IBM Research

SysML™によるモデル駆動型システムズ・エンジニアリング

2008年4月25日

日本アイ・ビー・エム株式会社東京基礎研究所
赤津　浩之、石川　浩

CONTENTS

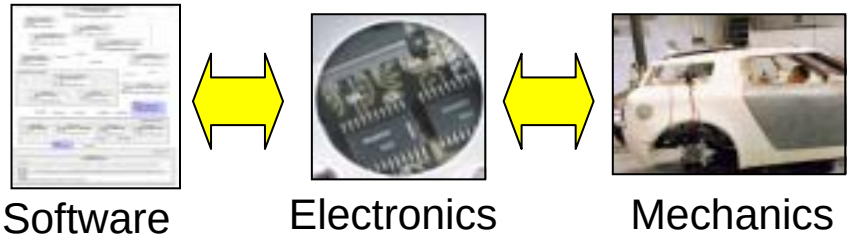
- メカ・エレキ・ソフトの統合設計環境が必要となる背景
- メカ・エレキ・ソフトのモデリング
  - Model-driven systems engineering (MDSE)の概要
  - SysML™・MARTE™の活用
  - コントローラー・プラント統合モデリング
  - トレーサビリティーとインパクト分析
  - モデルの実行と検証
- Eclipse™ベースでの統合設計環境
- まとめ

CONTENTS

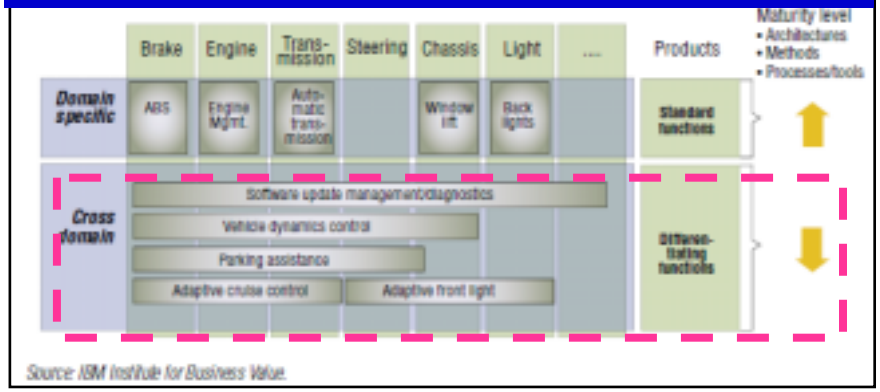
- メカ・エレキ・ソフトの統合設計環境が必要となる背景
- メカ・エレキ・ソフトのモデリング
  - Model-driven systems engineering (MDSE)の概要
  - SysML™・MARTE™の活用
  - コントローラー・プラント統合モデリング
  - トレーサビリティーとインパクト分析
  - モデルの実行と検証
- Eclipse™ベースでの統合設計環境
- まとめ

システムズの複雑化により、製品品質の維持や開発コストの低減が難しくなってきています

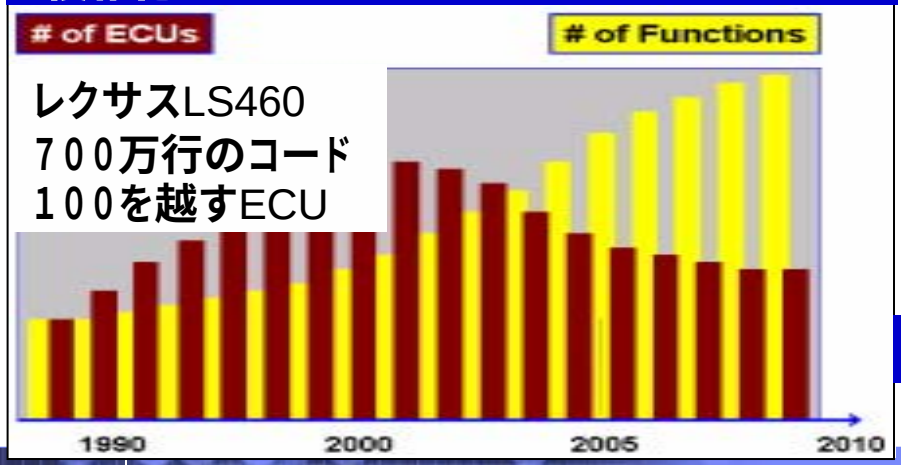
メカ・エレキ・ソフトが相互に関係する機能の設計が増えてきている



設計組織や領域を跨いだ機能の設計が増えてきている



差別化を支えるために、SWは膨大になり、ECUも複雑化してきている



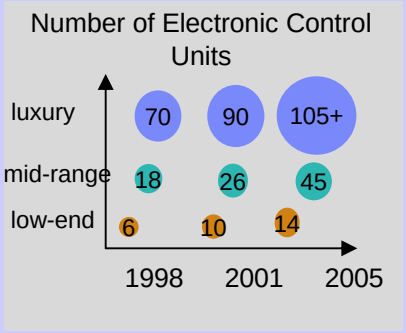
差別化のために、自動車設計は、複雑化している

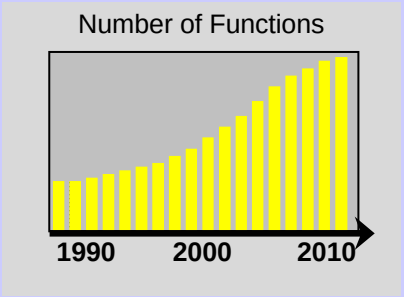
http://www-03.ibm.com/solutions/plm/doc/content/bin/g510_3987_embedded_systems_overhaul.pdf?g_type=rhc

This industry is reaching a critical state in several areas which is driving the imperative for change

Complexity of Product

The number of electronics controls units continues to grow new functions are continuously added

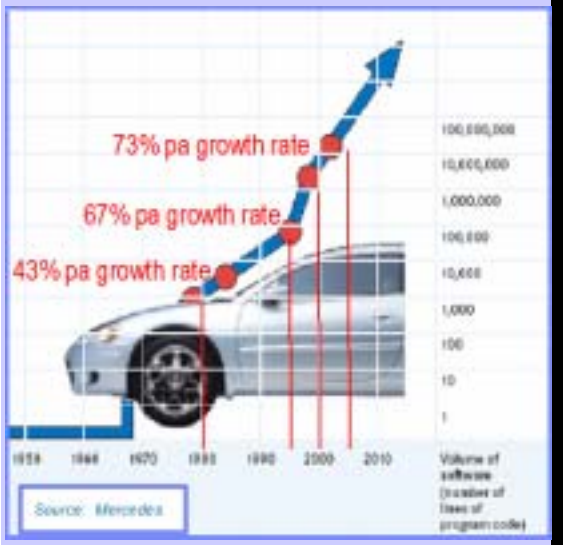




New System Architectures Required

Embedded Skills Needed

Lines of code are growing in near triple digits while skilled resources are growing in single digits



New Competencies & Approaches Required

Quality/Warrant Costs

Currently 30% of warranty issues are associated with Electronics and Software (around 10 B$ per year)

Plus costs of:

Recalls

Liability

Brand Damage

Potential to grow exponentially with code growth

Doing Nothing is Not Affordable

システムズの開発には、開発から量産段階に亘る、様々な課題の解決を求める声が上がっています

開発フェーズ		製品開発における課題
モデル構築期	企画・プランニング	要求管理(機能・非機能要求の設計との連携)
		要求変更時のインパクト分析
	システムズ設計	システムレベル・モデリング
		トレードオフ分析
		システムレベル・モデルの検証
		組み込みモデルの動的振る舞いの検証
		シミュレーションの連携
		アーキテクチャー設計
		OEM・サプライヤー間でのモデルを用いた製品仕様の共有
	ドメイン設計	マルチコア向け組み込みシステム設計
		ドメイン設計時の整合性とトレーサビリティー管理
モデル使用期	システムテスト・検証	テスト・パターン生成
		テスト・カバレージ向上、リソース低減
	量産	欠陥原因のトレーサビリティー管理
運用上の課題		ベース・派生品管理
		スタンダードに基づいた設計 (AUTOSAR etc.)
		アセットの検索機能
		プロジェクト管理
		成果物管理 (変更管理、版管理等)

CONTENTS

Model-Driven Systems Engineering (MDSE)とは、要求をモデルを用いて分析・整理・検証することで、要求を満足し、迅速に品質の高い設計を実現する手段です

