MathWorksの要件ベースのソフトウェア開発ソリューション

MathWorks Japan
アプリケーションエンジニアリング部（制御）
シニアアプリケーションエンジニア
大越 亮二

はじめに

本Webセミナーの対象者

- **MBDによる製品開発に従事されている方**
  (対象エリア: 主に制御ソフトウェア開発領域)
- **機能安全等各規格要求に対応すべく要件と各種成果物のトレーサビリティに関する MathWorksツールの使いどころを知りたい方**

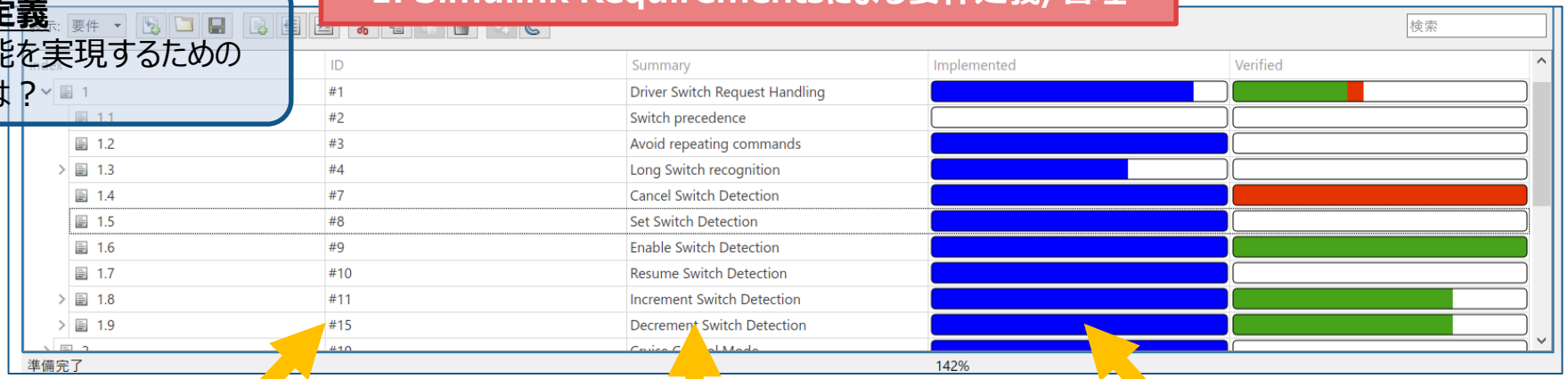
本Webセミナーでお伝えしたいこと

MBDによる要件ベースのソフトウェア開発を進める上で
MBD各開発工程において有用と思われるMathWorksツールの機能・使い方

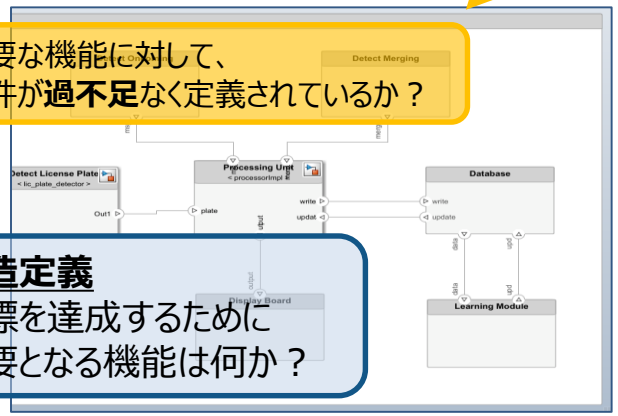
要件ベースによるソフトウェア開発のポイント

要件定義
各機能を実現するための
要件は？

1. Simulink Requirementsによる要件定義/管理



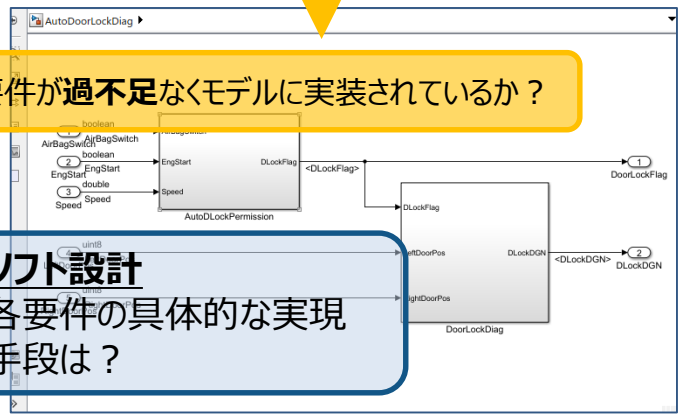
必要な機能に対して、
要件が**過不足**なく定義されているか？



構造定義
目標を達成するために
必要となる機能は何か？

2. System Composerによる機能定義

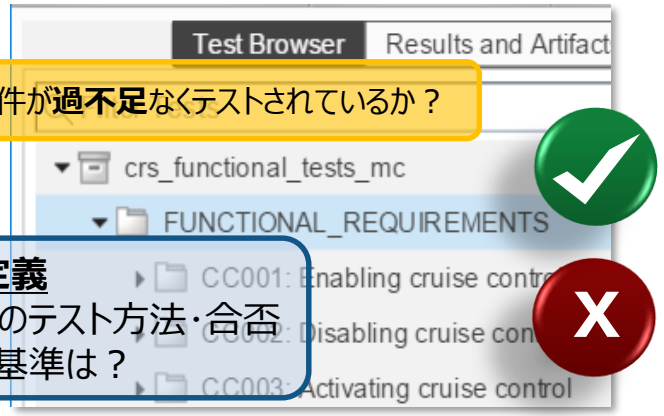
要件が**過不足**なくモデルに実装されているか？



ソフト設計
各要件の具体的な実現
手段は？

3. Simulink/Stateflowによる
モデリング・シミュレーション

各要件が**過不足**なくテストされているか？



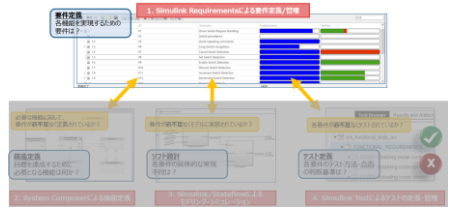
テスト定義
各要件のテスト方法・合否
の判断基準は？

4. Simulink Testによるテストの定義・管理

アジェンダ

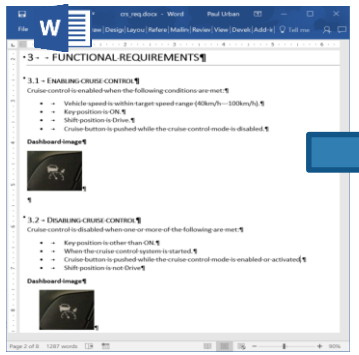
1. **要件定義/管理（Simulink Requirements）**
2. 構造・機能定義（System Composer）
3. モデリング・シミュレーション（ Simulink/SLDV ）
4. テストの定義・管理（ Simulink Test ）

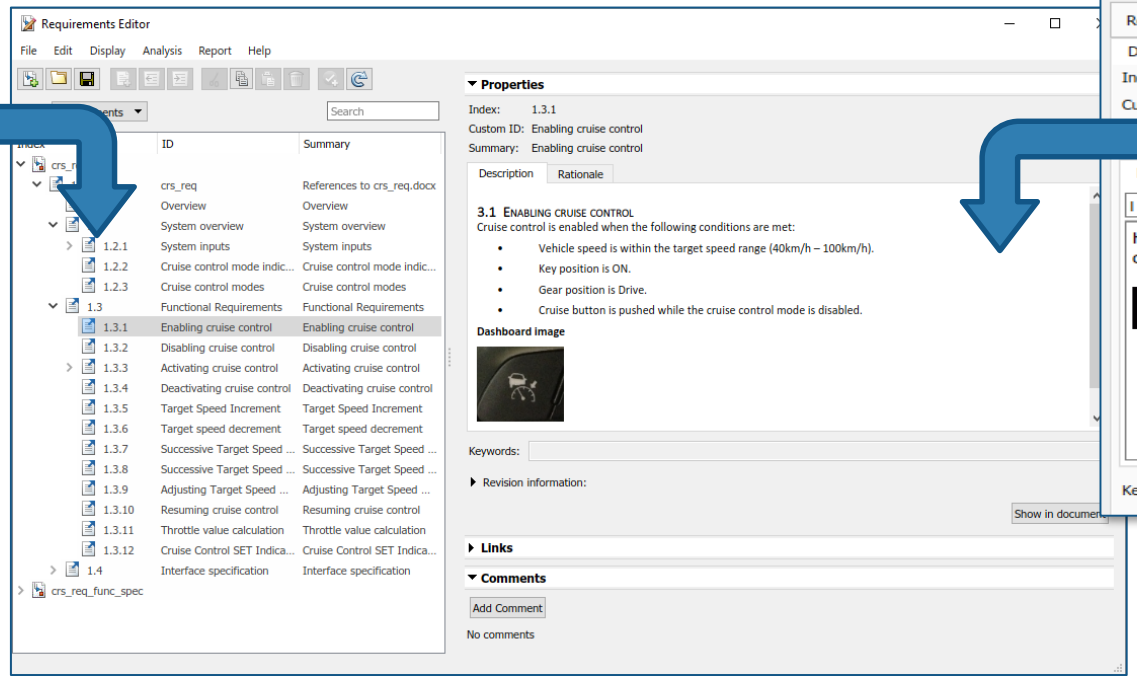
要件エディタによる要件の作成・外部仕様書の取り込み



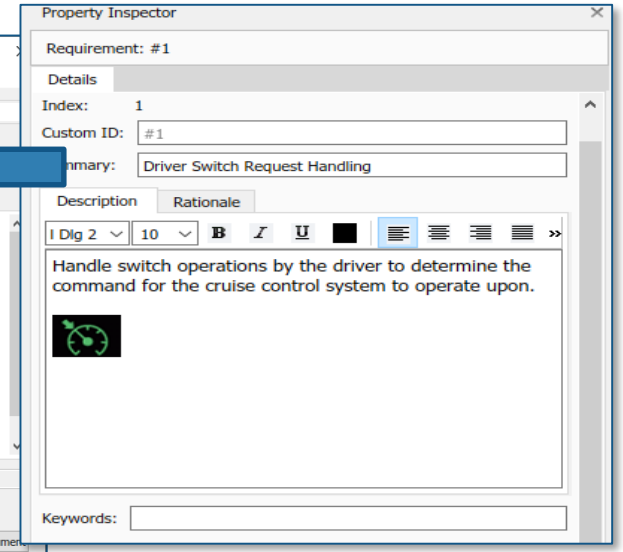
- **Simulink上で要件を作成(定義)できる**
- **Word/Excelで作成された外部仕様書は、要件エディタに取り込むことも可能**
  - 要件エディタ上の各要件を「読み取り専用」として取り扱うことで、外部仕様書をリファレンスにすることもできる
  - 外部要件を更新した場合、影響を受けそうなモデルやテストを検出可能(要トレーサビリティ設定)
- **ReqIF経由で外部要件管理ツールとの連携も可能**

