

# MBSE支援サービスと事例のご紹介

- 現代の自動車開発において、車載制御システムの位置づけは極めて重要となっており、その開発を効率的に行うために、ソフトウェア開発の効率化手法であるMBDとともに、上流の**システム開発を体系的に行うための手法として、MBSEが欠かせない技術**となっています。
- 近年、**多くの企業でMBSEの導入・活用が進められており**、先進的な企業様では「要求の体系化」「網羅的な振る舞い検討」「最適なテストケース生成」など、すでに**大きな効果が得られています**。
- しかしながら、「**MBSEに興味はあるがまだ導入していない」「どこから始めればよいか分からない」「導入を試みたがうまく開発に展開できていない**」という企業様もまだまだ多く存在しています。
- SOLIZEでは、**これからMBSEの導入を始めるお客さま向けに**、代表的なモデリング言語である**SysMLの「基礎教育」と「実践支援」**のサービスを提供しています。
- さらに、MBSEを活用して業務課題を解決する **「MBSEコンサルティング」** サービスも提供しています。

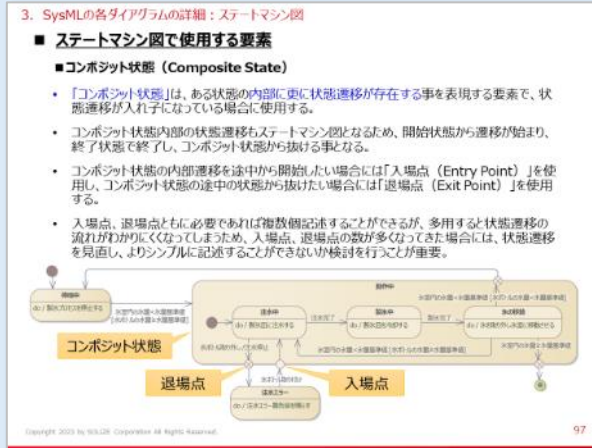


## MBSEの代表的なモデリング言語であるSysMLの基礎教育から実践までをご支援

### ① 基礎教育

#### SysML各種ダイアグラムの 読解/作成ができる技術の習得

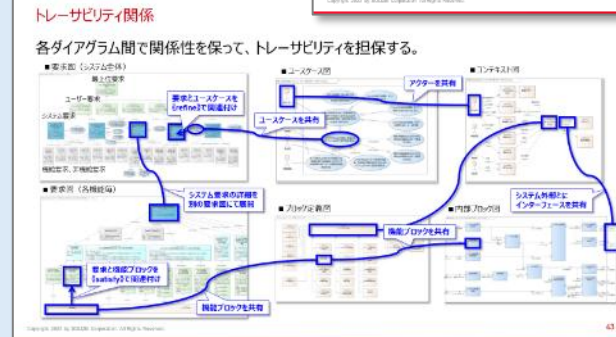
- 1 代表的なSysMLダイアグラムの種類と特長、使用ルールなど、**基礎をしっかりと習得**できる。
- 2 SysML作成ツールを利用した実践形式なので、**自分でダイアグラムが作成**できるようになる。
- 3 クルーズコントロールを題材にした応用問題により、**実業務に必要な考える力**を習得できる。



### ② 実践支援

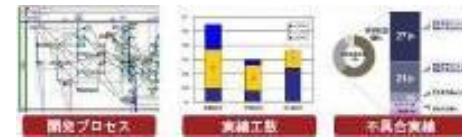
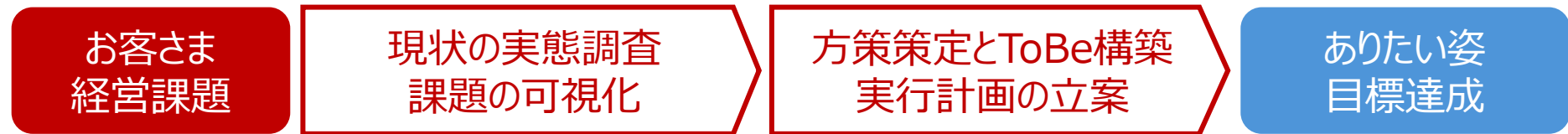
#### お客さまの製品のSysMLモデルを作成し 実業務に適用するためのノウハウを習得

#### ワークショップ形式でディスカッション



開発業務の実態把握・詳細分析から解決すべき課題を導き出し、  
ありたい姿に向けてお客さまの業務の変革を実現

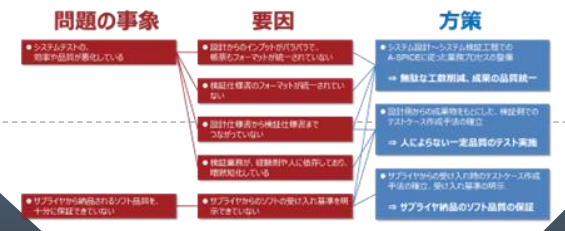
## 【To Be】



お客様の課題  
仕入れ先とのQA削減、開発スピード・品質を担保したい  
MBSE適用の効果見込み  
仕様書の初回QAを 77% 削減



対話型ヒアリング  
暗黙知のあぶり出し



蓄積データ分析  
事実の認識



関係者間での  
コミットメント調整

## 【As Is】

開発の実態

## お客さまの課題の解決に向け、適用効果の高いMBSE手法の導入を支援

特長① 要求の階層化

特長② 用語・構文のルール化

特長③ 利害関係者要求からの提供サービス定義

特長④ シーンに応じた車両の振る舞い定義

特長⑤ 入力条件に応じたシステム応答定義、  
車両要件抽出

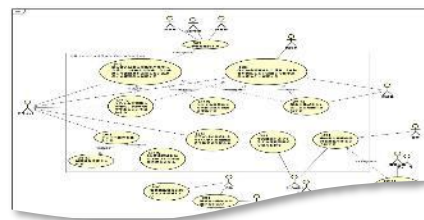
特長⑥ 実現方法の検討とサブシステムへの機能配置

特長⑦ DSMを用いた要求間の依存関係の管理

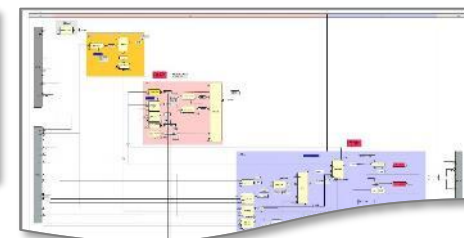
特長⑧ 機能成立性検証、非機能要件の抽出

特長⑨ 管理性の高いUSDM要求仕様書の作成

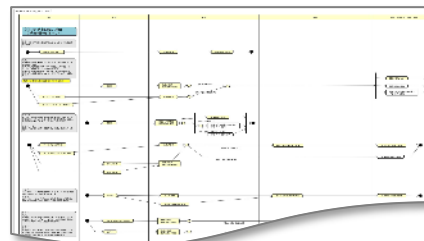
【ユースケース図】システムのサービス



【内部ブロック図】シグナルフロー



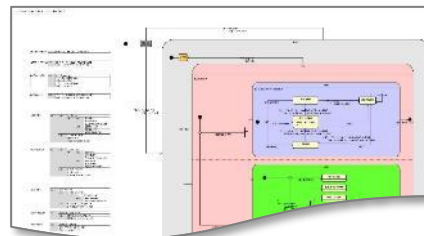
【アクティビティ図】ユースケースパターン



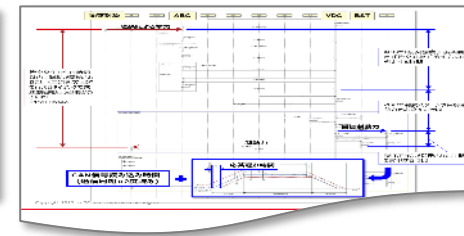
【DSM】サブシステムへの機能配置



【ステートマシン図】システム応答

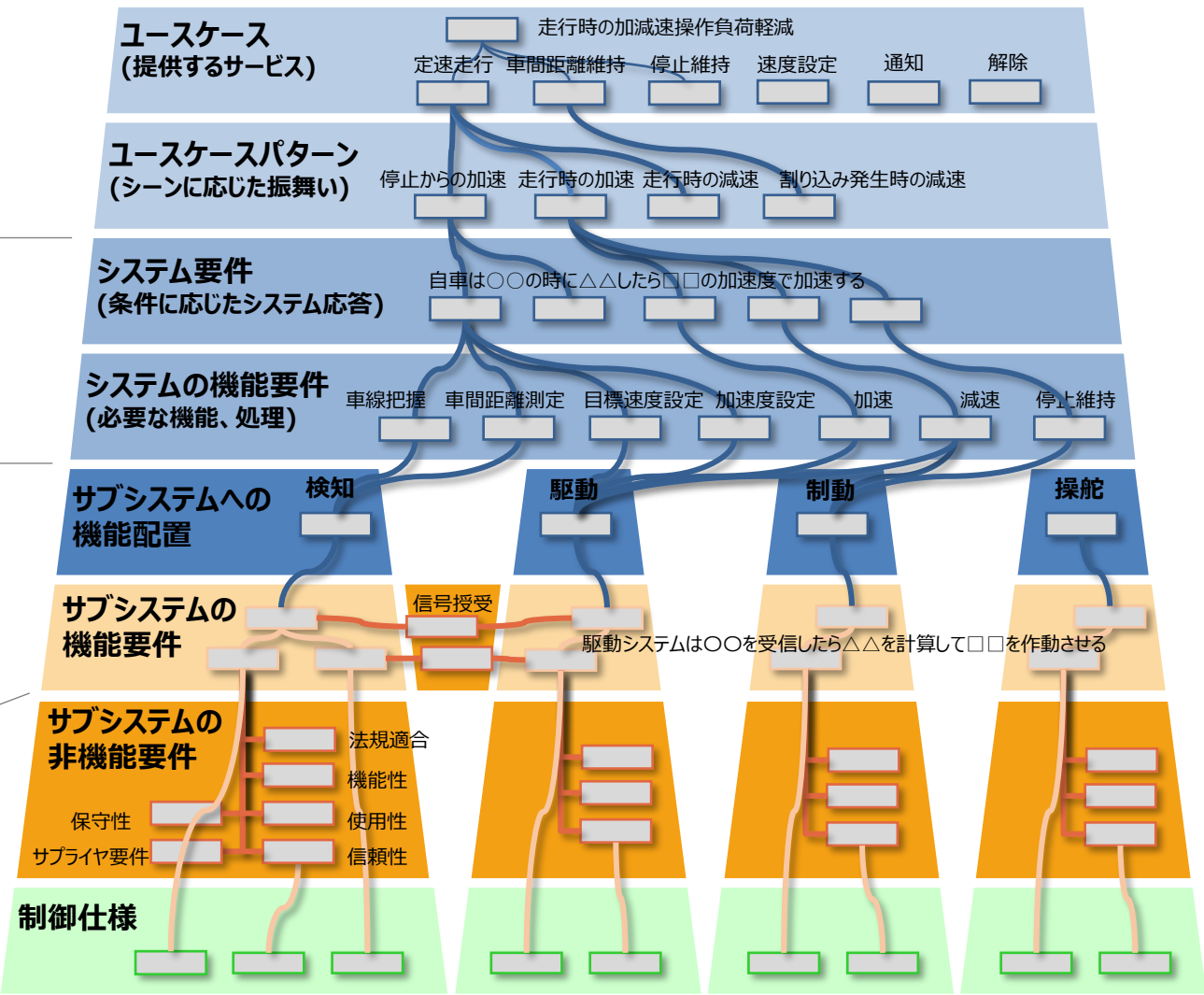
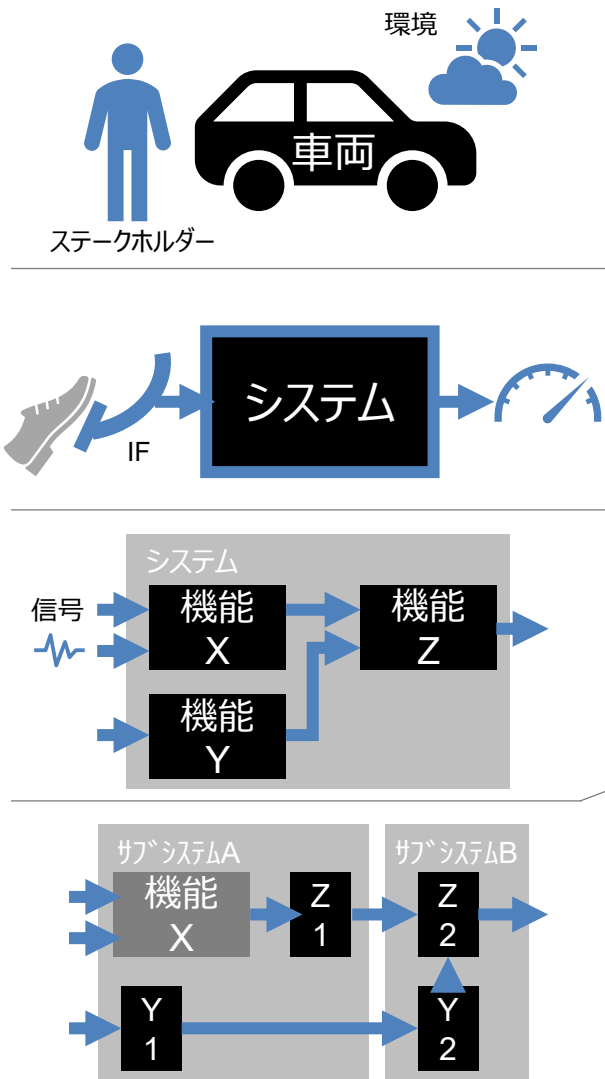


