

# RTミドルウェア SUMMER CAMP 2024

## SysML実習

日時: 2024年8月19日(月) 15:00～

株式会社 グローバルアシスト  
坂本 武志

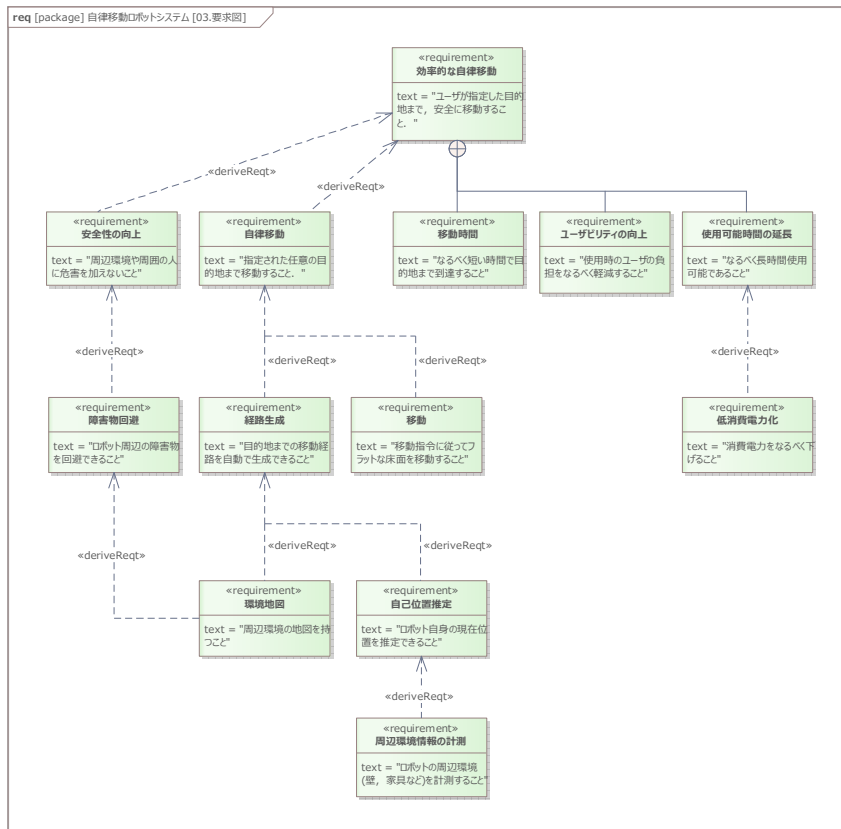


## 本実習の目的



- **サマーキャンプを通して開発を行う対象システムの明確化**
  - チーム内の意識のすり合わせ
    - どのような機能/性能を持つシステムを開発するのか？
    - どのような構成のシステムを開発するのか？
  - 対象システム/開発内容の(第三者への)説明
    - 何が特徴なのか？
    - どのようなコンポーネントを使用/作成するのか？
- **SysMLを使用する理由**
  - 分析, 設計内容の明確化
    - 分析, 設計の結果を, その場限りの独自記法で表現するのではなく, 国際標準となっている記法で表現して欲しい
  - 開発したRTコンポーネント公開時に, 設計情報も併せて公開して欲しい
    - 後からRTコンポーネントを利用する人に開発者の意図を伝えるため

