MBSE導入に失敗しないための上流設計検証手法のご紹介

2021/11/26

品質安全デザイン室
DynaSpecチーム
プロダクトマネージャー
市村健太朗

構造計画研究所 KOZO KEIKAKU ENGINEERING Inc.
株式会社　構造計画研究所
〒164-0001 東京都中野区中央4-5-3

❒Agenda

- ご提案の背景
- 形式検証とは？モデル検査とは？
- モデル検査手法を用いた検証事例の紹介
- 前半セッションのまとめ
- お知らせ

□ご提案の背景

ご提案の背景

製品開発で要求される
技術的ハードルは益々高まっている

一方で…

□ご提案の背景

- ✓ **求められる安全性も日増しに高まっている**
- ✓ **システムが複雑化すればするほど、**
  思いもよらない不具合が発生する。

❑不具合は開発工程のどこに混入されているか？

■上流工程で多くの不具合が混入されている。

✓上流工程で混入された不具合の検出が後工程になる程膨大なコストがかかる。

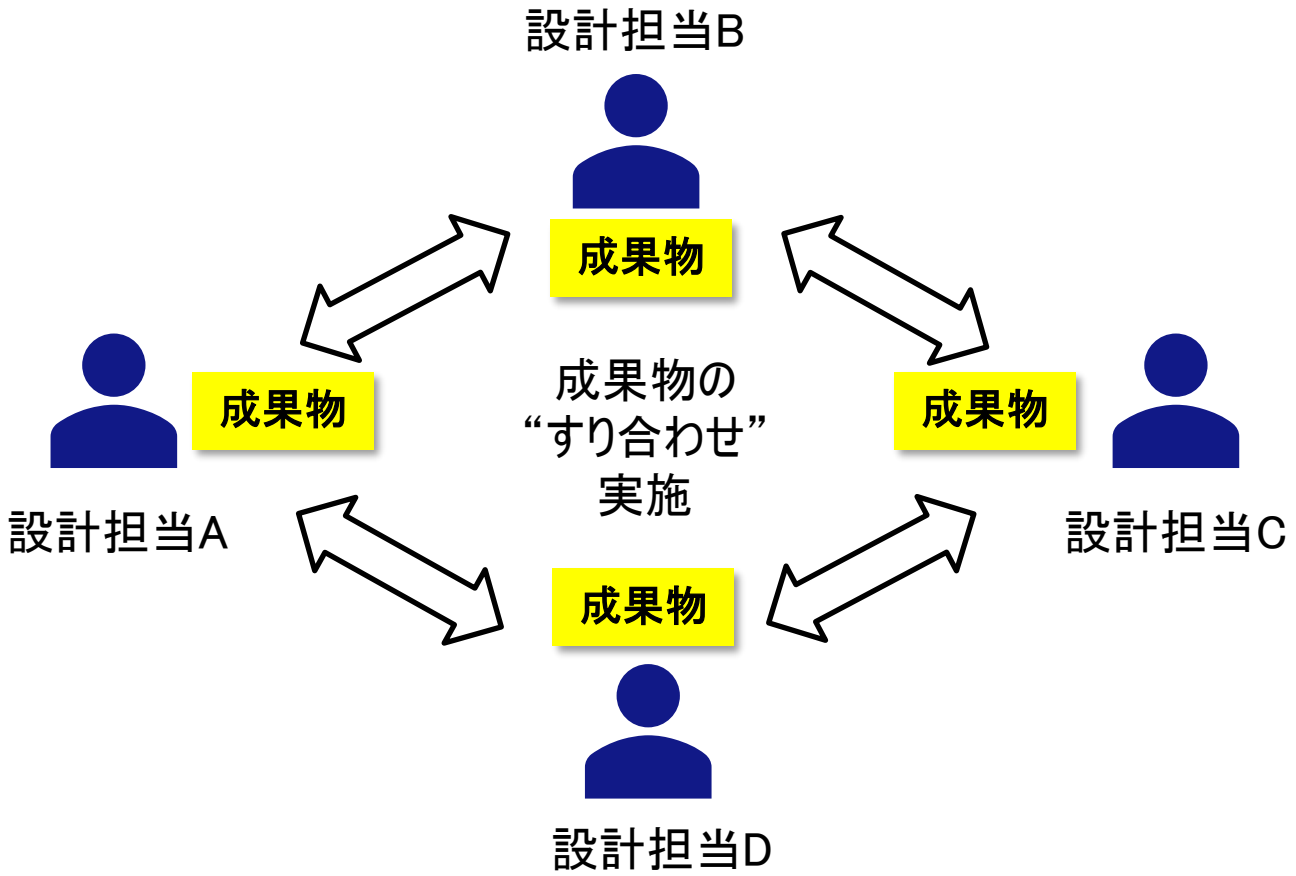
■IPAの業界アンケート「組込みソフトウェアに関する動向調査(2018)」

設問	順位	回答
組込みソフトウェア開発の課題	1位	設計品質の向上
モデルベース開発技術の導入目的	1位 2位	品質向上のため 上流工程での検証のため
不具合の原因	1位 増加	ソフトウェアの不具合 他製品・他システムとの接続に起因する不具合 表示・操作・使用環境等使用状況に関連する不具合
モデルベース開発・ツールの導入状況	1位 2位 3位	試験／評価ツール（シミュレータ等） 状態遷移モデル（図／表） UML SysML

❒従来開発とその課題

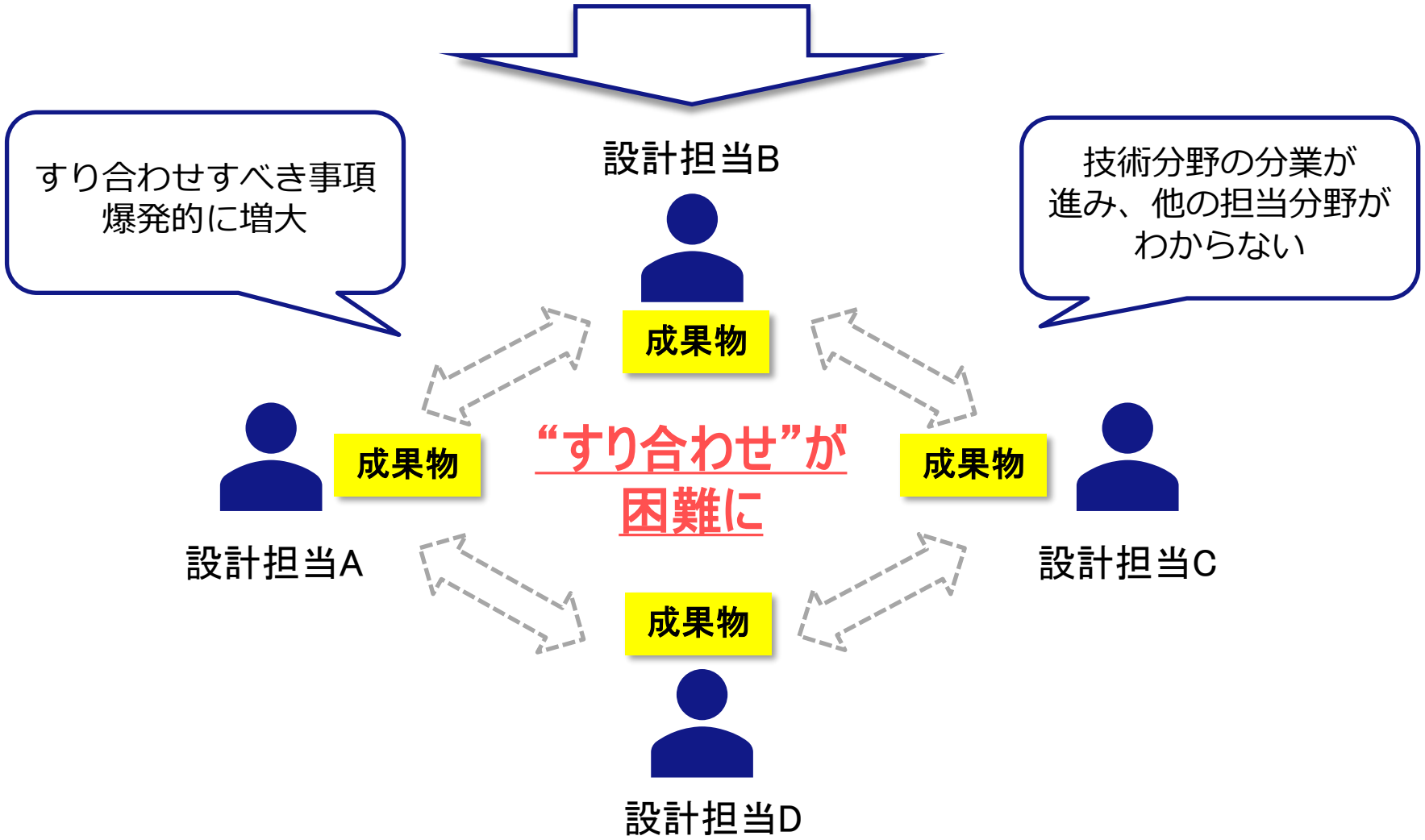
■従来開発(＝すり合わせ開発)

