

MBD中部コンファレンス
PMA2:MATLAB開発

Simulinkモデル開発における 工夫事例

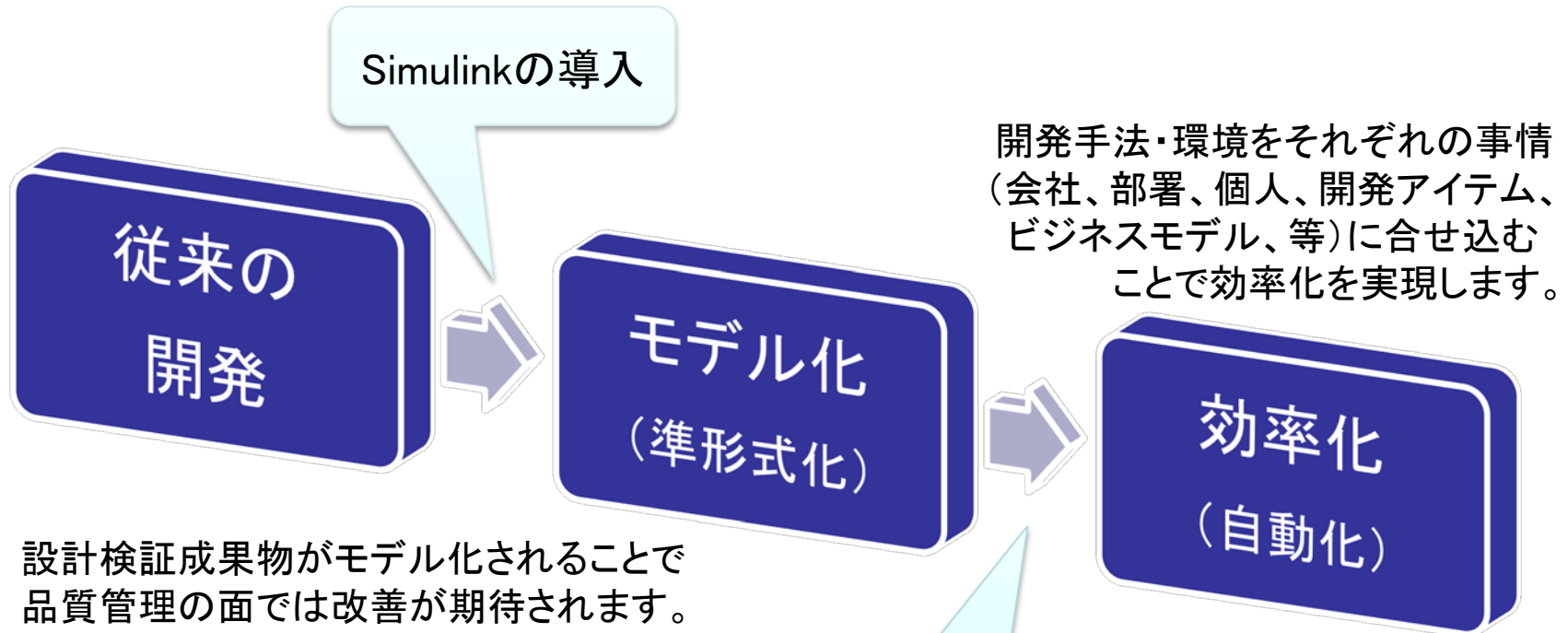
2014年12月18日

オムロンオートモーティブエレクトロニクス株式会社

開発統括室ボディコントロールシステム開発部

町井紀善

はじめに



設計検証成果物がモデル化されることで
品質管理の面では改善が期待されます。
→モデリング等の手間もあり作業は増加？

合せ込みの方法

- ・MATLABによる環境構築 →事例1、事例2、事例3
- ・Simulinkモデリングの工夫 →事例3
- ・市販ツールの活用 →事例4

事例1：回帰テスト環境 (1/4)