外部仕様書の取り込み



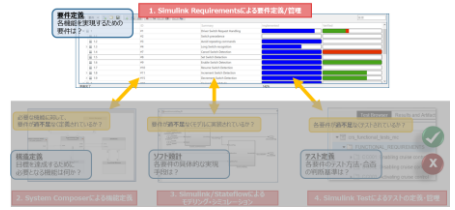


新規要件の作成

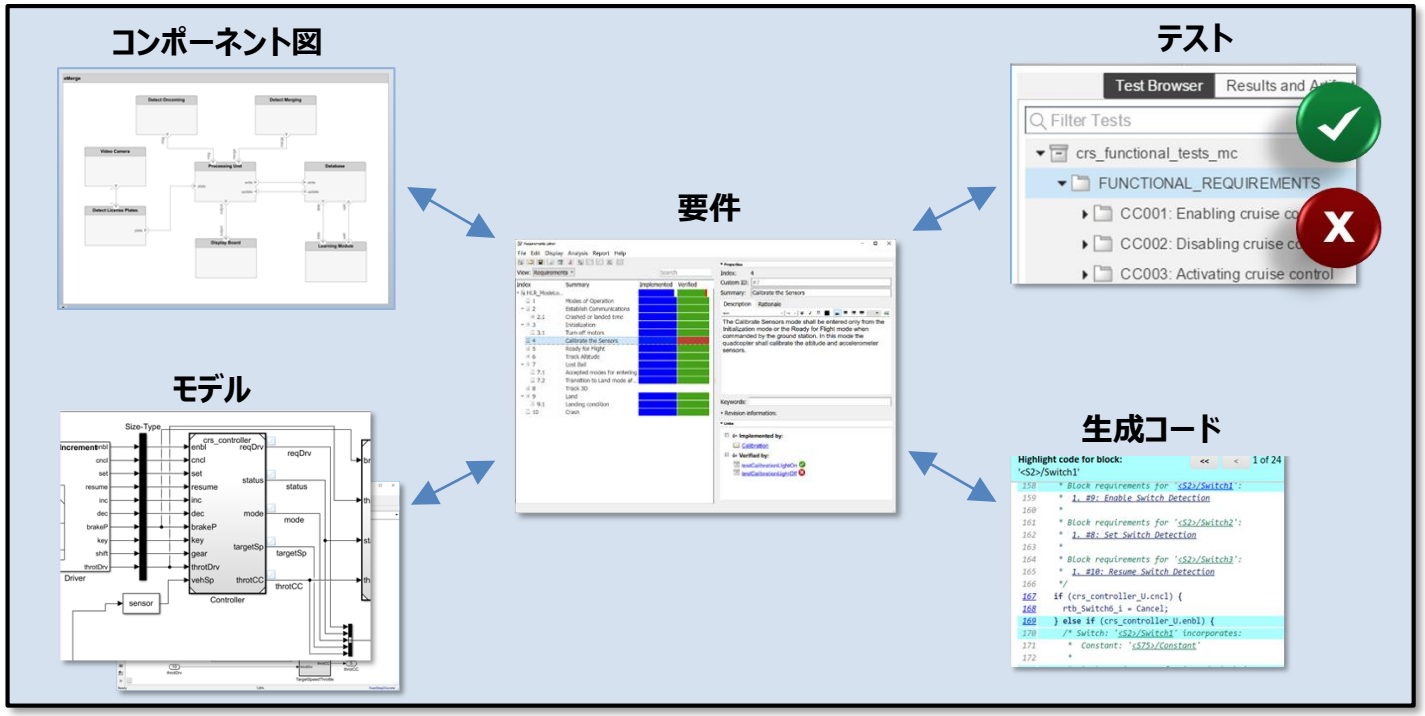


外部要件管理ツールとの連携

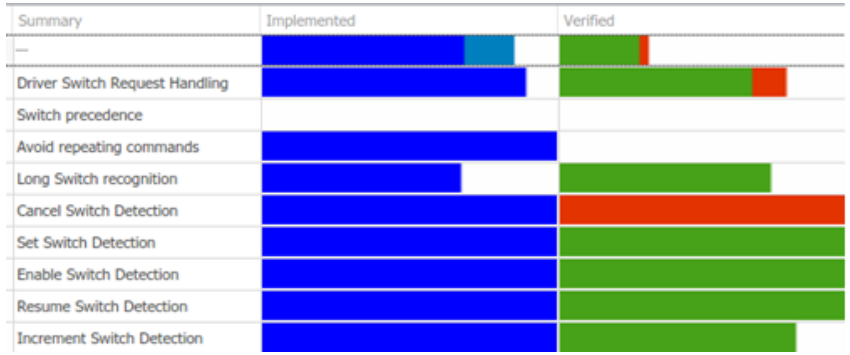
要件トレーサビリティ および 実装・検証状態の可視化



- 以下に対して、双方向のトレーサビリティを担保可能できる
  - 要件 ~ System Composerで作成したアーキ図内のコンポーネントダイアグラム
  - 要件 ~ モデル(Simulink/Stateflow)
  - 要件 ~ Test Manager内のテスト
- トレーサビリティ機能により、実装・検証状態を可視化できる



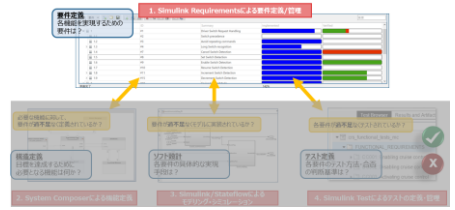
実装ステータス　検証ステータス



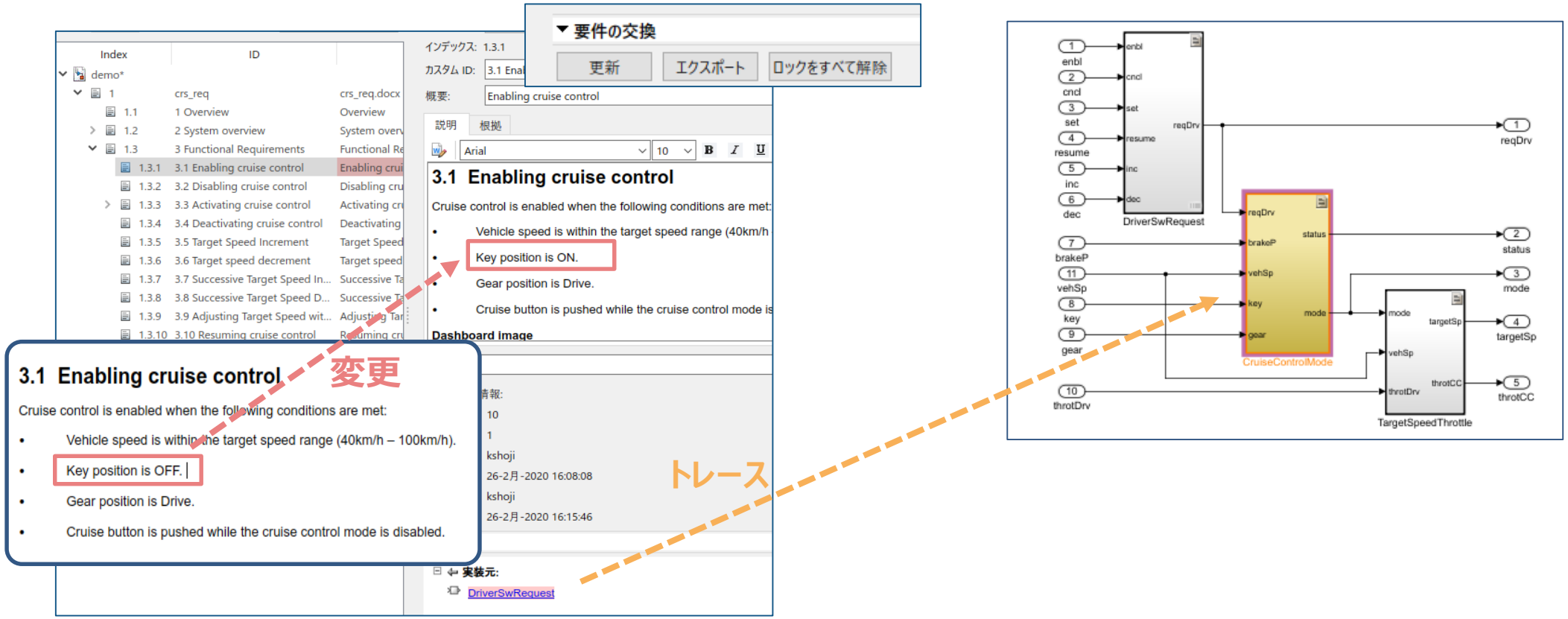
実装ステータス
- 実装済み
- 正当化
- リンクなし

検証ステータス
- 合格
- 失敗
- テスト未実施
- 正当化
- テストなし

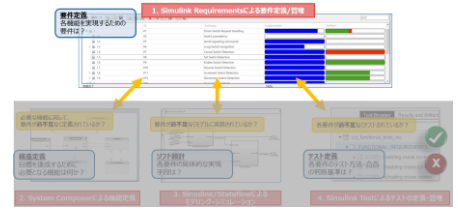
要件変更時の影響分析



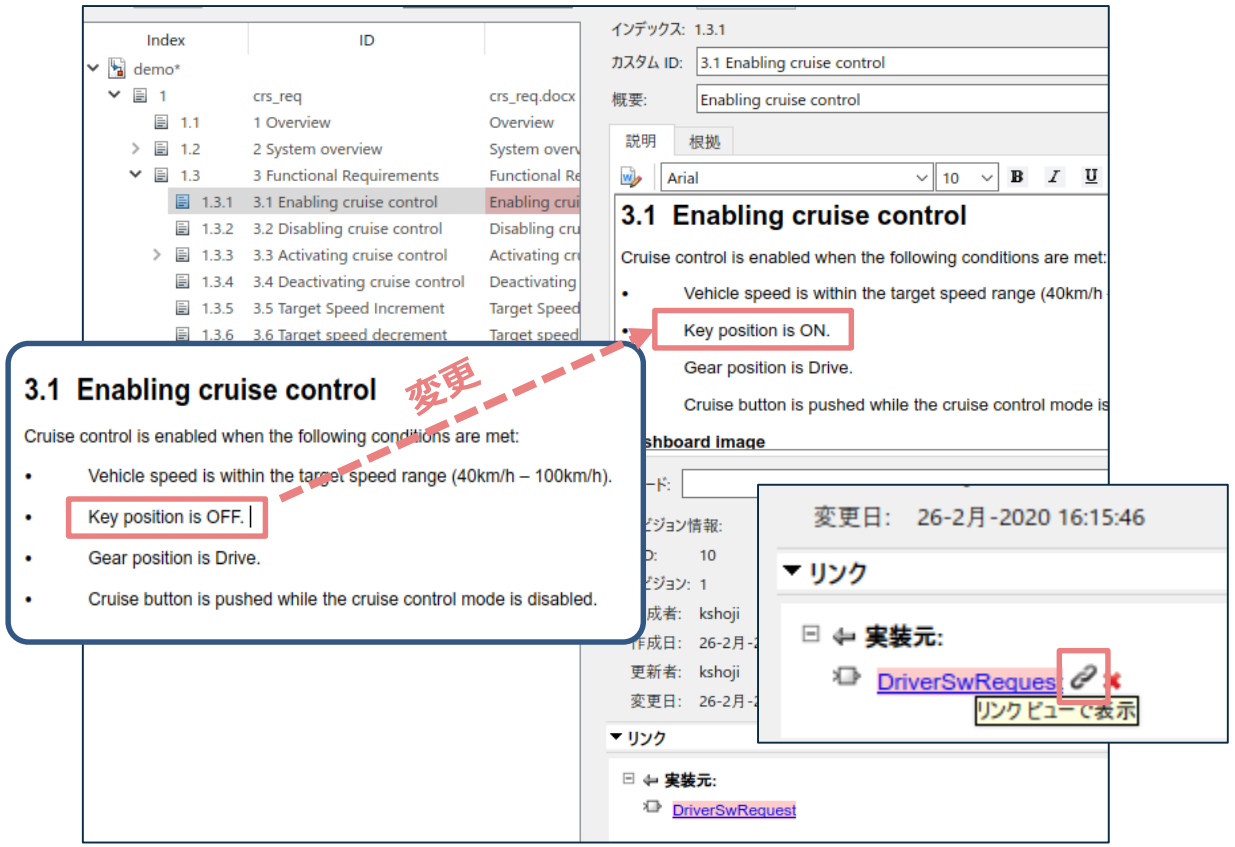
- **要件を変更した場合、影響範囲を特定可能**
  - 外部仕様書を「読み取り専用の参照」としてインポートしている場合は、要件エディタ上で「更新」を実行することによって変更点を検出可能
  - 要件に紐づけられているオブジェクト(アーキテクチャモデル、モデル、テスト)へとトレース可能

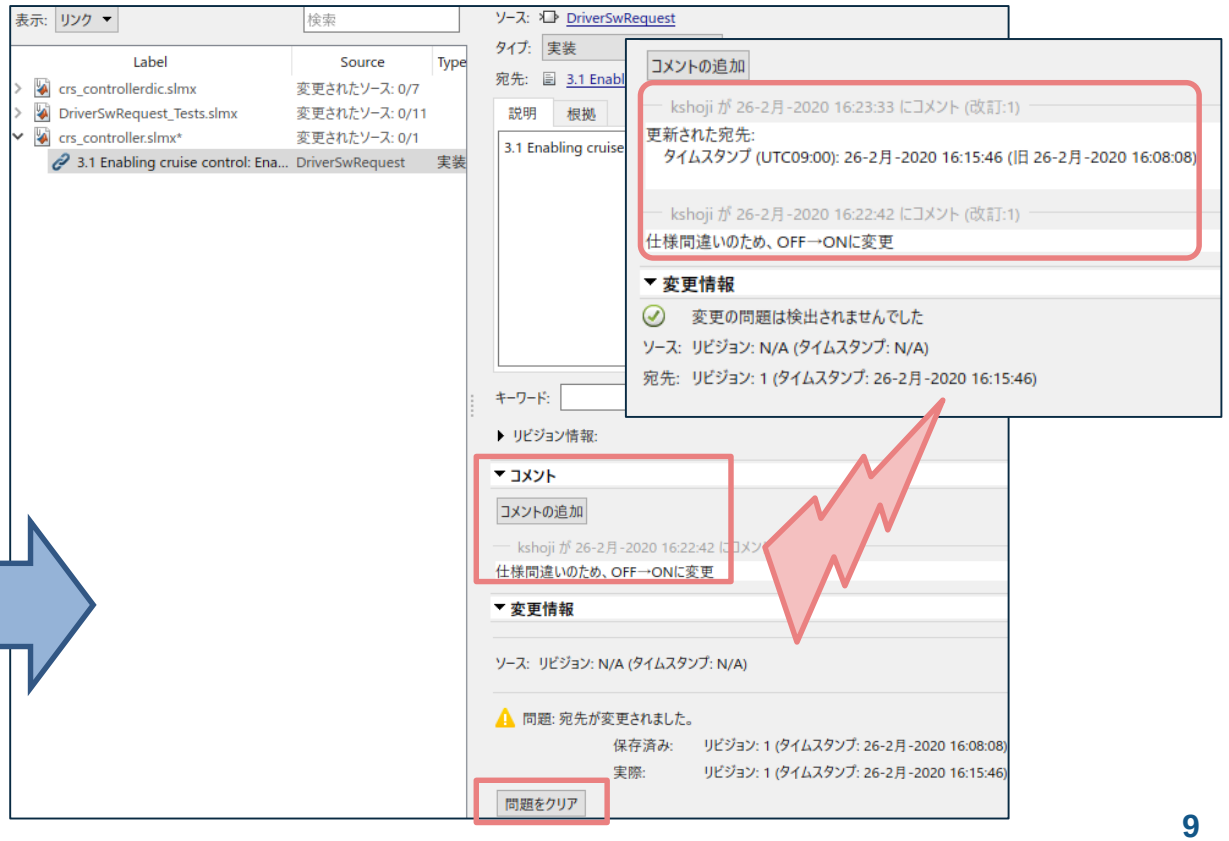


要件変更の管理



- **変更検出時、リンクビューに切り替えることで、変更点を管理できる**
  - コメントを追加して、「問題点をクリア」を実行することで、要件の履歴管理が可能
  - 誰がどの日時に変更したのかを履歴管理可能(変更内容は手動で入力)

