システム安全性分析手法
STAMP/STPA入門コース

2022年10月5日

安全性向上委員会
三原　幸博

一般社団法人
組込みシステム技術協会
Japan Embedded Systems Technology Association

講義内容

- システム安全性分析とは
- システム安全性分析の歴史
- 既存のシステム安全性分析手法の限界
- 新しいシステム安全性分析手法STAMP
  - 事故モデル
  - アクシデント、ハザード、安全制約（事例付き）
  - コントロールストラクチャ（事例付き）
  - 安全でない制御アクション（事例付き）
  - ハザード誘発要因（事例付き）

システム安全性分析とは

システムの安全性とは、
「システム(system)とは, “所定の任務を達成するために, 選定され, 配列され, 互いに連係して動作する一連のアイテム(ハードウェア, ソフトウェア, 人間要素)の組合わせ.” 安全性(safety)とは, ``人間の死傷又は資材に損失若しくは損傷を与えるような状態がないこと. “ 」（ JIS-Z8115）

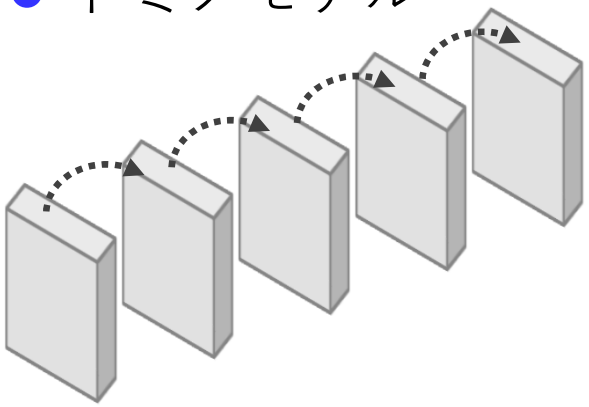
システムの安全性分析とは、
「システム(system)開発時に、実装に先立ってシステムが 安全性(safety)を脅かすハザード要因（リスク）が内在しているか否か、内在している場合はどのようなハザード要因（リスク） であるかを分析すること」

システムの安全設計とは、
「システム(system)の安全性分析を行い、発見されたハザード要因（リスク）を取り除く対策を設計に取り込むこと」

事故モデル（従来）

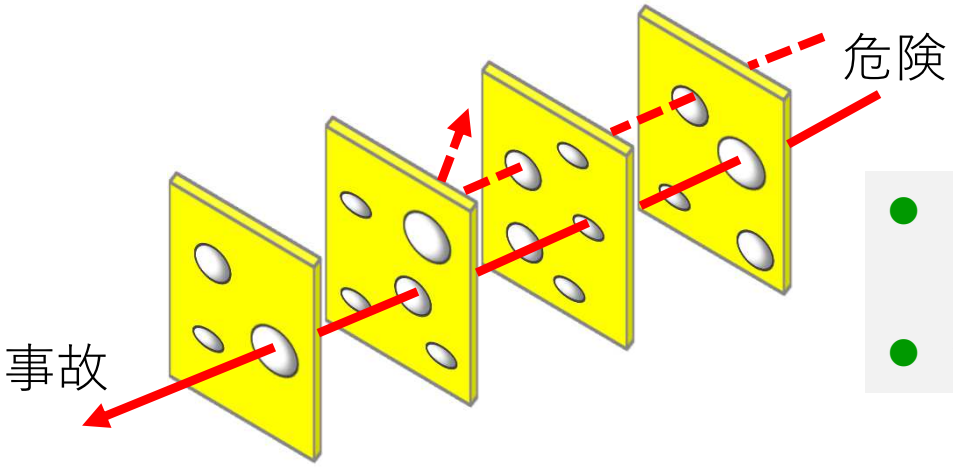
■ 従来の事故モデル … イベントチェーンの形式

● ドミノモデル



- **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



- **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**

システム安全性分析手法 （従来）

手法のタイプ	手法名	手法内容
演繹的手法	FTA (Fault Tree Analysis)	危険事象に対する原因となる事象とその事象に対する防護手段の検討を階層的に実施し、FTを作成する。FTができたらFTの構造分析を行い、定量化する。
	ETA (Event Tree Analysis)	まず初期事象を設定し、初期事象からの事故進展を考慮しながら、進展キーの項目を設定する。各進展キーの成功/失敗を統合することにより、シナリオを作成し、最終事象がどのような事象になるかを判断する。その後、定量分析や対策案の抽出といった分析を実施する。
帰納的手法	FMEA (Failure Mode and Effects Analysis)	システムの構成要素から出発してシステム全体に与える影響を調べる解析手法である。
両手法の使い分け	HAZOP法	設計からのずれの起こる箇所及びその原因と結果を明らかにするために、プロセスの各部を調査することである。
	FMFEA法 (Failure Mode Factors and Effects Analysis)	構成部品の故障モードについて、その要因及び影響の解析をそれぞれFTA,ETAを利用した複合型の安全性評価手法である。要因系と結果系の詳細な解析を進めることで、問題の発見と予測ができる。

システム安全性分析の歴史

～19分

1940 1950 1960 1970 1980 1990 2000 2010 2020

FMEA

FTA

HAZOP

ETA

FMFEA
(FTA + ETA)

- **Introduction of computer control**
- **Exponential increases in complexity**
- **Lots of new technology**

既存のシステム安全性分析手法の限界

環境変化による限界

- ネットワークを通じてシステムが連携する新たなサービス拡大
- 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
- システム相互間の複合原因によるシステム障害が増加
- 長期間の保守によるシステムの劣化が原因の障害が増加
- 同じ原因でのトラブルが多い
  - 個別視点の分析に留まっている
  - 原因分析が充分にできていない

既存のシステム安全性分析手法の限界

（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した機器
- **ソフトウェアは純粋に設計そのもの**