【シーケンス図】機能成立性の検証



特長① 要求の階層化

誰が行っても一定の「規則性」と「網羅性」を確保できるよう、階層化した要求抽出を行う





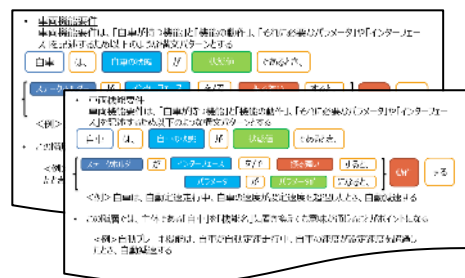
用語・構文のルール化を実施し、要求記載粒度を統一化

# 用語定義書

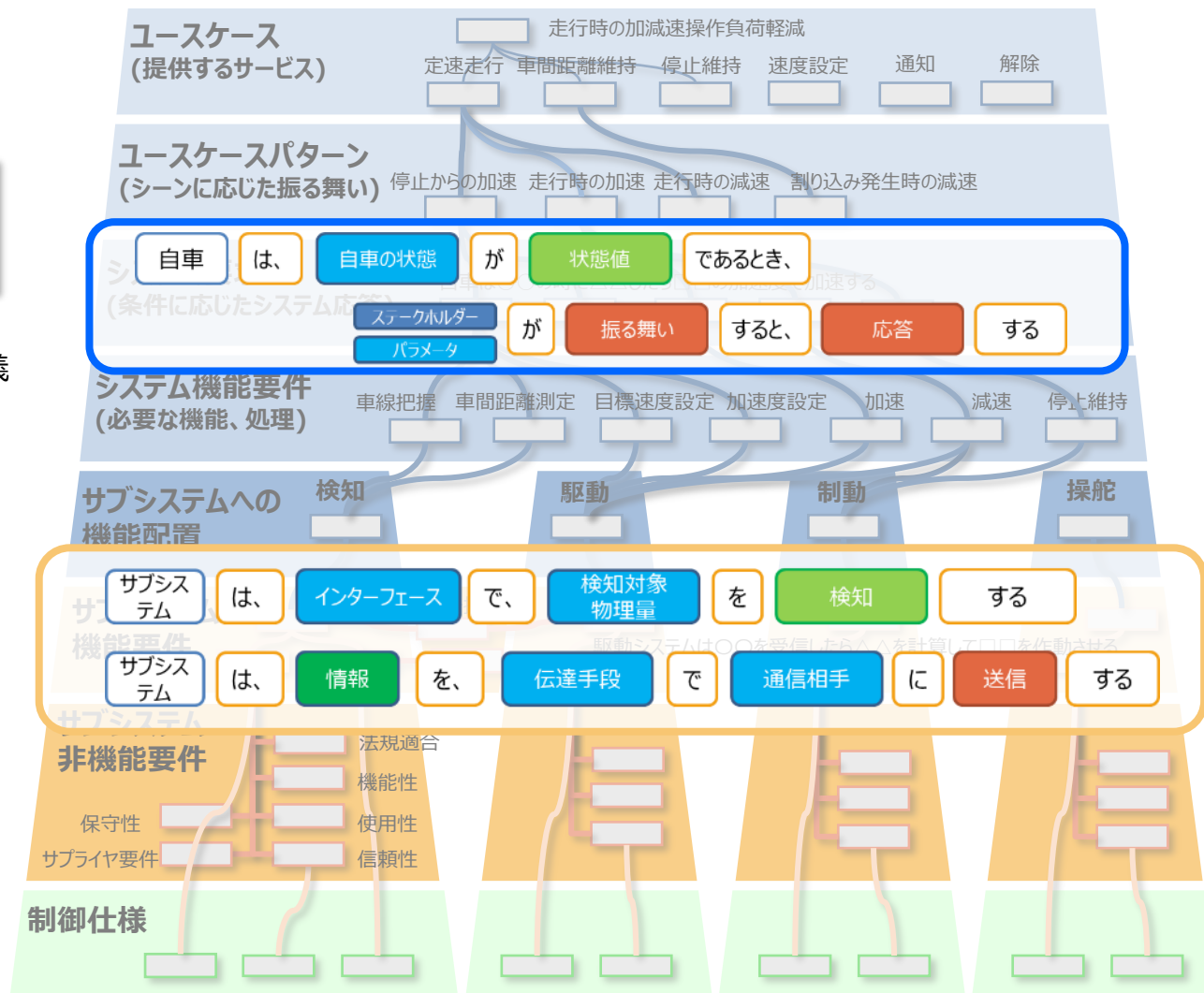
[illegible]

## 使用可能な用語・用語の関係性を定義

## 書義定文構



## 要求階層ごとの構文を定義



# 実施事例① 冗長電源制御システム開発プロセスの变革

トヨタ自動車株式会社



## 増大する要求に対応可能なソフトウェア構造実現に向けた設計変革機運の高まり

トヨタ自動車株式会社は、「幸せの量産」を使命とし、クルマの未来を変えていくために、モビリティ・カンパニーへの変革を目指しています。「カーボンニュートラル」と「移動価値の拡張」という二つの大きなテーマを掲げ、これまで培ってきたクルマの本質的な価値を基盤にもっと社会の役に立つ存在へクルマを進化させ、誰もが自由に、楽しく、快適に移動できるモビリティ社会を実現することを目指し、挑戦を続けています。

同社制御電子プラットフォーム開発部では、ソフトウェア・ハードウェアアップデートを見据えたクルマ全体の電気・電子システムの機能配置を考え、ソフトウェアファーストなクルマを実現する電子プラットフォームの構築を担っています。これからのクルマ開発において、多様な機能を協調して制御する技術の開発領域はさらに拡大していくとみられており、さまざまな要求や関係する機能の増大に伴い、ますます複雑化していくことが予想されています。その流れは電子プラットフォームの構築においても例外ではなく、増大する要求に対応可能なソフトウェア構造とそれを実現する設計スタイルの確立が急務となっていました。

## 増え続ける要求への対応と品質の確保が両立しきれない潜在リスクへの打ち手

従来、電源開発においては、市販実績のあるソースコードに追加、編集していくことで、品質と短納期開発の両立を実現してきました。一方で、ソフトウェアの大規模化、複雑化の進展はとどまるところを知らず、期日までに検証をやりきるためには、人と時間をかけ、力業で品質を確保せざるを得ない状況が多発するようになってきていました。

現時点でお客さまに提供している製品の品質に問題は発生してはなくても、今後これまで以上に複雑化していくソフトウェア構造に対して品質を確保し続けるためには、抜本的な変革が必要な状況であることは明白でした。

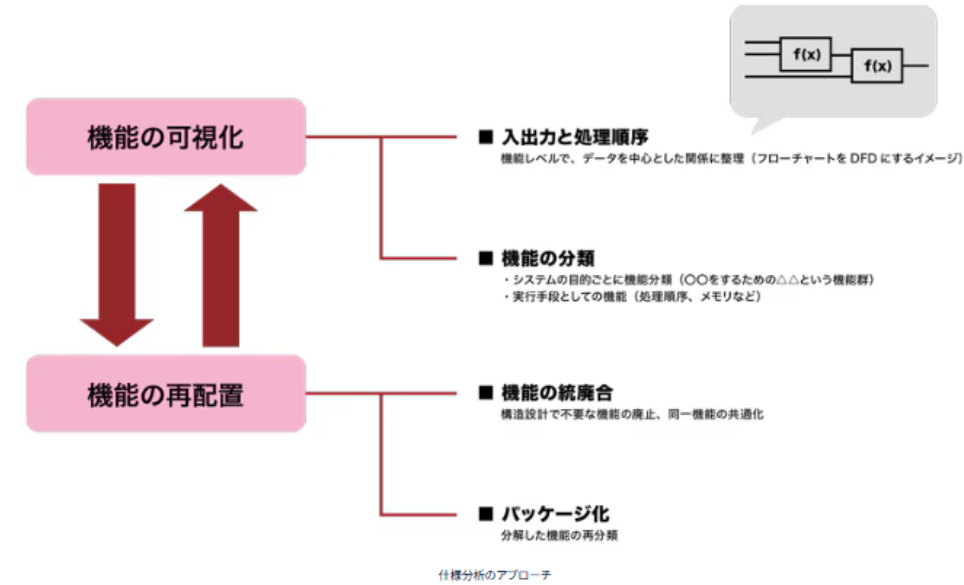
そのような背景から、機能配置の最適化、要求分析を起点とするトレーサビリティの確保、モデルベース開発プロセスのさらなる推進といった取り組みが、品質・生産性の向上やリードタイムの短縮につながることを、またその取り組み自体が技術の手の内化につながることを想定したうえで、3〜4年先を見据えた抜本的な変革活動がスタートしました。

👉 詳細はこちらから閲覧いただけます

トヨタ自動車株式会社 - 構造設計にフォーカスした、  
冗長電源制御システム開発プロセスの变革  
<https://ureka-technology.solize.com/projects/006>

## アーキテクチャ設計にフォーカスした、先行検証型の開発スタイルを実現する取り組み

当社は、対象とする冗長電源制御システムの制御仕様やソースコードを徹底的に分析したうえで、複雑化したシステムを独自の手法で紐解いていきました。その分析内容から、機能自体の定義やその影響範囲、要件の重複などが明確になったことで、機能配置の見直しや機能の統廃合を両社協働で進めることができました。結果として、機能は約2/3に集約され、見通しのよいアーキテクチャとして再構築することを可能としました。





# 実施事例② Automotive SPICEをベースとした検証プロセス変革

日産自動車株式会社



## イノベーションをドライブし続けるために、開発基盤の強靱化を図る

日産自動車株式会社は、「人々の生活を豊かに。イノベーションをドライブし続ける。」をコーポレートパーパスとして掲げ、創業以来培ってきた技術と、「他のやらぬことを、やる」という精神のもと、人々の移動の可能性や社会の可能性を広げるために、日々挑戦を続けています。

日産自動車のAD/ADAS&シャシー制御開発部では、クルマ全体としてドライバーが意図したとおりの走行を実現するために、ブレーキやパワートレイン等を統合的に制御するシステムの設計および検証業務を担っています。昨今これらのシステムには、よりきめ細かな制御だけではなく、たとえばドライバーの好みに応じた設定変更などのパーソナライゼーションも求められ、多機能化、複雑化の一途をたどっています。このようなシステムの開発に当たり、これまでは設計と検証が一丸となって品質を担保してきましたが、これから先は、より高い次元での品質確保と効率向上を実現する基盤の確立が急務となっていました。

## ベテランの経験とノウハウを若手技術者に継承し、将来的な競争力の強化を目指す

従来、日産自動車のシャシー制御システム開発の現場では、制御ロジックの設計者が検証内容を策定し、検証者が各検証環境に合わせた手順の作成および実施を行っている状況でした。その結果、第三者視点で検証内容が策定できないことに加え、システム全体で行うべき検証が、ソフトウェアのロジックに着目した検証に偏る傾向がありました。このため、ベテラン技術者が過去の経験やノウハウをもとに、第三者視点やシステムレベルでの検証項目を追加することで最終的な品質を確保してきましたが、ベテラン技術者の工数が限られる中で多機能化したシステムの品質を確保し続けるのは現実的に困難な状況が見えてきており、抜本的な変革の必要性が高まっていました。

このような背景から、若手技術者でもシステム検証を実施できるようにし、システムの品質を確保し続けること、さらには、若手技術者に業務を引き継ぐことで生まれるベテラン技術者の工数を新たなモビリティ開発に充てることで、将来的な競争力の強化を目指した変革活動がスタートしました。

👉 詳細はこちらから閲覧いただけます

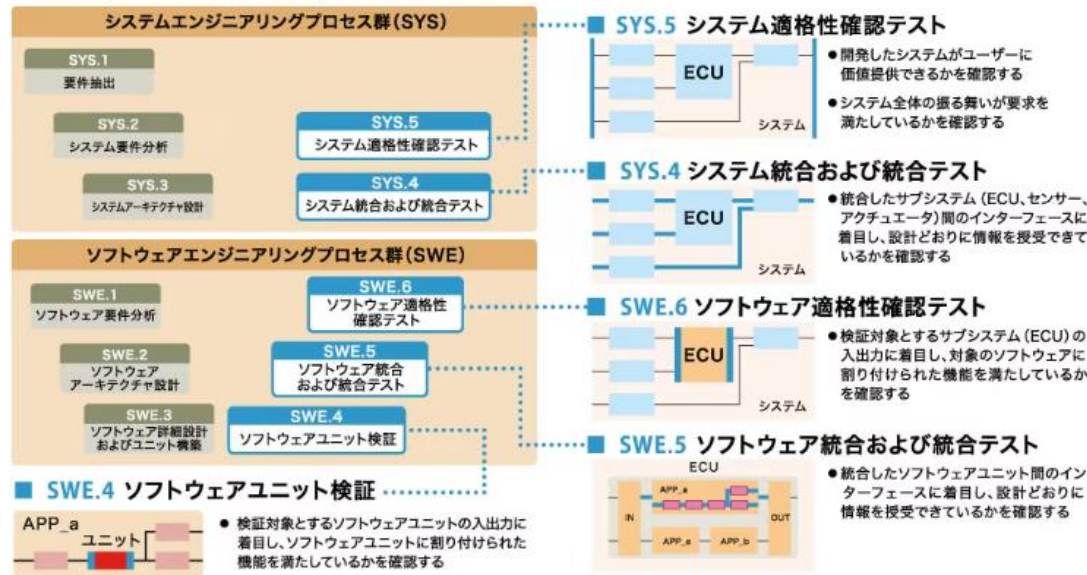
日産自動車株式会社 - Automotive SPICEをベースとした、シャシー制御システムの検証プロセス変革

<https://ureka-technology.solize.com/projects/022>

## Automotive SPICEをベースとした検証業務標準化の取り組み

当社は、Automotive SPICE (以下A-SPICE) をベースに、車載システム開発で求められる検証業務の階層を定義することから始めました。たとえば、A-SPICEの「SYS.5 システム適格性確認テスト」と「SYS.4 システム統合および統合テスト」のそれぞれにおいて、入力および測定の対象となる要素を洗い出し、業務の階層間で重複することがないように当てを行いました (SWE領域についても同様)。

### ■ PRM (プロセス参照モデル)



# SysML基礎教育

目的	SysMLダイアグラムの一般知識の習得
期間	・ 4時間×4日間
効果	・ 代表的なSysMLダイアグラムの種類と特長、使い方の概要を習得する ・ 各種ダイアグラムの読解、作成が行えるようにする
内容	■ Day1 ・ SysMLとモデリングの概要、SysMLのダイアグラムの概要 ・ 要求図（Requirement Diagram） ■ Day2 ・ ユースケース図（Usecase Diagram） ・ アクティビティ図（Activity Diagram） ■ Day3 ・ ステートマシン図（State Machine Diagram） ・ ブロック定義図（Block Definition Diagram） ■ Day4 ・ 内部ブロック図（Internal Block Diagram） ・ シーケンス図（Sequence Diagram）
提出物	・ 講義資料 ・ 受講者の提出物に対するフィードバック結果
使用ツール	・ Enterprise Architect、astah*SysML、（その他のツールも承ります）



## 1 代表的なSysMLダイアグラムの種類と特長、使用ルールなど、**基礎をしっかりと習得**できる

**SysMLとは**

**システムをモデリングするための記述言語**

システムの仕様化、分析、設計、妥当性確認や検証のために利用できるため、自動車・航空宇宙・通信分野などで幅広く活用されている

**SysMLの言語仕様**

ソフトウェア開発で用いられている「UML」を再利用した図と、新たに拡張した図から構成される

**SysMLのダイアグラム（＝図）の構成**

「構造図」「振る舞い図」「要求図」に分類され、お互いに関連している。

赤字はUMLに対して新規または変更があった図

※ OMG Unified Modeling Language (UML) Object Management Group (OMG) Copyright 2023 by SOLIZE Corporation All Rights Reserved.

