STAMPワークショップ
チュートリアル初級編
～システム思考によるソフトウェアの安全設計～

2018年12月3日

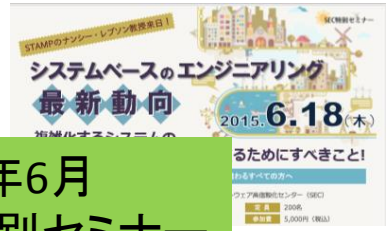
会津大学名誉教授　兼本　茂

STAMP/STPAをもう一度振り返ってみよう

- 背景
- 手順
- 事例

IPAでの活動（IoTシステム安全性向上技術WG）

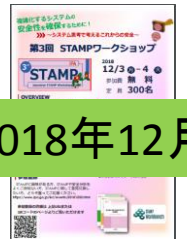
2015			2016			2017			2018		
4月		12月			3月	4月		12月			3月	4月		12月			3月



2015年6月
Sec特別セミナー
2016年1月
13thWOCS2

2016年12月

2017年11月



2018年12月

初級編

実践編

活用編

ツール

STAMP Workbench

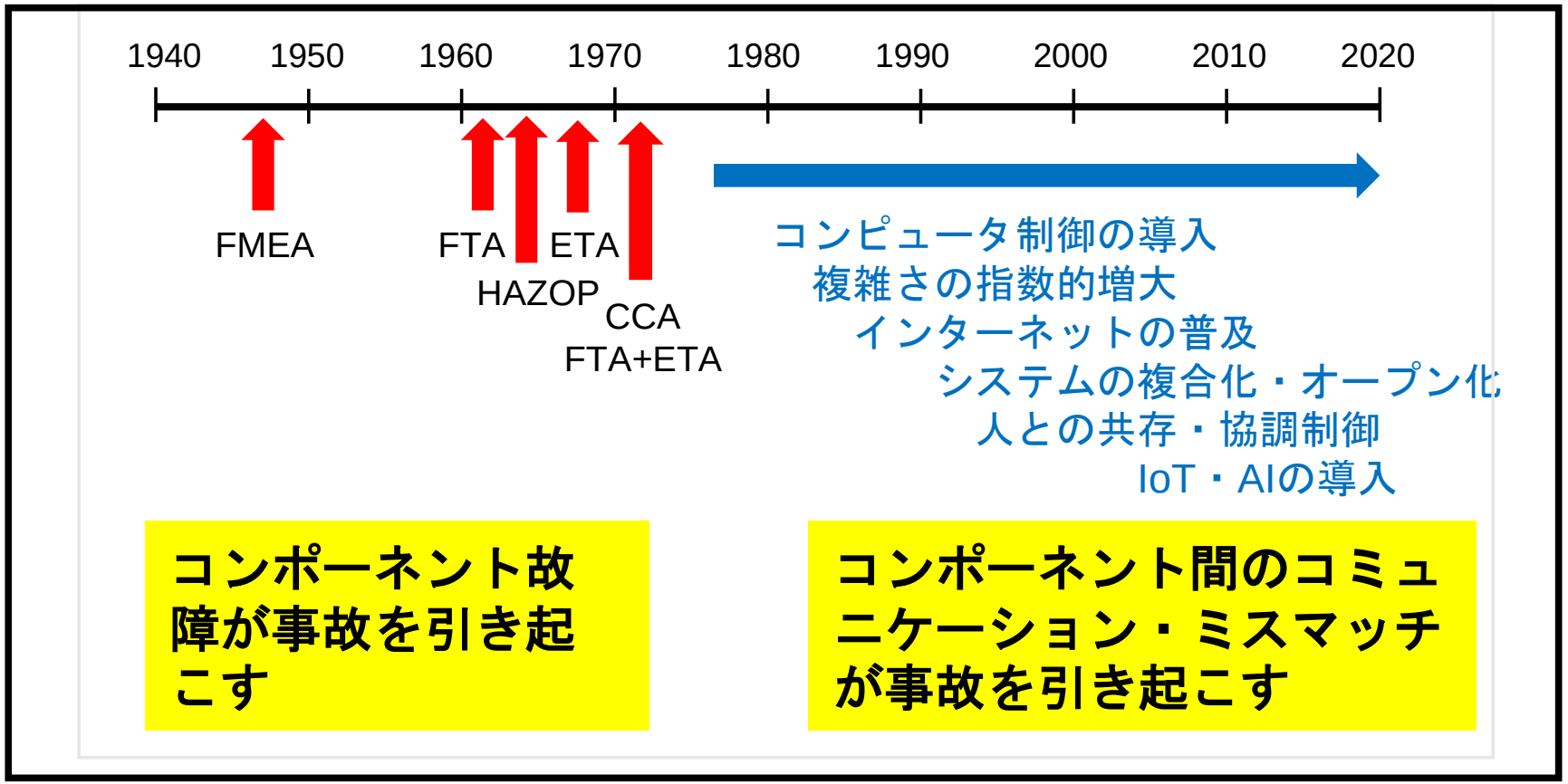
予定
（事例集）

安全設計入門
(JASA、予定)

IoT時代の環境変化と新しい安全解析法　STAMP

Sec-Seminar(2015.6.18) Lecture by Nancy Leveson

現状の安全分析ツールは、40-65年も昔に開発されたものであり、現代の新しい技術の入った複雑な工学システムの安全分析には限界がある　→　STAMPの登場



活動の中で得た格言

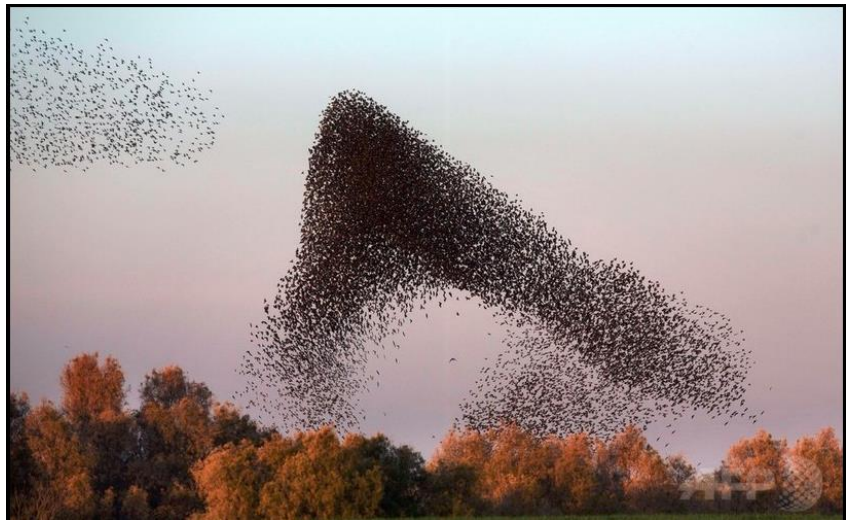
- コミュニケーションエラーが事故を起こす
- 複雑システムの事故は要求仕様の欠陥が原因である
- 安全は創発事象であり還元論だけでは分析できない
- 確率論は意味がない
- 想定外を想定せよ
- 後知恵で責任者を追求しても意味がない
- STAMP/STPAは、従来法（FTA,FMEA,HAZOP)と異なるパラダイムシフトである

活動の中で得た格言→本当に理解できているか？

- コミュニケーションエラーが事故を起
- 複雑システムの事故は要求仕 原因である
- 安全は創発事象であり還 は分析できない
- 確率論は意味がな
- 想定外を想定
- 後知恵で責任 追求しても意味がない
- STAMP/STPAは、従来法（FTA,FMEA,HAZOP)と異なるパラダイムシフトである

創発とは何か？

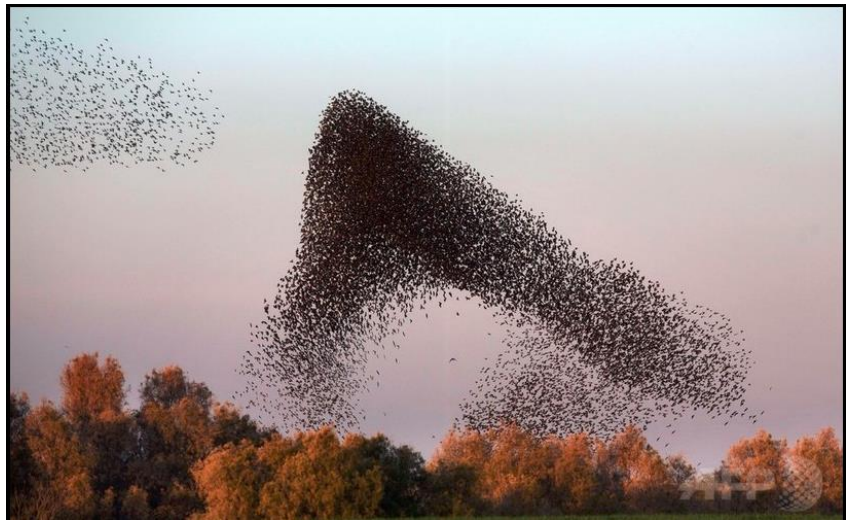
二種類の創発
工学的･社会学的手法で制御できるか？



ムクドリの群れ

チャレンジャー号の爆発事故

二種類の創発
工学的･社会学的手法で制御できるか？



ムクドリの群れ

チャレンジャー号の爆発事故

工学システムとしての分析対象

チャレンジャー号の爆発事故（1986年）

ブースターロケットのジョイント部で使われていたOリングが,低温環境での使用で破壊
→シール効果が不十分となって燃料が漏れ,これに炎が燃え移り爆発
→一見,単純なOリングの部品故障に思えるが、問題の根は深い
→NASAとOリング製作会社は、低温におけるOリングの硬化の問題を予め知っていた
→Oリング製作会社は、当日の気象条件を見て、打ち上げを中止すべきと勧告
→しかし,打ち上げが強行された

→NASA内での現場の技師と管理者との間の意志疎通が不十分
→次年度予算を取るための圧力で,技術上のリスクを低く見積もるという認知バイアス

→複数のコミュニケーションミスと機械の故障,人間や組織の判断ミスなどが複雑に絡まった結果発生した事故といえる
→機械・人・組織の相互作用をモデル化できれば、工学的・社会学的な制御は可能

ニューヨークの地下鉄の犯罪の低減

的外れという批判にも耳を貸さずに,まず地下鉄の落書き清掃作戦からはじめ,さらに,無賃乗車の撲滅にも取り組んだ.その結果,それまでは見過ごされていた軽犯罪で逮捕された人の数は増えたものの,重罪事件は減っていった.

この劇的な成功を支えたのは,犯罪学者のジョージ・ケリングが発案した「割れ窓理論」である.割れたまま放置された窓があっても誰も気にしなければ,まもなく他の窓も割れる.するとその無法状態の雰囲気が町中に伝わり,そこでは何でも許されるという信号を発しはじめ,より深刻な犯罪の呼び水になる.

→システミック理論の有用性を示す。意識下にあるコミュニケーション（相互作用）でも、その影響を明確化すれば、社会学的な安全制御が可能になる

何故、創発と還元論にこだわるか？

- 大規模・複雑システムでは、複数の視点で対象を分析することが必要（システムズエンジニアリングの基本）
- いろいろなネーミングがある！
  - 還元論　vs　創発
  - システマチック　vs　システミック
  - Analysis　vs　Synthesis（分析　vs　合成）
  - 信頼性工学　vs　安全制御工学

システムズ・アプローチ
～システミックとシステマチックの両面からの分析の大切さ～

「システミックxシステマチック」な問題解決を


SDM NEWS

2013年 1 月号

行事予定

2013年2月6日(水) 19:00～21:00
SDM研究所・GCOE共催公開講座
「ダイアログとデザインの未来 Vol.8 経済の未来」
@日吉キャンパス 協生館C3S10教室
http://www.sdm.keio.ac.jp/2013/02/06-132333.html
要事前登録 無料

2013年3月4日(月) 13:00～17:10
第5回環境共生・安全システムデザインシンポジウム「脳と心と幸福を考える」
@日吉キャンパス協生館 藤原洋記念ホール
http://www.sdm.keio.ac.jp/2013/03/04-164319.html
要事前登録 無料

研究科委員長兼研究所長からのメッセージ

「システミック×システマティック」な問題解決を

明けましておめでとうございます。今年のSDMのキャッチフレーズは"システム"。昨年は主にデザインという単語にフィーチャーした一年だったと言えるかもしれません。今年は初心に帰り、"システム"にフォーカスします。システムとは、要素間の[redacted]性によって創発する特徴を持つもの。技術システム[redacted]ステム、人間システムまで、インタラクション[redacted]は何でもシステムです。慶應SDMで重視[redacted]は、システムズエンジニアリングを基盤とするシステムズ・アプローチ。システムズ・アプローチには、ふたつの意味があります。システミックなアプローチと、システマティックなアプローチ。前者は、ものごとをシステムとして俯瞰的・全体的に見るということ。後者はロジカルに分解・統合するということ。慶應SDMでは両者を駆使してイノベーティブなデザインとサステナブルなマネジメントを実現します。システムズ・アプローチを身に着けた人材の育成、SDM学の基礎と応用に関する実践的な研究、そして企業や事業体との連携に基づく様々な階層での問題解決・社会変革。今年度もよりよい世界を構築するために邁進してゆく

慶応大学　SDMニュース
http://www.sdm.keio.ac.jp/pdf/sdmnews/SDM_News_201301.pdf

<u>システミック・アプローチ</u>
ものごとをシステムとして俯瞰的・全体的に見る
<u>システマチック・アプローチ</u>
ものごとをロジカルに分解・統合して見る

（参考）システムズエンジニアリング
「システムの実現を成功させることができる複数の専門分野にまたがるアプローチおよび手段 」（INCOSE SE Handbook,2000)

