



# 検証機能および変更容易性を有する 離散事象シミュレーションフレームワーク

鹿島建設 ITソリューション部 天野 和洋 amano@kajima.com

## 開発における問題点

1. 離散事象に対応した汎用シミュレーション言語は数多く存在するが、専用言語のため習得が煩雑であり、様々な変更要求への迅速な対応が難しい。
2. 一定の性質を調べるためには統計分析やアニメーション機能を用いてパラメータスタディから推察するものの、網羅的な検証ではないため再現性の低いクリティカルなケースを見逃す可能性がある。

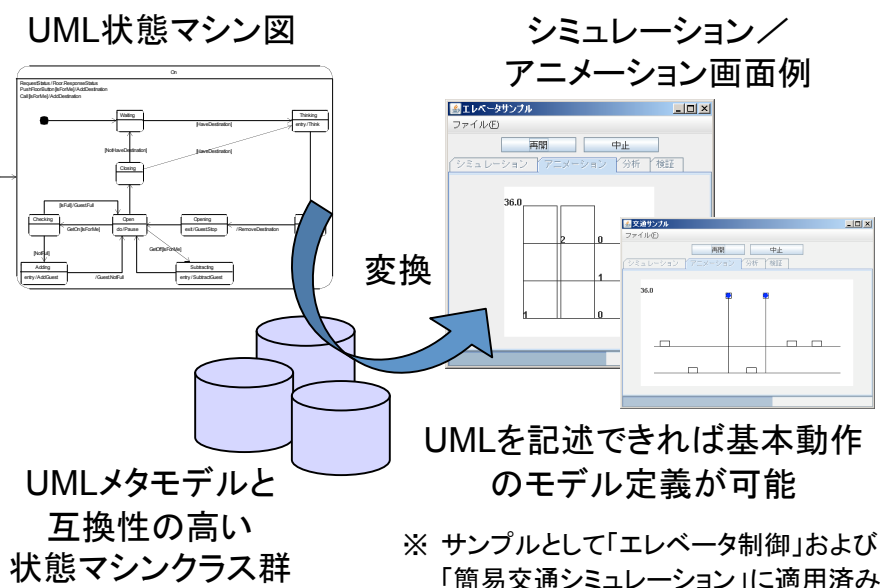
## 手法・ツールの適用による解決

1. 適用問題に対する変更容易性を、世界標準のモデリング言語であるUML状態マシン図のメタモデルとソフトウェアパターンを用いた設計によって実現した。
2. シミュレーション機能と網羅的な検証機能のシームレスな連携を、Javaプログラムを直接検証できる実装レベルの検証ツールJava PathFinderをフレームワークから制御することによって実装した。

## 主な機能と変更容易性

### 【フレームワークの主な機能】

- ▶ 適用問題を表現したモデルの入出力機能
  - ✓ UML状態マシン図によるモデルの定義
  - ✓ DAO(Data Access Object)パターンにより入出力ソース(XMLやDB等)を容易に変更可能
  - ✓ XMI(XML Metadata Interchange)サポート
- ▶ シミュレーション実行機能
  - ✓ 状態マシンに対応したイベント駆動型シミュレーションの実行
- ▶ 結果の分析・集計機能と網羅的な検証機能
  - ✓ Interpreterパターンにより分析内容・検証性質に関する文法表現の拡張性を確保
- ▶ アニメーション用結果出力機能



## 検証機能の実装

- ▶ モデル検査とは
  - ✓ システムの取り得る状態を網羅的に検査することによりシステムの性質を検証する手法
- ※ NASA Ames Research Centerが開発したJava PathFinder(JPF)は、Javaプログラムを直接検証可能なモデル検査ツール
- ※ JPFをフレームワークから制御することで検証機能を実装
- ※ フルスペックのシミュレータをそのまま検証するのは負担が大きすぎるため、非決定性に特化した抽象シミュレータを用意
- ※ MJI(Model Java Interface)を介して2つのシミュレータを同期することによって、シミュレーション機能と検証機能の完全にシームレスな連携を実現

