

Model-Based Designを活用したドア系製品の ソフトウェア開発事例

株式会社アイシン ボデーシステム開発部

伊神範光

CONTENTS

弊社の紹介

説明概要

開発工程紹介

今後の期待

まとめ

会社概要

社 名：株式会社アイシン

事業内容：自動車部品、エネルギー・住生活関連製品の製造・販売

設 立：1965年8月31日

本 社：愛知県刈谷市朝日町2丁目1番地

資本金：450億円

従業員数：単独 35,610名

連結 116,649名

連結子会社199社（日本、北米、欧州、アジア、南米）



2023年3月31日時点

事業内容

主な事業領域として「自動車部品の提供」と
「住生活・エネルギーの製品」まで幅広く扱っ
ています

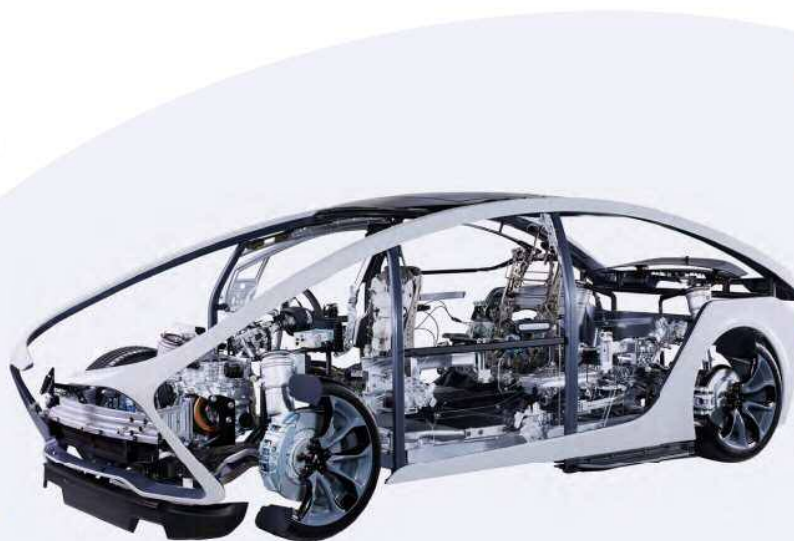