ソフトウェア集約システムの安全設計でのパラダイムシフト

- 従来の安全設計法（FTA、FMEA、HAZOPなど）は還元主義（Reductionism）である（N.G.Leveson)
  - 安全目標を要素に分解し、全ての故障を導出・低減する信頼性工学手法
  - システマチックアプローチに対応するが、ソフトウェア集約システムでは、全ての欠陥を見つけて除去することは困難でもある
- システム理論に基づく事故モデルと安全分析法STAMP/STPAは、トップダウンで安全制御構造を可視化して事故を防ぐ
  - システミックアプローチに対応し、事故シナリオを創発（Emergence)と捉える
  - 複雑システムを、抽象化と階層化した安全制御構造モデルで理解し、故障を低減するのではなく、事故を防ぐ方策を分析する（安全制御工学）
  - 仕様の欠陥や要素間の連係ミス（コミュニケーションエラー）により起こされる事故シナリオを抽出することが主眼であり、還元主義的手法と異なるパラダイムシフトでもある

活動の中で得た格言→本当に理解できているか？

- コミュニケーションエラーが事故を起こす
- 複雑システムの事故は要求仕様の欠陥が原因である
- 安全は創発事象であり還元論だけでは分析できない
- 確率論は意味がない
- 想定外を想定せよ
- 後知恵で責任者を追求しても意味がない
- STAMP/STPAは、従来法（FTA,FMEA,HAZOP)と異なるパラダイムシフトである

活動の中で得た**格言**→本当に理解できているか？

- コミュニケーションエラーが事故を起こす
- 複雑システムの事故は要求仕様の欠[illegible]因である
- 安全は創発事象であり還元論だけ[illegible]析できない
- 確率論は意味がない
- 想定外を想定せよ
- 後知恵で責任者を[illegible]も意味がない
- STAMP/STPAは[illegible]（FTA,FMEA,HAZOP)と異なるパラダイムシフトである

複雑システムを理解するには？

現代の工学システム

自動運転車、生活支援ロボット、列車、航空機、船舶、プラント・・

- ほとんどの便利な機能はソフトウェアが担っている
- 付随する安全機能もソフトウェアが担っている

ナンシー・レバソンは、これを、
「ソフトウェア集約システム（Software-Intensive system）」
と呼んでいる。　→これは複雑システムでもある。

より便利な機能を追い求めて、人の支援・代替機能や、
外部環境との協調・適応制御を十分な検証なしに実装
しがちになる（完全な検証ができない場合もある）。
→事故が起こった後、想定外との言い訳をしてしまう

ソフトウェア集約システム（Software-intensive system）の課題
～ソフトウェアは好むと好まざるとにかかわらず複雑化する～

- ソフトウェアコンポーネントとコンポーネント間通信は、市場競争の産業界では必然的に複雑化する
  - ソフトウェアは、人が行う機能を補助ないし代替しようとするが、エラーも含めて代替してしまう
  - ソフトウェアは、環境の状態を人間をまねて認識しようとするが、同等の認識はできない
  - ソフトウェアコンポーネントは、将来の拡張性やトラブル時の事後確認などを想定して、仕様にない機能も含めてしまう
  - ソフトウェアコンポーネント間、人・環境とソフトウェアコンポーネント間の通信データは、送る側の意図と異なる意図で受け取り側が利用することがある
- ソフトウェアエンジニアは、システムエンジニアとしての役割も期待される

問題点はわかったが、複雑システムの安全設計はどうする？

- STAMPによる複雑さの克服
  - システムズ理論や認知科学からの教訓によると、人間の心は、抽象化と階層化によって複雑さを理解(manage)する（N.G.Leveson)
  - トップダウンプロセスにより、高いレベルの抽象度から始めて、詳細モデルへ掘り下げる（階層化したシステムモデルの構築）
- 「概念の整合性（完全性）」は製品の品質保証やシステム設計のために最も重要である（F.Brooks)
  - トップダウンプロセスでのシステム設計が重要

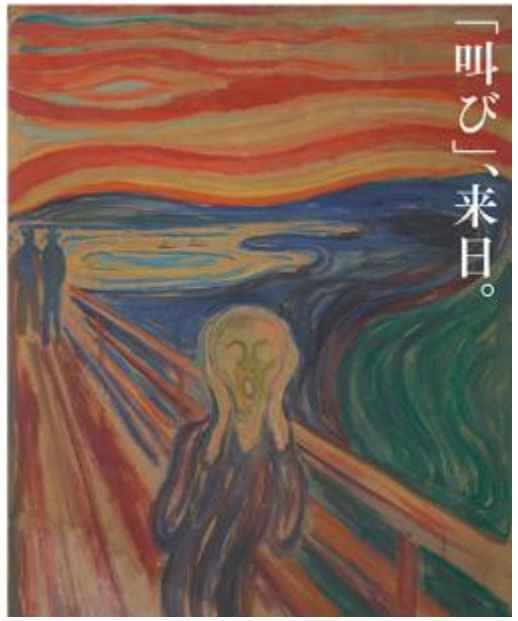
閑話休題

安全とは何か？

2018.11.13　Elick Hollnagel　IPA　FRAM勉強会

A need to be safe

A need to guard against a repeat occurrence



A need to feel safe

A need to find an explanation for what happened

絶対安全はないので
確率論で評価する

確率論に乗らない事象がある

後知恵で説明しても事前の予測には役立たない

想定外との言い訳をする

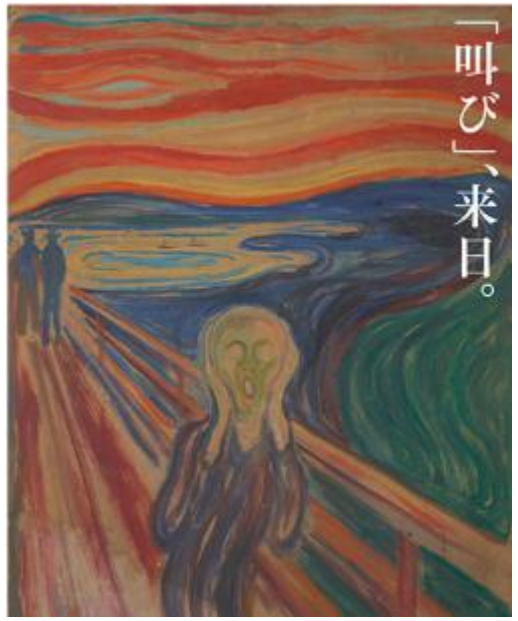
閑話休題

安全とは何か？

2018.11.13　Elick Hollnagel　IPA　FRAM勉強会

A need to be safe

A need to guard against a repeat occurrence



A need to feel safe

A need to find an explanation for what happened

絶対安全はないので
確率論で評価する

確率論に乗らない事象がある

後知恵で説明しても事前の予測には役立たない

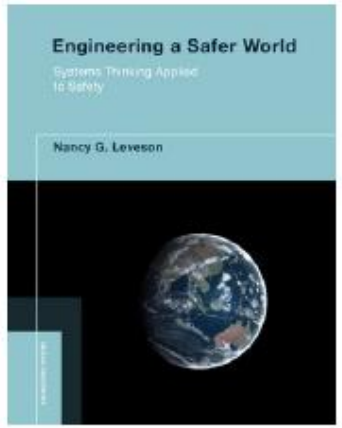
想定外との言い訳をする

新しい製品、新しい作業の安全評価には過去トラは役立たない
では、どうやって安全を論証するか？　ー＞STAMP

STAMP/STPAをもう一度振り返ってみよう

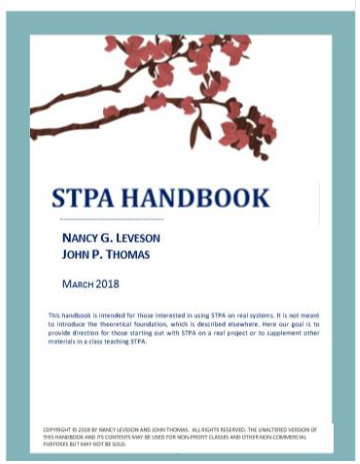
- 背景
- 手順
- 事例

三つの参考資料をもう一度読み直してみた



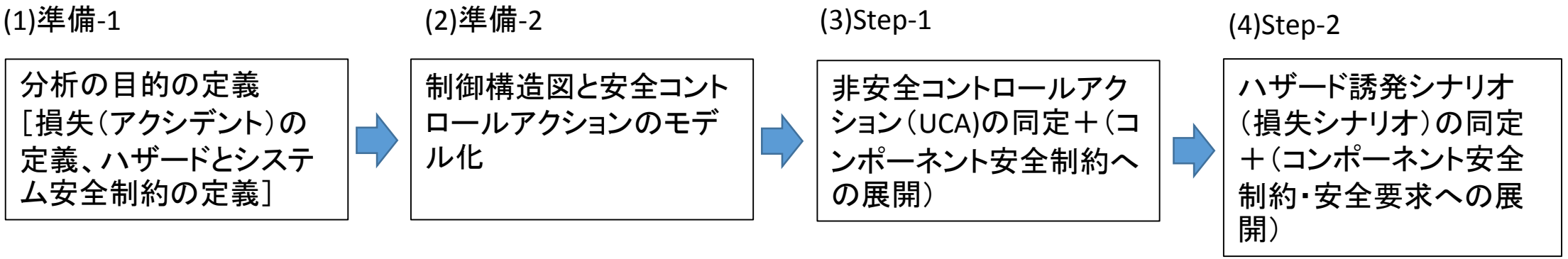
An STPA Primer
Version 1, August 2013
(updated June 2015)

http://psas.scripts.mit.edu/home/wp-content/uploads/2015/06/STPA-Primer-v1.pdf



http://psas.scripts.mit.edu/home/

STPA手順（Handbook）



(1)表面的には極めて簡単な手順で、4つの非安全コントロールアクション(UCA)が特徴的に見える。
(2)ハザード誘発シナリオは、従来のFTA,FMEAの思考法と大差ない。
しかし、
(3)アクシデント、ハザードの決め方、制御構造図の作り方などで、明示化されていないノウハウがある。
（これを理解するにはシステムズ理論の考え方の理解が重要である）

アクシデント、ハザード、安全制約の例

システム	損失（アクシデント）	ハザード	安全制約
ACC(自動追従運転)	L1:2台の車の衝突	H1:前方ないし後方の車との不適切な車間距離	SC1:二つの車は最小の車間距離を越えてはならない
化学プラント	L1:有害物質による人命の損失または危害	H1:プラントからの有害物質の気中や地中への放出	SC1:有害物質はプラントから過失によって放出されてはならない
列車のドア開閉システム	L1:乗客の列車からの転落	H1:ドアの間に人がいるときに閉まる H2:ドアが列車が動いているとき、または、プラットフォーム外で開く、 H3:ドアが緊急時（火災など）に開かない、	SC1:ドアの間に人がいるときにドアを閉めてはいけない SC2:ドアが開いているとき列車は出発してはならない、 SC3:列車が動いているときドアが開いてはならない、 SC4：緊急時にはドアが開いていないといけない
自動車のエアバッグ	A1:運転者の死傷	H1:衝突したのにエアバッグが開かない H2:通常走行時にエアバッグが開いてしまう H3:エアバックの異常な爆発（部品飛散）	SC1:衝突時にはエアバックが開く SC2:通常走行時にエアバッグは開かない SC3:エアバック開の際に部品を飛散させない

アクシデント、ハザード、安全制約の例

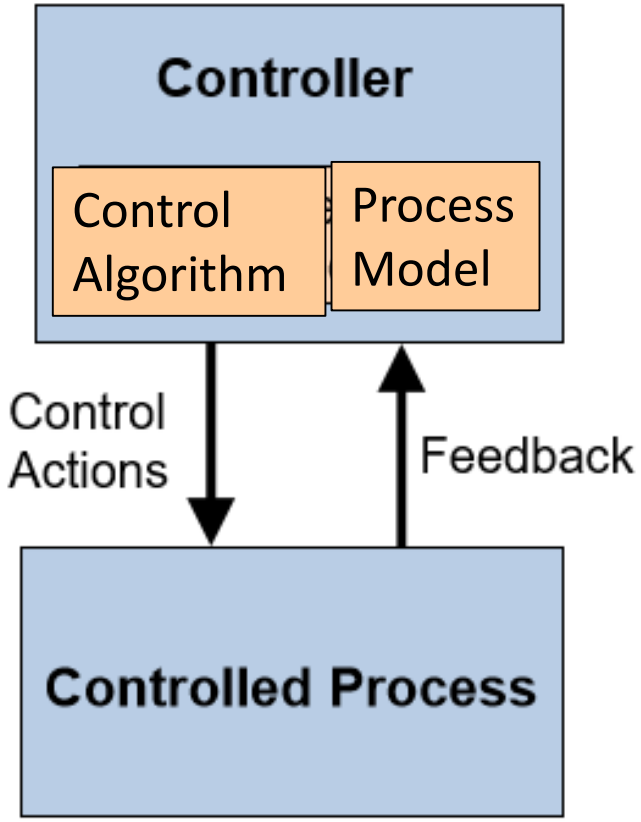
システム	損失（アクシデント）	ハザード	安全制約
ACC(自動追従運転)	L1:2台の車の衝突	H1:前方ないし後方の車との不適切な車間距離	SC1:二つの車は最小の車間距離を越えてはならない
化学プラント	L1:有害物質による	H1:プラントからの有害物質の	SC1:有害物質はプラントから過失に
	または		
	車から		
自動車のエアバッグ	A1:運転者の死傷		
		H3:エアバックの異常な爆発（部品飛散）	させない