## ■ 性能なども含めたシステムが満たすべき項目を表現



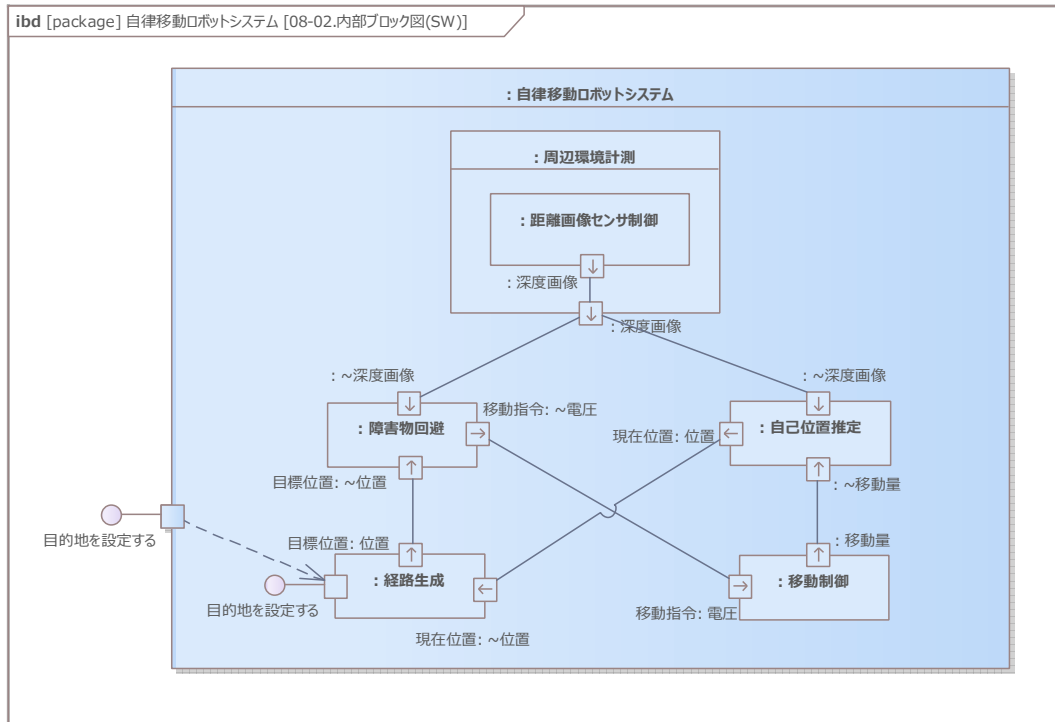
## ■ 目的

- 開発対象システムの説明
  - どのような機能/性能、特徴を持ったシステムを構築するのか？
- 開発時の代替案の検討

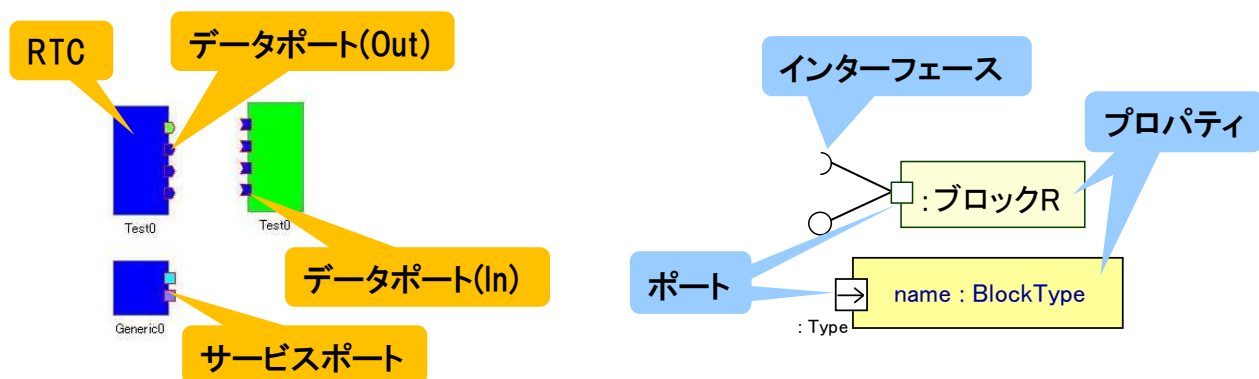
## ■ 作成手順

- 「何を」実現するシステムなのか？を検討
  - それぞれの要求を実現するためには、「何を」行わないといけないのか？を検討
- ※要求図では、具体的な実現手段は記述しないため、「どのように」実現するかの詳細は初期段階では気にしない

- システムを構成する要素間でやり取りする内容を表現
  - システム構成要素間の接続関係を明確にする
  - 構成要素間でやりとりする内容を明確にする



- RTSystemEditorで構築したシステム情報とほぼ同じ内容を表現



OpenRTM-aist	SysML
RTコンポーネント	プロパティ(ブロック)
データポート	ポート(連続型)
サービスポート	ポート(離散型)
サービスインターフェース	インターフェース