❒担当製品・ドメインの成果物を”すり合わせ”ながら、製品の品質を向上させる手法は、日本の製造業の強みであった



❒従来開発とその課題

システムの大規模複雑化に伴って…



□システムズエンジニアリング実践の必要性

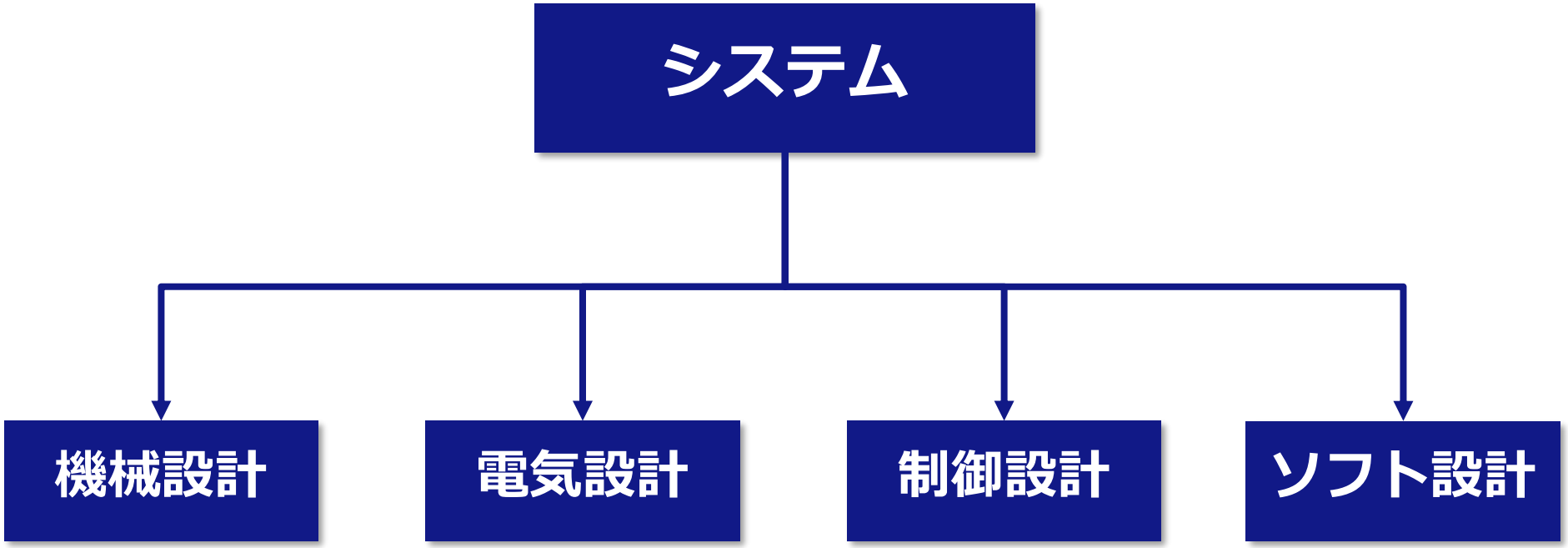
全体を俯瞰したい上で
要求分析から設計、開発までを
一貫して行う必要がある

⇩

システムズエンジニアリングの実践

□システムズエンジニアリングとは？

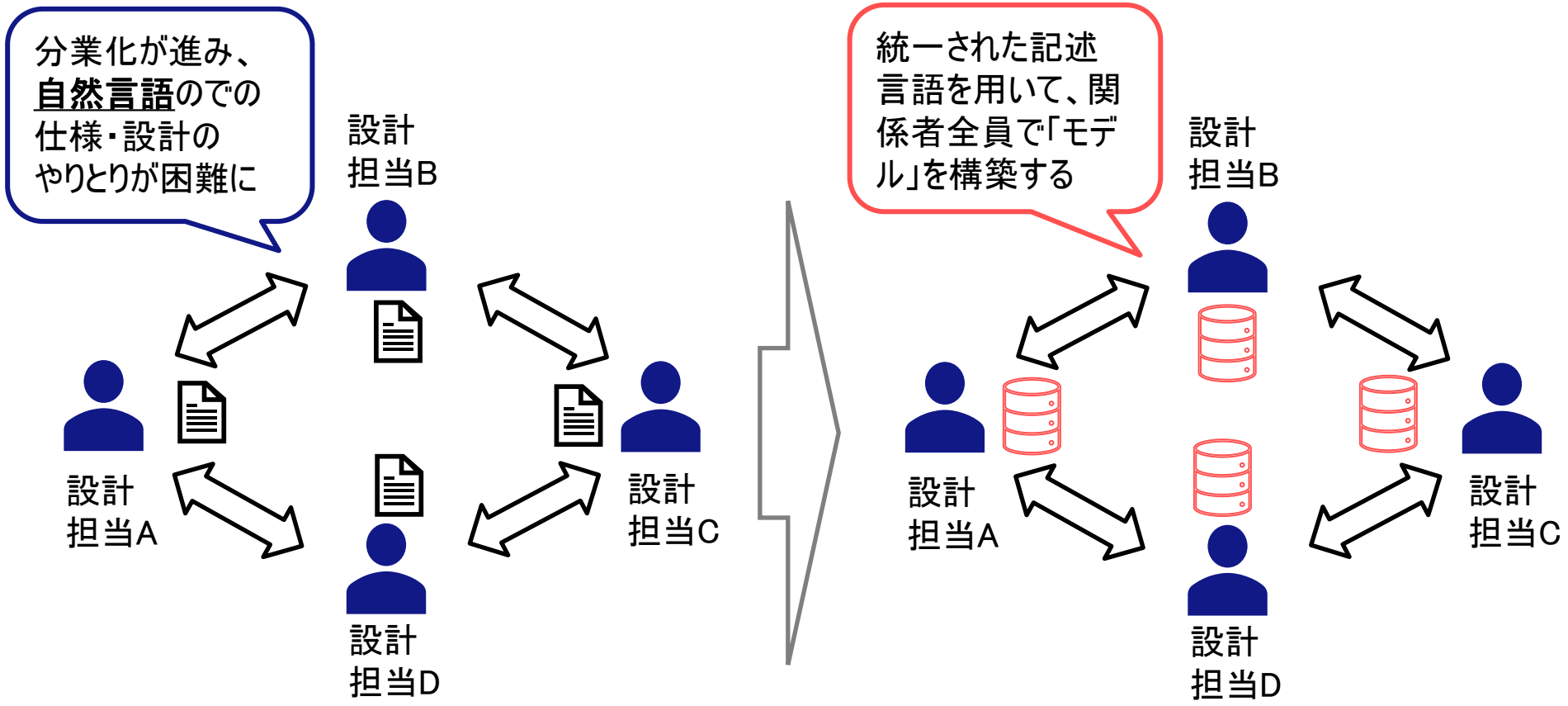
- 「システムを成功させるための複数の専門分野にまたがるアプローチと手段」
  - by INCOSE（The International Council on Systems Engineering）
- これまで各分野それぞれで行っていた開発業務を、他の担当部署と協業して製品の品質を向上させるため、**<u>システムの全体を把握した上で、各分野に適切に切り分ける</u>**こと。