損失（Loss）は、人命損失、環境汚染、経済的損失、ビジネスリスクなど何でもよい。

ハザードは、事象（イベント）や危険源ではなく、事故の一歩手前の放置してはいけない状態と考える。

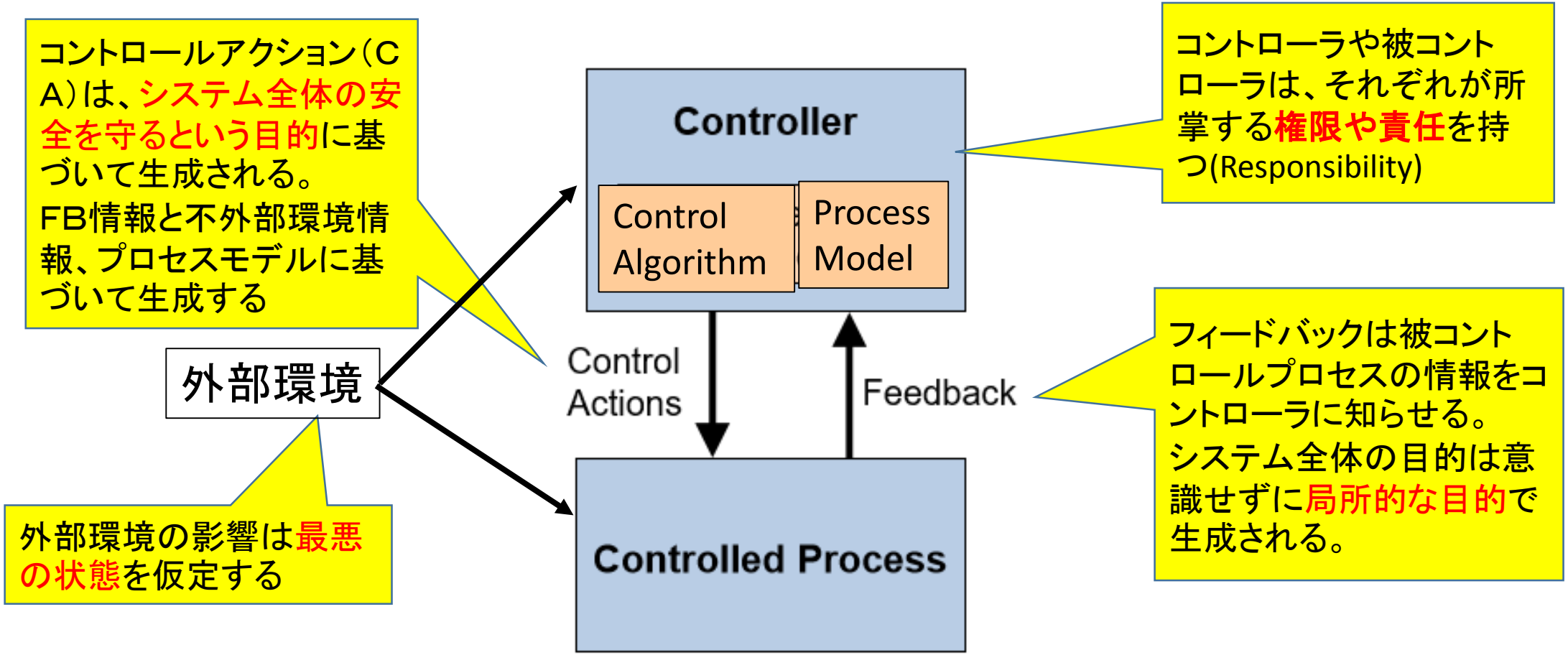
・システム外部の状態は制御できないので最悪状態を仮定する
・ハザードを誘発する要因まで言及すると発想を狭める（例：ブレーキ故障ではなく、加速・減速という表現）
・ハザードは、システム全体を見て10個程度以内に抑える。これが多すぎるのはシステムの理解の抽象度が足りないことを意味する

Basic STAMPの制御構造図



- コントローラは、その所掌責任・権限に基づいて、コントロールアクション(CA)によりシステムの安全を確保する。CAは、フィードバック情報とプロセスモデルに基づいて決められる。
- プロセスモデルが正しくないときに想定外の挙動が起こる(プロセスモデルは、被コントロールプロセスの挙動についての分析者のメンタルモデルでもある)
- 四つの不適切なコントロールアクション(UCA)
  - (N) Not providing
  - (P) Providing causes hazard
  - (T) Inadequate timing, too early or too late
  - (D) Inadequate duration, stop too soon or applying too long
- ソフトウェアと人間の挙動のモデルにより、ソフトウェアエラー、ヒューマンエラー、相互作用による事故などを説明する

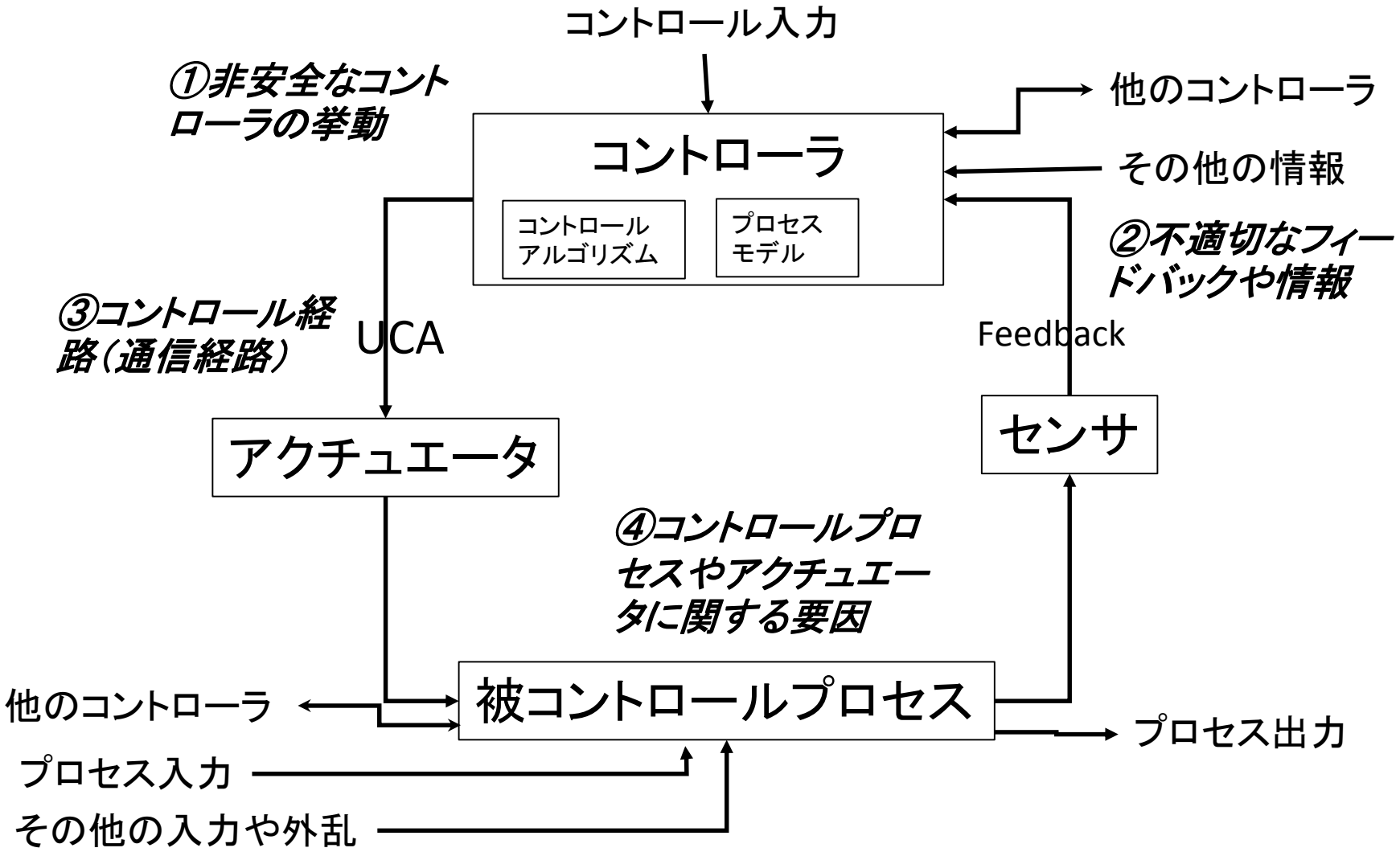
Basic STAMPの制御構造図（四つの大事な考え方）



一枚の制御構造図で、システム全体の安全制御メカニズムが見通せないといけない

→　立場の異なる人の間で共通の理解に基づいて相互レビューができる

ハザード誘発シナリオ


コントロール入力
①非安全なコントローラの挙動
他のコントローラ
コントローラ
その他の情報
コントロールアルゴリズム
プロセスモデル
②不適切なフィードバックや情報
③コントロール経路（通信経路）
UCA
Feedback
アクチュエータ
センサ
④コントロールプロセスやアクチュエータに関する要因
他のコントローラ
被コントロールプロセス
プロセス出力
プロセス入力
その他の入力や外乱

ハザード誘発シナリオ

ハザード誘発要因(HCF)をチェックシートやガイドワードとして使うのは間違い！（自由な発想を妨げてしまう）

本日の講演で訴えたいこと！

- 安全を確保するための制御構造図を作ることで、対象システムのシステミックな理解をし、それをステークホルダー間で共有することが最も大事である
  - 新しい製品開発で想定外のトラブルを減らすことに役立つ
  - 安全論証として最善の努力をしていることのエビデンスになり得る