（2）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

- 計画や理解や予測で、全ての望ましくないシステムの挙動を防ぐことは
- 全ての設計エラーをテストで駆除することは

できない

既存のシステム安全性分析手法の限界

現在事故には2つのタイプが有る

<u>コンポーネント故障事故</u>

- 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

古いタイプの事故

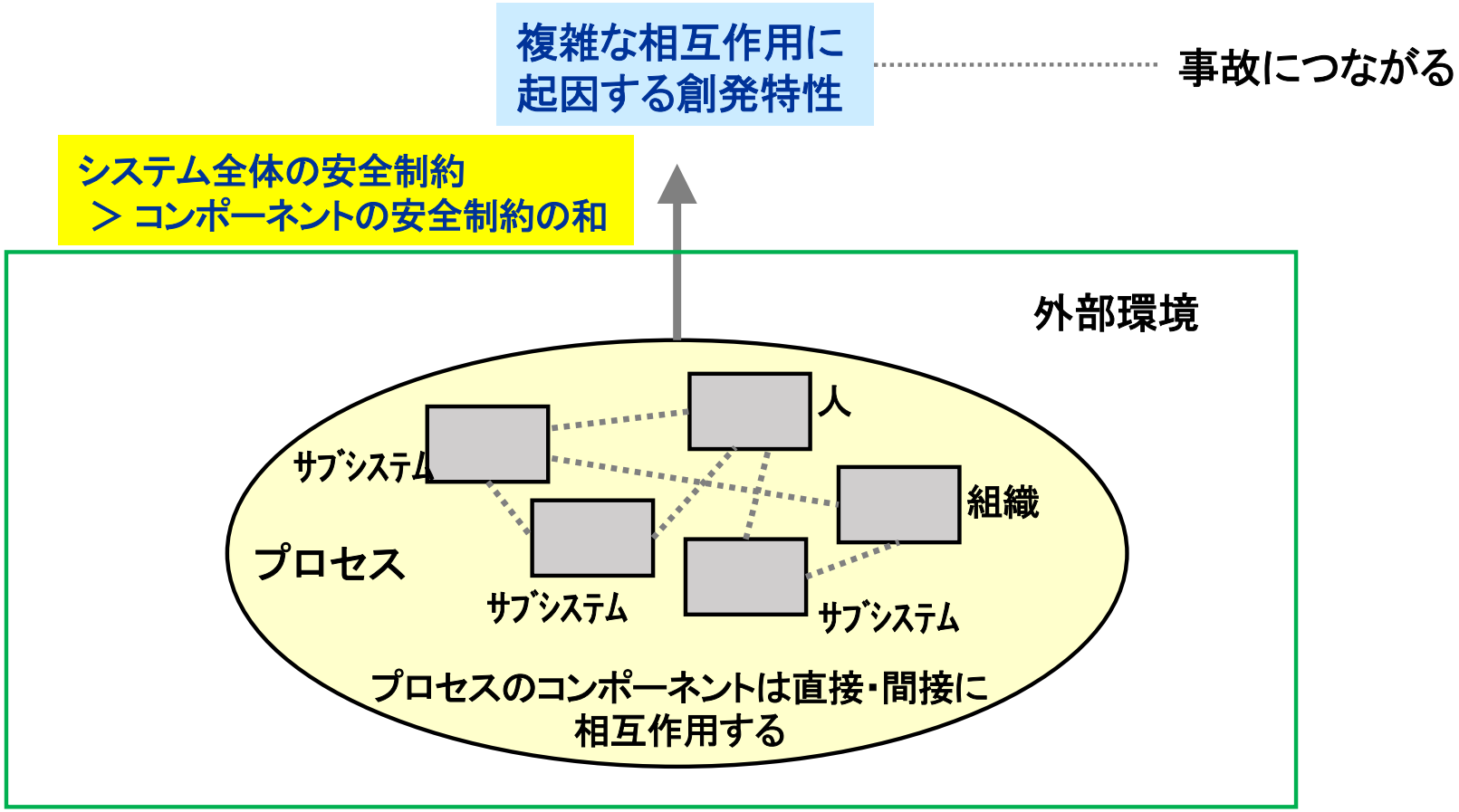
<u>コンポーネント相互作用事故</u>

- 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

新しいタイプの事故

既存のシステム安全性分析手法の限界 ～30分

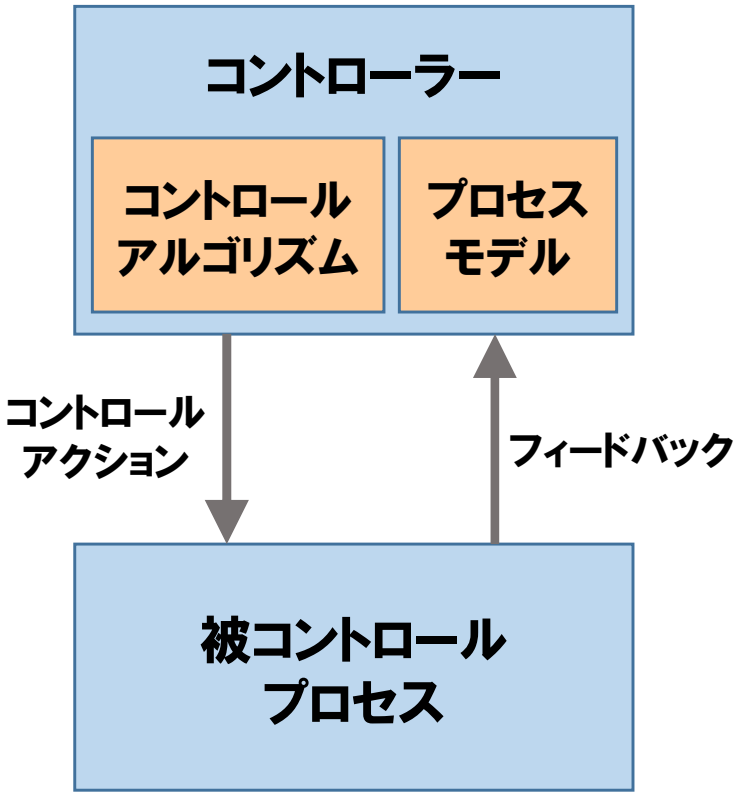
コンポーネント相互作用事故は、従来の事故モデルでは対応できない



新しいシステム安全性分析手法STAMP

- 事故モデル
- アクシデント、ハザード、安全制約（事例付き）
- コントロールストラクチャ（事例付き）
- 安全でない制御アクション（事例付き）
- ハザード誘発要因（事例付き）

