自宅からハンドル操作ブレーキの無効化（140万台のリコールに）https://blog.kaspersky.co.jp/blackhat-jeep-cherokee-hack-explained/8480/
- MiraiによるDDoS攻撃（2017年）
- 海外から調達した通信機器
- （中国の大手通信機器メーカー
- 「ファーウェイHUAWEI」のルーター）が
- アメリカで深夜に勝手に動き本国にデータ送信の疑惑

IoTセキュリティの課題

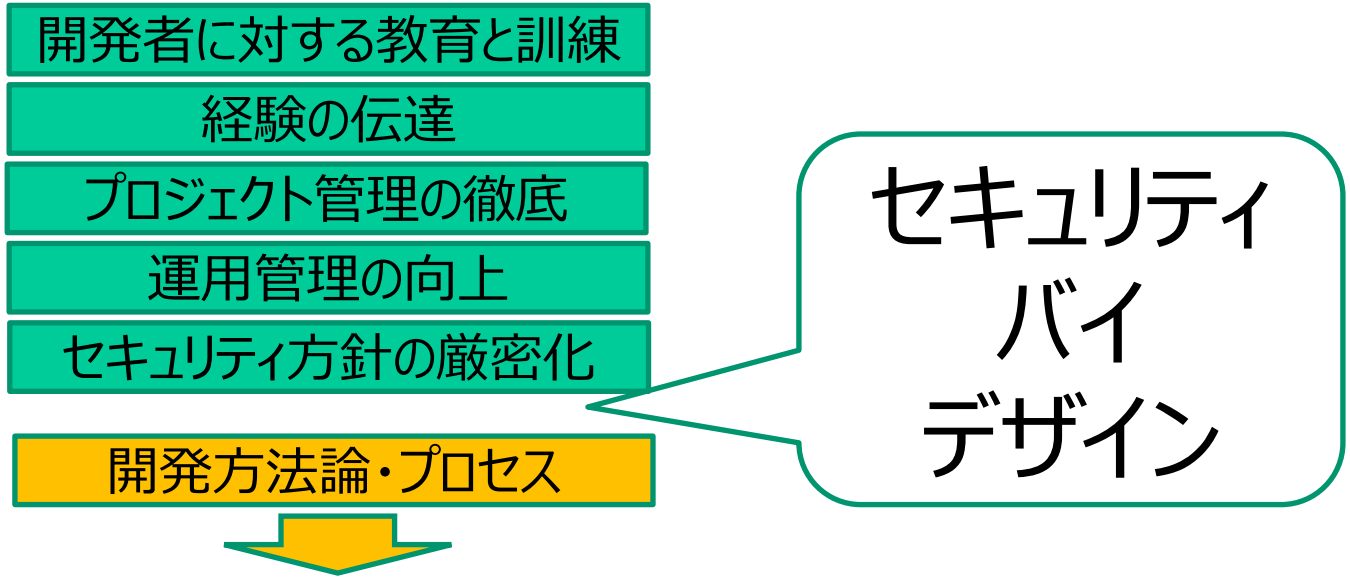
- 対象が情報だけでなく，実体を伴うモノになるため．与える被害も致命的であり，攻撃の被害は甚大にならざるを得ない→IoTセキュリティの確保は重大事
- IoTセキュリティの対象となる機器やシステムに対する脅威には盗聴や不正アクセスによる情報漏えいやプライバシー侵害，データやソフトウェア改ざんによる誤動作や予期せぬ停止
- 業界，製品・システムごとに要件が異なるため，セキュリティの対応レベルが異なり，標準化の動向も異なっている
- IoTのセキュリティを確保するための技術や手法，標準，基準はまだ確立されていない

脅威となる対象の洗い出し・目標を設定するセキュリティ要求分析手法・リスク評価の手法，要件を可視化する技術が必要）

安全なシステム・ソフトウェアの開発

より巧妙化する脅威に対して，より安全なソフトウェアを開発するにはどうしたらよいか？

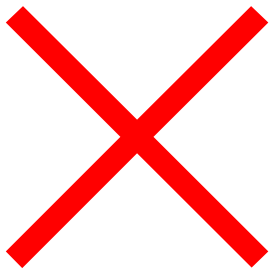
解決方法



システム・ソフトウェアの中に脅威への対抗手段を含めることがより根本的な対策になりうるから

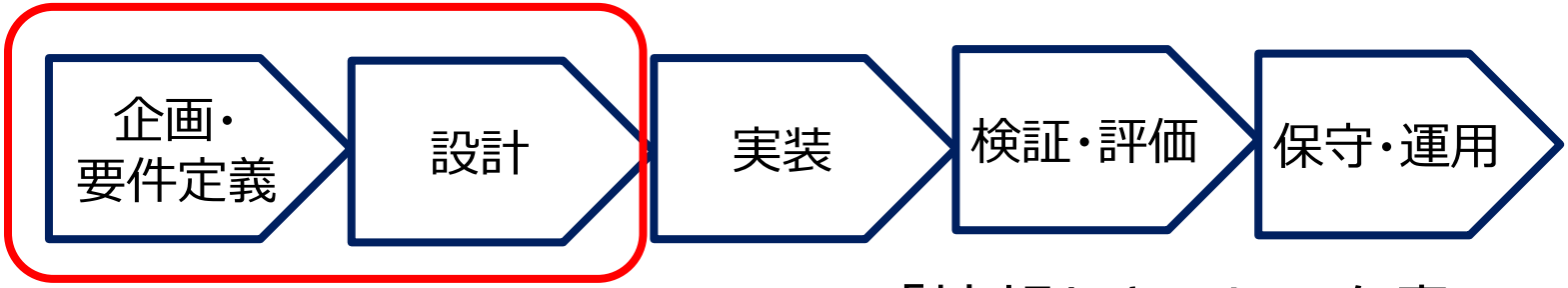
クイズ１

セキュリティ・バイ・デザインとは、ソフトウェア設計によってセキュリティを実現することである。

答えは 

セキュリティ・バイ・デザインの定義(NISC)

「情報セキュリティを企画・設計段階から確保するための方策」



＊「情報セキュリティ白書2019」p56

リスク分析を上流工程から行うこと
→脅威分析・セキュリティ要求分析

安全なIoT システムのためのセキュリティに関する一般的枠組

平成28年8月26日
内閣サイバーセキュリティセンター

1.目的

IoT(Internet of Things)システムについては、モノが接続されることから、IT と物理的システムが融合したシステムとして捉える必要があり、同システムが提供するサービスには、従来の情報セキュリティの確保に加え、新たに安全確保が重要となる。また、将来、個々のシステムが相互に接続されることを見据え、システム相互間の接続が新たな脆弱性となる懸念があることを踏まえ、セキュリティ・バイ・デザイン(Security by Design)の思想で設計、構築、運用されることが不可欠

3. 基本原則

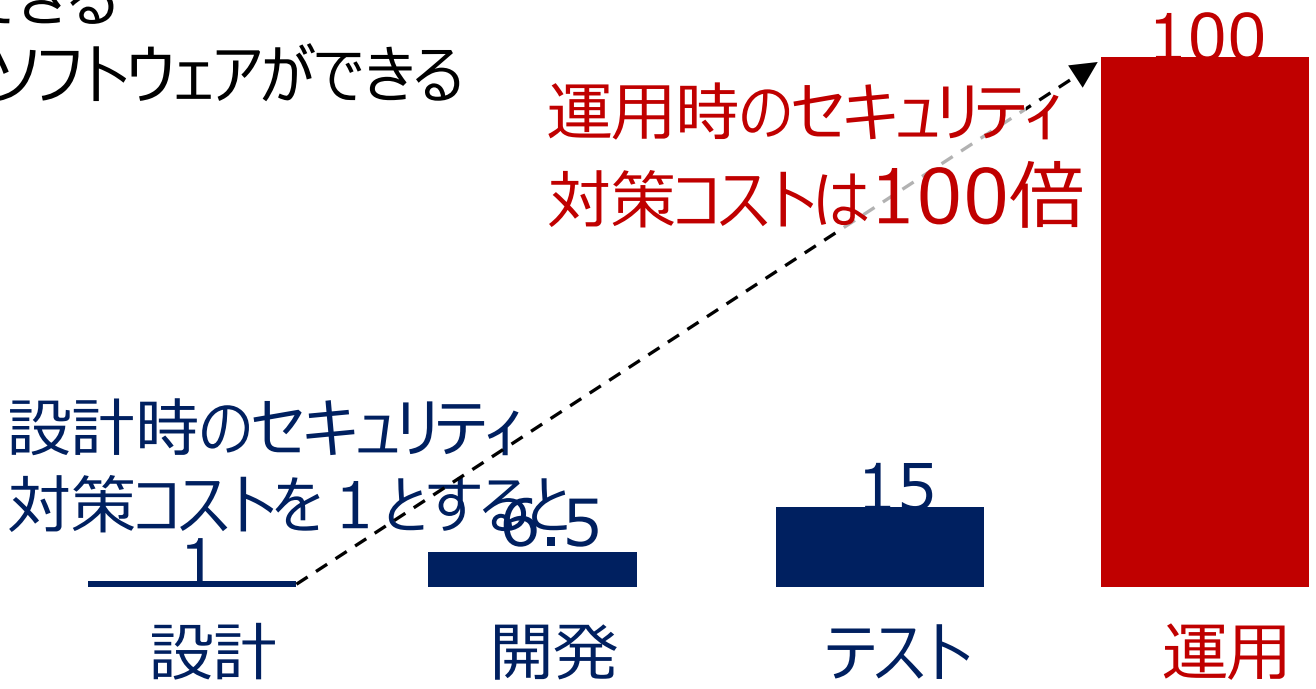
IoT システムの設計・構築・運用に際しては、セキュリティを事前に考慮するセキュリティ・バイ・デザインを基本原則とし、これが確保されていることが当該システムの稼働前に確認・検証できる仕組みが求められる。

www.nisc.go.jp/active/kihon/pdf/iot_framework2016.pdf

セキュリティ・バイ・デザインのメリット①

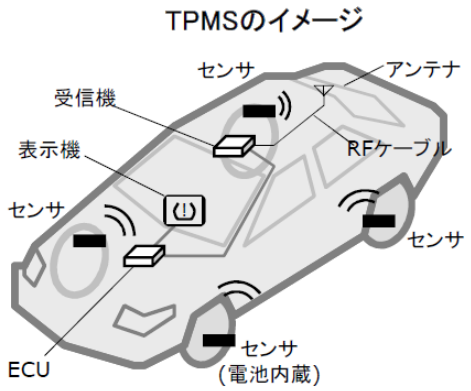
開発の早い段階から入れ込むので、

①手戻りが少なく納期を守れる
②コストも少なくできる
③保守性の良いソフトウェアができる



IPA資料「セキュリティ・バイ・デザイン入門」より

セキュリティ・バイ・デザインのメリット②



米国ではTREAD法によりタイヤ空気圧監視システムTPMS （Tire Pressure Monitoring Systems）の）装着が2007年から完全義務化。
・南カリフォルニア大とラトガーズ大は、TPMSのプロトコル解析に成功、近傍の車両の無線通信と区別するため、車輪毎に32bitの固有IDが割当てられていることを発見。
・車輪IDを読み出すことで、車両を追跡したり、誤った情報の送信により誤った警告灯の表示が可能。
攻撃ツールの開発に要した原価は$1,500程度。

セーフティの観点から設計すると車輪毎の固有ID付与は妥当だが、
セキュリティの観点からは車両の追跡や誤った警告灯の表示という脅威を生む。

CCDS資料より引用

問題の解決には、統合的観点でのセキュリティ・バイ・デザインが必要

設計思想の転換

Ⅱ．IoT高頼化機能要件

開発者向け 「つながる世界の開発指針」の実践に向けた手引き

- **開発指針のうち技術面での対策が必要となる部分をさらに具体化**
- **2017年5月８日公開：以下のURLにpdf版掲載**

  http://www.ipa.go.jp/sec/reports/20170508.html

つながる世界の開発指針

2016年3月

「つながる世界の開発指針」の実践に向けた手引き

2017年5月

①設計段階から考慮して欲しい要件とIoT高信頼化機能の具体例を解説

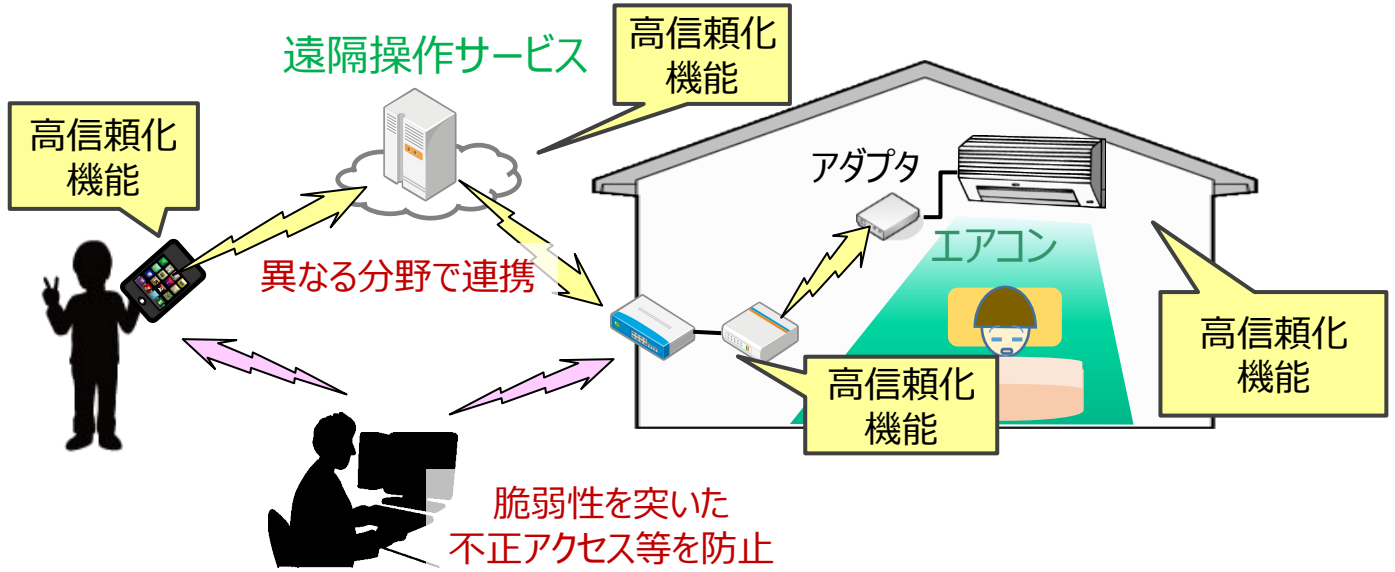
②IoT機器・システムやサービスのライフサイクルとクラウド・フォグ・エッジ等の機能配置を考慮し網羅的にイメージ

③IoTの分野間連携のユースケースと、リスクや脅威、機能定義や機能配置の具体例

つながる世界の開発指針～ＩｏＴ高信頼化機能

- IoT高信頼化機能とは
  IoT機器・システムが相互に連携する（つながる）環境において、安全安心を確保するための機能

- 「実践に向けた手引き」における用語
- IoT高信頼化機能は、様々なIoT機器・システムでの利用を想定

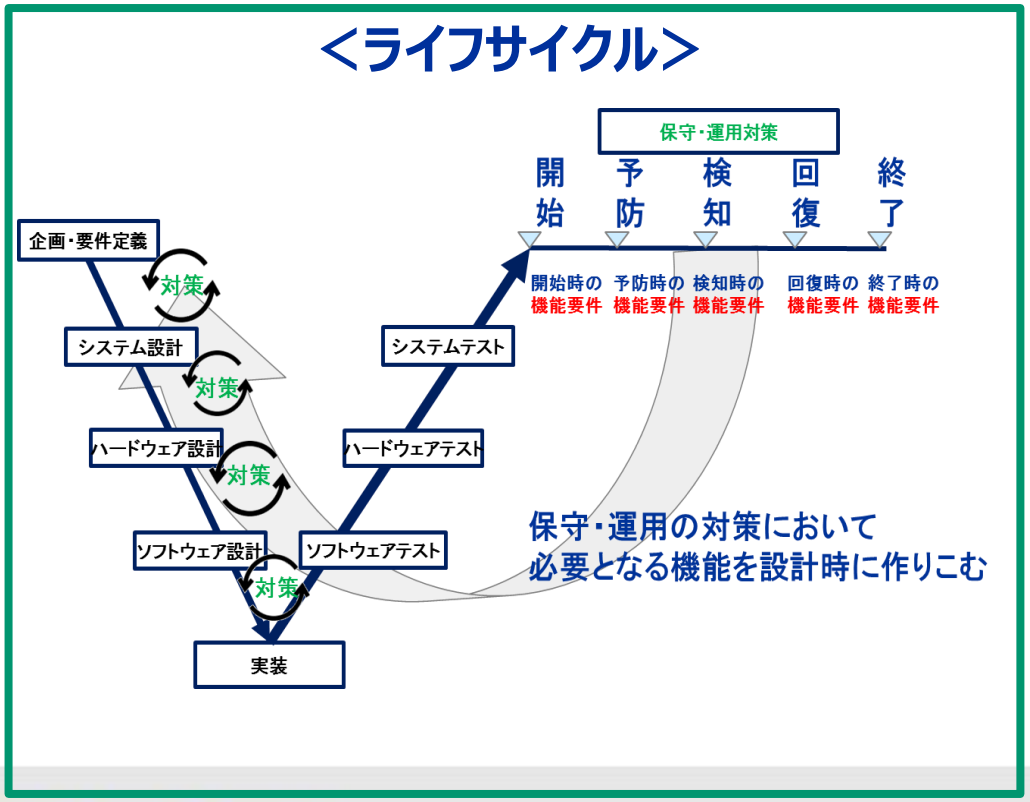


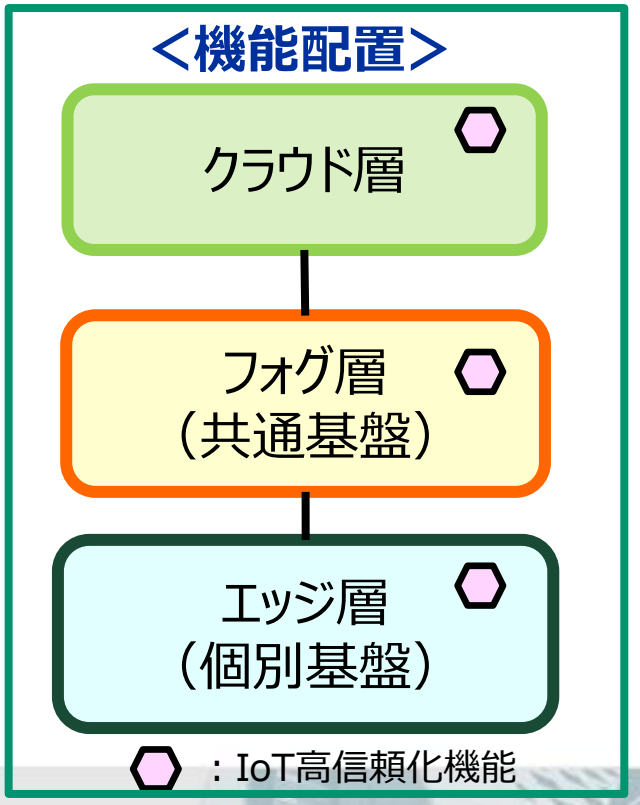
＊「つながる世界の開発指針」の実践に向けた手引き

ライフサイクルと実装位置でカバー

- IoT機器・システムやサービスのライフサイクルを考慮し、保守・運用の視点で開始と終了の間を予防・検知・回復にわけて整理
- クラウド・フォグ・エッジ等の機能配置を考慮