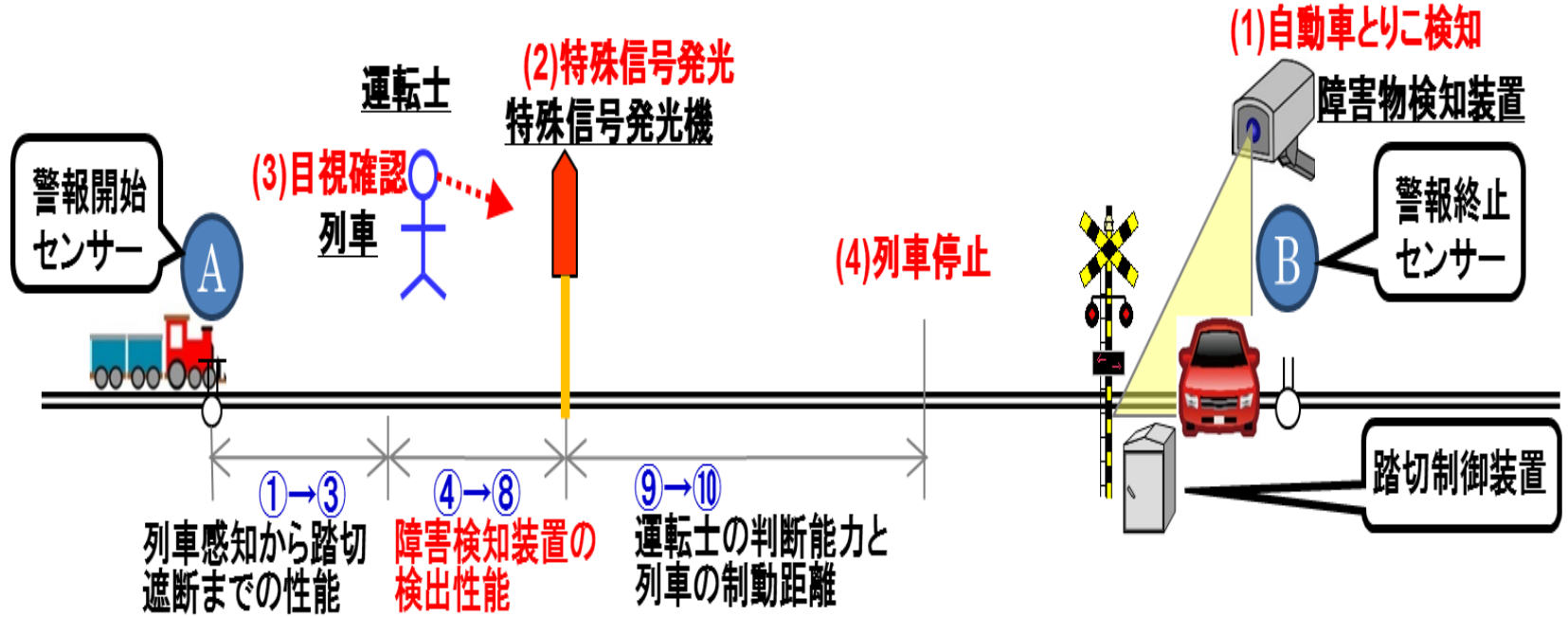
STAMP/STPAの事例を振り返ってみよう

- 背景
- 手順
- 事例

WGの活動の中での事例を振り返る

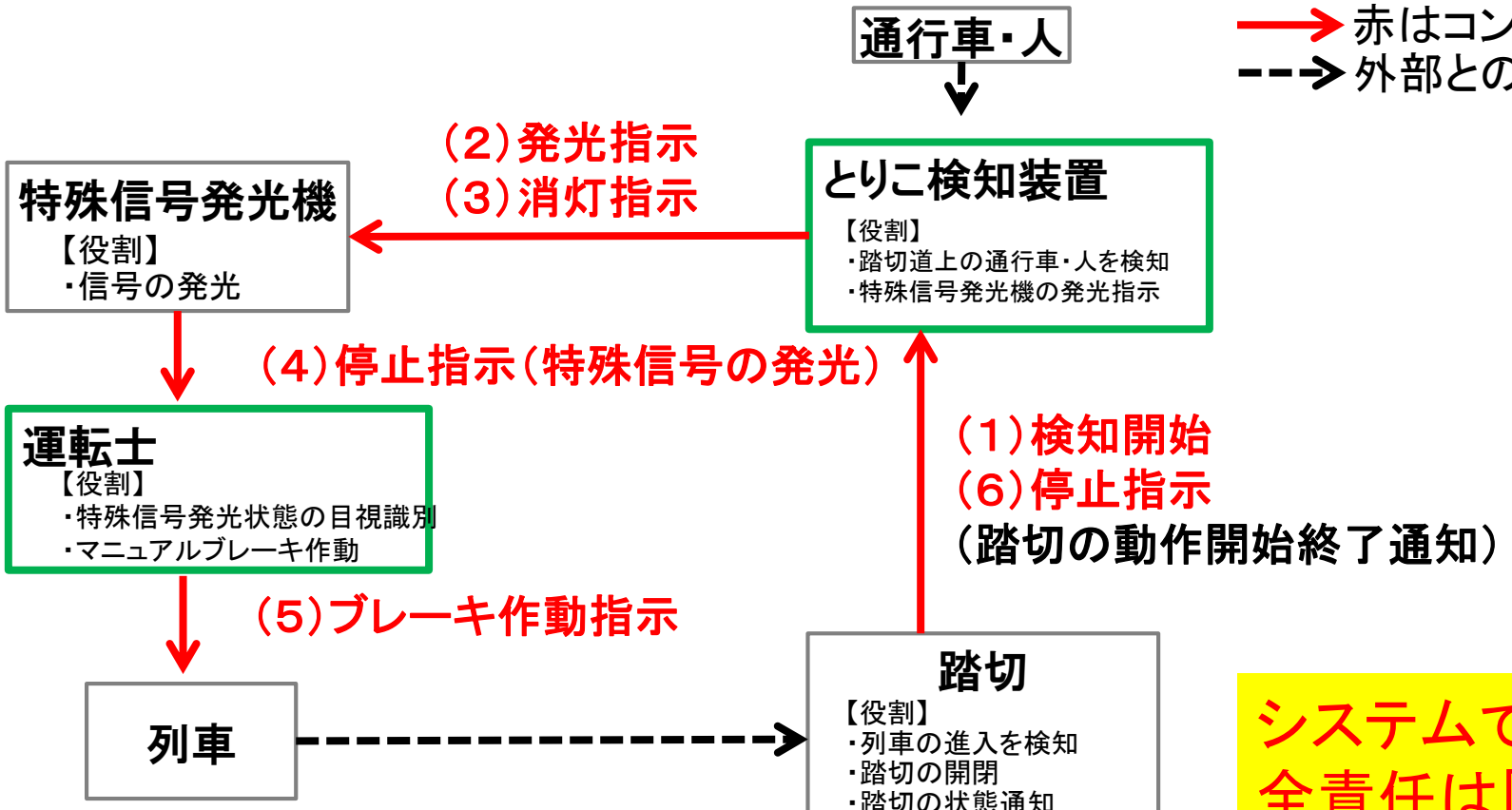
システムズ思考の事例：踏切の「とりこ」検知装置

踏切が閉まった時、踏切内に閉じ込められた人や車を検知し、特殊信号発光機で列車運転士に知らせるシステム



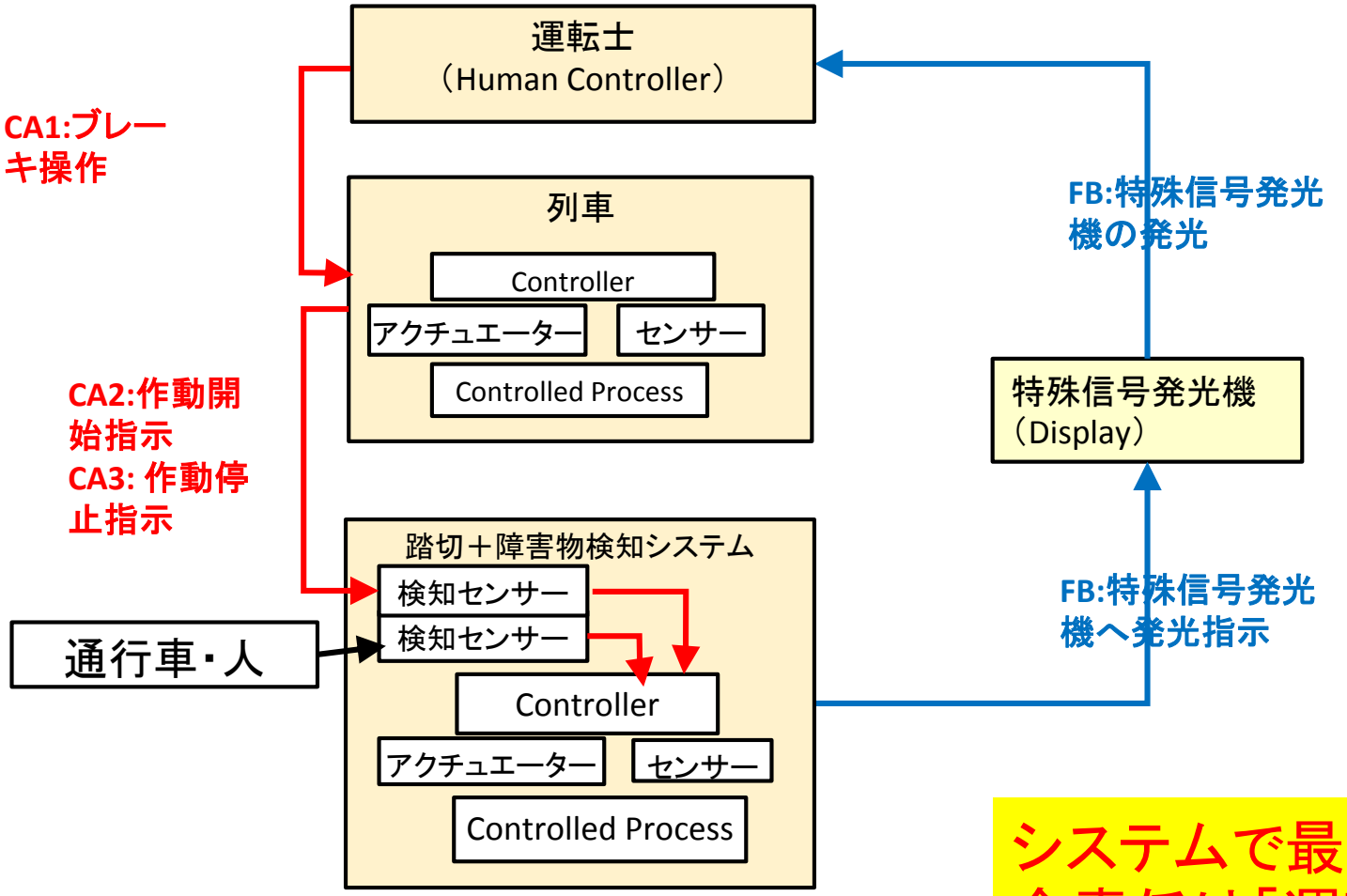
今回の分析範囲は④から⑨

検知の流れに沿った制御構造図（ハードウェア視点）



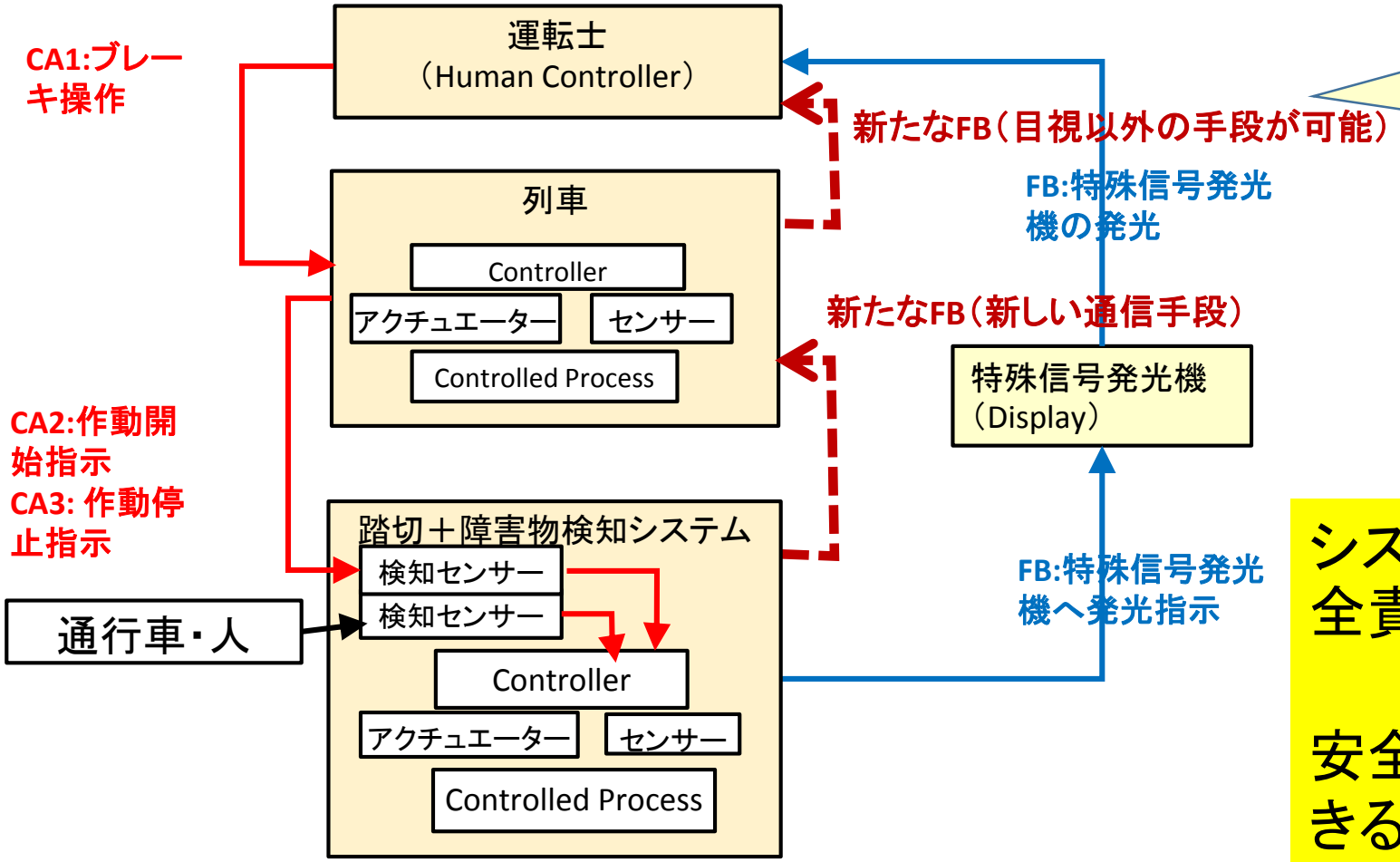
システムで最も大事な安全責任は「とりこ検知装置」と見える
（従来の技術者視点）

運転士を中心にした制御構造図（運転士視点）



システムで最も大事な安全責任は「運転士」である

運転士を中心にした制御構造図（運転士視点）

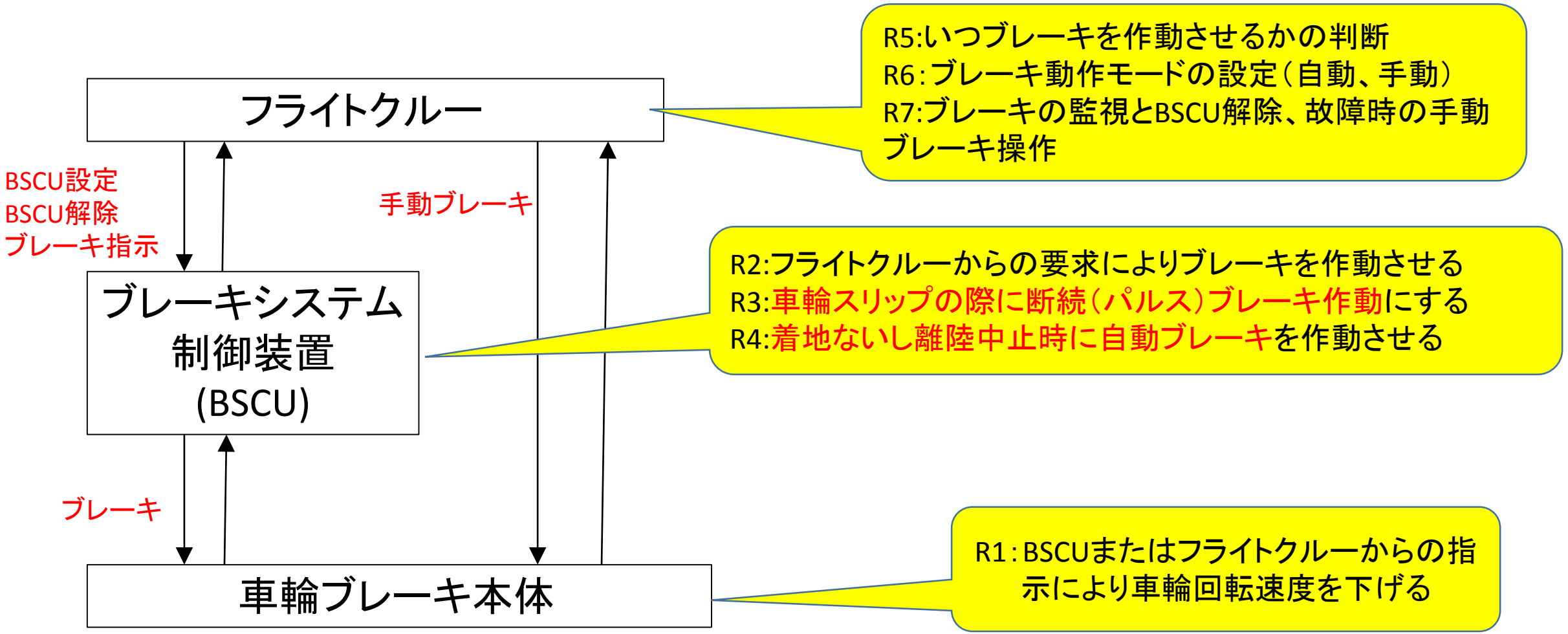


信号目視以外に、音声でも伝えられる。踏切の映像も伝えることが出来る。

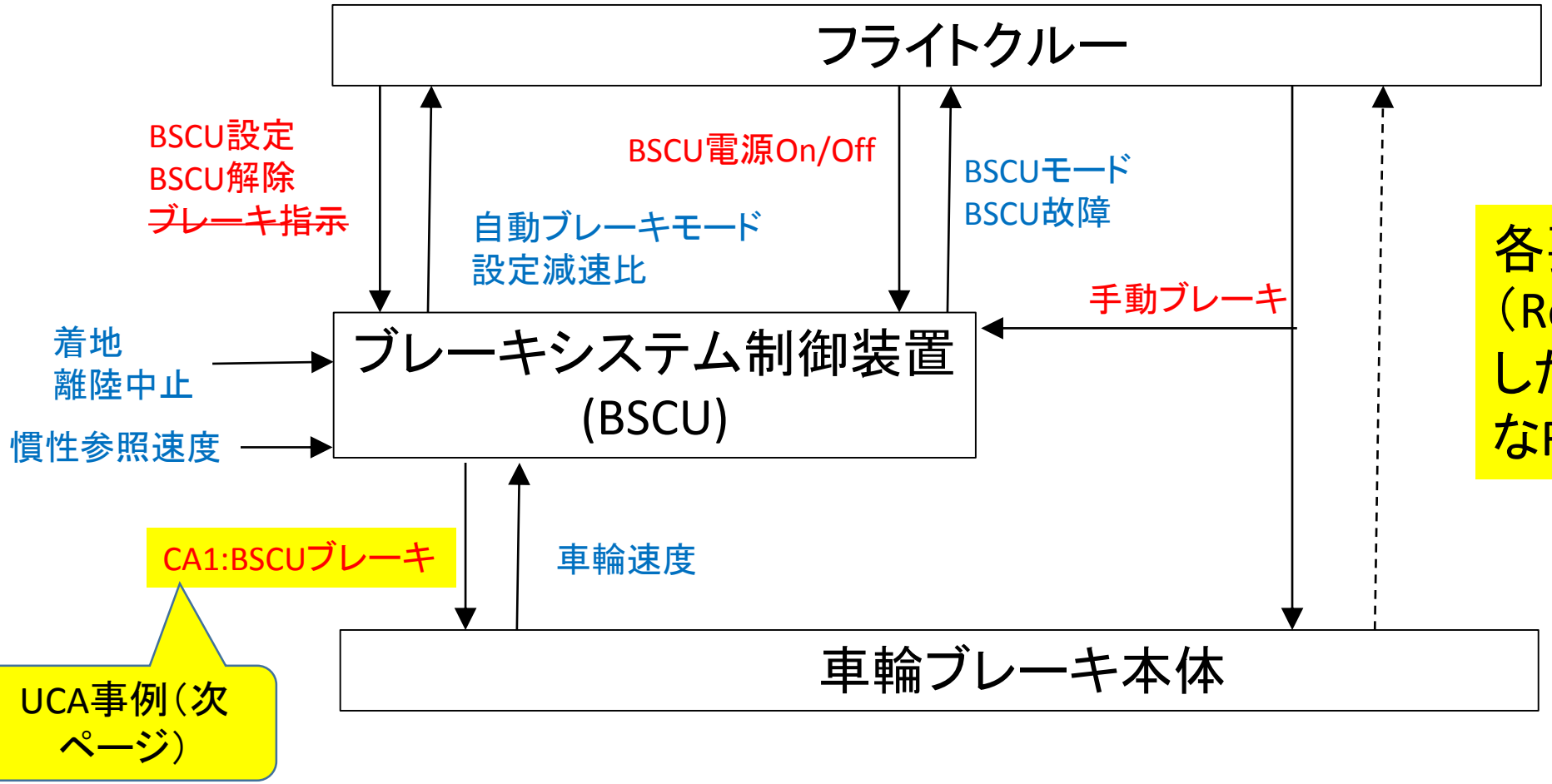
システムで最も大事な安全責任は「運転士」である

安全要求の改善が発想できる（想定外の環境変化への対応）

システムズ思考の事例：航空機の車輪自動ブレーキシステム