製品・事業

自動車部品ラインアップ

パワートレイン



CSS



車体



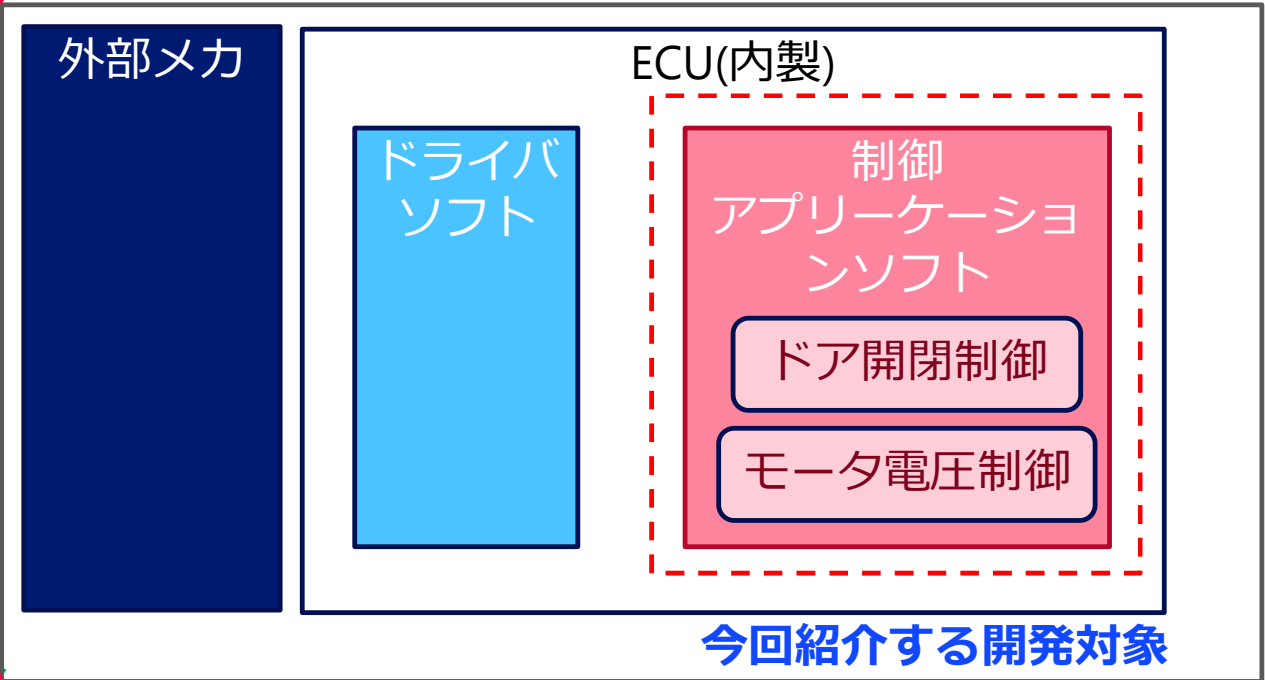
走行安全



アフターマーケット

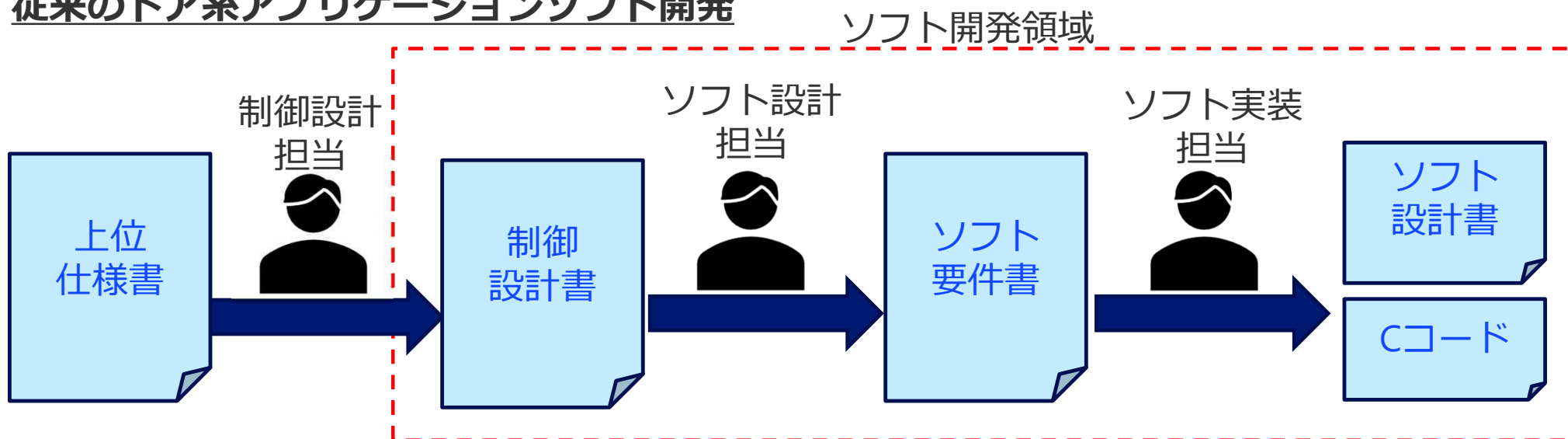


開発概要



導入の背景

従来のドア系アプリケーションソフト開発

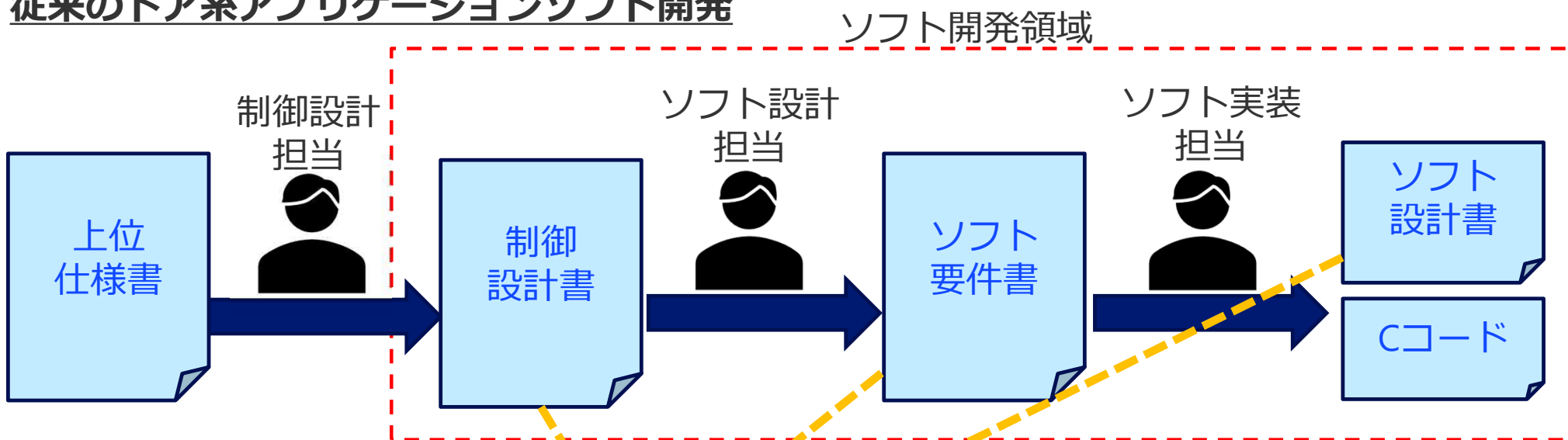


当時のソフト開発課題点

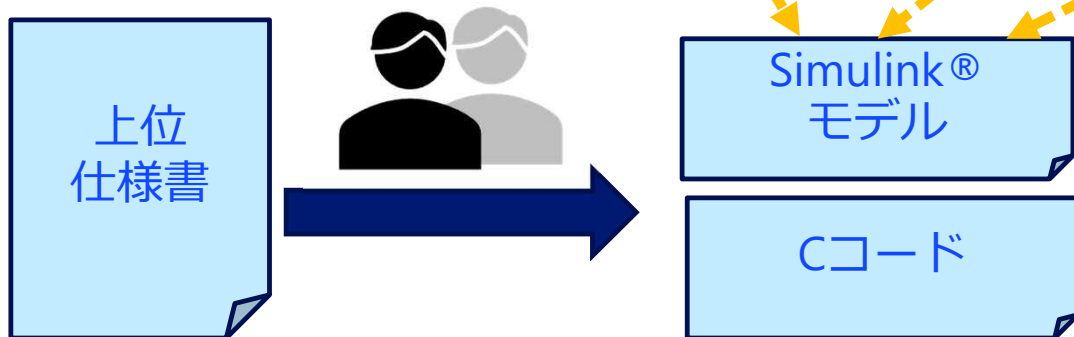
- 設計ドキュメントの多さに伴うメンテナンス性
各ドキュメントで担当跨ぎ→語句の解釈に関する問い合わせによる手間が大
- ドア機能の電動化領域の拡大
制御設計に人員をシフトさせていく必要あり

