システムズ理論によるソフトウェアの安全設計
～現状と今後～

2019年2月6日

余宮　尚志（（株）東芝　研究開発本部）
hisashi.yomiya@toshiba.co.jp
hisashi.yomiya@gmail.com

◆安全設計入門改訂版

電波新聞社 2010年7月発行

改訂版の主旨

企業でのソフトウェア技術者向けの安全設計の入門書として,
安全分析の基本手法(FTA,FMEA,HAZOP),
各種機能安全規格、
STAMP/STPA による新しい安全分析法などを広く学べる。

目次

1.安全の基本
2.安全規格体系と概要
6.自動車の機能安全(ISO 26262)
7.サービスロボットの機能安全(ISO 13482)
8.システム思考で考えるこれからの安全

2019 年 3 月発刊予定（電波新聞社）

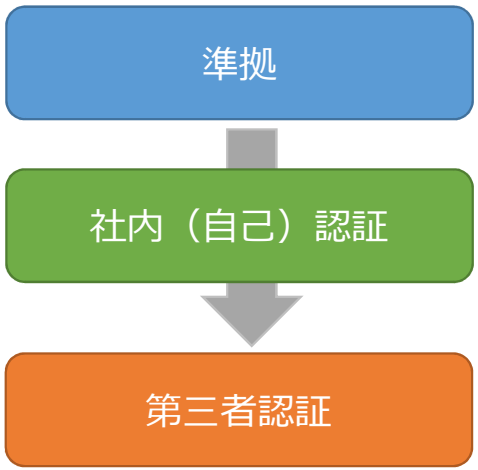
本講演でもこの一部をお話します

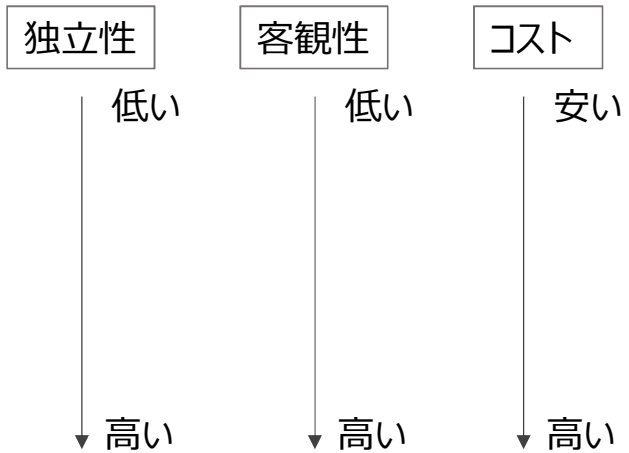
国際安全規格を採用する理由(1)

- 国際的な基準に適合していることを示すことができる
- 安全規格に準拠していることで、**製品の安全性を客観的に説明できる**
- 万が一、市場で事故が発生した場合などに**説明責任を果たしやすくなる**

国際安全規格を採用する理由(2)

- 立場によって国際安全規格に準拠する目的は異なる
- 当社では…
  - 安全性確保の手段、説明責任遂行時の根拠とする　⇒ **<u>安全論証</u>**
  - 入札条件や顧客要求である
  - 他社との差別化や営業戦略である





各々の安全規格の限界(適用範囲)

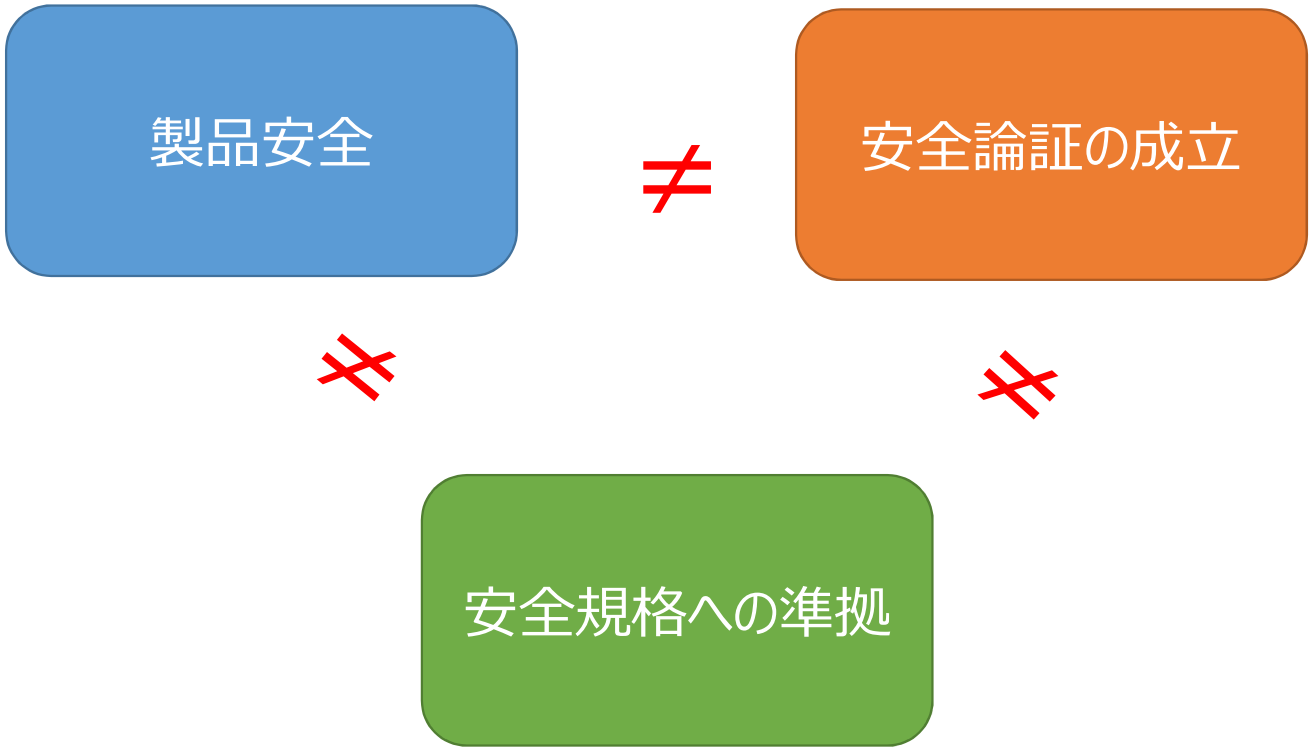
- 各々の安全規格には、適用範囲(Scope)が定められている
- IEC 61508
  - コンピュータを含む、安全機能が求められるさまざまな製品を対象としているが、コンピュータを含んでいない**製品(はさみなど)は対象外**であるし、**コンピュータで実現していない安全機能(逆流防止弁やクッション材など)の要求事項も規定していない**
- ISO 26262
  - IEC 61508の適用範囲であって、さらにISO 26262:2011では**車両総重量が最大3,500kgまでの量産される乗用車に限定されている**。ISO 26262:2018では、モータサイクルや、トラックやバス、トレイラーなどの商用大型車に適用範囲を広げているが、**モペットは除かれている**

