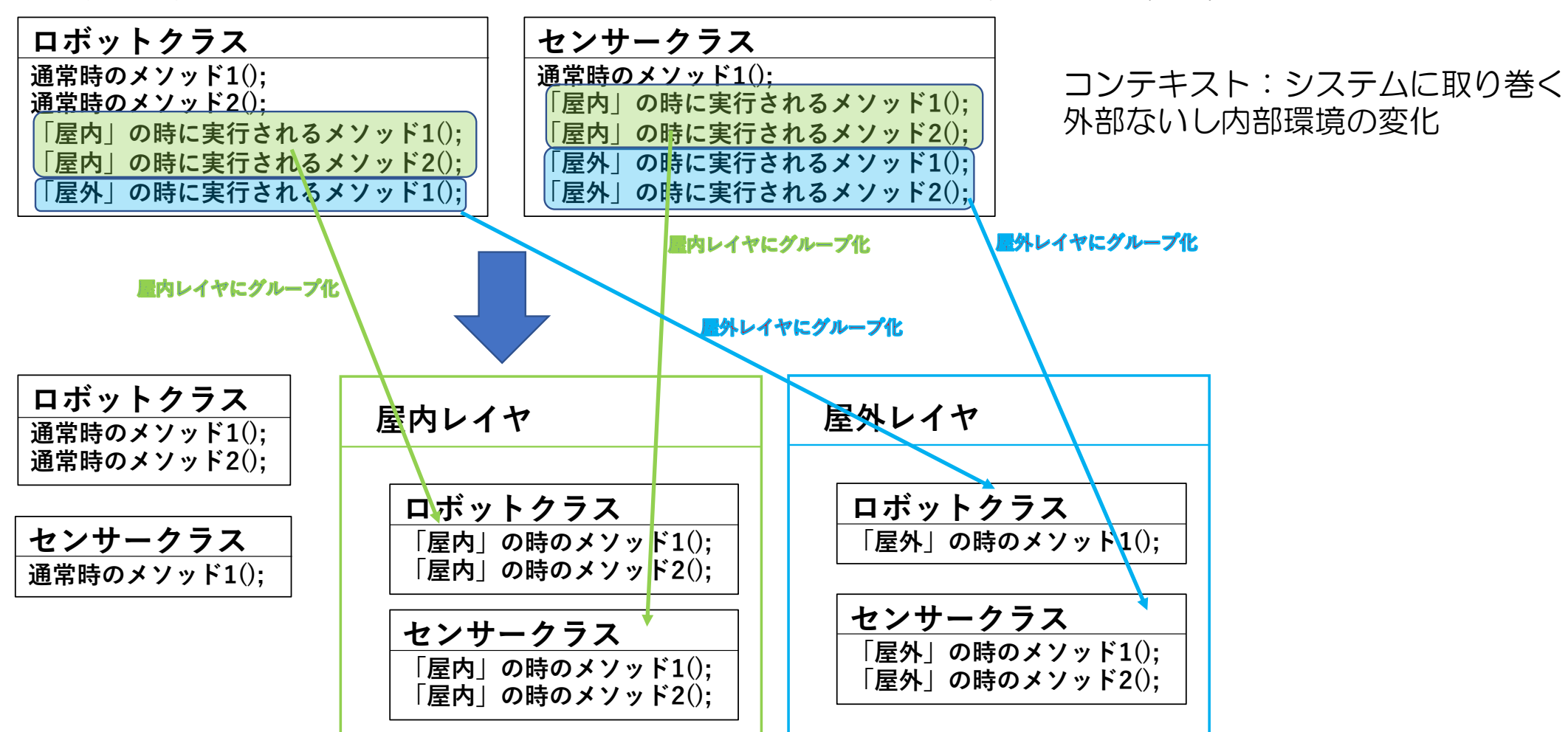


UMLクラス図、ステートマシン図を用いた文脈指向言語生成手法

文脈指向プログラミング

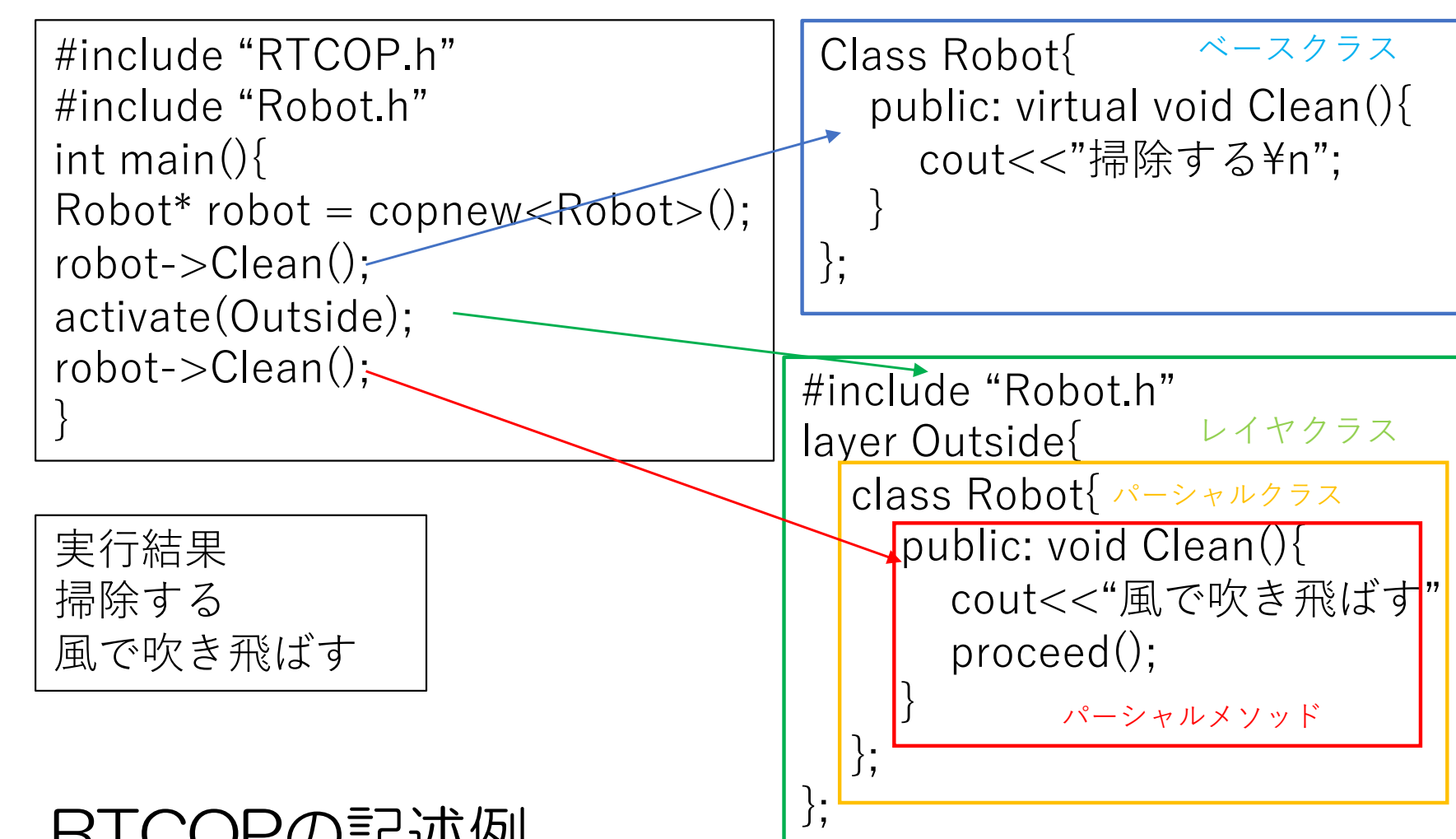
COP：Context-Oriented Programming
コンテキストに応じた振る舞いを表す
クラスやメソッドをレイヤとしてモジュール化
実行時のコンテキストに応じてレイヤを活性化