導入の背景

従来のドア系アプリケーションソフト開発



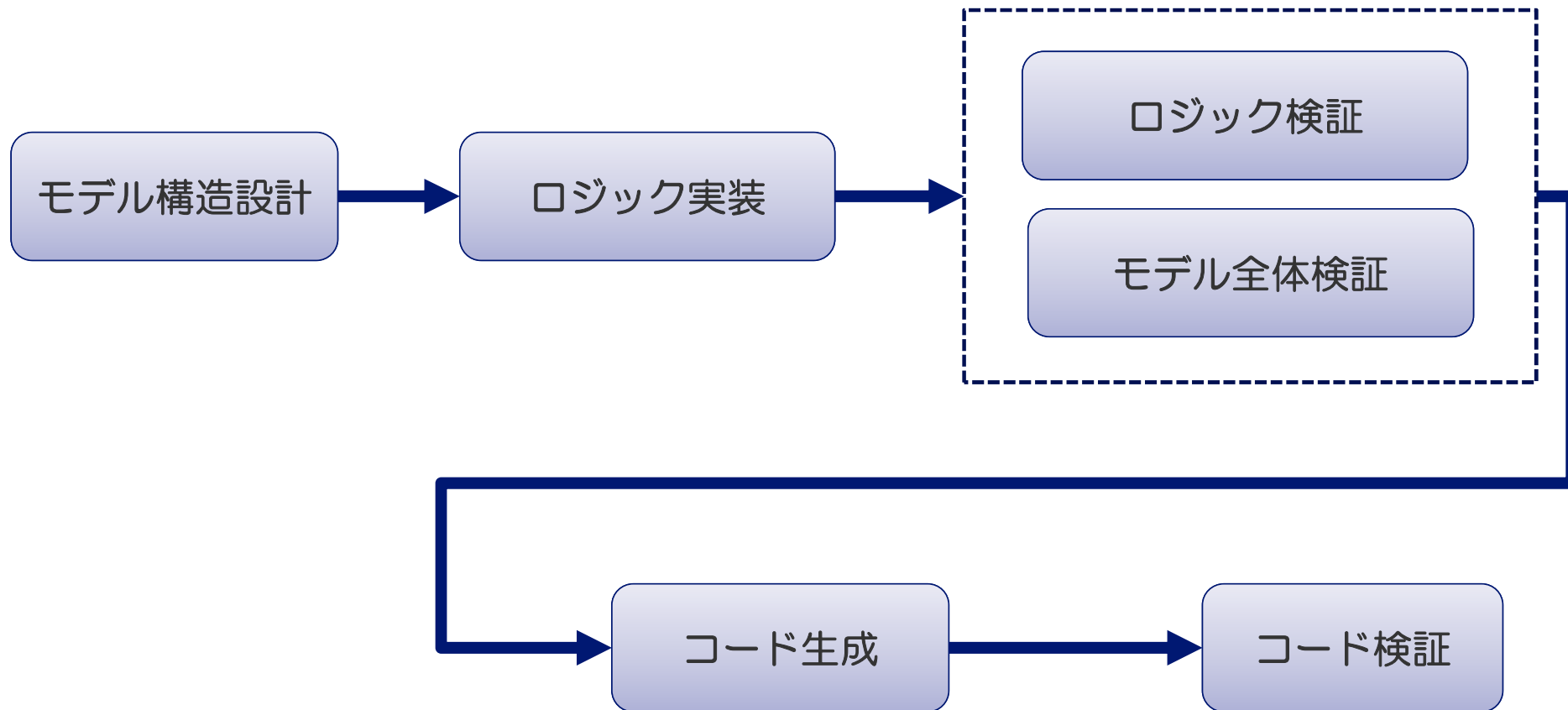
MBDを活用した開発



ソフト開発領域の各設計書をSimulink®モデルに置き換え。

⇒ **Simulinkモデルを共通言語とした制御～ソフトまでの一気通貫開発を実施**

モデルを活用したアプリソフト開発の全体像



モデル設計(モデル構造設計)