**SysMLの効果**

システムを多面的かつ抽象的に表現することで全体最適を実現する

システム構想資料

システムに求められるサービスを実現する手段は、一つとは限らない

要求

振る舞い

構造

最適化実現手段（仕様）を導き出す

システム仕様

Copyright 2021 by SOLIZE Corporation All Rights Reserved.

**3. SysMLの各ダイアグラムの詳細：ステートマシン図**

■ **ステートマシン図（State Machine Diagram）**

- 「ステートマシン図」は、ある要素が、生成されてから破棄されるまでの様子の変化（ライフサイクルを表現するためのダイアグラム）。
- 対象要素が取り得る様子や、様子に変化するきっかけを明確化するとともに、各場面で実行する振る舞い（アクティビティ）を表現することができる。
- 後述のシーケンス図では、システムを構成する複数の要素間のやり取り（相互作用）を表現するが、ステートマシン図では、その中のある一つの要素に注目して振る舞いを明確化する。

Copyright 2023 by SOLIZE Corporation All Rights Reserved.

**3. SysMLの各ダイアグラムの詳細：ステートマシン図**

■ **ステートマシン図で使用する要素**

■ **コンポジット状態（Composite State）**

- 「コンポジット状態」は、ある状態の内部に更に状態遷移が存在する事を表現する要素で、状態遷移が入れ子になっている場合に使用する。
- コンポジット状態内部の状態遷移もステートマシン図となるため、開始状態から遷移が始まり、終了状態で終了し、コンポジット状態から抜ける事となる。
- コンポジット状態の内部遷移を途中から開始したい場合には「入場点（Entry Point）」を使用し、コンポジット状態の途中の状態から抜けたい場合には「退場点（Exit Point）」を使用する。
- 入場点、退場点ともに必要であれば複数個記述することができるが、多用すると状態遷移の流れがわかりにくくなるため、入場点、退場点の数が多くなってきた場合には、状態遷移を見直し、よりシンプルに記述することができないか検討を行うことが重要。

Copyright 2023 by SOLIZE Corporation All Rights Reserved.

## 2 SysML作成ツールを利用した実践形式なので、自分でダイアグラムが作成できるようになる

3. SysMLの各ダイアグラムの詳細：要求図

■ 要求図の演習課題 ①

以下の要求図を模写してください。(15分)

Copyright 2023 by SOLIZE Corporation All Rights Reserved. 55

3. SysMLの各ダイアグラムの詳細：ユースケース図

■ ユースケース図の演習課題 ①-1

以下のユースケース図を模写してください。(10分)

Copyright 2023 by SOLIZE Corporation All Rights Reserved. 72

3. SysMLの各ダイアグラムの詳細：アクティビティ図

■ アクティビティ図の演習課題 ①

以下のアクティビティ図を模写してください。(15分)

Copyright 2023 by SOLIZE Corporation All Rights Reserved. 89

3. SysMLの各ダイアグラムの詳細：ステートマシン図

■ ステートマシン図の演習課題 ①

以下のステートマシン図を模写してください。(15分)

Copyright 2023 by SOLIZE Corporation All Rights Reserved. 105

3. SysMLの各ダイアグラムの詳細：内部ブロック図

■ 内部ブロック図の演習課題 ① (例1 フロープロパティでの表記)

ブロック定義図の演習課題①で作成した図を利用して、ポートの型をValue型で定義し、以下の内部ブロック図を模写してください。(30分)

Copyright 2023 by SOLIZE Corporation All Rights Reserved. 137

3. SysMLの各ダイアグラムの詳細：シーケンス図

■ シーケンス図の演習課題 ①

以下のシーケンス図を模写してください。(20分)

Copyright 2023 by SOLIZE Corporation All Rights Reserved. 159

## 3 クルーズコントロールを題材にした演習課題により、実業務に必要な考える力を習得できる

### 3. SysMLの各ダイアグラムの詳細：アクティビティ図

#### ■ アクティビティ図の演習課題 ②

「全車速追従機能付クルーズコントロール」機能のアクティビティ図を以下の情報を基に作成してください。(40分)

##### 作成するアクティビティ図

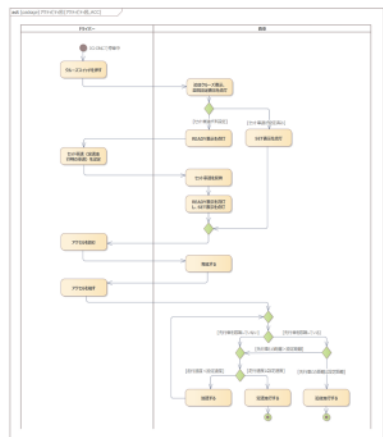
「車両停止状態でのACC開始から、条件分岐による定速走行と先行車追従走行まで」情報

1. パーティションは「ドライバー」と「車」とする
2. 開始ノードは「IG-ONにて停車中」とする
3. ドライバーが「クルーズスイッチを押す」と自車は「追従クルーズ表示と車間設定表示を点灯」させる
4. 自車は、セット車速（定速走行時の車速）が未設定の場合は「READY表示を点灯」させ、セット車速が設定済みの場合は「SET表示を点灯」させる
5. セット車速が未設定の場合に、ドライバーが「セット車速を設定」すると、自車は「セット車速を反映」したのち、「READY表示を消灯し、SET表示を点灯」させる
6. ドライバーが「アクセルを踏み」と、車両は「発進する」
7. 車両発進後に、ドライバーが「アクセルを離す」と、自車は分岐条件の組み合わせに応じて、「加速する」「定速走行する」「追従走行する」のいずれかのアクションを行う
8. 「先行車を認識している」かつ「先行車を認識していない」かつ「定速走行する」とし、「走行」
9. 「先行車を認識している」かつ「定速走行する」とし、「走行」
10. アクションが「加速する」の場合、アクティビティ図を終了

### 3. SysMLの各ダイアグラムの詳細：アクティビティ図

#### ■ アクティビティ図の演習課題 ② 模範回答

「全車速追従機能付クルーズコントロール」機能のアクティビティ図を以下の情報を基に作成してください。(40分)



Copyright 2023 by SOLIZE Corporation All Rights Reserved.

8

### 3. SysMLの各ダイアグラムの詳細：ステートマシン図

#### ■ ステートマシン図の演習課題 ②

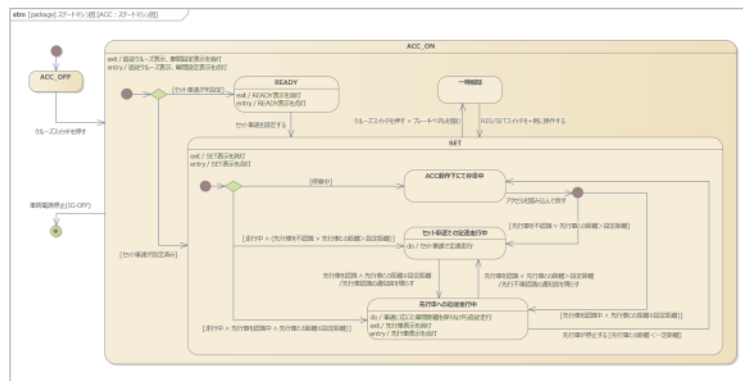
「全車速追従機能付クルーズコントロール」機能のステートマシン図を以下の情報を基に作成してください。(40分)

16. 「SET」状態に入った際に、「走行中 かつ（先行車を不認識 又は 先行車との距離>設定距離）」であれば、「セット車速での定速走行中」状態となる
17. 「SET」状態に入った際に、「走行中 かつ 先行車を認識中 かつ 先行車との距離≤設定距離」であれば、「先行車への追従走行中」状態となる
18. 「セット車速での定速走行中」状態の間は、「セット車速で定速走行」を行う
19. 「先行車への追従走行中」状態に入った際に、「先行車表示を点灯」させる
20. 「先行車への追従走行中」状態から出る際に、「先行車表示を消灯」させる
21. 「先行車への追従走行中」状態の間は、「車速に応じた車間距離を保ちながら追従走行」を行う
22. 「ACC作動下にて停車中」状態にて「アクセルを踏み込んで放す」を行った際、「先行車を不認識 又は 先行車との距離>設定距離」であれば、「セット車速での定速走行中」状態となる
23. 「ACC作動下にて停車中」状態にて「アクセルを踏み込んで放す」を行った際、「先行車を認識中 かつ 先行車との距離≤設定距離」であれば、「先行車への追従走行中」状態となる
24. 「セット車速での定速走行中」状態にて、「先行車を認識 かつ 先行車との距離≤設定距離」になったら、「先行車認識の通知音」を鳴らす。「先行車への追従走行中」状態となる
25. 「先行車への追従走行中」状態にて「先行車不認識の通知音」を鳴らす
26. 「先行車への追従走行中」状態にて「ACC作動下にて停車中」状態となる
27. 「ACC\_ON」状態にて、「車」

### 3. SysMLの各ダイアグラムの詳細：ステートマシン図

#### ■ ステートマシン図の演習課題 ② 模範回答

「全車速追従機能付クルーズコントロール」機能のステートマシン図を以下の情報を基に作成してください。(40分)



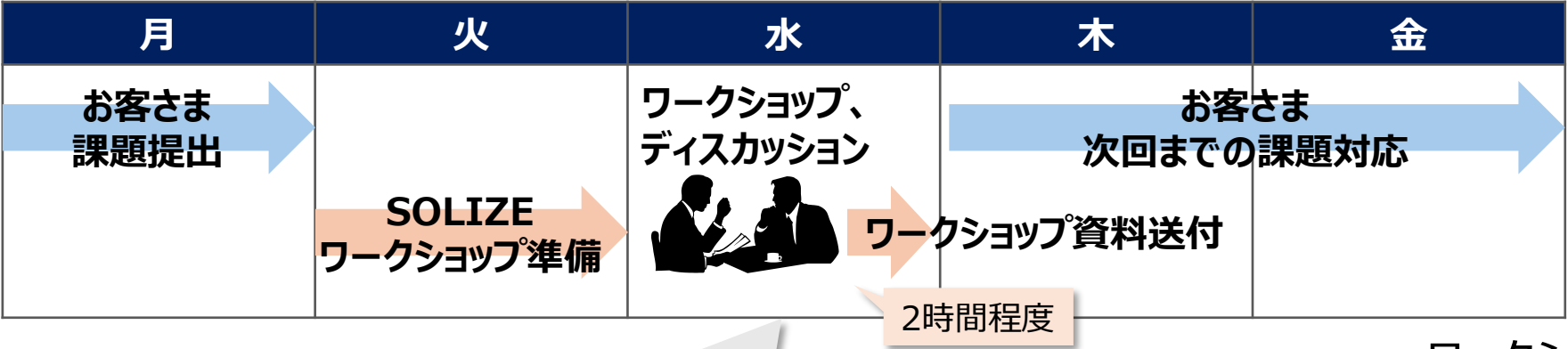
Copyright 2023 by SOLIZE Corporation All Rights Reserved.

