設計モデル検証(基礎編)

平成２４年度シラバス

2012 年 1 月 13 日

国立情報学研究所

トップエスイープロジェクト

代表者　本位田　真一

1. 講座名

設計モデル検証(基礎編)

2. 担当者

吉岡　信和、田辺　良則、宇佐美 雅紀、早川 昌志、長久 勝

3. 本講座の目的

本講座では、ネットワーク家電の制御ソフトウェアを題材とした産業ソフトウェアの設計モデル検証問題を扱う。近年ますます大規模化・複雑化するソフトウェアの設計において、その動作の正しさを保証することがより困難になっているが、設計モデルの正しさを自動的に検証するツールSPINを使用し、実際のシステム開発に適用する方法を習得する。しかし、SPINを使いこなすには多くの難しさがあり、普及への障害となっている。そこで本講座では、標準のモデル記述言語UMLを全面的に採用し、実際に組込ロボット上でのソフトウェア開発を通して、ソフトウェア設計検証の難しさを体感し、SPINツールの適用における難しさを解決するための検証プロセス、およびノウハウを習得する。また、グループ討議を通じた例題演習により、議論を通してモデル検査技術の理解を深め、検証プロセスの実用的ノウハウを体得できる効果が期待できる。

4. 本講座のオリジナリティ

SPIN などのモデル検査ツールを使用した設計モデル検証については、従来多くの同様な講座が開講されている。しかしそれらの講座にはさまざまな問題点があったため、受講生が習得した内容を速やかに開発現場で適用するための障害となっていた。本講座では、それらの問題点を解消し、本講座受講後 SPIN ツールを速やかに開発現場で適用できるように配慮している。**表 1** に、既存の講座の問題点と、本講座における解を示す。

表 1 既存の講座の問題点と、本講座における解

既存の講座の問題点	本講座における解
モデル検査ツール独自の言語・記法のみを使用するため、そのような言語や記法を使用しない、現場での開発工程に組み入れるのが難しい	UML を全面的に採用し、UML 設計モデルの検証、およびバグ修正までサポートするため、現実的な開発プロセス(特に MDA)へのシームレスな組入れが可能
モデル検査手法が解決する、現状のソフトウェア開発における問題点を体感できないため、モデル検査の有用性を実感するのが難しい	同じ情報(UML設計モデルとJava実装)を用いて、テスト・デバッグとモデル検査の両方を実習するため、従来のテスト・デバッグに比べての、モデル検査の有用性を体感可能
組込みシステムをターゲットとした講座であっても、高々PC 上での実習しか行わないため、組込みシステムに対する有効性を実感するのが難しい	組込み機器の事例としてライントレースロボットを使用した実習を行うため、組込みシステムに対する有効性を体感可能
現実的な問題をそのまま使用せず、抽象化・簡略化されたトイ例題を題材にしているため、実際のシステム開発に対する有用性を実感するのが難しい	トイ例題を用いたチュートリアル、トイ例題に対するUML設計モデルを用いた検証プロセス、そして現実的な例題に対する UML 設計モデルを用いた検証プロセス、といった段階的な講座進行となっているため、実際のシステム開発に対する有用性を実感できるまで、無理のない習得が可能

5. 本講座で扱う難しさ

近年、家電の制御ソフトウェアは大規模化、複雑化が進んでいる。特に、複数の家電機器がネットワークに接続され、相互に連携して動作することで 1 つの機能を実現する、ネットワーク家電市場が急速に立ち上がりつつある。従来の家電機器の制御ソフトウェアは単体で動作するように設計されていたのに対して、ネットワーク家電では、他の機器との連携動作のためのプロトコルなどが必要になるため、制御ソフトウェアが非常に複雑になる。さらに、ホームネットワークでは、動的に変更される接続相手の識別や、不安定なネットワーク環境などを考慮する必要がある。このような環境に対して高信頼でかつ安全なソフトウェア設計を実施するために、全動作パターンを人手で検証する従来型の方法論は膨大な工数を必要とし、もはや現実的手段ではない。