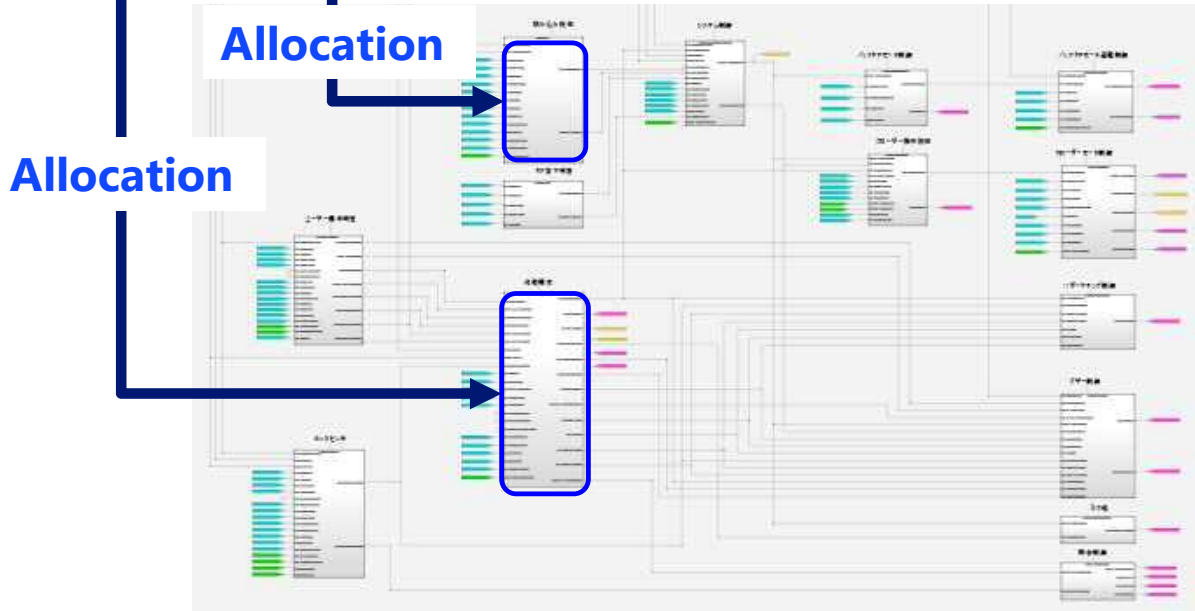
上位仕様書



論理アーキ(制御設計)
制御の機能配置
物理アーキ(ソフト設計)
IFの影響

- 評価/検証容易性
- 制御機能単位で整理、可読性
- 機能間の影響の最小化
- 移植性

制御/ソフト設計両面から
保守性の高い構造を構築

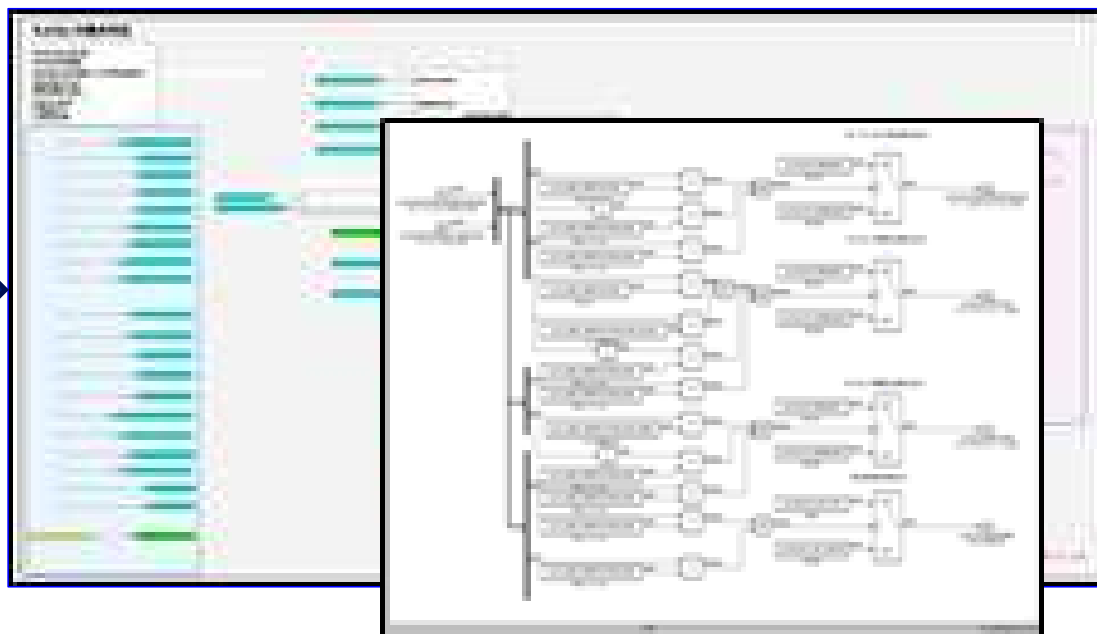


モデル設計(ロジック実装)