❒MBSEとは？

■ Model Based Systems Engineering

- □ システムズエンジニアリングを実践するためのアプローチ
- □ ドキュメントのかわりに、統一された記述言語で対象システムの「モデル」を構築し、開発に携わる関係者がアクセス・アップデートし、開発を進めていくプロセス



❒MBSE導入ステップと一般的な取組み

Step.1

[MBSEの導入プロセス検討]

- 一般的な取組み
  - 一般的には下記のMBSEプロセスが参照されることが多い
  - Harmony for Systems Engineering
    - ✓ IBM社の提供するMBSEプロセス
  - OOSEM（Object-Oriented Systems Engineering Method）
    - ✓ INCOSEが提供するMBSEプロセス

Step.2

[モデリング言語の習得]

- 一般的な取組み
  - SysMLの習得
  - 人材教育によって、関係者全員がSysMLを読み書きできるようにする必要がある

❒MBSEの導入メリット

■コミュニケーションの促進⇒全体を見通した設計

- ❒関係者間で共通の記述言語でやりとりすることで、効率的なコミュニケーションが促進される。
- ❒コミュニケーションの促進により、**全体を見通した設計**が出来るようになる。

■システムの要求・振る舞いの可視化

- ❒システムの構成要素である各サブシステムが満たすべき**要求と振る舞いを簡易に把握**することができる。
- ❒システム開発プロセスの中で**要求のトレーサビリティ**が確保される。

❒MBSE導入における課題

Step.1 Step.2

[MBSEの導入プロセス検討]

- 一般的な取組み
  - 一般的には下記のMBSEプロセスが参照されることが多い
  - Harmony for Systems Engineering
    - ✓ IBM社の提供するMBSEプロセス
  - OOSEM（Object-Oriented Systems Engineering Method）
    - ✓ INCOSEが提供するMBSEプロセス

[モデリング言語の習得]