→ **網羅的にイメージした検討や、経済合理性や寿命を考慮した現実的な検討を支援**





＊「つながる世界の開発指針」の実践に向けた手引き　より

設計段階から考慮してほしい要件

- 利用開始後の利用条件や環境の変化を見据えた設計が必要
- 保守運用における5つの視点「開始」「予防」「検知」「回復」「終了」で整理し、それをさらに12の機能要件に細分化

IoT高信頼化要件		IoT高信頼化を実現するための機能要件	対応IoT高信頼化機能
開始	導入時や利用開始時に安全安心が確認できる	初期設定が適切に行われ、その確認ができる	1、2
		サービスを利用する時に許可されていることを確認できる	3、4
予防	稼働中の異常発生を未然に防止できる	異常の予兆を把握できる	5、6、7、8、9
		守るべき機能・資産を保護できる	4、5、6、10
		異常発生に備えて事前に対処できる	11
検知	稼働中の異常発生を早期に検知できる	異常発生を監視・通知できる	12、13
		異常の原因を特定するためのログが取得できる	5、6
回復	異常が発生しても稼働の維持や早期の復旧ができる	構成の把握ができる	14
		異常が発生しても稼働の維持ができる	8、15、16、17
		異常から早期復旧ができる	11、18、19、20
終了	利用の終了やシステム・サービス終了後も安全安心が確保できる	自律的な終了や一時的な利用禁止ができる	18、21、22
		データ消去ができる	23

＊「つながる世界の開発指針」の実践に向けた手引き　より

記載例

［要件3］稼働中の異常発生を早期に検知できる

(1) 概説

IoT システムは、一般に多数のセンサーなどの IoT 機器で構成されることが多く、かつ、他の IoT システムと連携してサービスが提供され、複雑な構成となることがある。これらの複雑化した構成の中で、一部の IoT 機器の障害/故障やセキュリティ異常が連携システム全体に影響を及ぼすことが想定されるので、異常の早期発見と被疑箇所の特定が重要となる。そのためには、普段からのシステムの監視や異常の原因を切り分けるための動作ログを収集することが必要である。

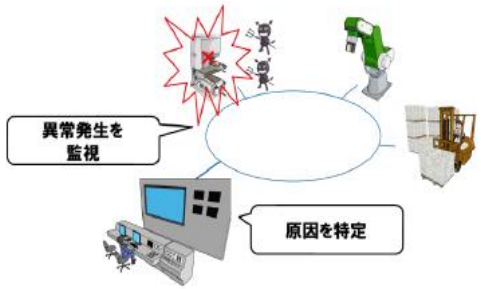


図 3-3 監視と原因特定

(2) 要求される機能

【機能要件6】異常発生を監視・通知できる

IoT 機器・システムの監視では、システムの各構成要素が保有する障害/故障やセキュリティ異常などの監視機能の能力を見極めて、システム全体としてもれなく監視が行き届くような設計が必要である。また、監視対象を明らかにして、その状態を逐次確認することが可能な状態可視化機能が必要である。なお、IoT の監視では、IoT 機器・システムの障害/故障やセキュリティ異常だけでなく、IoT 機器・システムなどに対する制御の競合が発生していないことを監視できる機能も必要である。制御の競合の検知については後述する。

【IoT 高信頼化機能】監視機能[12]、状態可視化機能[13]

IoT高信頼化要件

IoTとして実装上考慮すべき点

(3) 実装上の考慮事項

① **個々での異常検知ができない場合の考慮**

例えば、多数のセンサーが接続されている場合に、個々のセンサーの障害を監視することが困難な場合がある。そのような場合に、複数のセンサーからあげられてくるセンシング値を比較し、「外れ値（統計において他の値から大きく外れた値）」などにより、センサーの異常の推測を行うことが考えられる。このように、個々の IoT 機器での異常検知ができない場合に、異常情報以外の値や複数の情報を用いて、異常を推測することが考えられる。

【IoT 高信頼化機能】監視機能[12]

② **競合への考慮**

IoT では、各種サービスが複雑に連携するケースが想定され、一つの IoT 機器・システムが同時に複数のサービスから相反する指示を受けることがある。例えば、住宅の中で、快適サービスからは、暑くなってきたので窓を開ける指示と、一方、防犯サービスでは、窓を閉める指示が出されるなど、制御が競合する場合がある。競合の状況が検知できることが重要であり、競合状況としては、互いに正常な指示の競合だけでなく、一方が不正な指示による競合も想定が必要となる。

要求される機能

IoT高信頼化を実現するための機能

＊「つながる世界の開発指針」の実践に向けた手引き

２３のIoT高信頼化機能

- 初期設定や認証など、具体的な機能を紹介

IoT高信頼化機能					
1	初期設定機能	9	ウイルス対策機能	17	冗長構成機能
2	設定情報確認機能	10	暗号化機能	18	停止機能
3	認証機能	11	リモートアップデート機能	19	復旧機能
4	アクセス制御機能	12	監視機能	20	障害情報管理機能
5	ログ収集機能	13	状態可視化機能	21	操作保護機能
6	時刻同期機能	14	構成情報管理機能	22	寿命管理機能
7	予兆機能	15	隔離機能	23	消去機能
8	診断機能	16	縮退機能		

＊「つながる世界の開発指針」の実践に向けた手引き　より

記載例

(9) ウイルス対策機能

目的	ウイルス感染の被害を防止する。
説明	ウイルス対策には、検出（侵入、実行、潜伏を含む）、駆除がある。検出には以下のような方式がある。 ・ ホワイトリスト方式 – 特にリソースの少ない IoT 機器の場合においては、登録されたソフトウェアのみ実行を許可することで、未知のウイルスの実行を防止する。 ・ ブラックリスト方式 – ウイルスチェックには、既知のウイルスをパターンファイルに登録し侵入、実行、潜伏を検出する。 ブラックリスト方式は、リソースが少ない場合には実装が難しいことが想定される。
参考	制御システム向けの端末防御技術「ホワイトリスト型ウイルス対策」とは？ http://monoist.atmarkit.co.jp/mn/articles/1404/07/news004.html

IoTについて考慮した事項

各機能の説明は簡潔にまとめ、詳細については参考情報を記載

(12) 監視機能

目的	機器・システムの異常を検知する。
説明	監視機能には以下のような機能がある。 ・ 異常の検知機能 – 障害/故障の検知（ログ分析含む） – セキュリティ異常の検知（ログ分析含む） – 制御の競合の検知　等 ・ 検知した異常の通知機能
参考	

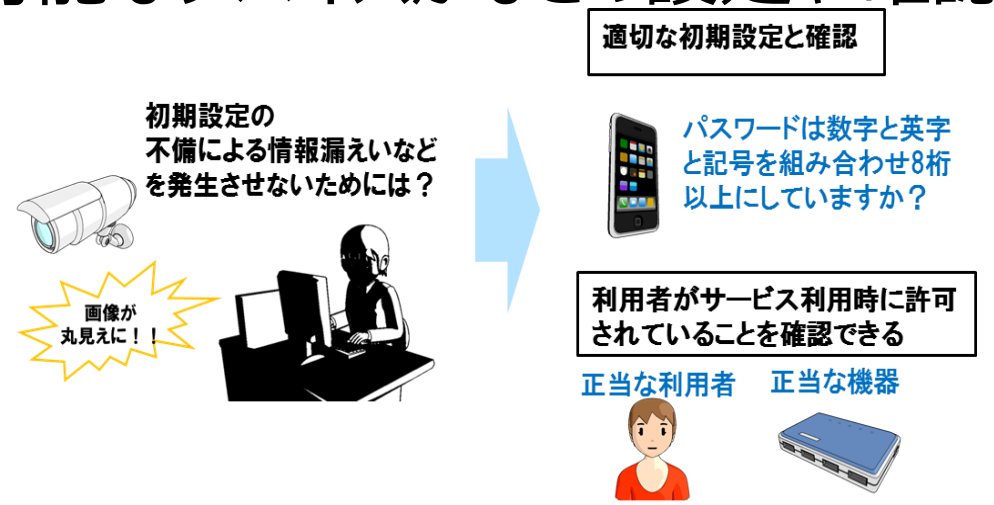
セキュリティだけではなく、セーフティやリライアビリティに関する事項も含む

一般論だけでなく、ユースケース分析から明らかになった事項も記載

＊「つながる世界の開発指針」の実践に向けた手引き　より

［開始］導入時や利用開始時に安全安心が確認できる

- 初期設定の不備をなくすとともに、許可された者か・アクセス可能なデバイスかなどの設定や確認を行う



	1 初期設定機能	2 設定情報確認機能	3 認証機能	4 アクセス制御機能
【機能要件1】初期設定が適切に行われ、その確認ができる	✓	✓		
安全安心に係る初期設定が適切に行われる				
初期設定が適切であることを確認できる				
【機能要件2】サービスを利用する時に許可されていることを確認できる			✓	✓
接続されるときに本人や正しい機器であるかを確認できる				
設定された情報にもとづき利用の許可/制限ができる				
相互の信用度を確認して接続の可否判断ができる				

＊「つながる世界の開発指針」の実践に向けた手引き　よ

IoT高信頼化機能
（３）認証機能

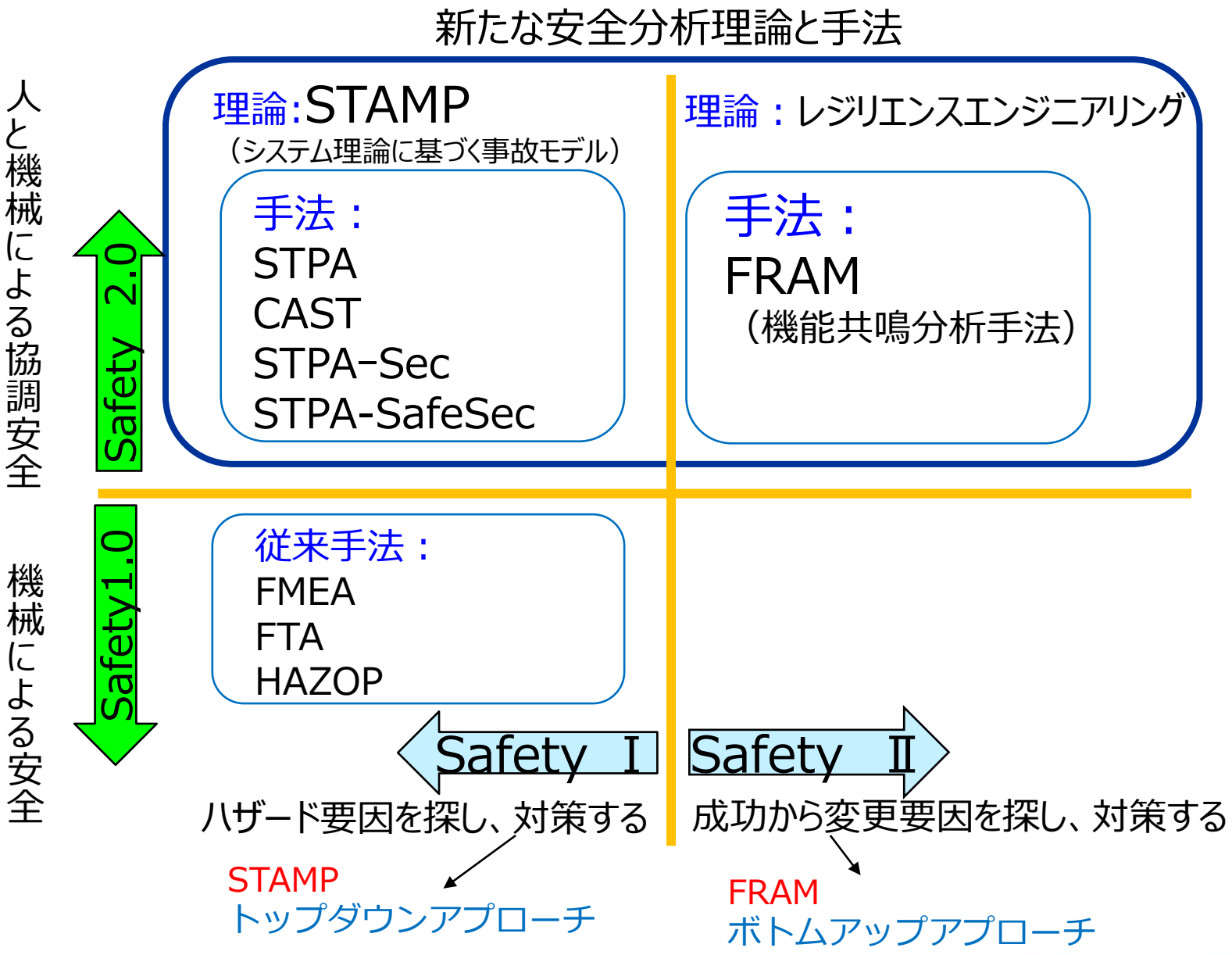
目的	利用者、機器などを一意に識別し、本人、あるいは、正しいものであるかどうかを確認する。
説明	認証方式には以下のような方式がある。 ・ 接続するIoT機器のなりすまし防止 – IoT機器の識別子による認証 – クライアント証明書による認証 – メッセージ認証 ・ 利用者のなりすまし防止 – ID・パスワードによる認証 – ICカードなどの所有物による認証 – 生体認証 ・ 接続する相手のシステム・サービスのなりすまし防止 – 接続する相手のシステム・サービス相互で鍵・電子証明書等を使用した認証 上記の、いくつかの方式を組み合わせた多要素認証などの方式もある。 また、一定回数以上の認証に失敗した場合にロックする機能などがある。
参考	・ [IoT推進コンソーシアム]IoTセキュリティガイドライン http://www.meti.go.jp/press/2016/07/20160705002/20160705002-1.pdf ・ [CSA]IoT早期導入者のためのセキュリティガイダンス https://www.cloudsecurityalliance.jp/newsite/wp-content/uploads/2016/02/Security_Guidance_for_Early_Adopters_of_the_Internet_of_Things_J_160224.pdf ・ [CRYPTREC]電子政府推奨暗号の利用方法に関するガイドブック http://www.cryptrec.go.jp/report/c07_guide_final.pdf

IoT機器が利用者を識別するときや、IoT機器がクラウドに接続するときや、IoT機器間での接続など様々なパターンを考慮しよう

＊「つながる世界の開発指針」の実践に向けた手引きより

Ⅲ. 安全性理論：
STAMPとレジリエンス・エンジニアリング

安全(セーフティ)理論・手法とSafety2.0・Safety Ⅱの関係



安全とSafety2.0

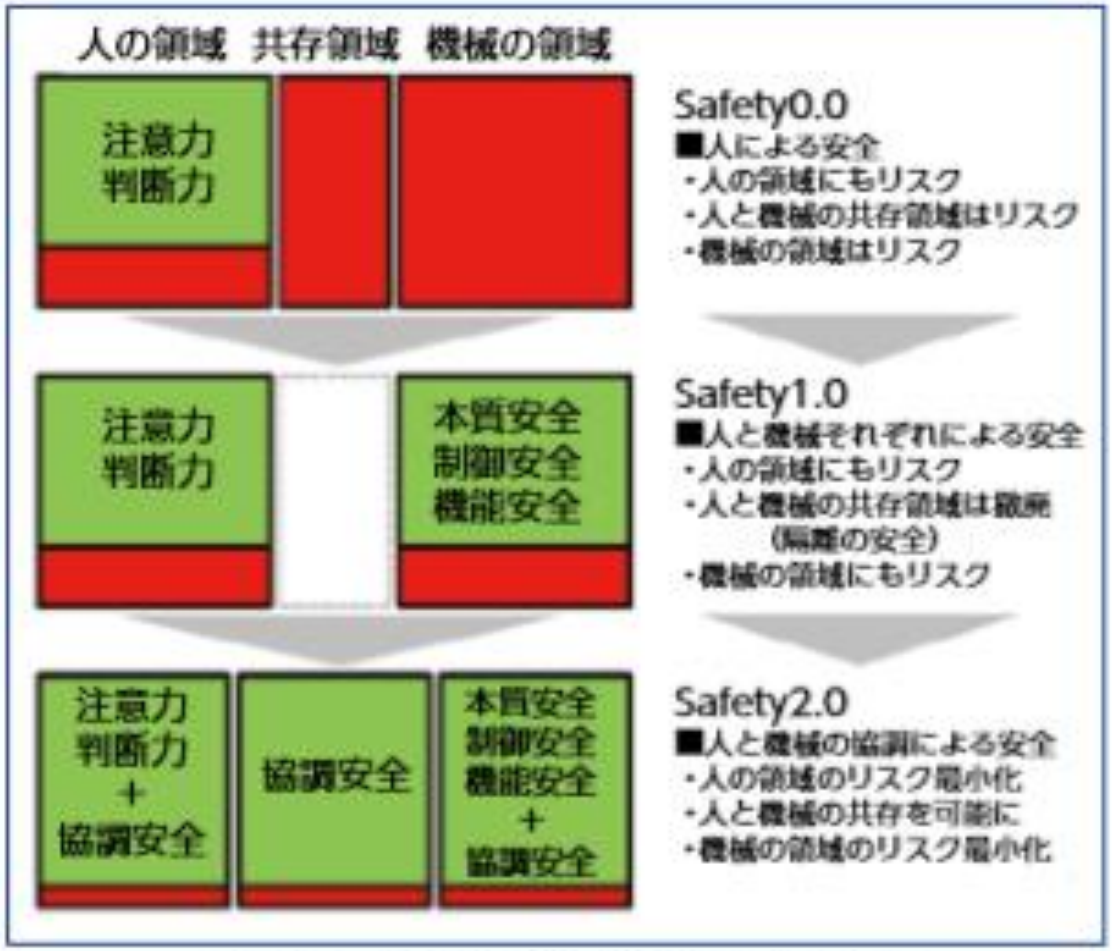


図 5.2-6 Safety0.0/1.0/2.0 の概念の比較

- 「本質制御」：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
- 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、 複雑システムの安全性向上技術とSafety2.0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

モデルとは何か？

モデル：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
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