ユニット設計ガイドライン



ルール適用



以下を考慮したモデリング手法のルール化

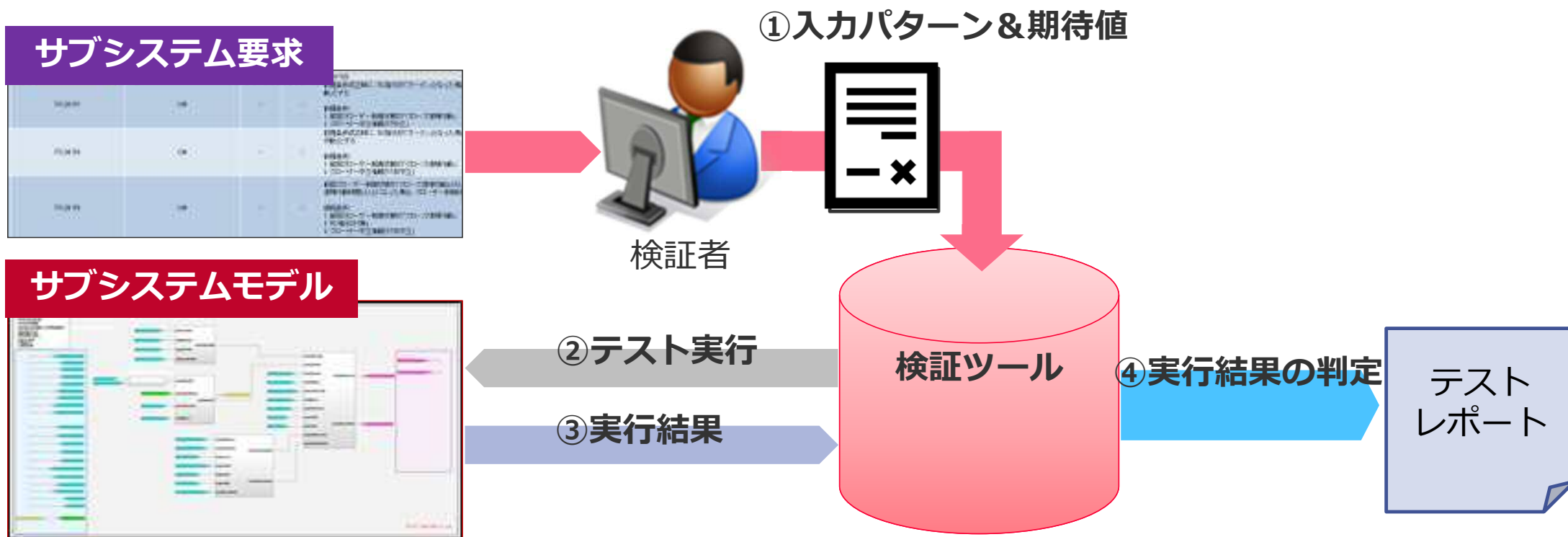
- 可読性/検証容易性
- 処理負荷/メモリ効率
- コーディングルール準拠

**機能モデルから実装用モデルへの
作り替えが不要なモデリングを実現。**

ロジック検証

モデル構造設計にて分解された機能単位での検証

⇒**Simulink® Test™**等検証ツールを活用し、**期待値の判別を自動化**

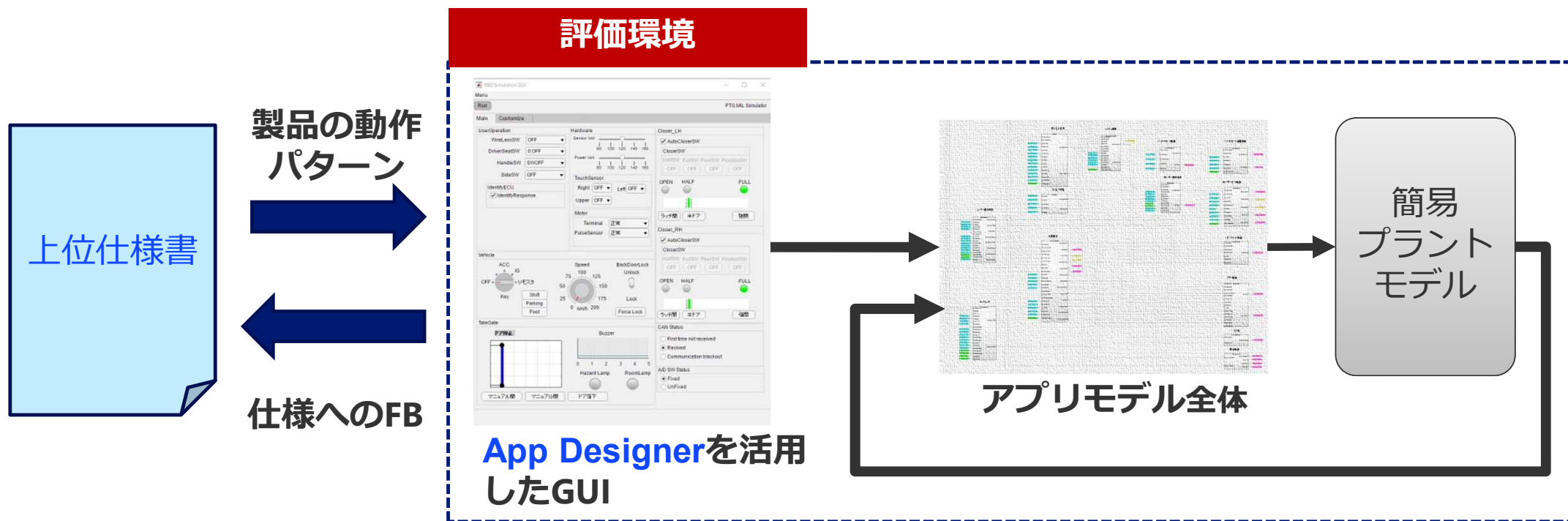


※演算処理等の検証はSimulinkシミュレーションで検証実施

モデル全体検証

上位仕様書に基づき、アプリモデル全体を検証
製品として正しい動きをしているかの確認。

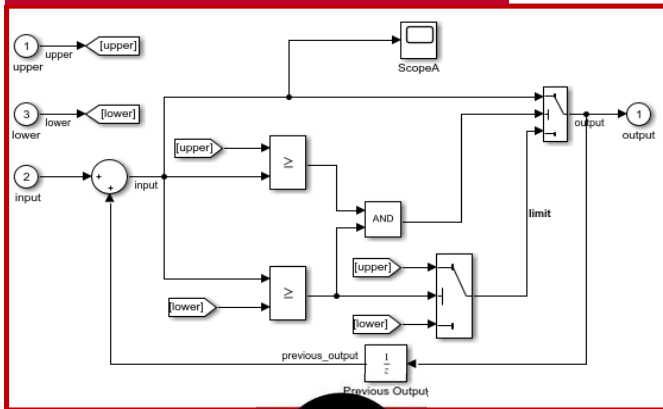
⇒ **App Designer** を活用し、**実製品ユーザーの複雑な操作を模擬**
実機完成前にモデルや仕様自体の問題点を早期検出可



モデル検証

作成したモデルが正しい書き方をしているかの確認
→ユニット設計ガイドラインへの準拠チェック

サブシステムモデル



チェック
結果

チェック結果

モデル アドバイザー - untitled

ファイル 編集 実行 設定 強調表示 ヘルプ

検索:

モデル アドバイザー

- 製品別
- タスク別
- AI_DOOR_RuleCheck
 - SimulinkブロックとDDとの差異を検出
 - LookUPブロックの丸め誤差設定を確認
 - ブロックのDescriptionを確認
 - 使用禁止ブロックの存在を確認
 - ブロックの制約事項を確認
 - Switchブロックの記述制約事項を確認
 - Relationブロックの記述制約事項を確認
 - Logicalブロックの記述制約事項を確認
 - LookUpTableブロックの記述制約事項を確認
 - IEブロックの記述制約事項を確認

Switchブロックの記述制約事項を確認

解析

Switchブロックがガイドライン上の制約を満たしていることを確認します

このチェックを実行

結果: 失敗

警告

Switchブロックの設定がガイドラインに違反しています。以下:

- untitled/Switch

推奨アクション

+

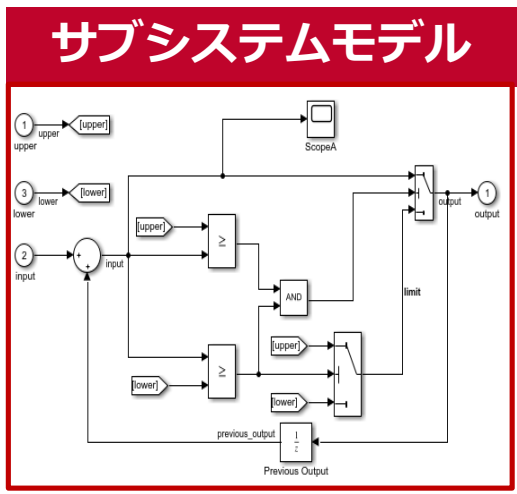
ユニット設計ガイドライン



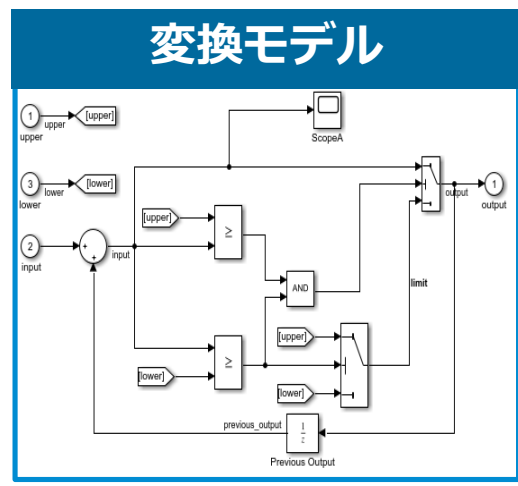
Simulink® Check™による
モデルアドバイザ機能のカスタマイズ設定を活用
→社内独自ルールも自動チェック可とし、
品質と効率向上。