実現したい事

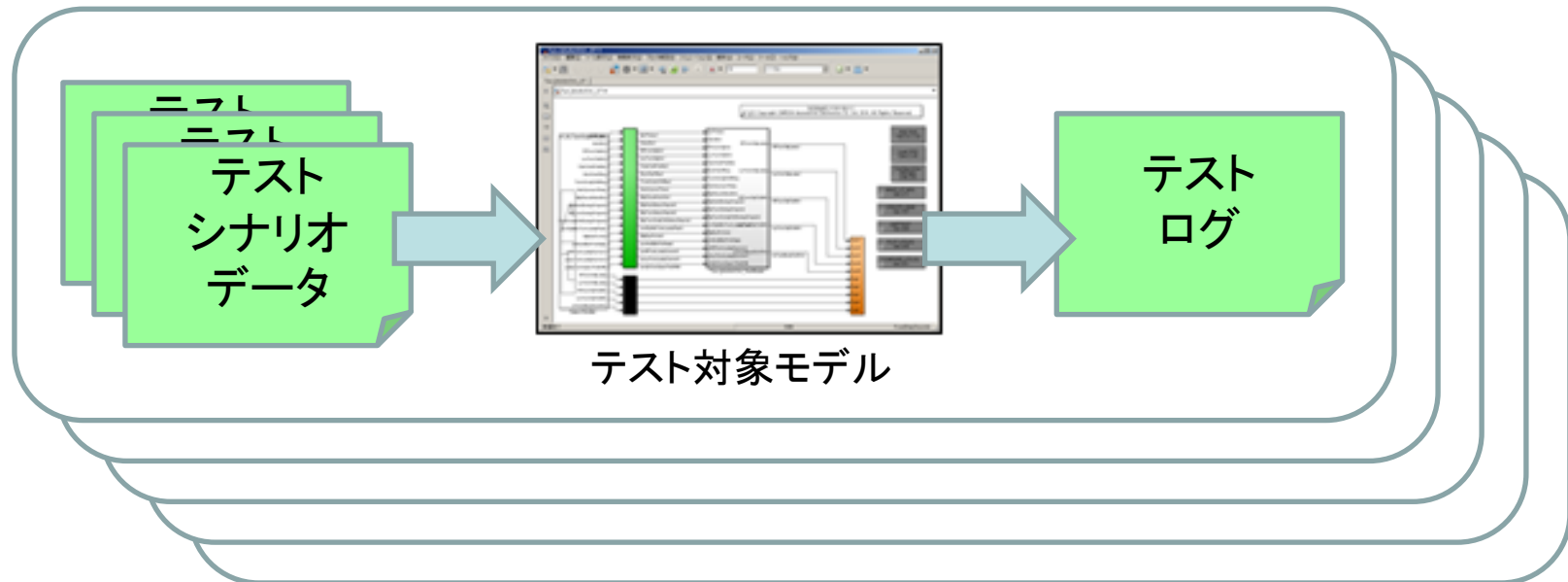
テスト工数削減のためにシミュレーションを自動実行させたい

- ・1つのモデルにテスト波形が多数
- ・テスト対象モデルが多数

- ・モデルカバレッジの記録
- ・各ログデータの波形生成
- ・各ログデータの過去との比較
- ・期待値と出力値の比較

→これら一般的な事柄も合わせて実施したい

- ・現在担当しているアイテム(多機能ECU)独自の事情
- ・数が多いため、自動化による効果大きい



事例1：回帰テスト環境（2/4）

過去の環境

過去に「1モデル多テストシナリオ」の自動シミュレーション環境を構築済み。
過去の環境を拡張して新仕様を実現できないか検討したが・・・

- ・環境構築にまとまった時間を確保できない（環境構築の専任者がいない）
- ・Simulinkの古い機能を使っていて、ツールで処理する機能範囲が広い

→今後を考え、新規に環境を構築

項目		過去の環境（R14SP3）	新環境（R2007b+）
目的		「1モデル多テストシナリオ」の自動シミュレーション	「多モデル多テストシナリオ」の自動シミュレーション
仕様	テストシナリオデータ書式	CSVファイル＋ 独自規格テキストファイル	
	テスト信号生成	From Workspaceブロック	
	ロギング方法	Scopeブロックの保存機能	
	:	:	



事例1：回帰テスト環境 (3/4)