- 一般的な取組み
  - SysMLの習得
  - 人材教育によって、関係者全員がSysMLを読み書きできるようにする必要がある

SysMLを描くだけで満足してしまっていませんか…？

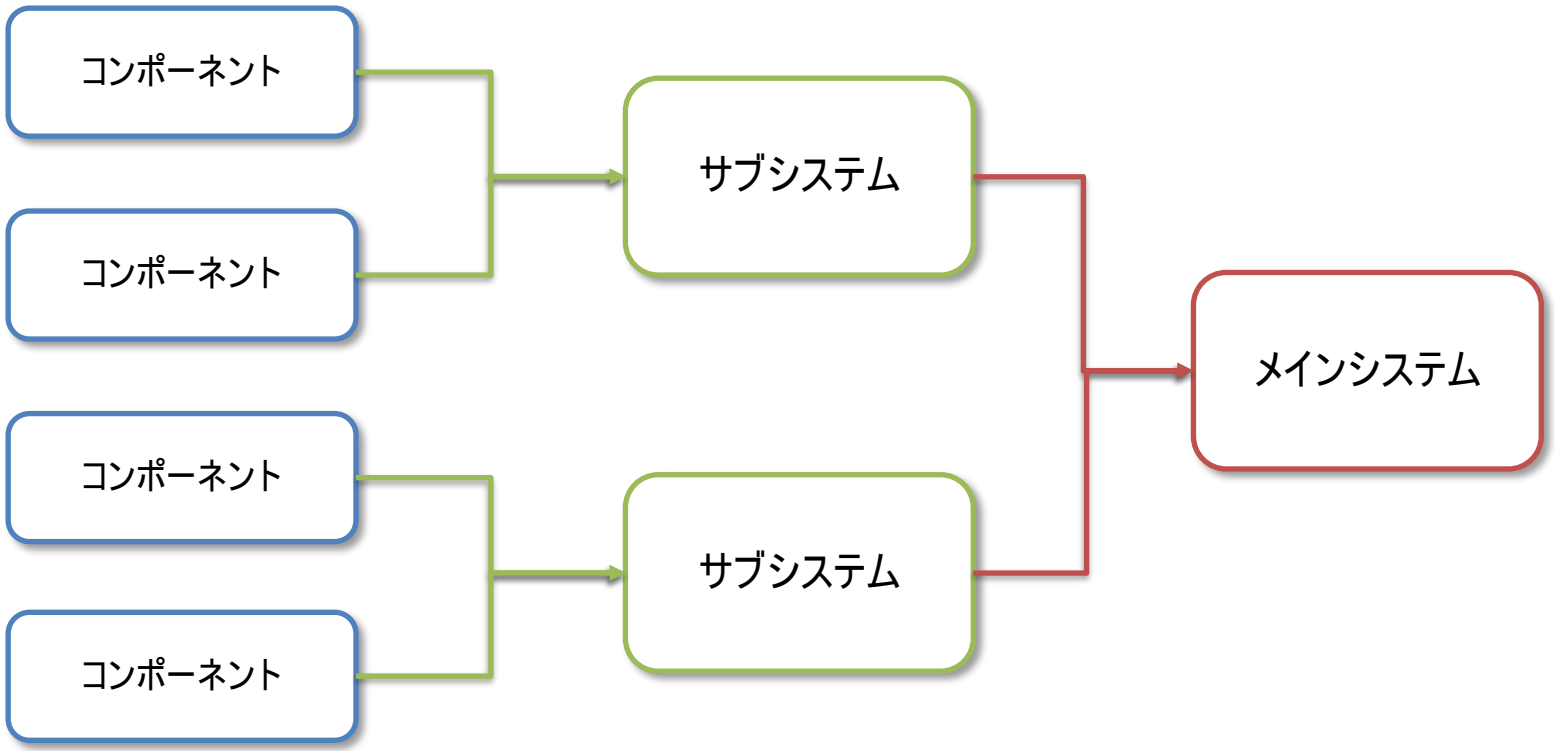
❑お客様からよく聞く課題

設計が要求仕様を満たしているのかチェックが難しい

複数の制御の組み合わせで機能を実現している制御のチェックが難しい

□システム設計で起こり得る問題

■システム設計ではシステムを階層化し、各サブシステム、コンポーネントが満たすべき要求を導出した上で設計を行う。

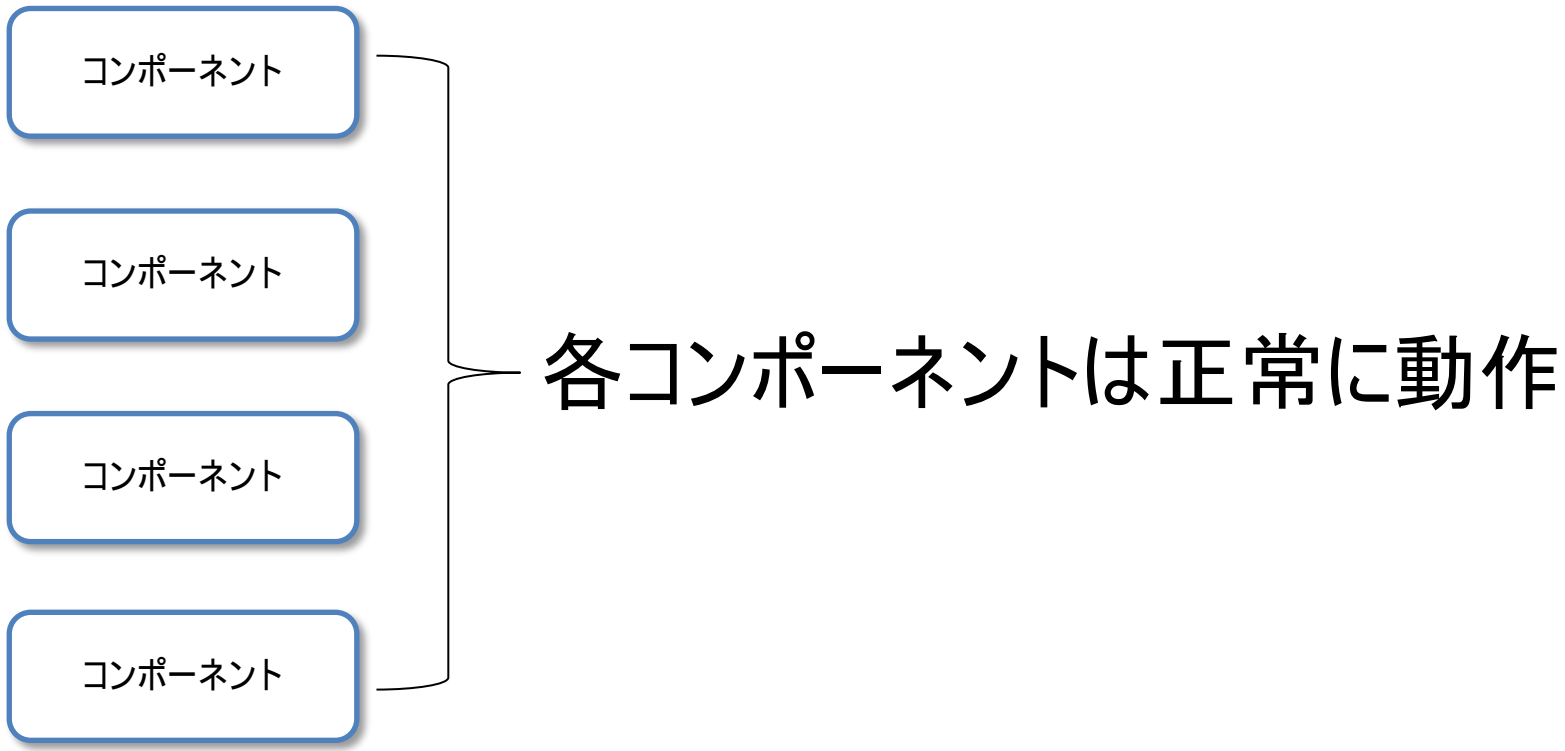


◻なぜシステムレベルでの設計・検証が重要なのか？

コンポーネント単体が
正常に動作する

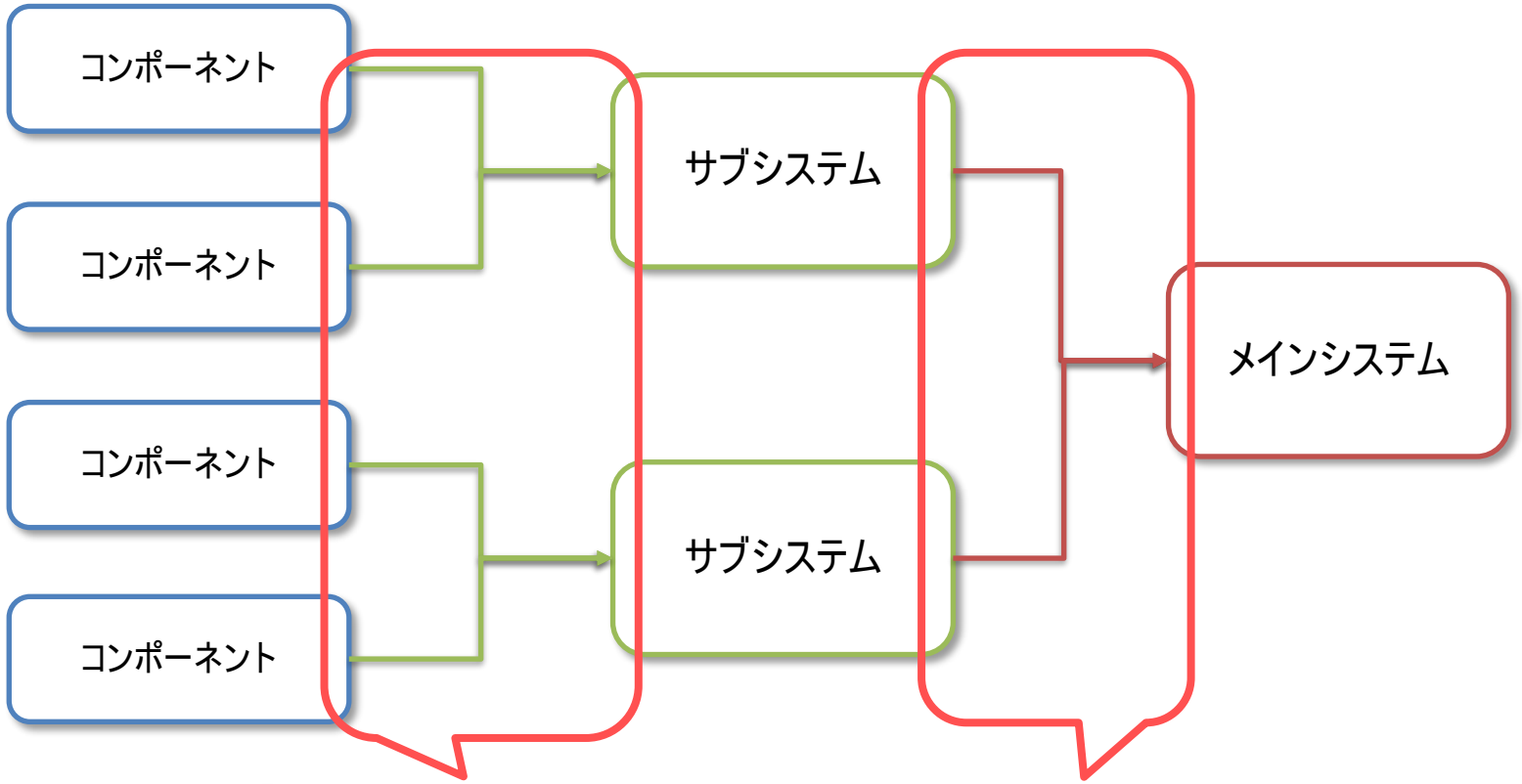
≠

統合システムが
正常に動作する



□なぜシステムレベルでの設計・検証が重要なのか？

コンポーネント単体が正常に動作する ≠ **統合システムが正常に動作する**



サブシステム・メインシステムに統合する際に
不具合は発生しやすく、これらは人手では発見困難

❑SysMLモデル検証の必要性

SysML等を用いて全体を俯瞰した設計を行い、
人手でチェックするだけでは、発見困難な不具合は見逃してしまう

⇩

人手やレビューではチェックが困難なシステム設計を
正しく検証するためにはどうすれば良いか？

⇩

**SysMLモデルの検証を属人化せず、
「システマティックに」かつ「厳密に」検証する手法が必要**

❒MBSE導入成功への最後のステップ

Step.1 | Step.2 | Step.3

[MBSEの導入プロセス検討]

- 一般的な取組み
  - 一般的には下記のMBSEプロセスが参照されることが多い
  - Harmony for Systems Engineering
    - ✓ IBM社の提供するMBSEプロセス
  - OOSEM（Object-Oriented Systems Engineering Method）
    - ✓ INCOSEが提供するMBSEプロセス

[モデリング言語の習得]

- 一般的な取組み
  - SysMLの習得
  - 人材教育によって、関係者全員がSysMLを読み書きできるようにする必要がある

[課題]