IBM MDSE methods & tools

図面と文章による従来の設計プロセス

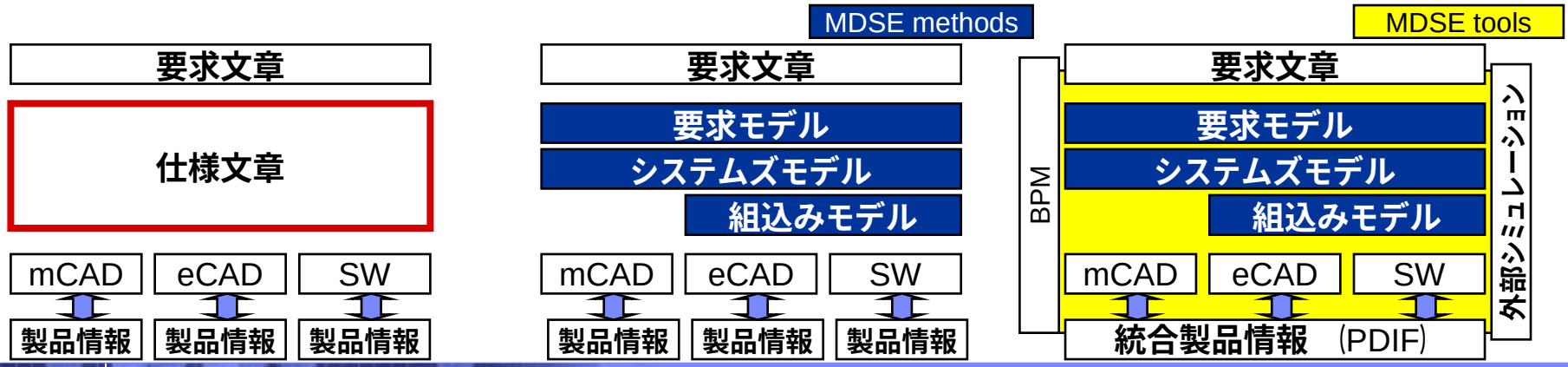
- 図面を用いた構造表現と文章による要求仕様
- 人による要求仕様の分析と、不明確な図面との関連付け
- ドメイン間の不明確な設計および変更の整合性管理

要求・構造・振舞・制約・機能のモデリング

- プラットフォームに依存しない分析モデリング手法を用いた要求分析による要求内容と矛盾の明確化手法
- ドメイン（物理）モデルと分析モデルを連携したモデルの最適化手法
- トレードオフ分析手法
- 派生管理手法
- 成果物管理手法

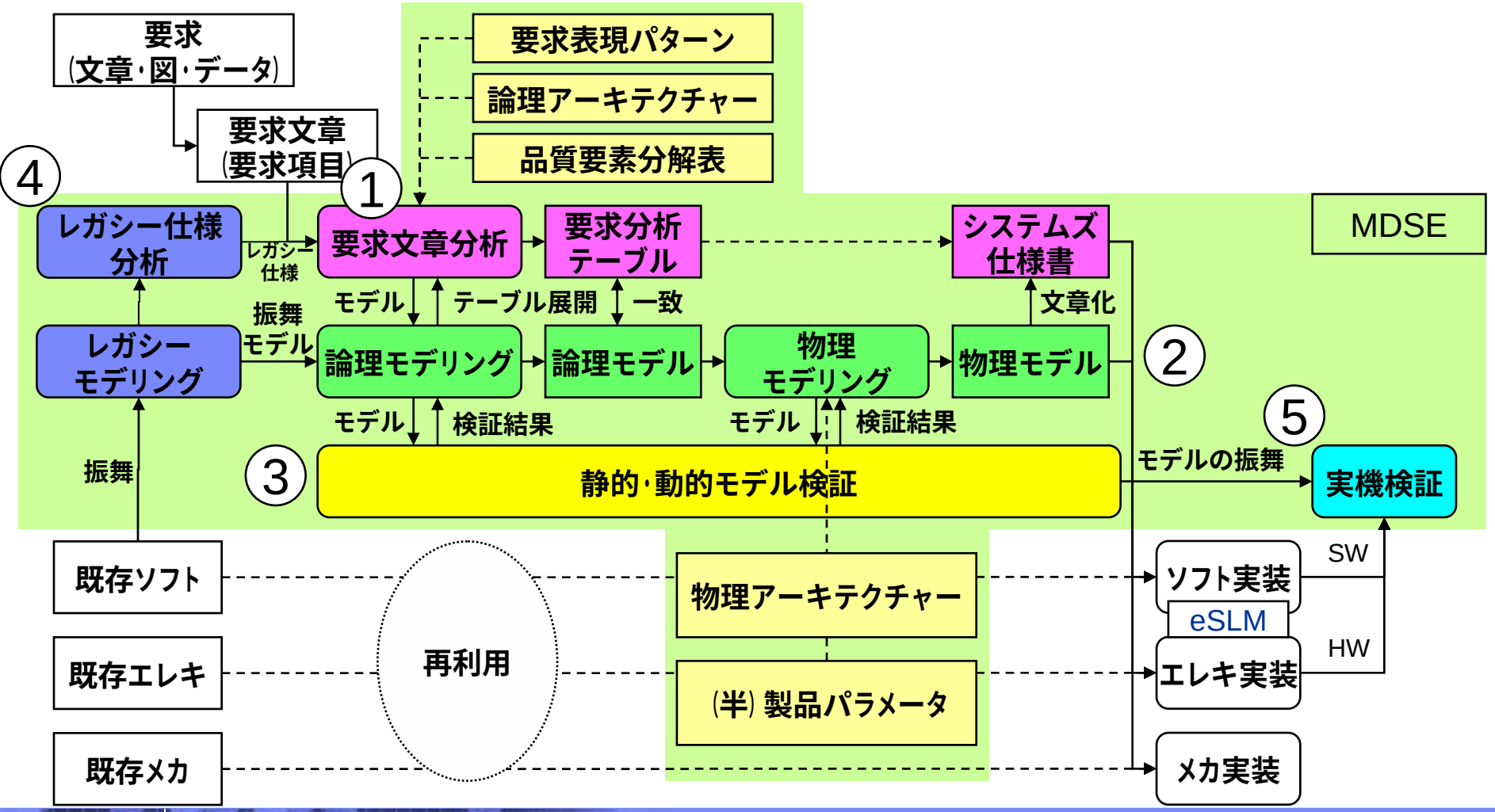
ツールを用いたモデルの実行

- MDSE toolsとIBM SW製品群によるモデルの実行
- モデルの動的検証とその結果による実機検証の効率化
- 製品情報との連携
- ドメイン毎の設計ツールとの連携
- 統合製品情報との連携
- 設計プロセス手法のガイド
- 外部シミュレーションとの連携 (Whole vehicle simulation)



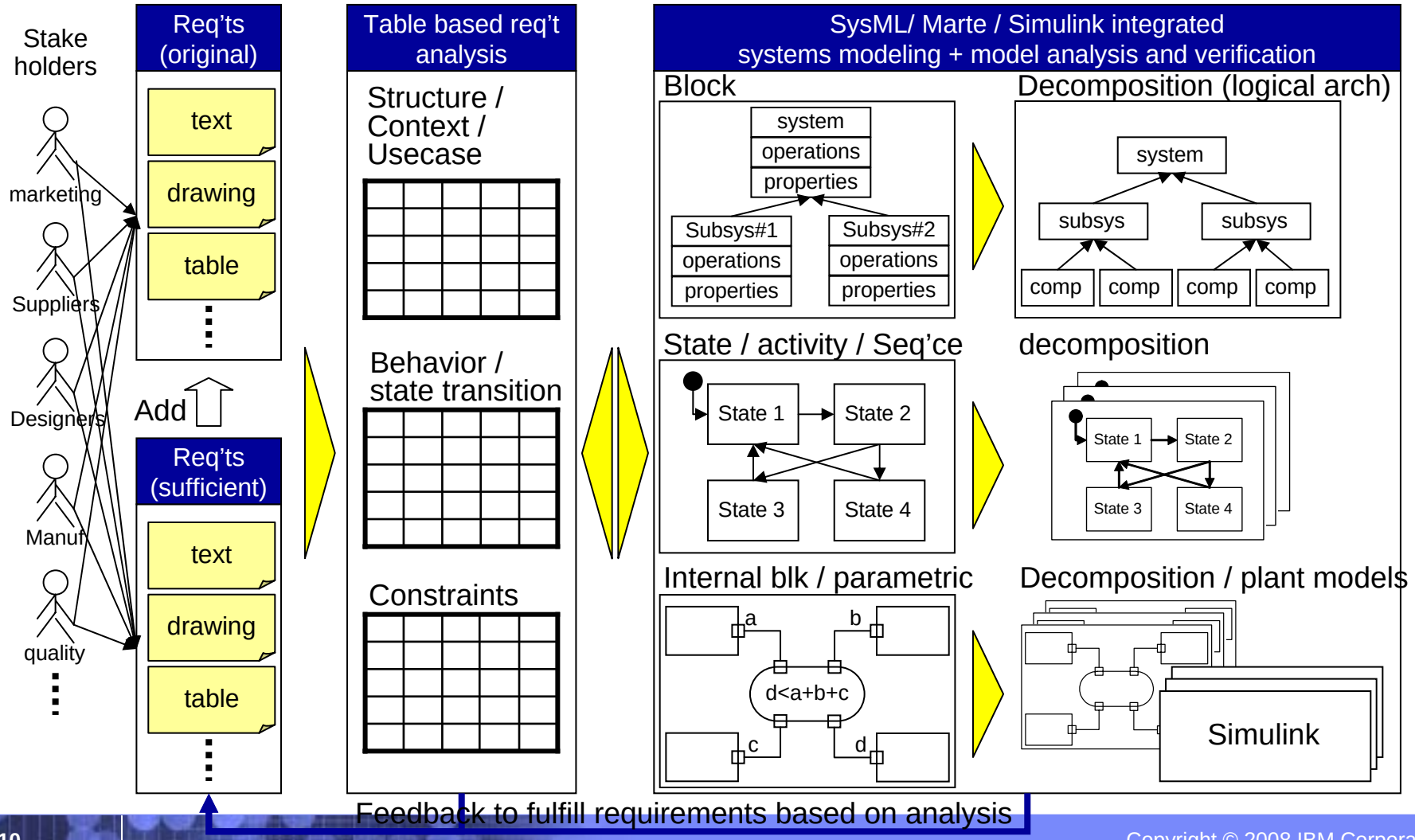
MDSEのモデリング・プロセス全体像

MDSEは、①要求文章分析・②モデリング分析・③モデル検証によって、システムズ・レベルの仕様書を精緻化することに加え、④レガシー・モデリングと⑤実機検証で、システムズ開発の効率と品質を向上させます