改善ポイント

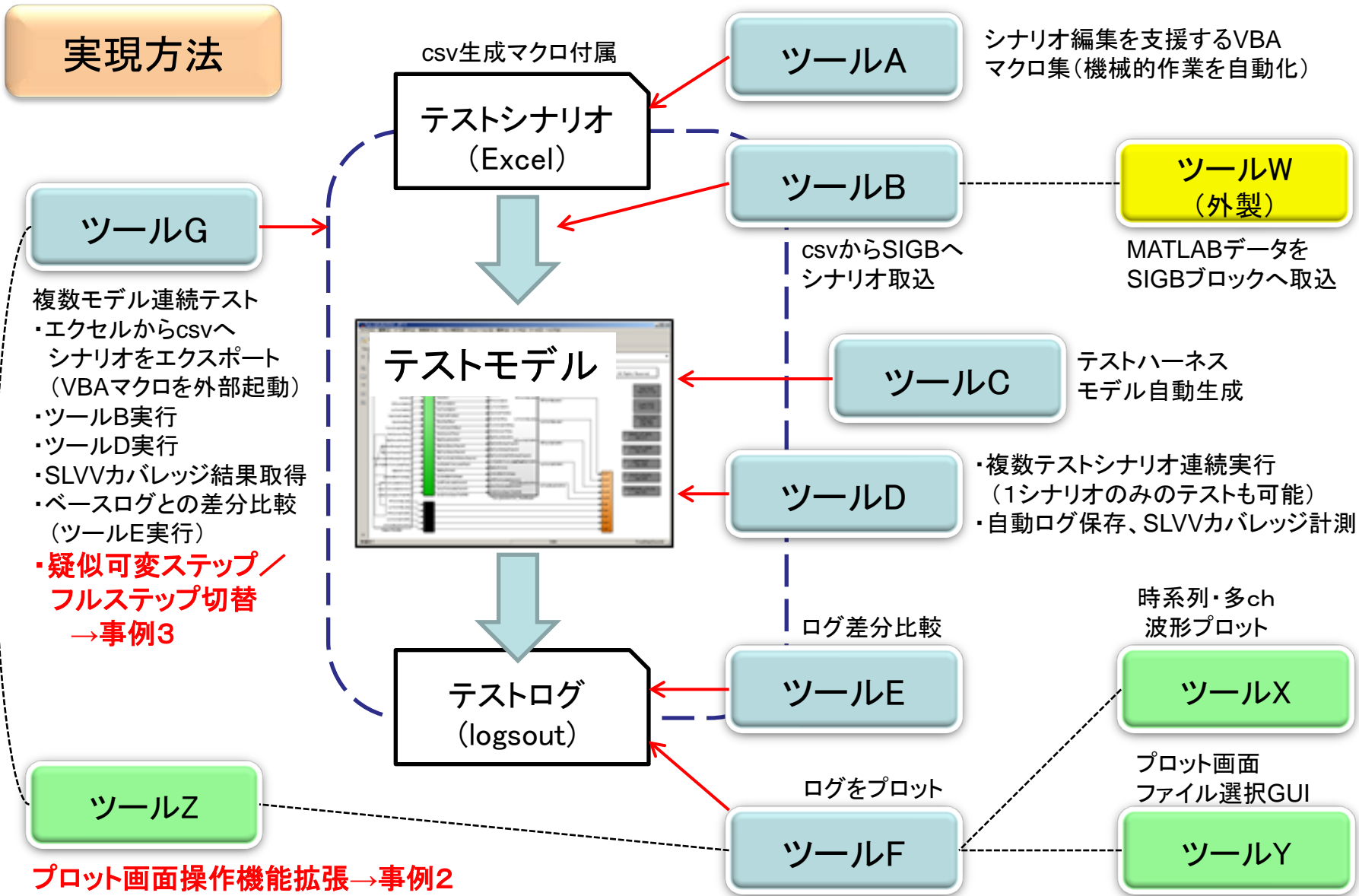


段階的な導入や機能拡張が可能。管理が楽。

項目		過去の環境 (R14SP3)	新環境 (R2007b+)
目的		「1モデル多テストシナリオ」の自動シミュレーション	「多モデル多テストシナリオ」の自動シミュレーション
ツール化プロセス (考え方)		実現したい事 (要求仕様) → ツール化 (ツール仕様)	実現したい事 → 手作業手順化 → ツール化
ツールの構成		全体で1つのツールとして管理 (Mファイルは分割)	手作業に対応した機能毎に独立したツールとして管理
仕様	テストシナリオデータ書式	CSVファイル + 独自規格テキストファイル	エクセルファイル
	テスト信号生成	From Workspaceブロック	Signal Builderブロック
	ロギング方法	Scopeブロックの保存機能	信号ロギング機能
	:	:	:

- ・ 独自ルールを削減
- ・ Simulinkの新機能を採用

事例1：回帰テスト環境 (4/4)



事例2:プロット画面の操作機能拡張 (1/3)