- **SysMLモデルを作っても、設計が要求を満たしているかがわからない**

⇩

[KKEの提案]

作成したSysMLモデルを

- 形式検証・モデル検査技術

で検証する

❒設計情報の検証について

- 「ドメイン毎に細分化されたのち、各設計情報を物理シミュレータ等で検証すること」は一般的になってきている。
- 「しかし、システム設計レベルの抽象度の高い設計情報をレビュー以外の検証手法で検証することはまだ一般的ではない。
  - ❒モデリングツールの機能を用いてシミュレーションを行う場合もある
  - ❒但し、**シミュレーションだけでは抽出しきれない不具合がある**
    - ➡システム設計段階で不具合を抽出できないと、
      - ☑不具合が混入した設計情報を下流に流してしまい、大きな手戻りが発生する
      - ☑テスト工程でも発見困難な不具合を抽出できず、重大なインシデント発生に繋がる

⇩

シミュレーションよりも強力な検証技術
すなわち「モデル検査」技術が必要

□SysMLの一般的なモデル化プロセス

要求定義・分析

- 要求図
- ユースケース図

✓ 利害関係者や要求から上記の図を用いて、機能要求や非機能要求を導出する。

システム設計

論理設計

- ブロック定義図
- 内部ブロック図
- ステートマシン図
- アクティビティ図

✓ 機能に着目して、システムを分割し、システム間の接続関係、インターフェースを検討する。
✓ 必要に応じてシステムを階層化し、サブシステムが満たすべき要求(制約)を検討する。
✓ 前工程で導出した要求に対し、振る舞いを検討する。

物理設計

- ブロック定義図
- 内部ブロック図
- パラメトリック図

✓ システムの物理構成を検討する。
✓ 前工程で検討した論理構成(機能構成)を物理構成に割り当てる。

❑SysMLの一般的なモデル化プロセス

要求定義・分析

- 要求図
- ユースケース図

システム設計

論理設計

- ブロック定義図
- 内部ブロック図
- ステートマシン図
- アクティビティ図

物理設計

- ブロック定義図
- 内部ブロック図
- パラメトリック図

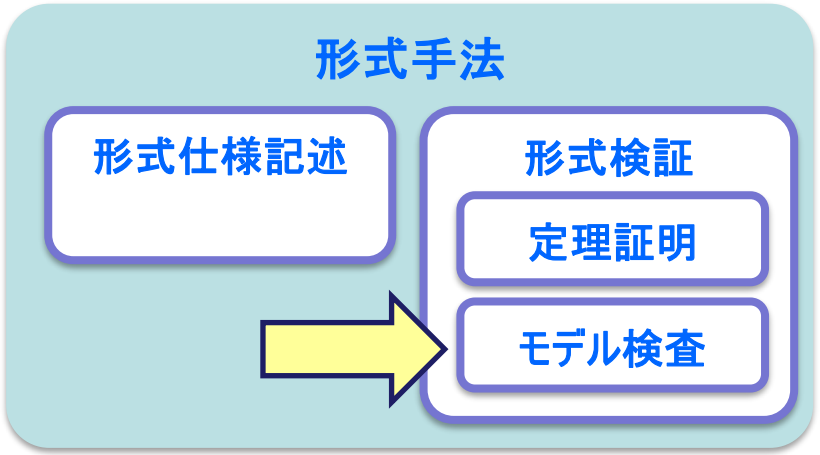
システム設計の論理設計、
すなわち、システムの振る舞いを検証する手法
「モデル検査」について本日はご紹介します

□形式検証とは？モデル検査とは？

❒形式検証とは？モデル検査とは？

■そもそも形式検証とは？

- ❒形式手法の中の検証に関する手法



■形式手法

❒形式仕様記述

➡仕様や設計を、あいまいな自然言語ではなく厳密な文法を持つ形式言語できちんと書く。

☑不正確さ，不整合も表面化する。システムに対する理解が深まる

❒形式検証

➡「きちんと」書いた仕様、設計を元に、厳密な分析・検証を行う

☑定理証明：関数の正しさをテストによらず数学的手法で証明する

☑モデル検査：システムの振る舞いを網羅的に再現し、検査する

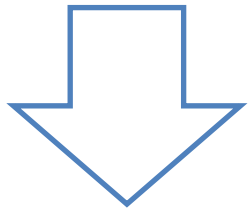
■ シミュレーションとモデル検査の違い

□ シミュレーション

- 与えた条件・シナリオでどのような結果に繋がるかを見る
- シナリオを人手で作成すると多大な工数がかかり、抜け漏れも発生する

□ モデル検査

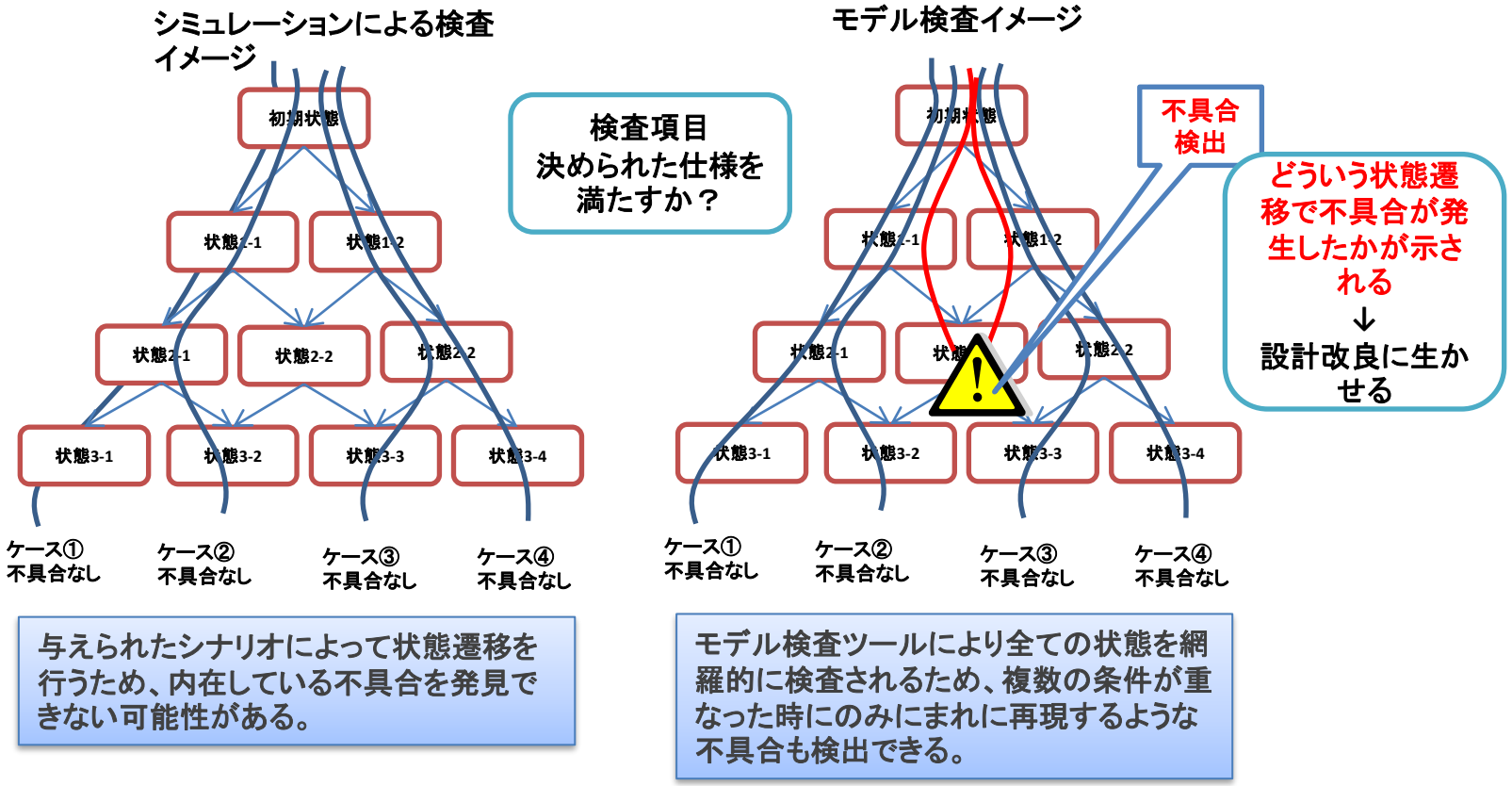
- 確率は低くても理論的に起きる可能性があるかどうかを検出する。
- シナリオは非決定的要素を含む独立プロセスとしての外部環境モデルを与えることで自動的に網羅シナリオで検査される。