航空機の車輪自動ブレーキシステムの制御構造図（フィードバック情報の具体化）



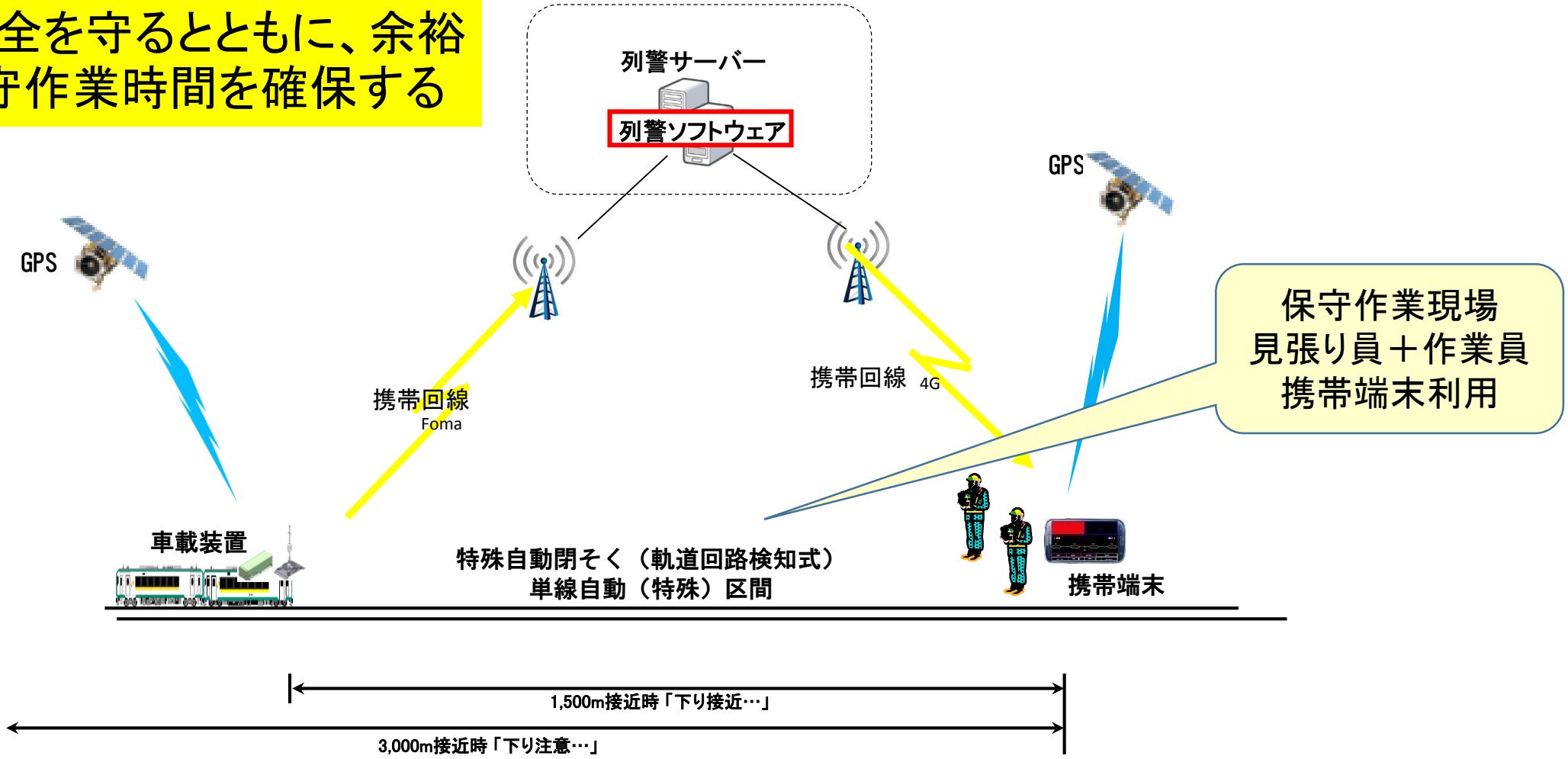
各要素の責任（Responsibility)を考慮したCAの具体化と必要なFB情報の抽出

航空機ブレーキ操作時のUCAの例

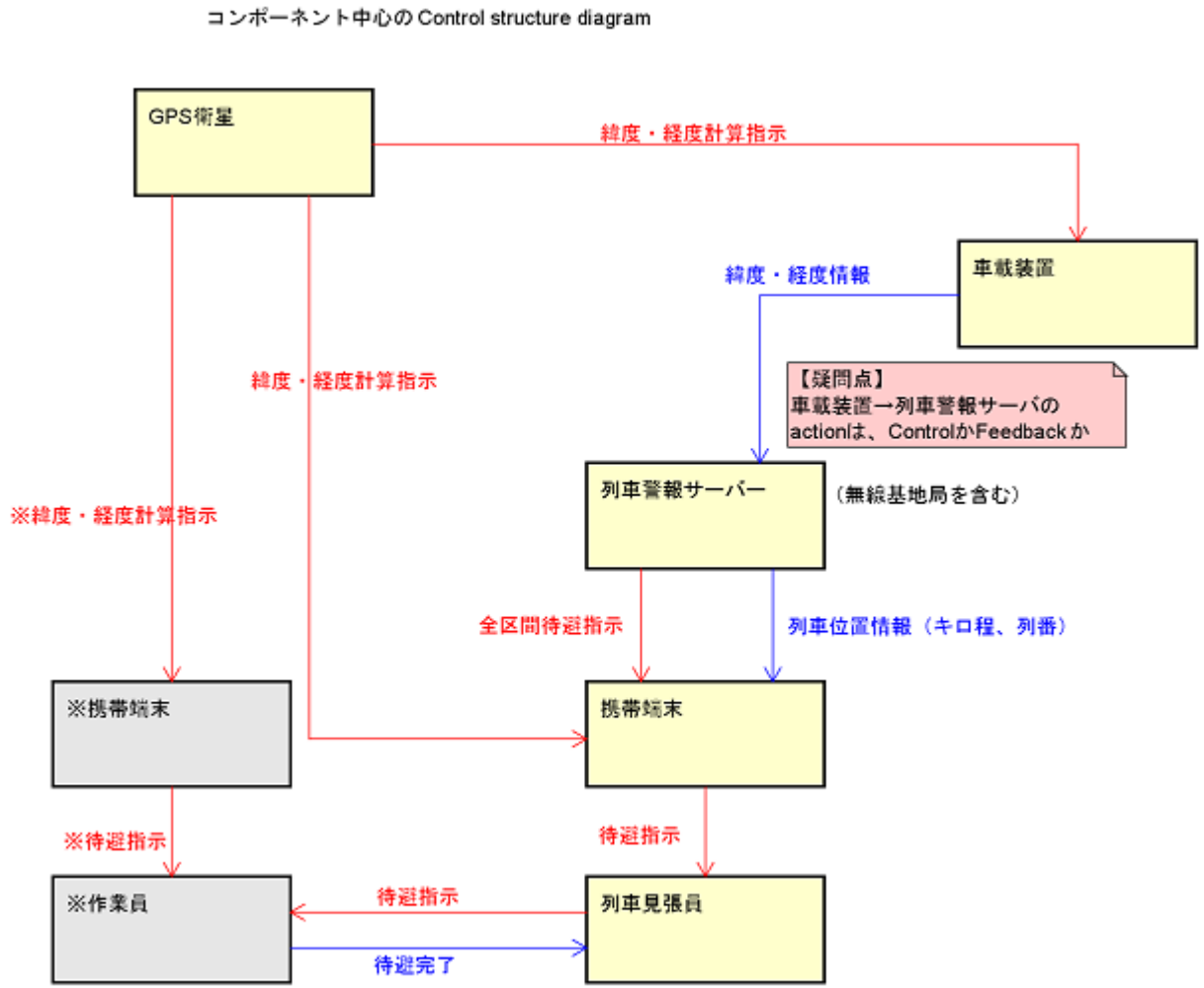
コントロールアクション	与えられないとハザード(N)	与えられるとハザード(P)	早すぎ、遅すぎ、誤順序(T)	早すぎる停止、長すぎる運用(D)
CA1:BSCUブレーキ	UCA1-N-1： 着陸滑走中BSCU が作動している時、BSCU自動ブレーキがブレーキコントロールを出さない[H-1]	UCA1-P-1： 通常離陸中にBSCU自動ブレーキがブレーキコントロールアクションを出す[H-3] UCA1-P-2：着陸滑走中にBSCU自動ブレーキが、不十分なブレーキレベルでブレーキコントロールアクションを出す [H-1] UCA1-P-3： 着陸滑走中にBSCU自動ブレーキが、方向性のある、または非対称のブレーキとなるブレーキコントロールアクションを出す [H-1, H-2]	UCA1-T-1： 着陸後BSCU自動ブレーキがブレーキコントロールアクションを出すのが遅過ぎる(>TBD 秒) [H-1]	UCA1-D-1： 着陸時にBSCU自動ブレーキが、ブレーキコントロールアクションを止めるのが早過ぎる(駐機速度TBDkm/Hrに達する前) [H-1]