現在のコンテキストに動的に適應することを
可能にするプログラミング技術

RTCOP

組込みソフトウェアへのCOP適應を可能にするために
提案されたフレームワーク

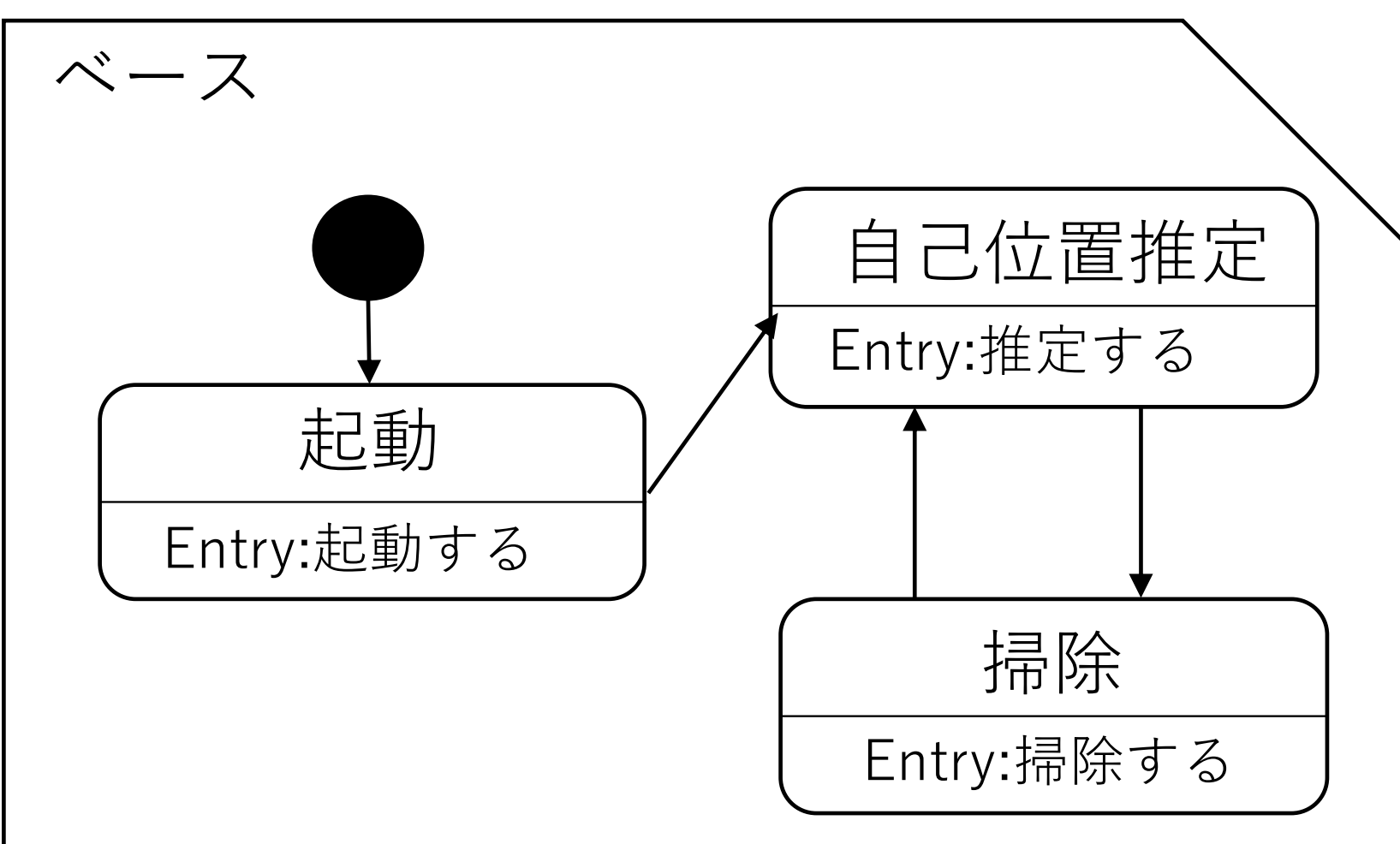


RTCOPの記述例

layerのスコープ内で記述された部分クラス、
部分メソッドをレイヤとしてモジュール化

レイヤ階層型ステートマシン図