## ■ 作成手順

- 使用するハードウェアの検討
  - どの機材を利用するのか？不足している機材は何なのか？
- 必要な機能の検討
  - どのようなRTCで構成するか？
  - RTCを新規開発するのか？既存のRTCを再利用するのか？
- RTC間のインタラクションの検討
  - RTC間でやり取りするデータ、コマンドの検討
    - ◇ まずはPlatform(RTミドルウェアなど)に依存した内容ではなく、より抽象度の高い内容で検討
- 使用するPlatformに対応した内容の検討
  - 使用するコンポーネント、各データ型、インターフェースの詳細をPlatformに対応した形にマッピング

# 設計時の注意点

## ■ 検討事項

- 必須機能の判断
  - 最低限どの機能が必要なのか？
  - 代替案の検討
    - ◇ **必須機能とオプション機能**の検討
      - ◇ 初期段階では、どうしても偏ったシステム構成としがちなため
      - ◇ 各RTCの粒度(機能)をあまり大きくしすぎない
- 開発工数の見積もり
  - 各機能を開発するのに必要な工数はどの程度か？
  - 再利用RTCの検討
    - ◇ 開発工数を下げるため、既存のRTCで使えるものがないか検討
- 優先順位の決定
  - どのRTCから開発を行うのか？
- 作業分担の検討
  - 誰がどのRTCを開発するのか？

## ■ 作業工数の見積もり

- (できれば)担当者を決める前に、全員で各RTC、各機能の開発時間の見積もりを行う
- 今回は期間が1週間と短いため、**時間単位での見積もり**を行う
  - 通常の開発では、日単位で見積もりを行う事が多い
- 全員で各RTC、各機能について、作業時間の見積もりを行う
  - 既存のコンポーネントを再利用する場合でも、再利用可能かどうかを**検証するのにも時間が掛かってしまう**場合がある
    - ◇ 使用する環境、インターフェース(データ型、操作の詳細)が合致しているか確認
- 実際に開発にかかった時間を計測しておき、**見積もり値との差異を記録**する
  - この差異を元に、何度か再見積もりを行う
    - ◇ 今回は期間が短いので、可能であれば毎日再見積もりを行う
    - ◇ 現状の進捗で、予定していた機能を開発できるか？を確認
    - ◇ 工数的に厳しくなった場合には、代替手段を検討
  - 工数の見積もりは、個人によって傾向が異なる
    - ◇ 自分自身の傾向を把握し、見積もりの精度を向上させることが重要

## ■ 作業スケジュールの検討

- 各機能の担当者の決定
  - 見積もり工数を参考に、期間内に作業が終わるか？を確認
  - サマーキャンプ期間中に開発可能な時間数:
    - ◇ 8/20(火) 2h(+4h), 21(水) 6.5h(+5h), 22(木) 6.5h(+5h), 25(金) 3h
    - ◇ **計:18h (32h)/人**
  - もしも工数的に終わらないのであれば、そもそものスケジュールに無理がある
- 各機能の開発順番の決定
- マイルストーンの検討
  - 複数RTCを用いた結合試験のタイミングなど
- マニュアルなどのドキュメント類を作成する時間も検討

## ■ 作業工数の見積もり方法

### ● プランニング・ポーカー

- 各メンバーが対象コンポーネント/機能の開発工数見積もりを実施
- 見積もった工数トランプで**一斉に提示**
  - ◇ 使用できる数字は、1, 2, 3, 5, 8, 10, 絵札の7種類
  - ◇ 全員の見積もり工数, 考えを確認するために, 見積もった工数を一斉に提示
    - ◇ 一部の人の意見が通りやすくなってしまいう事を避けるため
- 工数の議論, 検討
  - ◇ 提示された見積もり工数がばらついた場合は, 何故その工数と考えたのか理由を確認
    - ◇ 単純に平均値を計算するのではない
    - ◇ チームメンバ全員が納得できる工数をチームとしての見積もりとする
- 対象コンポーネント, 機能の細分化
  - ◇ 見積もり工数が10を超えた場合には, 可能であれば対象機能を詳細化/細分化し, 再度見積もりを実施
    - ◇ 規模が大きい機能は, 見積もり精度が悪くなるため

## ■ 検討結果を基に, 2日目15:00から, 今回開発するシステムを発表して頂きます