11

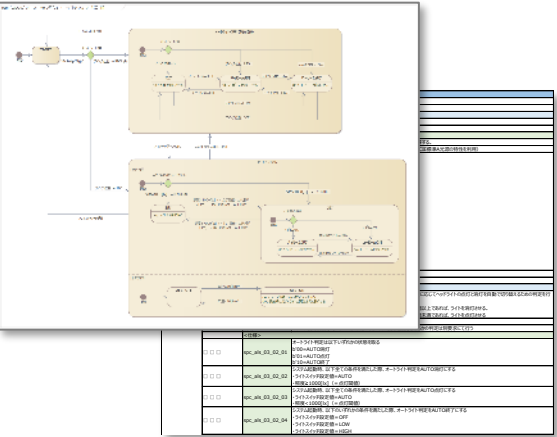


# SysML実践支援

お客さまの製品のSysMLモデルをSOLIZEと共に作成することで、  
実業務に適用するためのノウハウを習得する

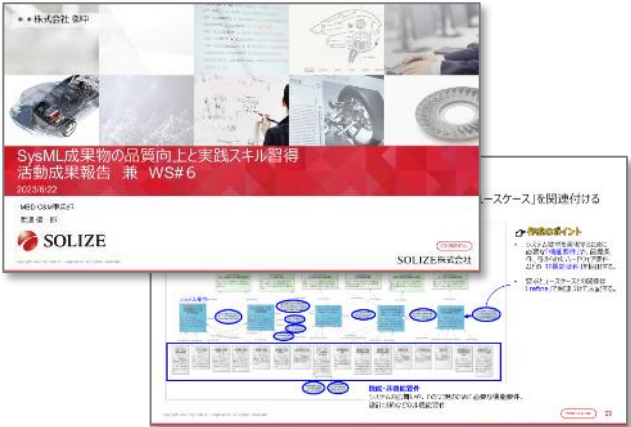


SysML成果物（課題）



- SOLIZE→お客さま
  - ・お客さまが作成したSysML成果へのフィードバック
  - ・次回ワークショップまでの課題提示、サンプル紹介
- お客さま→SOLIZE
  - ・課題対応、提出

ワークショップ資料  
（宿題へのフィードバック）



## 実績

- AD/ADASに関するシステム（AEBS、LKS etc.）
- EVに関するシステム（BMS、TMS etc.）

## よくある状況

- 海外OEMからの引き合いが増えており、それに伴いA-SPICEへの対応が求められている。
- A-SPICEの能力レベル1 相当の成果物（要件仕様書・制御仕様書）は存在する。
- 仕様書はWordで作成されており、基本はテキストベースで構成されているが、部分的にはSysML図での表現を含む。

## よくある課題

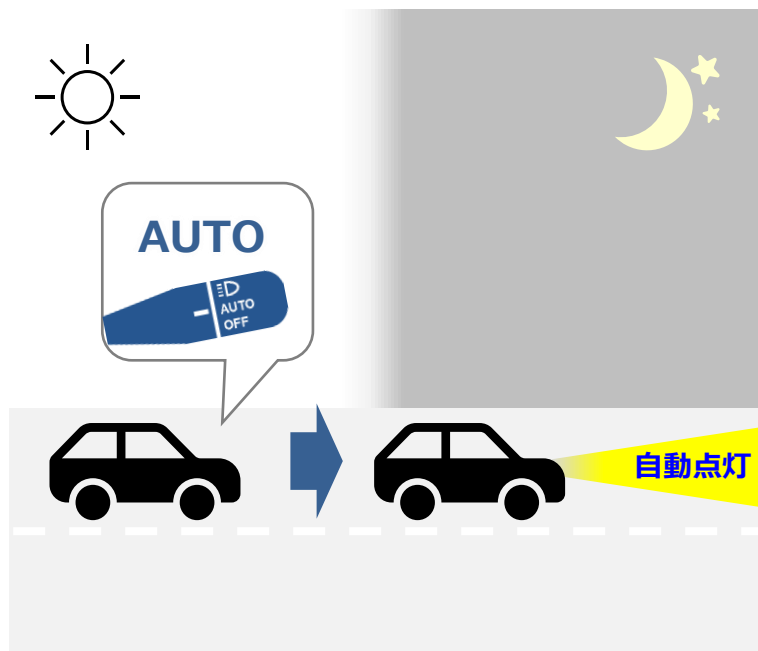
- 開発対象システムの範囲（SoI：System of Interest）がわかりにくい。
- 関係する外部システムや利害関係者が十分に表現されていない。
- 対象システムに求められる、ユーザー要求や車両としての要求が把握できない。
- 要求図は各階層で何を表現するかが明確ではなく、網羅性の確認が行いにくい構成となっている。
- ユースケースが十分に抽出されていない、ユースケースごとにどのような振る舞いを行いたいかが十分に表現されていない。
- 要求や振る舞いを実現するための機能を、体系的に関連づけて管理できていない。
- SysMLの各成果物の関係性が不明確。



ヘッドライトの以下の機能を題材に、SysML実践支援の流れをご紹介します

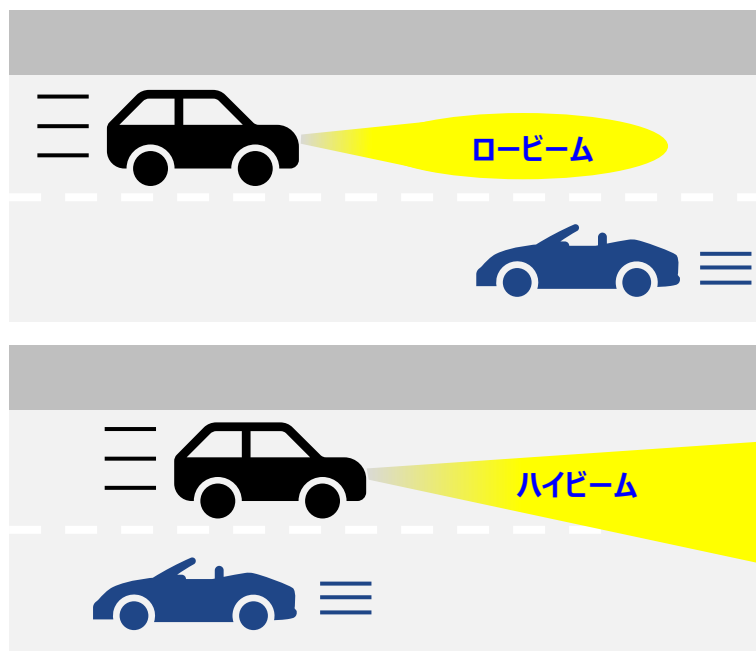
## オートライト

周囲の明暗に合わせて自動でヘッドライトを点灯/消灯する



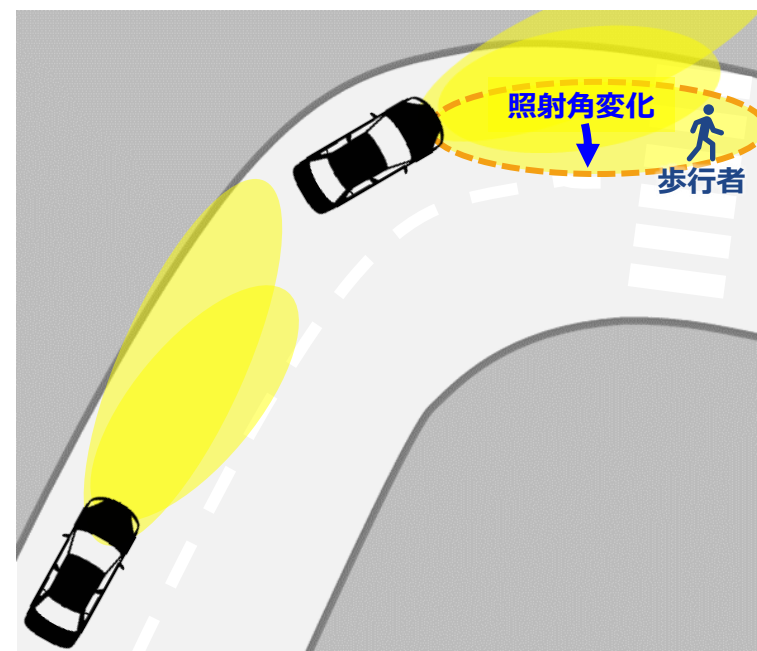
## オートハイビーム

前方の車や対向車がない場合は自動でハイビームにする



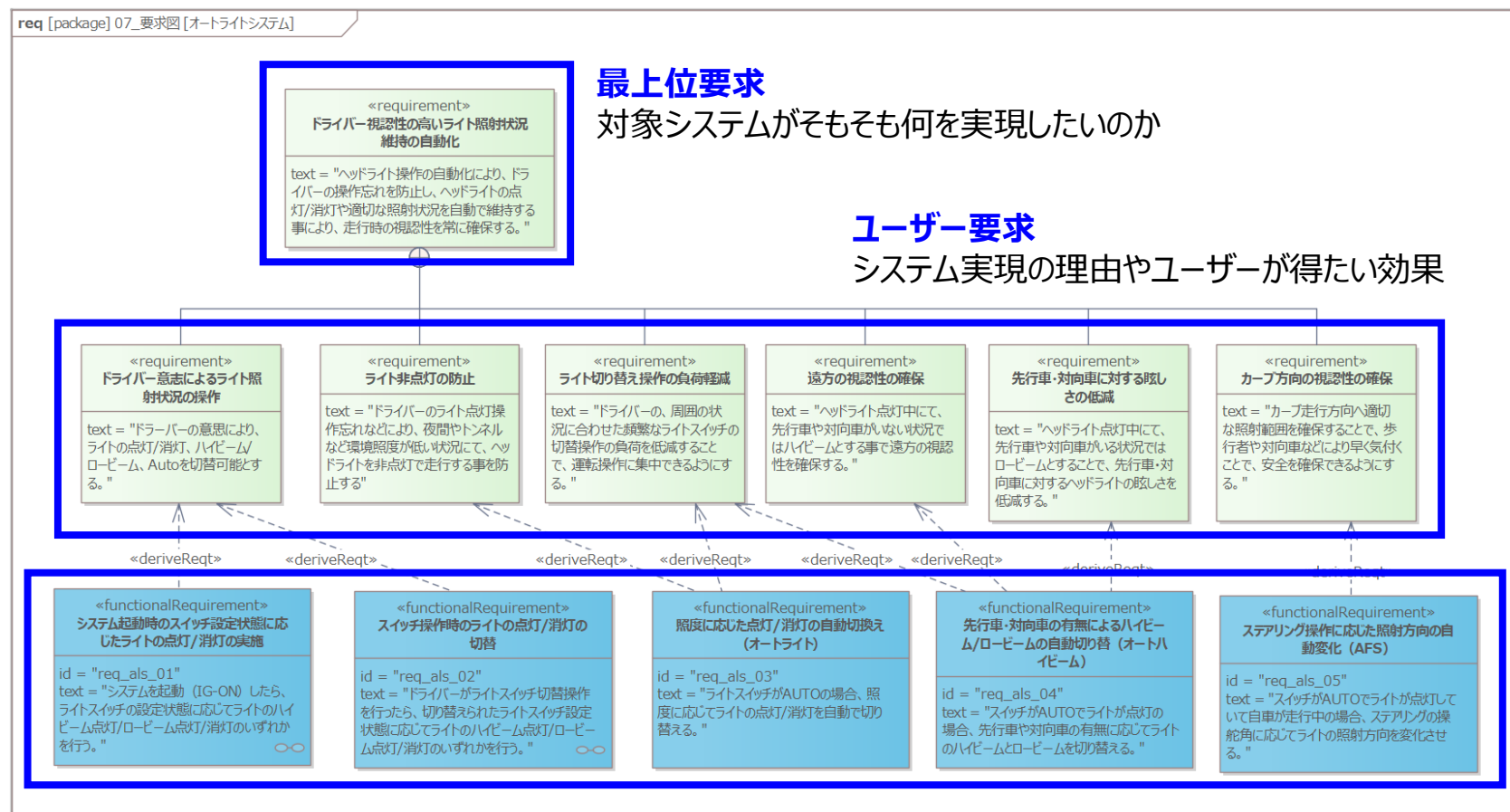
## AFS (Adaptive Front-lighting System)

コーナーで車速や舵角に応じてライトの照射角度を変化させる



## システム実現の「理由」や「ユーザーが得たい効果」を明確にし、システム要求と関連づける

### ■ 要求図（ユーザー要求）



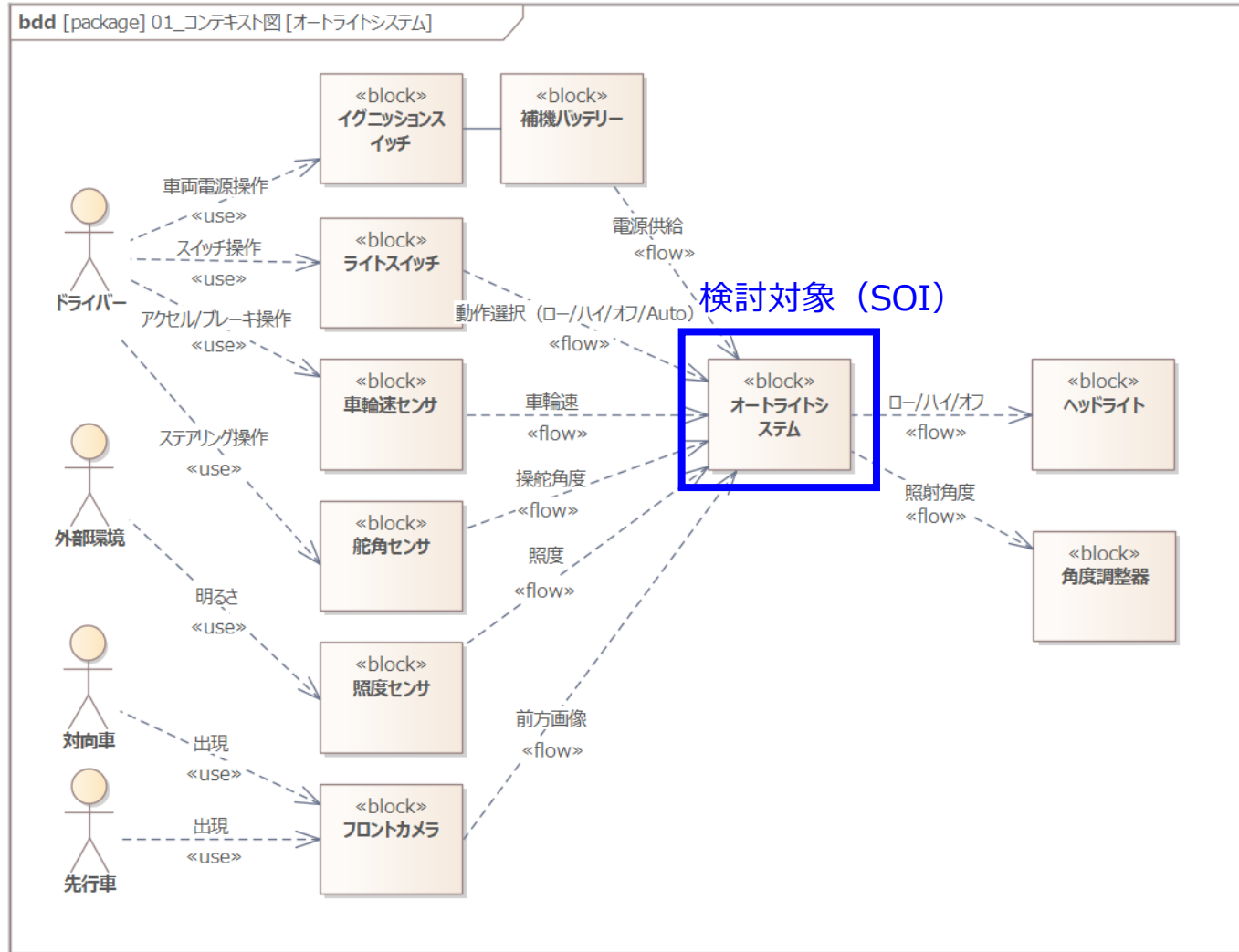
### ■ 要求図

- システムが満たすべき「要求」「機能」「要求と機能の関係」を表現するためのダイアグラムであり、検討すべき内容を網羅的に表現する。
- 開発内容の妥当性を検証するためにも使用されるため、開発が進むにつれてさまざまな情報が追加されていく。