システムズ思考の事例：GPS列警システム

目的
GPSによるより正確な位置情報で、作業員の安全を守るとともに、余裕を持った保守作業時間を確保する



GPS列警システム（コンポーネント中心の制御構造図）

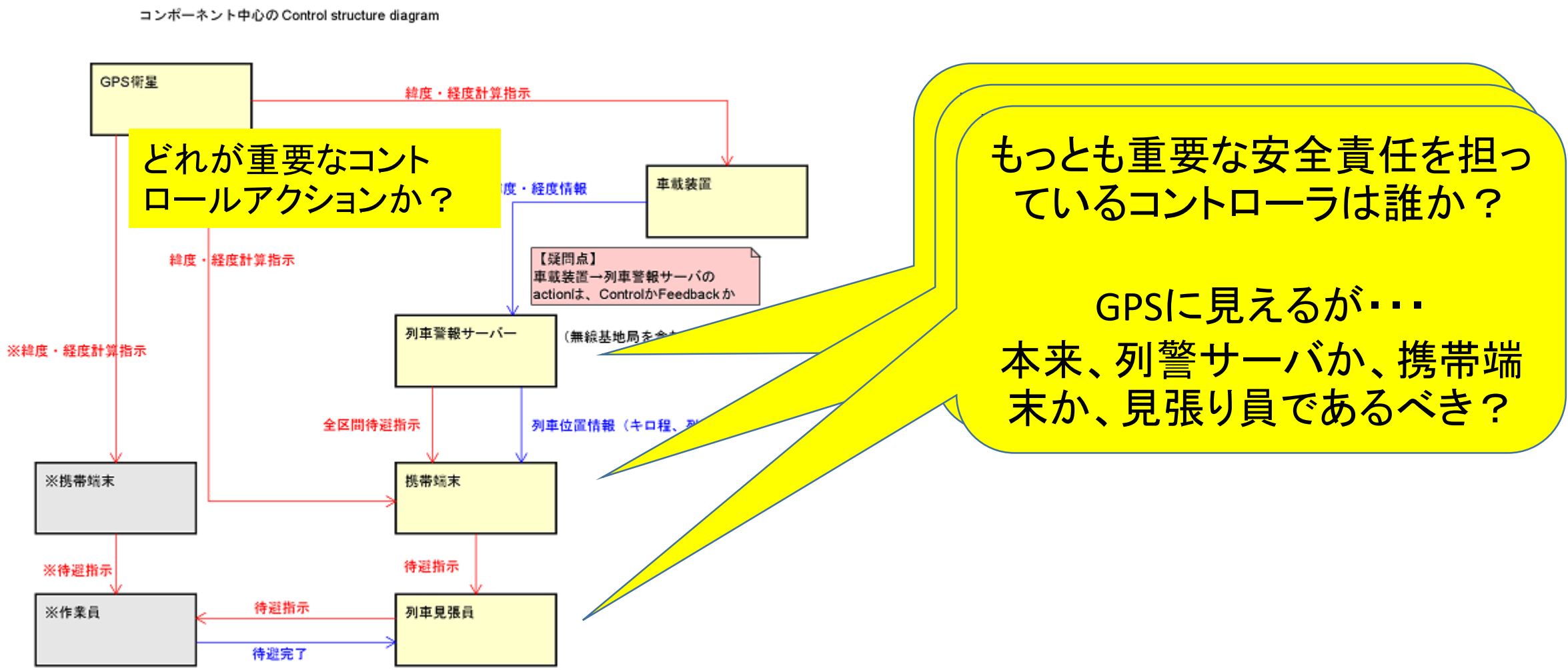


ハードウェアに沿ったブロック図になっている（機能の抽象化が十分でない）。

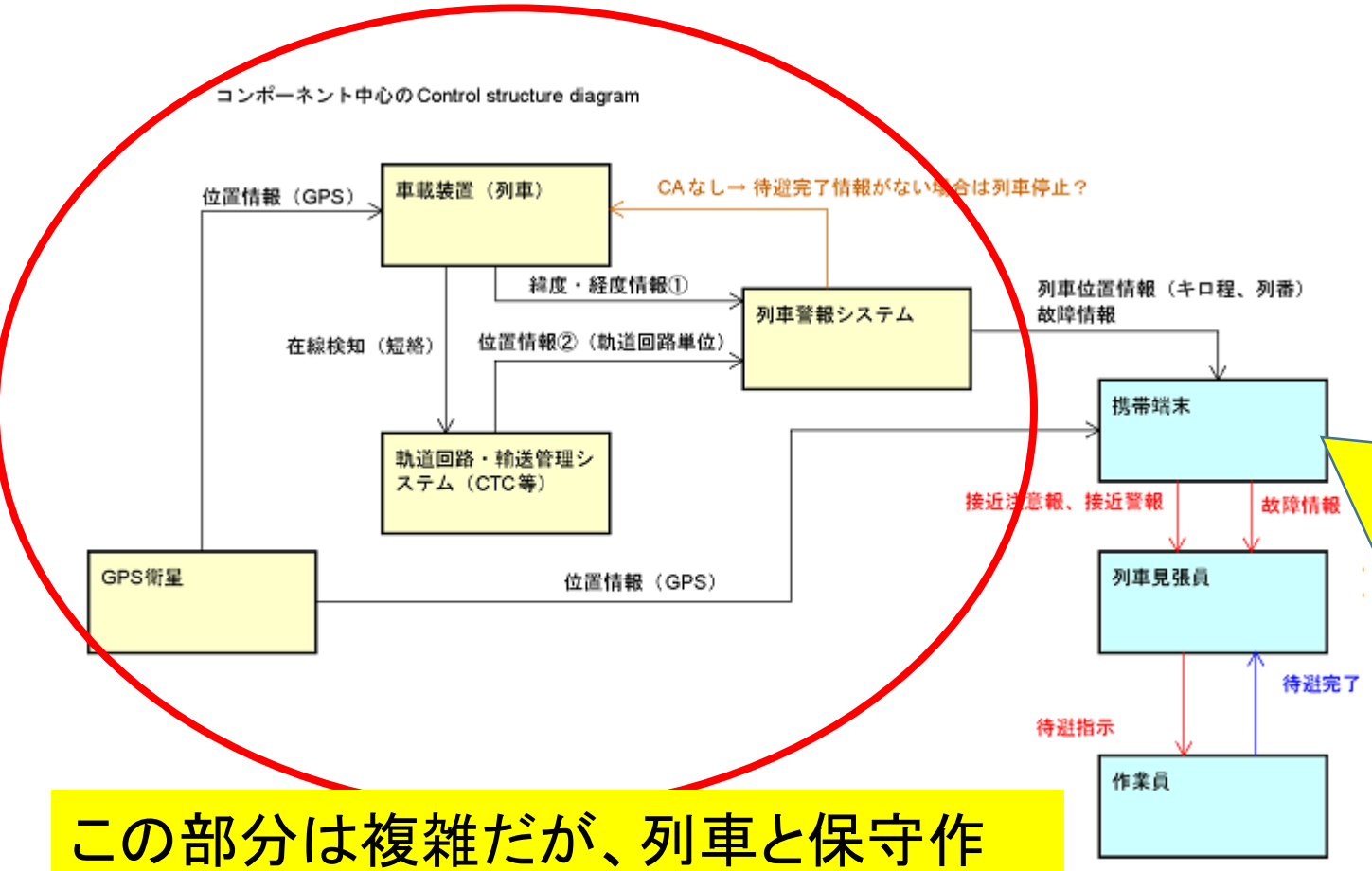
CAとFBの区別が不十分。

主なコントローラ（安全責任）は誰か。

GPS列警システム（コンポーネント中心の制御構造図）



GPS列警システム（携帯端末中心の制御構造図）



この部分は複雑だが、列車と保守作業の位置を計測するだけの役割

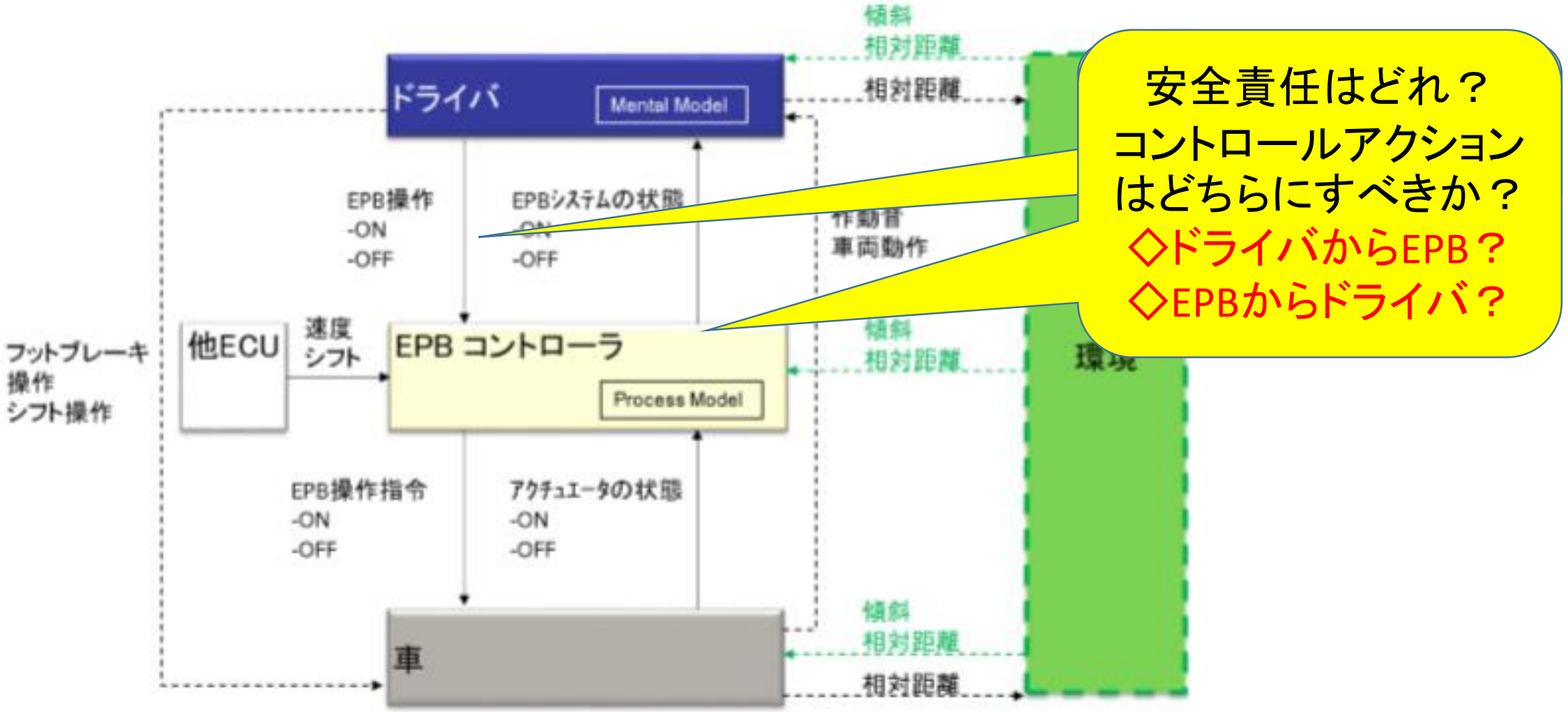
もっとも重要な安全責任を担っているコントローラは誰か？

携帯端末とみなすと安全責任がわかり易くなるが・・・

保守作業の時間は多くとれそうだが、故障によるリスクは増えそう

STAMP/STPAで考えてみよう！

システムズ思考の事例：電動パーキングブレーキシステム（EPB）



STPA手順の再考

(1)準備-1

分析の目的の定義
[損失（アクシデント）の定義、ハザードとシステム安全制約の定義]

(2)準備-2

制御構造図と安全コントロールアクションのモデル化

(3)Step-1

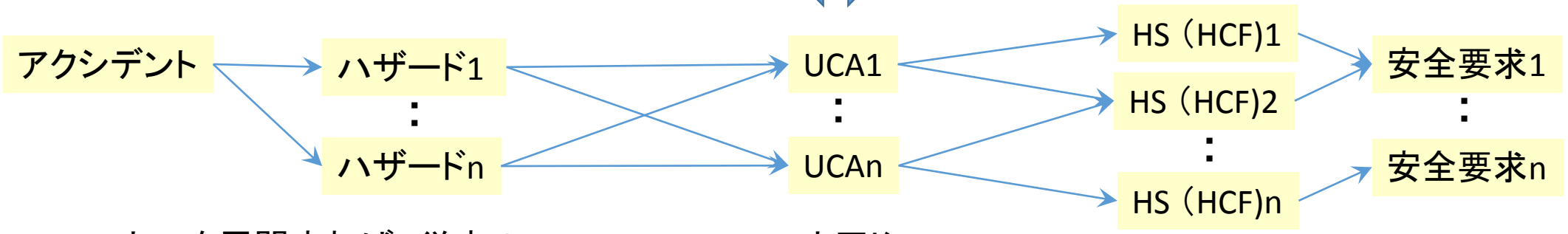
非安全コントロールアクション（UCA)の同定＋（コンポーネント安全制約への展開）

(4)Step-2

ハザード誘発シナリオ（損失シナリオ）の同定＋（コンポーネント安全制約・安全要求への展開）

HSに関わらずUCAを回避する（安全制御工学、レジリエンス工学）

HS(HCF)を全て評価し回避するのが信頼性工学



UCAとHSを展開すれば、従来のFTA、FMEA、HAZOPと同じ

後知恵で考えると、どんな手法で考えても同じ、だが、
事故前に発想できるかどうかの違いがある（プロアクティブな安全設計）

STPA手順の再考

制御構造図による安全メカニズムの可視化と共有
- 目的（どんな事故を防ぐか）
- 各要素の安全責任と権限を定義
- 安全責任を達成するためのCAと、CAを作るためのFBを明示化
- 外部環境は最悪の状態を仮定