STAMPの事故モデル



- コントローラーは、コントロールアクション（CA）を決めるのにプロセスモデルを用いる

  プロセスモデル … 制御対象の状態や動作を抽象化したもの

- この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
- 4タイプの非安全なコントロールアクション
  - 与えられないとハザード
  - 与えられるとハザード
  - 早すぎ、遅すぎ、誤順序でハザード
  - 早すぎる停止、長すぎる適用でハザード

- ソフトウェアと人間の挙動の適切なモデルによって、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

STAMP/STPAの手順

- **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：**
  非安全な制御動作（UCA）の抽出
- **Step 2：**
  ハザード誘発要因（HCF）の特定

最新のSTPA　HANDBOOKでは、STEPの名称がSTEP1～4となっているが、ここではIPAのSTAMP解説シリーズの表記に従った。

例題 ― 電車ドア開閉システム

～35分

電車で、ドアを開閉する部分に着目して、そのシステムの安全性を解析する。

- 走行中に開いてはいけない
- 駅間で開いてはいけない
- 緊急事態では、停車中に開くことができる

…

システムをどのように考えるか？
最初は非常に少ない情報だけでよい。

例えば、アクチュエーターがあることは分かっているが、
それがどのように動くかはまだ分からなくてよい。
形式的なモデルもまだない。

列車ドアに関するアクシデント事例

韓国大邱(テグ)市地下鉄火災
日時:2003 年2 月18 日(火)9:53 ごろ
場所:韓国大邱市中央路駅
被害:死者192 名、負傷者148 名
韓国大邱市地下鉄中央路駅で、アンシム(安心)行き地下鉄の車両にガソリンを撒いて放火する事件が発生した。火災発生直後に隣接駅を出発した下り車両が中央
路駅に停車し、火災のため運行ができなくなり、延焼した。この際に、車両のドアが開放されず、適切な誘導が実施されなかったため、多数の乗客が車両に取り残さ
れ火災に巻き込まれ焼死した。この火災で192 名が死亡、148 名が負傷した。

列車ドアに関するアクシデント事例

状況により変わる安全制御行動
～小田急･火災事故～

2017年9月10日　16:00過ぎ、乗客約300人

プラットフォーム外で停車の際はドアを開けると危険
しかし、火災時は、ドアを開けて退避させないと危険
→状況（Context)依存で矛盾する安全制御行動
→人やソフトウェアによる複雑な安全制御行動は副作用を伴う可能性もある、

従来の分析法（FTA等）の限界
次世代の安全解析法としてのSTAMP/STPAの可能性

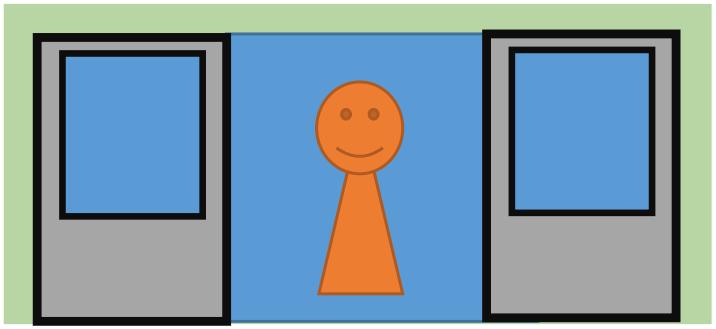
Step 0　準備1の概説

作業名称	アクシデント、ハザード、安全制約の識別
目　的	アクシデントを定義する 安全制約を導き出す
入　力	① 要求仕様書 ② [ドメイン専門家]
処　理	① 分析しようとするアクシデントが何であるかを定義する ② アクシデントと成り得るハザードには何があるかを考える ③ ハザードの裏返しとなる程度の粒度で安全制約を導き出す
出　力	① アクシデント、ハザード、安全制約の一覧表
備　考	• アクシデント：喪失（Loss）を伴うシステムの事故 • ハザード：アクシデントにつながるシステムの状態 • 安全制約：システムが安全に保たれるために必要なルール 例えば、踏切制御システムにおいて踏切がいつまでも開かないのは、サービス利用者・提供者に経済的損失を与えたり、精神的苦痛を与えることになることもあるが、人の生命に関わる事柄に焦点を絞ったときには『アクシデントではない』と定義できる。

アクシデント、ハザード

- アクシデントとは
  - 望まれない・計画されていないイベントで、ロスへ至るもの
  - ロス：人命が失われる、人が負傷する、環境汚染、ミッション失敗等
- ハザードとは
  - システムの状態または条件の集まり
  - 最悪の環境条件の下で、アクシデントへ至る
- 具体例
  - アクシデント：乗客が列車から落ちる
  - ハザード1：列車が動いているのにドアが開く
  - ハザード2：誰かがドア付近にいるのにドアが閉じる

アクシデント、ハザード



電車ドア開閉システム

ロスとして、金銭や社会的信用を含めると・・・
「平常時に駅停車中に列車内に閉じ込められる」もアクシデントとなる

今回は、「平常時に駅停車中に列車内に閉じ込められる」はアクシデントとしないので、「平常時に駅停車中にドアが開かない」はハザードではない

- アクシデントは？
  - A-1：列車から降りるときに負傷（H-2）
  - A-2：閉まるドアに当たる（H1）
  - A-3：緊急時に列車内に閉じ込められる（H-1, H-3）
- ハザードは？
  - H-1：ドア付近に人がいるのに、ドアが閉まる（A-2, A-3）
  - H-2：列車が走行中あるいは駅にいないのに、ドアが開く（A-1）
  - H-3：緊急時なのに，旅客あるいは乗務員が列車外へ出られない（A-3）

安全制約

- 安全制約とは
  - ハザードを防止する(安全が守られる)ために必要となるルール

- 具体例
  - 列車が走行中、あるいは駅に停車していないときには、ドアは閉まっている
    (ハザードの裏返し)