### 👉 作成のポイント

- 先に要求の階層を設定し、その階層を意識して作成することで、要求の粒度がそろい、第三者が見ても把握しやすい体系だった要求図となる。
- ユーザー要求を最初から考えにくい場合は、システム要求（実現するサービス）を最初に定義して、それが必要な理由やユーザーが得られる効果を検討するとやりやすい。

## 開発対象システムと他システム、ステークホルダー間の関係を洗い出す



### ■ コンテキスト図

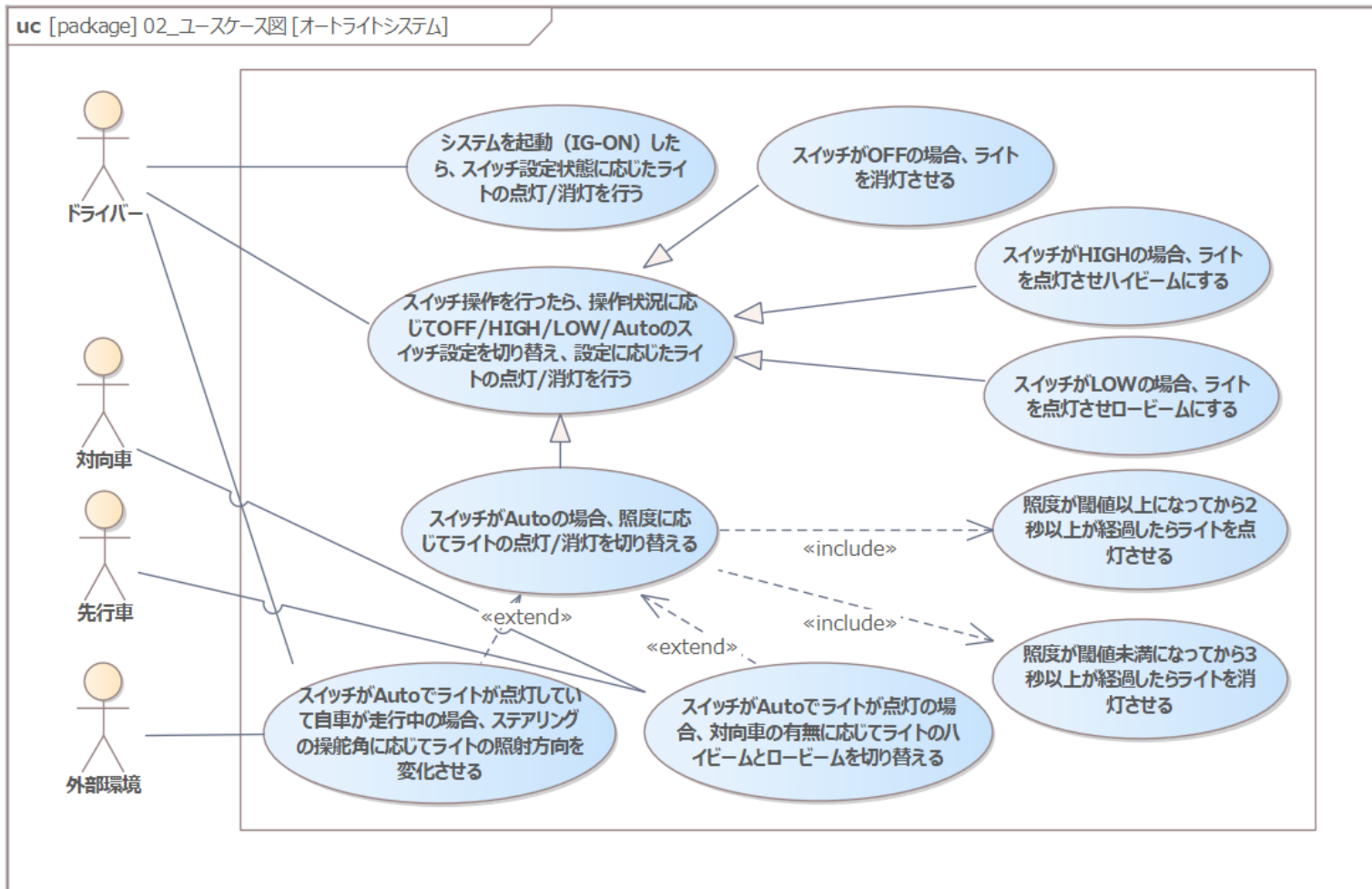
- 「コンテキスト図」は、対象システムが動作する環境・外部要素を明示するためのダイアグラム。
- 対象システムに影響を与える要素、対象システムと何らかの相互作用する要素を記述する。
- ただし、すべての環境要素を記述するのではなく、以降の分析/設計作業において、設計者が着目し、検討を行うべき要素のみを記述する。

### 👉 作成のポイント

- 開発対象がサブシステムの場合は、どこまでの範囲を対象システムとしてとらえるべきかが不明確になりやすい。
- 検討対象システム (SOI) を明確化し、ステークホルダーや他システムとの関係性を明確にすることにより、関係者間で開発の全体像が共有でき、どこに注力して開発すべきかといった議論が行いやすくなる。



## 外部からの要求に対して「システムが提供するサービス」を表現する



### ■ ユースケース図

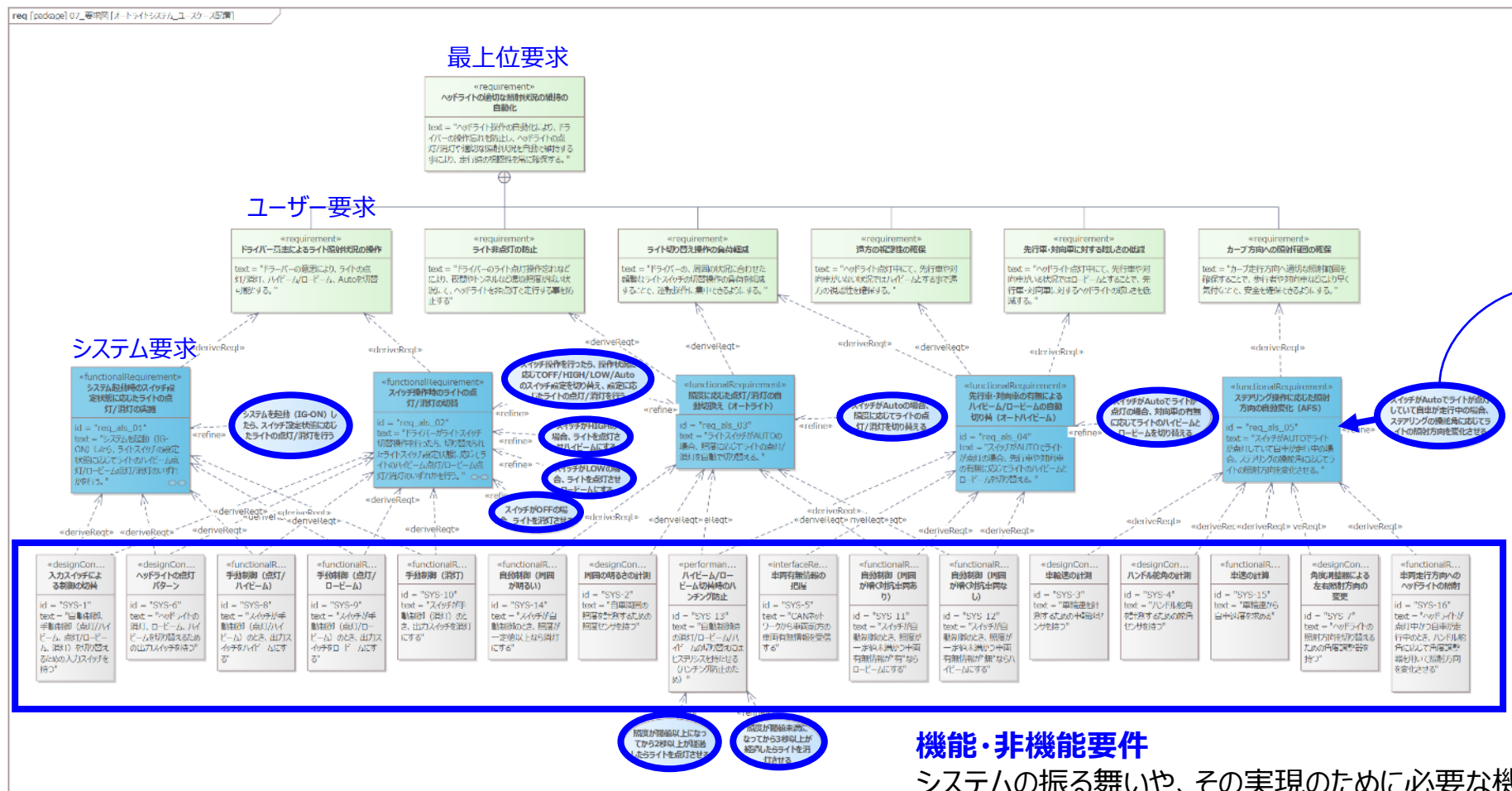
- 「ユースケース図」は、対象システムが外部に提供する機能を表現するためのダイアグラム。
- ユースケース図を用いることで、開発対象範囲を明確にすることができる。
- どのユーザーにどのような機能を提供するのか、システムはどのような外部要素を必用とするのか、を明確化することができる。

### 👉 作成のポイント

- 利害関係者が対象システムに対してどのような操作を行うと、対象システムはどのようなサービスを提供するか、という視点で記載する。
- 要求との重複があるのは問題はない。後ほど要求図上で関連づけを行う。
- 要求図との違いは、要求図では要求の実現手段や非機能要件も表現するが、ユースケース図ではシステム内部はブラックボックスとし、表現しない。

システム要求を実現するための機能分析を行い、「要求」「機能」「ユースケース」を関連づける

## ■要求図（ユースケースとの関連付け）



## 👉 作成のポイント

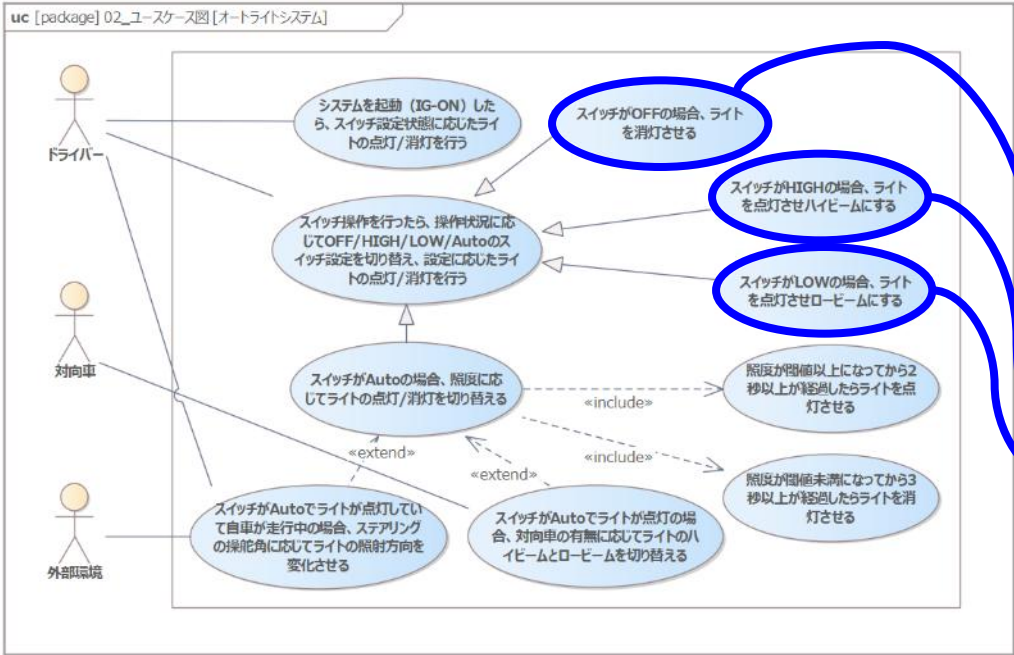
- システム要求を実現するために必要な「機能要件」や、前提条件、与えられたハードウェア要件などの「非機能要件」を抽出する。
- 要求とユースケースとの関係は「refine」で関連づけて表記する。