安全規格の限界

- 各々の安全規格の限界(適用範囲)とは、当該の安全規格における要求事項を定めた適用範囲外を指す
- 実際の製品開発では**安全規格の限界にかかわらず包括的に安全方策を導入して、リスクを軽減する必要がある**
- 現行における、ほとんど全ての安全規格の限界(適用範囲外の内容)　例
  - **ヒューマンエラーへの対応**

    合理的に予見可能でない誤使用や、予見可能であってもあらゆるヒューマンエラーへの対処を安全規格は要求していない
  - **新しい特性を持つ製品への対応**

    自動運転車や人工知能(AI)にかかわる製品等、これまで安全規格で扱いにくかった製品が市場に増えているが、こうした製品にも該当する安全規格がない



※一部に包含関係はある

(参考)安全規格とサイバーセキュリティ

- 安全規格は、使用者が〝合理的に予見可能な誤使用も含めて〟正しい使用方法を行うことを前提としている
- 実際には、製品が悪意のあるサイバーセキュリティの脅威にさらされることで、許容できないリスクが顕在化することが起こるかもしれない
- このような、サイバーセキュリティの脅威や脆弱性を適用範囲として取り込んでいない

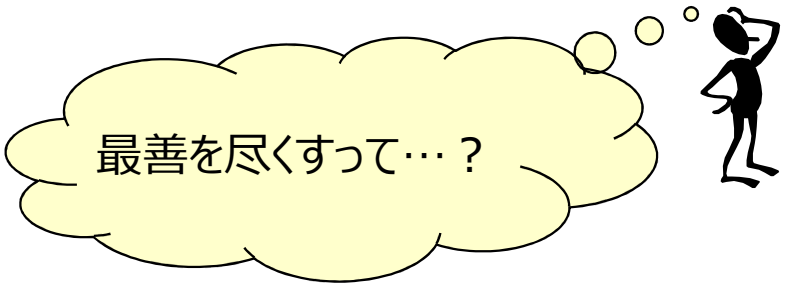
自動運転車やAIに関連した製品のように、安全規格の発行よりも、新しい特性を持つ製品の開発が先行している事例がある。特にサイバーセキュリティによる脅威を考慮した安全確保は、安全規格の確立も含めて今後の新しい流れとなるかもしれない

安全規格と製品安全を支える**安全論証**の考え方(1)

- 安全論証とは、**安全性を設計文章等のエビデンスを用いて、論理的かつ合理的に説明できること**(説明責任が果たせること)である

- すなわち、受容できないリスクがないことをエビデンスを用いて論理的、合理的に説明できることである

安全論証を成立させるために、最善を尽くしていることが、論理的な説明に有利となる

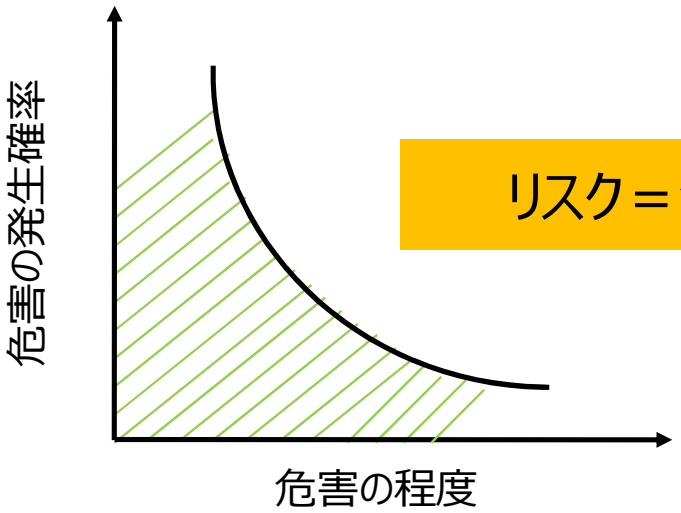


安全規格と製品安全を支える安全論証の考え方(2)

- 安全論証には、以下の二つの考え方が内在している

1. (想定する危険事象に対する)危害の発生確率と危害の程度の組み合わせが受容できるよう設計されていること　⇒　**安全要求の設定が可能**
2. 想定する危険事象に想定外がないこと

⇒　**想定外に対しては安全要求の設定ができない**
(想定外を減らす最善の努力をしたという論証は提示すべき)



リスク＝危害の発生確率およびその危害の程度の組み合わせ

- 「安全」とは、「受容できないリスクがないこと」（斜線部）

※安全規格によっては異なる定義(文言)となっていることがあります

ソフトウェアにおける安全論証の考え方

- 一般論として、ソフトウェア特有の安全論証に対する考え方はない(共通である)

- 安全規格の要求事項を満たすこと、安全規格にある各フェーズの目的を達成すること
- ソフトウェアに関わる安全要求の実現を確実にすること
  - 特に、ソフトウェアにバグがないこと
  - ソフトウェアにバグがないことの確からしさを、客観的に説明できること

  ※ バグとは特に、**安全要求の侵害に至るものを重視**
- ソフトウェア業界の品質に関わる慣例(例:開発プロセスによる組織の成熟度モデル)を取り入れていること　etc