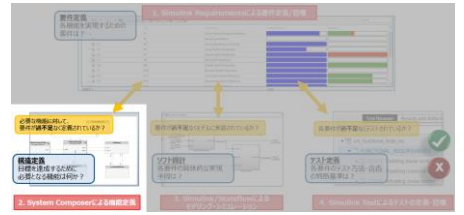




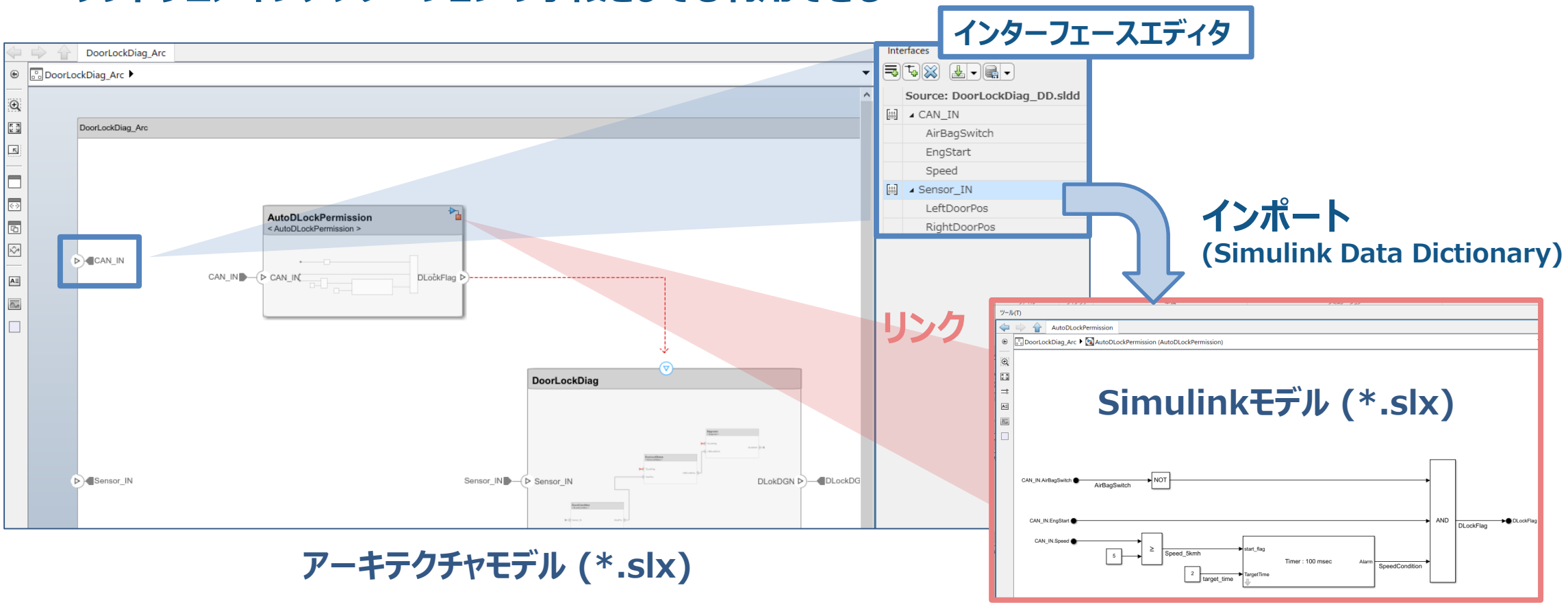
アジェンダ

1. 要件定義/管理（Simulink Requirements）
2. **構造・機能定義（System Composer）**
3. モデリング・シミュレーション（ Simulink/SLDV ）
4. テストの定義・管理（ Simulink Test ）

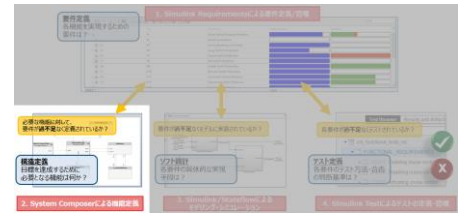
アーキテクチャモデルの作成



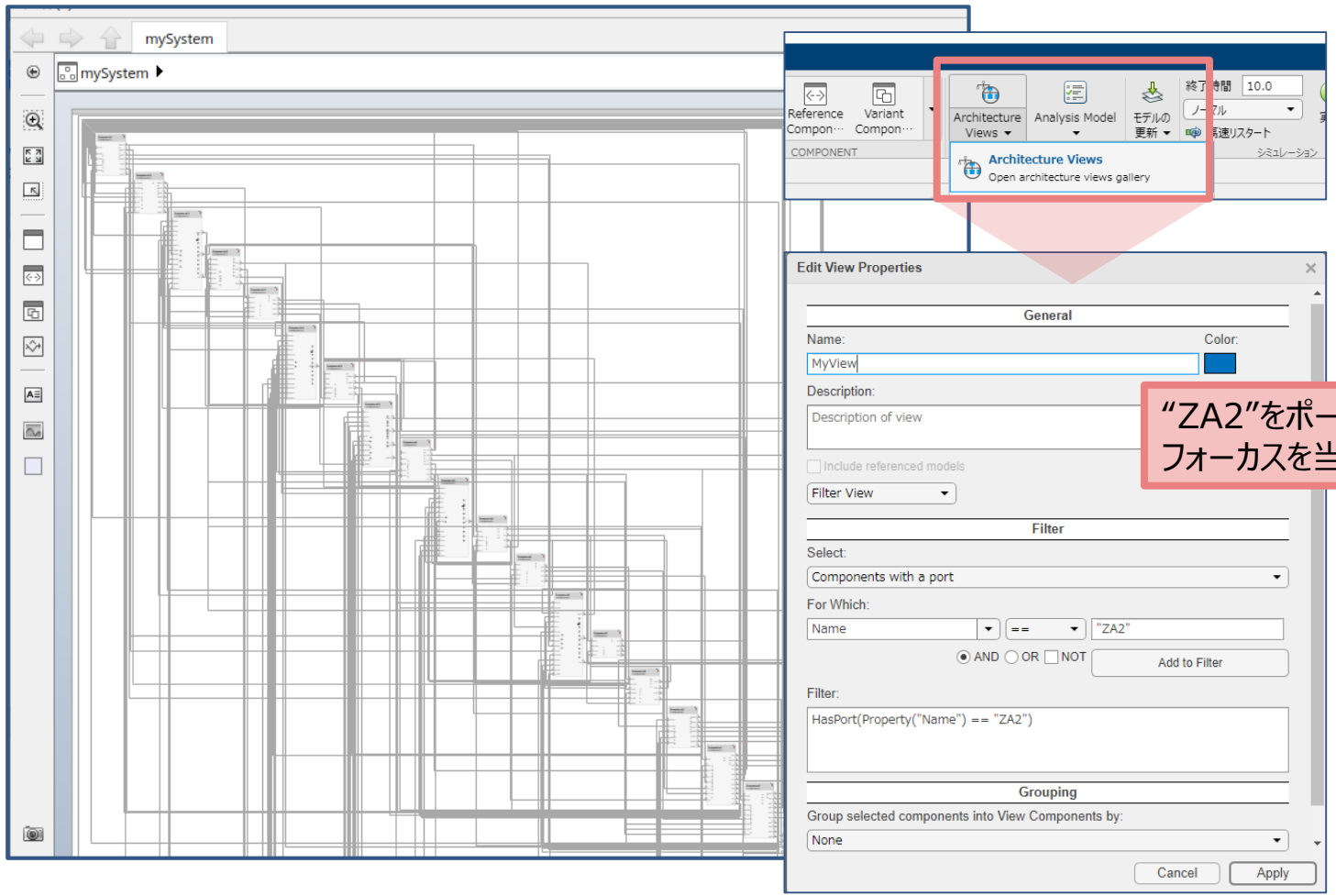
- **アーキテクチャモデルを作成することで、SWの機能構造を明確化できる**
  - Simulinkに似た専用のエディタにより、簡単にアーキテクチャモデルが作成可能
- **アーキテクチャモデル内で定義したコンポーネントとモデルファイルはリンクできるので、ソフトウェアインテグレーションの手段としても利用できる**



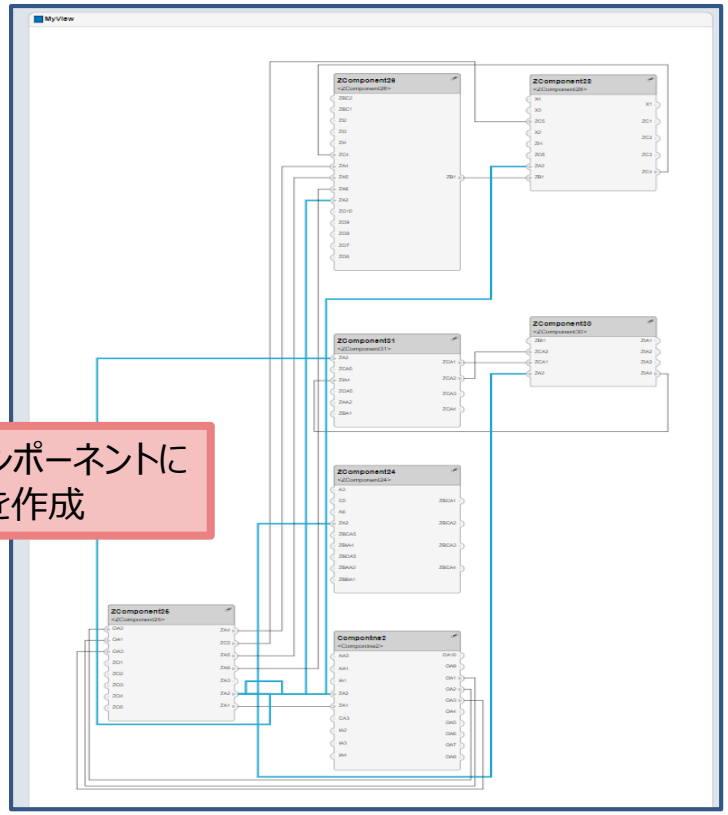
アーキテクチャビュー機能による依存関係分析



- 条件を満たすコンポーネントのみにフォーカスを当てたビューを生成できるので、アーキテクチャモデルを変更することなく必要な情報のみにアクセス可能

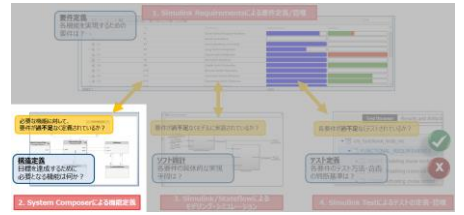


“ZA2”をポートに持つコンポーネントにフォーカスを当てたビューを作成

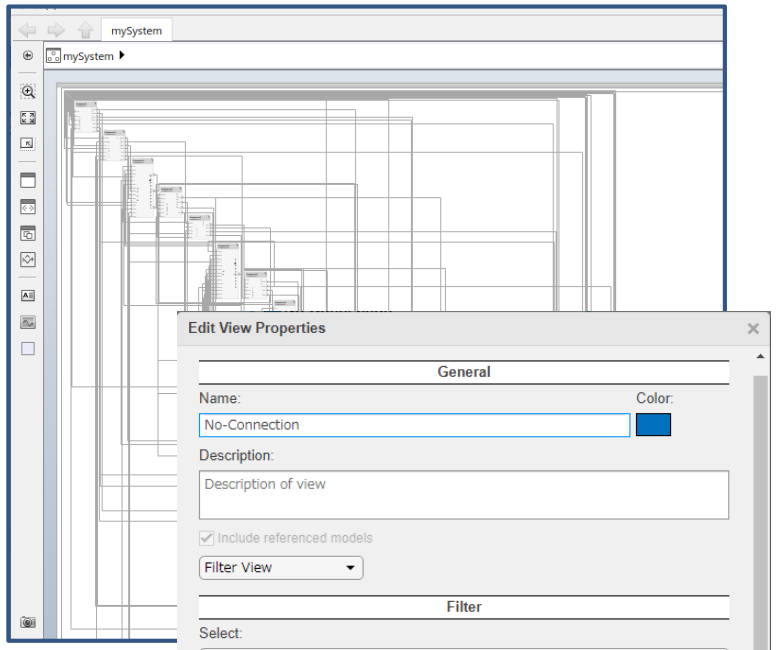


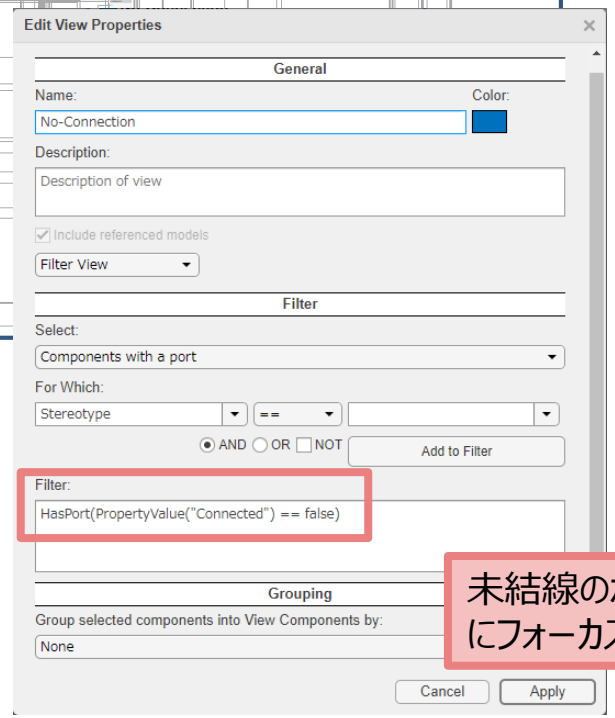
※：ラインのハイライトは手動でおこなっており、現Verでは機能としては実装されていません