## 機能・非機能要件

システムの振る舞いや、その実現のために必要な機能要件、  
設計制約などの非機能要件

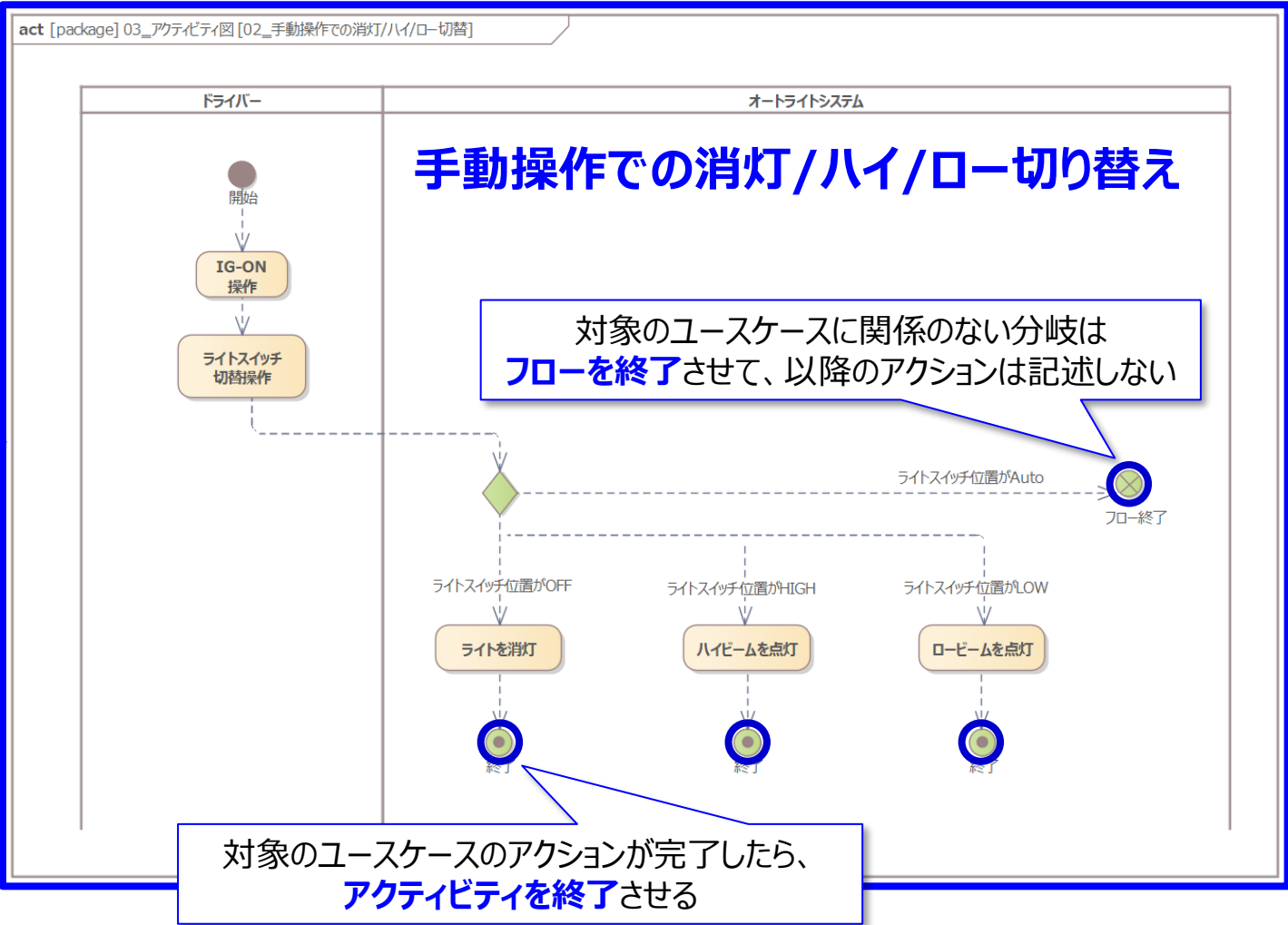
# システムの振る舞い定義

ユースケースごとの振る舞いを「アクティビティ図」で表現し、関係者間で共通認識のもと、理想とする振る舞いへと洗練させる



## ■ アクティビティ図

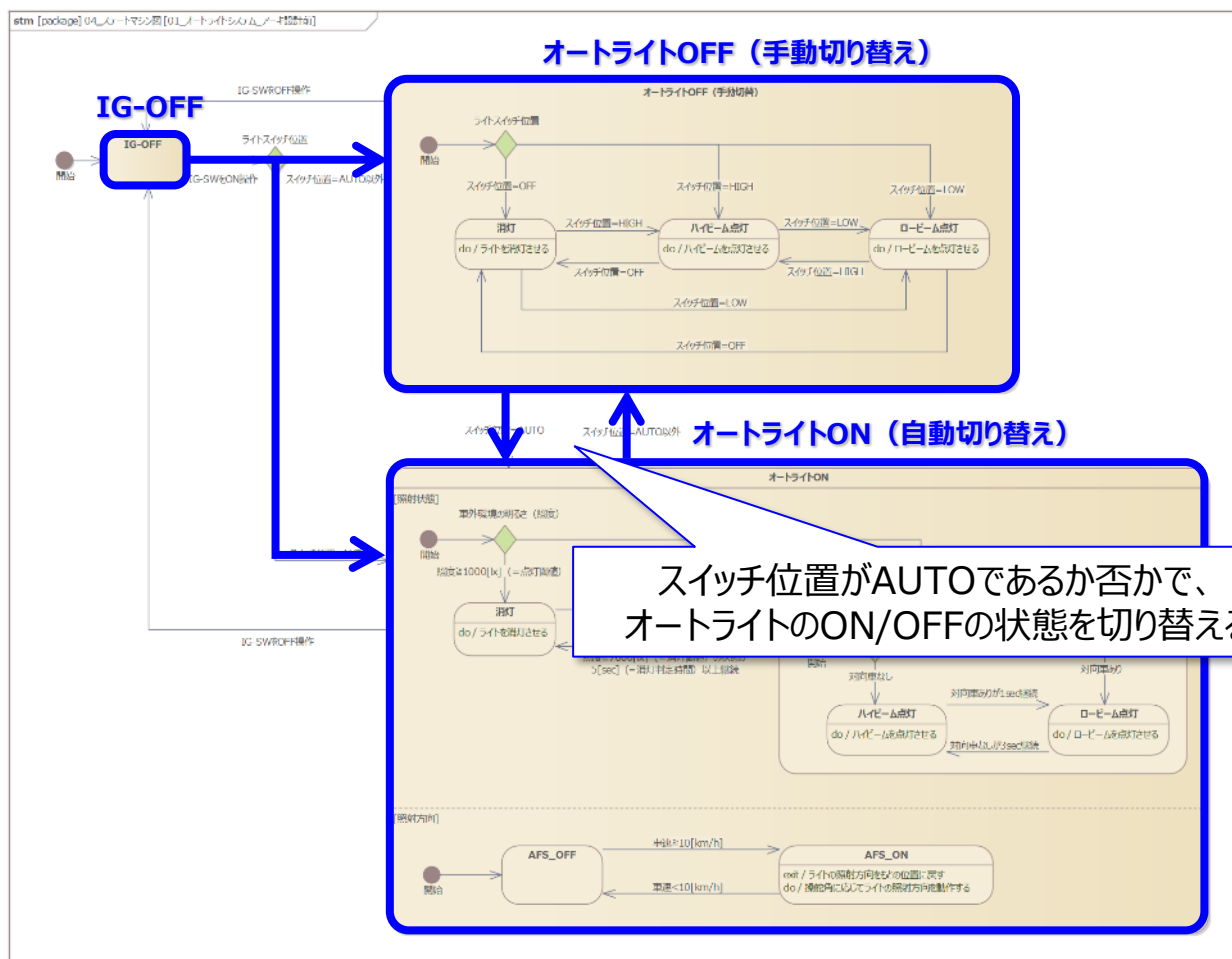
- 「アクティビティ図」は、処理の流れや実行手順を整理して表現するためのダイアグラム。
- ビジネスプロセスの記述や、機能を実現するためのアルゴリズムの検討など、さまざまな場面で利用できる。





# 車両の振る舞いの集約

ユースケースごとのアクティビティ図が総合的に成り立つよう  
入力条件の分岐とそれに応じた車両挙動を一つの「ステートマシン図」にまとめる



## ■ステートマシン図（状態遷移図）

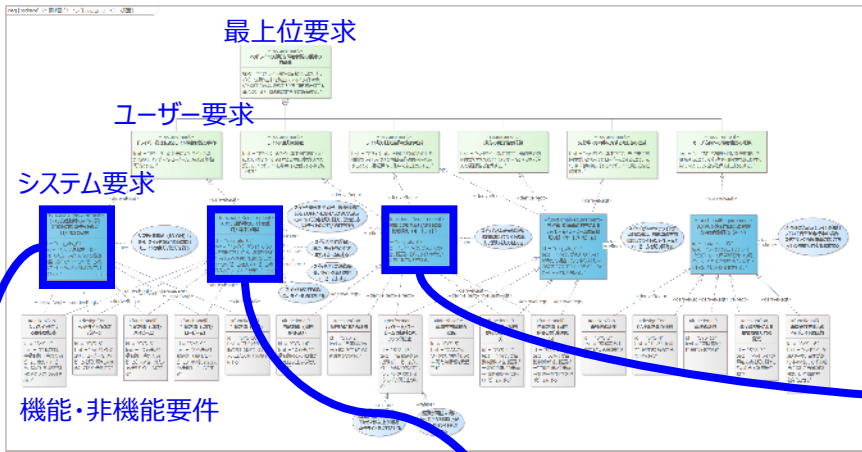
- イベントにより引き起こされる **システムの状態遷移** を表現するためのダイアグラム。
- 状態が変化していく流れ（遷移）と、状態が遷移するきっかけ（トリガー）を記述することが可能。
- 各状態で**実行する振る舞い（アクティビティ）** も記述することができる。

## 👉作成のポイント

- 各アクティビティ図が**総合的に成り立つ**よう、一つの**システム応答のステートマシン図**にまとめる。
- ステートマシン図の状態は、「**システムのOFF/ON**」、「**システムがONの時の振る舞いの種類**」、の構成で定義する。

システムが満たすべき「要求」や「機能」を分析し、「要求と機能の関係」を構造化する

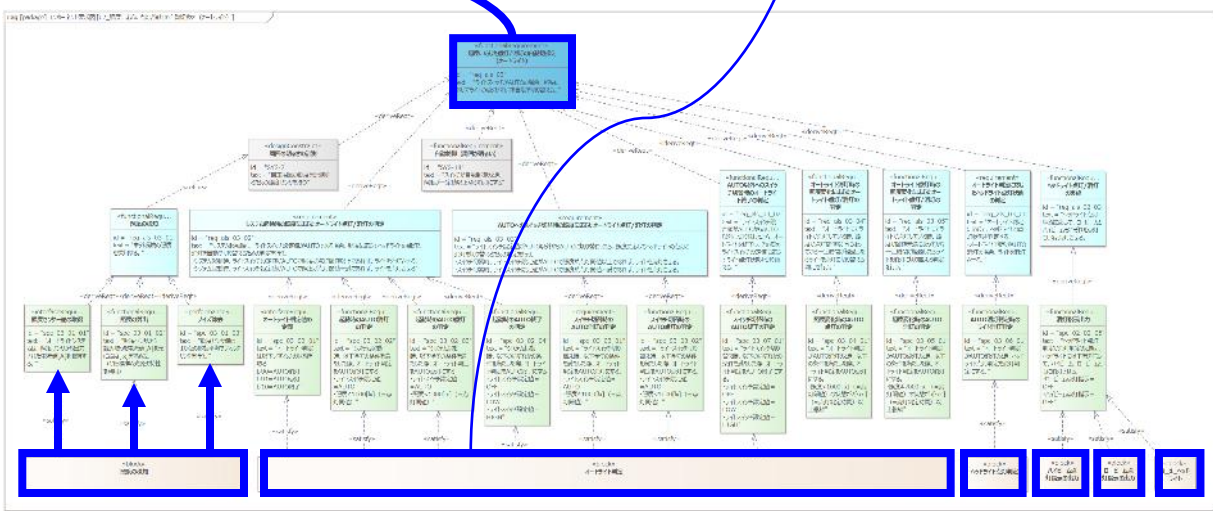
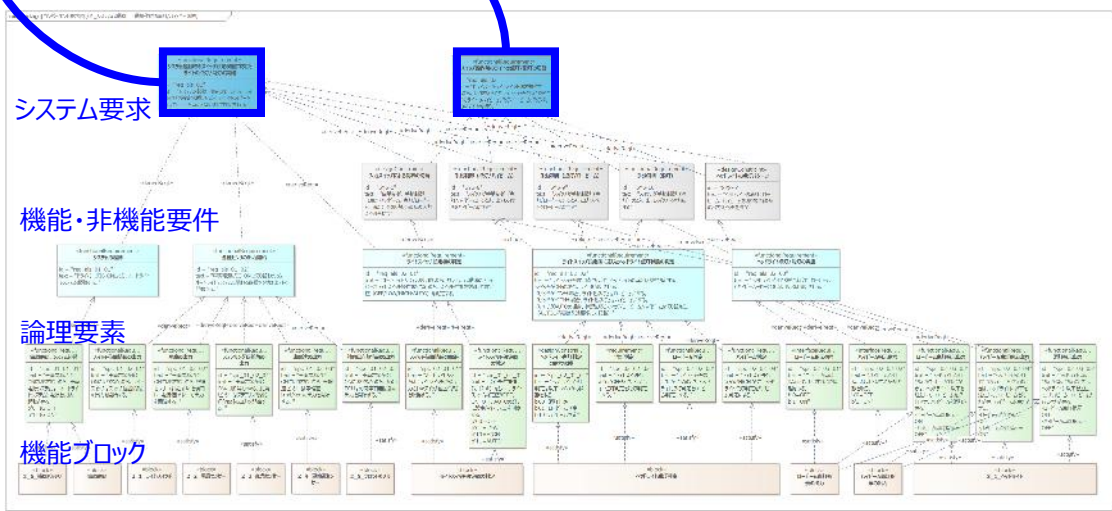
■要求図（システム全体）



作成のポイント

- 一つの要求図にすべての階層の要求を記載すると、図が大きくなりすぎて管理性や視認性が低下する場合は、システム全体の要求と、機能ごとの要求を別の要求図にわけて記載する。
- 機能ごとの要求図では、機能・非機能要件を実現方法である論理要素に機能展開したのち、整理・集約し、機能ブロックへ機能配置する。
- 要求（論理要素）と機能ブロックとの関係は「satisfy」で関連づけて表記する。