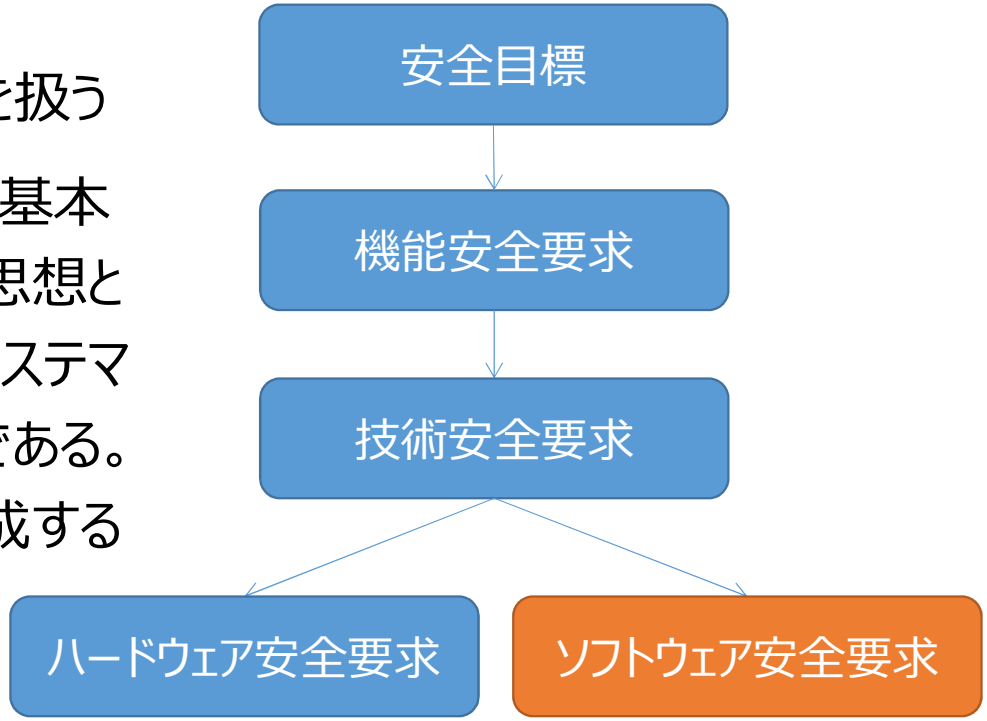
注）ソフトウェアに関わる安全要求に対するバグ

バグがないことを含めて、安全論証を支えている考え方は〝最善を尽くしているか”

ソフトウェアの安全設計~安全規格の目線から(1)

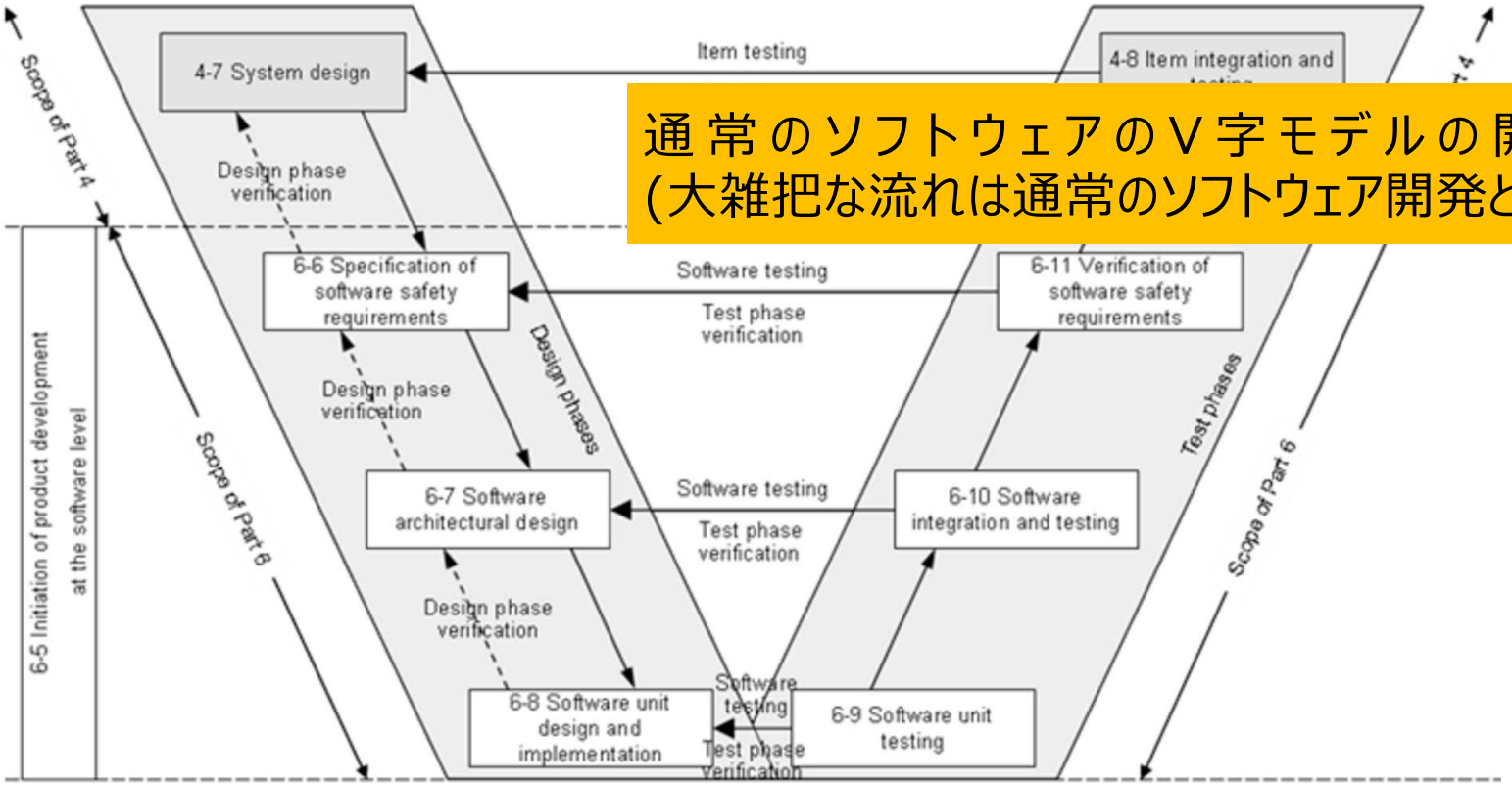
- ソフトウェア安全要求を(技術安全要求から)導出し、ソフトウェア設計を行う
- ソフトウェア設計では、システマティックフォールトを扱う
- 安全を達成するための安全分析も含まれるが、基本的にバグがないようソフトウェアを設計することが思想となっている。安全規格の要求事項の数々は、システマティックフォールトの混入を低減するための事項である。それによって、ソフトウェア安全要求を確実に達成することを説明する

注）ソフトウェアに関わる安全要求に対するバグ



ISO 26262の安全にかかわる要求

ソフトウェアの安全設計~安全規格の目線から(2)