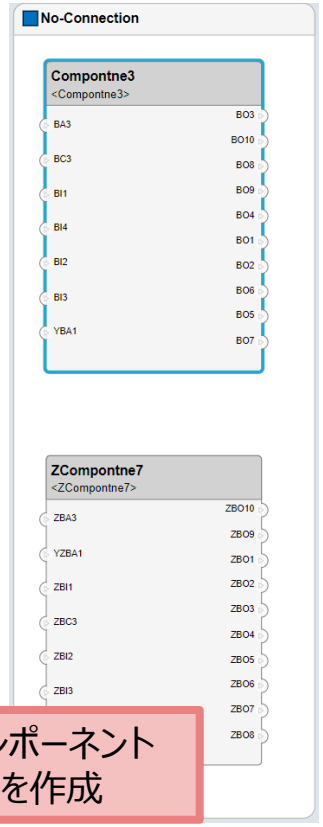
アーキテクチャビュー機能を利用したI/Fチェック



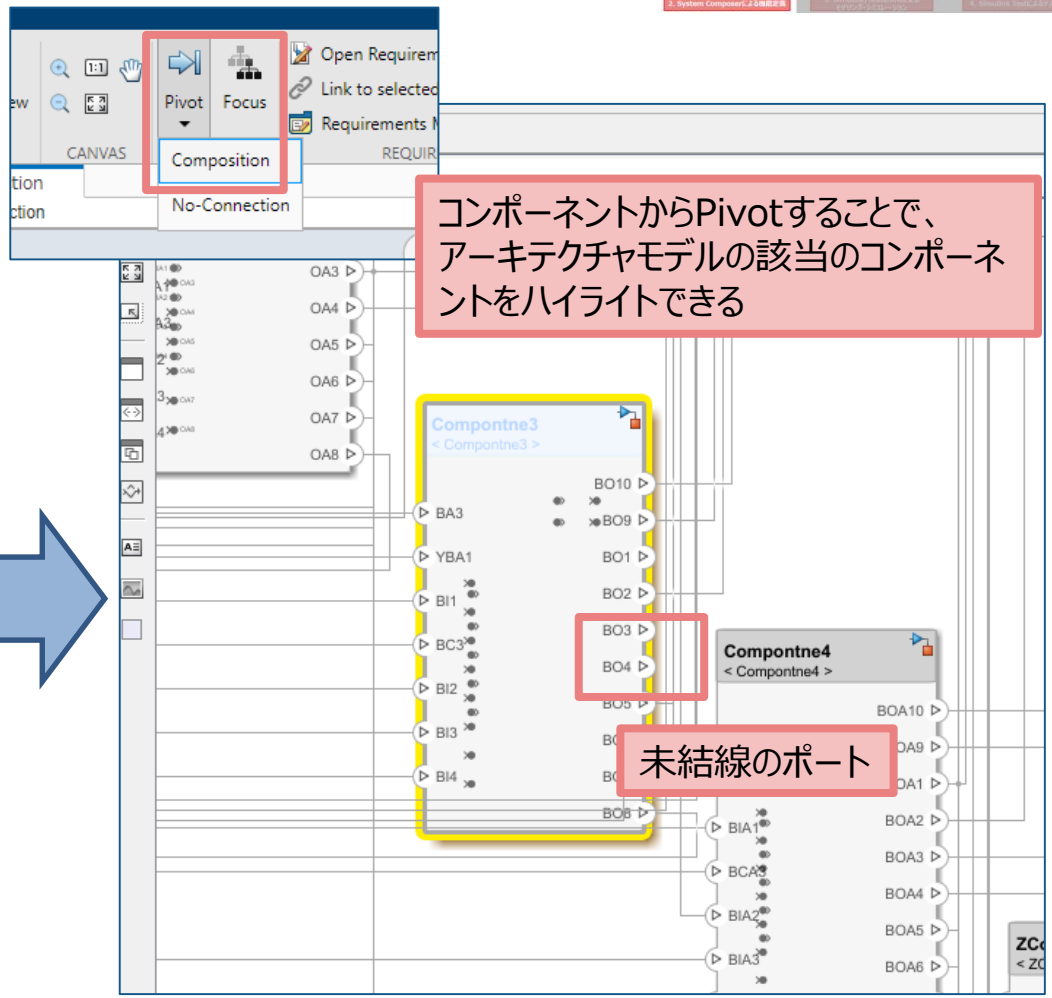
- アーキテクチャビュー機能は、未結線のポートを持つコンポーネントの検索も可能







未結線のポートを持つコンポーネントにフォーカスを当てたビューを作成



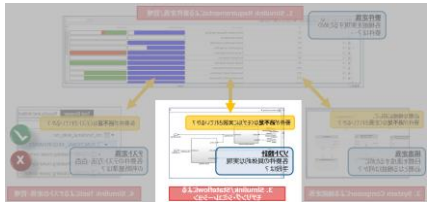
コンポーネントからPivotすることで、アーキテクチャモデルの該当のコンポーネントをハイライトできる

未結線のポート

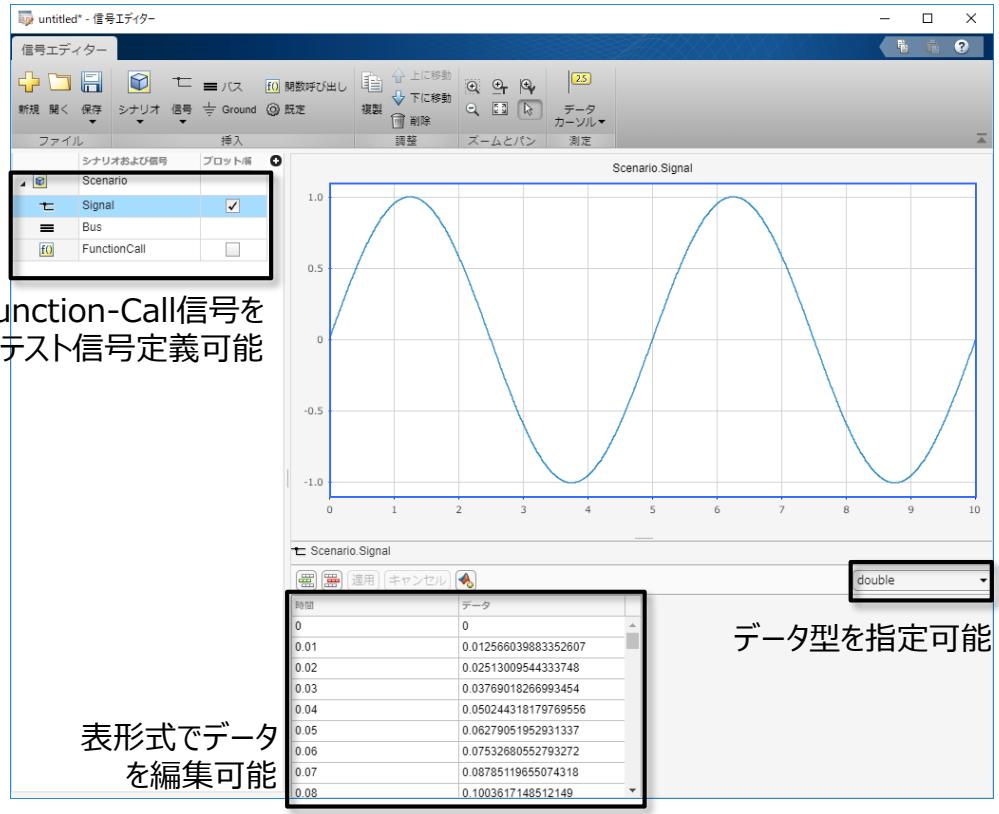
アジェンダ

1. 要件定義/管理（Simulink Requirements）
2. 構造・機能定義（System Composer）
3. **モデリング・シミュレーション（ Simulink/SLDV ）**
4. テストの定義・管理（ Simulink Test ）

Signal Editorによるテストケース作成



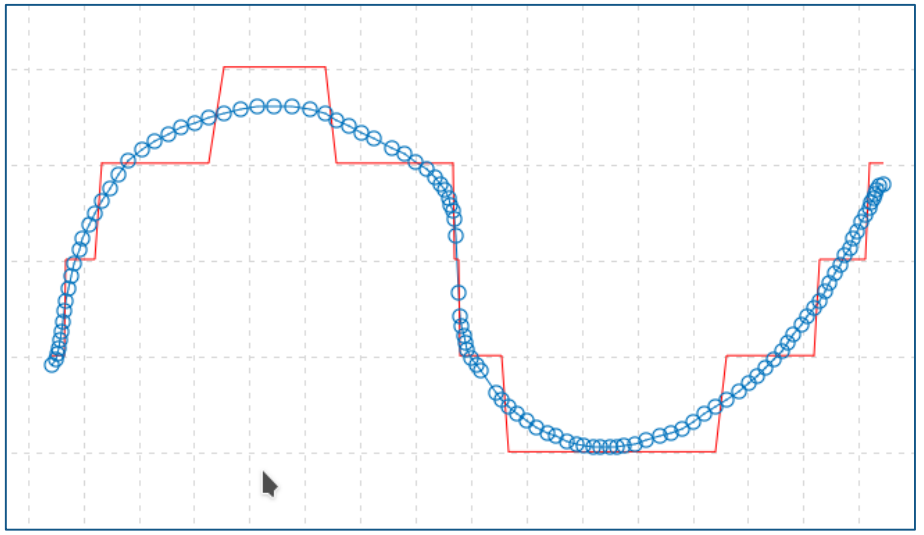
- **Signal Builderの下記課題を解消し、テスト入力をより設計・管理しやすく**
  - 表形式によるデータ点編集、データ型指定、バス信号・Function-Call信号の定義・出力
- **Signal Builder → Signal Editor 変換コマンド (R2018a)**
  - **`signalBuilderToSignalEditor`**