制御ソフトウェアの複雑さの例を図 1 に示す。図 1 は、各機器の振る舞いが複雑であるためにシステム全体の動作を把握することが困難であることを示す例である。今、3 台のネットワーク家電機器、TV、DVD レコーダ、HD レコーダ一体型 TV から構成されたホームネットワークがあり、2 人のユーザがホームネットワーク上の異なる機器に対して操作を行っている。1 人は、DVD レコーダを操作して、HD レコーダ一体型 TV から録画済みの番組をコピーし、DVD ディスクに記録している。もう 1 人は、HD レコーダ一体型 TV に予約録画を行っている。もし、HD レコーダ一体型 TV がコピー処理中に予約録画が開始された場合、HD レコーダ一体型 TV に組み込まれた 1 つのメディアプロセッサを 2 つの処理が奪い合うという、資源獲得の競合が発生しうる。HD レコーダ一体型 TV は、複数の資源獲得要求とそれに伴う処理を適切な順序で誤りなく実行するように設計されなければならない。

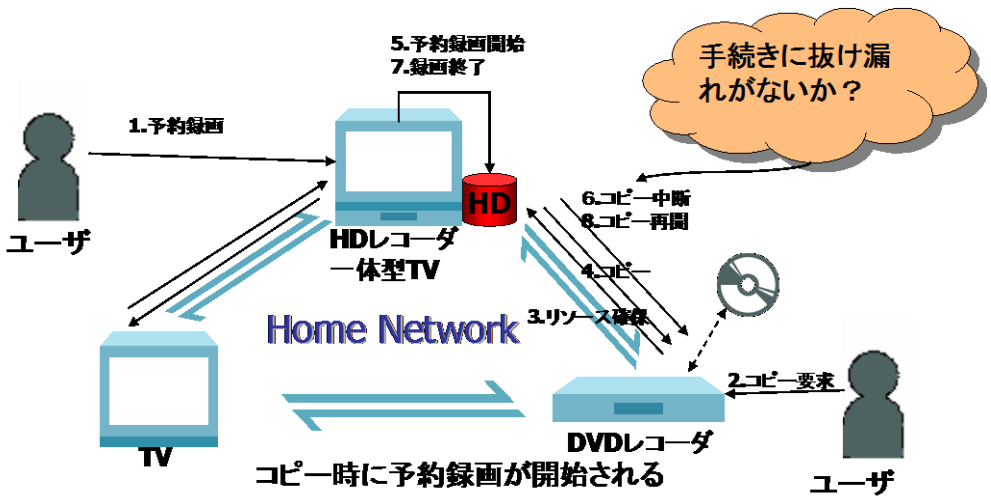

図 1　ネットワーク家電の複雑さの例

6. 本講座で習得する技術

将来のネットワーク家電では、接続する機器や機器間で交換する情報の数・種類が従来型の家電機器よりも大幅に増加するため、その振る舞いの設計が大規模化、複雑化し、人手によるレビュー等では設計誤りの発見が非常に困難になってきている。モデル検査は、システム上で起こり得る状態を網羅的に調べることによって設計誤りを発見する自動検証手法の一種である。近年、高品質な検証ツールが開発され利用されて、モデル検査を実用システムに適用した多数の成功事例が報告されている。

しかし、大規模ソフトウェア開発にモデル検査を利用する場合には、検証の目的や範囲を明確化した上で、戦略的な検証実施計画を立てることが求められる。なぜならば、依然として設計検証は時間と手間とを要する非常にコストの高い活動であるからである。検証すべき設計仕様の適切な部分が適切な条件下で検証されなければ、その活動に膨大な時間をかけたとしても検証結果は意味のないものになってしまう。したがって、ソフトウェアの設計検証には、ソフトウェア開発プロセスと同様に、様々な角度から検証問題を捉えるという実践的なスキルと経験が求められる。