基本の振る舞いを表すベースレイヤにCOPの機能を持
ったレイヤの振る舞いを差分のモデルとして重ねて
レイヤのモデルを表現

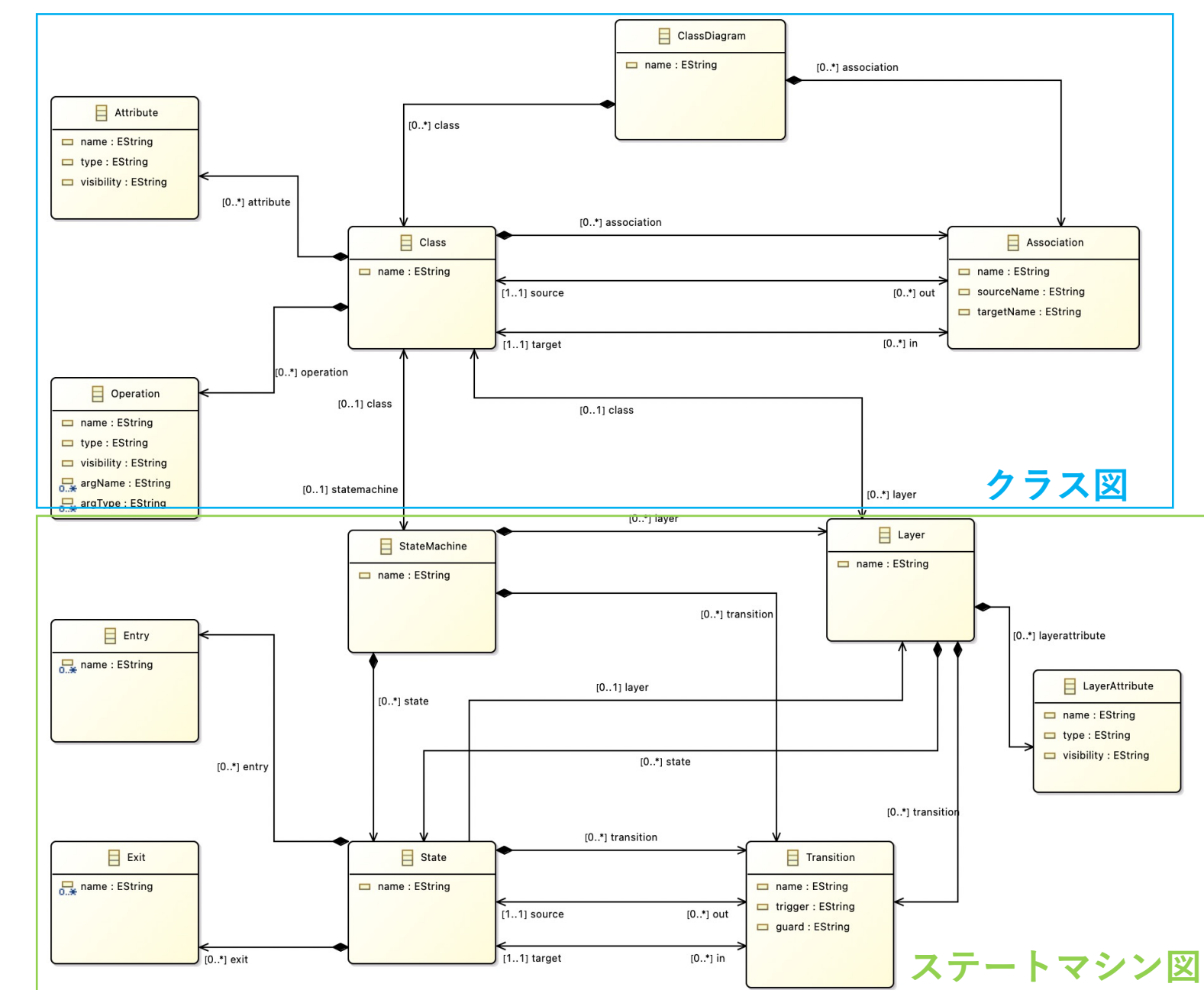


UMLステートマシン図をCOP用に拡張

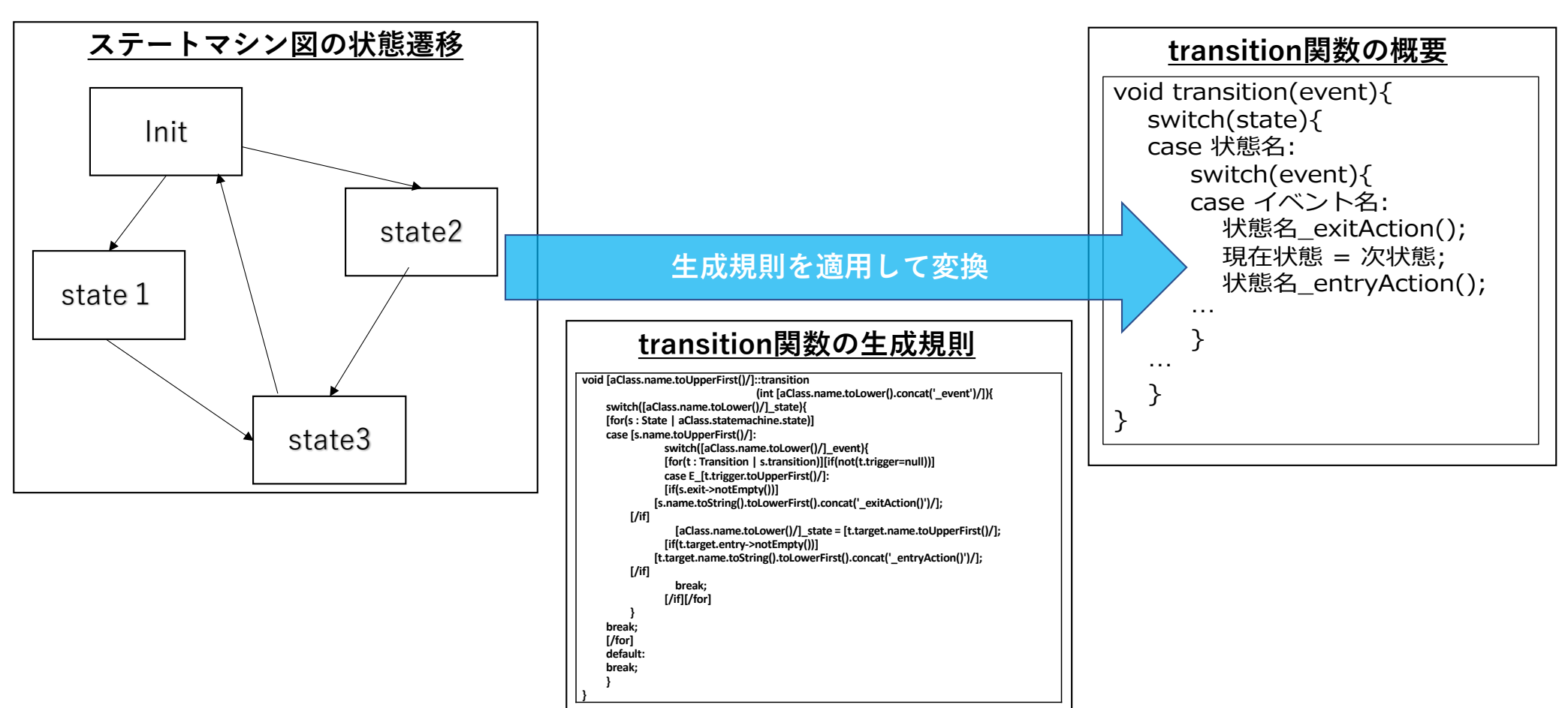
コード生成手法