理論的に起き得る振る舞いを
自動的・網羅的に再現させて不具合を検出する

□シミュレーションとモデル検査の違い

■「シミュレーション」と「モデル検査」



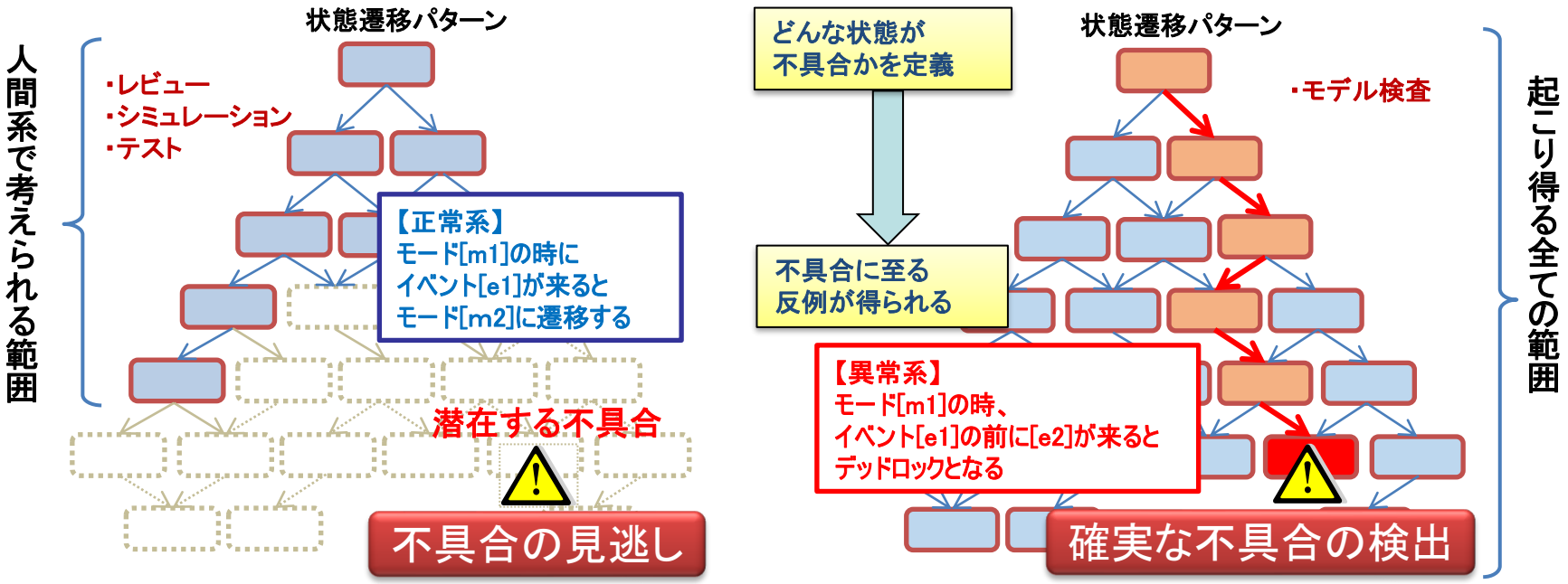
□従来手法（レビュー、シミュレーション、テスト等）とモデル検査の違い

■ 従来手法（レビュー、シミュレーション、テスト等）による品質保証の限界

- □ 人間系に依存する従来の品質保証では、複雑な要因が絡んで発生する可能性のある不具合を見逃してしまう！（特に異常系の動作）

✓ 人間が考えられる範囲でのシナリオに基づく検証

✓ モデル化したシステムの振る舞いの網羅探索による検証（モデル検査）

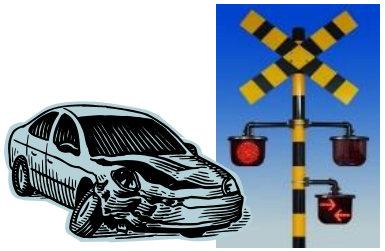


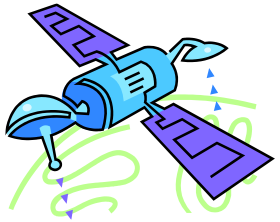
❒形式検証の適用対象

- 適用効果が高い分野
  - 高い安全性が要求される**交通システム**
    - 自動車のECU
    - 鉄道の列車制御システム（連動装置、信号制御装置）　・・・・
  - 1回の不具合発生が致命傷になる**航空宇宙システム**
    - 航空管制システム、人工衛星制御システム　・・・・
  - 不具合による社会的影響度が甚大な**社会インフラ・基幹システム**
    - 原子力発電所制御システム、銀行オンライン情報システム　・・・・

セーフティクリティカルシステム

ミッションクリティカルシステム





- 形式手法が推奨されるソフトウェア開発国際標準
  - 機能安全：IEC 61508（一般）、ISO 26262（自動車）、IEC 62279（鉄道）他

不具合の発生が許されないシステムの品質保証の有効手段

❑モデル検査が適用できる対象システムの特性

適用可能、効果大	適用性・効果に制限あり
離散	連続、ハイブリッド
論理	物理・熱・電気
抽象	詳細
ソフトウェア	ハードウェア
開発上流フェーズ	開発下流フェーズ
全体システム・統合制御	サブシステム・単体制御
シーケンス制御	フィードバック制御
非リアルタイムシステム	リアルタイムシステム
MBSE	MBD
システマティック故障	ランダム故障
ステートマシン図、アクティビティ図	パラメトリック図、微分・運動方程式
ミッション/セーフティクリティカル	ノン　ミッション/セーフティクリティカル