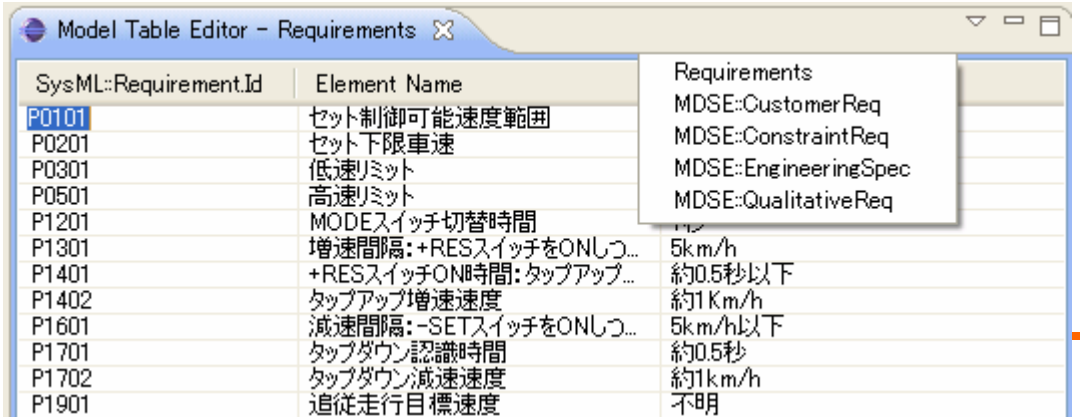
システムズモデリングの概要

MDSEのシステムズモデリングは、要求文章の分析を行い、それを荒い粒度から徐々にコンポーネントへ向けてモデリングしていき、システムズのアーキテクチャーを整理しつつ、モデルの精度を高めていきます



主なツールの機能①：表・マトリックス形式でのモデルの編集・分析

モデルのサイズが大きくなるシステムズ・モデリングでは、モデル全体を表形式で編集・分析できる機能が便利となり、さらに表がモデルのViewとなっていて、内容の変更がモデル自身と不整合を起こさない仕組みであることが重要となります



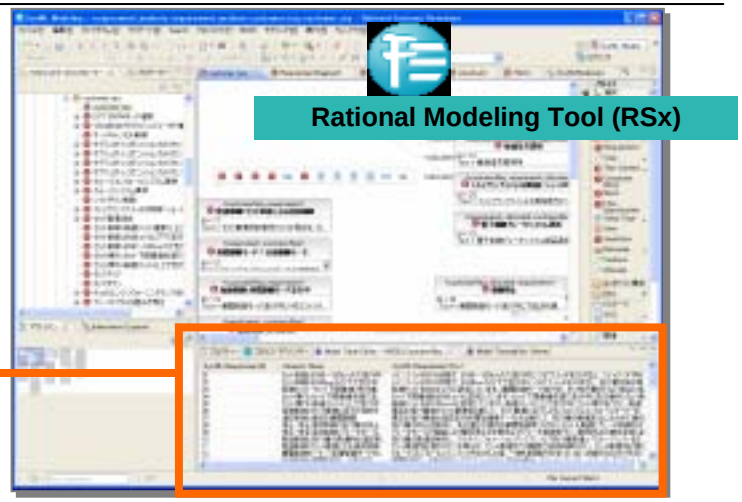
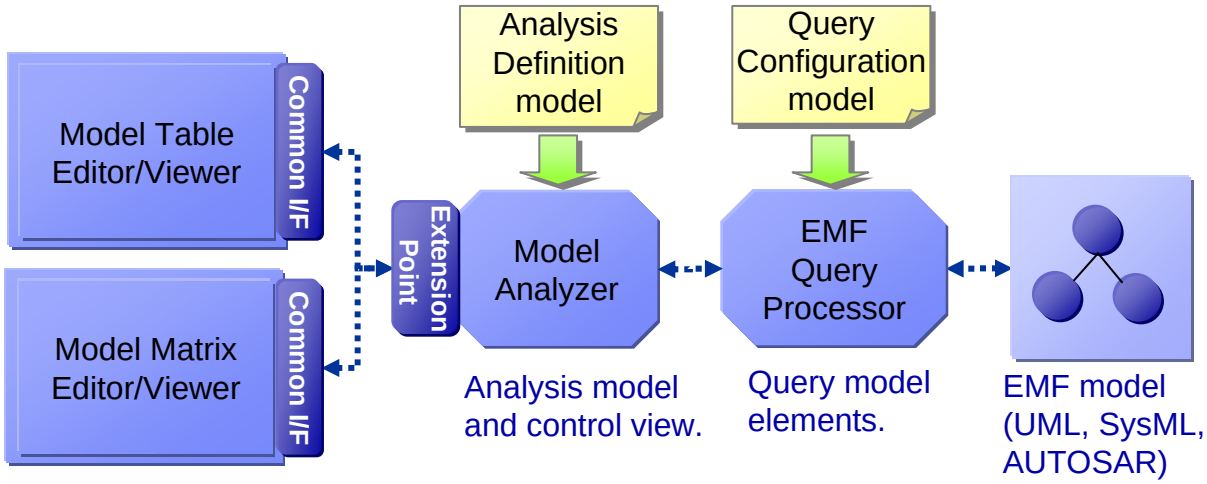


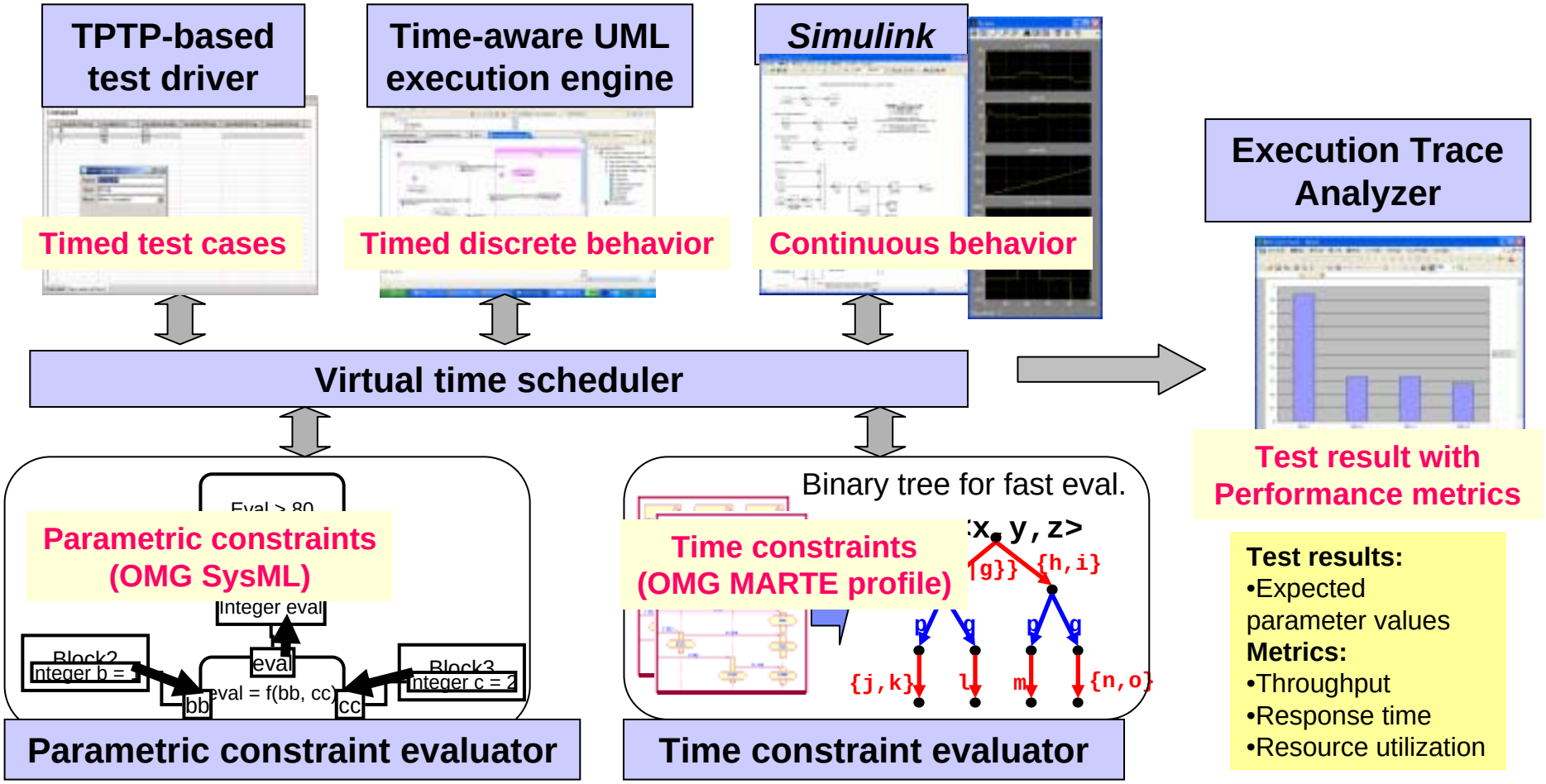
Table / Matrix based model edit / analysis / verification

- Enable to see preferable model properties and edit them with table view according to query configuration model.
- Enable to check model statistics (e.g. the number of ports and interfaces and dependencies) according to analysis definition models.
- Custom viewer and analyzer can be added through extension point.