実現したい事

波形の拡大縮小、スクロール
操作を簡単に行いたい。

- ・ボタンをクリックする操作が手間
- ・デバッグ作業(不具合要因の検討)に意識を集中したい

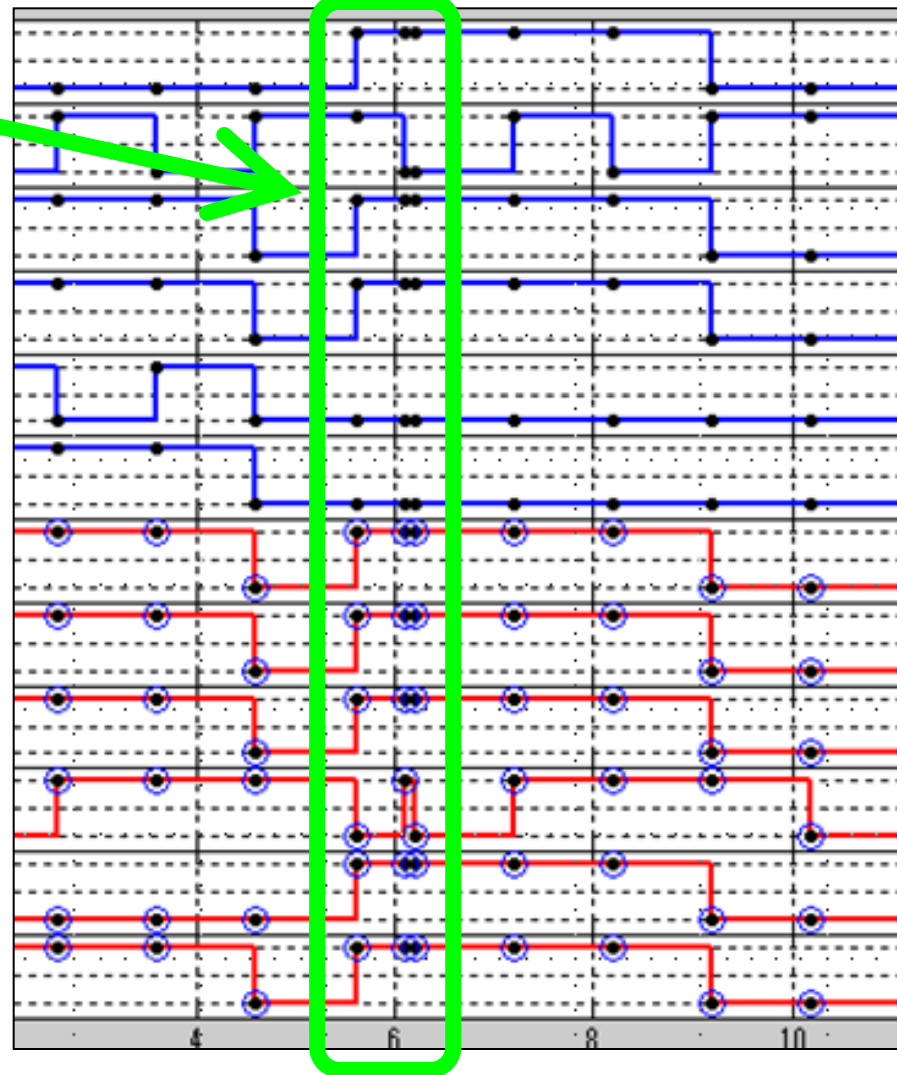
◆MATLABの機能

ズーム

移動



Simulinkシミュレーションのロギング波形



事例2:プロット画面の操作機能拡張 (2/3)

実現方法

Figure画面の操作
機能を拡張する
汎用ツールを作成

動作対象

動作方向

座標軸

XY軸

ウィンドウ

X軸のみ

Y軸のみ

同時押し操作

無し

+[Shift]

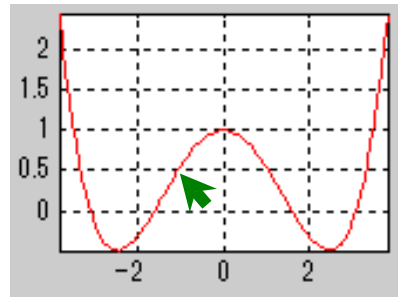
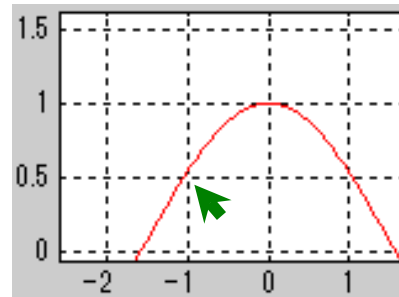
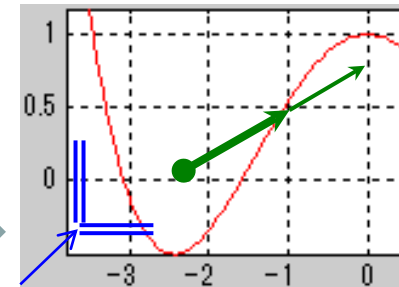
+[Alt]