(1)準備-1
分析の目的の定義
［損失（アクシデント）の定義、ハザードとシステム安全制約の定義］

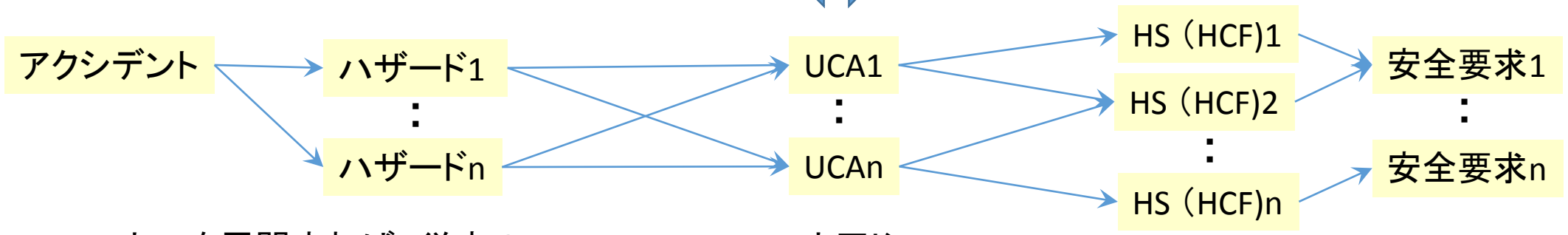
(2)準備-2
制御構造図と安全コントロールアクションのモデル化

(3)Step-1
非安全コントロールアクション（UCA)の同定＋（コンポーネント安全制約への展開）

(4)Step-2
ハザード誘発シナリオ（損失シナリオ）の同定＋（コンポーネント安全制約・安全要求への展開）

HSに関わらずUCAを回避する（安全制御工学、レジリエンス工学）

HS(HCF)を全て評価し回避するのが信頼性工学



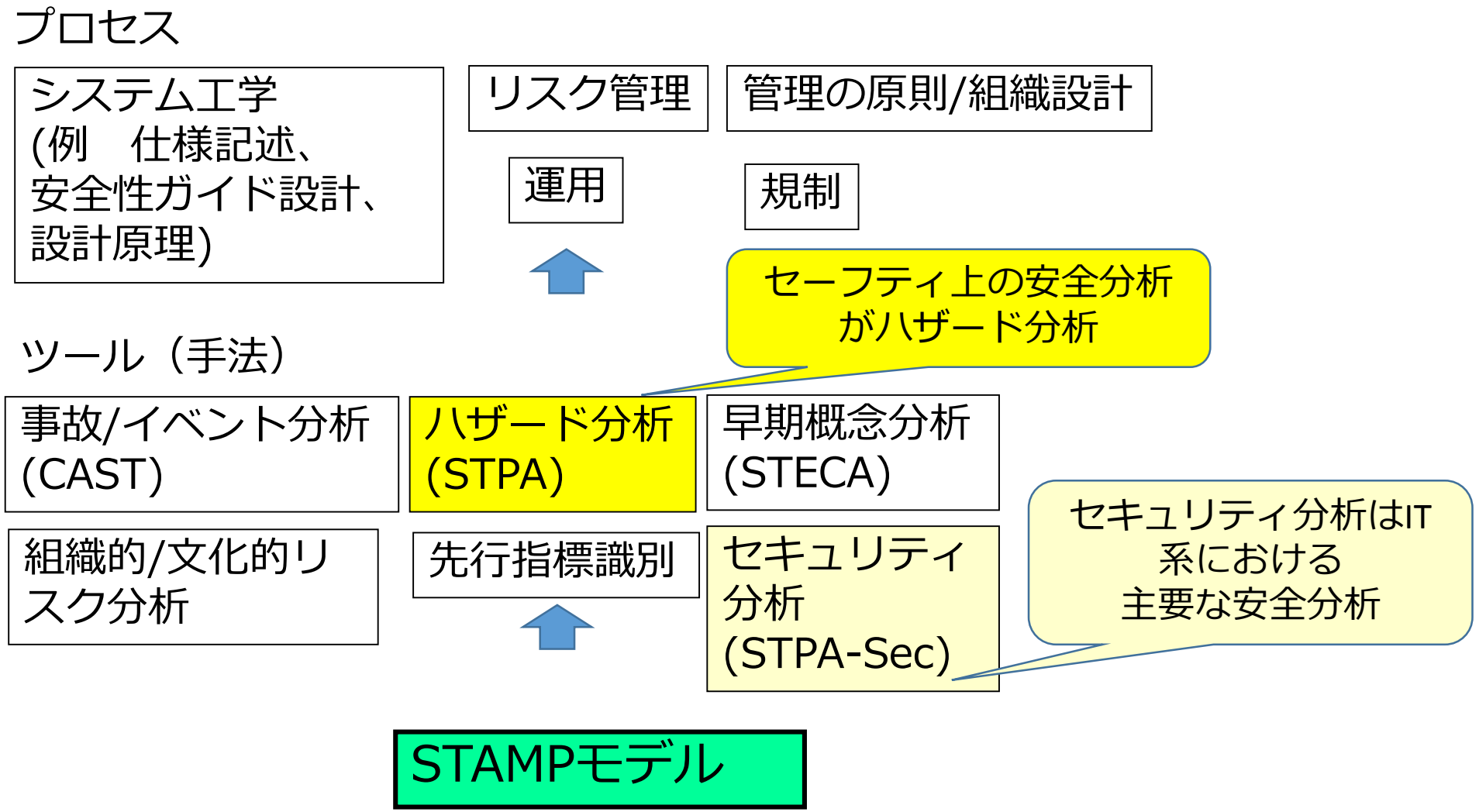
UCAとHSを展開すれば、従来のFTA、FMEA、HAZOPと同じ

後知恵で考えると、どんな手法で考えても同じ、だが、
事故前に発想できるかどうかの違いがある（プロアクティブな安全設計）

今後の複雑システムの安全設計のまとめ

- 安全制御メカニズムの可視化（抽象化・階層化した制御構造図）
  - 一枚の絵で安全制御メカニズムを理解し、共有する
  - 異なる立場のレビューアでも、相互の誤解なく理解できる
  - 誰が誰を制御するかという責任と権限の所在が明確
  - 想定外の事故を想定する自由な議論ができる
- 安全論証に役立つ
  - どんな事故を防ぐか、その前段階のハザードをどうやって防ぐかがわかる（UCA、ハザードシナリオ（HS)、コンポーネント安全制約/要求など）
  - 前提とした外部環境の状態の明示化（最悪の状態、複数の条件の重複などの仮定）
  - 論理的な整合性（トレーサビリティ、網羅性、排他性の明示化）

おまけ：STAMPに基づく分析の道具立てとプロセス



STAMP/STPAの可能性

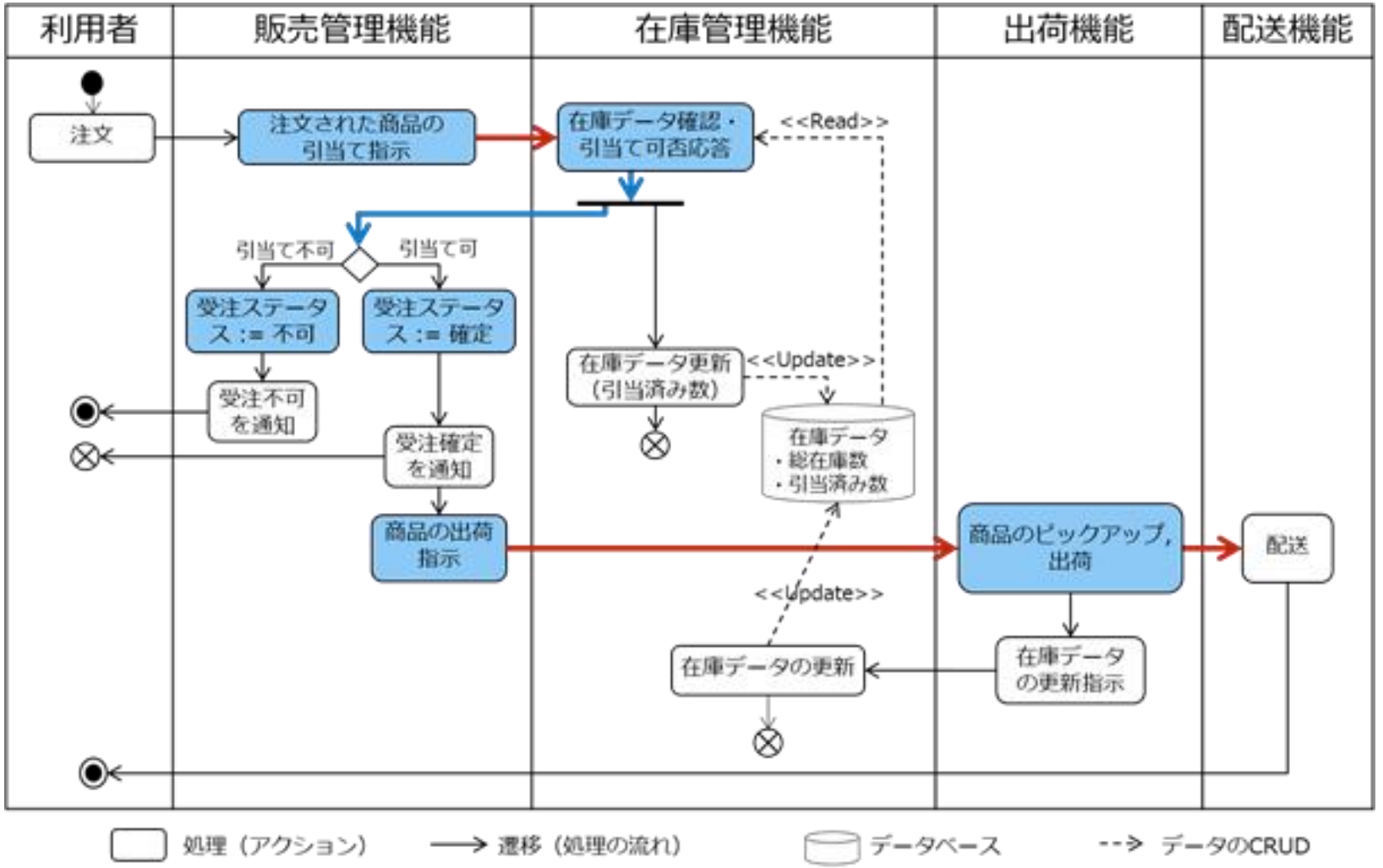
- 過去トラのない新しい製品の安全分析（特にコンセプトフェーズでの利用）
- 想定の難しい環境が関係する現場作業
- 複雑な非定型作業が含まれる保守作業
- 組織、人、コンピュータが絡んだシステム（エンタープライズ系も含む）
- セーフティとセキュリティを同時に考えないといけない製品