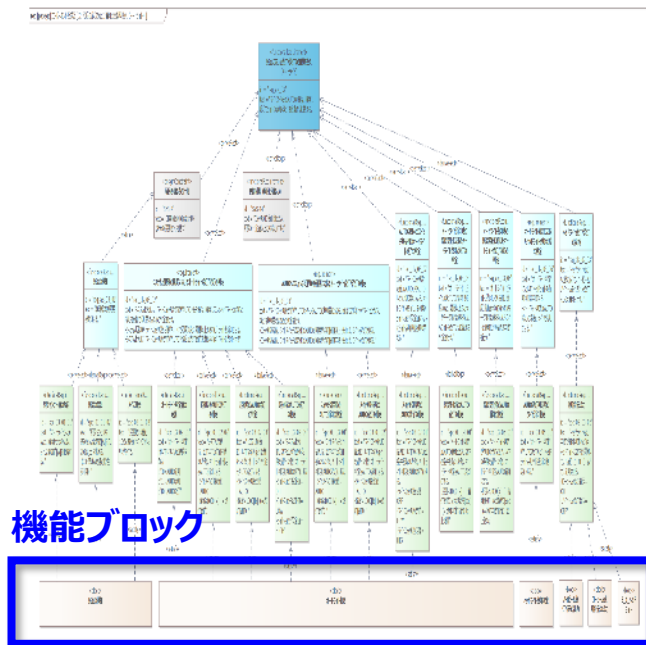
■要求図（機能ごと）



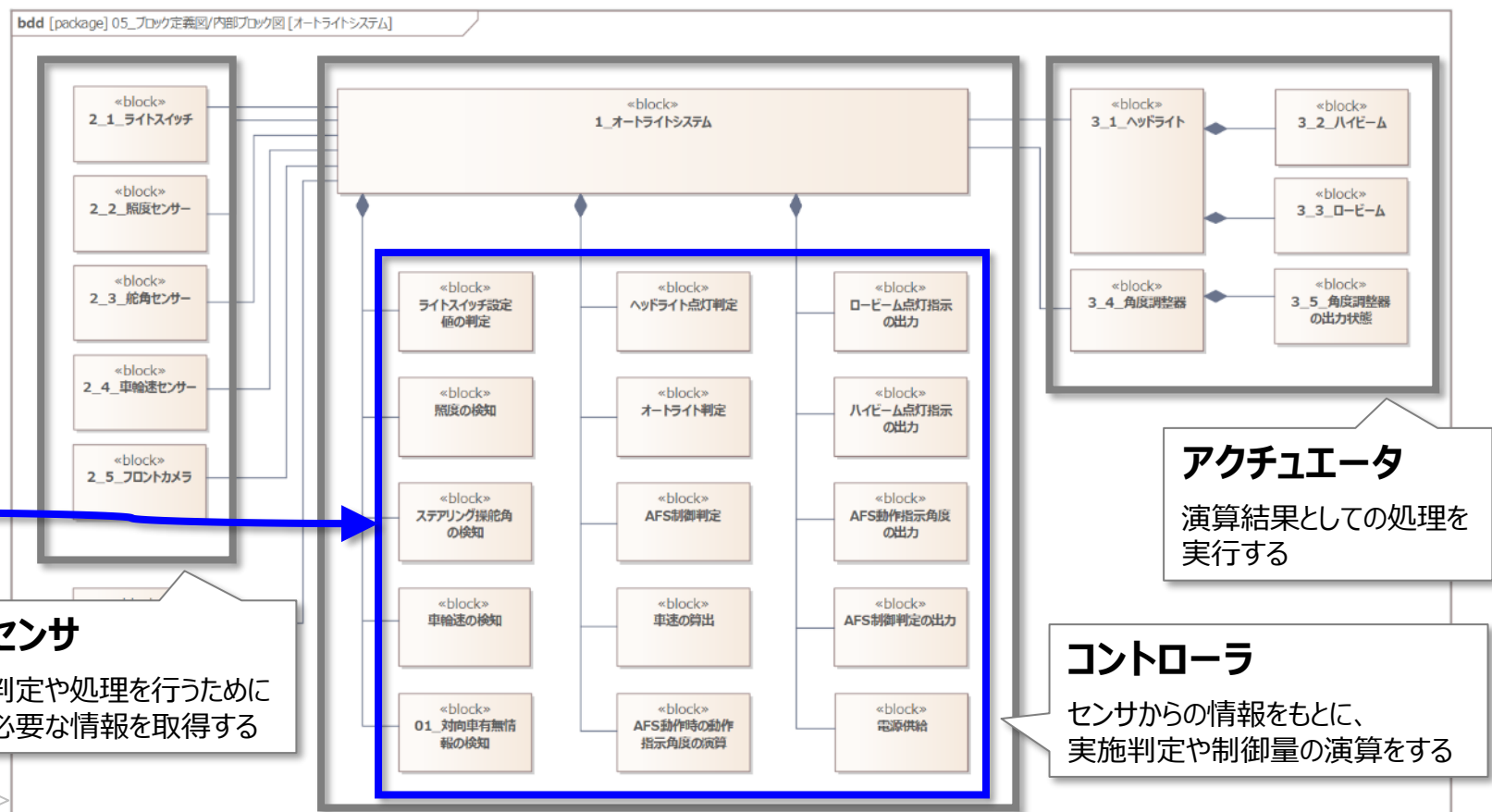
# ブロック定義図の作成

システム機能要件とシステム構成との関係性を整理し、「ブロック定義図」で表現

## ■ 要求図



## ■ ブロック定義図



## ■ ブロック定義図

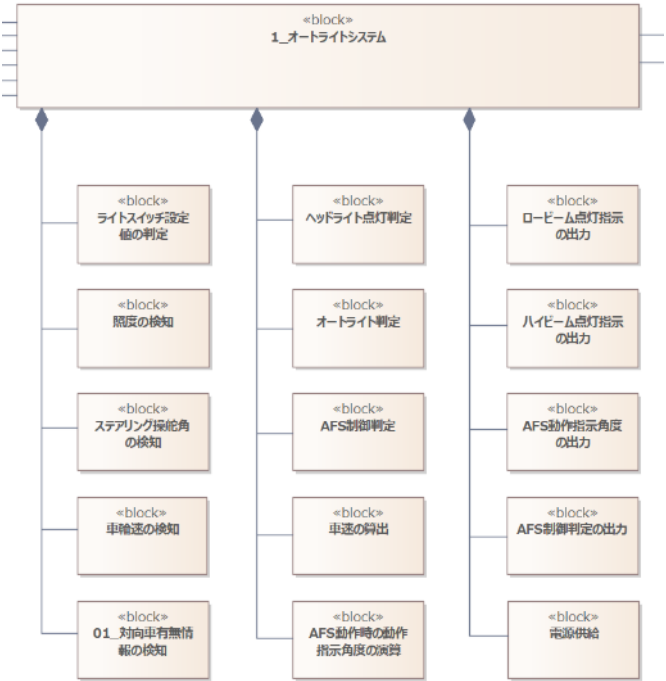
- システムの構成要素を定義するためのダイアグラム。
- 他システムとの関係や、ブロックの内部構造を表現できる。



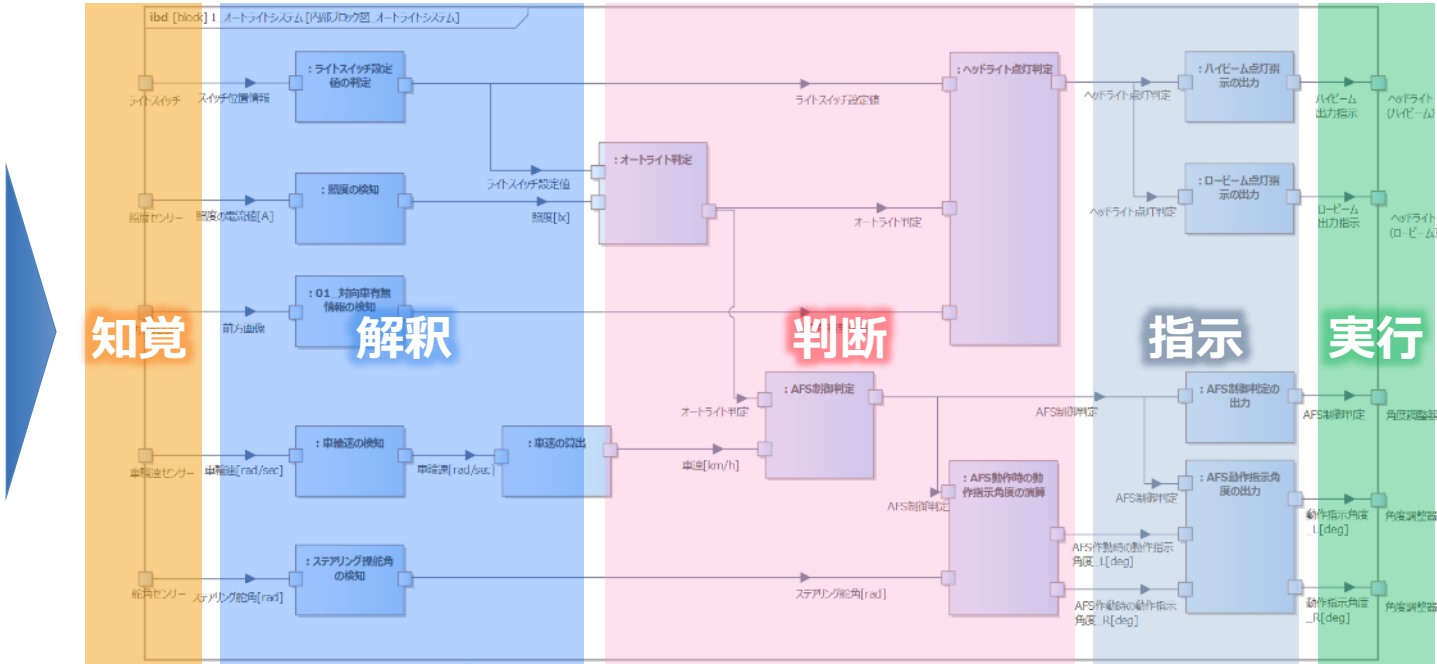
# 内部ブロック図の作成

機能ブロックを整理して並べ、論理アーキテクチャとしての「内部ブロック図」を作成する

■ブロック定義図



■内部ブロック図



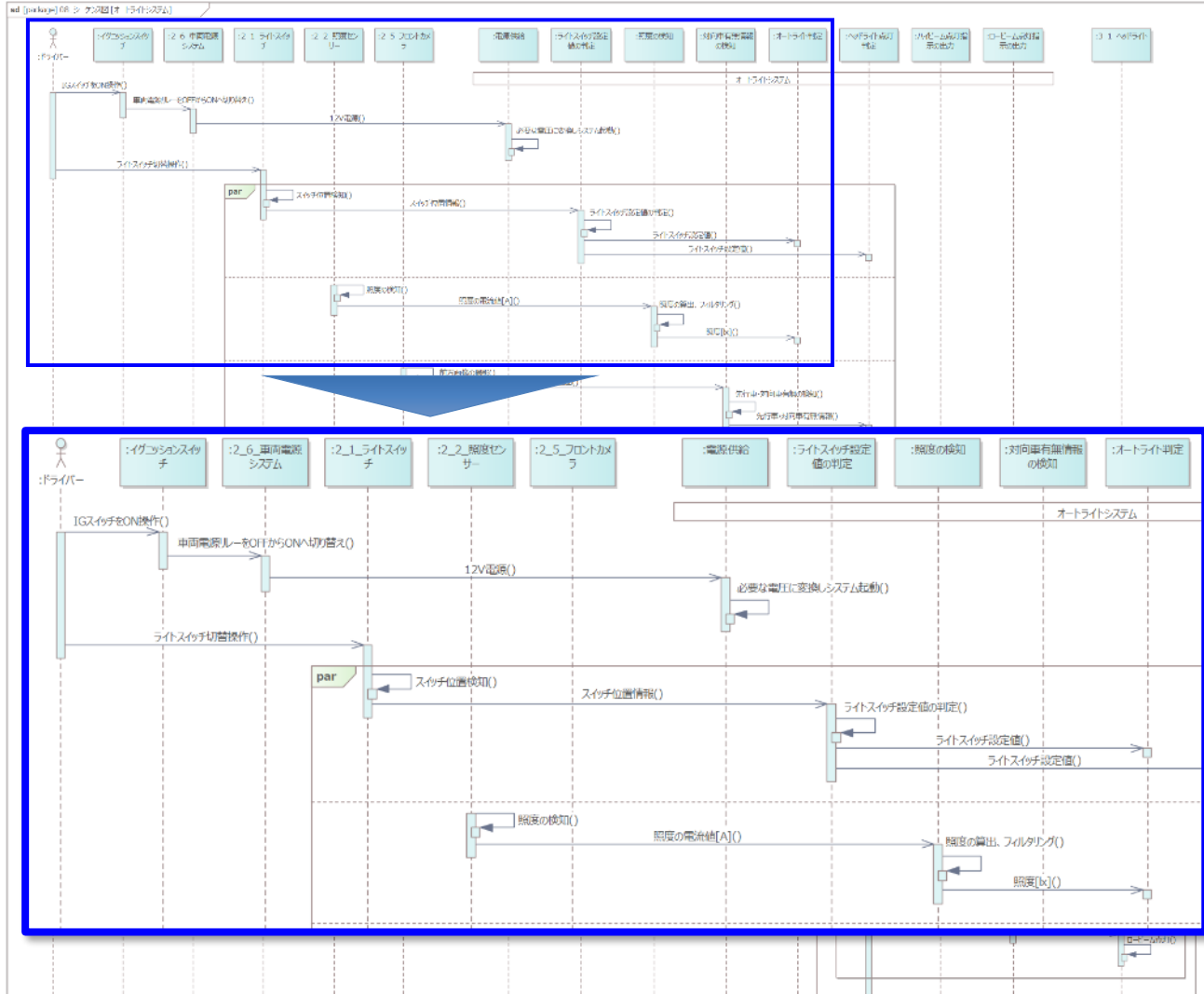
■内部ブロック図

- ブロック定義図で定義したシステムの構成要素がやり取りする内容を表現するためのダイアグラム。
- 構成要素間の接続関係を明確にするとともに、構成要素間でやり取りするデータやシグナルを明確化する。