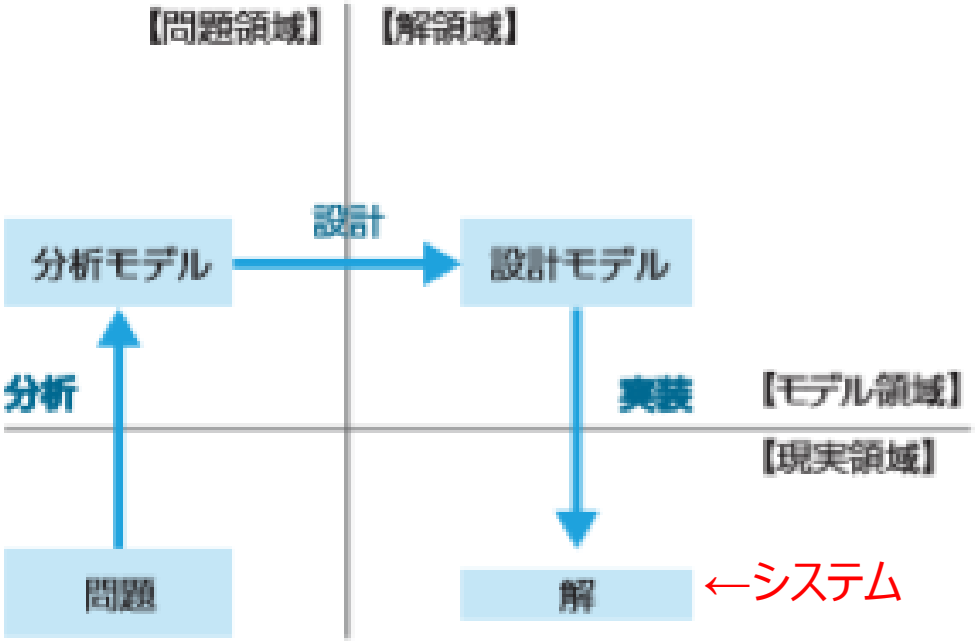
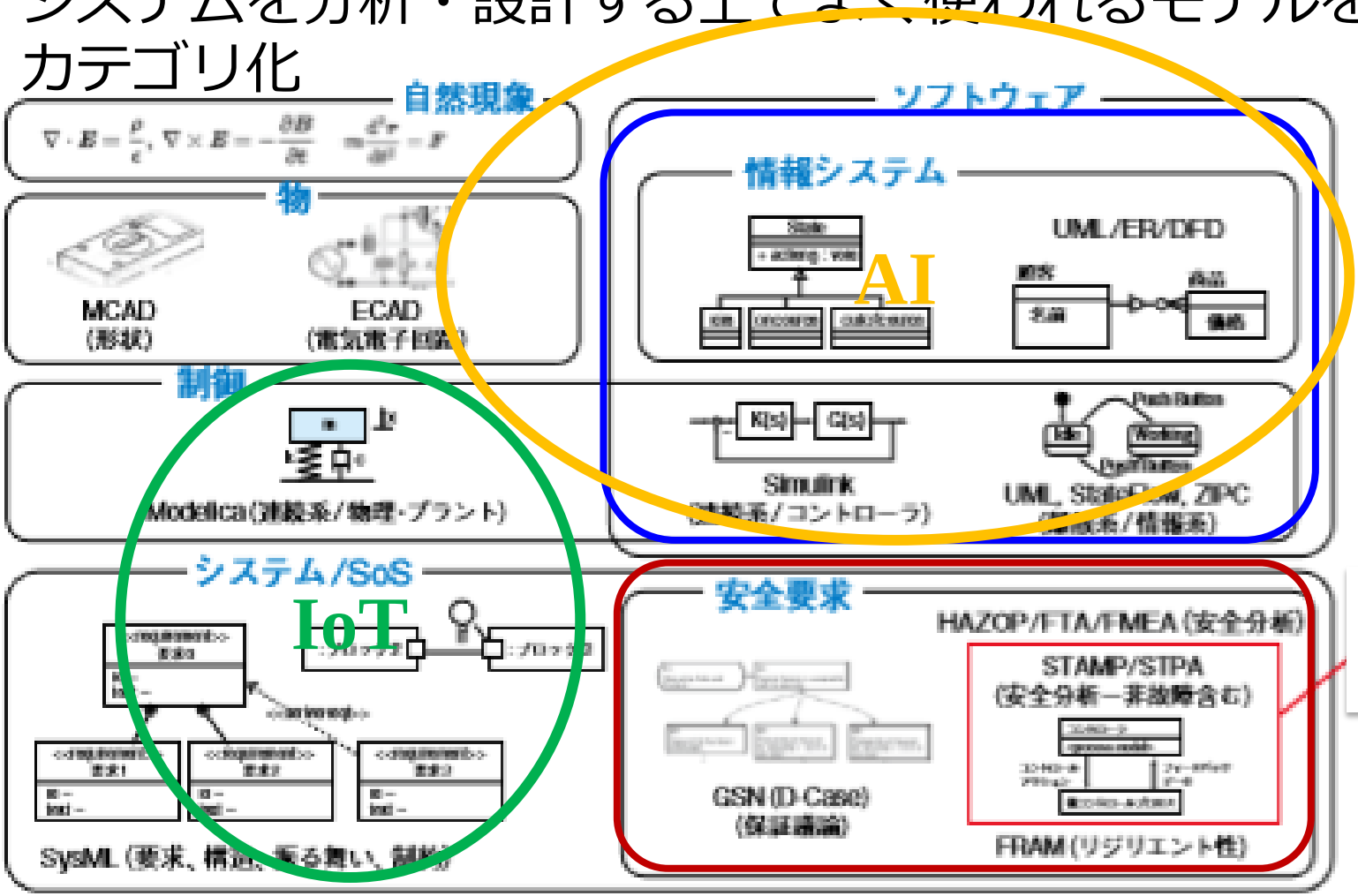


図 4.1-1　モデルの役割

IPA,はじめての
STAMP/STPA（活用編）
より

セーフティ・セキュリティとIoT/AI

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



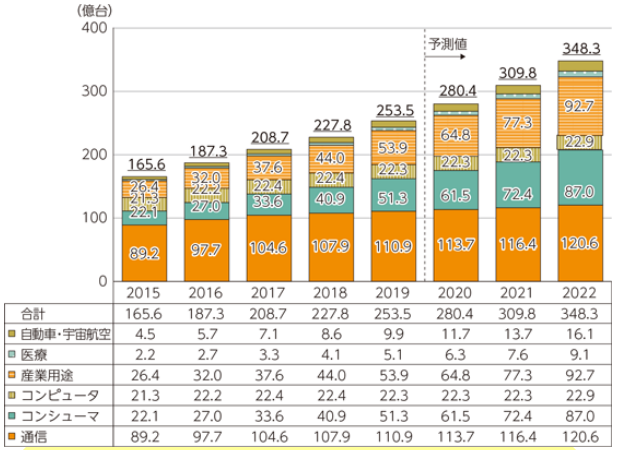
（サイバー）
セキュリティ
要求

セーフティ
要求

図 4.1-2　システムのモデリング言語

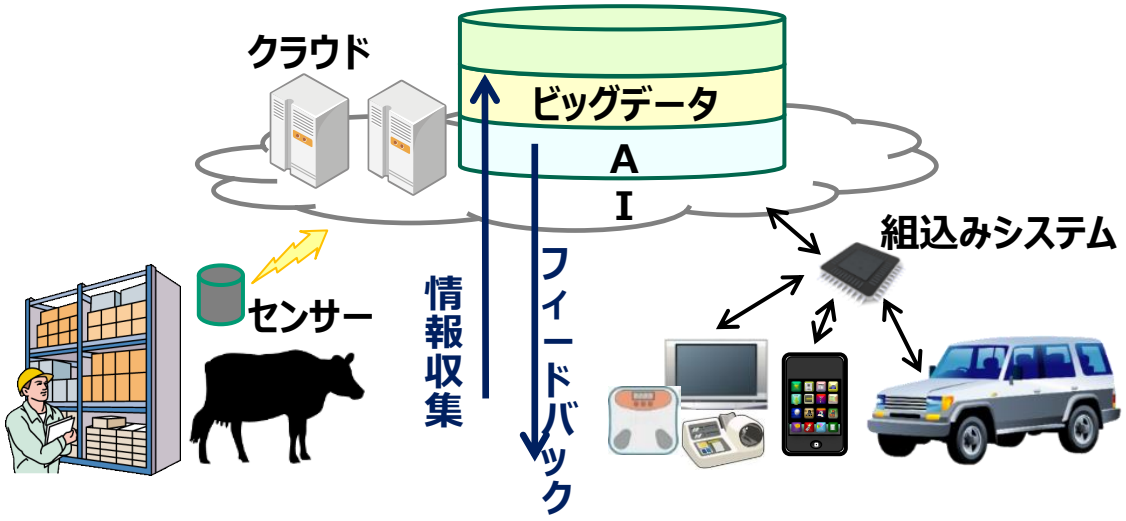
新しい安全解析手法の必要性

- 新たなシステムの基幹を担う要素が**ソフトウェア中心**に変化
- システム相互間の**コミュニケーションミス**によるシステム障害が増加
- 想定外の原因でのトラブルが無くならない



	2015	2016	2017	2018	2019	2020	2021	2022
合計	165.6	187.3	208.7	227.8	253.5	280.4	309.8	348.3
自動車・宇宙航空	4.5	5.7	7.1	8.6	9.9	11.7	13.7	16.1
医療	2.2	2.7	3.3	4.1	5.1	6.3	7.6	9.1
産業用途	26.4	32.0	37.6	44.0	53.9	64.8	77.3	92.7
コンピュータ	21.3	22.2	22.4	22.4	22.3	22.3	22.3	22.9
コンシューマ	22.1	27.0	33.6	40.9	51.3	61.5	72.4	87.0
通信	89.2	97.7	104.6	107.9	110.9	113.7	116.4	120.6

世界のＩｏＴデバイス数
〔総務省〕
２５億（２００９年）
⇩
254億（2019年）



＊IoTデバイス数は、「医療」、「産業用途」、「コンシューマ」及び「自動車・宇宙航空」で高成長が見込まれている

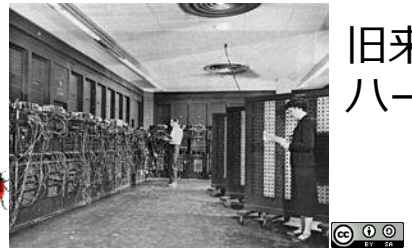
➢ 安全対策をしていても事故を防げない現状から、**複雑化したシステムに対応した新しい解析手法・事故モデル**が必要

1．STAMP出現の背景

現在の分析手法が確立されたのは40-65年前
コンピューターシステムはハードウェア主体からIoT時代へ

従来の手法
FMEA, FTA, HAZOPなど

アクシデントは構成機器の故障やオペレーションミスに起因すると仮定



旧来は
ハードウェア主体

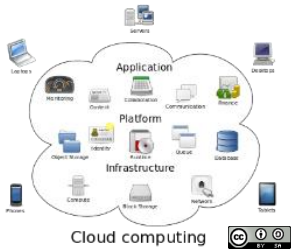
1940 • FMEA 1950 1960 • FTA • HAZOP • ETA 1970 • Bow Tie (CCA) FTA+ETA

既存のハザード分析手法は40～65年前のもの

パラダイムシフト

新たな手法
STAMPに基づく分析

アクシデントは構成要素間の相互作用から創発的に発生



1980 1990 2000 2010 • STAMP 2020

大規模・複雑化が進むIoT時代のコンピューターシステムへの対応が求められる

２．ソフトウェアの持つ特性(1/3)

（１）ソフトウェアは“故障”しない

◆ソフトウェアは物理事象を抽象化して設計

◆ソフトウェアは純粋に設計そのもの

汎用機器 ＋ ソフトウェア ＝ 特定目的機器

（２）ソフトウェアが主要な役割を果たすシステムの事故の殆どには誤った要求仕様が含まれている

ソフトウェアを修正しようとする試み

ソフトウェアを高信頼にしようとする試み

→ 殆どシステムの安全の向上に寄与しない

２．ソフトウェアの持つ特性(2/3)

（３）ソフトウェアはそれ自身どこまででも大規模・複合化・複雑化可能

全ての望ましくないシステムの挙動を防ぐ

全ての設計エラーをテストで駆除

もはや
無理

**事故
発生！！**

事故には2種類ある

コンポーネント故障事故

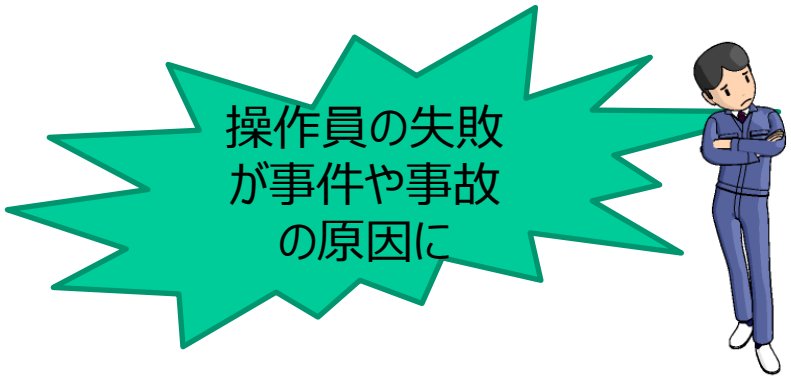
- 単一/複数コンポーネント故障
- 通常ランダム故障と見做す

コンポーネント相互作用事故

- コンポーネント間の相互作用から発生
- 相互作用とダイナミックな複雑さに関連

2．ソフトウェアの持つ特性(3/3)

（４）ソフトウェアはシステムにおける人間の役割を変化させる

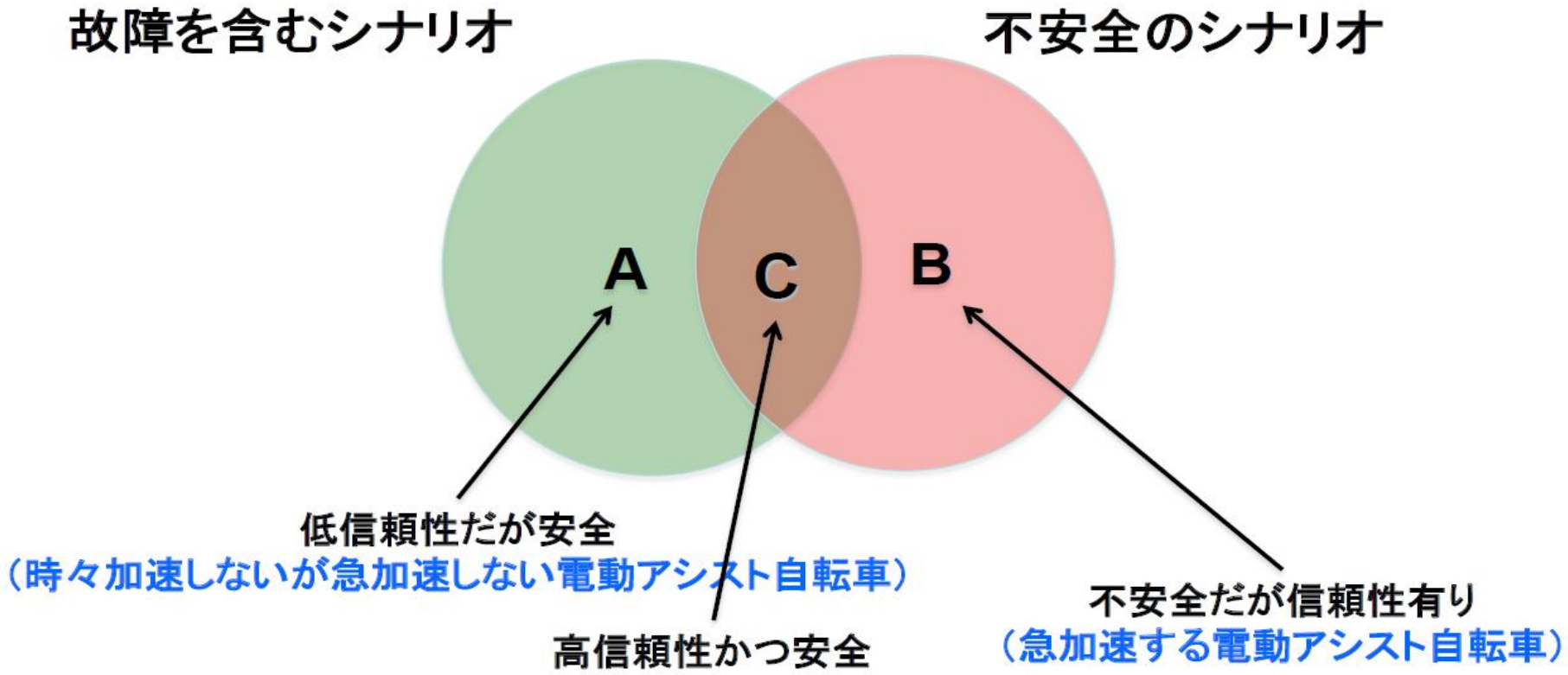


- 警告、再教育、解雇
- より自動化することで操作員の役割を限定
- 更に規則や手順を増やして操作員の作業を厳格化

システムの視点から見る操作員のエラー

- ◆ 操作員エラーは、兆候であり、原因ではない
- ◆ 全ての振る舞いがシステムとその状況から影響を受ける
  - –操作員が起こすエラーによってソフトウェア中心システムにおける操作員の役割が変化している
  - –操作員のエラーが避けられないあるいは設計者より操作員が事故の責任を問われる
- ◆ 操作員のエラーに関して何かするとしたら人々が働いているシステムをしっかり観察しなければならない
  - –設備の設計
  - –手順の有用性
  - –目的の不整合や操作へのプレッシャーの存在
- ◆ ヒューマンエラーは再設計すべきシステムの兆候である

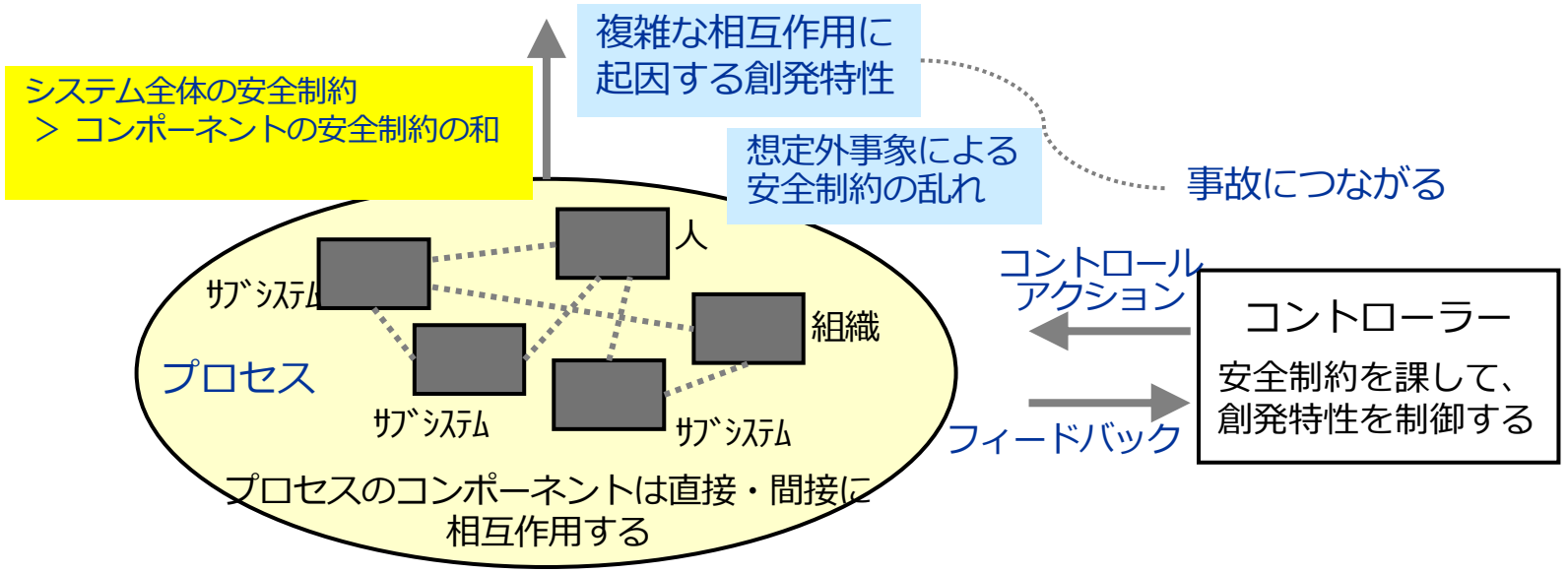
3．安全性と信頼性



コンポーネントや機能の故障を防ぐだけでは不十分

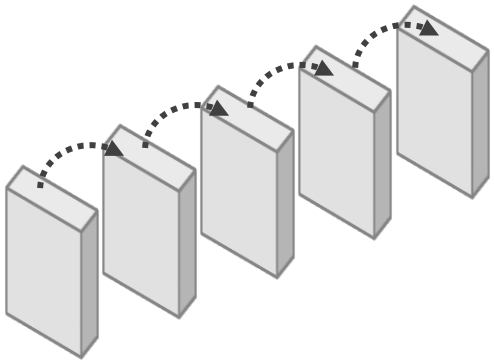
ＳＴＡＭＰ … 新しい事故モデル

- Ｓｙｓｔｅｍｓ-Ｔｈｅｏｒｅｔｉｃ Ａｃｃｉｄｅｎｔ Ｍｏｄｅｌ ａｎｄ Ｐｒｏｃｅｓｓｅｓ
  「システム理論に基づく事故モデル」 ＭＩＴの Ｎａｎｃｙ Ｌｅｖｅｓｏｎ 教授が提唱
  - 事故の要因となるのは：
    - コントロールストラクチャーやコントロールアクションが安全制約を実施できない
    - コントロールストラクチャーが徐々に劣化する
    - コントローラー間でコントロールアクションの調整が不十分である