主なツールの機能②：システムズ・モデルの静的・動的検証

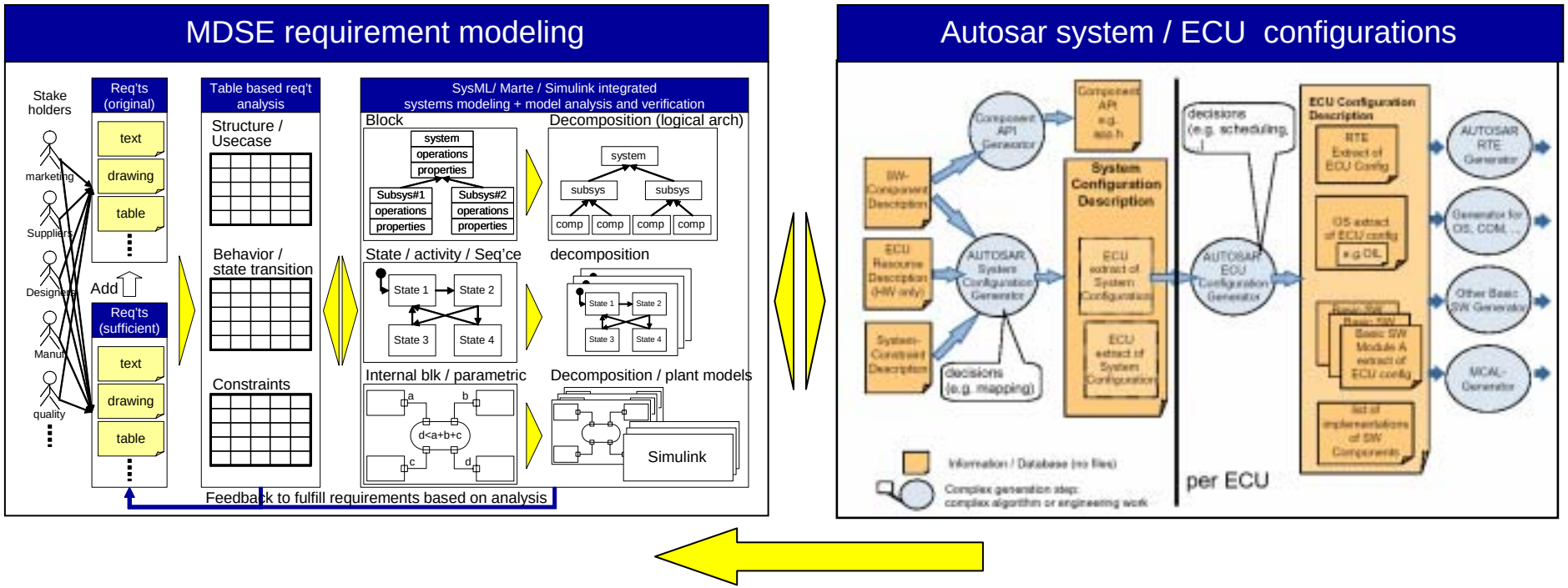
SysML™・Marte™で書かれたコントローラーと、Simulink™で書かれたプラントの両モデルを連携して、SysML™で表現されるパラメータ制約や、Marte™で表現される時間制約を、分析・検証できる仕組みが必要となります



Autosar™プロセスとの組合せ

Autosar™プロセスと組み合わせることで、要求からAtomic SW Component、さらに各ECUへのインプリメンテーションを、一連のプロセスの中で、Traceableに行うことが出来ます

Map to physical ECU / network configurations



Re-verify constraints after physical mapping

CONTENTS

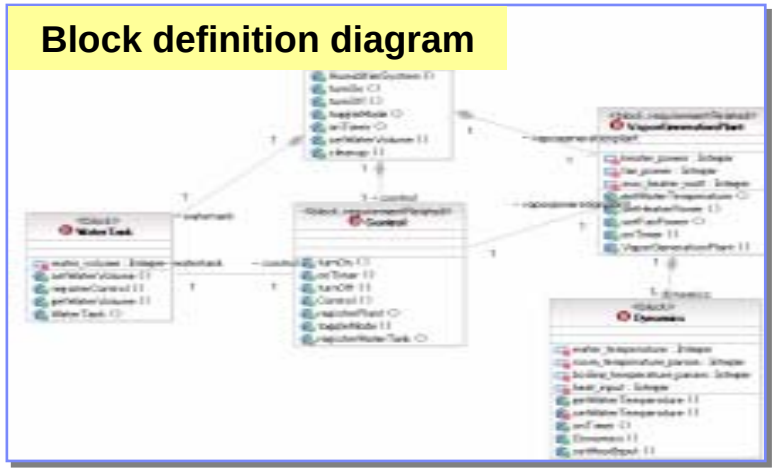
- メカ・エレキ・ソフトの統合設計環境が必要となる背景
- メカ・エレキ・ソフトのモデリング
  - Model-driven systems engineering (MDSE)の概要
  - SysML™・Marte™の活用
  - コントローラー・プラント統合モデリング
  - トレーサビリティーとインパクト分析
  - モデルの実行と検証
- Eclipse™ベースでの統合設計環境
- まとめ

