

【講座シリーズ担当】

・・・

[講座シリーズリーダー]

吉岡 信和

国立情報学研究所 准教授

研究分野：ソフトウェア工学、エージェント技術、セキュリティパターン

・・・

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, モデル検査, 様相論理, 形式検証技術

・・・

田原 康之

国立大学法人 電気通信大学 大学院情報システム学研究科

社会知能情報学専攻 システム設計基礎学講座 准教授

研究分野：ソフトウェア工学、形式検証

・・・

磯部 祥尚

産業技術総合研究所

情報技術研究部門ミドルウェア基礎研究グループ 主任研究員

研究分野：並行システム検証, プロセス代数, 定理証明器, モデル検査器

・・・

長谷川 哲夫

株式会社 東芝 ソフトウェア技術センター

研究分野：ソフトウェア工学、エージェント技術、分散処理技術

## 【本講座コースの基本思想・特徴】

この講座シリーズでは、ソフトウェアの振る舞いをどうモデル化し、検証するかについて学習します。モデル化には、検証する性質に応じて、UML の状態チャートやタイミング図、GSP などの並行プロセス記述を用います。そして、そのモデルの検証方法として、最近、急速に実用化が期待されているモデル検査ツールを用います。なお、本稿において「検証」という用語は、デッドロックが起こらないなど、ソフトウェアが与えられた性質を満たすかどうかを、可能な限り厳密に確認する、という意味で使用します。また「モデル検査」とは、ソフトウェアなどのモデルの振る舞いに関して全ての状態を探索することにより、数学的に厳密な検証を実現する1つの技術を指します。したがって、一般的な意味でのソフトウェアの「検査」、すなわち「テスト」とは異なることに注意して下さい。

このようにモデル検査ツールは、テストと異なり数学的に完全性を示すことが可能です。さらに、証明ツールと異なり状態探索のために特別な操作・指定を行う必要がなく、数学の専門家以外が使える検証ツールとして注目されています。しかし、モデル検査ツールを使いこなすためには、ソフトウェアエンジニアの重要な能力であるモデル化能力や、問題領域をとらえ、解決手段を発見する能力が必要となります。

本講座シリーズでは、単にモデル検査ツールの使い方を学ぶのではなく、ツールを通してこのようなエンジニアの能力を育てることを目的としています。具体的には、UML などで記述した設計モデルを、検査したい項目から、その構造やデータ、制御フローに着目した検証部分の切り出しや抽象化の方法も学びます。このような検証部分の切り出しや抽象化は、検証のための時間やメモリを大幅に軽減し、状態爆発による検証不能を避けるために重要な作業です。さらに、モデル検査では、取りうる状態を網羅するために外部環境を含め全ての状況(閉じたモデル)を規定する必要があります。検査したい項目から、このようなモデルの作成や検証する具体的な性質をブレイクダウンする過程を通じて、何を検証したいのかという問題領域を正しくとらえる能力や、そのための解決手段を発見する能力を養うのです。そして、検証に必要な数学的背景知識も単に大学でのオートマトンの授業と異なり、具体的な検証方法との関係とともに工学のための活きたサイエンスとして学びます。

最終的には、各講座を通して、たくさんのツールを使うことで、検証内容に応じたツールの使い分けができるようになるようになります。

## 【該当講座】

- ・ 設計モデル検証（基礎） [ 吉岡 信和 ・ 田原 康之 ・ 田辺 良則 ]
- ・ 設計モデル検証（応用） [ 吉岡 信和 ・ 田原 康之 ]
- ・ 並行システムの検証と実装 [ 磯部 祥尚 ]
- ・ 性能モデル検証 [ 長谷川 哲夫 ]

## 【演習のひとこま】

### バスを介したオーディオ機器制御プロトコル

近年の高性能な家電では、従来の個々の機能の組み込みモジュールのリアルタイム性に加え、バスを介した様々なモジュールを正しく組み合わせ連携させる必要があります。この演習では、実問題である高性能オーディオを例題に、機能モジュールを組み合わせても時間制約を満たし、正しくサービスが動作することを確認します。この例題の難しさは、振舞いと

リアルタイム制約を同時に考慮する必要があり、かつ、複数の送信者、受信者など多数のモジュールの処理タイミングのズレが複雑に影響しあっている点にあります。

これに対して、時間制約を考慮したモデル検査ツールである UPPAAL を通して、具体的に時間制約をどうモデル化するか、そもそも何が制約なのかを考え、その背景にある時相論理、時間オートマトンなどを、実習を通して学びます。