通常のソフトウェアのV字モデルの開発である
(大雑把な流れは通常のソフトウェア開発と変わらない)

ISO 26262:2011, Part6, Figure 2

ソフトウェアの安全分析手法事例

- ソフトウェアの安全分析の主目的(主にアーキテクチャ設計レベル)
  - ① ソフトウェアにおける安全性の向上、安全の達成
  - ② ソフトウェアにおけるバグゼロの達成

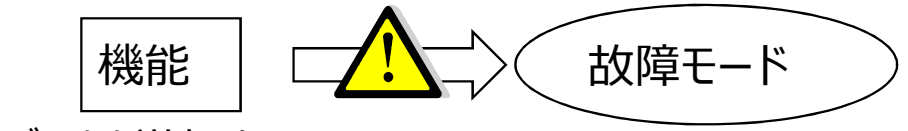
次以降のスライドでは、ソフトウェア安全分析のひとつの考え方を紹介する
注) 安全論証の補強としても有効だが、安全論証には安全分析で最善を尽くすこと以外に、さまざまな視点がある(例：テストケースの十分性)

ソフトウェアで安全分析をするときの課題(FMEAを例として)

1. 故障モード（故障）の抽出が難しい

【故障の捉え方(例)】

ハードウェア	ソフトウェア
経年劣化あり	経年劣化なし
物理特性の変化	条件の組み合わせ



データを送信する

データが送信できない

部品の果たす機能の裏返しになってしまう

×発想が広がらない
×壊れ方のパターンが定まらない
（影響の評価や対策の検討も十分にできない）

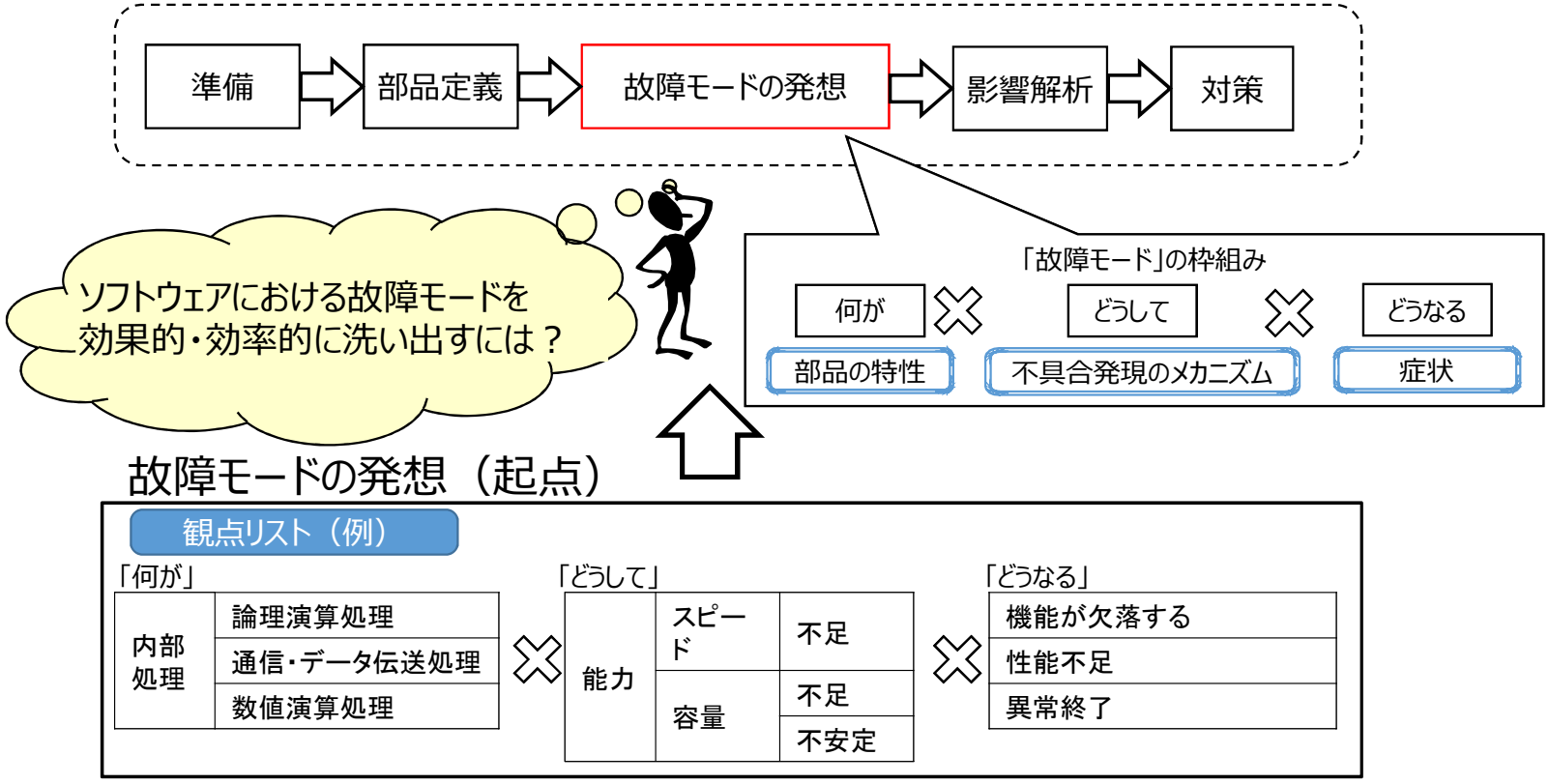
2. 限られた開発工数の中でやりきれない

ハードに比べて設計の自由度が高いため、考えるべき範囲や組み合わせが多い

⇨ 匠が最適な設計解を与えて解決してきた

SQiPシンポジウム2016発表資料を改変

ソフトウェアのFMEAと発想の観点(当社の事例)



過去トラなどをベースにしたシステマティック故障の洗い出しを実施

観点リストの期待効果

- 〝製品分野を考慮した〟観点リストを使うことで、
  - 製品の特性によって起こりやすい不具合の再発・未然防止につながる
  - 開発組織やエンジニアの特性によって起こりやすい不具合の再発・未然防止につながる

観点リストとは、故障モードを発想しやすくするための、技術的な観点(起点)を記述した目録

観点リストの開発手順(当社の事例)