SysML™の活用

SysML™では、(1) 構造と (2) 振舞いに加えて、(3) 要求と設計の連携と (4) パラメータ制約をひとつの言語で記述できます

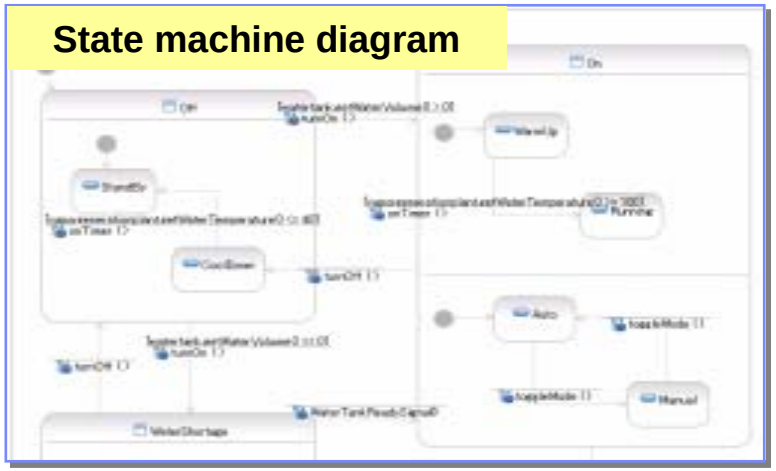
(1) 構造

Block definition diagram



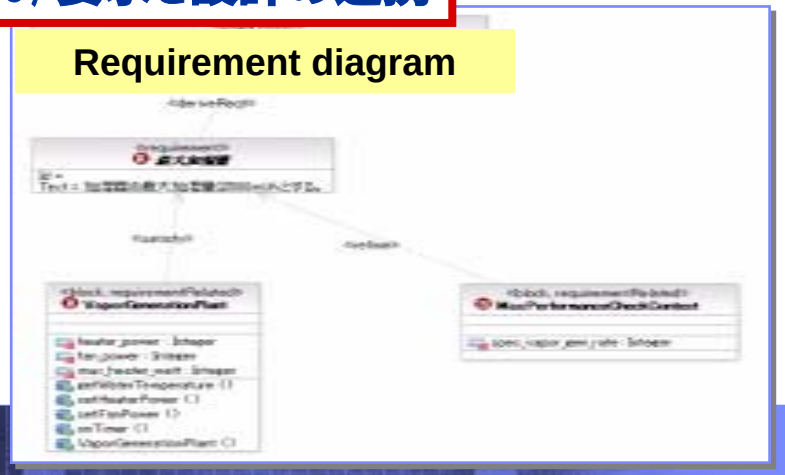
(2) 振舞い

State machine diagram



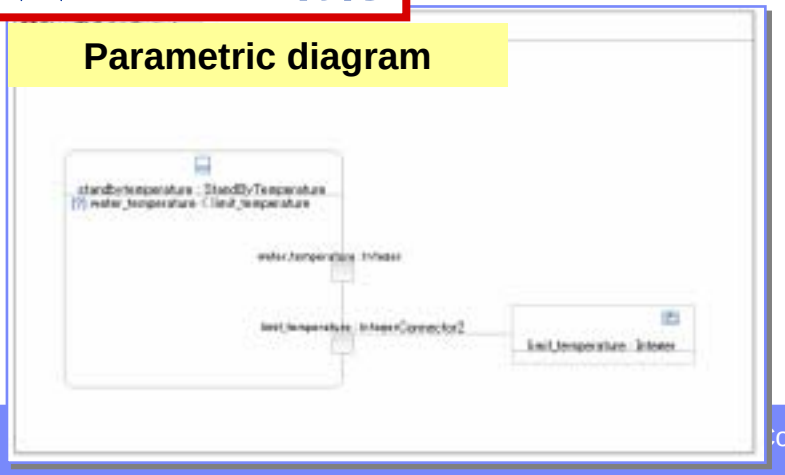
(3) 要求と設計の連携

Requirement diagram



(4) パラメータ制約

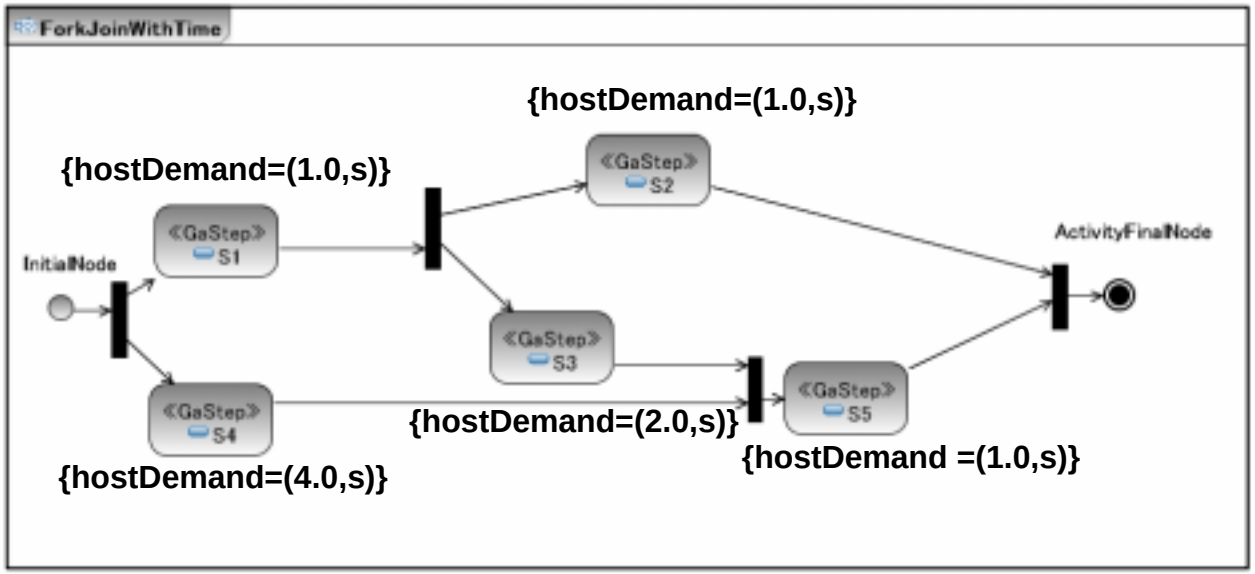
Parametric diagram



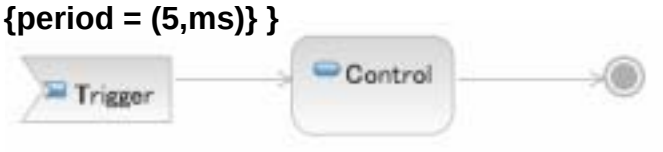
MARTE™によるTimed検証のためのモデリング

動作に関する時間を指定するMARTE™プロファイルを利用した時間指定の具体例です

- 所要時間の指定



- 周期的タスクの周波数の指定



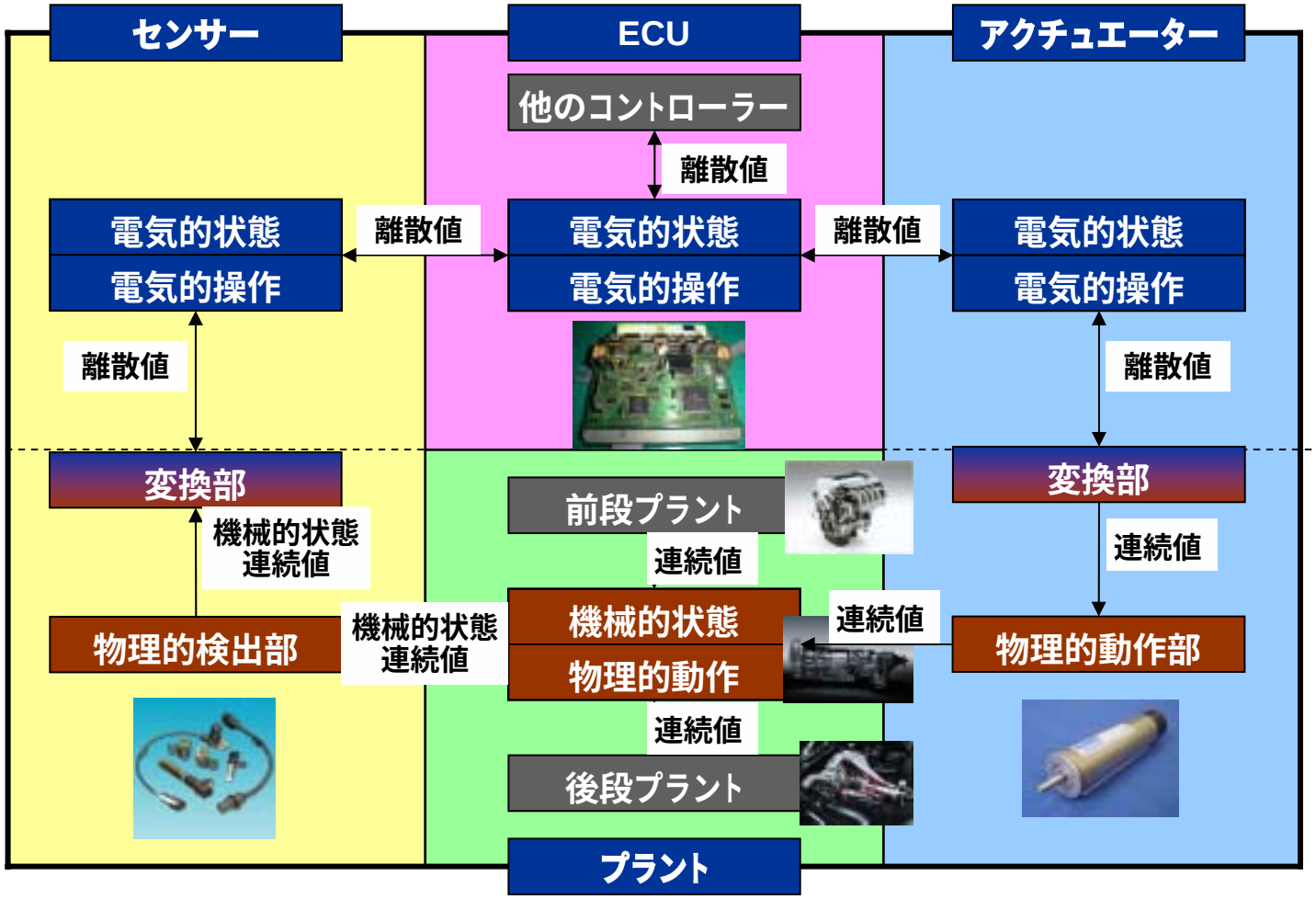
CONTENTS

制御視点でのモデリング

コントローラー・プラントモデリングにおいて、制御視点でモデリング要素を考える場合、離散系と連続系に分け、それぞれを異なった環境でモデリングしていくことになります

離散系
(SysML™/UML™ **でのモデリング**)