+[Ctrl]

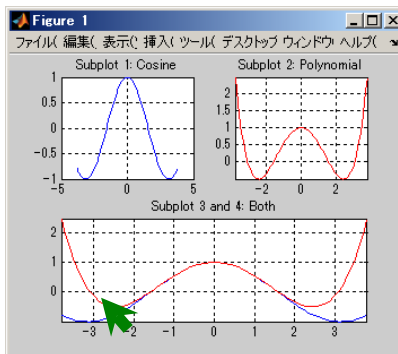
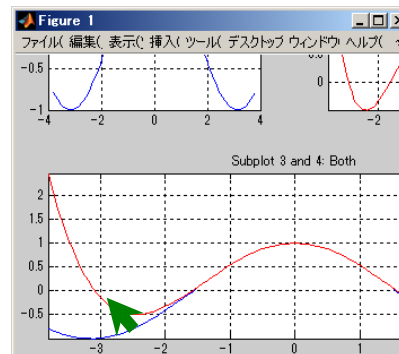
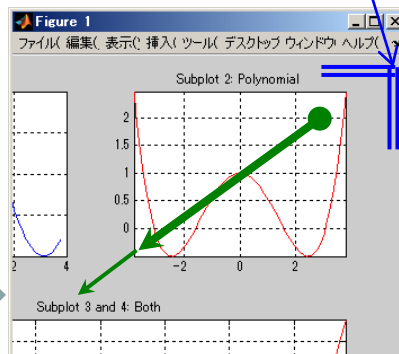
2 × 3 = 6種類の動作モード

任意の組合せ
で割り付け可

◆座標軸モード(従来の機能)

ホイール操作
でズームドラッグ操作
で移動オブジェクトの端点を認識
して表示範囲を制限

◆ウィンドウモード(スマホ風の機能)

ホイール操作
でズームドラッグ操作
で移動

事例2:プロット画面の操作機能拡張 (3/3)

開発手法

◆仕様書

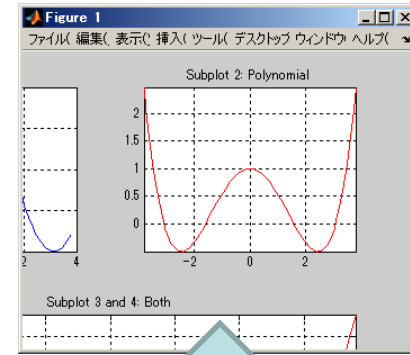
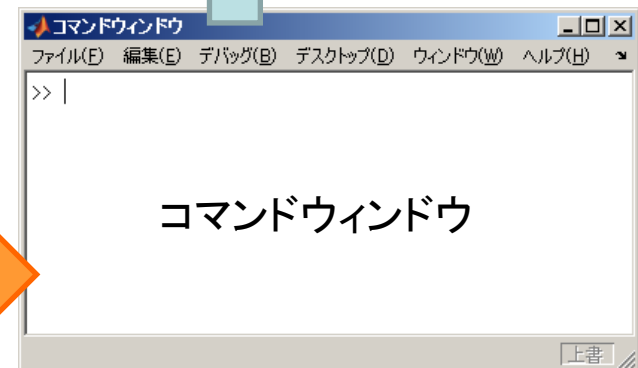
Mコードのヘルプテキスト領域を仕様書とすることで、仕様書の管理をシンプルにしています。

◆コマンドAPI

GUIツールであってもコマンドAPIを設けます。これにより、ツールの用途が広がります。

◆テスト

テスト用スクリプトでツールの各機能を実行し、プロファイラでカバレッジを確認します。(マウス等の操作が必要なケースは、DOSコマンドで操作できるフリーソフトを使用)