バス信号やFunction-Call信号を含め、各種のテスト信号定義可能

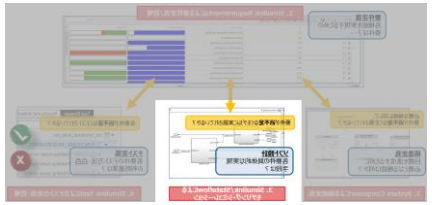
データ型を指定可能

表形式でデータを編集可能

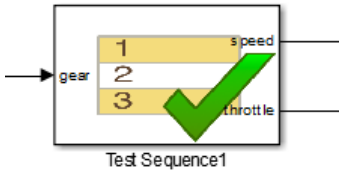


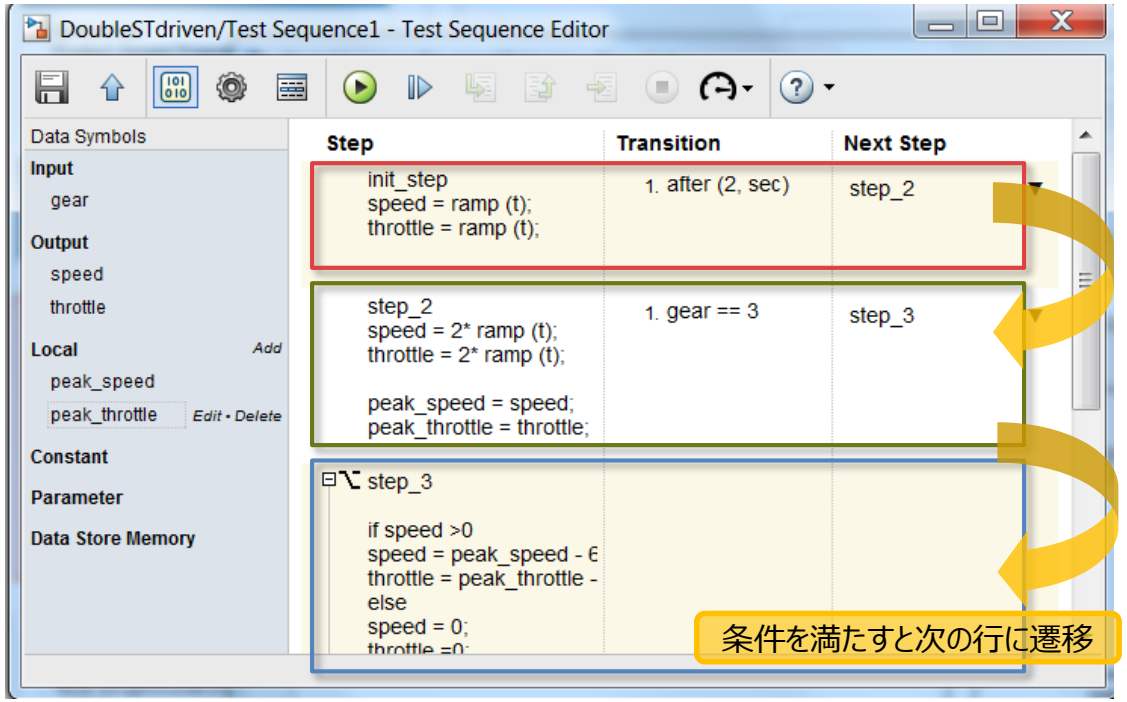
フリーハンドによるテスト信号作成 (R2019a)

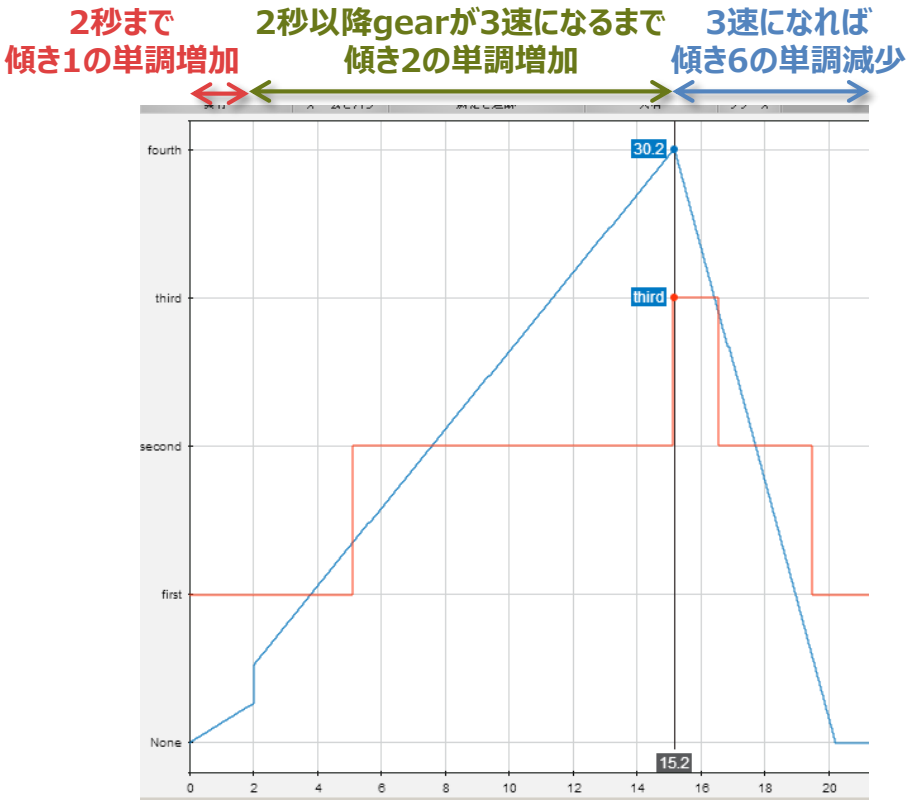
Test Sequenceによるテストケース作成



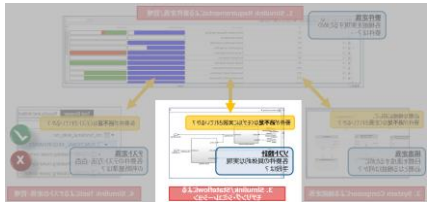
- **シーケンシャルなテストパターンを作成可能**
  - 状態遷移表を用いてシナリオベースでのテストを設計
  - 出力や内部状態の変化によって、テストパターンを切り替え可能
- **診断の挿入による信号チェック・不具合検証も可能**







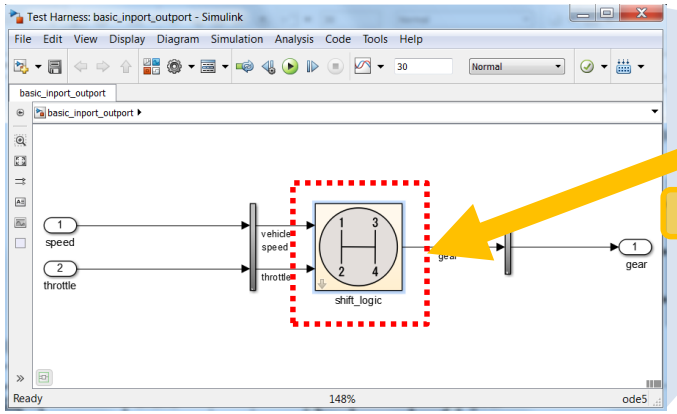
テストハーネス機能によるテスト用モデルの作成

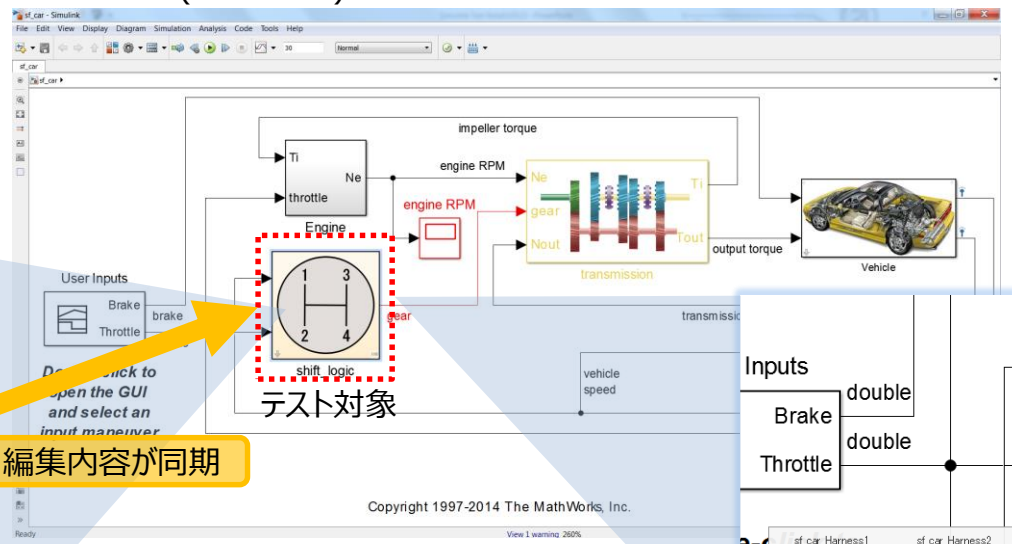


- **サブシステムの単体テストをより実現・管理しやすくなります**
  - サブシステムごと・目的ごとにテストハーネスを作成可能
  - 必要に応じてテストハーネスの保存場所をモデルの内部⇔外部に切り替える可能
  - テストハーネスとメインモデルとは常に同期

メインモデル (モデル本体)

テストハーネス

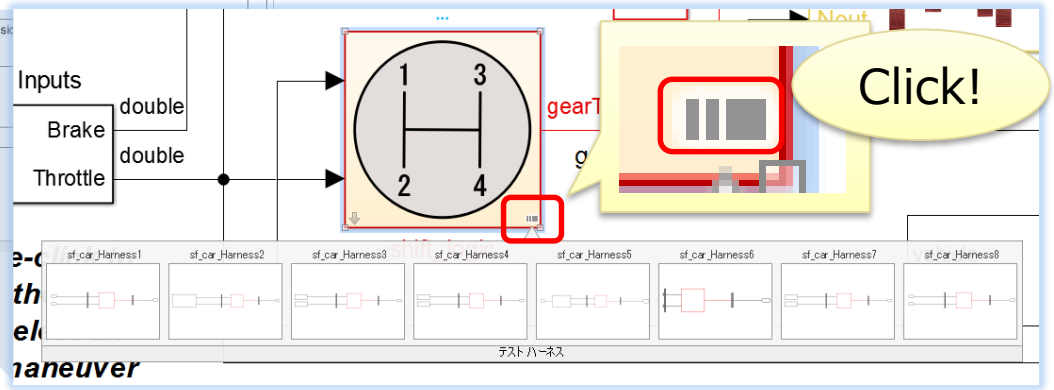




テスト対象

編集内容が同期

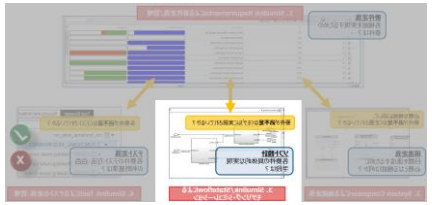
テストハーネスリスト



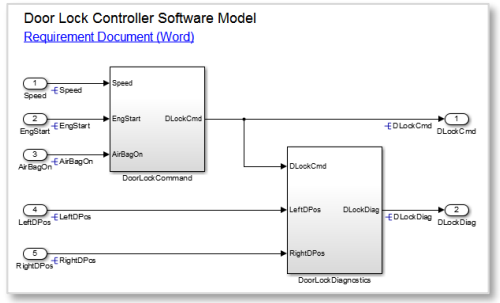
テスト対象に入力と出力を繋げた
ハーネスモデルを自動作成

一つのサブシステムに複数のテストハーネスを
作成することが可能

設計エラー検出によるランタイムエラーチェック



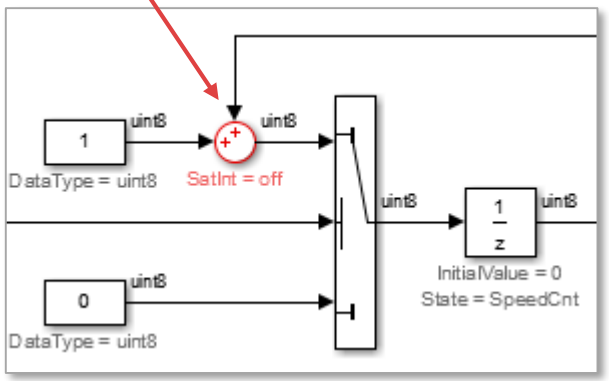
- **I/Fの見直し/回避処理実装の有無のチェックに利用できる**
  - 静的解析をおこなうため、Simulinkの診断機能と異なり、テスト入力の用意は不要
  - デッドロジック、整数オーバーフロー、ゼロ除算、配列の範囲外アクセスがモデルに含まれていないかをチェック
- **エラーが検出された場合、エラー再現用のテストデータを生成可能**