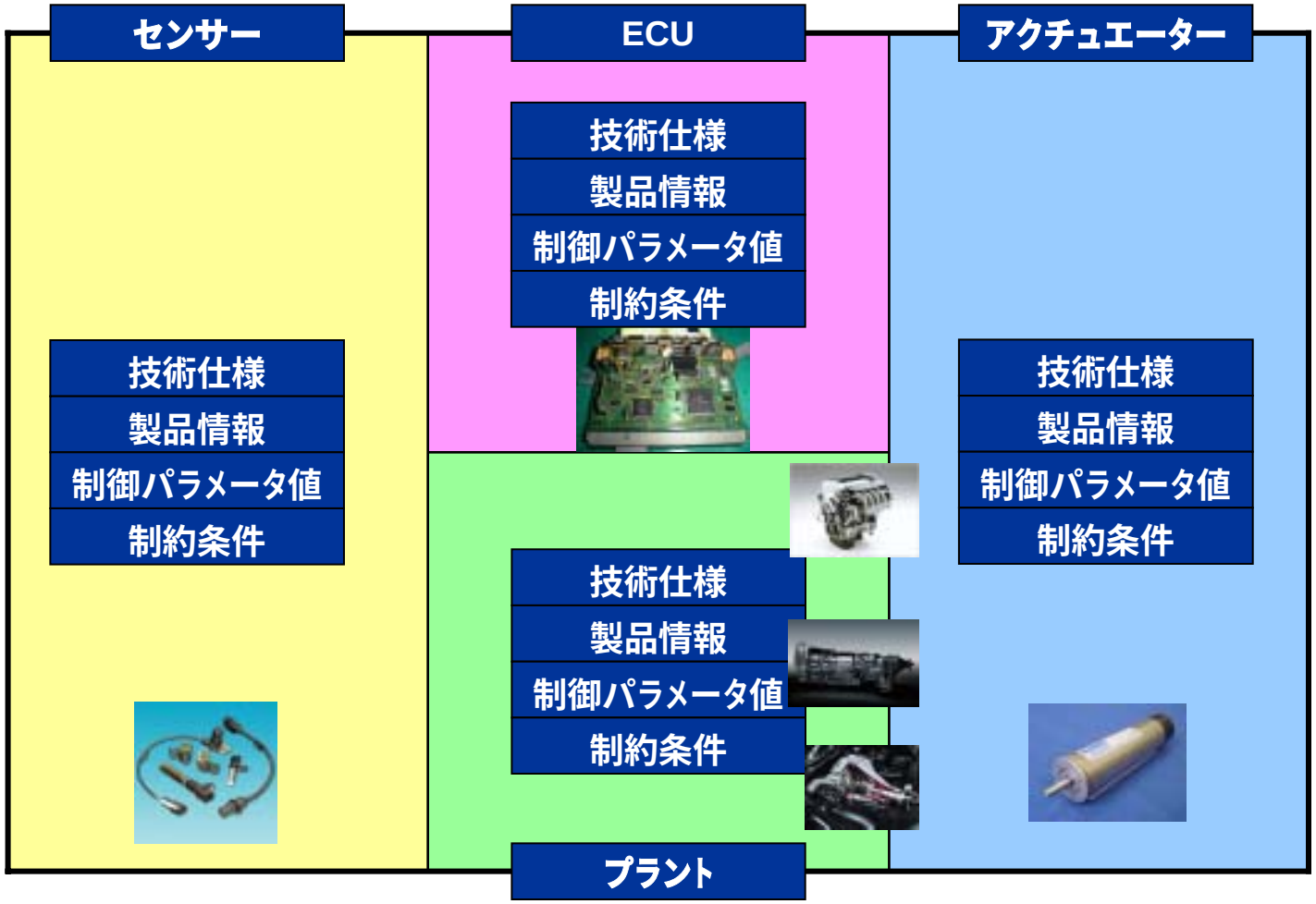
連続系
(Simulationでの**モデリング**)



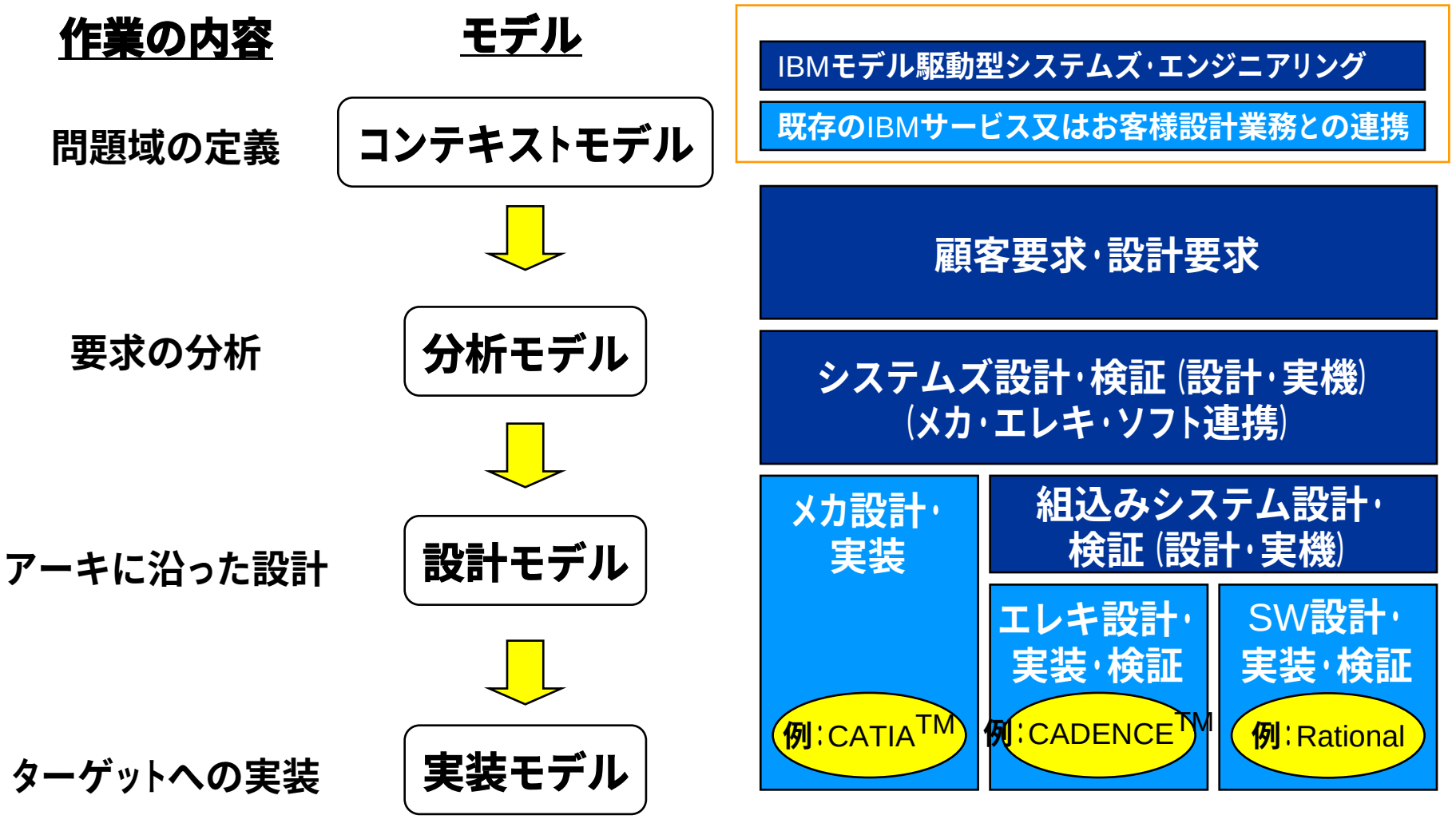
パラメトリック制約視点でのモデリング

技術仕様・製品情報・制御パラメータ値・制約条件など、システムズとしての設計制約の評価のために必要なブロックのプロパティー値は、SysML™モデルに記述します

コントローラー・プラントを含めて、SysML™、UML™でモデリング



モデル駆動型システムズ開発は、設計の精度を段階的に高めていくように、設計の完成度を4つの段階で進めていきます

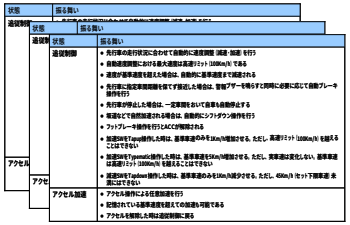


論理アーキテクチャーを活用したコントローラ・プラントモデリングの例

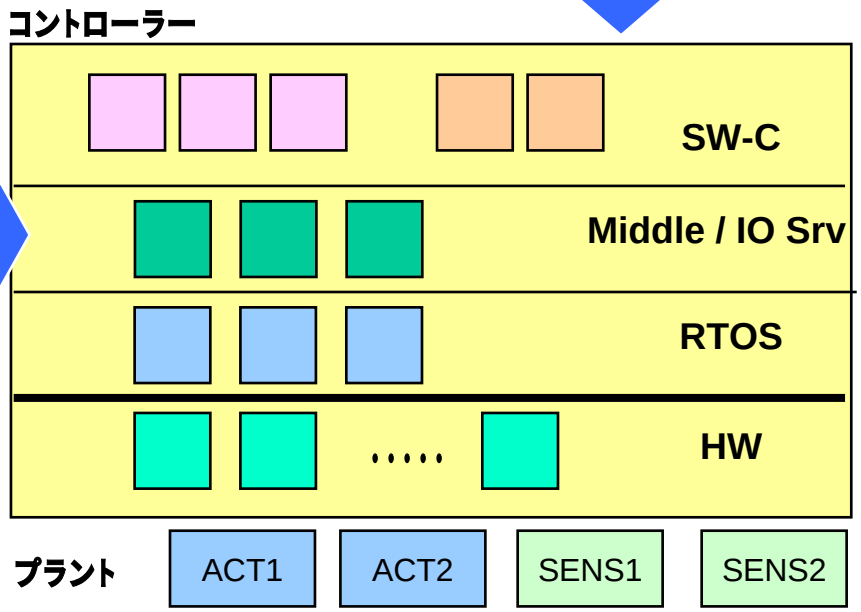
論理アーキテクチャー・リファレンスと仕様検討表を参照しながら、状態の振舞を実現するコントローラ・プラントに対応する論理コンポーネントを設計します

アーキテクチャー構成要素	アーキテクチャー構成要素の説明
コントローラー機能	• I/Oはデジタル信号で、ECUに割り当てられるべき機能
ネットワーク機能	• I/Oはデジタル信号で、ネットワークに割り当てられるべき機能
スイッチその1	• ステートを持たず、イベントのみを発生させるスイッチ
スイッチその2	• ステートを持ち、さらにイベントも発生させるスイッチ
I/O機能その1	• INPUTはデジタル信号で、OUTPUTはアナログ信号のプラント • 主にアクチュエーター
I/O機能その2	• INPUTはアナログ信号で、OUTPUTはデジタル信号のプラント • 主にセンサー
プラント機能その1	• INPUTもOUTPUTはアナログ信号で、運動モデルだけの制御 • エンジンなど
プラント機能その2	• INPUTもOUTPUTはアナログ信号で、運動モデルと状態を持つ • トランスミッションなど