Figure
ウィンドウFigure
画面操作
機能拡張
ツールプロファイラ
カバレッジ
レポートテスト用
スクリプト

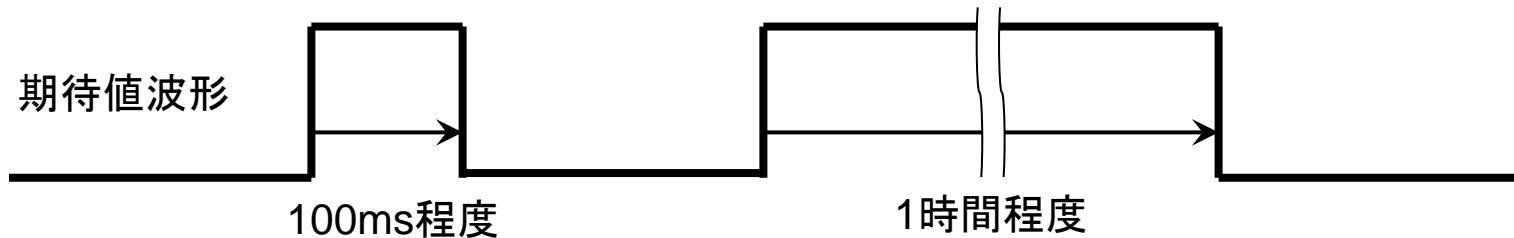
事例3：広レンジのタイマ制御モデル (1/2)

Sensing tomorrow™

※仕様の数値は架空のケースです。

背景

- ・100ms～1時間程度のタイマ制御を持つアプリケーション機能のモデル開発
- ・タイマ閾値の最小分解能は50ms
(シミュレーション時に連続系のプラントモデルは用いなくてよい)



改善課題

- ・普通にモデリング(カウンタで時間を計測)すると、ソルバの最大ステップを50ms以下にする必要があり、1時間タイマの機能確認に72000以上のシミュレーションステップが必要となります。
→シミュレーション実行時間とPCリソース負荷に対する影響が大きい。

事例3: 広レンジのタイマ制御モデル (2/2)

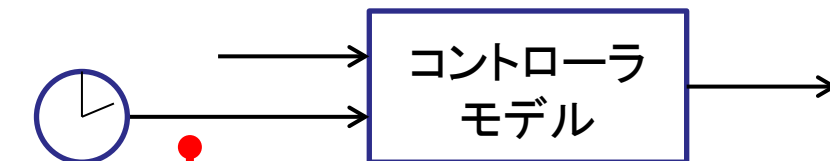
実現方法

- ・モデリングの工夫によりシミュレーション負荷を削減
- ・シミュレーションの仕方によって厳密なテスト(負荷大)も実施可
→事例1の回帰テスト環境で2つのテストモードを選択可能

コントローラモデルのモデリング

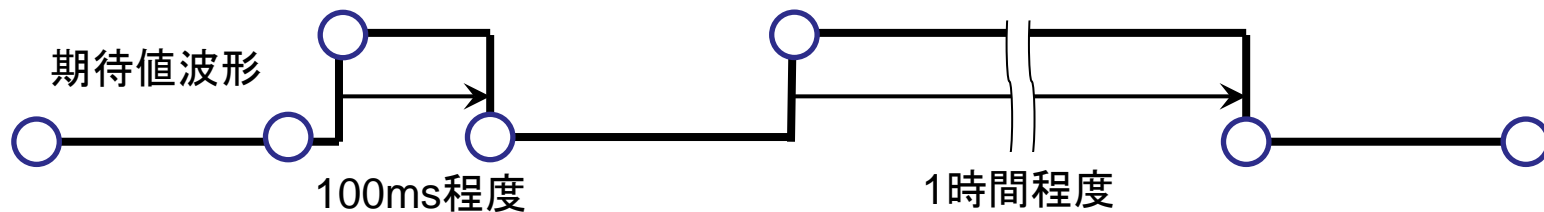
- ・時刻情報としてフリーランカウンタ信号を外部から入力
- ・過去のフリーランカウンタ値との比較で経過時間を判定

同一モデル&テストシナリオに対し、両方のテストを実行可能



カウンタ
信号

テストモード	シミュレーション方法	負荷	用途
フルステップ	ソルバを50ms周期固定とし、カウンタ信号を毎周期+50ms	高	実機に近い動作としてテストするとき
疑似可変ステップ	ソルバは1s周期固定(ダミー設定)とし、カウンタ信号を任意に定義	低	デバッグなど作業時間重視のとき



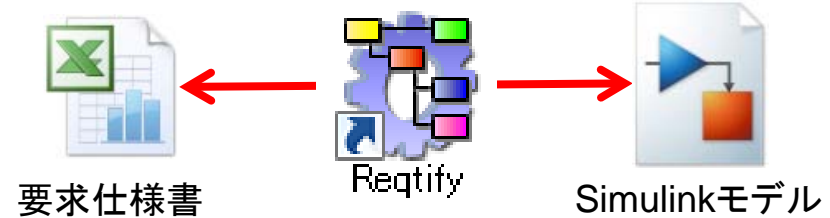
事例4: 要件管理ツールReqtifyとの連携 (1/2)