エンタープライズ系/ネット通販システムアクティビティ図
（はじめてのSTAMP実践編）

アクシデント:
注文した商品が、利用者に配送されない

ハザード:
受注ステータスが確定と決定された注文に対して、商品が出荷されていない状態

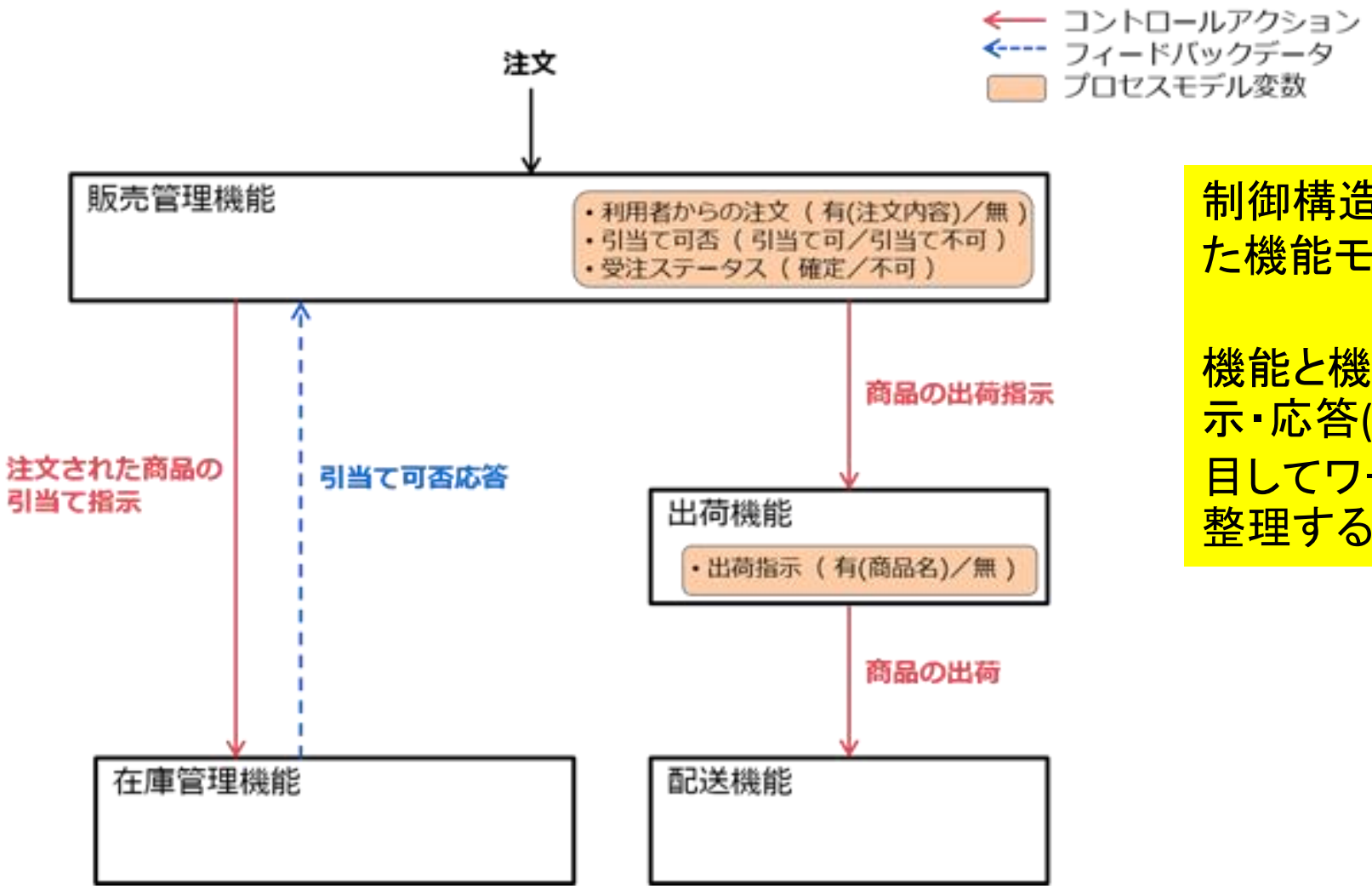
開発者にとってわかり易いが、ユーザの理解はどうか？



事例　ｴﾝﾀｰﾌﾟﾗｲｽﾞ系/ネット通販システム制御構造図
（はじめてのSTAMP実践編）

アクシデント:
注文した商品が、利用者に配送されない

ハザード:
受注ステータスが確定と決定された注文に対して、商品が出荷されていない状態

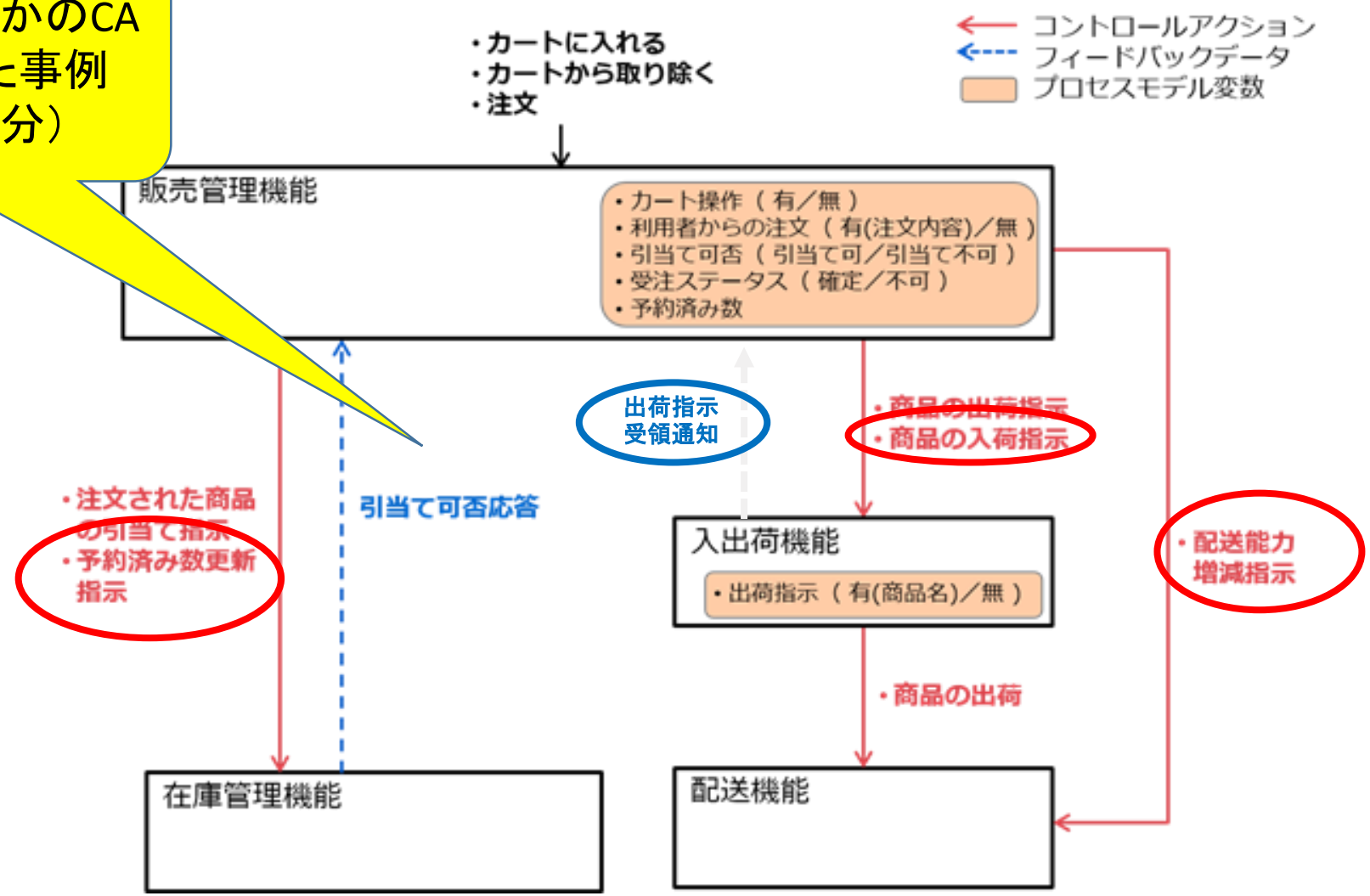


制御構造図(抽象化した機能モデル）

機能と機能間の指示・応答(CAとFB)に着目してワークフローを整理する

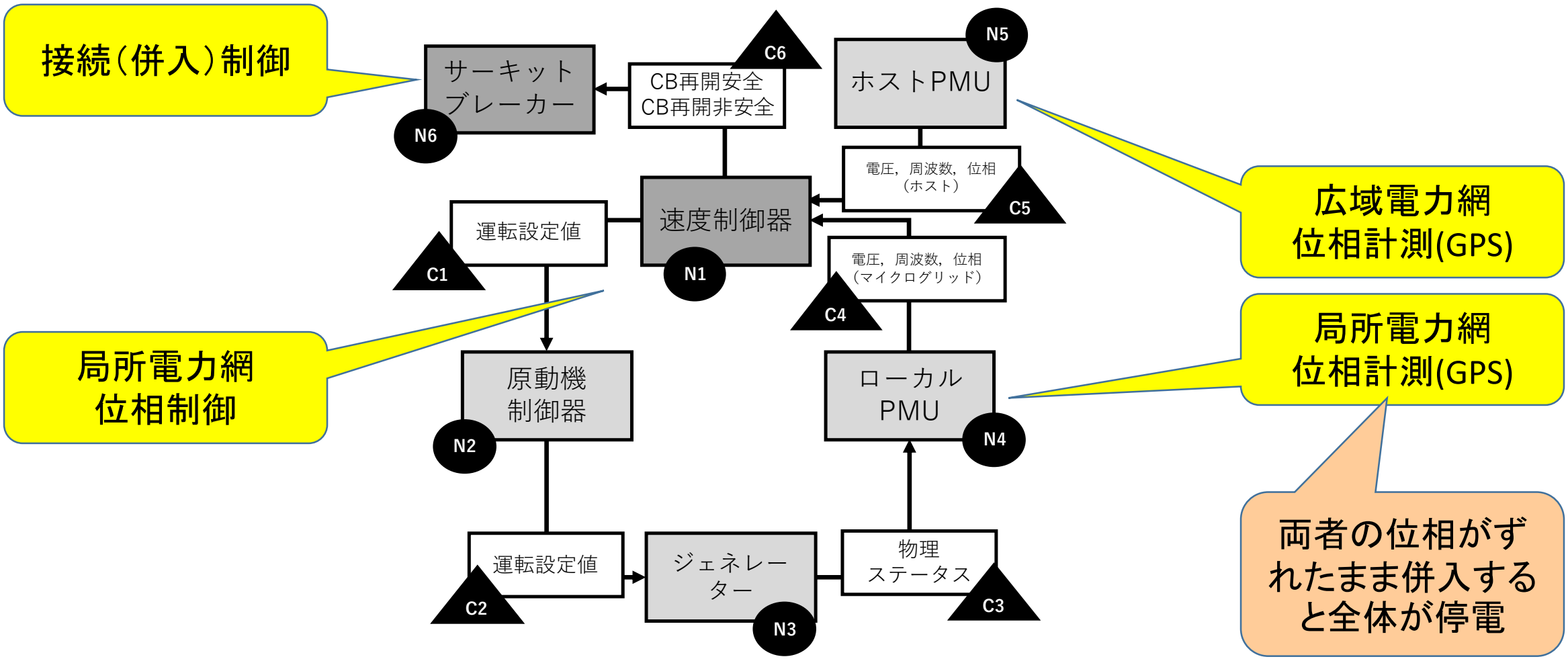
事例　エンタープライズ系/ネット通販システム制御構造図（はじめてのSTAMP実践編）

分析により、いくつかのCAとFBが追加された事例（〇で囲った部分）

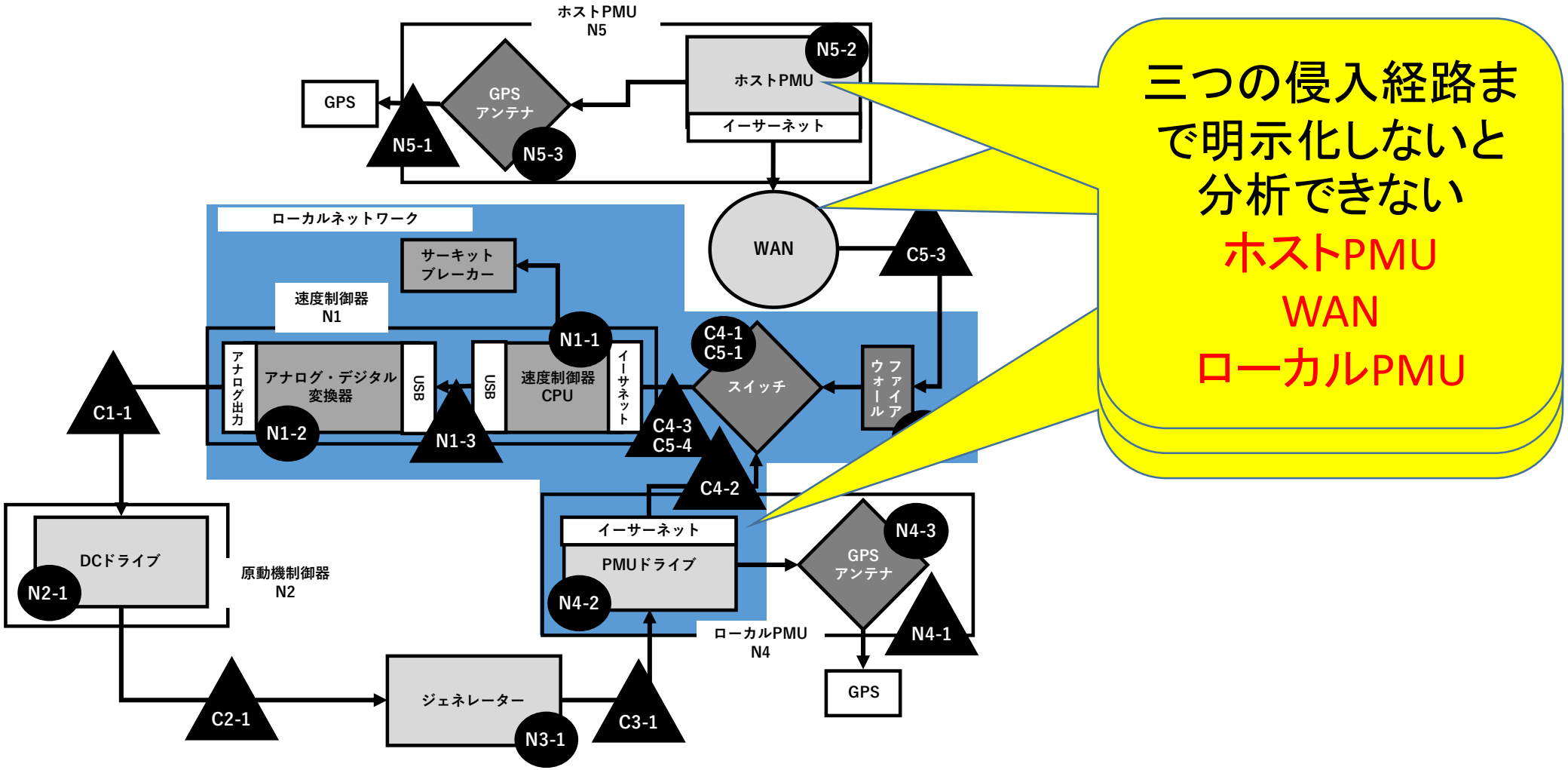


STPA-safesec（セーフティとセキュリティの統合分析）

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（はじめてのSTAMP実践編）



STPA-safesec（セーフティとセキュリティの統合分析）
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
（はじめてのSTAMP実践編）



今後の複雑システムの安全設計のまとめ(追加)

- 安全制御メカニズムの可視化(抽象化・階層化した制御構造図)
  - 一枚の絵で安全制御メカニズムを理解し、共有する
  - 異なる立場のレビューアでも、相互の誤解なく理解できる
  - 誰が誰を制御するかという責任と権限の所在が明確
  - 想定外の事故を想定する自由な議論ができる
- 安全論証に役立つ
  - どんな事故を防ぐか、その前段階のハザードをどうやって防ぐかがわかる(UCA、ハザードシナリオ(HS)、コンポーネント安全制約/要求など)
  - 前提とした外部環境の状態の明示化(最悪の状態、複数の条件の重複などの仮定)
  - 論理的な整合性(トレーサビリティ、網羅性、排他性の明示化)
- 産業界での活用事例の情報共有は、互いに役立つはず!

ご清聴ありがとうございました