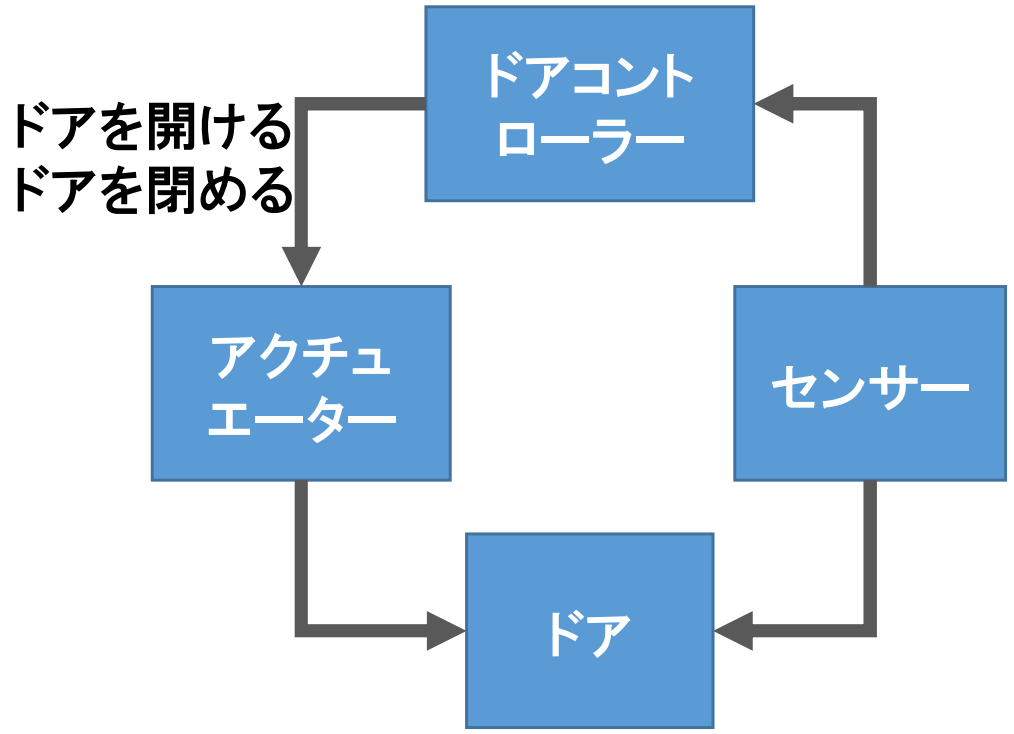
Step 0　準備2の概説

作業名称	コントロールストラクチャーの構築
目　的	登場人物間の依存関係を制御構造図で表す。 制御主体と制御対象の間で行われる制御（サブシステム間の相互作用）には何があるかを明確化する。 その後の分析作業において理解し易いイメージを共有する。
入　力	① 要求仕様書
処　理	① 要求仕様書から登場人物（ブロック）を抽出する ② 要求仕様書から各ブロックの役割を抽出する ③ 役割を果たすために必要な制御、役割を果たした結果のフィードバックを抽出する ④ 制御、入出力情報（情報を与えるのみで制御を行うわけではない）の違いを分別する ⑤ ブロック間を矢印線で結び、制御・入出力を表す
出　力	①制御構造（コントロールストラクチャー）図
備　考	ブロックの数は4つ程度が良いと言われている。 それ以上多くなる（抽象度を下げる）と、以降で分析すべき組み合せが多くなり、集中しにくく検討漏れを起こしかねないので、工夫が必要になる。

コントロールストラクチャー

- コントロールストラクチャーとは
  - コンポーネント間の機能動作を示したシステムの設計図
    - コンポーネント間でやり取りされる制御の指示やフィードバックなどを矢印で結んで表す