コード生成(ACG)

モデル設計時点で**実装用モデルへの作り替えが不要なモデル**となっているため
Simulinkモデルに対し**コード化設定の埋め込み**を実施するだけで
量産ソースコードが作成可能

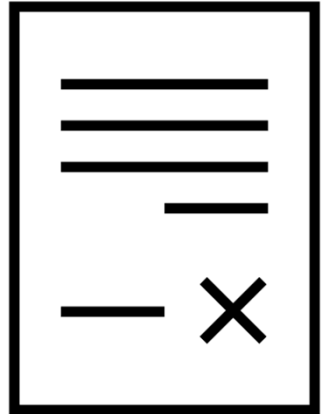


→
コード化設定
埋め込み



→
コード
自動生成

量産ソースコード

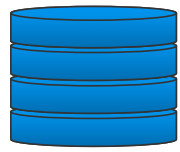


コード化設定埋め込みはツールによる半自動化
(※正しく埋め込めたかは人の目で確認)
→従来開発におけるソフト設計書から
Cコードを作る工程の時間を大幅に短縮

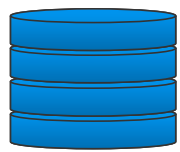
コード評価(単体・結合)

モデルと量産ソースコードに同一の入力を与え、出力結果の一致を確認。
⇒ロジック検証されたモデルと量産ソースコードが一致したことをもって
ソースコードの検証を行う(Back to Backテスト)

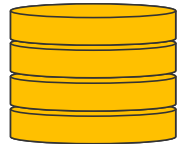
ロジック検証で
利用した
入力パターン



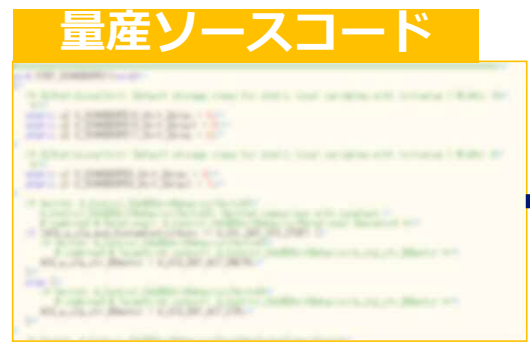
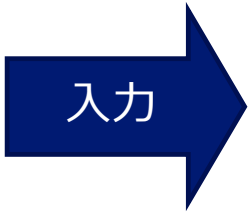
モデル全体検証
で利用した
入力パターン