（0）　“組込みソフトウェア一般向け観点リスト”を用意する

（1）　**過去の不具合を真因解析**し、その結果を3つの属性で表現する

（2）　**手順（1）に対する応用例・類推を考える**

抽象度が低い観点リスト

（3）　手順（1）と手順（2）の結果を抽象化する

（4）　“製品分野を考慮した観点リスト”に反映すべき結果を選択する

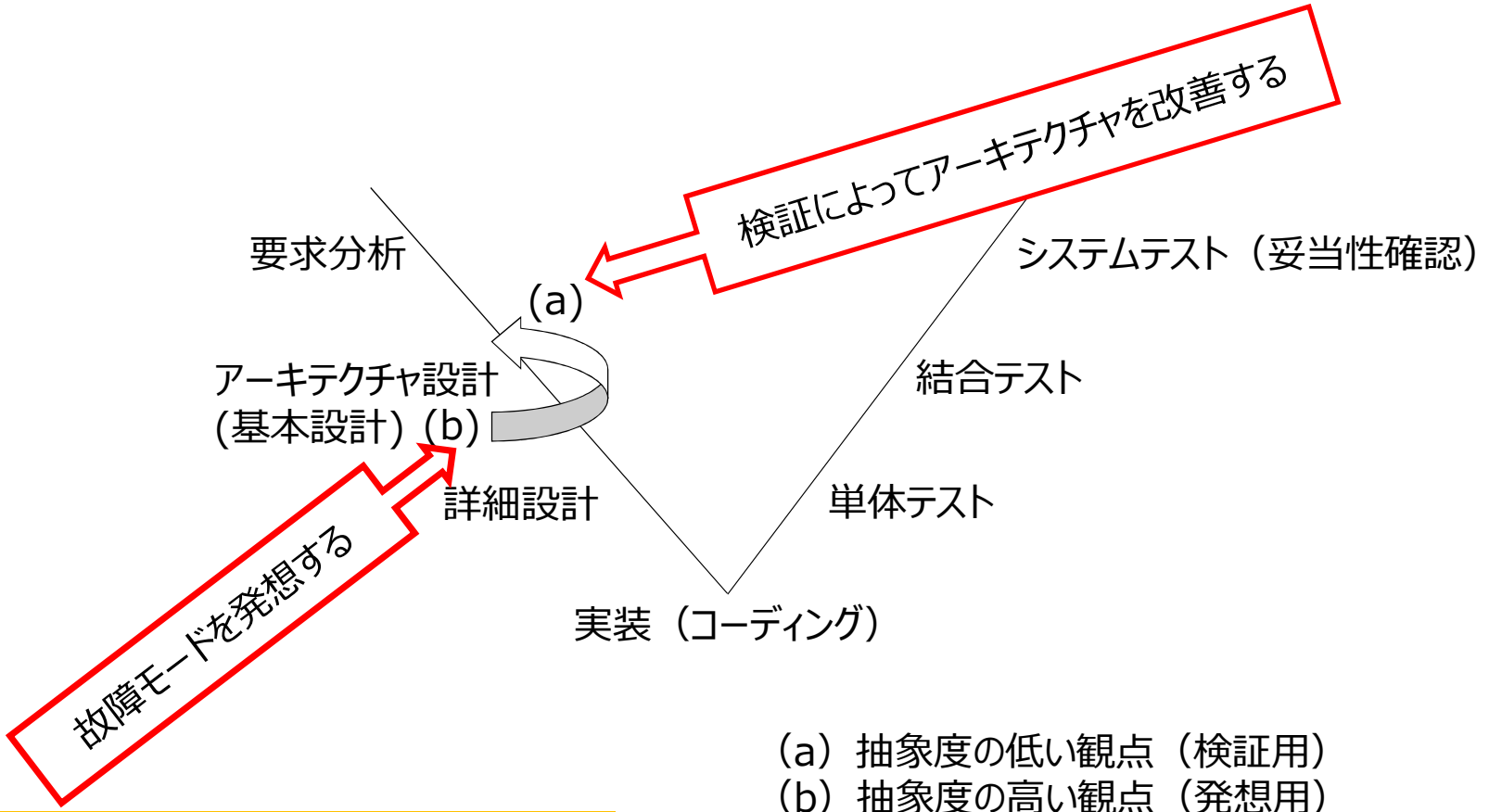
（5）　手順（0）における組込み一般向け観点リストを更新する

抽象度が高い観点リスト

- 経験や実績等をより活かすために以下の内容も手順（1）に含める
  - **エキスパートの経験や知見**

信頼性・保全性シンポジウム(第47回)発表資料を改変

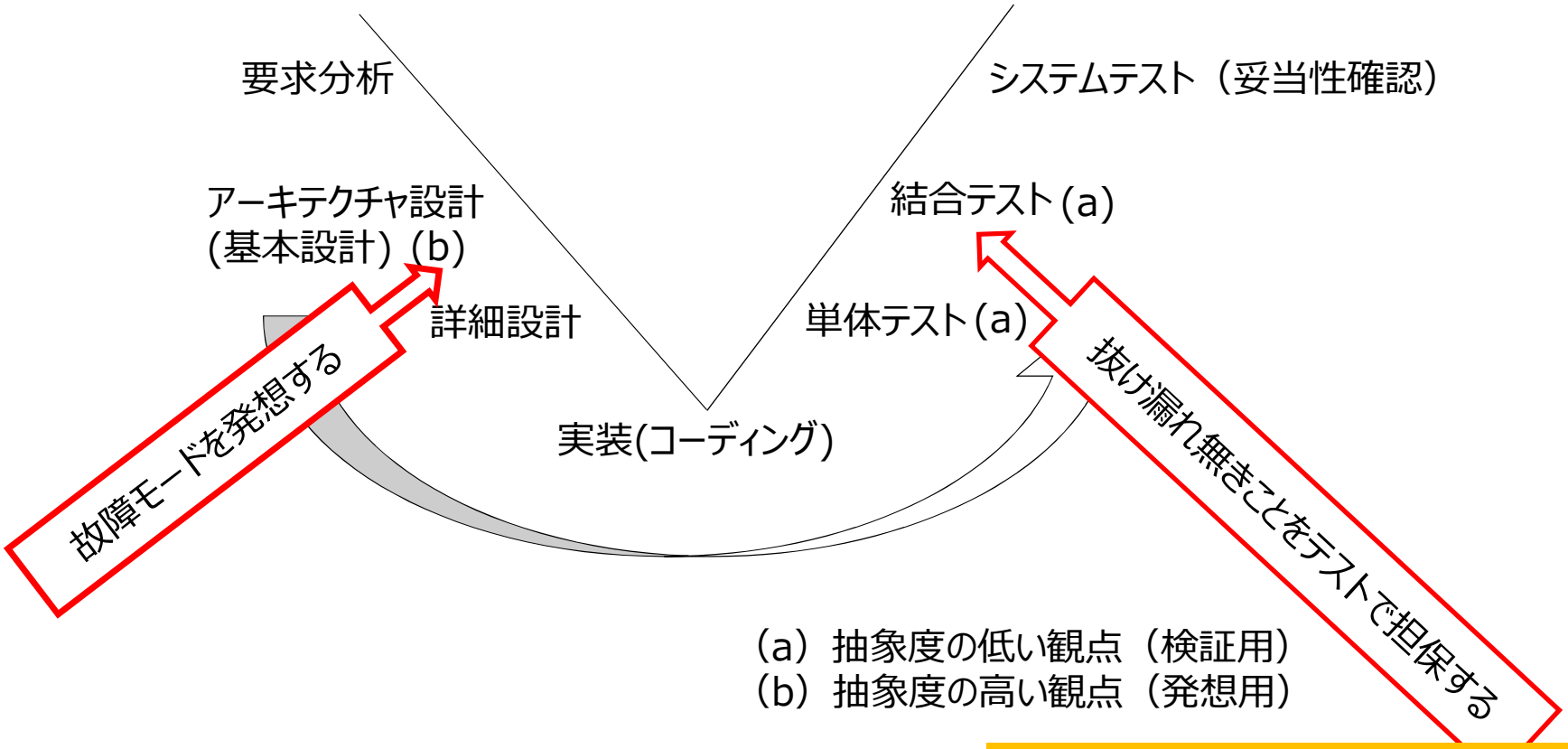
観点リストの適用ケース(1)