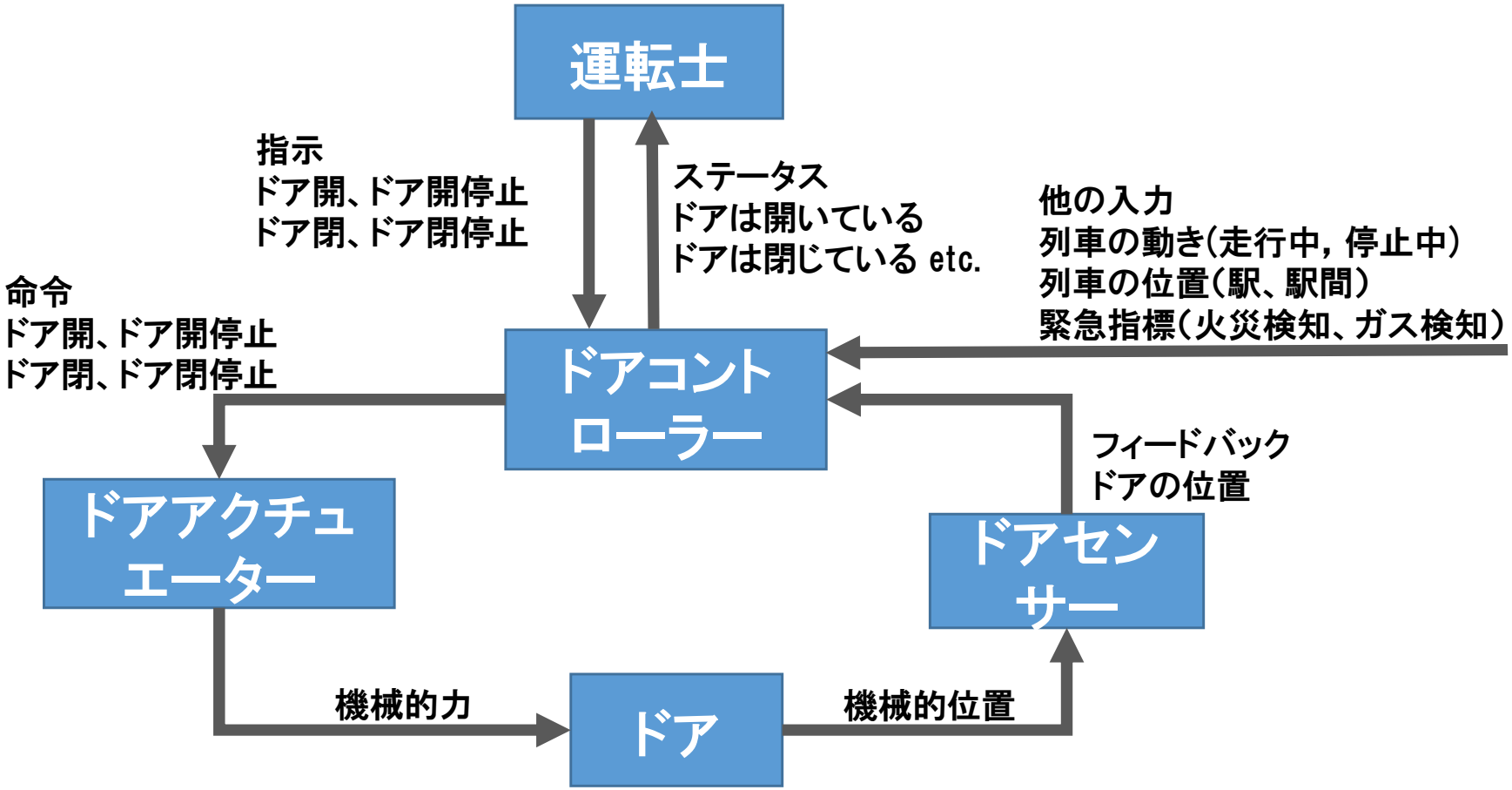
- 具体例
  - ドア開閉システム
  - 粒度（詳細さ）はいろいろ



コントロールストラクチャーの作成

- 登場人物と依存関係をまとめる
  - 要求仕様から機能に対応して登場人物(ブロック)を抽出する
  - 指示とフィードバックによってブロックを結合する
  - 注意すべきこと
    - コンポーネント間関係の抜けを防止する
    - コンポーネントとその中身の整合性を維持する
    - 適切な粒度を選択する

先の例を詳しくしてみる



Step 1 の概説

作業名称	UCA（Unsafe Control Action：非安全制御動作）の抽出
目　的	ハザードにつながり得る制御動作の不具合を識別する（発想する）
入　力	① UCAを導き出すための4つのガイドワード（4分類） ② アクシデント、ハザード、安全制約の一覧表 ③ 制御構造図
処　理	① UCA 識別の表を準備する ② 最上列に4つのガイドワードを記す ③ 最左行に制御構造図中にある制御をすべて記す ④ 各マスごとに、当該（最左行）の制御動作が当該（最上列）の状況になった場合、いずれかの安全制約違反になり得るかを考える ⑤ 安全制約違反になり得るならば、UCAであると判断する
出　力	①縦軸：制御行動、横軸：ガイドワードとしたUCA一覧表
備　考	想定外を排除することを忘れないように。

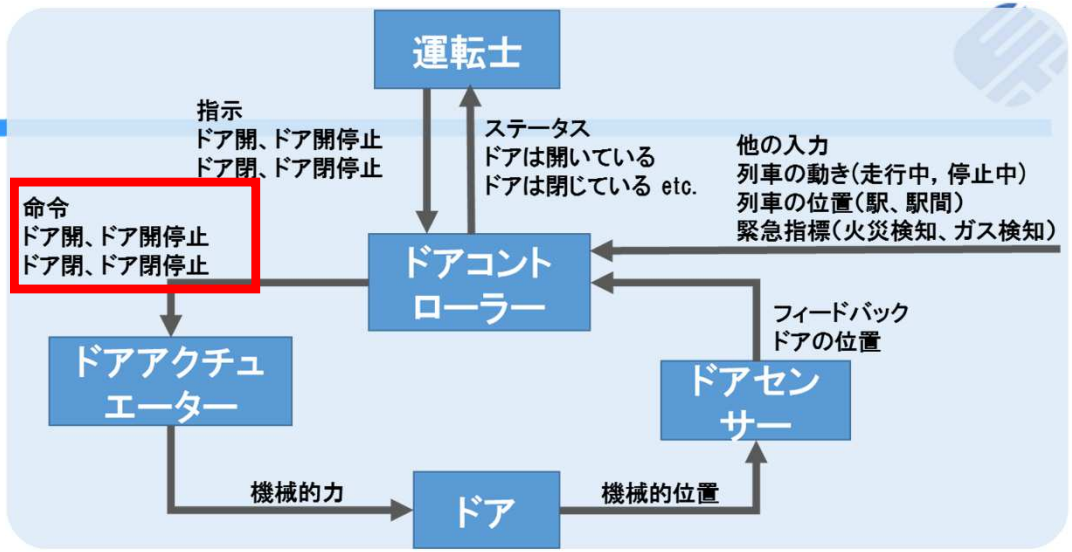
非安全なコントロールアクション

- 非安全なコントロールアクションとは
  - ハザードに至る、コントロールアクションの状態・条件

- 具体例
  - 列車が動いているのに、ドアが開く
    コントロールアクション「ドアが閉まる」に対して、「与えられるとハザード」を考える
  - 警報が鳴らないのに、列車が踏切を通過する
    「警報を鳴らす」に対して、「与えられないとハザード」を考える

UCA抽出の手順

H-1：ドア付近に人がいるのに、ドアが閉まる
H-2：列車が走行中あるいは駅にいないのに、ドアが開く
H-3：緊急時なのに、旅客あるいは乗務員が列車外へ出られない



4つのタイプ

コントロールアクション	与えられないとハザード	与えられるとハザード	早すぎ、遅すぎ、誤順序でハザード	早すぎる停止、長すぎる適用でハザード
ドア開	緊急時にもかかわらず、ドアコントローラーがドア開命令を与えない［H-3］			
ドア閉				

UCA識別のヒント
- H-3：緊急時なのに、旅客あるいは乗務員が列車外へ出られない
- 「緊急時」とは？ 火災検知、ガス検知、地震検知、・・・
  - 詳細化してみる ⇒「列車が駅で停止中に火災検知された」
  - 分解してみる ⇒ 列車の位置＝駅、列車の動き＝停止中、火災＝検知
    列車の動き＝走行中 等も考える必要があることが分かる