Sensing tomorrow™

実現したい事

※Reqtifyはダッソー・システムズ製の要件管理ツールでSimulinkモデルファイルに対応しています。

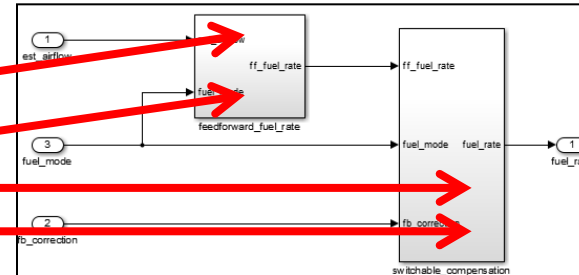
- 要件管理ツールReqtifyを使って要求仕様をトレースする
(諸事情によりツールが先に確定)



- 要求トレースの管理単位は「データフローレイヤ(処理)を有するサブシステム」

ID	タイトル	説明
R01	○○○	○○○○○○○○○
R02	△△△	△△△△△△△
R03	×××	×××××××

要求仕様



Simulinkモデル

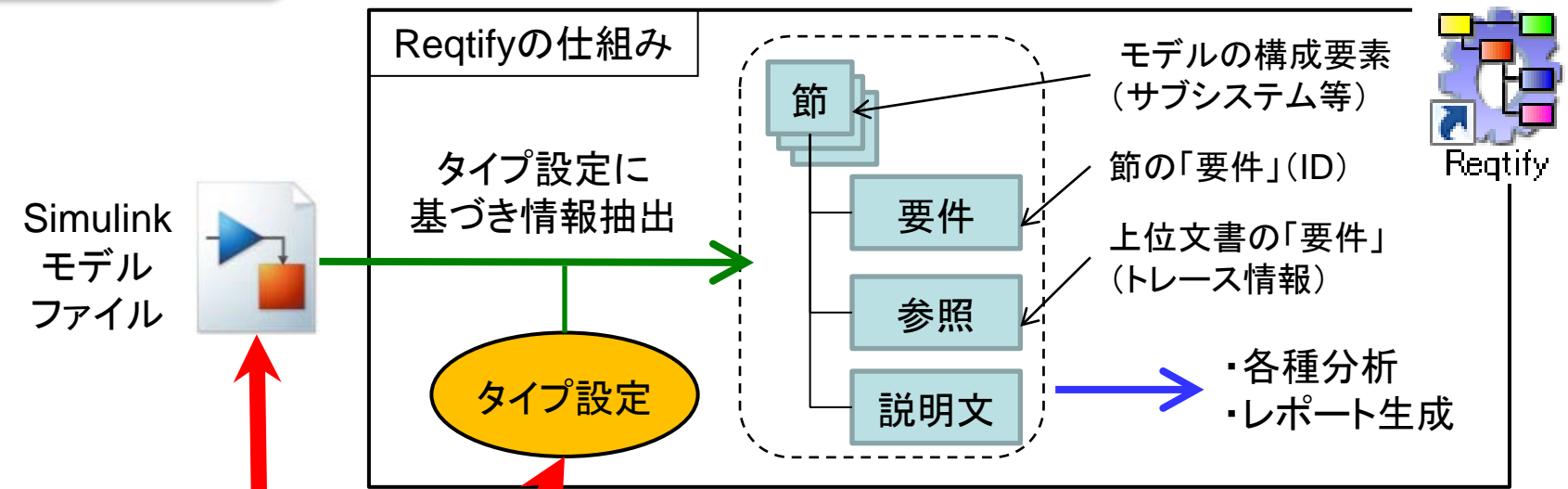
- 要求IDはモデルレイヤ上に可視化された状態にする

Reqtifyのデフォルト設定では、説明プロパティやDocブロックなど、レイヤ上に可視化されない箇所に要求IDを記述するようになっており、そのままではニーズを満たせない。

事例4：要件管理ツールReqtifyとの連携 (2/2)