（a）抽象度の低い観点（検証用）
（b）抽象度の高い観点（発想用）

抽象度の高い観点リスト(発想用)で未然防止

観点リストの適用ケース(2)



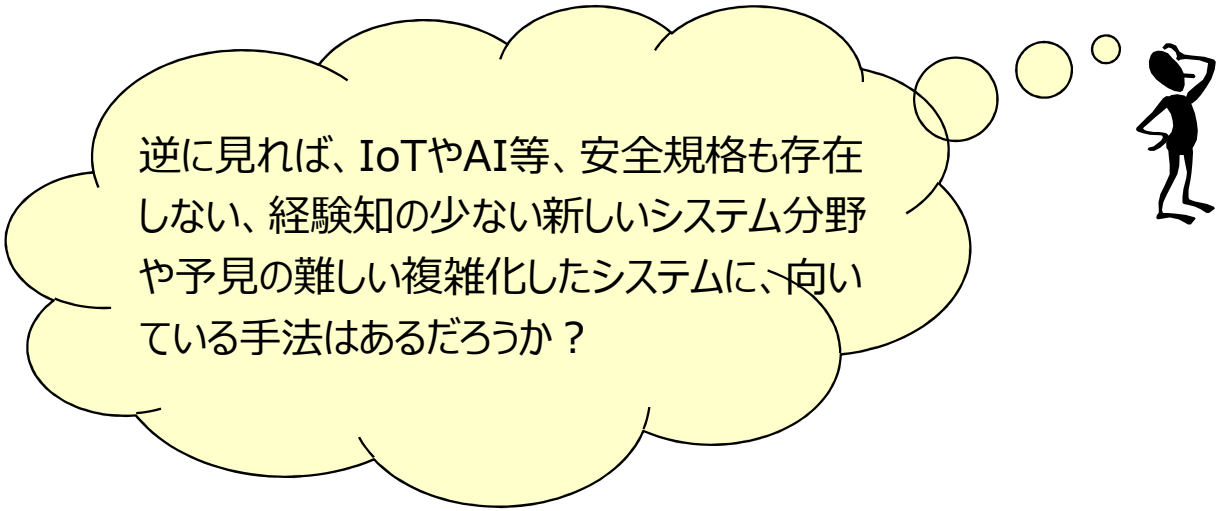
(参考)2つの観点リストまとめ

	抽象度が高い観点リスト	抽象度が低い観点リスト
メリット	未然防止につながる	発想が容易
デメリット	発想が難しい	未然防止につながりにくい
目的	発想用	検証用
適用フェーズ	V字モデルの左側上部	V字モデルの左側上部 またはテスト工程
抽象度の方針	十分に高める	具体的にする
観点の数	絞る（A4一枚程度）	制限無し（できるだけ多く）
注意事項	抽象度は徐々にあげる	観点は追加し続ける