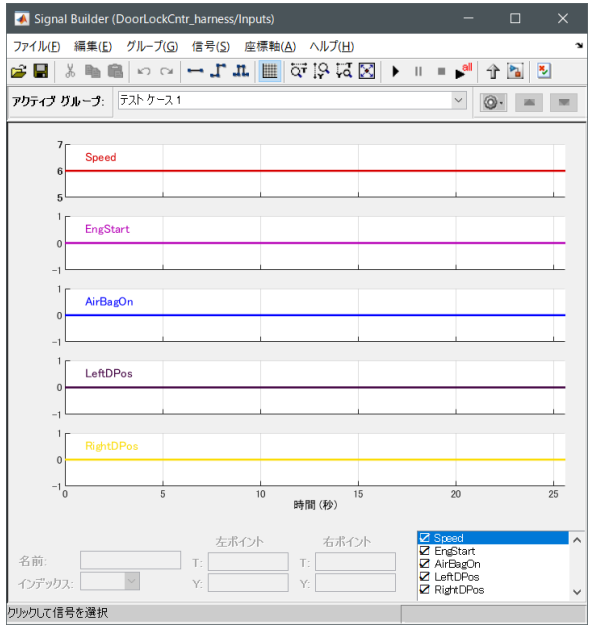
モデル

解析

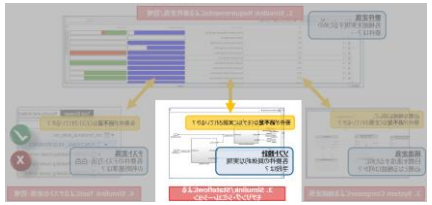
整数オーバーフローリスク有り



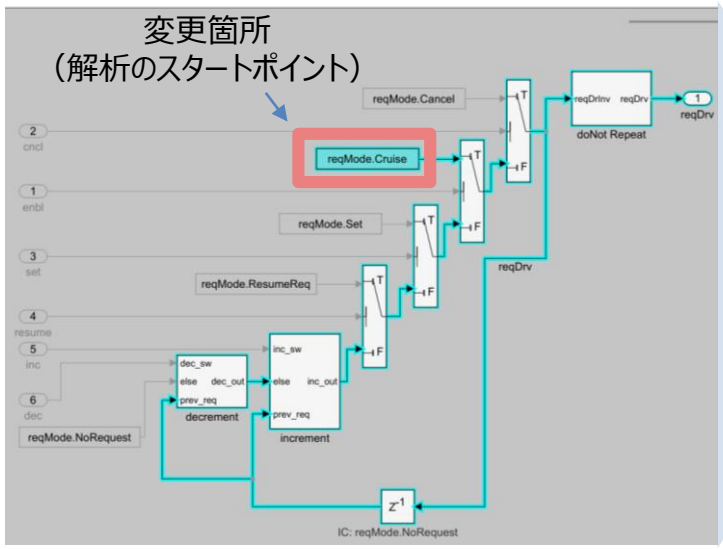
テスト生成

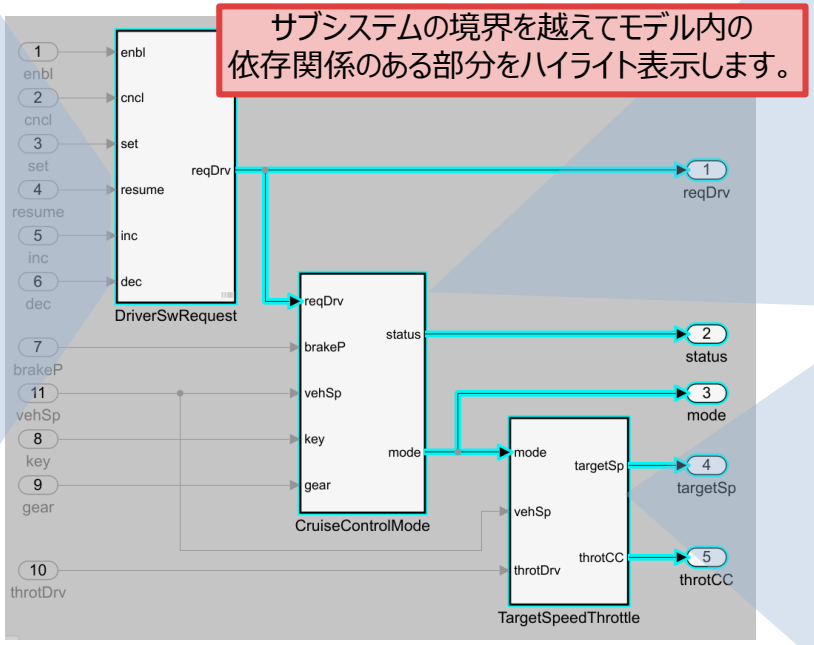


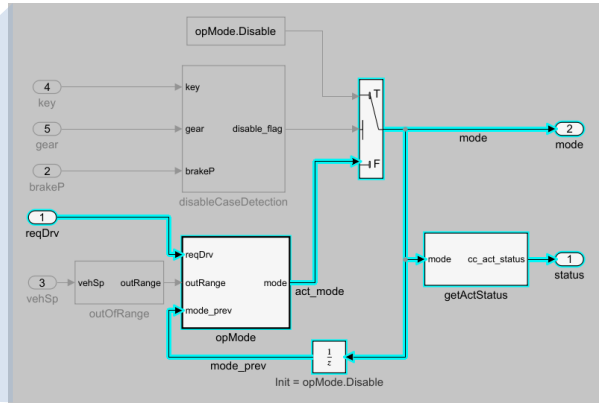
モデルスライサによるデータフローの影響範囲解析

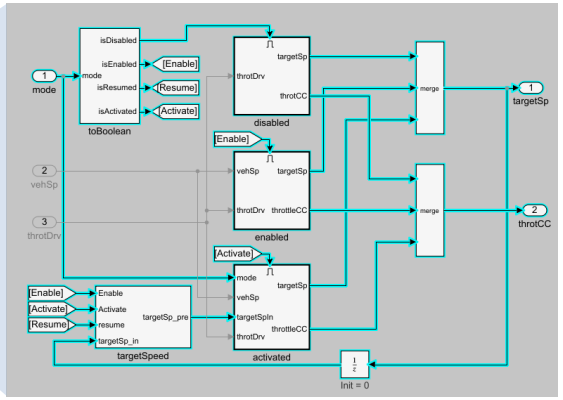


- **特定のブロックや信号に依存関係のある部分を洗い出すことが可能**
  - 修正・変更部分が及ぶ影響範囲を特定可能
  - 任意のシミュレーション時間における影響範囲も特定可能
- **関連するモデル要素のみを残してモデルをスライシングすることも可能**
  - 軽量したスライスモデルをシミュレーションし、影響の結果を確認可能





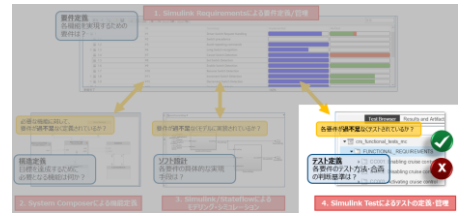




アジェンダ

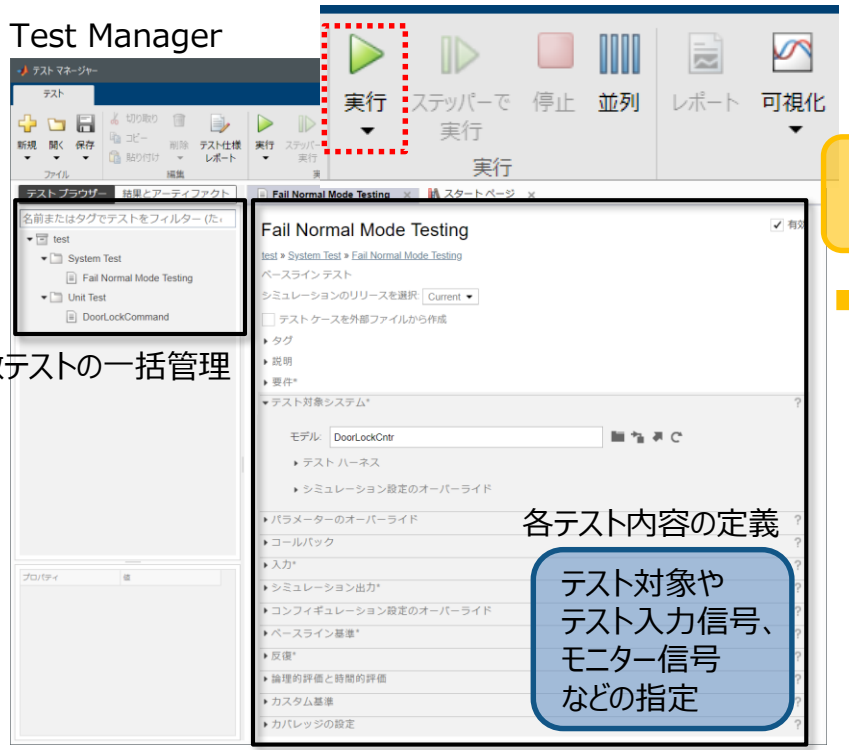
1. 要件定義/管理（Simulink Requirements）
2. 構造・機能定義（System Composer）
3. モデリング・シミュレーション（ Simulink/SLDV ）
4. **テストの定義・管理（ Simulink Test ）**

Test Managerによるテストの管理や自動化



- **テスト定義を資産として管理や再利用が可能**
  - 増大する検証項目の管理が可能
  - テストファイルの再利用で回帰テストを簡単に実現可能
  - テスト定義の共有によるミスコミュニケーション防止や工数削減が可能
- **テスト結果レポートを生成可能**

Test Manager



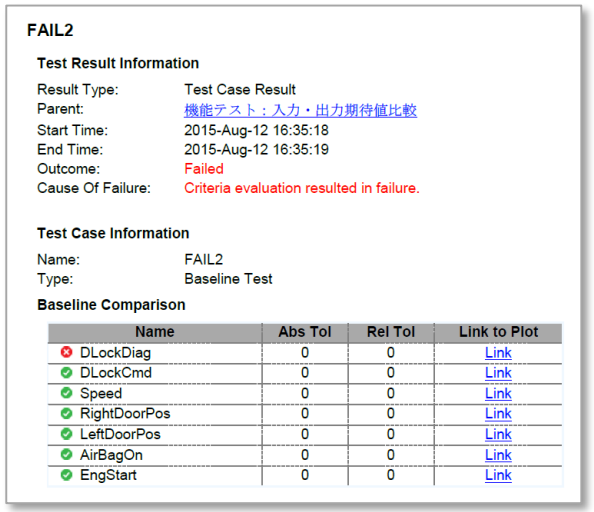
複数テストの一括管理

各テスト内容の定義

テスト対象や
テスト入力信号、
モニター信号
などの指定

テストの一括実行や
テストレポートの生成も可能

テスト結果レポート



FAIL2

Test Result Information

Result Type:	Test Case Result
Parent:	機能テスト：入力・出力期待値比較
Start Time:	2015-Aug-12 16:35:18
End Time:	2015-Aug-12 16:35:19
Outcome:	Failed
Cause Of Failure:	Criteria evaluation resulted in failure.

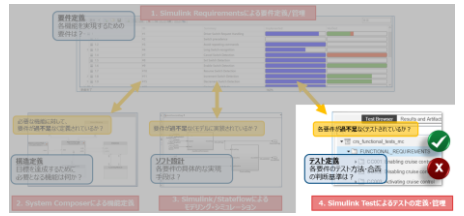
Test Case Information

Name:	FAIL2
Type:	Baseline Test

Baseline Comparison

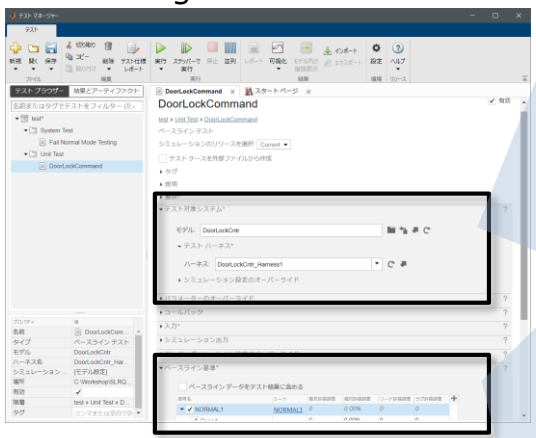
Name	Abs Tol	Rel Tol	Link to Plot
DLockDiag	0	0	Link
DLockCmd	0	0	Link
Speed	0	0	Link
RightDoorPos	0	0	Link
LeftDoorPos	0	0	Link
AirBagOn	0	0	Link
EngStart	0	0	Link

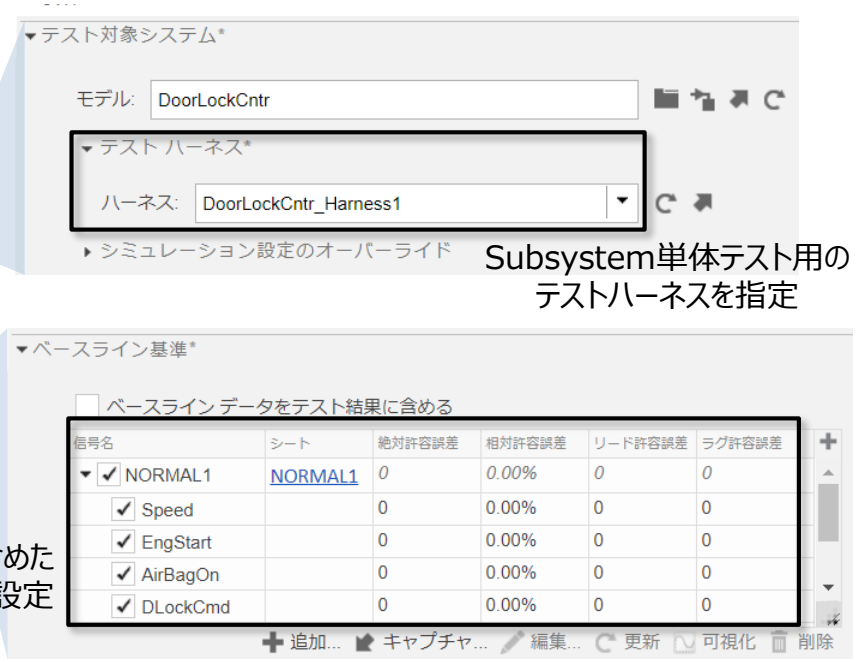
Test Managerによる単体検証



- **Subsystemごとの単体検証を自動化可能**
  - テストの目的に合わせて作られたSubsystem用のテストハーネスをテスト対象として設定可能
  - 出力の期待値をベースラインとして指定すれば、シミュレーション結果との一致性を自動判定可能
  - 許容誤差範囲を設けることが可能。（時間軸やY軸、絶対許容誤差や相対許容誤差を両方設定可能）