コントロールアクションがハザードへ至る条件を記載する

UCA

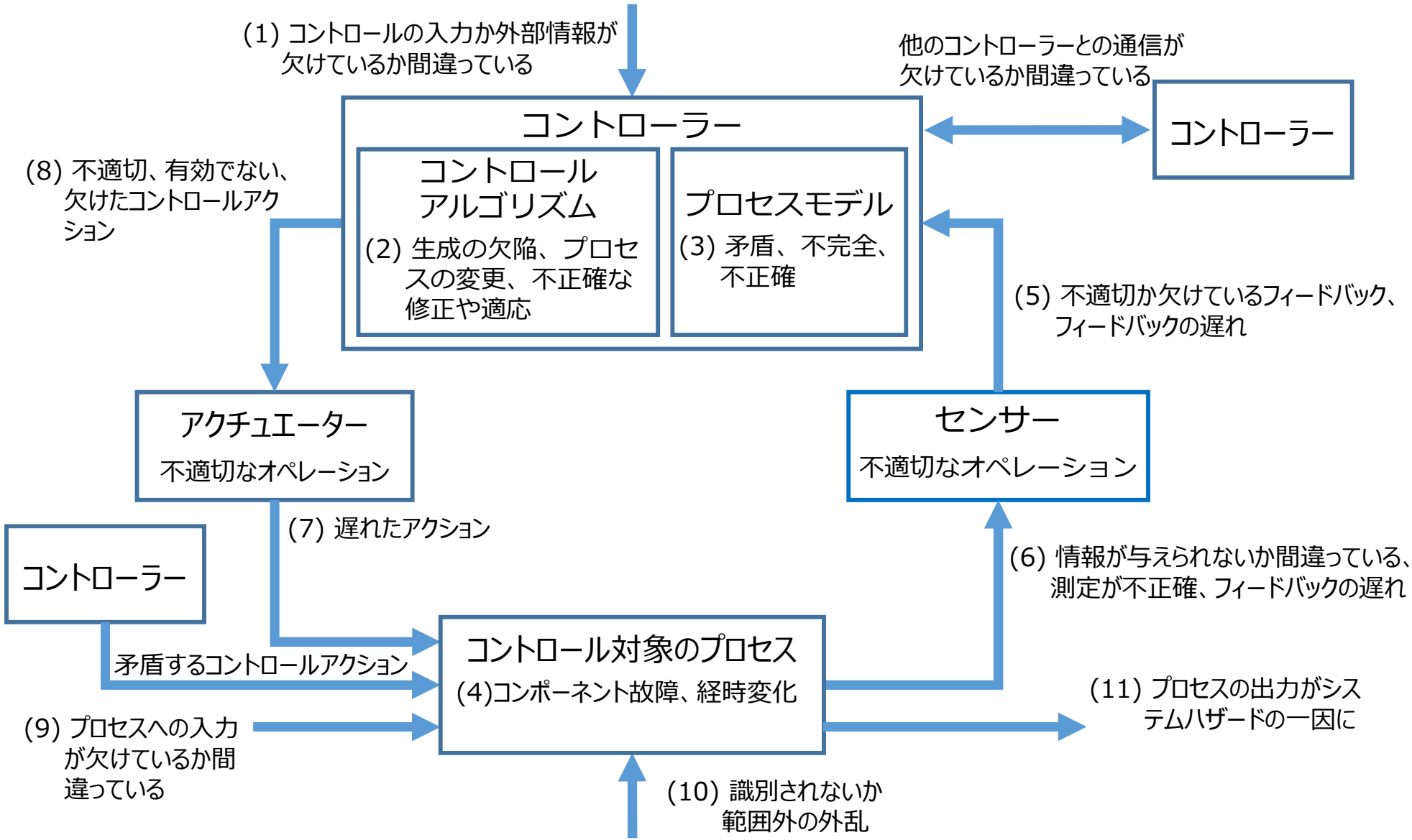
UCAにラベルを付けると便利（変更に強い）
コントロールアクション×4タイプ×通番
例：UCA-CA01-N、UCA-CA02-P など

制御動作	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、誤順序でハザード(T)	早すぎる停止、長すぎる適用でハザード(D)
ドア開（CA1）	UCA-CA1-N1: 列車がプラットフォームで停止したのにドア開が命令されない [ハザードへ至らない] 緊急避難のためのドア開が命令されない [H-3] 人または障害物が扉口にいてドアが閉じつつあるのにドア開が命令されない [H-1]	列車が動いているのにドア開が命令される [H-2] UCA-CA1-P2: 列車がプラットフォームで停車していないのにドア開が命令される [H-2]	列車が停止前あるいは動き出した後(「列車が動いている」と同じ)にドア開が命令される [H-2] 列車が停止後ドア開が遅すぎで命令される [ハザードへ至らない] 緊急時にドア開が遅すぎで命令される [H-3]	通常のドア開停止よりも早すぎるドア開停止が命令される [ハザードへ至らない] 緊急停止時に早すぎるドア開停止が命令される [H-3]
ドア閉（CA2）	列車が動く前にドア閉または再度のドア閉が命令されない [H-2]	人または障害物が扉口にいるのにドア閉が命令される [H-1] 緊急避難時にドア閉が命令される [H-3]	乗客が乗降りを終える前にドア閉が早すぎで命令される [H-1] 列車が動き出した後にドア閉が遅すぎで命令される [H-2]	ドアが完全に閉まっていないのに早すぎるドア閉停止が命令される [H-2]