※ 限定的に適用することは可能

モデル検査に合ったシステムへの適用で最大限の効果を発揮！

□モデル検査手法を用いた検証事例の紹介

❒形式検証の適用対象

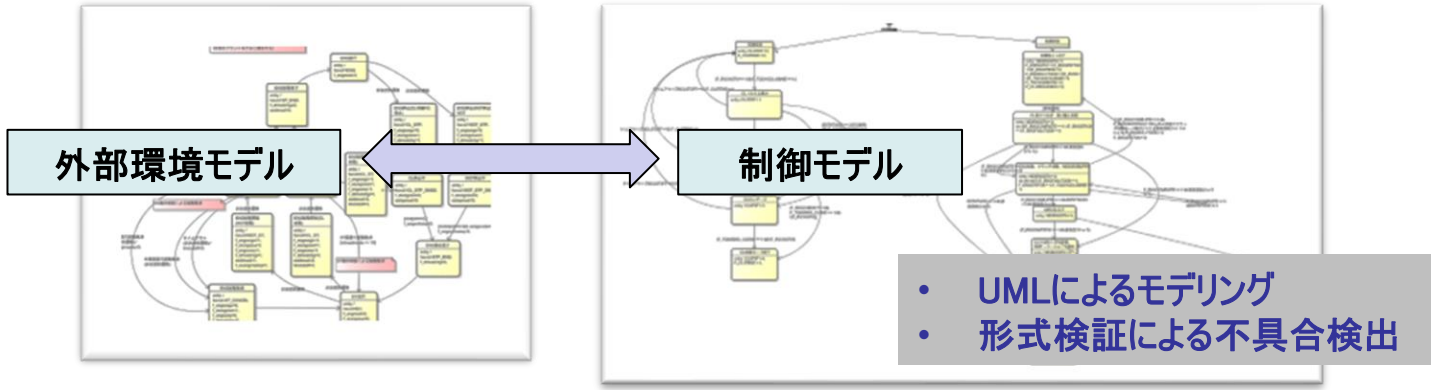
【参考】 KKEの形式検証適用事例

- どのようなシステム、スコープ、目的でモデル化・検証が行われているかの事例

■ 自動車分野
- HV の走行モード切替制御（エンジン駆動⇔モーター駆動）の検証
- EV のシステム起動／停止制御の検証

■ 鉄道分野
- 踏切の警報／遮断制御の検証

■ 航空宇宙分野
- 自律飛行安全システム（飛翔体の位置推定ロジック）の検証
- 自律飛行安全システム（飛行中断判定ロジック）の検証



❑～ 自動車OEM事例 ～　形式検証導入の背景

■顧客：自動車OEM

■背景／課題：

- コンセプト通りに制御仕様が書けているかのチェックが困難
- 特に複数の制御の組み合わせで機能を実現している制御

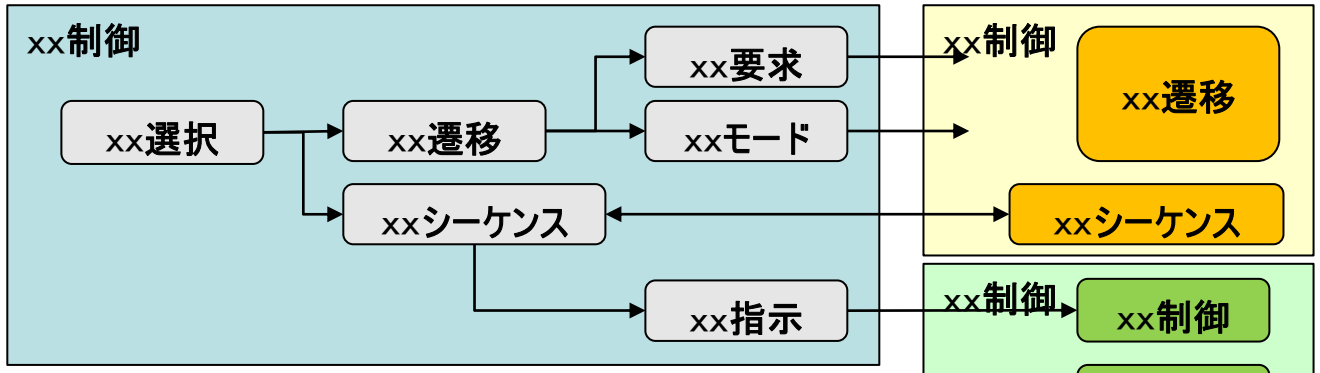
① **HVの走行モード切替制御（エンジン駆動走行⇔モーター駆動走行）**

- 駆動力マネジメント制御、エンジン制御、モーター制御・・・等

② **ＥＶのシステム起動／停止制御**

- エネルギーマネジメント制御、バッテリー制御、ＤＣ／ＤＣ制御・・・等

コンセプト：
制御構造

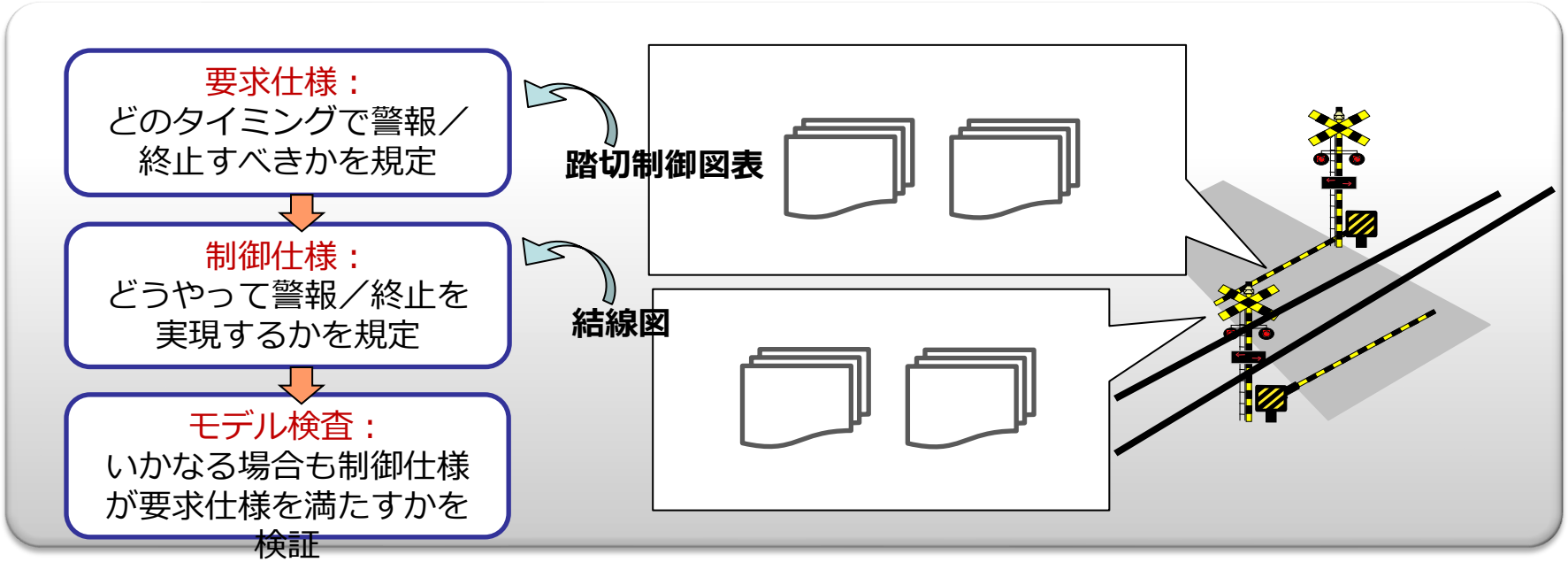


モデル検査による制御設計上の不具合検出を実現

❐～ 鉄道事業者事例 ～　大規模駅構内踏切の警報／遮断制御

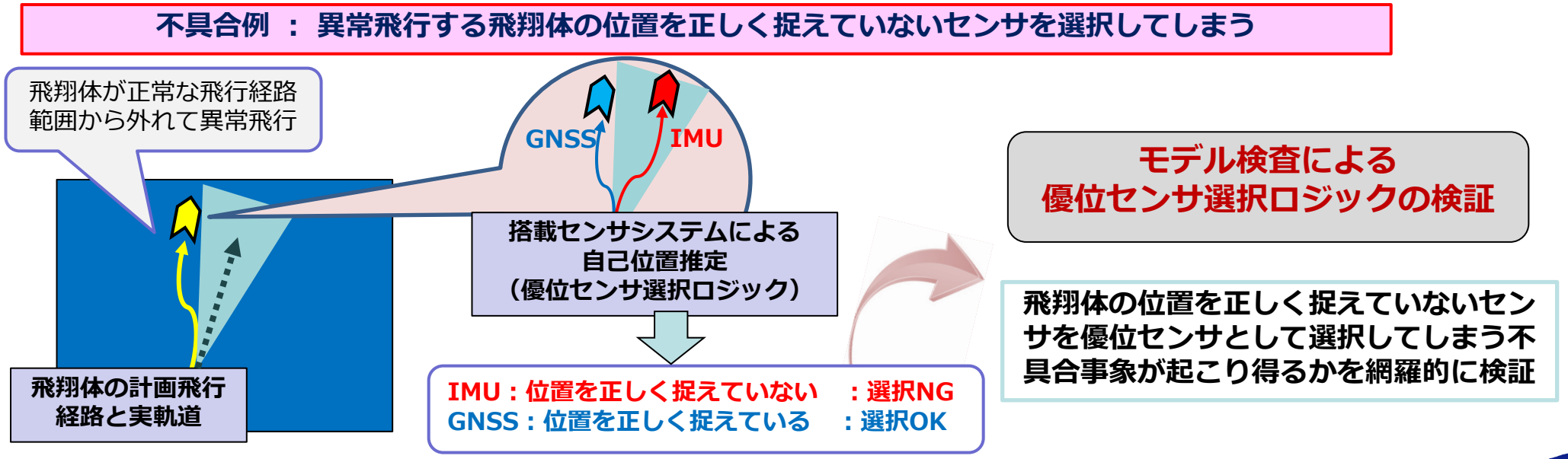
- 顧客：東日本旅客鉄道株式会社
- 背景と目的
  - 踏切の警報等の制御論理は「踏切制御図表」「結線図」によって表現される。
  - 人間系では輸送障害時に発生するような極めてまれな運行状況までを想定した安全性の検証は困難である。