追加パターン



Simulink® Design Verifier™等で自動生成



出力値を自動
比較・判定

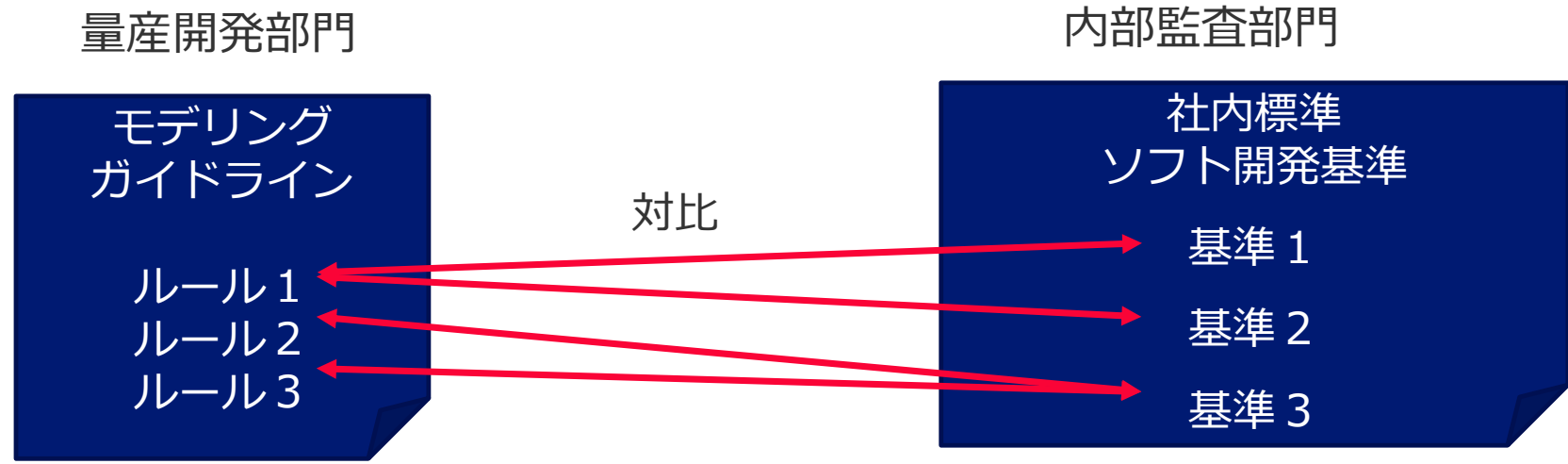
設計・評価工程外の取り組み

内部監査部門との連携

従来のCコード開発における品質保証規約と
モデル自体やモデリングガイドライン等との対比を行い、内部監査部門と認識合わせ

⇒設計アウトプットをモデルに置き換えたことによる
社内標準開発基準への準拠の確認方式を新たに構築

例：



開発の中の工夫

Simulinkモデルのみで開発を行わない

ViewerとしてExcel等の積極活用

→モデルに馴染みのない人とのレビュー・読み合わせを容易にした

例：外部ソフトとのIF仕様読み合わせ

モジュール間のI/F一覧表



設計情報
入力

設計情報
出力



カスタマイズしたブロック
をモデルに配置してクリッ
クするだけで自動生成

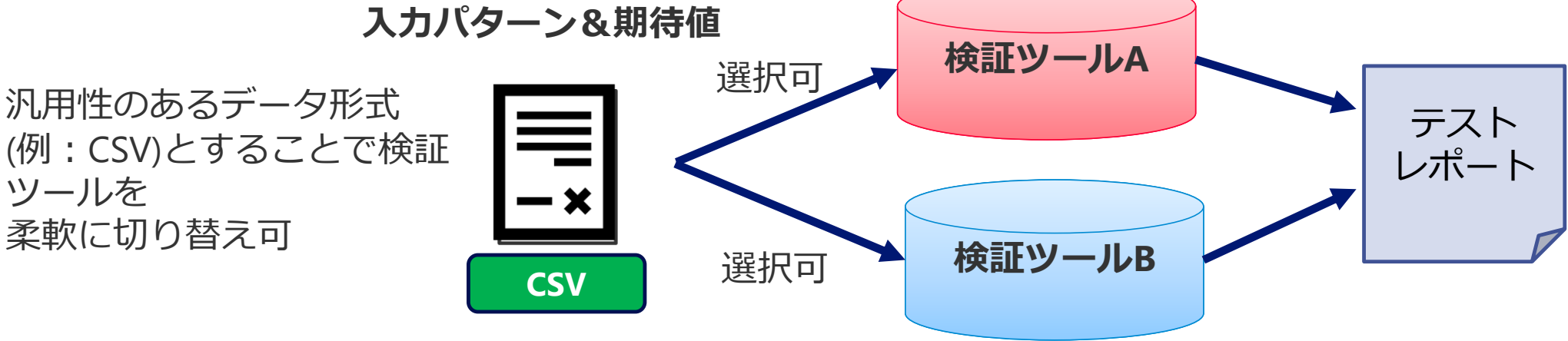
モデル設計内容⇔EXCELドキュメントの自動反映等の仕組みを構築
→説明のための資料を作る工数を削減。

開発の中の工夫

開発ツールに極力依存しない形式で設計・検証データを保存

⇒ 各開発プロジェクトの人員体制、利用可能なツールライセンス数に応じて柔軟に開発ツールを切り替え可能とした

例：モデル検証

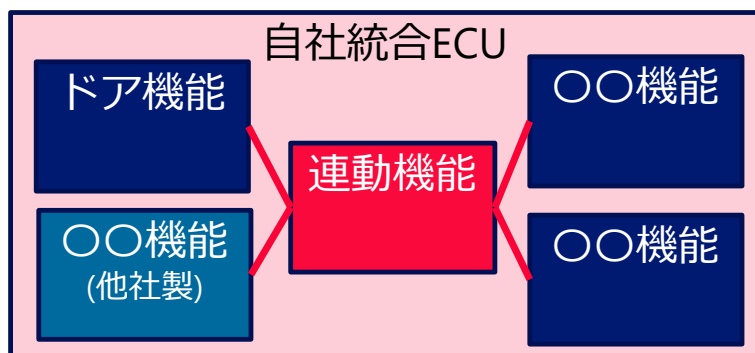


今後の期待

■自動車業界の傾向

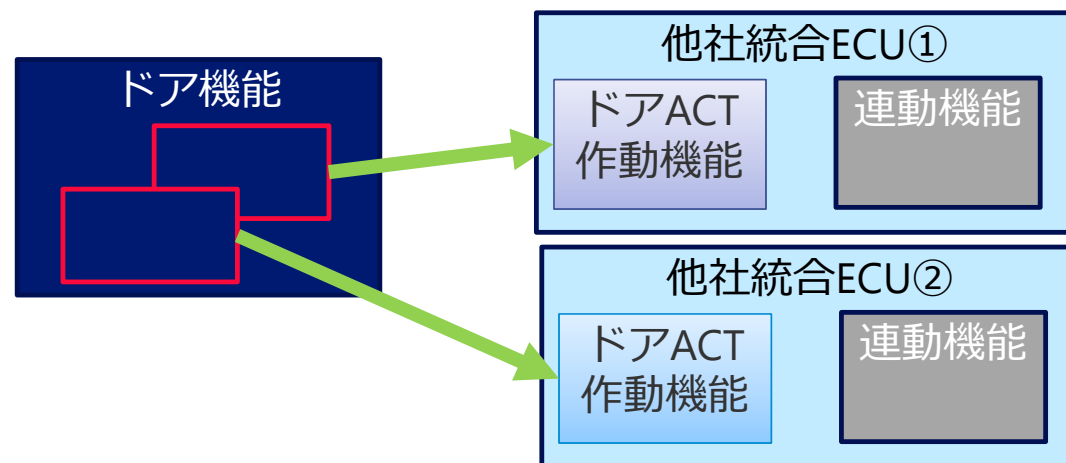
1つのECUへの複数機能の集約(統合ECU化)

Case 1 : 自社ECUで複数機能の統合



自社で製品をまたいだ統合制御ECUの設計
→連動機能の設計(新規領域)に
リソース集中が必要

Case2 : 他社統合ECUへの機能単独の提供



相手に合わせた自製品の一部機能の切り売り
→製品単位でなく、機能単位での保証・提供
が必要

今後の期待

■自動車業界の傾向

1つのECUへの複数機能の集約(統合ECU化)

