

【講座シリーズ担当】

・・・1・・・

[講座シリーズリーダー]

田原 康之

電気通信大学 大学院情報システム学研究科 准教授

研究分野：ソフトウェア工学、特に形式手法、要求工学

・・・2・・・

田辺 良則

国立情報学研究所 特任准教授

研究分野：ソフトウェア検証、様相論理

・・・3・・・

Cyrille Artho

独立行政法人 産業技術総合研究所 情報セキュリティ研究センター 研究員

研究分野：形式検証技術

・・・4・・・

橋本 祐介

日本電気株式会社 サービスプラットフォーム研究所

研究分野：

・・・5・・・

加瀬 直樹

株式会社東芝 ソフトウェア技術センター

研究分野：ソフトウェアテスト技術

## 【本講座コースの基本思想・特徴】

本講座シリーズでは、実装工程、すなわちコーディング、テスト、およびデバッグ作業に関する技術を学習します。トップエスイーは、ソフトウェア開発プロセスのうち、分析・設計を中心とした、上流工程に重点を置いています。トップエスイーでは、モデリング能力の向上により、プロセスを通じて高品質な成果物を効率的に開発できるソフトウェア技術者の育成を目指していますが、モデリング能力は主として上流工程で重要となるからです。一方実装工程では、最終成果物であるプログラムを扱いますので、モデリング能力は不要に思えます。しかし、プログラムは最も抽象度の低いモデルです。より抽象的な他のモデルとの整合性をとり、プログラムそのものの品質を高めるためには、やはり実装工程でもモデリング能力は重要です。このような見地から、本講座シリーズでは、モデリング技術としての実装工程の手法やツールを学習します。プログラムをモデルとして捉える方向として、次のようなものがあります。すなわち、プログラムのテストをテストケース・テストスイート、およびテスト計画などさまざまな観点から捉えたテストモデル、プログラムの実行による変数値の変化の状態遷移モデル、およびプログラムの実行前・実行中・実行後のそれぞれにおいて成立すべき性質を表現する「設計に基づく契約 (Design by Contract, DbC)」モデルです。これらのモデルは、全て理論的背景に基づいていて、高品質なプログラムを効率的に実現することを目的として利用されており、これらを扱うツールも近年充実してきていますので、トップエスイーで学習するのに最適なものです。

そこで本講座シリーズでは、それぞれのモデルに対応して、テストモデル検証、実装モデル検証、およびプログラム解析の3講座を用意しました。まずテストモデル検証講座につきましては、アドホックに実施されがちなテストモデル検証作業に対しまして、直交表やテスト駆動開発プロセスといった、体系的なテストモデルを習得し、モデルに基づいてテスト作業を効率的に実施するためのツールを使いこなせるように実習を行います。実装モデル検証講座では、Java プログラムを状態遷移モデルとして扱い、実行時に遷移し得る状態を網羅的に検査することにより、期待される性質を検証するツールである、Java PathFinder (JPF) の習得を行います。さらにプログラム解析講座で

は、やはり Java プログラムの DbC モデルを記述する Java Modeling Language (JML) を扱うツールの実習により、プログラムの正しさの確認を行う手法を習得します。

以上のように本講座シリーズは、ツール実習を通じて、実装工程に必要なモデリング能力を習得することにより、高品質なプログラムの開発を効率的に進めることができるソフトウェア技術者の育成を目的としています。

## 【該当講座】

- ・ 実装モデル検証 [ 田辺 良則 ・Cyrille Artho ]
- ・ プログラム解析 [ 橋本 祐介 ]
- ・ テスティング [ 加瀬 直樹 ]

## 【演習のひとつま】

### 具体的な Java マルチスレッドプログラムに対し、 デッドロックと活性違反の有無を検証する(モデル検査による検証)

Java 言語の大きな特徴の 1 つが、マルチスレッド機構による並行プログラム記述が可能な点です。つまり、1 本の制御の流れに従って動作するスレッドを、同時に複数個動作させることにより、並列に計算を実行でき、また同時に起こる複数の外部の活動を制御できます。しかし、並行プログラムについては、非決定性や資源共有の問題が発生します。非決定性とは、同一の入力に対して、異なる動作結果が生じうることです。これは、動作単位の実行順序が異なることがあるために起こります。また複数のスレッドが 1 つの資源を共有する際には、一貫性を保つことが重要です。このために相互排除機構を使用することになりますが、複数の資源を奪い合って処理が止まってしまうデッドロックや、一部のスレッドが資源を占有して他のスレッドが資源を利用できなくなる活性違反などの不具合が発生しえます。

実際のプログラムが、デッドロックや活性違反なしに、一貫した資源の利用を必ず行うかどうかを検証することは、従来のテスト手法では困難でした。JPF のようなモデル検査ツールを利用することにより、厳密かつ自動的な検証が可能になります。

本演習では、1 車線しかない 1 本の橋に、両方向から自動車が通行する状況のシミュレーションプログラムや、複数のユーザが複数の DVD レコーダを用いたコピーを行うプログラムを例題として用います。これらのプログラムに JPF を適用することで、デッドロックや活性違反が発生しないことを検証します。さらに前者では、橋に両方向から同時に自動車が侵入しないこと、また後者では各 DVD レコーダは同時に 1 人のユーザしか使用しないことの検証を実施します。さらに、これらの問題にテスト・デバッグ手法を適用する試みと比較することにより、モデル検査手法の得失を議論します。