Test Manager



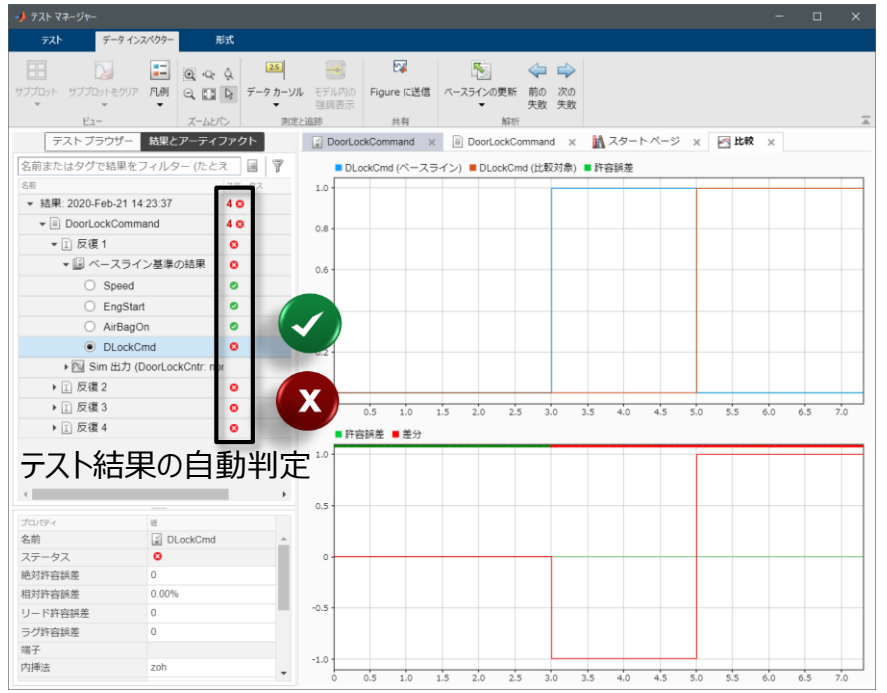


Subsystem単体テスト用のテストハーネスを指定

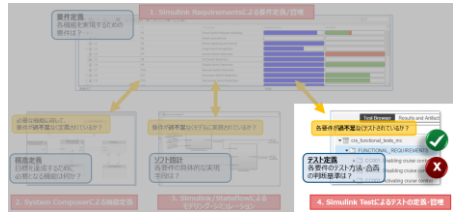
許容誤差を含めたベースラインを設定

テスト実行

テスト結果



Test ManagerによるExcelテストデータの読込み



- **Excelでのテストデータを管理し、直接、Simulinkモデルとマッピング可能**
  - 入出力の信号データタイプや補完方法をExcel上で定義可能
  - シートごとでのインポート可能
  - 入出力データをまとめて定義可能
- **モデルからExcelフォーマットのテスト入力テンプレートも生成可能**

入出力データを纏めて定義

	A	B	C	D	E	F	G	H
1	Time	Speed	EngStart	AirBagOn	LeftDPos	RightDPos	DLockCmd	DLockDiag
2		Type: single	Type: boolean	Type: boolean	Type: single	Type: single	Type: boolean	Enum: DiagStatus
3		Interp: zoh	Interp: zoh	Interp: zoh	Interp: zoh	Interp: zoh	Interp: zoh	Interp: zoh
4	0	0	0	0	0	0	0	NORMAL
5	0.1	0	0	0	0	0	0	NORMAL
6	0.2	0	0	0	0	0	0	NORMAL
7	0.3	0	1	0	0	0	0	NORMAL
8	0.4	0	1	0	0	0	0	NORMAL
9	0.5	5	1	0	0	0	0	NORMAL
10	0.6	5	1	0	0	0	0	NORMAL
11	0.7	5	1	0	0	0	0	
12	0.8	5	1	0	0	0	0	

NORMAL1 | NORMAL2 | FAIL1 | FAIL2

データタイプ・補間方法のインポート対応

シートごとでのインポート可能

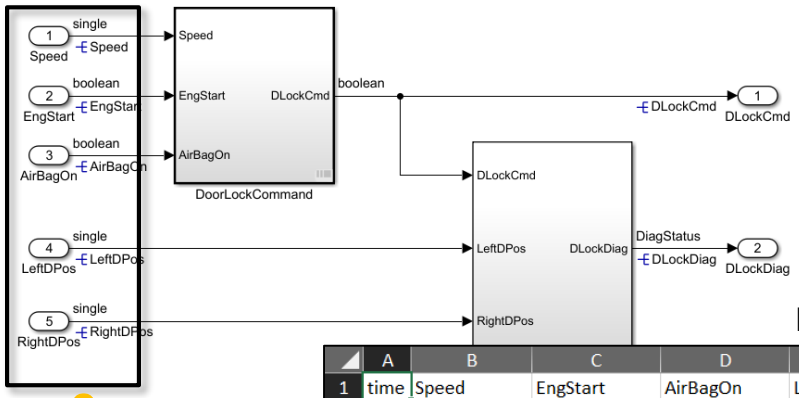
繰り返しシミュレーション実施時のインポートしたシートごとで定義した入力やベースラインを設定可能

反復*

テーブルによる反復*

名前	説明	外部入力	ベースライン
反復 1	なし	NORMAL1	NORMAL1
反復 2	なし	NORMAL2	NORMAL2
反復 3	なし	FAIL1	FAIL1
反復 4	なし	FAIL2	FAIL2

モデル

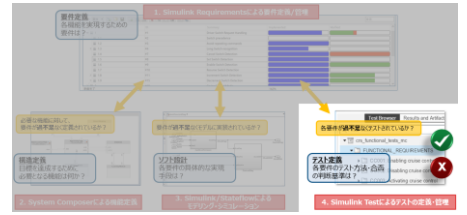


Excelテンプレート

	A	B	C	D	E	F
1	time	Speed	EngStart	AirBagOn	LeftDPos	RightDPos
2		Type: single	Type: boolean	Type: boolean	Type: single	Type: single
3		Source: Input				
4			Interp: zoh	Interp: zoh		
5	0	0	0	0	0	0
6	10	0	0	0	0	0
7						

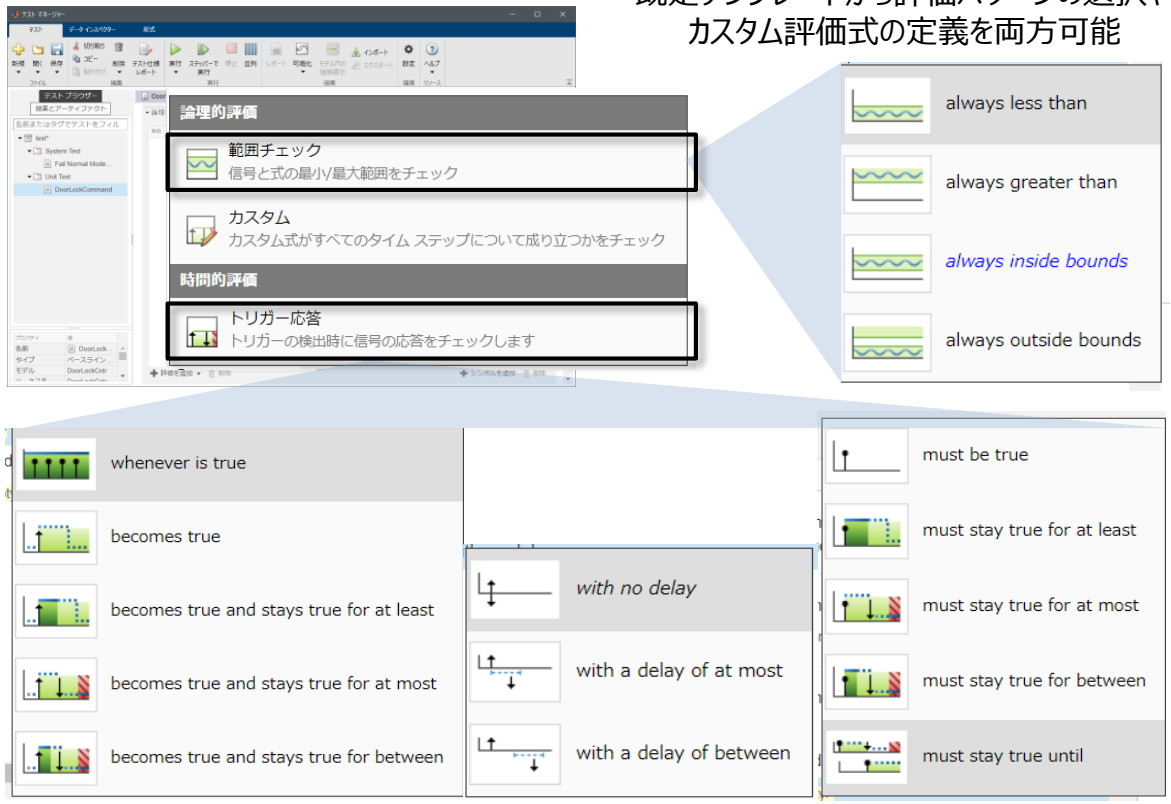
モデルの入力信号の設定からExcelベースのテストテンプレートを生成可能

Test Manager上で要件に基づく評価項目を定義



- **要件を自然言語風テンプレートで記述し、シミュレーションとともに成否を評価可能**
  - 信号値の上下限・範囲内チェック
  - ～秒経過したら～する
  - 特定条件が～回発生後～になる、など。