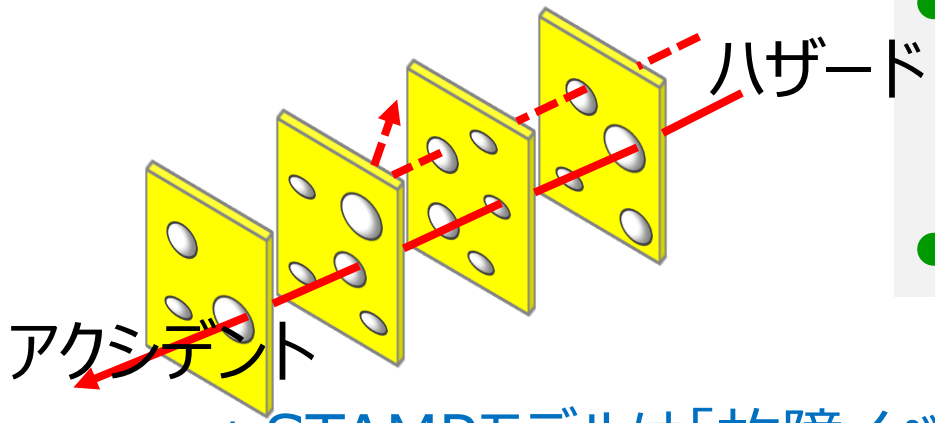


従来の事故モデル

■ ドミノモデル



■ スイスチーズモデル



ドミノモデルは

- 原因－結果（次の原因）－ …の系列をドミノ倒しにたとえる
  このドミノ倒しのどこかで手を打てば事故が避けられるとする
- 根本原因分析といわれる事故分析の各手法は、この考えに立っている

スイスチーズモデルは

- 防御壁とそこでの漏れをチーズの穴にたとえる
  穴が重なって見通せたときに事故となる
- 個々の穴をふさぐことで対策とする

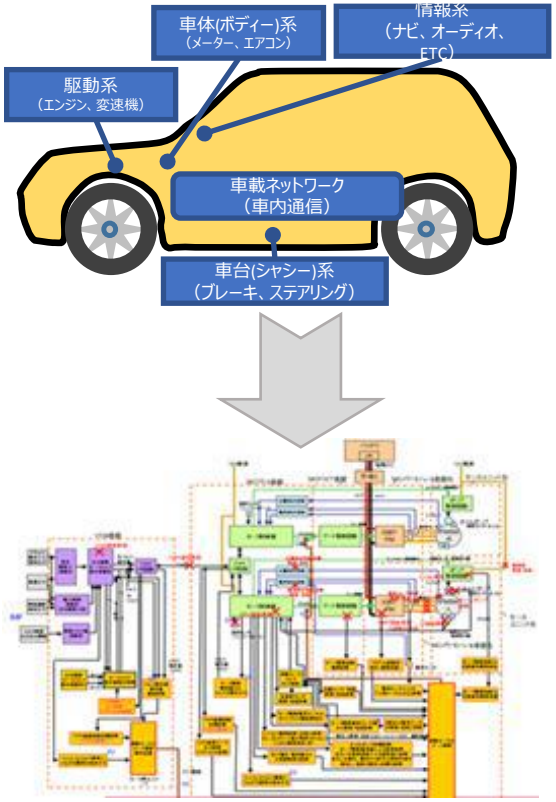
＊STAMPモデルは「故障イベントのチェーン」モデルに代わって登場

従来ＳＴＡＭＰの考え方

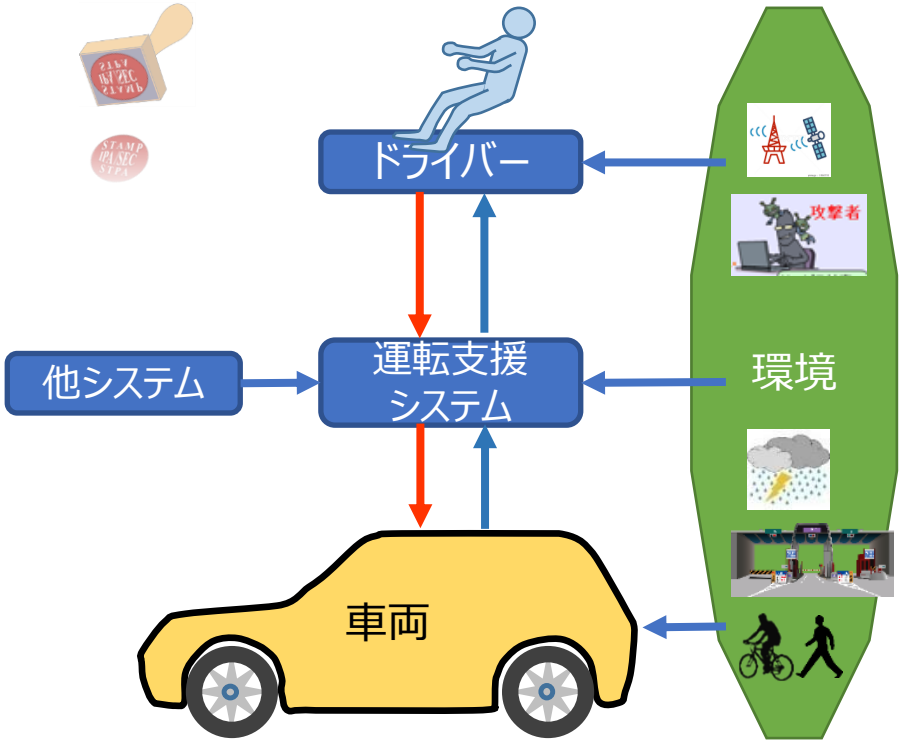
- 現在の複雑な組込みシステムは、ソフトウェアや人間・組織を含むサブシステムやコンポーネントで構成されており、コンポーネントに不具合がなくともサブシステムやコンポーネントの相互作用によってハザードが発生する。
- 従って、従来型のリスク分析手法では限界がある。

既存手法とSTAMPの考え方の違い

自動車システムを大規模な機能ブロック図等で表現し、システム中心視点の安全分析を行う

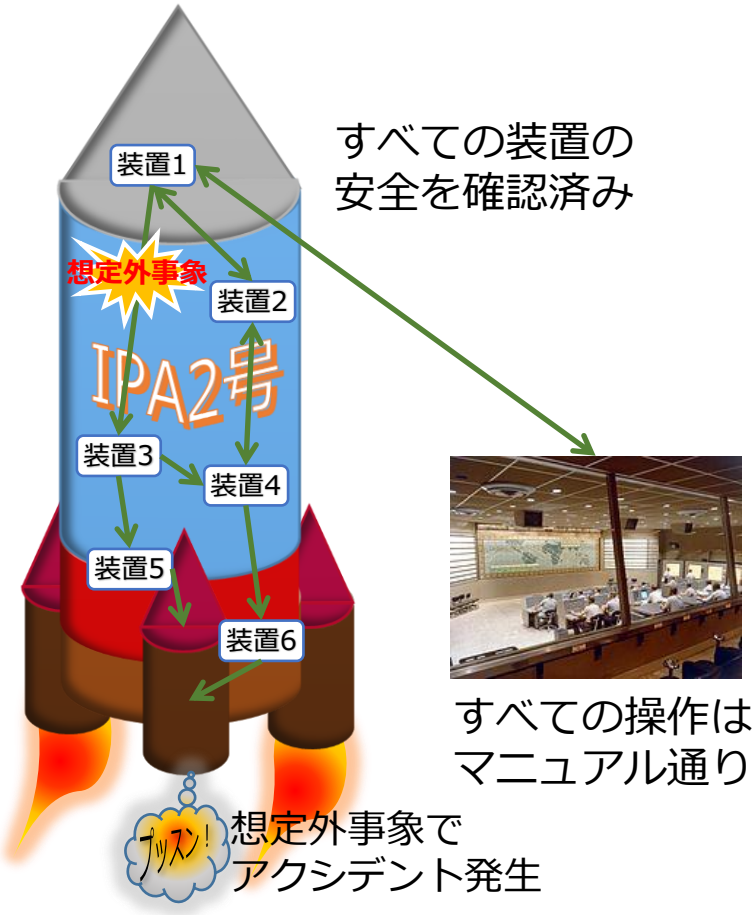


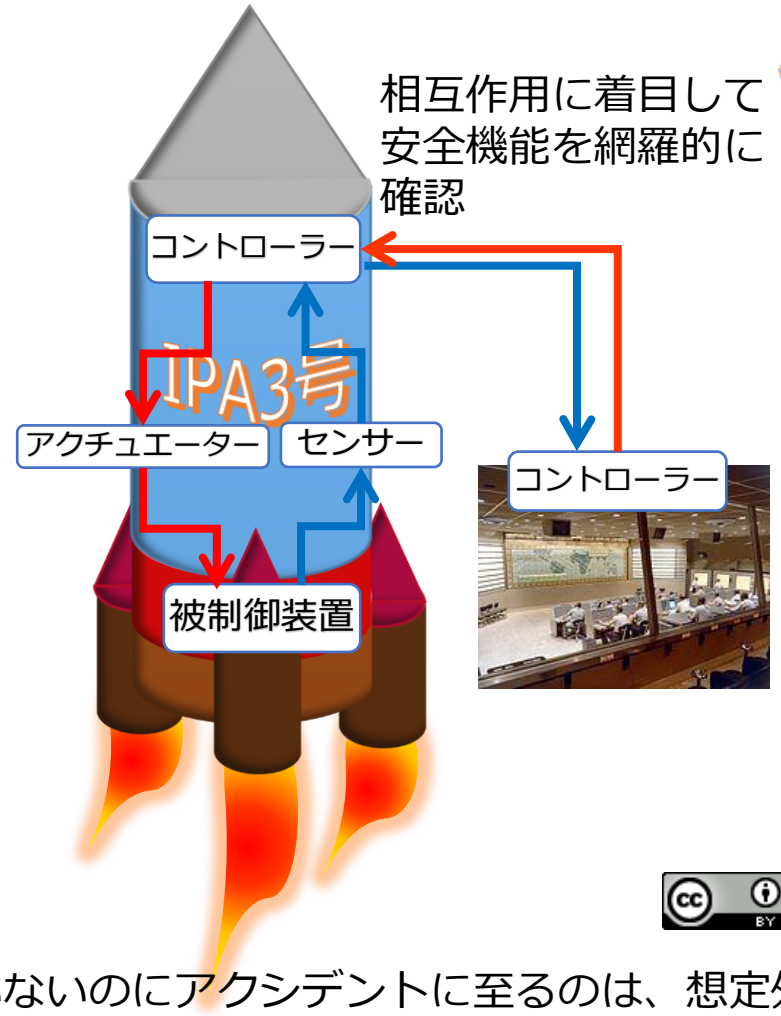
運転自動化が進むと更に大規模、複雑になり、人や環境、またそれらとの相互作用にも着目した安全分析が必要になる



STAMPのSTはSystems-Theoretic（システム理論に基づく）の略で、分析対象システムの安全に関係するすべてを含めて考えるべきと提唱しており、今後の複雑なシステムの安全分析に適している

STAMP/STPAの紹介　～宇宙航空編～





構成機器の故障に着目したら構成機器が故障していないのにアクシデントに至るのは、想定外事象

↓

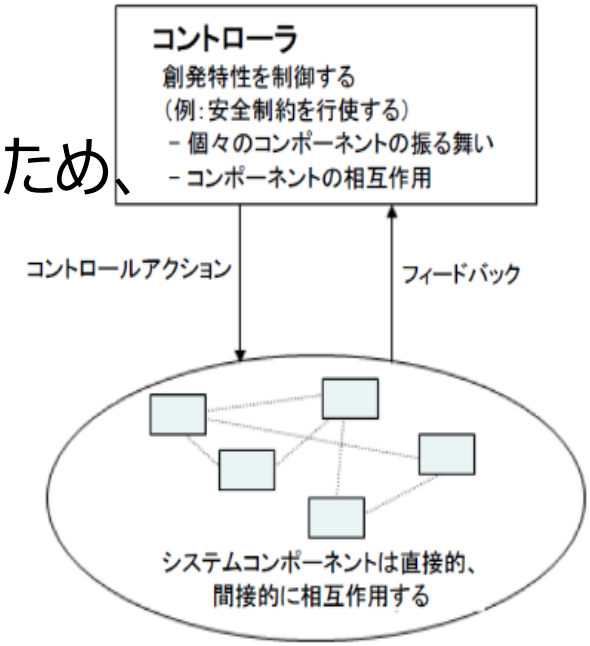
システムを安全に維持するための相互作用に着目して網羅的に確認することで想定外を削減

STAMP/STPA適用の方向性

創発特性とは、個々のコンポーネントの総和ではなく、コンポーネントが相互作用する時の「創発」する特性
創発特性は、システムの部品同士がどのように相互作用し調和するのかという関係によって生じる。