## 作成のポイント

- 機能ブロックを「知覚→解釈→判断→指示→実行」の順で整理して並べると、理解しやすい構成となる。

機能ブロック間の情報と処理の流れを「シーケンス図」で表現して確認する

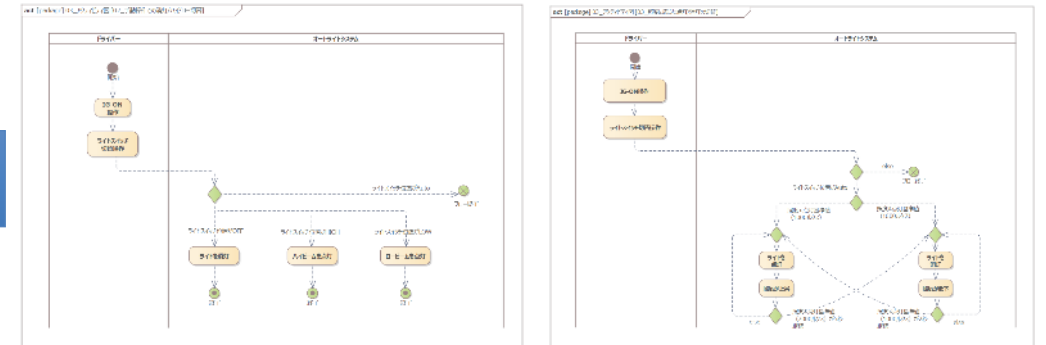


## ■シーケンス図

- 「シーケンス図」は、ある「機能」を実現するための処理の流れ（相互作用）を時系列に沿って表現するダイアグラム。
- ダイアグラムの上から下に向かって時間が流れるため、時間に沿った形で処理の流れを表現することができる。
- 要素間のやり取り（呼び出し関係、データの受け渡し関係など）や各要素の振る舞い（操作、受信）を明確にすることができる。

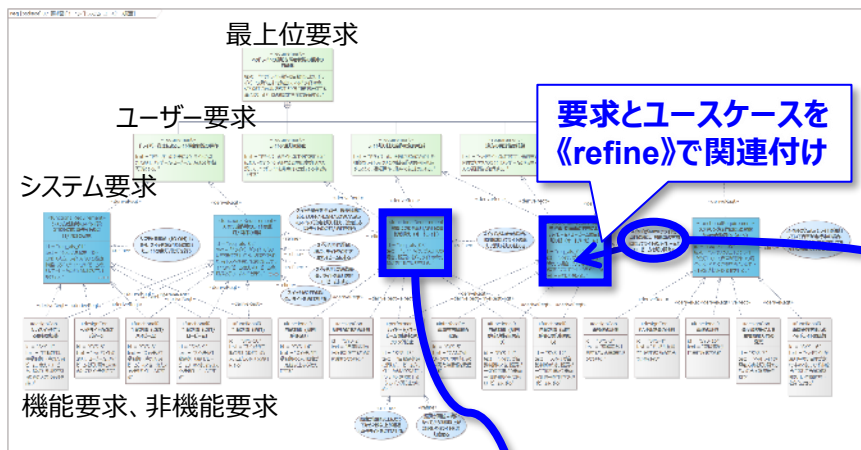
## 👉作成のポイント

- アクティビティ図で表現した初期検討時の操作や処理の流れをもとに、シーケンス図にてコンポーネント間での情報と処理の流れとして詳細に表現することで、検討したアーキテクチャで処理が正確に実行できるかを確認する。

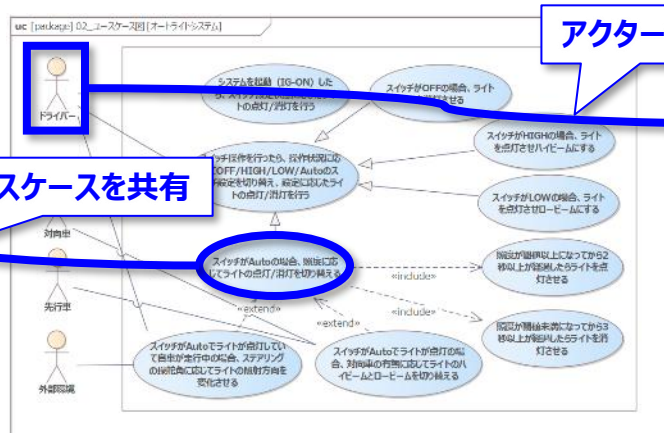


## 各ダイアグラム間で関係性を保って、トレーサビリティを担保する

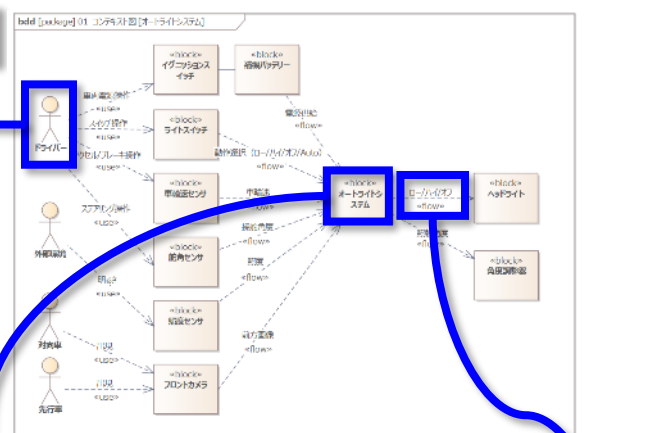
### ■ 要求図 (システム全体)



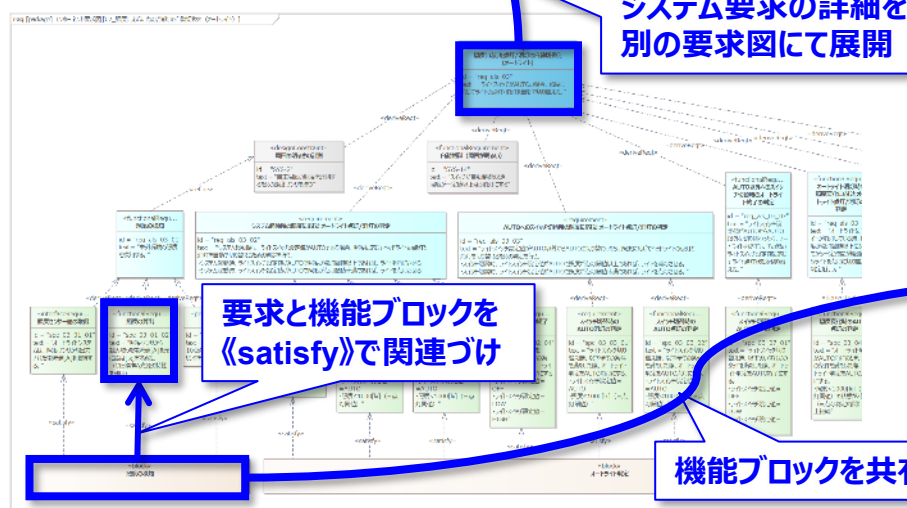
### ■ ユースケース図



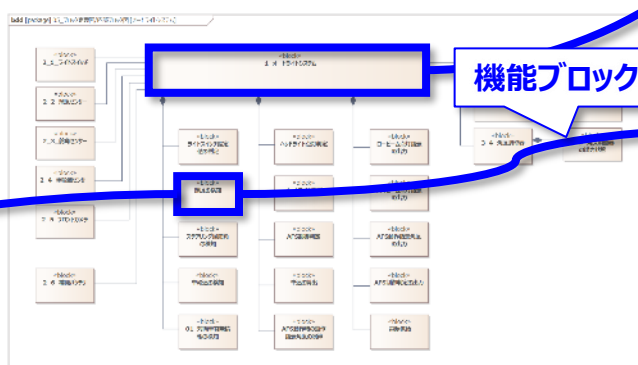
### ■ コンテキスト図



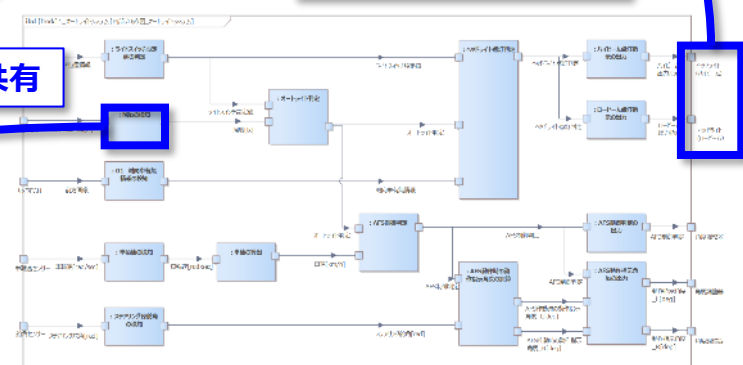
### ■ 要求図 (機能ごと)



### ■ ブロック定義図



### ■ 内部ブロック図

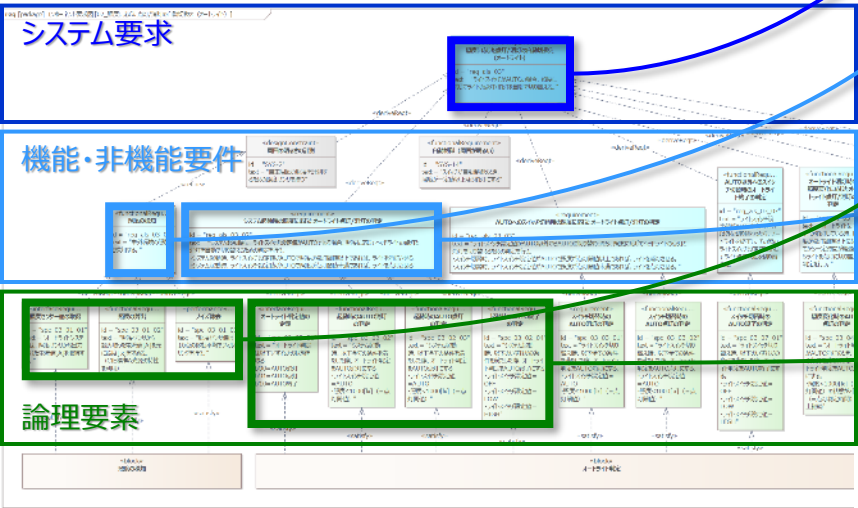


必要に応じて、要求・仕様の内容を可読性の高い「USDM」形式にて記述する

■ USDM (Universal Specification Describing Manner)

- 直訳すると「万人に通じる仕様を描写する作法」
- システム要求を仕様化するための記述手法であり、「要求」「理由」「仕様」を階層化することで、「読みやすく」「背景・理由が把握しやすく」「抜けものない」記述ができる。
- 要求は、動詞と目的語に着目し、システム動作の一連の流れである「振る舞い」として表現する。

■ 要求図（機能ごと）



■ USDM形式の要求仕様書

上位要求

下位要求

仕様

下位要求

仕様

<照度に応じた点灯/消灯の自動切替>		
上位要求	req_als_03	スイッチがAUTOの場合、照度に応じてライトの点灯/消灯を切り替える。
理由		ドライバーのライト点灯操作忘れなどにより、夜間やトンネルなど環境照度が低い状況にて、ヘッドライトを非点灯で走行する事を防止するため。ドライバーの、周囲の状況に合わせた頻繁なライトスイッチの切替操作の負担を軽減することで、運転操作に集中できるようにするため。
下位要求	req_als_03_01	車外環境の照度を検知する。
		照度により、点灯/消灯のON/OFFを判定するため。
		<仕様>
	spc_als_03_01_01	オートライトシステムは、照度センサから出力された電流値[A]を取得する。
	spc_als_03_01_02	照度センサから読み取った電流値[A]を元に照度[lx]を求める。（CIE標準A光源の特性を利用）
	spc_als_03_01_03	照度センサ値は10点の移動平均でフィルタリングを行う。
		照度センサ値の平滑化のため。
下位要求	req_als_03_02	システム起動時に、ライトスイッチ設定値がAUTOである場合、照度に応じてヘッドライトの点灯/消灯を自動で切り替えるための判定を行う。
理由		・システム起動時、ライトスイッチ設定値がAUTOで照度が点灯閾値以上であれば、ライトを点灯させる。 ・システム起動時、ライトスイッチ設定値がAUTOで照度が点灯閾値未満であれば、ライトを点灯させない。 ・エンジン起動時に、自動でオートライトのON/OFFを行いたい。
		<仕様>
	spc_als_03_02_01	オートライト判定は以下いずれかの状態を取る b'00= AUTO消灯 b'01= AUTO点灯 b'10= AUTO終了
	spc_als_03_02_02	システム起動時、以下全ての条件を満たした際、オートライト判定をAUTO消灯にする ・ライトスイッチ設定値= AUTO ・照度 ≥ 1000[lx] (= 点灯閾値)
	spc_als_03_02_03	システム起動時、以下全ての条件を満たした際、オートライト判定をAUTO点灯にする ・ライトスイッチ設定値= AUTO ・照度 < 1000[lx] (= 点灯閾値)
	spc_als_03_02_04	システム起動時、以下のいずれかの条件を満たした際、オートライト判定をAUTO終了にする ・ライトスイッチ設定値= OFF ・ライトスイッチ設定値= LOW ・ライトスイッチ設定値= HIGH



**1 「SysML基礎教育」**にて、  
SysMLモデルおよび各種ダイアグラムの読解・作成ができるようになる。

**2 「SysML実践支援」**にて、  
お客さまの製品のSysMLモデルをSOLIZEとともに作成することで、  
実業務に適用するためのノウハウを習得する。



SysMLを**お客さま自身で活用**できるようになり、**効果を実感**できる

SOLIZEのMBSE手法を詳しく知りたい方は、オンデマンド配信セミナーをご視聴ください。

### ■ 体系的・網羅的なシステム要求抽出

[「自動車のブレーキ機能」を題材としたシステム要求抽出の方法](#)

<https://www.youtube.com/watch?v=Dmy2C3Tn90w>

### ■ アーキテクチャ設計とテストパターン作成

[「自動車のオートライト機能」を題材としたアーキテクチャ設計とテストパターン作成](#)

<https://www.youtube.com/watch?v=5KhaUPHXUqc>

### ■ 高齢者向け、電動キックボードを題材にしたMBSE-MBD連携事例

[【SOLIZE × MathWorks 共催セミナー】MBSEとMBDの連携エンジニアリング](#)（前半）

<https://www.youtube.com/watch?v=OpKD2i9gKBw>

[【SOLIZE × MathWorks 共催セミナー】](#)（MathWorks様 後半）

<https://content.mathworks.com/viewer/67b6e09e38c0d27b7658d5f4#1>

サービス・資料に関するお問い合わせはこちらまでお願いいたします

**SOLIZE Ureka Technology 株式会社 営業担当**

**[contact.mbd@solize.com](mailto:contact.mbd@solize.com)**

**MBD/MBSEに関する資料ダウンロード/お問い合わせ**



SOLIZE

Ureka Technology