観点リストによって、過去トラ、匠の知識、業界の慣例等を取り入れることが可能
アーキテクチャ設計では過去トラと類似トラブルも防ぎ、テストとしての十分性を補完できる

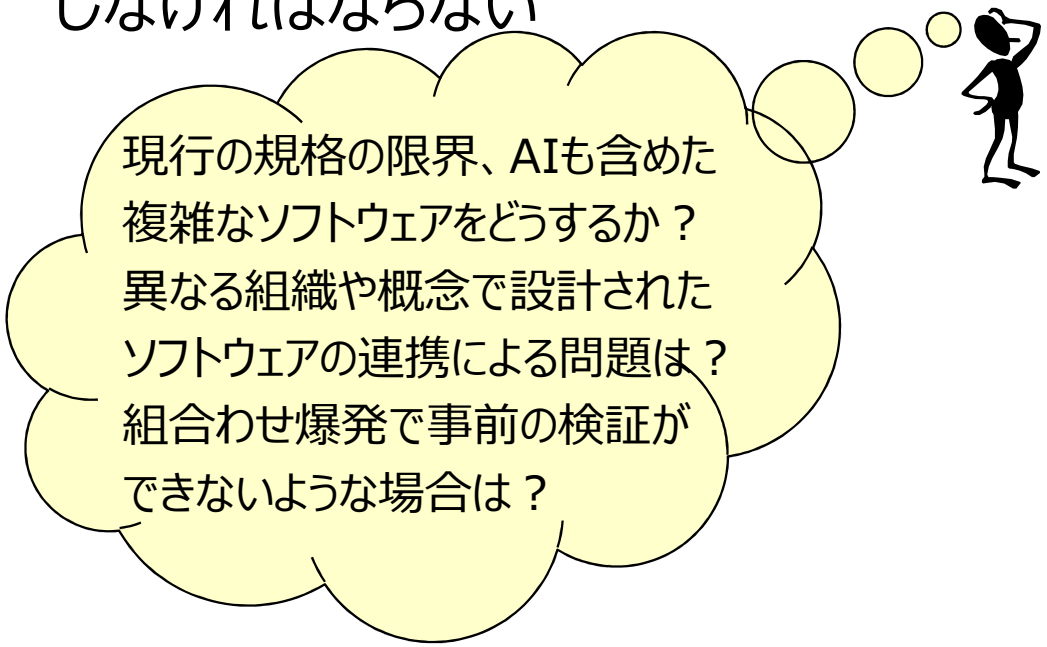
開発現場における従来の安全分析手法

- 事故やバグが起きたら対策を採るが、FMEAやFTAらの分析にはそれらの知見が反映されている　※ソフトウェアでも同じ
- この方法は、十分な過去トラがある等、経験知が高い製品では十分な安全確保のための手法と成り得る



(参考)これからの安全設計におけるソフトウェア

- IoTやAIの時代となって、大規模そして複雑化・多様化するソフトウェア
- 同時に、ソフトウェアは安全や信頼性だけでなく、サイバーセキュリティも考慮しなければならない



今すぐに答えを得るのは難しい

(参考)安全目標を網羅的に定義するには

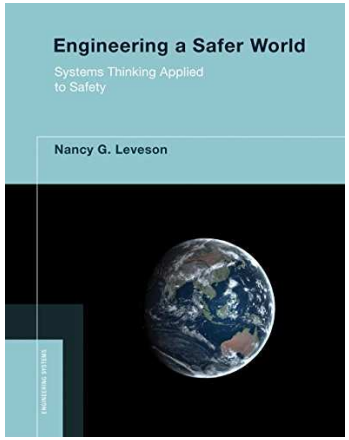
- 他のシステムや、他の技術領域に対する理解も必要になるだろう
- 異なる組織や概念で作られたソフトウェアの連携によって、安全を脅かすようなことは？

  ⇒ 範囲を限定しない安全論証は、困難を極める

これらは今後の課題。予測できない組み合わせは、STAMPの活用も検討すべきだろう

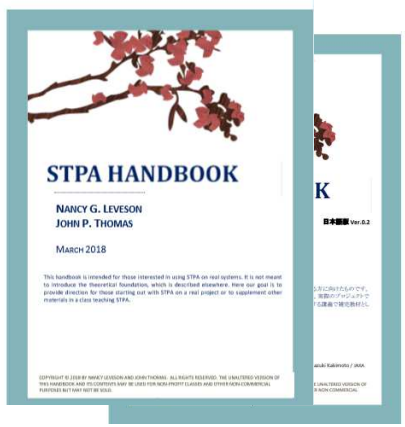
STAMP/STPAへの期待
～**システムズ理論**に基づく事故モデルとプロセス～

- 背景
  - ソフトウェアは複雑化する
  - システミックとシステマチックの違い
- STAMP/STPAのシステムズ理論とは？



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/

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

ソフトウェア集約システムの課題
～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
- 複雑システムではコミュケーションエラーが事故を引き起こす

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 × システマチック」な問題解決を


SDM NEWS

1

2013年 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、要素間の関係性によって創発する特徴を持つもの。技術システムから社会システム、人間システムまで、インタラクションを含むものはすべてもシステムです。慶應SDMで重視するのは、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

慶応義塾大学 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

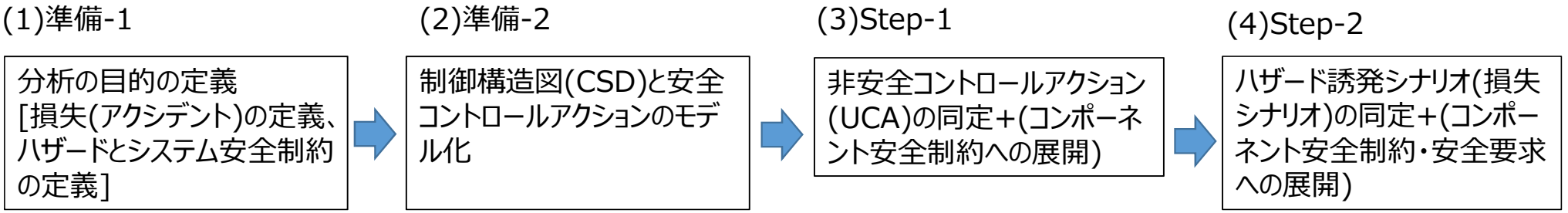
<u>システマチック・アプローチ</u>
ものごとをロジカルに分解・統合して見る

(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)
  - 安全要求を要素に分解して設計する**信頼性工学**の手法に基づいて全ての故障をなくす
  - **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
- 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、**安全制御工学**の手法で事故を防ぐ
  - 仕様の欠陥や、要素間の連係ミス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考え方である

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)を起点にしてハザード誘発シナリオを抽出し、事故を防ぐ方法を考える
2. ハザード誘発シナリオは、従来のFTA, FMEA, HAZOP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

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又は危害	H1:プラントからの有害物質の気中や地中への放出	SC1:有害物質はプラントから過失によって放出されてはならない
自動車のエアバッグ	A1:運転者の死傷	H1:衝突したのにエアバッグが開かない H2:通常走行時にエアバッグが開いてしまう H3:エアバックの異常な爆発(部品飛散)	SC1:衝突時にはエアバックが開く SC2:通常走行時にエアバッグは開かない SC3:エアバック開の際に部品を飛散させない