安全性だけでなく、セキュリティ、プライバシー、保守性や運用性なども創発特性である。

STAMPはいかなる創発特性にも適用できるため、
STPAは、サイバーセキュリティも含む、
いかなるシステム特性にも使用可能



有効性・優位性の検証はこれから行う必要がある

STAMP/STPA適用の可能性

・エンタープライズシステムの運用の分析
・開発プロジェクト（体制）のリスク分析にも適用可能

MITがボーイング社の運用フローの分析に適用して効果をあげた事例がある

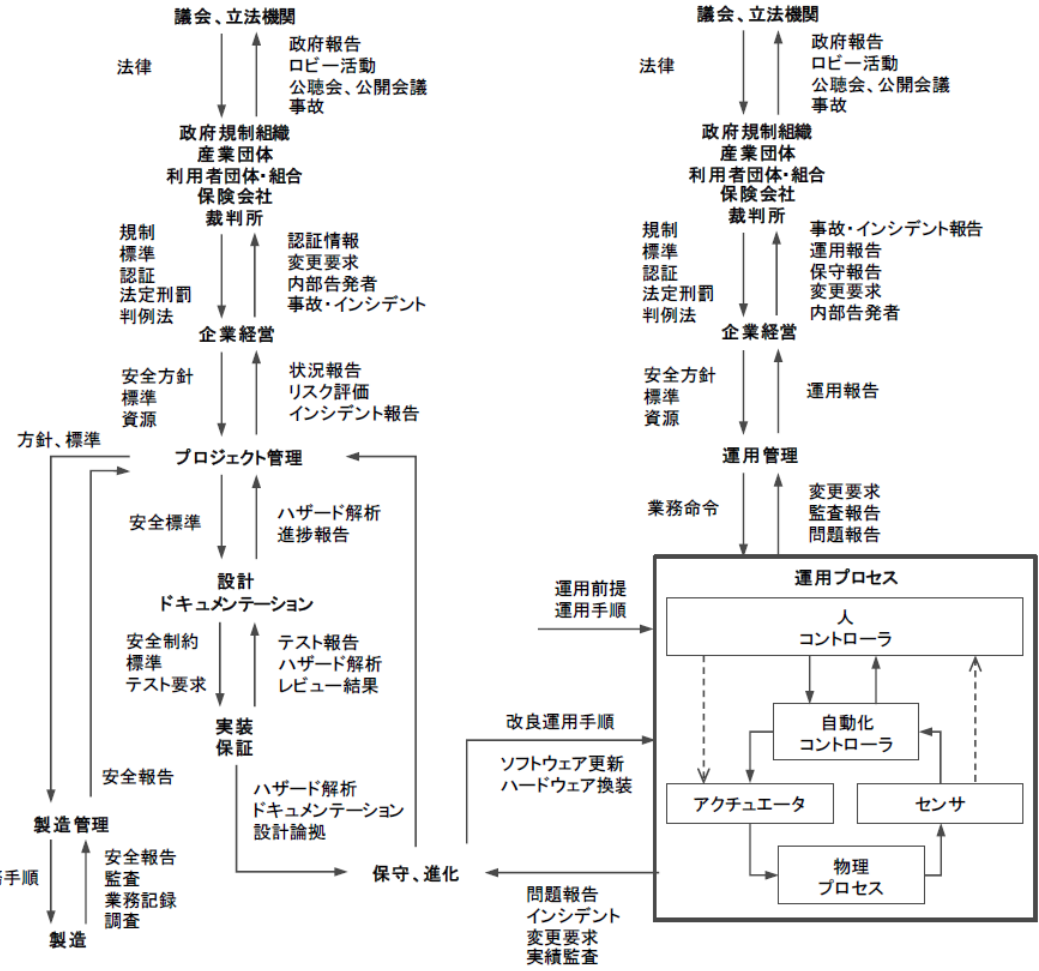


図 3.2:安全性に関するコントロールストラクチャーの例

STPAガイドブックより

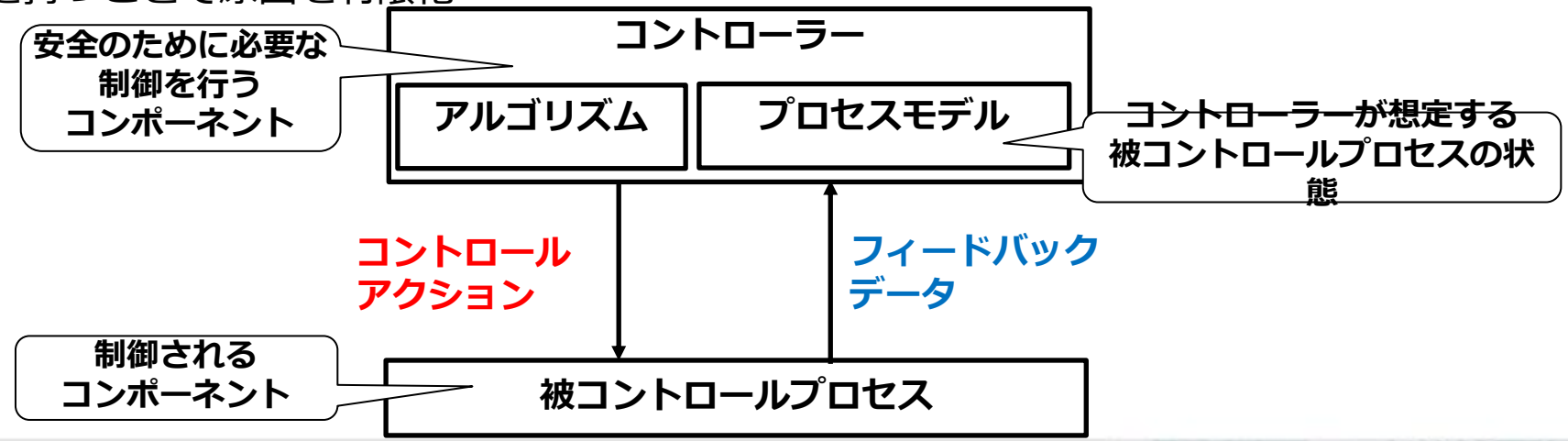
STAMPの概念

「STAMP：システム理論に基づく事故モデル」

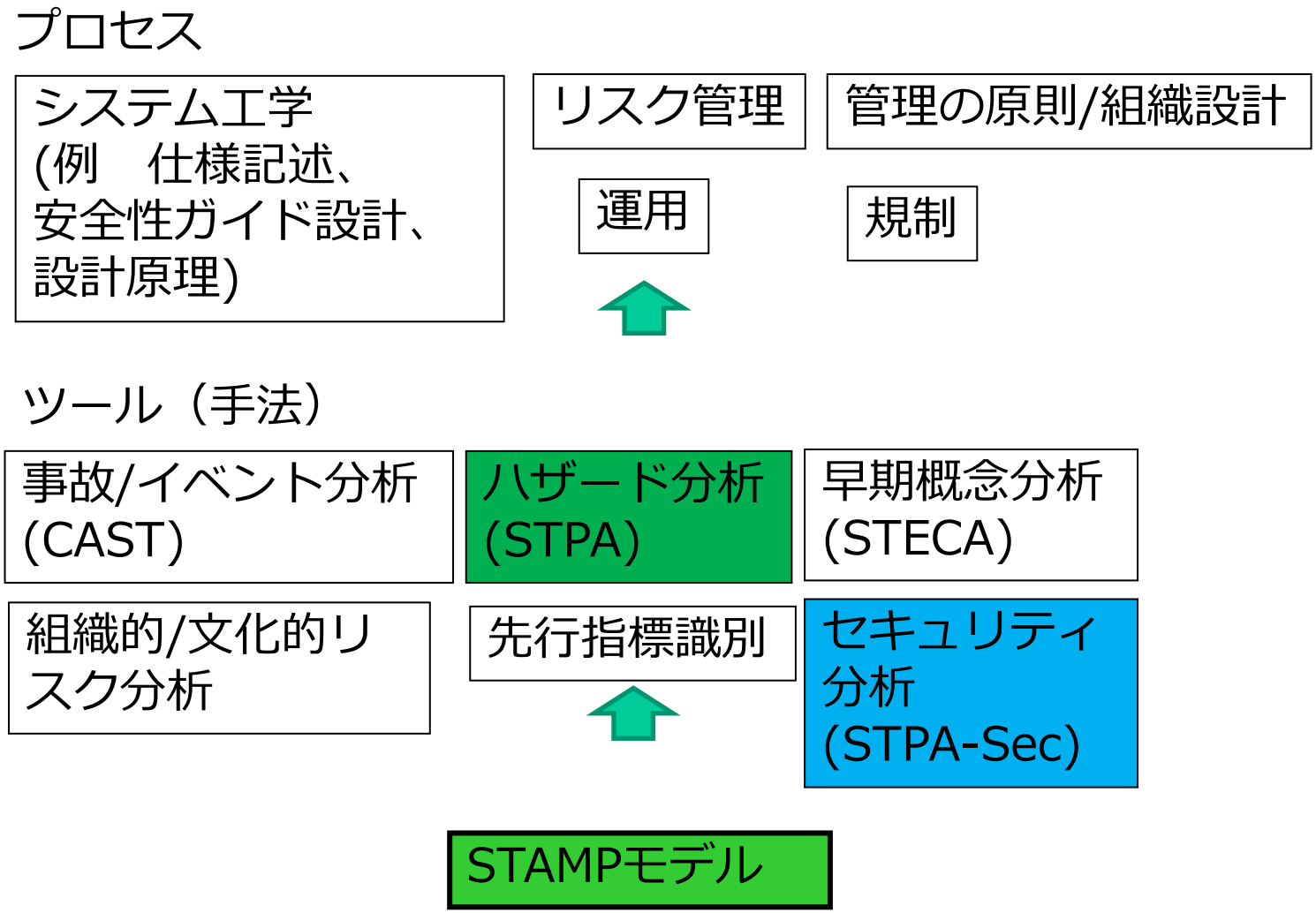
STAMP（**S**ystems-**T**heoretic **A**ccident **M**odel and **P**rocesses）：システム理論に基づく事故モデル

前提　システム事故の多くは、構成要素の故障ではなく、システムの中で安全のための制御を行う要素（制御要素と被制御要素）の相互作用が働かない事によって起きる

- 「要素（コンポーネント）」と「相互作用（コントロールアクション）」に着目してメカニズムを説明
- 「アクションが働かない原因」＝「コントロールアクションの不適切な作用」という視点を持つことで原因を有限化



STAMPに基づく分析の道具立てとプロセス



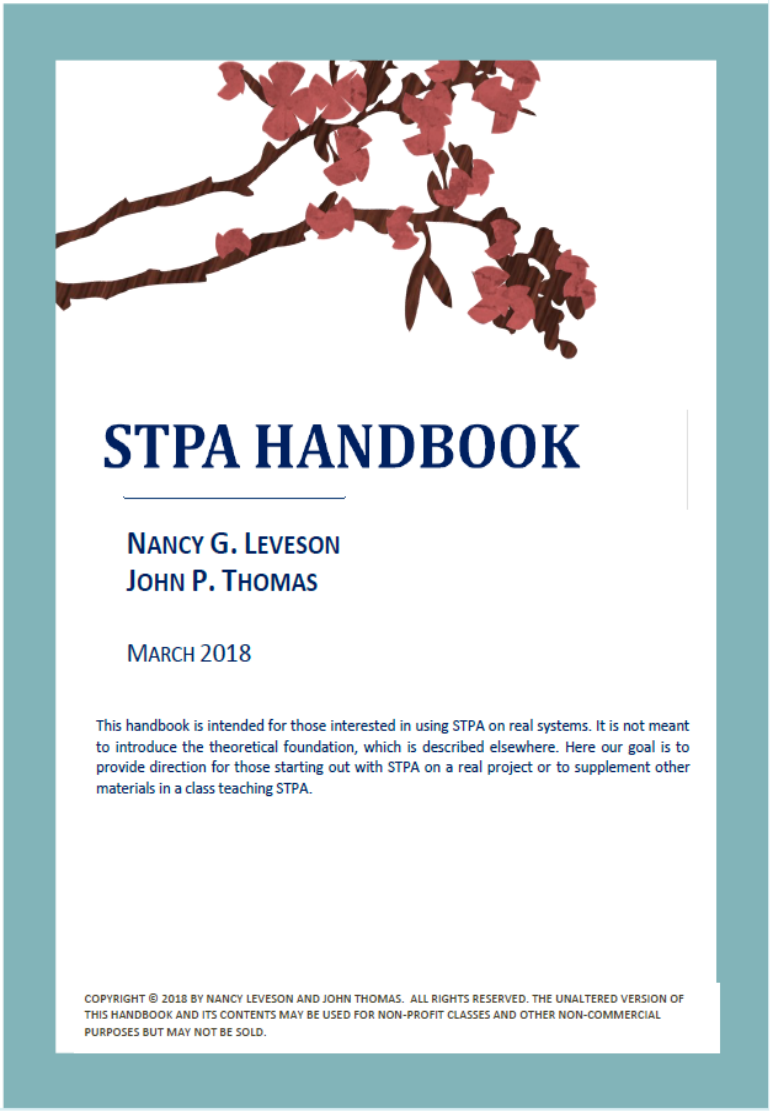
STAMP/STPAの紹介　～海外の普及状況～

欧米では宇宙、航空、鉄道など大規模インフラの安全設計や事故分析へのSTAMP活用が普及しつつあるが、日本では未だ認知度は十分ではなく、取り組みが遅れている

- STAMPがハザード分析に使われた重要システムの例
  - 宇宙開発（例：HTV　こうのとり）
  - 軍事分野（例：無人航空機）
  - 原子力発電所

- 試験的導入が進む世界の公的機関、民間企業の例
  - 米国FDA（例：医薬品リコール）
  - 米国FAA（例：次世代航空交通システム）
  - 米空軍（例：飛行機運用手順）
  - 航空産業（例：ボーイング、キャセイパシフィック、ブラジルの航空管制）
  - 自動車産業（例：欧米の自動運転、ステアリング制御、ブレーキ制御、ディーゼルエンジン）
  - ロシアの巨大パイプラインプロジェクト
  - 鉄道（例：中国の高速鉄道事故分析）
  - 自動車機能安全規格（SOTIF：Annexに提示）
  - 米国SAE　自動車技術会・機能安全委員会
  - ペガサスプロジェクト（ドイツの自動運転国家プロジェクト）

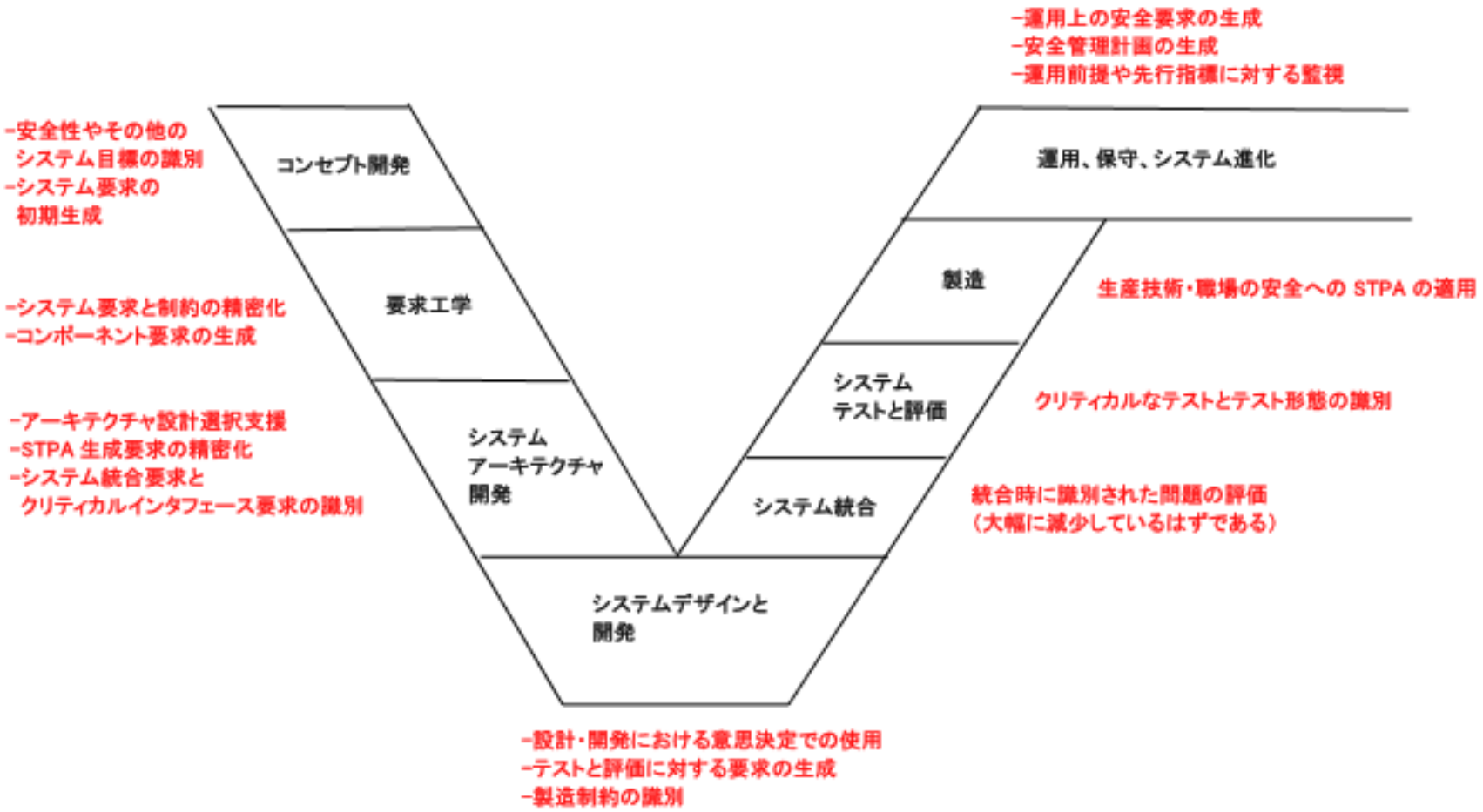
STPA HANDBOOK



目次

https://psas.scripts.mit.edu/home/materials/

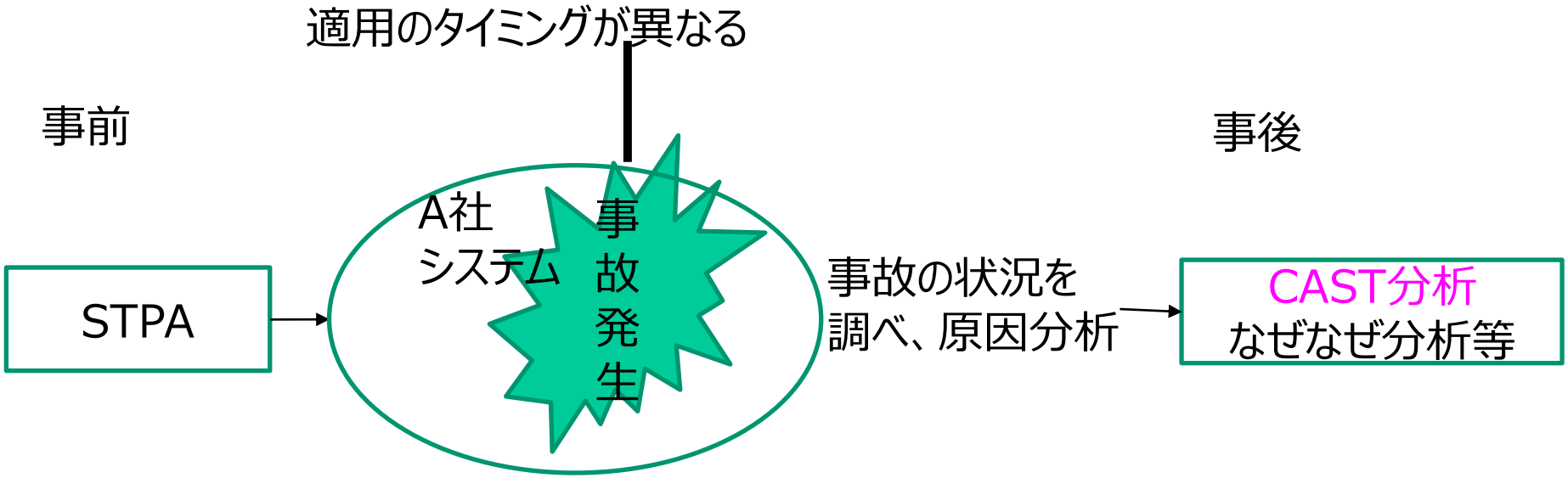
STPAのシステムエンジニアリングプロセス統合



CASTの概要

システム理論に基づく原因分析手法である CAST（Causal Analysis using System Theory）はSTPA とは異なり，STAMP 事故モデルの考えに基づいた事後分析手法

CAST は事故全体の理解 を可能とするフレームワークとプロセスを提供し，事故の原因分析を安全制御構造の破綻にフォーカスし，先入観や偏見による影響や偏りを小さくする事故分析技術



過去の事故に対するCAST分析は、さらなる損失を防ぐために排除または管理する必要がある妥当なシナリオを識別することによってSTPAプロセスを支援することができる

国立研究開発法人産業技術総合研究所（産総研）の情報システムに対する外部からの不正なアクセスの概要

発生日時：2018年2月6日
産総研の主たる情報システムである
① クラウドサービスを利用するメールシステム
② 独白に構築する内部システムの双方に順次不正なアクセスが行われた

---

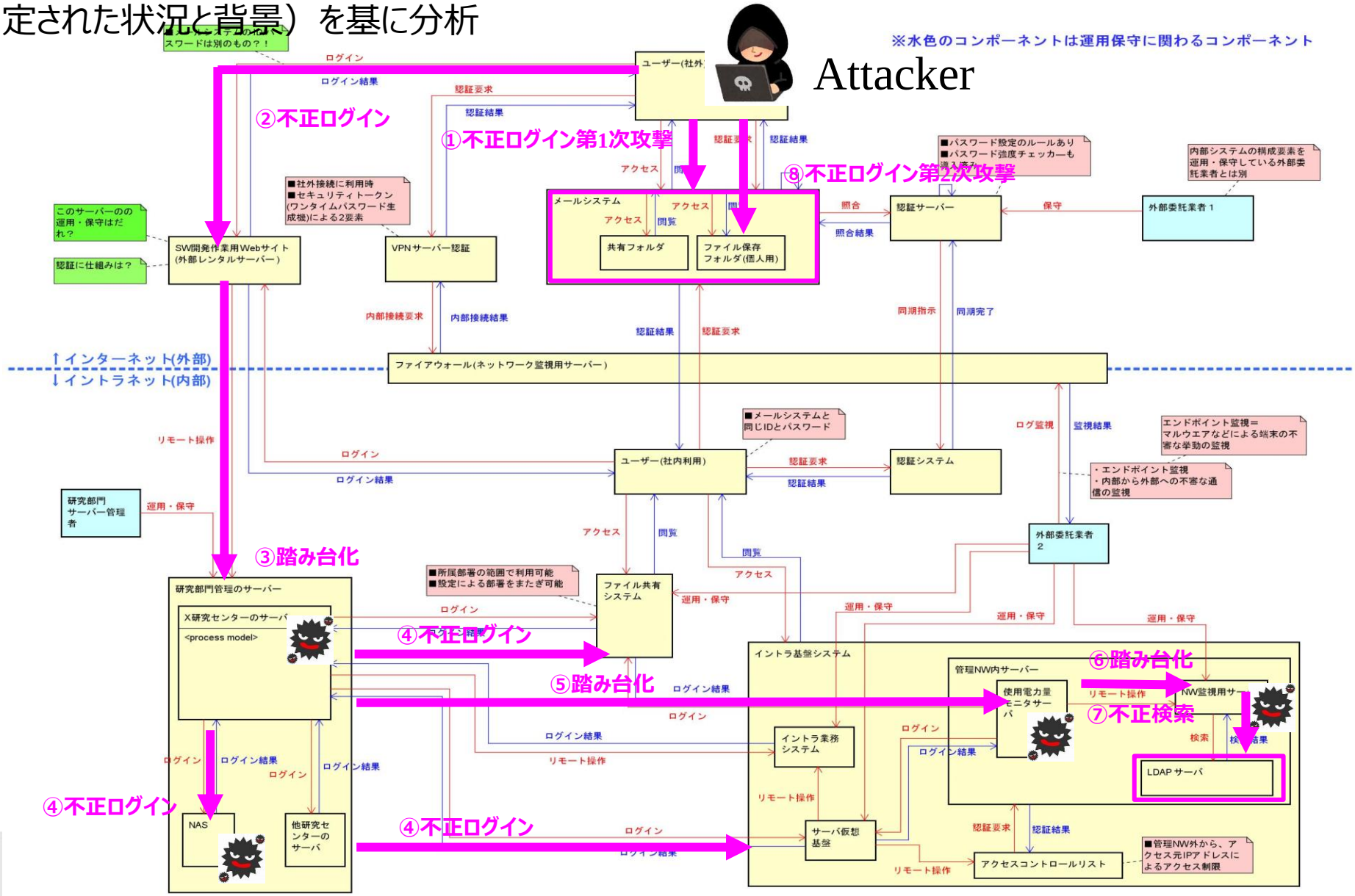
① 職員のログインIDの窃取
② パスワード試行攻撃によるパスワード探知
③ 職員のログインID・パスワードを用いた、内部システムへの不正侵人
④ 内部システムのサーバの「踏み台」化
⑤ メールシステム及び内部システムの複数のサーバに保管したファイルの窃取又は閲覧