コントローラー系はAutosar™へマッピングする



仕様検討表

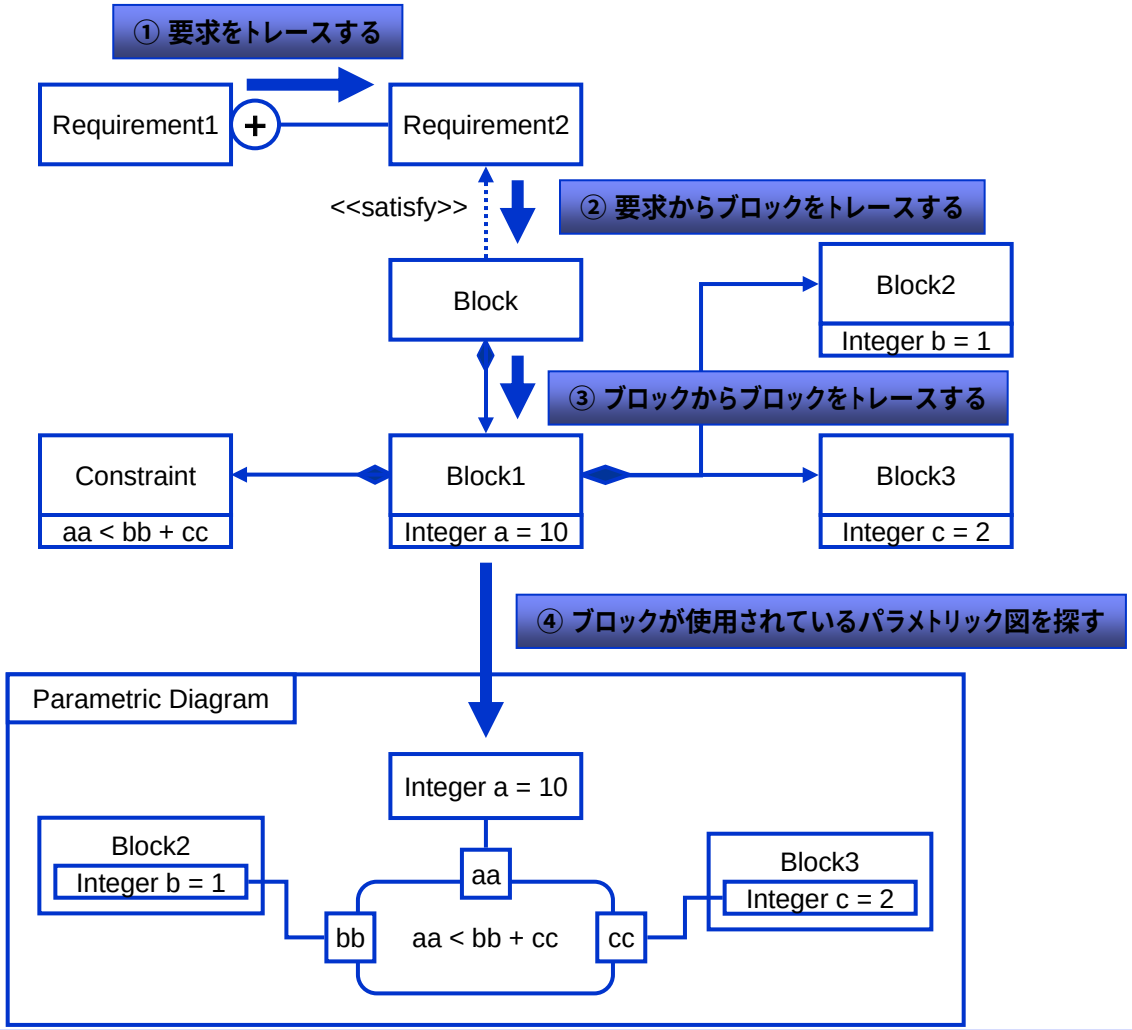


CONTENTS

- メカ・エレキ・ソフトの統合設計環境が必要となる背景
- メカ・エレキ・ソフトのモデリング
  - Model-driven systems engineering (MDSE)の概要
  - SysML™・Marte™の活用
  - コントローラー・プラント統合モデリング
  - トレーサビリティーとインパクト分析
  - モデルの実行と検証
- Eclipse™ベースでの統合設計環境
- まとめ

要求を反映する制約ブロックの特定

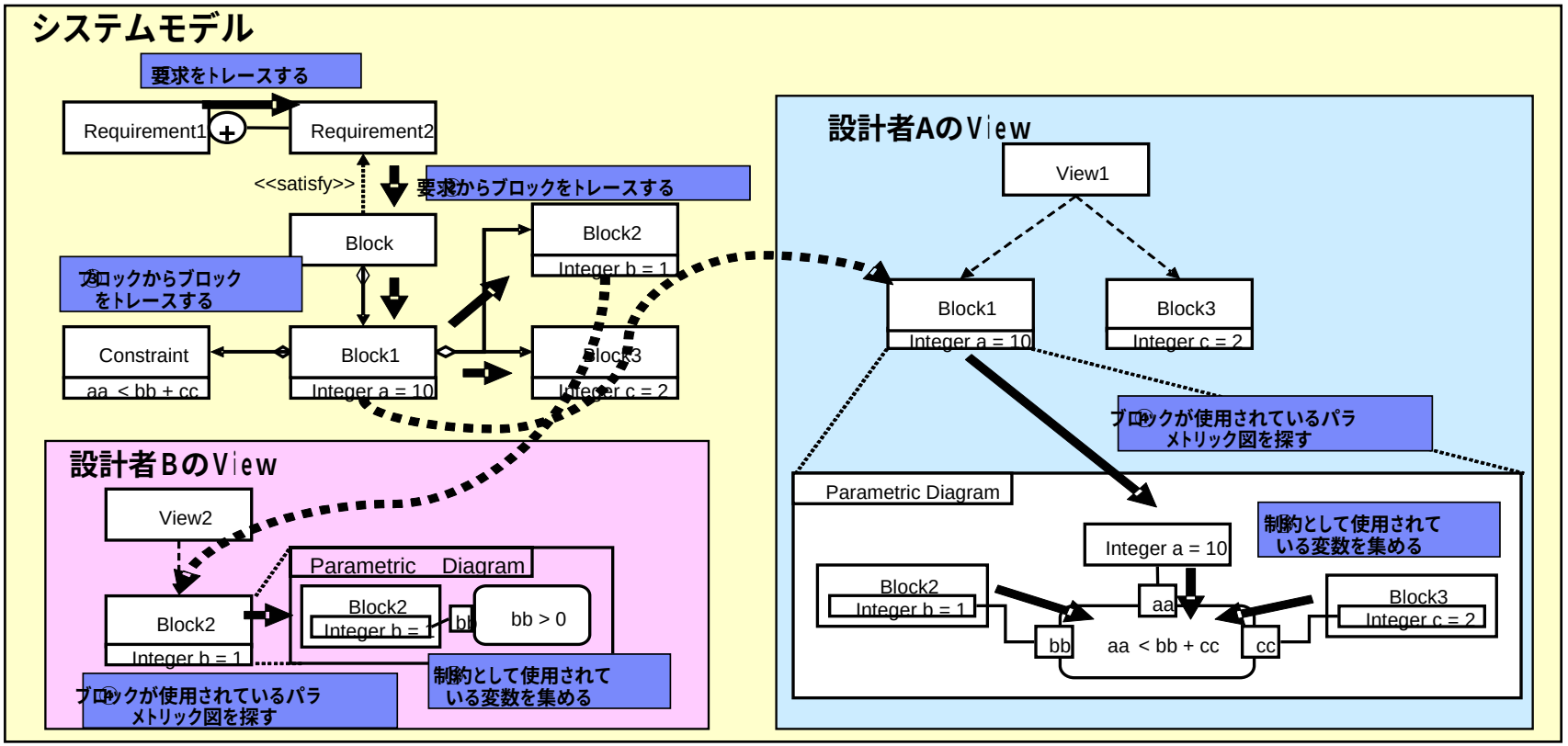
ある要求を起点として、関連付けられた要求やブロックをたどることにより、要求を反映する制約ブロックを特定します