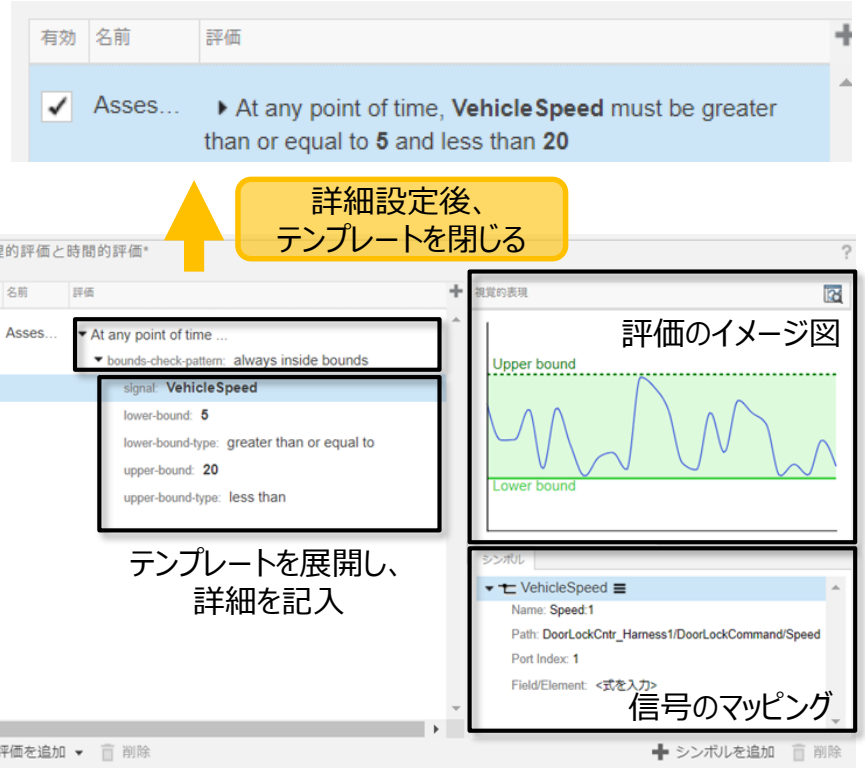
Test Manager



既定テンプレートから評価パターンの選択や
カスタム評価式の定義を両方可能

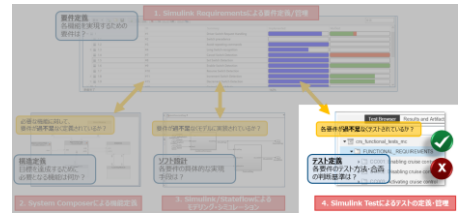
評価項目
の詳細定義

評価項目が自然言語で表現される

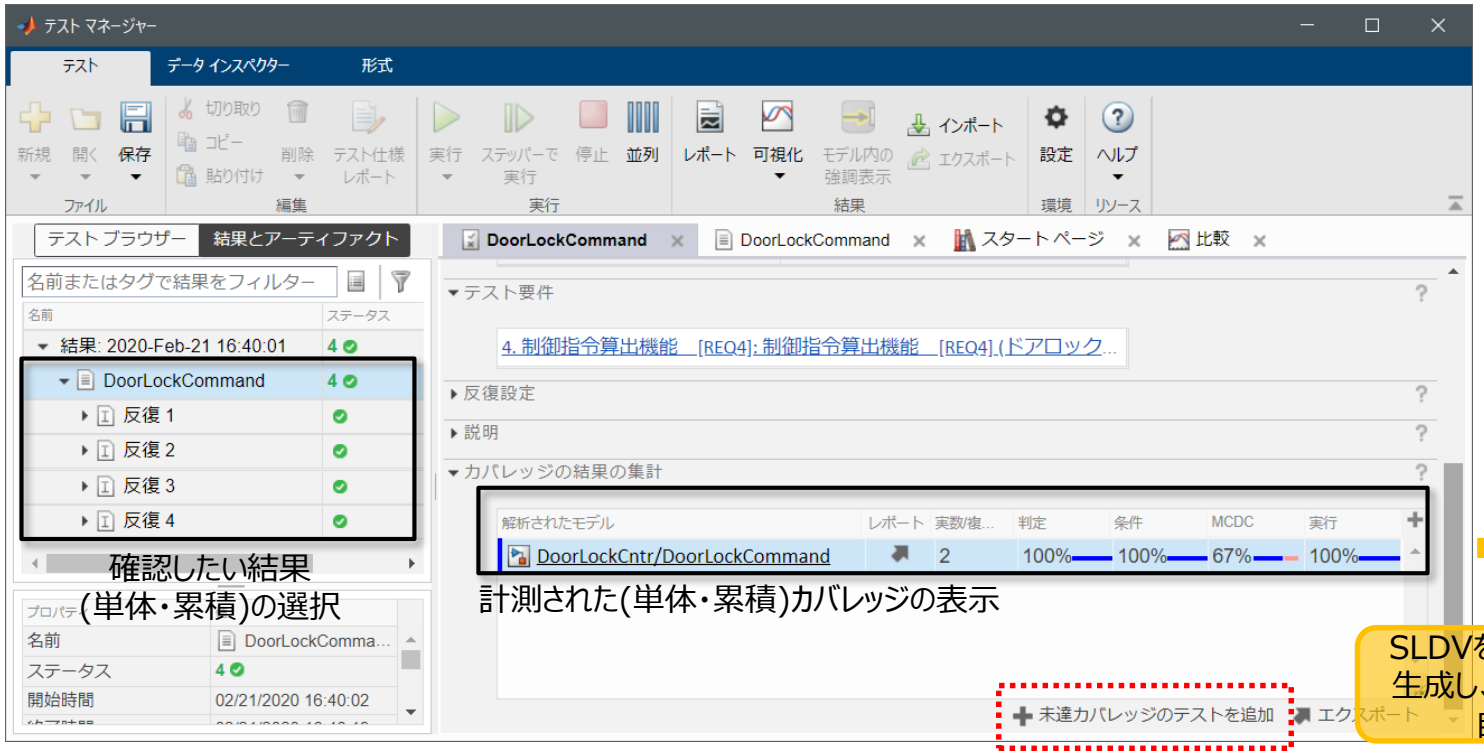


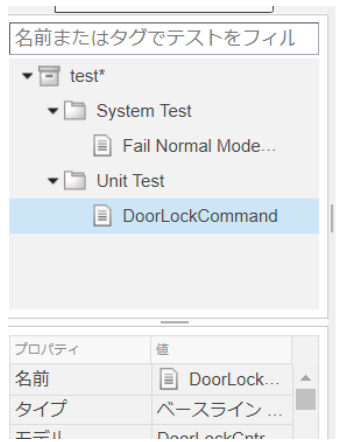
Test Manager上でのカバレッジ補完

Simulink Test
Simulink Design Verifier



- **テストの実行と伴にカバレッジを計測可能**
  - 一括実行した複数のテスト(※)で得られたカバレッジの自動累積。
  - 複数回で実行したテストで得られたカバレッジを簡単に加算可能。
  - カバレッジの確認や詳細レポートの生成はTest Managerのテスト結果画面で行うことが可能。
- **カバレッジが100%でない場合、テストの追加生成もTest Manager上から実行可能**

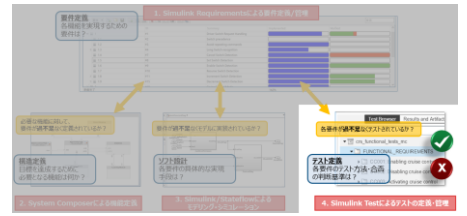




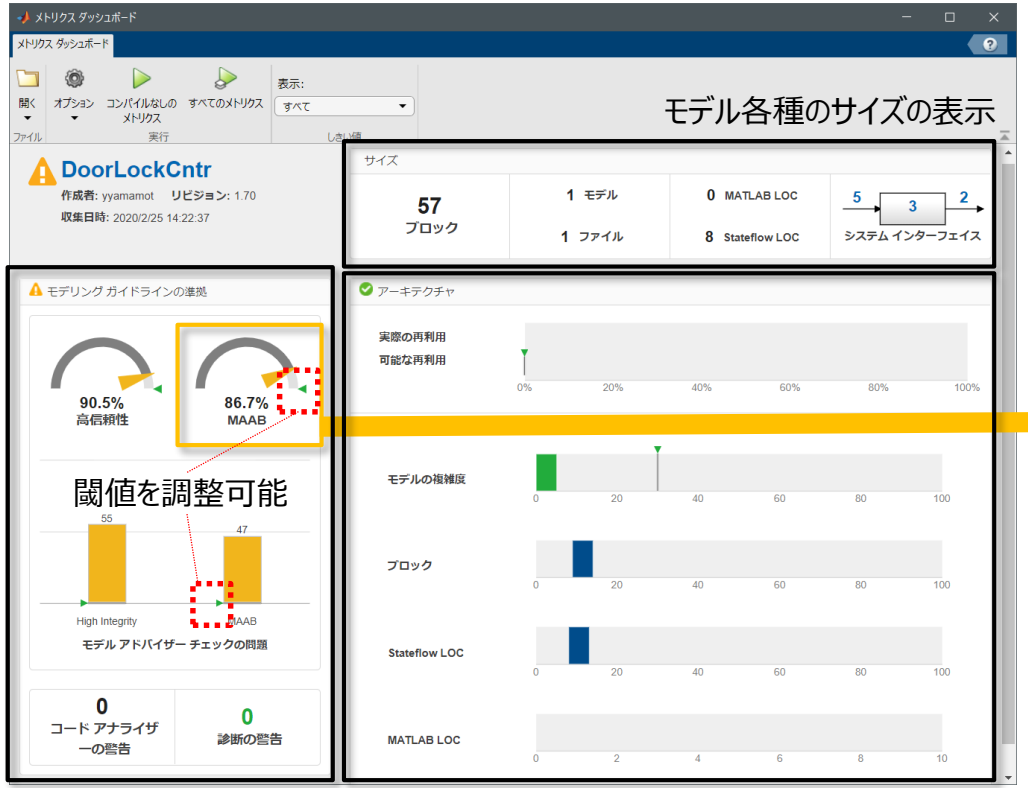
SLDVを用いてテストを追加生成し、Test Managerへ自動登録する

メトリクスダッシュボードによるモデルの定量化測定

Simulink Check



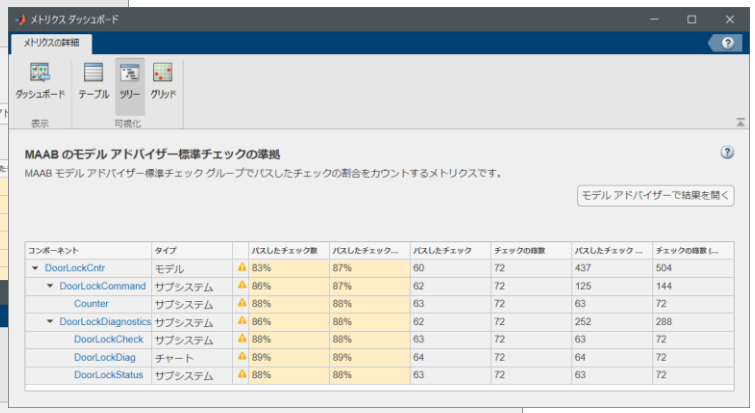
- **モデルのメトリクスを測定可能**
  - モデルのサイズ、複雑度、可読性を定量化可能
  - モデルガイドラインの準拠結果をまとめる
  - ダッシュボードで各測定結果をグラフ表示して可視化可能



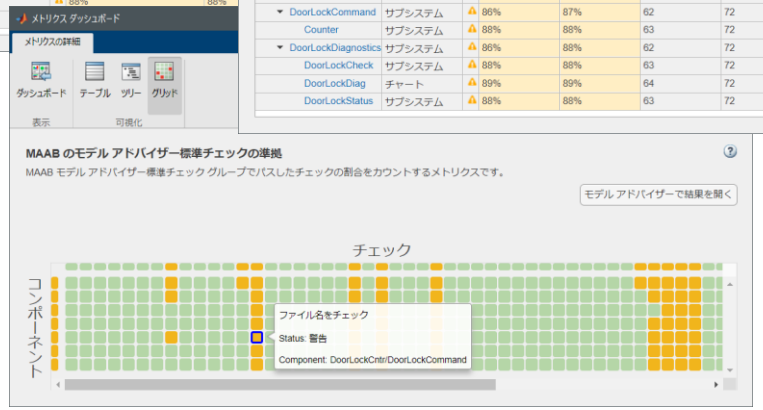
モデルガイドラインの準拠結果まとめ　　モデル各種のサイズの図示化

テーブル表示

ツリー表示



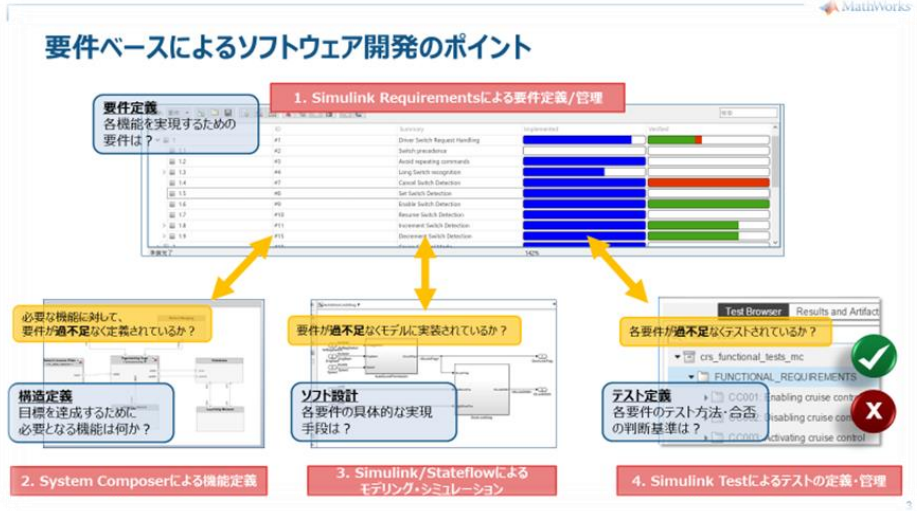
モデルガイドラインの準拠結果を
更にテーブル・ツリー・グリッド
フォーマットで纏めることが可能



グリッド表示

まとめ

- **本Webセミナーの説明内容の振り返り**
  - ✓MBDによる要件をベースとした開発を進める上で有用と思われるMathWorksツールの機能および使い方をご紹介
- **MATLABプロダクトの強み**
  - ✓SimulinkをはじめとしたMathWorksツールは、要件をベースとした開発が効率的に行える環境を提供。
  - ✓QCDの向上を目的とした製品開発へのMBD取り組みを促進。



規格認証を前提とした製品開発における
MBD適用に際し、MathWorksツールを
是非ご活用下さい。

関連情報

- **製品紹介ページ**
  - **Simulink Requirements（要件の作成と管理およびモデルとのリンク）**
  - https://jp.mathworks.com/products/simulink-requirements.html
  - **System Composer（システムおよびソフトウェアアーキテクチャの設計と解析）**
  - https://jp.mathworks.com/products/system-composer.html
  - **Stateflow (ロジックベースの制御)**
    https://jp.mathworks.com/products/stateflow.html
  - **Simulink Check (モデリング標準への準拠性を検証)**
    https://jp.mathworks.com/products/simulink-check.html
  - **Simulink Coverage (モデルおよび生成コードのテストカバレッジを測定)**
    https://jp.mathworks.com/products/simulink-coverage.html
  - **Simulink Design Verifier (設計エラーの特定、テストケースの生成)**
    https://jp.mathworks.com/products/sldesignverifier.html
  - **Simulink Test (統合・単体テスト管理とテストの自動化)**
    https://jp.mathworks.com/products/sldesignverifier.html
  - **その他のプロダクト**
    https://jp.mathworks.com/products.html

関連情報

- **オンデマンドWebセミナー**
  - **今からはじめるSimulink入門**
    https://jp.mathworks.com/videos/introduction-to-simulink-120092.html
  - **Stateflowで状態遷移設計をより便利に**
    https://jp.mathworks.com/videos/introduction-of-stateflow-106650.html
  - **制御モデルを活用した量産／組込みソフト開発ソリューション**
    https://jp.mathworks.com/videos/embedded-production-software-development-by-model-based-design-107673.html
  - **Simulink Testでモデルのテスト実行・管理をより便利に**
    https://jp.mathworks.com/videos/model-test-and-test-management-made-easy-with-simulink-test-101294.html

  - **30日無料の評価版を試す**
  - **製品およびサービスのお見積もり**

Accelerating the pace of engineering and science