といった一連の不正行為が行われた。

産総研の情報システムに対する不正なアクセスに関する報告より

抽象的コンポーネントレベルでの分析

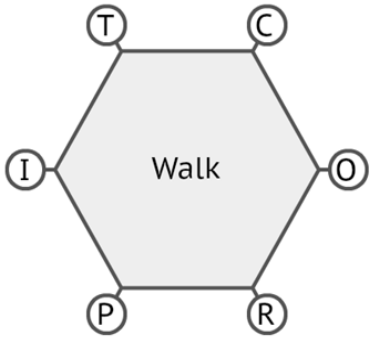
コンポーネント単体ではなく,複数のコンポーネントが関わり合って発生した事象＝抽象的事象と捉え,4つの観点（安全上の責務,非安全なコントロールアクション,プロセス/メンタルモデルの欠陥,意思決定された状況と背景）を基に分析



レジリエンスエンジニアリング

- レジリエンスとは復元力、回復力を意味する言葉であり、東日本大震災などの破壊的な被害から注目をあびるようになった概念
- レジリエンスエンジニアリングの提唱者で世界的権威：エリック・ホルナゲル教授
- レジリエンスエンジニアリングが提唱することの一つは、安全を旧来の考え方から「物事が正しい方向へと向かうことを保証する、すなわち、うまくいっていることから学ぶ」という新たな考え方への変革を促すこと。複雑な工学システムにおいては、重大な事故や損失が起こってから対応するのでは社会的影響が大きすぎるため、この考え方を取り入れることが必須

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。
FRAM分析は、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
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。



- I Input : 機能のトリガー
- P Precondition : 機能の実行に必要な前提条件
- R Resource : 機能の実行に必要な資源
- T Time : 機能の実行における時間制約
- C Control : 機能の実行を制御する制御パラメータ
- O Output : 機能の出力

IPA,はじめてのSTAMP/STPA（活用編）より

Ⅳ. セーフティ・セキュリティ統合フレームワーク

セーフティ＆セキュリティ

セーフティとセキュリティの概念

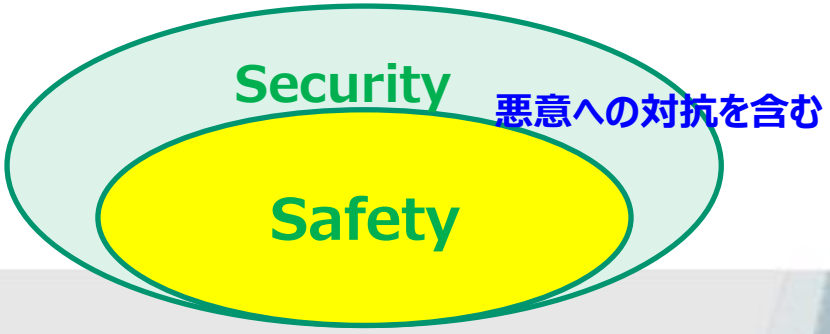
- 安全：受容できないリスクがないこと（ISO／IECガイド51）
- 守る対象で考えてみると．．．

Safety セーフティ：
偶発的なミス、故障などの悪意のない危険に対する安全

Security セキュリティ：
悪意をもって行われる脅威に対しての安全

- **英語の語義**

SE（離れる）＋**CURE**（気がかり）→ **Securityは安心？**



クイズ2

セーフティとセキュリティを連携して行うことが難しい理由は以下のどれか？

①担当部署が異なり、つながりがない
②技術や知識が異なり、相互理解が困難
③標準規格が異なり、業務プロセスも別々
④セーフティ設計だけでも大変なのにセキュリティ設計を考慮するのは困難

答えは全部

どこが違うから難しい？

セーフティとセキュリティの特徴の比較

相違点	セーフティ	セキュリティ
保護対象の違い	人命、財産（家屋等）など	情報の機密性、完全性、可用性など
原因の違い	合理的に予見可能な誤使用、機器の機能不全	意図した攻撃
被害検知の違い	事故として表れるため、検知しやすい	盗聴や侵入など、検知しにくい被害も多い
発生頻度	発生確率として扱うことができる	人の意図した攻撃のため確率的には扱えない
対策タイミング	設計時のリスク分析・対策で対応	時間経過により新たな攻撃手法が開発されるので継続的な分析・対策が必要
フォーカスポイント	ハードウェアまたは人中心	ソフトウェア中心
対策の仕方	網羅的で徹底した対処	ベスト・エフォート的対処
歴史的視点	長い歴史があり、多くの標準的で伝統的な分析方法が存在する。	コンピュータとインターネットの進化により考慮する必要性が生じた
標準	ドメインごとに標準があり、準拠が必須	マネジメント標準は概ね普及しているが、開発と設計に関するITセキュリティの国際基準は十分な普及はしていない
リスク分析	FTA、FMEA、HAZOPなど、多くの従来手法がある	セキュリティが追加されたセーフティ手法やソフトウェアエンジニアリングが考案されているが普遍的ではない

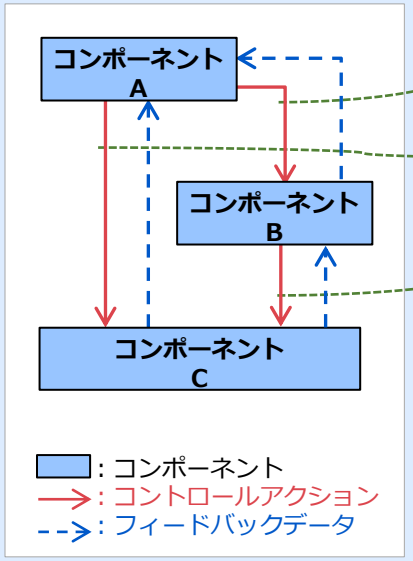
STAMP/STPAの実施手順

【Step0-準備1】➡ 【Step0-準備2】➡ 【Step1】 ➡ 【Step2】

システムレベルのアクシデント、ハザード、安全制約の識別

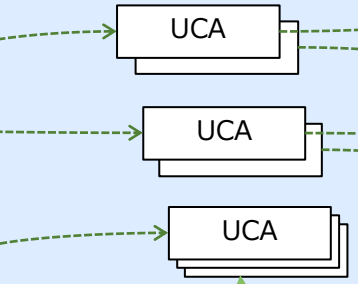
コンポーネント間の制御関係を表すモデルの構築

※コントロールストラクチャー

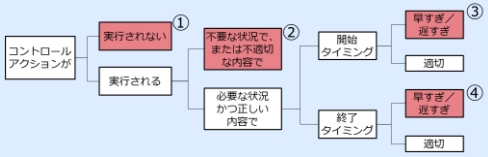


ハザードにつながるコントロールアクションの識別

※UCA
(Unsafe Control Action)

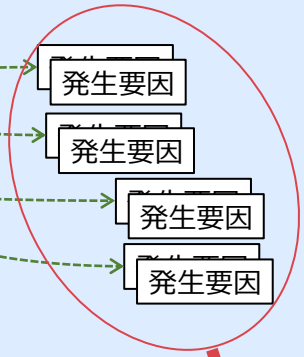


危険な制御のパターンを利用して分析



UCAの発生要因の分析

※HCF
(Hazard Causal Factor)



コンポーネントが満たすべき安全要件を導出

STAMP/STPAの実施手順(1)

【Step0-準備1】

システムレベルのアクシデント、ハザード、安全制約の識別

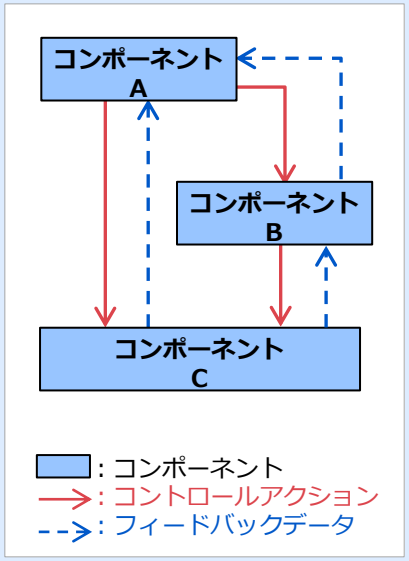
アクシデント	ハザード	安全制約
自動車が外部環境（歩行者/他の車/周辺物）と衝突/接触する	自動車が、ブレーキをかけても、外部環境の前で停止できない (H1)	自動車が、外部環境と衝突しないようにブレーキをかける（外部環境までの距離や相対速度を制御する） (SC1)
	ブレーキがかからない (H2)	運転手と自動車の両方がブレーキをかけられない状態にならない (SC2)
	急ブレーキにより後方車両から追突される (H3)	SC1に違反しない程度に緩やかに減速する (SC3)

STAMP/STPAの実施手順(2)

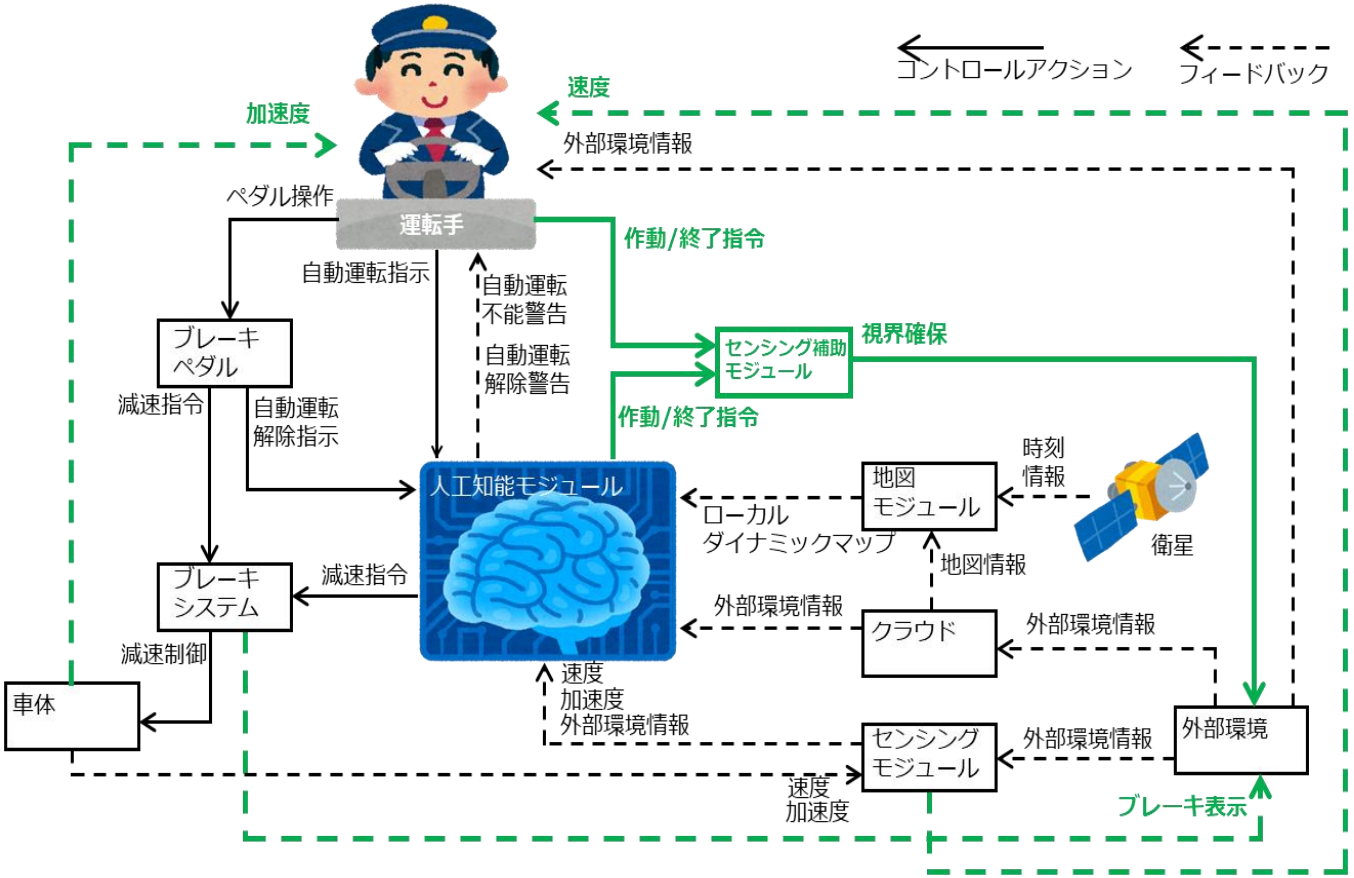
【Step0-準備2】

コンポーネント間の制御関係を表すモデルの構築

※コントロールストラクチャー



今回の実験でのコントロールストラクチャー



(*) 緑色: 昨年度のCSからの追加

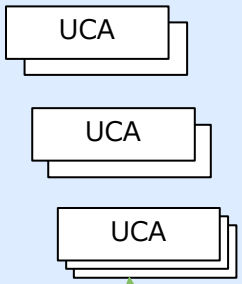
STAMP/STPAの実施手順(3)

【Step1】

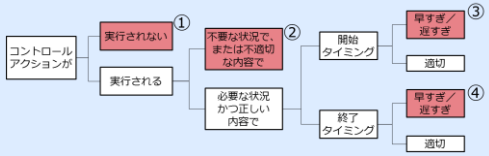
Unsafe Control Actionの一覧（一部抜粋）

ハザードにつながるコントロールアクションの識別

※UCA
(Unsafe Control Action)



危険な制御のパターンを利用して分析



No	CA	Not Providing	Providing causes hazard	Too early / Too late	Stop too soon / Applying too long
1	運転手によるブレーキペダル操作	(UCA1-N) 自動運転不能時に運転手がペダルを踏まないと減速指令が出ず外部環境と衝突する. [SC1][SC2]	(UCA1-P) 自動運転中に意図しないペダル操作が発生し，自動運転が解除されブレーキが作動しなくなり外部環境と衝突する [SC1][SC2]	(UCA1-T) 非自動運転時にペダル操作が遅すぎる場合，減速指令が遅れ，外部環境と衝突する [SC1]	(UCA1-D) 非自動運転時にペダルを踏む時間が不足すると，減速指令が不足して外部環境と衝突する [SC1] ブレーキを踏む時間が長すぎると必要以上に減速し，渋滞の原因となる
2	ブレーキペダル操作によるブレーキシステムへの減速指令	(UCA2-N) 減速指令がないと，そのまま外部環境と衝突する [SC1]	(UCA2-P) 不必要に強い減速指令が出され，後方車両から追突される [SC3]	(UCA2-T) 運転手のペダル操作に対して減速指令が遅すぎた場合，外部環境との適切な距離が保てず衝突する [SC1]	(UCA2-D) 十分な減速が行われる前に減速指令が終了し，外部環境との適切な距離が保てず衝突する [SC1] 必要な減速が完了した後も減速指令を出し続け，加速が困難になる
3	ブレーキシステムによる車体の減速制御	(UCA3-N) 減速制御が行われないと，そのまま走行方向の外部環境と衝突する [SC1][SC2]	減速指令を受けてないのに減速が発生し，交通渋滞となる (UCA3-P) 不必要に強い減速が生じ，後方車両から追突される [SC3]	(UCA3-T-1) 減速指令に対して減速が遅すぎる場合，外部環境との適切な距離が保てず衝突する [SC1] (UCA3-T-2) 外部へのブレーキ表示前に減速を始め，後方の車両から衝突される [SC3]	(UCA3-D) 減速指令が終了する前に減速が終了し，外部環境との適切な距離が保てず衝突する [SC1] 減速指令が終了した後も減速制御を行い，加速が困難になる
4	運転手による人工知能モジュールへの自動運転指示	自動運転指示が行われないと，自動運転が開始されない	意図しない自動運転が開始され，運転手が混乱する	-	-
5	人工知能モジュールによるブレーキシステムへの減速指令	(UCA5-N 自動運転時に人工知能が減速指令を出さないとそのまま外部環境と衝突する. [SC1]	(UCA5-P) 不必要に強い減速指令が出され，後方車両から追突される [SC3]	(UCA5-T) 減速指令が遅れた場合，前方の外部環境との適切な距離が保てず衝突する [SC1]	(UCA5-D) 十分な減速が行われる前に減速指令が終了し，外部環境との適切な距離が保てず衝突する [SC1] 必要な減速が完了した後も減速指令を出し続け，加速が困難になる
6	ブレーキペダル操作による人工知能モジュールへの自動運転解除指示	自動運転の解除指示が行われないと，運転手がブレーキ操作をすることができない	(UCA6-P) 意図せず自動運転が解除され，衝突回避のためのブレーキ指令をだすことができない [SC1][SC2]	-	-
7	人工知能モジュールによるセンシング補助モジュールの作動/終了指令	視界確保無しで人工知能モジュールが外部環境を認識できない場合，作動指令がなされないと人工知能による自動運転が不可能となる	不要な視界確保動作が行われ，部品の寿命が縮む	-	-
8	運転手によるセンシング補助モジュールの作動/終了指令	(UCA8-N) 非自動運転かつ運転手が視界の確保無しで運転ができない状況となった場合に作動指令が出せないと，運転手が外部環境を認識できず，ブレーキをかけずに外部環境と衝突する [SC1]	不要な視界確保動作が行われ，部品の寿命が縮む (UCA8-P) 非自動運転かつ運転手が視界の確保無しで運転ができない状況となった場合で，運転手の視界確保可能な状況となる前に終了指令が出されると，運転手が外部環境を認識できず，ブレーキをかけずに外部環境と衝突する [SC1][SC2]	-	-
9	センシング補助モジュールによる外部環境の視界確保	(UCA9-N) 視界の確保が行われないと，運転手/人工知能が外部環境を認識できず，ブレーキをかけずに衝突する [SC1][SC2]	(UCA9-P) 運転手/人工知能が外部環境を認識できないほど視界の確保機能が働き，ブレーキをかけずに衝突する [SC1][SC2]	(UCA9-T) 視界の確保作動のタイミングが遅すぎ，運転手/人工知能が外部環境を認識できない状態となり，ブレーキをかけずに衝突する [SC1][SC2]	-

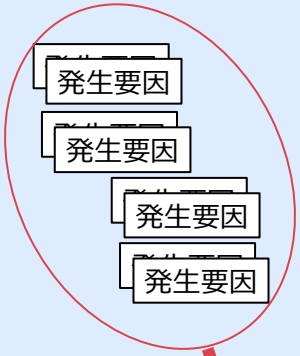
STAMP/STPAの実施手順(4)

【Step2】

Hazard Causal Factorの一覧（一部抜粋）

UCAの発生要因の分析

※HCF (Hazard Causal Factor)