UMLクラス図とレイヤ階層型ステートマシン図から
実行可能なCOP言語記述を生成する

Eclipse Siriusでメタモデルを定義し、モデルを作成



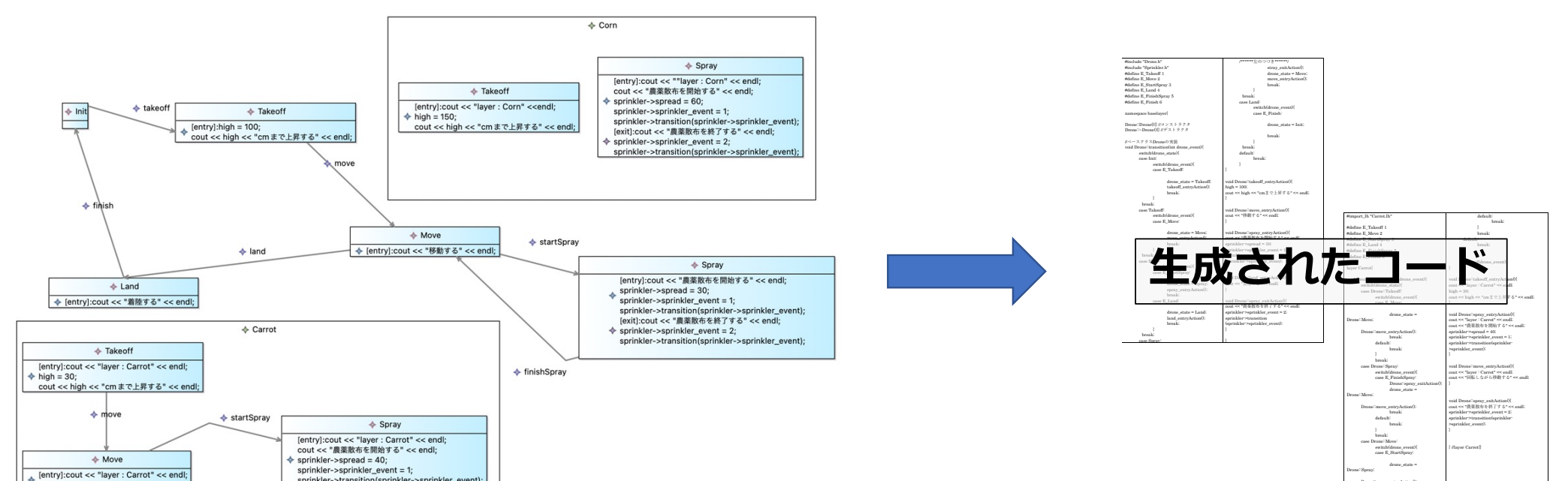
Acceleoで生成規則を定義し、モデルに適用



状態遷移⇒switch-case文で生成、状態名⇒列挙型で生成
イベント名⇒定数マクロで生成
入退場時処理⇒返り値なしの関数で生成など、、、

検証

飛行エリアで振る舞いを変わるドローンモデルに適用し、
RTCOPフレームワークを用いて実行可能



今後の課題

- メタモデルを詳細化⇒複雑な振る舞いのモデル化を可能に
- レイヤ活性化の優先度のセマンティクスを定める
- Javaによる中間コードを生成規則に定める
- モデル上でのシミュレーション実行、
実行状況の可視化による検証を可能にする