実現方法

市販ツールを利用することで、ツールを作成して環境を構築するよりも、簡単にやりたい事を実現



データフローレイヤに配置

Model Infoブロックのテキストから情報を拾うように設定

モデルファイルの内部構造を知っていると設定が簡単 → 環境構築の知識が有効

機能名：○○機能
 要求ID：R01,R02
 説明：
 ・○○○○○○○○○○
 ・△△△△△△△△

Model Infoブロック

要件

参照

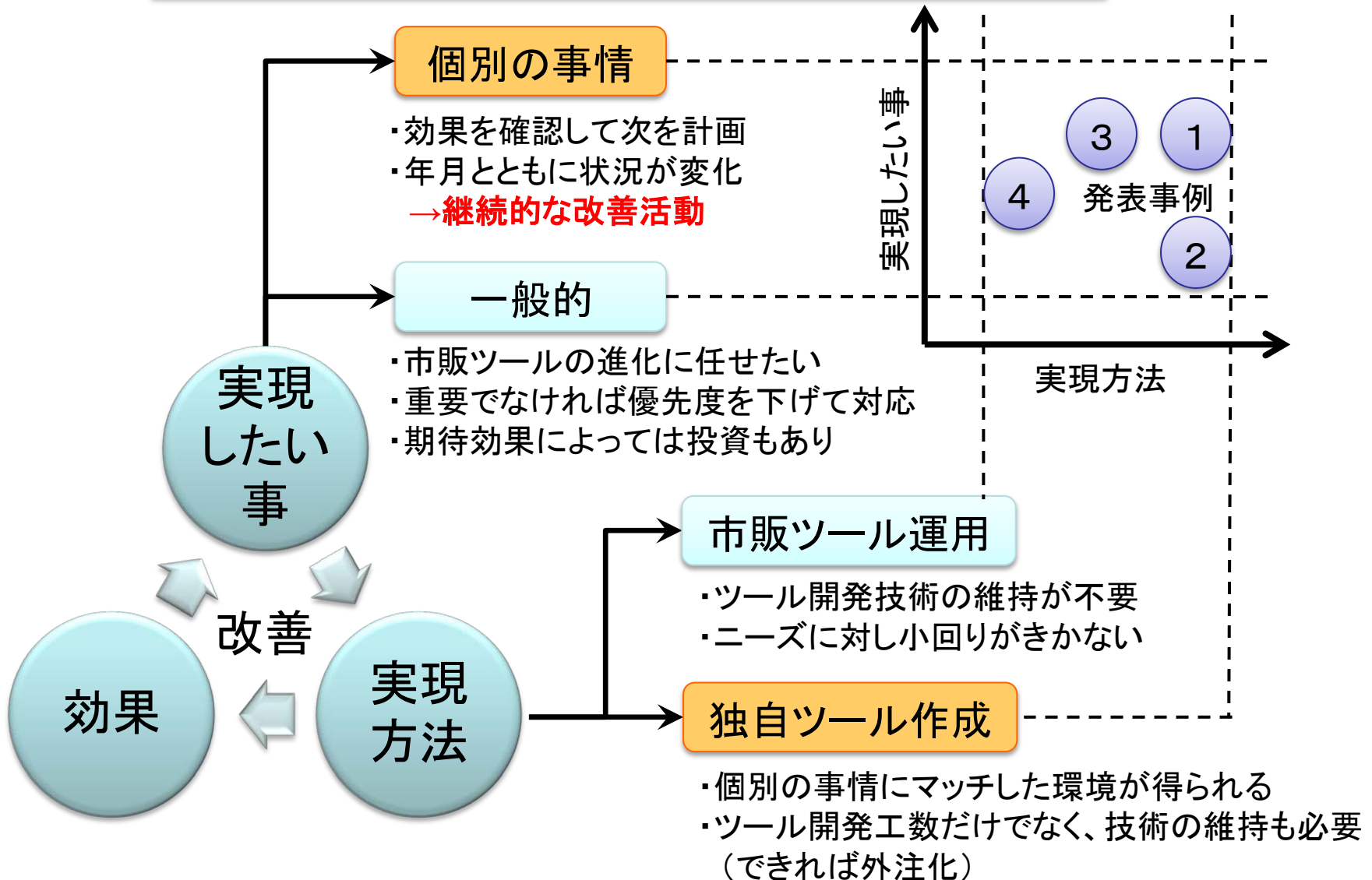
説明文

要求仕様書へリンク (トレーサビリティ管理)

Excel一覧表を生成 (モデル機能一覧として利用)

まとめ

改善には様々なパターンがあります。それぞれの特徴を考慮して改善を継続する事がポイントと考えています。



ご静聴ありがとうございました