本講座では、設計検証に関わる活動をソフトウェアプロセスの 1 つと捉えた、検証プロセスについて学ぶ。検証プロセスは次の 5 つのステップから構成され（図 2)、各ステップで必要な具体的作業の順序と内容とを体系的に習得する。

(1) 検証目的を明確化する検証要求分析ステップ
(2) 制約や外部環境をモデル化する検証モデル設計ステップ
(3) モデル検査ツールの入力言語による検証モデルの実装ステップ
(4) モデル検査ツールによる検証ステップ
(5) 検証エラーを分析して、設計誤りの原因を究明する設計誤り発見ステップ

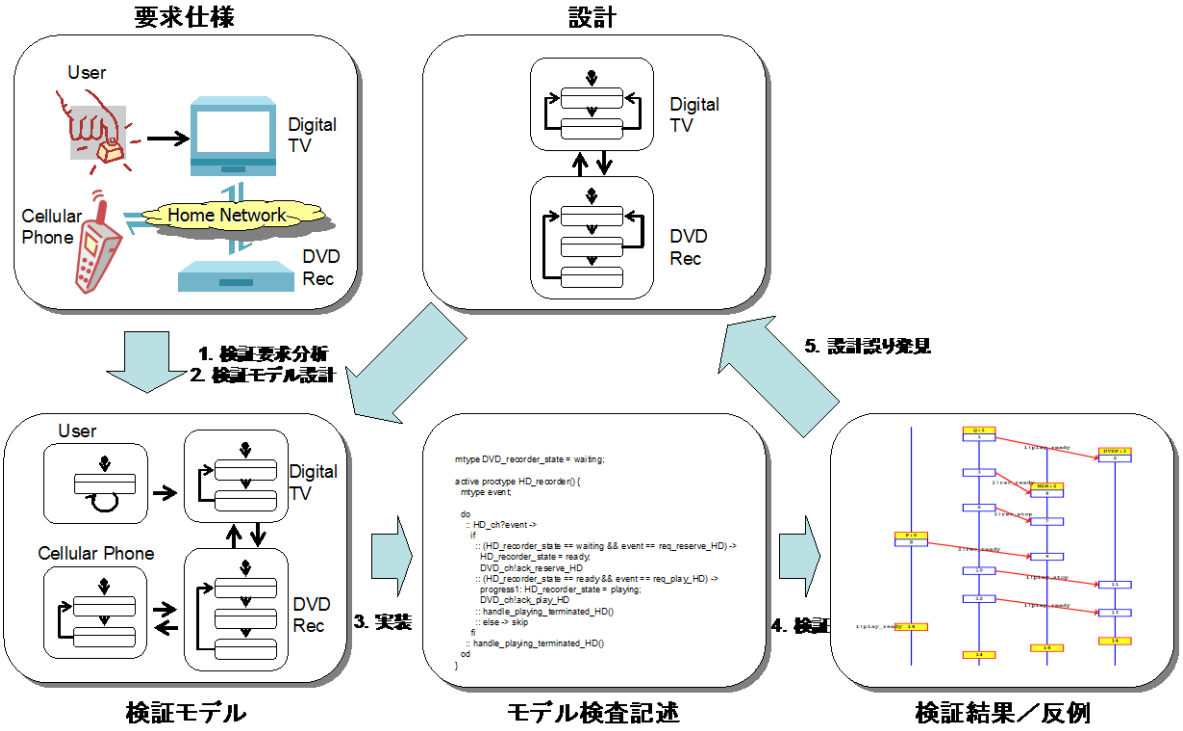

図 2　モデル検査を利用した設計検証プロセス

本講座では、具体的なモデル検査ツールとして、SPIN を使用する。SPIN の特徴は次の通りである。

- C や Java などのプログラミング言語に近い言語 Promela (Protocol/Process Meta Language)により、検証対象となるモデルの記述が可能
- 検証機能が豊富(シミュレーション、デッドロック検証、線形時相論理(Linear Temporal Logic, LTL)式検証など)
- iSpin ツールによる GUI サポート(メニューとボタンによる操作、反例トレースからのシーケンス図生成など)
- 実適用事例が豊富(火星探査機などの宇宙システム、交換機システム、治水システムなど)
- オープンソースライセンス