「モデル検査」という論理の変化を網羅的に検査することができる技術を用いて、踏切制御論理の安全性を向上させる。



❑～ 航空宇宙事例 ～　冗長センサシステムによる飛翔体の位置推定

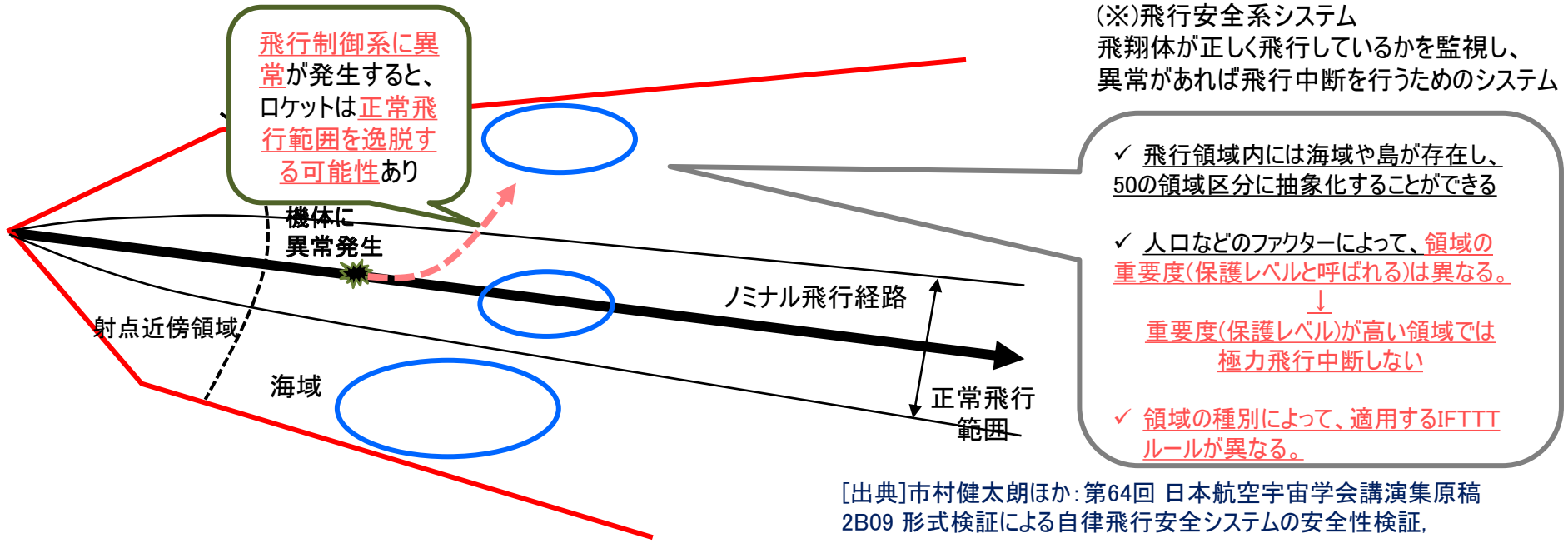
- 顧客：（国研）宇宙航空研究開発機構（JAXA）様、宇宙技術開発（株）（SED）様
  - 経済産業省からの委託事業**「宇宙産業技術情報基盤整備研究開発事業（民生部品等を活用した宇宙機器の軌道上等実証）」**の成果
- 形式検証の適用対象：自律飛行安全システム
  - 自律飛行する飛翔体の位置推定を、 複数のセンサ（GNSS：衛星測位/IMU：慣性航法）を利用した冗長センサシステムにより実現する。
    - ➡ **「優位センサ選択ロジック」**により複数のセンサの中から正しい位置情報を示す優位センサを選択する。
  - 故障・データ異常が起きても安全側に制御する。
    - ➡ 避けたい事象（検証目的）：**飛翔体の位置を正しく捉えているセンサを選択できていない。**



❑～ 航空宇宙事例 ～　自律飛行安全システムの検証

[検証概要]

- 自律飛行安全システムには、ロケットシステムから提供される各種情報に基づき地上の管制者が判断する**10の飛行中断クライテリア(IFTTTルール)**がソフトウェアとして実装されている。
- この自律飛行安全システムに対し、本件では以下の検証を行い、**不具合が存在しないことを明らかにした。**
  - **すべてのシステムの中で2故障の範疇で、意図しない飛行中断、飛行続行が起こらないこと**
  - **飛行安全系システム内(※)で1故障が発生しても、ミッション(ロケットを周回軌道にのせること)が達成できること**
  - **すべてのシステムの中で2故障が発生しても安全性が担保されること**



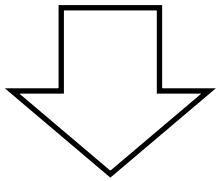
(※)飛行安全系システム
飛翔体が正しく飛行しているかを監視し、異常があれば飛行中断を行うためのシステム

- ✓ 飛行領域内には海域や島が存在し、50の領域区分に抽象化することができる
- ✓ 人口などのファクターによって、領域の重要度(保護レベルと呼ばれる)は異なる。
  ↓
  重要度(保護レベル)が高い領域では極力飛行中断しない
- ✓ 領域の種別によって、適用するIFTTTルールが異なる。

[出典]市村健太朗ほか:第64回 日本航空宇宙学会講演集原稿
2B09 形式検証による自律飛行安全システムの安全性検証,

❒前半セッションのまとめ

- システムが複雑化していく中で、MBSEに取り組み始めた企業様が増えている。
- しかし、SysMLモデルを構築することは出来ても、システム設計レベルでこれを検証することはまだまだ一般的ではない。
- また、シミュレーションだけでは抽出し切れない不具合が実際に存在する



システム設計に対して、「モデル検査」技術を適用することで、
・不具合を残したまま下流の工程に進み、手戻りが発生してしまうこと
・発見困難な不具合を見逃して、重大なインシデントが発生してしまうこと
を防ぐことができる。

□後半セッションの概要

■Enterprise Architect ＋ モデルベース形式検証ツールDynaSpecの導入により、設計品質、開発効率を向上する取組みについてご紹介致します。

■サンプルモデルとデモを通して、以下について紹介致します。

- □対象システムをSysMLモデル化する過程
- □更に、DynaSpecを用いてSysMLモデルを検証するための方法、過程

□～モデルベース形式検証ツールDynaSpec「個別」無料体験セミナーについて

■12月6日から、**期間限定で**弊社モデルベース形式検証ツールDynaSpecのオンライン無料個別体験セミナーを実施致します。

- □お申込みは**12月1日**から開始致します。
- □https://kke.lmsg.jp/v2/seminar/11672/jWW39c25

【オンライン開催】
モデルベース形式検証ツール『DynaSpec』
「個別」無料体験セミナー

～ご参加の方皆様に「DynaSpec」
無償体験版をご提供します！～

12月1日（水）8:00 に受付開始