アクシデント、ハザード、安全制約

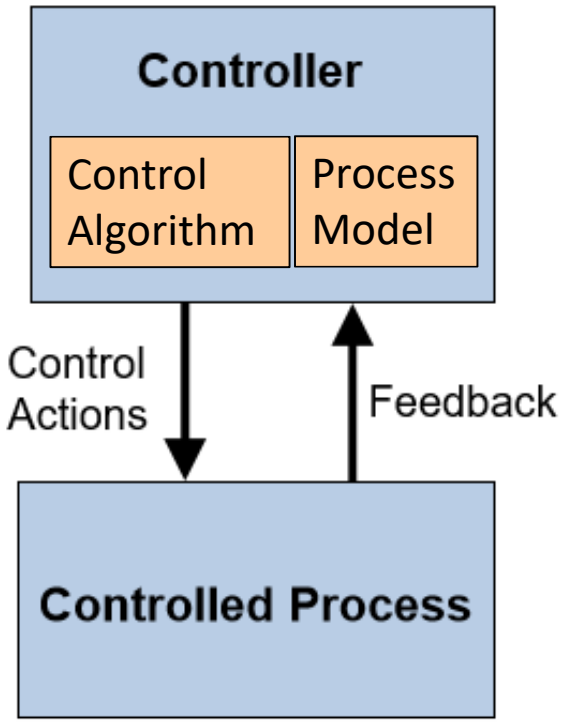
システム	損失(アクシデント)	ハザード	安全制約
ACC (自動追従運転)	L1:2台の車の衝突	H1:前方ないし後方の車との不適切な車間距離	SC1:二つの車は最小の車間距離を越えてはならない
化学プラント	L1:有害物質による人命の損失又は危害	H	
自動車のエアバッグ	A1:運転者の死傷	H 開 H か H (	

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

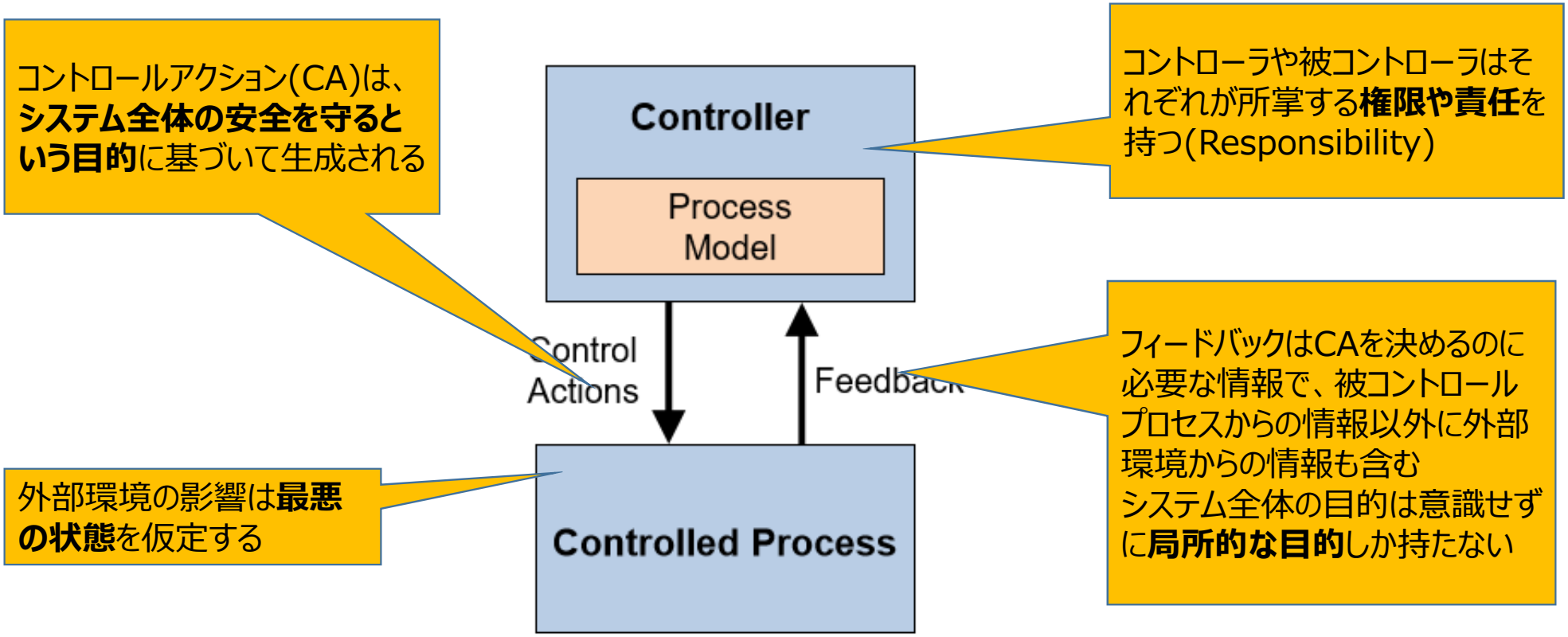
- システム外部の状態は制御できないので、最悪状態を仮定する
- ハザードを誘発する要因まで言及すると発想を狭める（例：ブレーキ故障ではなく、加速・減速という表現）
- ハザードは、システム全体を見て10個程度以内に抑える。これが多すぎるのはシステムの理解の抽象度が足りないことを意味する

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）
- 4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（4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

システムズ思考の事例：踏切の「とりこ」検知装置他

電動パーキングブレーキの事例他

「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

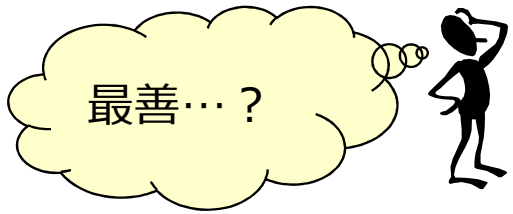
※2019年3月に最新刊(PDF)が公開予定

(参考)安全論証とSTPAに対する考察

- STAMP/STPAは…
  - 全ての部品や機能の故障(故障モード)を洗い出すといった分析に向いているわけではない
  - 例えば、ハザードに対して分析と対策が尽くされたことを説明できないかもしれない

STAMP/STPAの実施だけで安全論証は成立しない

- 国際安全規格では…
  - ある特定の制限/条件の下で、ハザードに対して分析と対策が尽くされたことを説明する
  - 新しい、今までに経験のない製品、ステークホルダー全員が素人であるような製品開発には対応しない (ex 自動運転車など)



従来の分析手法だけでは難しい

※第二回STAMPワークショップの発表スライドより引用

まとめ

1. 現行の安全規格には限界があり、新しい特性を持つ製品への対応はできていない
2. ソフトウェアだけに特別な安全論証の考え方はなく、バグがないことを含めて〝最善を尽くしているか〟がひとつの視点である
3. ソフトウェアの安全分析方法として、観点リストに〝最善を尽くす〟ための考え方(一例)を紹介
4. 新しい特性を持つ製品、これからのソフトウェアに対する安全分析にも、STAMPの考え方が役立つ可能性がある
5. STAMPは安全論証の一部をささえる

ご清聴ありがとうございました

※本発表資料の執筆協力：会津大学名誉教授　兼本茂先生