7. 前提知識

本講座の受講生は、以下の項目を習得済みであることが望ましい。

- 以下に関する基礎理論
  - 並行プログラム：非決定性、資源共有・排他制御、通信、安全性、生存性、公平性
  - オートマトン：意味論(形式言語理論)、通信
  - 時相論理：構文、意味論、パターン(安全性、生存性など)
- モデル検査技術の原理
  - 時相論理式のオートマトンへの変換
  - 受理言語探索
  - BDD (Binary Decision Tree)
  - On-the-fly 手法
  - 部分順序による状態数削減手法
- UML：特にステートチャートの意味論
- プログラム抽象解釈手法

なお、これらの項目は、全て「基礎理論」講座で習得可能である。

8. 講義計画

・ 概要

第1回：導入：並行分散システムの設計の難しさ、ＳＰＩＮ入門（１）

第2回：基礎 - ＳＰＩＮ入門（２）： デッドロック、進行性、公平性

第3回：基礎 - ＳＰＩＮ入門（３）： Naver Claim、ω-Acceptance

第4回：基礎 - ＳＰＩＮ入門（４）： 時相論理、iSpin

第5回：設計モデルの検証（１）：検証プロセス概要、検証要求分析

第6回：設計モデルの検証（２）：検証対象モデルの設計、検証記述、シミュレーション

第7回：設計モデルの検証（３）：検証、反例分析

第8回：設計モデルの検証（４）：設計モデルの修正

第9回：ライントレースロボットを使った事例と演習（１）

第10回：ライントレースロボットを使った事例と演習（２）

第11回：ライントレースロボットを使った事例と演習（３）

第12回：応用演習（１）

第13回：応用演習(２)

第14回：応用演習(３)

第15回：発表・議論とまとめ

・ 詳細

第1回：導入、SPINチュートリアル

導入(座学)

- 検証とは?モデル検査とは?(復習)テスト・デバッグとの違い
- なぜ検証が必要か?
- なぜ検証プロセスが必要か?

SPINチュートリアル

- Promelaの構文
- Spinの実行の仕方(個人実習)
- 検証以外の機能(構文チェック、シミュレーションなど)(個人実習)
- 検証機能(個人実習)
  - ✧ 検証式の記述の仕方
  - ✧ 検証のやり方
  - ✧ 出力の読み方
- 演習：トイ例題を用いた検証（個人実習、宿題）

第2回：基礎 - ＳＰＩＮ入門（２）： デッドロック、進行性、公平性

- SPINにおける検証の詳細：デッドロック、進行性
- 並行性に関する仮定:強公平性と弱公平性
- 演習：トイ例題を用いた検証（個人実習、宿題）

第3回：基礎 - ＳＰＩＮ入門（３）： Naver Claim、ω-Acceptance

- SPINによる性質の検証：Naver Claim
- モデル検査の動作原理：ω-Acceptance
- 演習：トイ例題を用いた検証（個人実習、宿題）

第4回：基礎 - ＳＰＩＮ入門（４）： 時相論理、iSpin

- 時相論理による性質の記述
- iSpinを使った検証の効率化
- 演習：トイ例題を用いた検証（個人実習、宿題）

第5回：設計モデルの検証（１）、検証プロセス概要、検証要求分析

- 検証プロセスとは何か？
- 検証の要求の分析と検証対象モデルの設計方針
- 検証モデルの完全化、外部環境の考慮
- 演習：現実的な例題(基本例題)を用いた仕様の検討(グループ実習、宿題)