コンポーネントが満たすべき安全要件を導出

UCAx	HCF (1) コントロールの入力が外部情報が欠けているか間違っている	(2) コントロールアルゴリズムの生成の欠陥、プロセス変更、不正確な修正や適応	(3) プロセスモデルの矛盾、不完全、不正確	(4) コンポーネント故障、経時変化	(5) 不適切か欠けているフィードバックのフィードバックの遅れ	(6) 情報が与えられない/間違っている、測定が不正確。フィードバックの遅れ	(7) 遅れたアクション	(8) 不適切、有効でないがたコントロールアクション	(9) プロセスへの入力が欠けている/間違っている	(10) 識別されない/許範囲外の妨害	(11) プロセスの出力がシステム/サブ-ドウン区に	(12) アクチュエーターの不適切なオペレーション	(13) センサーの不適切なオペレーション	(14) 他のコントローラーの通信が欠けている/間違っている	(15) 矛盾するコントロールアクション	Spoofing（なりすまし）	Tampering（改ざん）	Repudiation（否認）	Information Disclosure（情報漏えい）	Denial of Service（サービス拒否）	Elevation of Privilege（特権の昇格）
(UCA1-D) 非自動運転時にペダルを踏む時間が不足する、減速指令が不足して外部環境と衝突する [SC1]		運転手が前による路面摩擦の低下を考慮せずにブレーキ操作をする	運転手が前方の車両の減速度が増加していることに気がつかない					運転手がペダルを踏んでいないが、ブレーキのあそび部分しか踏んでいない									運転手の意識をそらし、ブレーキから足を離すようにしむける				
(UCA1-N) 自動運転不能時に運転手がペダルを踏まない減速指令が出ず外部環境と衝突する [SC1][SC2]	悪天候などの外部環境の情報が解釈でなく、運転手が危険原因知らない	運転手が十分な操作知識を持っておらず、ペダルを踏む場所を誤って覚えている	運転手が危険な領域にたが自動運転を過信して、ブレーキを踏まない		スピード表示が実際より低速となっており、危険な速度であることを察知しない			他のペダルと踏み間違える								外部環境の一部になりすまし、誤った外部環境情報をクラウドへ送信。自動運転不能な状態であっても人工知能に自動運転可能であると判断させ、自動運転を継続させない	-クラウドからの情報を改ざんし、自動運転不能な状態であっても人工知能に自動運転可能であると判断させる。-外部環境からセンシングモジュールへの情報を改ざんし、自動運転不能な状態であっても人工知能に自動運転可能であると判断させ、自動運転不能警告を出させない。-外部環境からセンシングモジュールへの情報を改ざんし、自動運転不能な状態であっても人工知能に自動運転可能であると判断させ、自動運転不能警告を出させない。-自動運転不能警告を改ざん（運転手の気をそらす・警告手段を破壊するなど），運転手への警告を遮断する			-人工知能モジュールに誤判断（物理的なものを含む）を与え、自動運転不能である ことを報知できなくさせる。-運転手に対して大量or致命的な情報を与え、正常な判断をすることが不可能な状態にする	
(UCA1-P) 自動運転中に意図しないペダル操作が発生し、自動運転が解除され、ブレーキが作動しなくなり外部環境と衝突する [SC1][SC2]		自動運転解除警告以外に気をとられ、自動運転が解除されたことを認識しない	運転手の足が意図せずペダルに触れてしまう								車の振動などの際性力でペダルが動作する			自動運転解除警告が表示されない			-運転手以外の何かがペダルに触れる-自動運転解除警告を表示させない				
(UCA1-T) 非自動運転時にペダル操作が遅すぎる場合、減速指令が遅れ、外部環境に衝突する [SC1]	外部環境が見えなく、運転手の判断が遅れる	体調などで運転手の判断能力が低下しており、ペダルを踏むのが遅れる			実際の速度よりも低速で速度表示がなされ、運転手が危険回避までの猶予があると勘違いする		ペダルのあそびが大きく、運転手の指示よりも減速指令が遅れる				経年劣化でペダルが硬くなり、ブレーキ指示を出すまでに時間がかかる	速度表示をせず、運転手が速度を認識できなくなる	自動運転不能の警告が遅れ、運転手が非自動運転であることの認識が遅れる			風などで信号と運転手の間に障害物が入るor信号の向きがかわることにより、運転手が危険を察知するのが遅れる			信号機が破壊し、運転手を迷わせる	
(UCA2-D) 十分な減速が行われる前に減速指令が終了し、外部環境との適切な距離が保てず衝突する [SC1]														運転手が誤れた足で操作することによりペダルが滑り、減速指令が解除される				接点不良などの異常が生じ、減速指令が出力されない										
(UCA2-N) 減速指令がないと、そのまま外部環境と衝突する [SC1]																		接点故障などにより、ペダル操作がなされても、減速指令を出力がされない							ブレーキシステムへの減速指令出力を改造し、指令がブレーキシステムへ届かないようにされている			ブレーキペダルが破壊されている	
(UCA2-P) 不必要に強い減速指令が出され、後方車両から追突される [SC3]				ブレーキシステムの故障により緩やかな減速が不能となる				-ブレーキペダルの遊びによる減速指令の遅れが生じることによる運転手の焦りにより、ペダルが強く踏み込まれる-雨により制動距離が長くなったことにより生じた運転手の焦りにより、ペダルが強く踏み込まれる									ペダルの反力が弱く、必要以上に運転手がペダルを踏み込む				ブレーキシステムへの指令が改ざんされ、不要な減速指令が出力される				
(UCA2-T) 運転手のペダル操作に対して減速指令が遅すぎた場合、外部環境との適切な距離が保てず衝突する [SC1]								-ペダルの遊びが大きく、減速指令が遅れる-雨でペダルが滑り、減速指令が遅れる																				

セキュリティ・バイ・デザインを可能とするSTAMPのセキュリティへの取り組みの経緯

CSS2017：安全解析手法STAMP/STPAに対するセキュリティ視点からの脅威分析の拡張提案

2018年3月CSEC研究会：安全性解析手法STAMP／STPAへの脅威分析（＝STRIDE）の適用

2018 年 12月　Threat analysis using STRIDE with STAMP/STPA, APSEC2018Workshop
（STAMP/STPA-SafeSecとの比較論文）

STAMP S&S～レベル３自動運転事例によるセーフティ・セキュリティ統合リスク分析

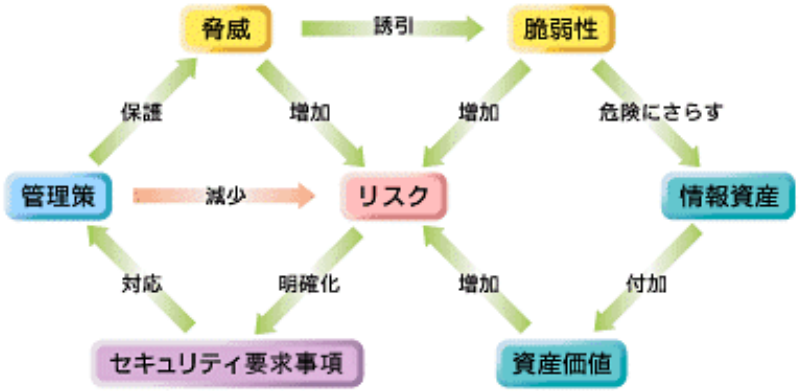
セーフティ&セキュリティのリスクの同時分析。STRIDEを組合せ：
実験1の実施

STAMP/STPAを用いた場合と用いない場合のセーフティ・セキュリティのリスク分析に差があるのか：
実験2の実施

脅威分析と脆弱性分析

- 脅威≠脆弱性
  - 脅威は攻撃者のおこすもの
  - 攻撃者は脆弱性を探して悪用する

情報セキュリティリスクの大きさ=
情報資産の価値×脅威の大きさ×脆弱性の度合い

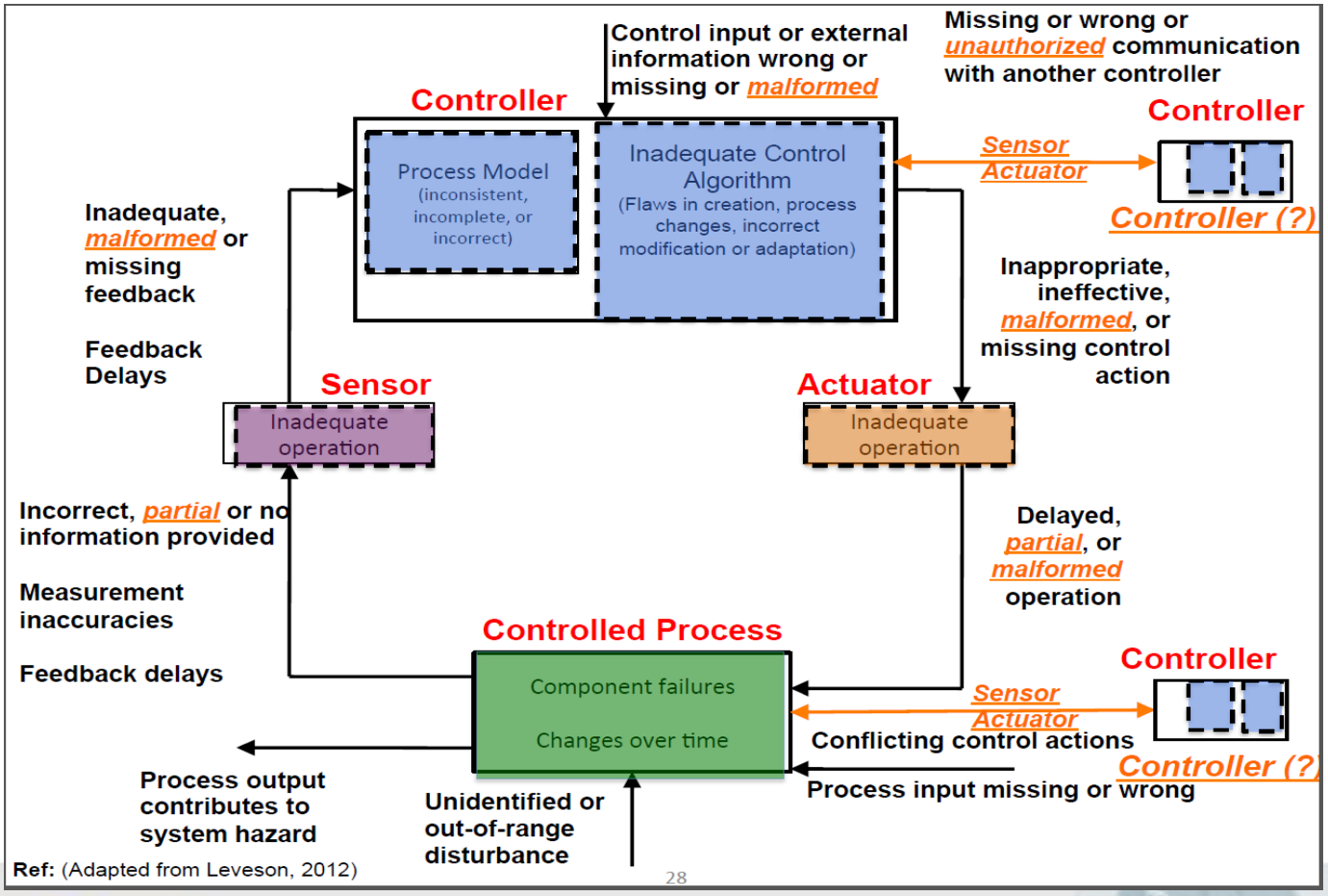


情報資産、脆弱性、脅威の識別とその因果関係（GMITS Part1）

- セキュリティ設計とは、設計対象で起こりうる脅威を認識し、脅威の対抗に必要なセキュリティ要件を定義すること

STPAとSTPA-Secの違いは？

◆STPAとSTPA-Secの分析手順は基本的には変わらない。
◆ただしセキュリティー上の脅威抽出に必要な分析の視点が追加→
malformed（不正な形式の）unauthorized（正当な権限のない）



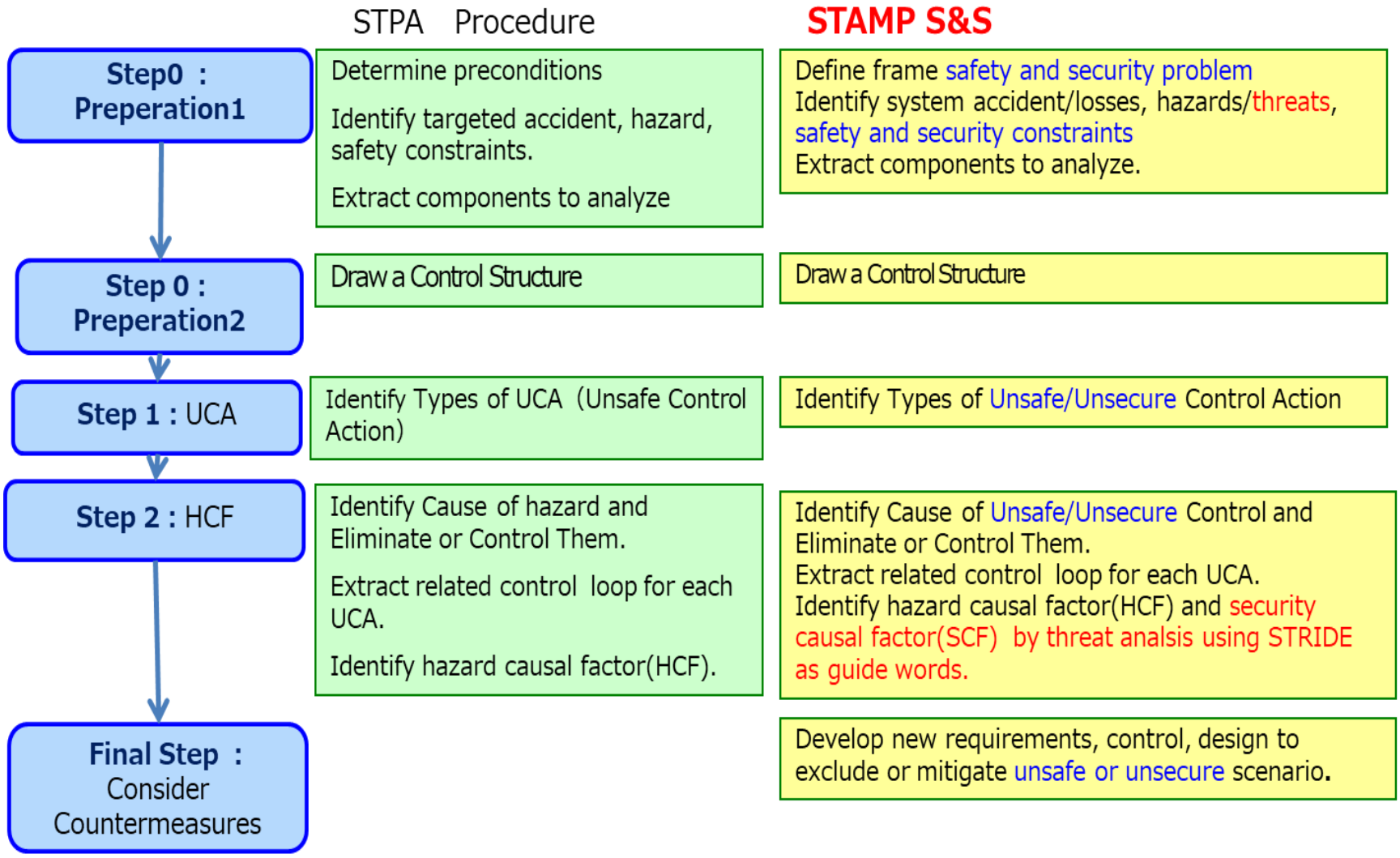
STRIDE

マイクロソフト社が定義する脅威モデル

脅威	訳	説明
Spoofing Identity	なりすまし	コンピュータに対し、他のユーザを装うこと
Tampering	改ざん	データを意図的に操作すること
Repudiation	否認	ユーザがあるアクションを行ったことを否認し、相手はこのアクションを証明する方法がないこと
Information Disclosure	情報の暴露	アクセス権限を持たない個人に情報が公開されていること
Denial of Service	サービス不能	攻撃により正規へのユーザへのサービスが中断される
Elevation of Privilege	権限の昇格	権限のないユーザがアクセス権限を得ること

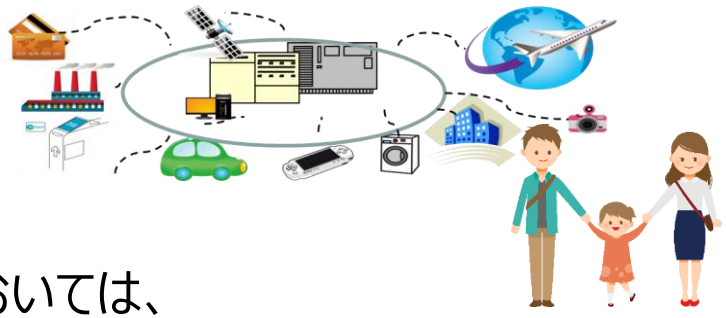
＊STRIDEはIoTセキュリティ用が公開されている。*Microsoft，Azure IoT リファレンス アーキテクチャの脅威のモデル化，https://docs.microsoft.com/ja-jp/azure/iot-suite/iot-security-architecture#threat-modeling-the-azure-iot-reference-architecture

脅威分析を追加


STPA Procedure
STAMP S&S
Step0 : Preperation1
Determine preconditions
Identify targeted accident, hazard, safety constraints.
Extract components to analyze
Define frame safety and security problem
Identify system accident/losses, hazards/threats, safety and security constraints
Extract components to analyze.
Step 0 : Preperation2
Draw a Control Structure
Draw a Control Structure
Step 1 : UCA
Identify Types of UCA (Unsafe Control Action)
Identify Types of Unsafe/Unsecure Control Action
Step 2 : HCF
Identify Cause of hazard and Eliminate or Control Them.
Extract related control loop for each UCA.
Identify hazard causal factor(HCF).
Identify Cause of Unsafe/Unsecure Control and Eliminate or Control Them.
Extract related control loop for each UCA.
Identify hazard causal factor(HCF) and security causal factor(SCF) by threat analsis using STRIDE as guide words.
Final Step : Consider Countermeasures
Develop new requirements, control, design to exclude or mitigate unsafe or unsecure scenario.