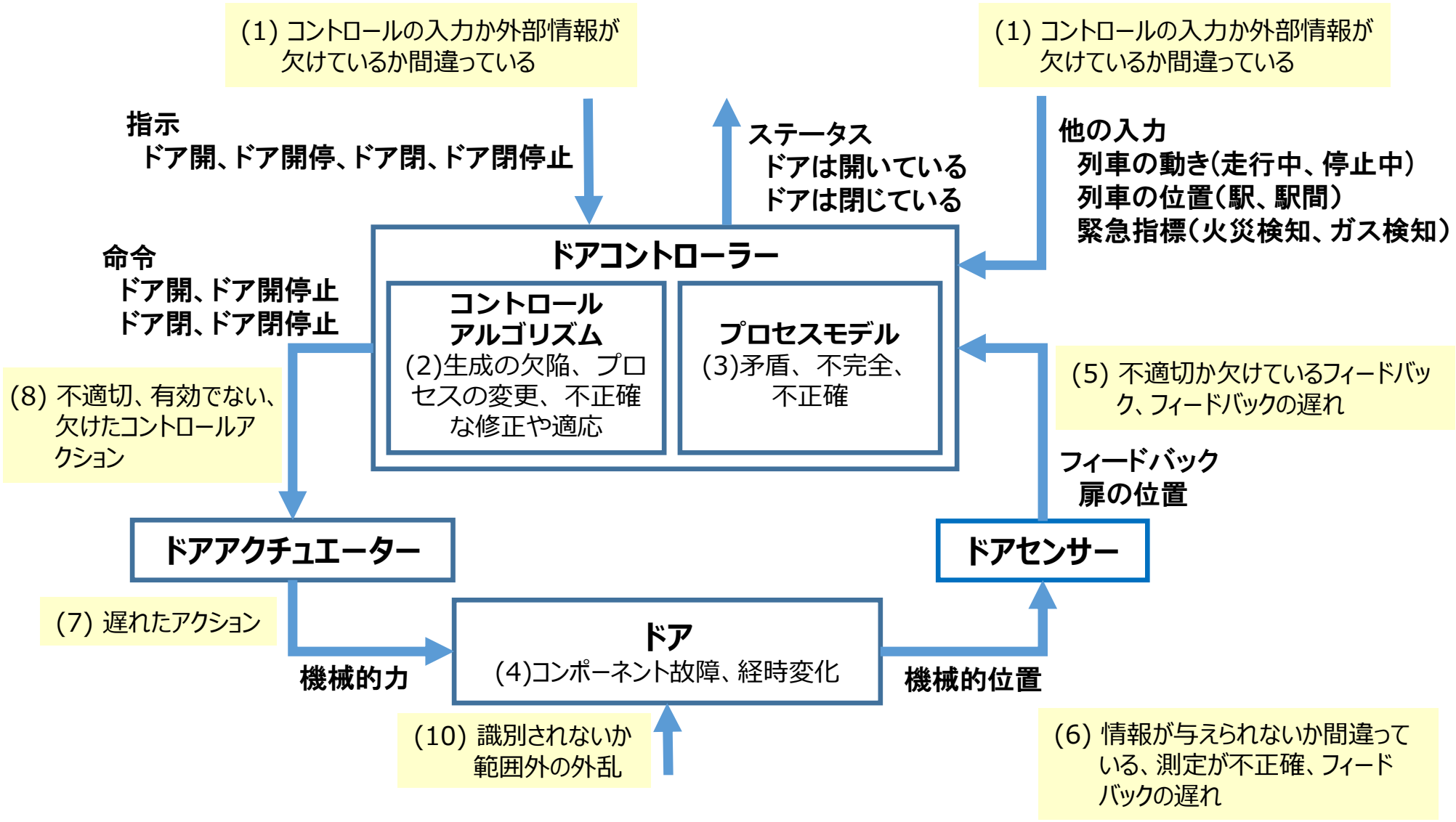
Step 2 の概説

作業名称	HCF（Hazard Causal factor：誘発要因）の特定
目　的	どのようなHCFがあったらUCAになり得るのかを考え、ハザードシナリオを作る
入　力	① HCF特定のための11個のガイドワード ② 制御構造図 ③ UCA一覧表
処　理	① 制御構造図からコントロールループを抜き出して、その中の各制御に該当するガイドワードを割当てる ② [制御構造図中の各制御に該当するガイドワードを割当てる] ③ Step1で識別したUCA毎に、ガイドワードを一つずつあてはめてみてハザードとなり得るかを考える ④ ハザードとなり得るならば、どういう条件下で当該ガイドワードの事象が発生し、その後どういうシステム挙動になったらハザードとなって、アクシデントにつながるかのシナリオを作る
出　力	① 縦軸：UCA、横軸：ガイドワードとしたハザード要因の一覧表 ② ハザードシナリオ
備　考	すべてのUCAに、それぞれガイドワードのすべてをあてはめて考える。