どの関連をたどるのかは、トレースポリシーに従います

設計者間の不整合の検出

コンカレント設計プロセスにおいて、複数の設計者による変更間の整合性を、パラメトリック図に定義されている制約とモデル間の関連から、確認することが出来ます

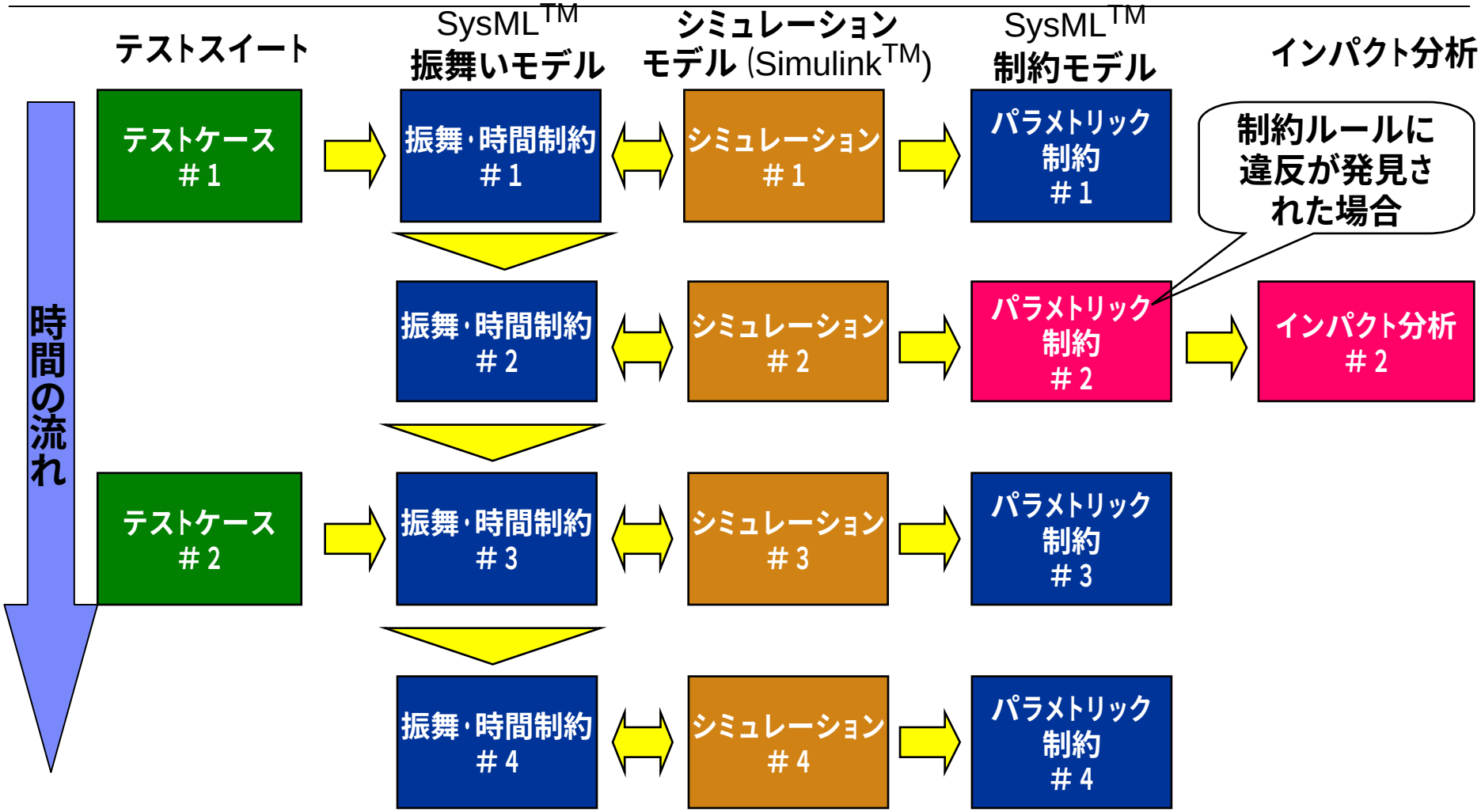


CONTENTS

- メカ・エレキ・ソフトの統合設計環境が必要となる背景
- メカ・エレキ・ソフトのモデリング
  - Model-driven systems engineering (MDSE)の概要
  - SysML™・Marte™の活用
  - コントローラー・プラント統合モデリング
  - トレーサビリティーとインパクト分析
  - モデルの実行と検証
- Eclipse™ベースでの統合設計環境
- まとめ

動的検証の流れ

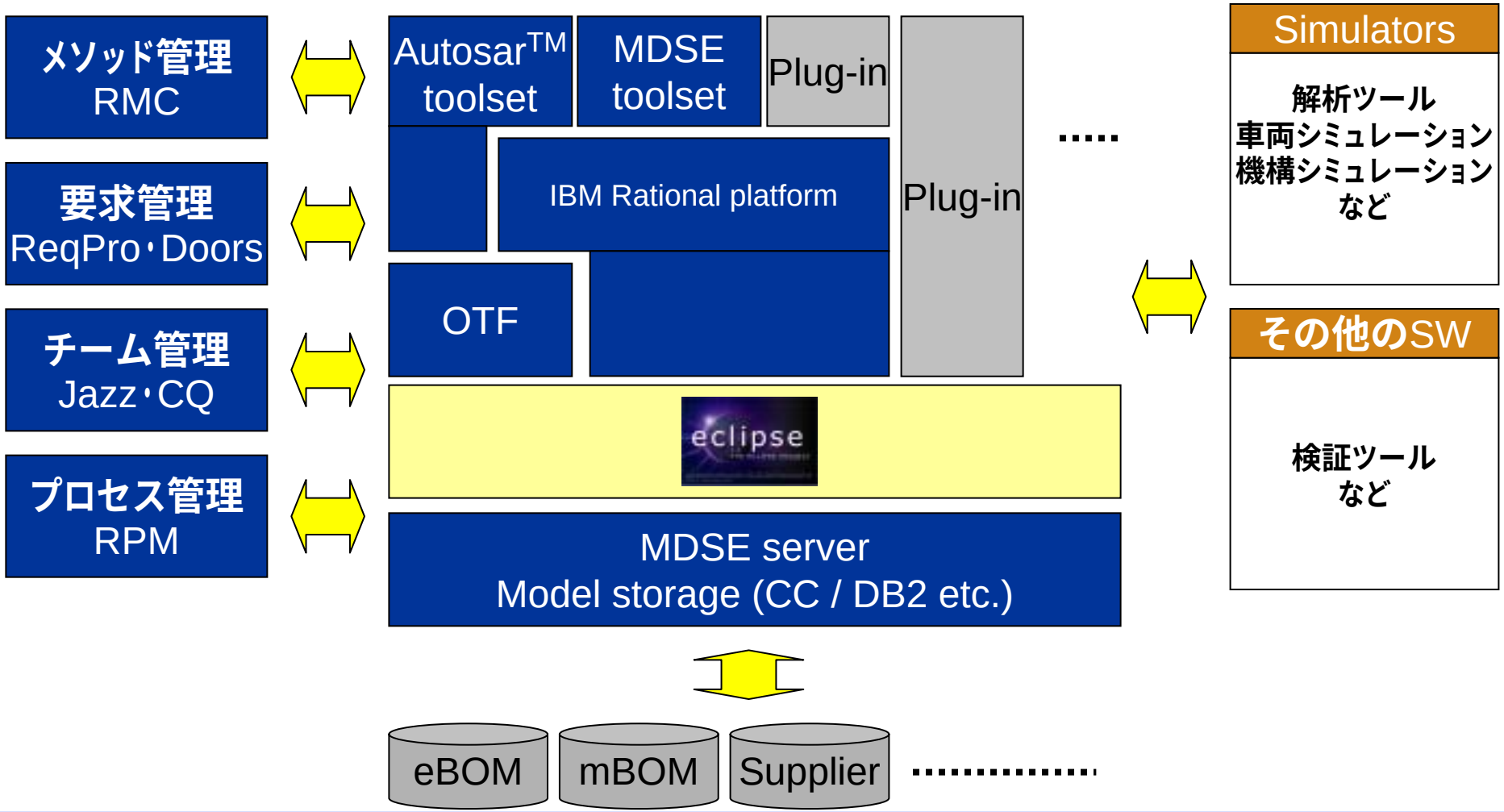
テストケースを入力にして、システムズの振舞いを、外部シミュレーションと連携しながら動的にシミュレーションし、各振舞いステップにおける時間およびパラメトリックな制約を検証して、制約ルールに反している場合は、関連のモデル要素と要求項目を分析・表示します



CONTENTS

- メカ・エレキ・ソフトの統合設計環境が必要となる背景
- メカ・エレキ・ソフトのモデリング
  - Model-driven systems engineering (MDSE)の概要
  - SysML™・Marte™の活用
  - コントローラー・プラント統合モデリング
  - トレーサビリティーとインパクト分析
  - モデルの実行と検証
- Eclipse™ベースでの統合設計環境
- まとめ

Open Tool Framework (OTF)とEclipse™上での、システムズの統合設計環境の開発を行っています



CONTENTS

- メカ・エレキ・ソフトの統合設計環境が必要となる背景
- メカ・エレキ・ソフトのモデリング
  - Model-driven systems engineering (MDSE)の概要
  - SysML™・Marte™の活用
  - コントローラー・プラント統合モデリング
  - トレーサビリティーとインパクト分析
  - モデルの実行と検証
- Eclipse™ベースでの統合設計環境
- **まとめ**

まとめ

- Model-Driven Systems Engineering (MDSE)とは、システムズに対する要求を、モデルを用いて分析・整理・検証することで、要求を満足し、迅速に品質の高い設計を実現する手段です
- SysML™を用いて、(1) 構造と (2) 振舞いに加えて、(3) 要求と設計の連携と (4) パラメータ制約をひとつの言語で記述できます
- MARTE™を用いて、時間制約などを記述できます
- コントローラー・プラントを統合してモデリングする手法が必要となってきます
- モデルリングとその実行・検証をサポートするツール群を、オープンな環境の上に開発しています
  - トレーサビリティー
  - 静的・動的検証
  - トレードオフ分析
  - など