Case 1 : 自社ECUで複数機能の統合

Case 2 : 他社統合ECUへの機能単独の提供

アプリケーション開発に求められること

現状のモデル構造設計より、より細分化された機能分解

→要件&評価項目の管理も細分化。移植性も必要とされる

既存機能を組み合わせた新規サービスの開発

→既存の機能の保証(従来の機能は正しく動くのか?)を両立させ

新規サービス部(例:連動制御)にリソース集中が必要

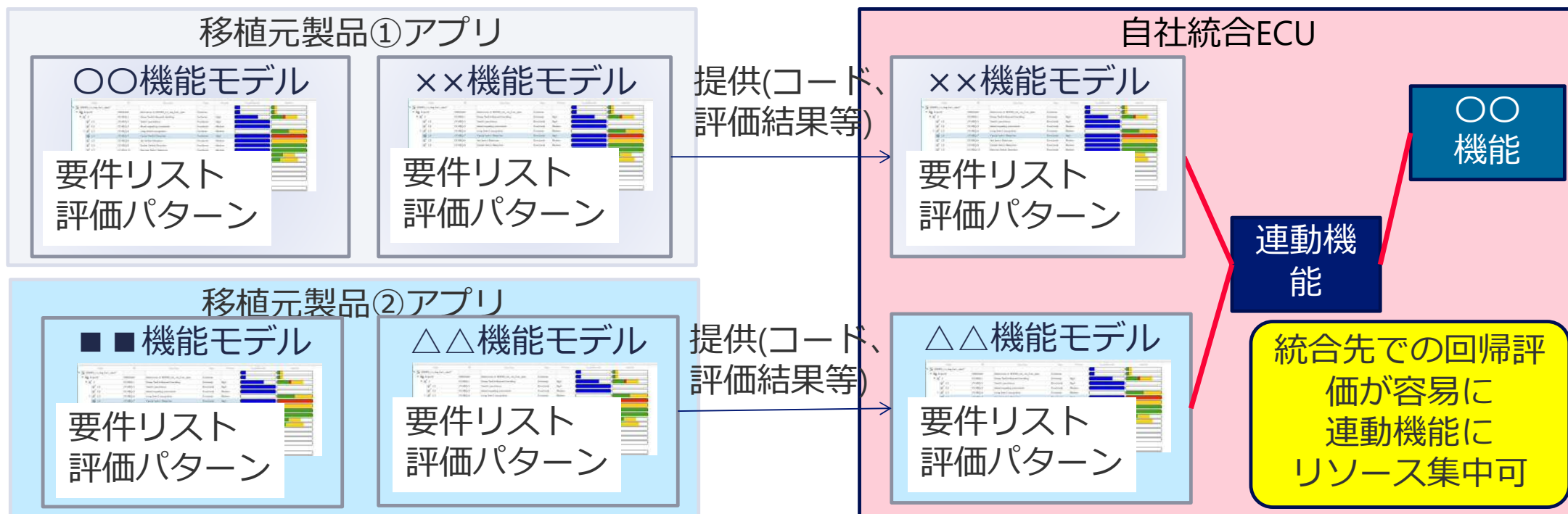
今後の期待

Requirements Toolbox及びSimulink Testを活用し

→より細分化された機能での流用性向上

システム統合において**既存機能の機能保証**にかかるリソースを最小限にする

Case1:自社ECUで複数機能の統合



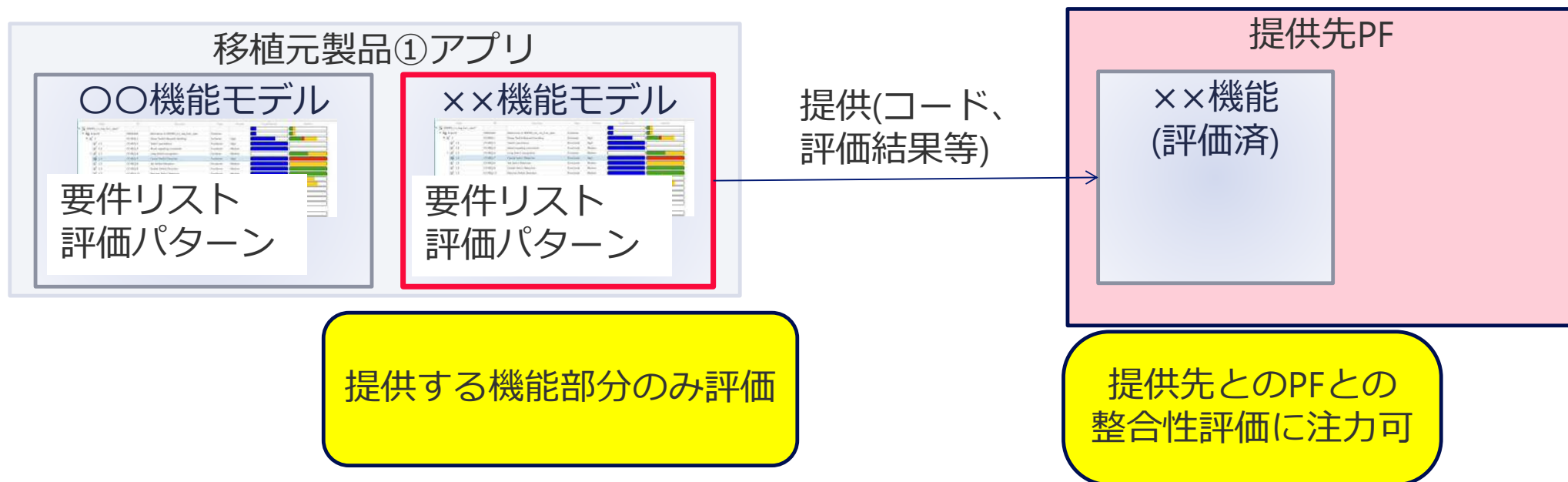
今後の期待

Requirements Toolbox及びSimulink Testを活用し

→より細分化された機能での流用性向上

システム統合において**既存機能の機能保証**にかかるリソースを最小限にする

Case2 : 他社統合ECUへの機能単独の提供



まとめ

1. 従来複数の設計書を経由していたソフトウェア開発を Simulinkモデルに一本化することで、制御設計～実装まで一気通貫の開発を実現。

実装工程にかけるリソースを上位仕様設計へシフトさせた

2. Simulinkの設計・評価工程においてMathWorks殿のツールを活用し

チェックや評価効率を向上させた

3. 今後のドア系アプリケーションの開発の展望における課題解決においても

MathWorks殿のツールが役立つ見込みがある

ご清聴ありがとうございました