安全制約を破られる原因発見のガイダンス



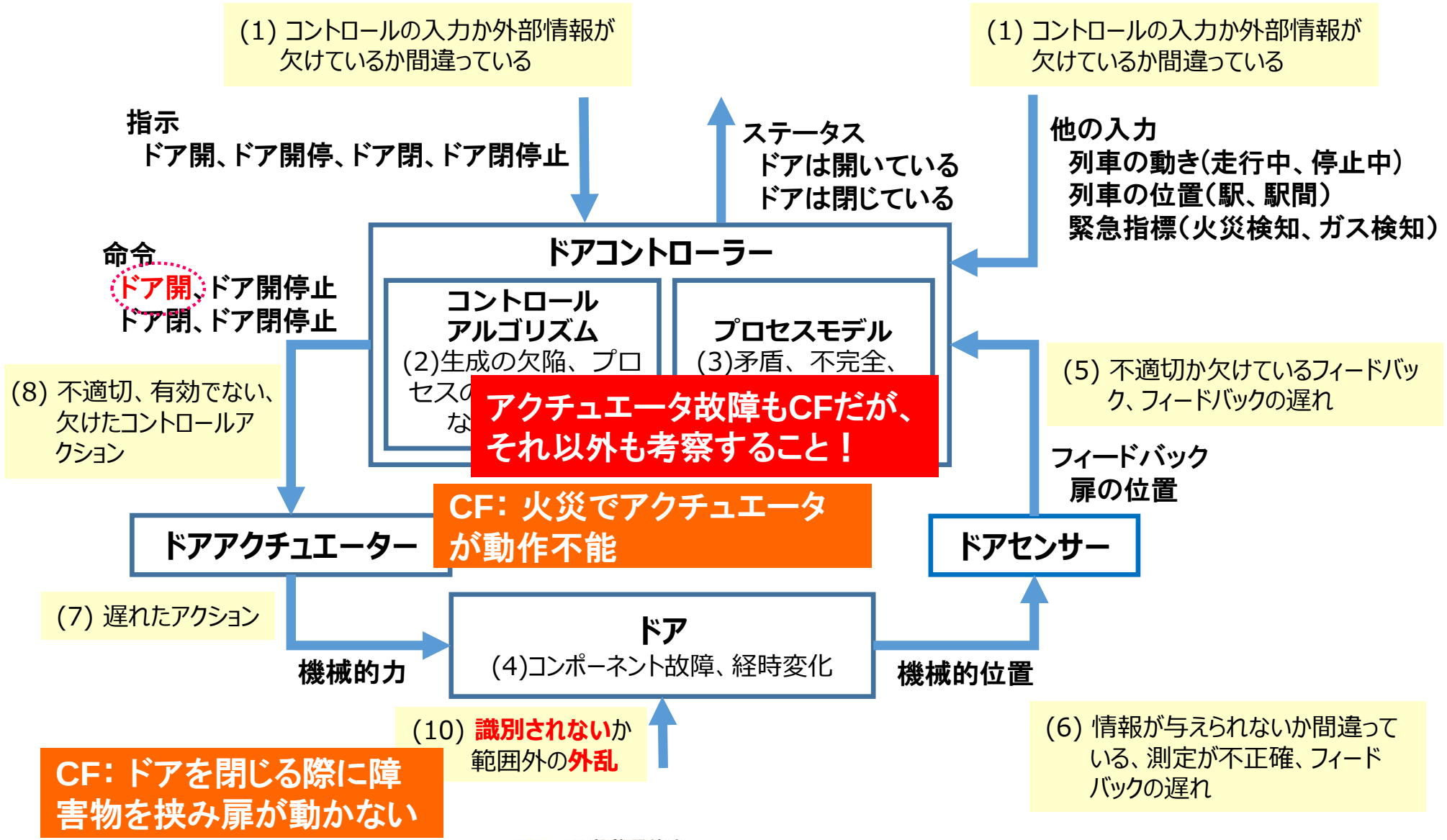
HCF特定の準備



HCF特定の例1

シナリオ：ドアコントローラーはドア開命令を出したがドアが開かない

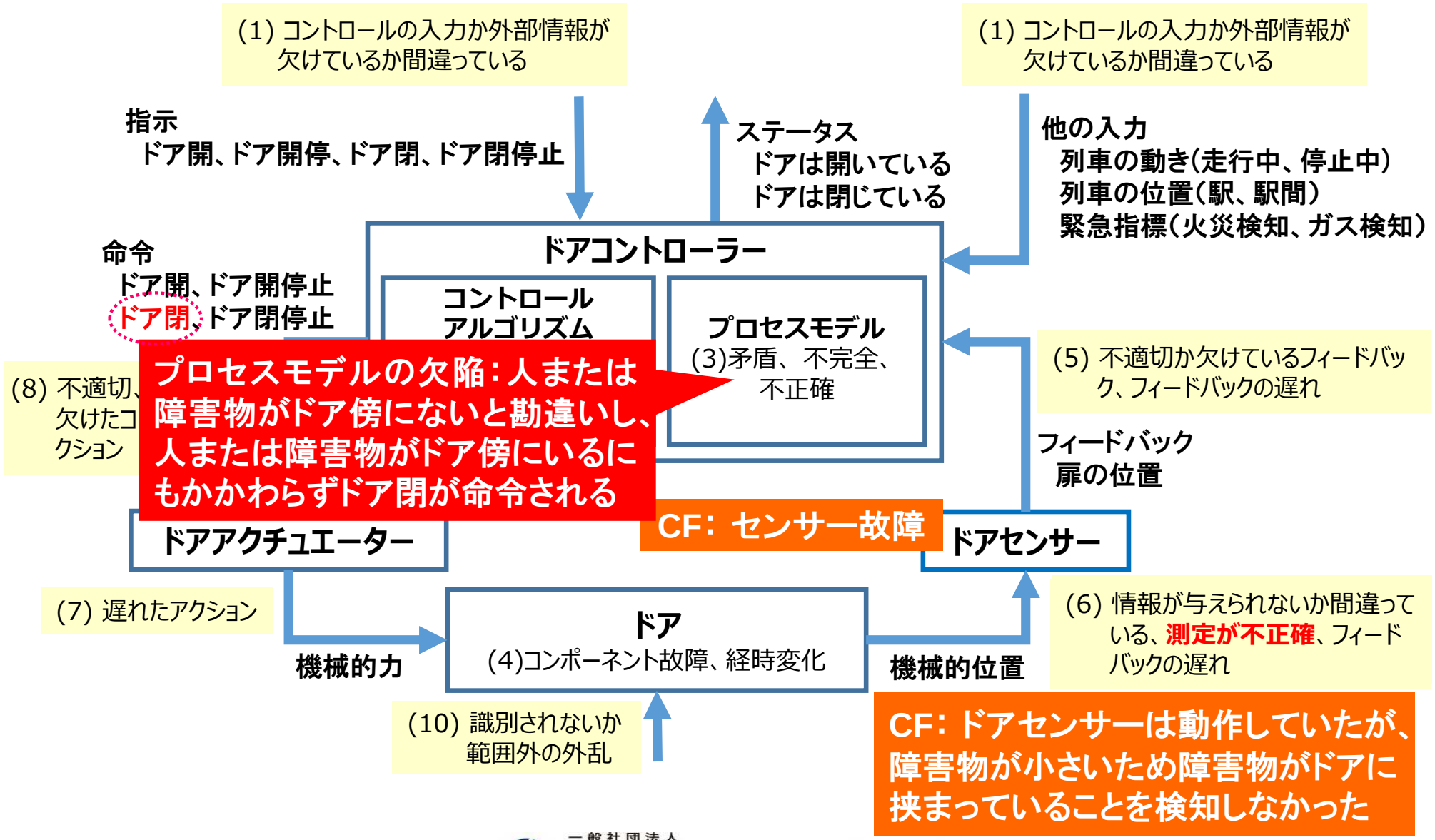
UCA：緊急時にもかかわらず、ドアコントローラーがドア開命令を与えない［H-3］



HCF特定の例2

シナリオ：人がドア傍にいるのに、「人はドア傍にいない」というフィードバックがコントローラーになされた

UCA：人または障害物が扉口にいるのに、ドア閉が命令される［H-1］



終わり