STAMP S&S

- STAMP S&S:システム理論に基づく安全性，リスク，事故分析などの様々な分析技術とその技術による取り組み
- STAMP S&SのSTAMPはSystem Theoretic Accident Model and Process ではなく、System Theoretic Architecture Model and Process
- S&S: Safety, Securityの他, System, Software, Service, Stakeholder, Society, Specification, Standard, Scenarioの略称を指す.
- 目的:STAMPの適用範囲の広さをベースにしてSTAMPの各種分析方法等をより広範囲に異なる観点で適用することでSTAMPの可能性を引き出しその具体的な適用方法を確立
- 機器＋システム＋人＋組織
- 多様なコンポーネントの相互作用を分析



＊STAMP S&Sは、Safety＆Securityの範疇においては、
STPA-Secによるセキュリティ分析のプロセスに対して、脅威分析を実施することである
脅威分析の方式として、STRIDEによる脅威モデリングを実施した。

実験結果：

問3: 被験者1人当たりのリスク件数

	STAMP S&Sを用いた分析	STAMP S&Sを用いない分析
平均（件数）	70.7	6.1

11.7倍！

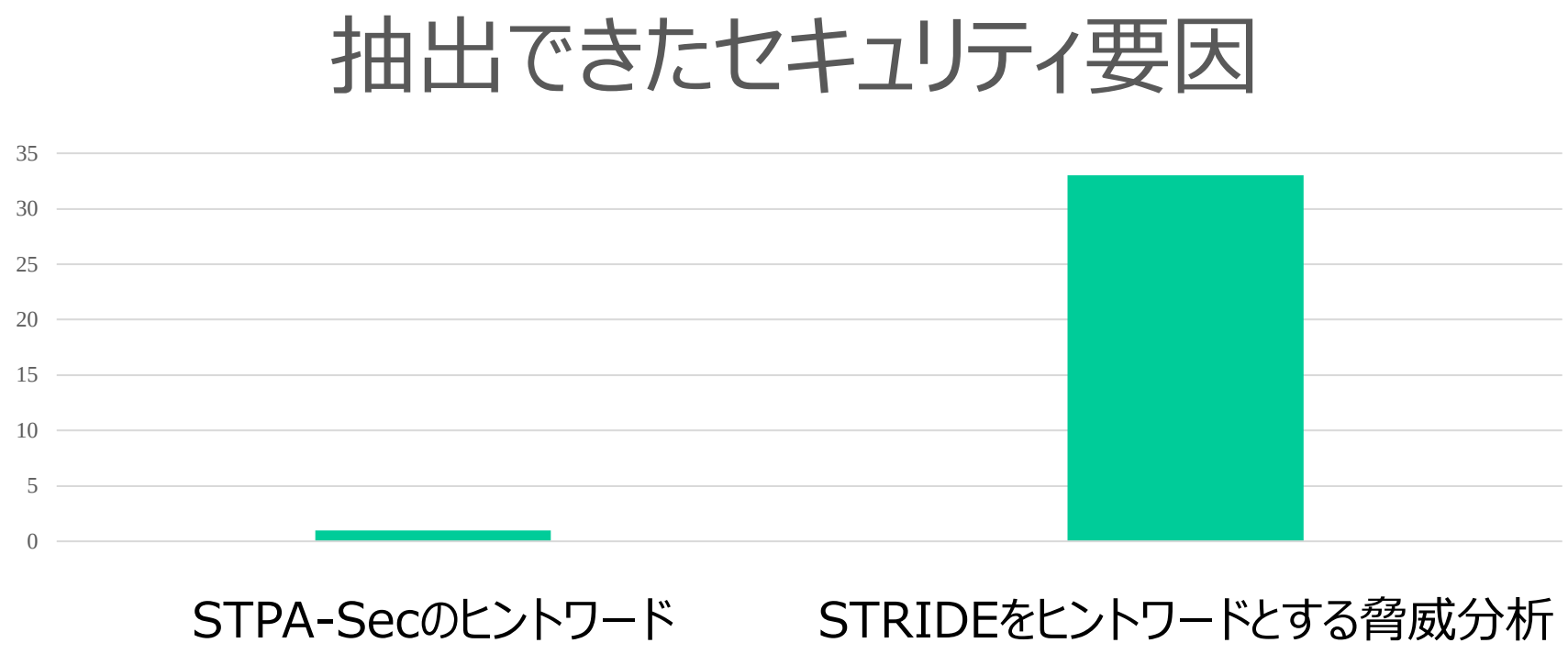
問4: 被験者1人当たりのリスク対策件数

	STAMP S&Sを用いた分析	STAMP S&Sを用いない分析
平均（件数）	63.8	4.4

14.5倍！

リスク件数・リスク対策件数ともに
STAMP S&Sを用いた分析の方が多い

セキュリティ要因抽出結果

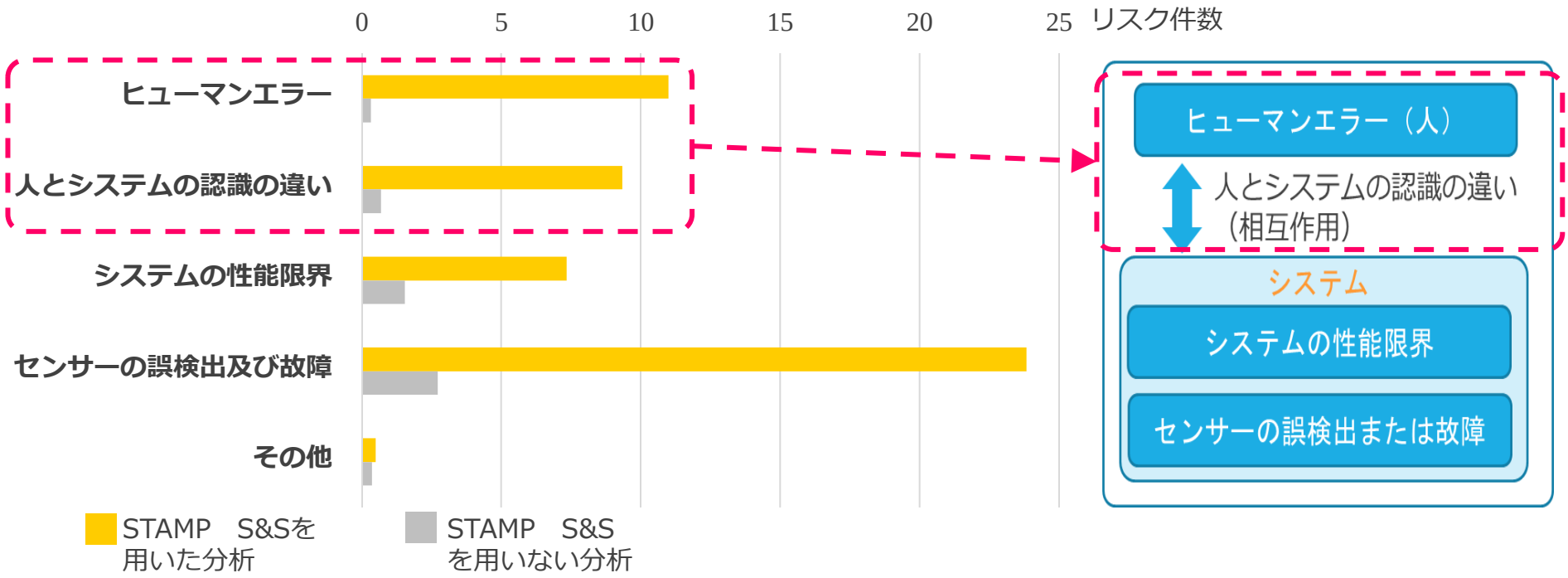


1件

３３件

集計結果のグルーピング

被験者1人当たりのリスク件数の内訳(セーフティ)



機能安全規格や従来の安全分析手法では対象としていないヒューマンエラー、人とシステムの認識の違い、システムの性能限界（SOTIF)の洗い出しを可能

まとめーSTAMP S&Sの適用効果

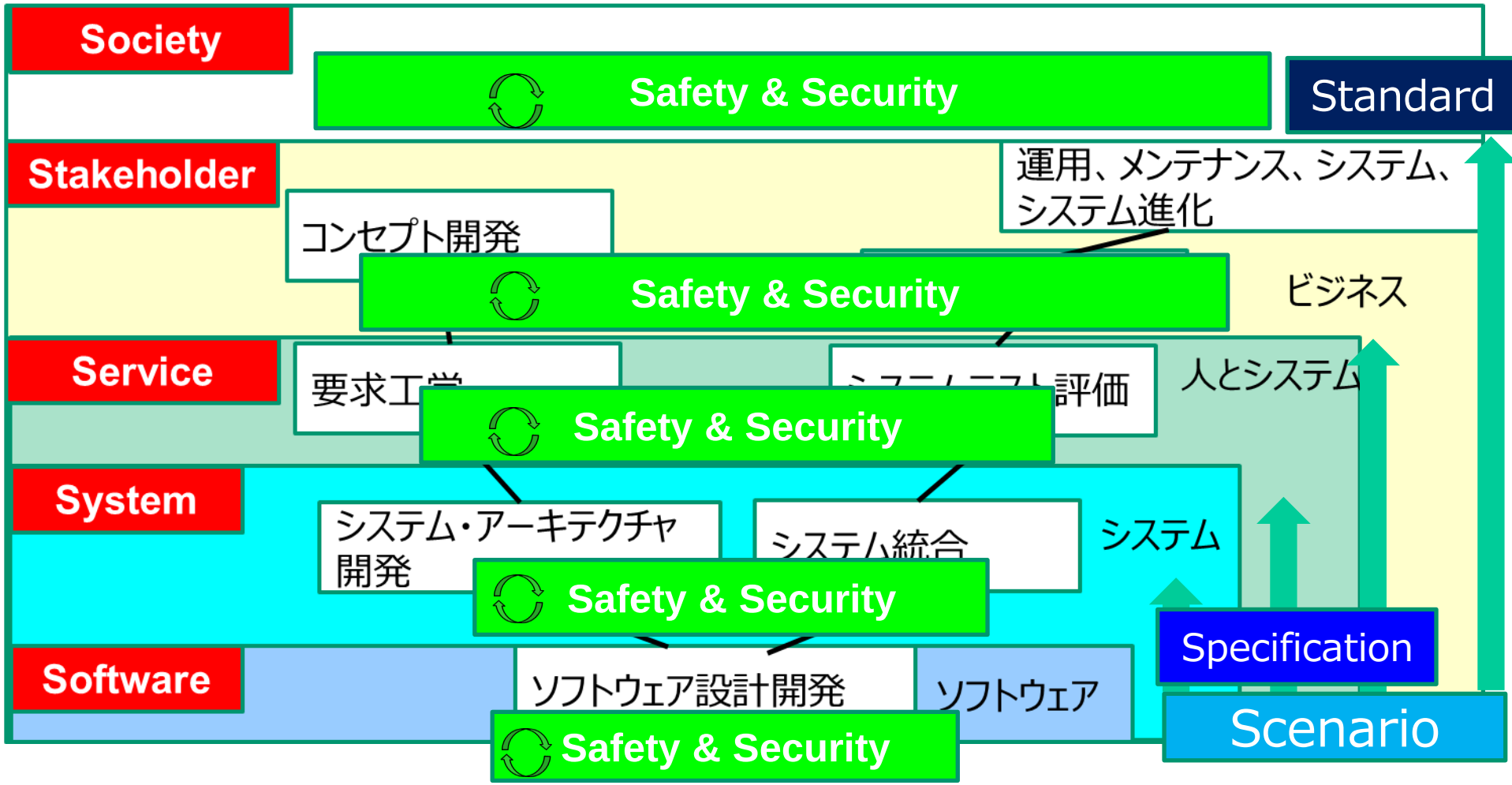
＜実験1＞

- STAMP/STPAに脅威分析を追加実施することで、コンセプト・企画段階でのセキュリティ要求分析（＝セキュリティ・バイ・デザイン）実施可能
- 脅威分析はSTPA　STEP２でSTRIDEをヒントワードとして利用することで、はるかに多くの（３０倍）脅威を抽出できた。
- 共通のCS図上でセーフティとセキュリティを同時分析可能

＜実験２＞

- 機能安全規格や従来の安全分析手法では対象としていないヒューマンエラー、人とシステムの認識の違い、システムの性能限界（SOTIF)の洗い出しを可能（セーフティの長所）
- セキュリティでは専門家ではなくても全員セキュリティリスクを抽出できた（セキュリティの長所）

社会への影響も含めた階層的モデルの必要性



どの階層においても、改善をしつつ、セーフティとセキュリティを確立し、社会への影響をシナリオ分析しニーズをとらえる。アウトプットは仕様や標準となる。

→社会技術システムの安全安心の確立。

STAMP Workbenchの基本コンセプト

STAMP支援ツール

STAMP Workbench

図表作成・編集の煩わしさから解放
分析者は思考のみに専念

Download URL & QR code

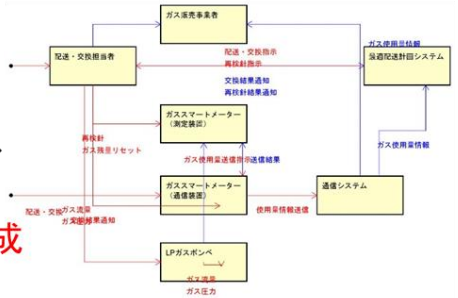
https://www.ipa.go.jp/sec/tools/stamp_workbench.html

- ツールがSTAMP/STPAの手順を誘導
- 要求仕様を整理するコンポーネント抽出表から、CS図の雛型を自動生成
- 人は考えることだけに専念できる

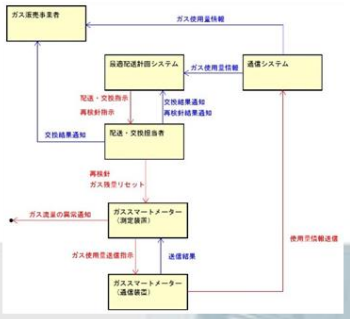
自然言語の要求仕様書

表形式で整理

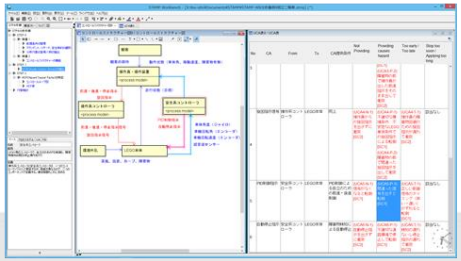
雛型を自動生成



優れた編集機能でCS図を完成



変更は全ての図表にリアルタイムに自動反映
IDの振り直しもリアルタイムに自動実行



表作成やID採番は自動化

STAMP/STPAガイドブック

「は P/STPA」入門編

- 理するためのSTPA手順解説書
- 分かり易い事例を用い、
  TPAの手順を具体的に解説

http p/sec/reports/20190428.html

「はじめてのSTAMP/STPA（実践編）」

- STAMPをやってみる（実践）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

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（活用定着）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

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

「STAMPガイドブック ～システム思考による安全分析～」

～STAMPの本質を理解してさらなる有効活用を～

- 「システム思考に基づく安全分析の本質」・・・システム思考の一般的な説明と安全分析との関係の解説
- 「STAMPの効果的な活用事例と解説」・・・システム思考が活かされたSTAMP分析事例の紹介
- 「STAMP/STPA演習教材」・・・STAMP/STPAを習得するための教材の紹介
- 「システム思考によるこれからの安全」・・・レジリエントなセキュリティの解説

<https://www.ipa.go.jp/ikc/reports/20190329.html>

機械学習を用いたシステムの
高品質化・実用化を加速する
“Engineerable AI”技術の開発

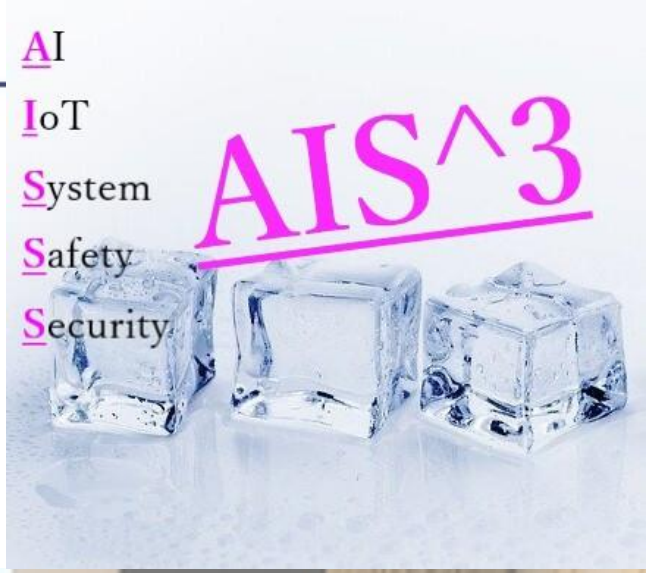
eAI

supported by

JST MIRAI 未来社会創造事業

QAML

AI
IoT
System
Safety
Security
AIS^3



機械学習システムセーフティ＆セキュリティWG
株式会社NTTデータ
有人宇宙システム株式会社

第2回
AI/IoTシステム
安全性シンポジウム

第2回AI/IoTシステム安全性シンポジウム概要（オンライン開催：Zoom+YouTube Live）

- 参加無料<https://qaml.jp/2020/08/29/ais3_2020_presentaion/>
- 11/10(火) 機械学習と安全性

  招待講演：浦本さん（前人工知能学会会長、三菱ケミカル）

  国立情報学研究所：石川准教授（eAIプロジェクト代表）

  パネルディスカッション
- 11/11（水）第1回Asian STAMP Workshop

  STAMP提唱者レブソン教授講演、Asian STAMP Workshop

  日本のSTAMPワークショップ（午後は日本語一般発表）
- 11/12(木) 第2回FRAM Workshop

  レジリエンス・エンジニアリング提唱者ホルナゲル教授講演、

  一般発表

シンポジウムの申し込み者数：国内626名＋英語参加11名

昨年は300弱→636人と倍増！！

AI／IoTシステム安全性グループ：→なんと846人

参考文献(STAMP関連)

1) Nancy G. Leveson, Engineering a Safer World, MIT Press, 2012.
2) STPA handbook, http://psas.scripts.mit.edu/home/
3) Nancy G. Leveson, STAMP Intro and Overview of STPA and CAST, http://psas.scripts.mit.edu/home/
4) William Young, Nancy G. Leveson. Systems Thinking for Safety and Security, Proceedings of the 29th Annual Computer Security Applications Conference （ACSAC 2013）, pp.1-8 （2013）.
5) Ivo Friedberg, Kieran, Paul Smith, David Laverty, and Sakir Sezer. STPA-SafeSec: Safety and security analysis for cyber-physical systems, Journal of Information Security and Applications, Volume 34, Part 2, pp.183-196 （2017）.
6) はじめてのSTAMP/STPA（入門編）, http://www.ipa.go.jp/sec/reports/20190428.html
7) はじめてのSTAMP/STPA（実践編）, http://www.ipa.go.jp/sec/reports/20170324.html
8) はじめてのSTAMP/STPA（活用編）, http://www.ipa.go.jp/sec/reports/20180328_2.html
9)STAMPガイドブック ～システム思考による安全分析～, https://www.ipa.go.jp/ikc/reports/20190329.html
10)金子朋子,「セキュリティ・バイ・デザインとアシュアランスケース」, SECジャーナル47号
11)金子朋子, 中沢潔, JETROニューヨーク事務所所長, ニューヨークだより5月特別号「システムの相互作用に着目したこれからの安全（STAMP） https://www5.jetro.go.jp/newsletter/nya/2018/IT/NYdayori_201805_Special.pdf
12)金子朋子,髙橋雄志,大久保隆夫,勅使河原可海,佐々木良一,情報処理学会(CSS2017）巻2017号：2　ページ：ROMBUNNO.3C3-2安全解析手法STAMP/STPAに対するセキュリティ視点からの脅威分析の拡張提案, 2017
13)金子朋子,早川拓郎,髙橋雄志,大久保隆夫,佐々木良一,安全性解析手法STAMP／STPAへの脅威分析（＝STRIDE）の適用, Vol.2018-CSEC-80,No.6,1-8, 2018
14)Threat analysis using STRIDE with STAMP/STPA, APSEC2018Workshop, 2018
15)A five-layer model for analyses of complex socio-technical systems ,Tomoko Kaneko, Nobukazu Yoshioka, PLoP2020,2020年9月
16)STAMP S&S: Safety & Security Scenario for Specification and Standard in the society of AI/IoT ,Tomoko Kaneko, Nobukazu Yoshioka(CFSE2020)
17)STAMP S&S: Layered Modeling for the complexed system in the society of AI/IoT Tomoko Kaneko, Nobukazu Yoshioka

ご清聴ありがとうございました!!

Q&A

何でもきいてくださいね