

## 論文の内容の要旨

論文題目      Applying Model Transformations to Construct  
Role-Based Systems

(役割に基づくシステム構築のためのモデル変換  
手法適用に関する研究)

氏      名      モンプラターンチャイ      スッパシット

現在のソフトウェア開発では、オブジェクト指向技術がソフトウェア工学の有力な方法として広く活用されている。オブジェクト指向ではオブジェクト同士が相互作用によってある目的に向かって動作するという形態をとり、現実の世界の物事を記述するのに向いている。しかし、オブジェクトの状態および動作は最初に定義されたものに固定されるため、それが環境の変化に応じて適応する状況には対応できないという問題点がある。適応的役割モデルは、役割の概念を導入することによって実世界の現象をシミュレートし、オブジェクトの環境適応を実現する新しい計算モデルである。このモデルに基づいてわれわれはすでに一つのフレームワークを提案しているが、そこではモデル要素が既知であるか、システム記述から直接的に識別することができるかと仮定して、Epsilon と呼ぶ役割モデルに基づくシステムを構築するための方法を提供した。しかし、特に問題領域が複雑な場合には、この構築フレームワークを適用する前にモデルの構成要素を正確に識別することが難しい。本研究はそのような実際的な状況で、いかにモデルを構築しシステム開発につなげるかというプロセスを確立したいという動機から出発している。

本研究の主要な成果は、役割モデル Epsilon に基づいたシステムの設計および構築のためのフレームワークを提案したことである。このフレームワークは、システムの要求仕様から戦略的かつ体系的に対象システムを表す Epsilon モデルを導出する方法論を提供する。モデル変換手法を

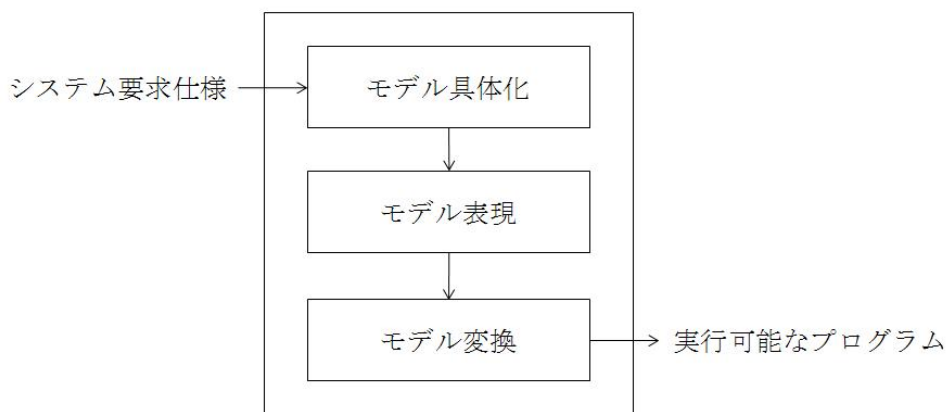


図1: フレームワークの概要

用いてモデルからシステムへの実装が自然に行えるため、実行可能なプログラムを自動生成するツールも開発している。本フレームワークを用いることにより、開発者は抽象的なレベルでのモデル構築に傾注することができ、さらに変換ツールによって実装レベルにモデルを変換することができる。図1に示すように、本フレームワークは主に以下の3つの連続した段階で構成されている。第一および第二段階は人間による作業であり、第三段階はツールにより自動化された作業である。

### 1) モデルの具体化

この段階では、i\*フレームワークという目標指向要求分析手法を改良したトップダウンの方法で抽象化レベルを洗練化することにより、システムの要求から段階的にモデル化を行うための手法を提供している。この手法によりシステム要求から **Epsilon** モデルの構成要素が抽出され、それが段階的に具体化されて対象問題領域のモデルが構成される。この段階のアウトプットは、モデル構成要素の集合で記述された **Epsilon** の概念モデルである。

### 2) モデル表現

この段階では **UML (Unified Modeling Language)** の図式表現を用いて、第一段階で得た要素から構成される **Epsilon** モデルを表現するための方法を提供する。この方法で各モデル要素が適切な **UML** の構成要素に写像され、対象システムを表現する **Epsilon** モデルのアーキテクチャとして **UML** 図に組み込まれる。この段階のアウトプットは、**UML** 図として表された **Epsilon** のアーキテクチャモデルである。

### 3) モデル変換

この段階では、既存の **UML** 描画ツールと本研究で開発した変換ツールを用いることにより、第二段階で得た **UML** 図を **Java** プログラムのソースコードに変換する。**UML** 図はまず、**UML** ツールによって中間表現に変換され、さらに開発された変換ツールで **Java** プログラムに変換される。この変換過程は、以下の3つの内部プロセスによって構成されている。

- ・前処理: **UML** ツールで生成された中間生成物を別の形式に変換する。

- ・ モデル変換 : Epsilon モデルを表す前処理された中間生成物を, Java モデルを表す別の形式に変換する.
- ・ 後処理 : モデル変換された形式を, Java プログラムのソースコードに変換する.

第一と第三のプロセスは Java で実装した. モデル変換プロセスには, **Model-Driven Engineering** : モデル駆動工学 (MDE) によるモデル変換技術を採用し, モデル変換言語 ATL により実装した. このツールで生成された Java プログラムには, これまでのモデルでは明示的に指定できないビジネスロジックを実現するためのコードを追加する必要がある. 本フレームワークではそのためのガイドラインも提供する.

本フレームワークの説明には, 会社組織の社員と管理者という役割で構成される具体的な状況を例題として用いている. それとは別に, より実際的なシステム開発における本フレームワークの適用可能性と有効性を評価するため, 事例研究を行った. 人工知能の分野から交通渋滞監視システムを事例として取り上げ, このシステムを設計構築するために本フレームワークを適用した経験を報告している. この適用経験とシステムの実行に関する実験により, 本フレームワークが複雑で現実的なシステム開発にも適用できることを示した.

先行研究として提案した既存のフレームワーク, およびマルチエージェントシステムや自己適応システムなどの役割に基づくシステムと類似するシステム開発のための他のアプローチとの比較も議論している. 本研究で提案したアプローチは役割モデルの概念を採用し, それに基づいたシステム開発に直接的な手段を提供しているので, 役割モデルが適合する問題領域に対しては優位性があることが示された. 提案手法の形式化や, フレームワークの全般的な自動化などは, 興味深い今後の課題としている.