第6回：設計モデルの検証（２）：検証対象モデルの設計、検証記述、シミュレーション

- 検証対象モデルのセマンティクスの確認と設計
- 検証記述（Promela）への変換
- 簡単なモデルで動作を確認（個人実習）
- 検証対象モデルから検証モデルへの変換(個人実習)
  - ✧ 抽象化、検証記述のノウハウを習得
- 検証記述の動作確認、シミュレーションと
- 演習:基本例題に対する検証対象モデルの設計と検証記述（グループ実習、宿題）

第7回：設計モデルの検証（３）：検証、反例分析

- 検証の実施と反例分析(個人実習)
- 不都合の原因の分析
- 演習：基本例題を用いた検証、反例分析（グループ実習、宿題）

第8回：設計モデルの検証（４）：設計モデルの修正

- 検証と反例分析の続き(個人実習)
  - ✧ 検証式記述ノウハウ、反例分析ノウハウを習得
- 原因の追究と設計モデルのバグの指摘まで(個人実習)
- 演習：基本例題を用いた反例分析とモデルの修正（グループ実習、宿題）

第9回：ライントレースロボットを使った事例と演習（１）

- 対象ドメイン「ETロボコン」の概要説明
- 演習：状態遷移の設計とモデル検査項目の作成(グループ実習、宿題)

第10回：ライントレースロボットを使った事例と演習（２）

- モデル検査項目の検討とPromelaへの変換

第 11 回：ライントレースロボットを使った事例と演習（３）

- Spin によるモデル検査
- モデルに準じた実装と動作確認

第 12 回：応用演習（１）

- 演習：応用課題の検証対象モデルの作成と検証記述への変換（グループ実習、宿題）

第 13 回：応用演習（２）

- 演習：応用課題の検証対象モデルの作成と検証記述への変換（グループ実習、宿題）

第 14 回：応用演習（３）

- 演習：応用課題の検証の実施と設計モデルの修正（グループ実習、宿題）

第 15 回：発表・議論とまとめ

- グループ発表：応用課題の検討結果の発表　および討議
- 最後に講座全体のまとめ(座学)
  - 習得内容の総括
  - 関連技術
  - 次に履修が期待される講座
- 宿題：講座で習得できたこと、SPIN、および検証プロセスの有用性の評価

9. 教育効果

本講座を受講することにより、実際のシステム開発、特に組込みソフトウェアの開発に適用可能な、設計モデル検証プロセスを習得できる。その結果、開発現場において、モデル検査ツールを活用することにより、信頼性の高いシステムを、効率的に開発することができるようになる。

10. 使用ツール

SPIN：モデル検査ツール

- 使用する上での難しさ
  - モデル記述言語 Promela、および LTL 検証式による記述が難しい
  - 検証の結果がエラーの場合、反例を設計モデル修正に反映するのが難しい
- 使用上必要なノウハウ
  - 検証モデル記述ノウハウ
    - 状態の表現
    - 遷移の表現
    - 通信の表現
  - 検証式記述ノウハウ
    - 検証項目の種類(安全性、生存性など)
    - 検証式パターン
  - 検証モデル抽象化
    - 内部動作の除去
    - データ領域の有限化
  - 検証効率化
    - 部分順序法
    - 状態削減
  - 反例分析
    - 変数値追跡
    - メッセージ追跡
- 選択理由、実用性：5 章に記載の SPIN の特徴を参照

11. 実験及び演習

2〜3 名程度の少人数で構成されたグループ単位で演習課題の実験と議論を行い、設計検証の難しさを実際に体感しながら、設計検証プロセスを体得する。まず、検証を行っていないネットワーク家電向けソフトウェア設計の誤りを発見する。次に、SPIN を利用してソフトウェア設計の検証を行い、上記で発見した設計誤りを検出できることを確認する。グループ内で SPIN を適用した結果について比較評価を行い、検証プロセスの有用性と適用性を議論する。議論を通してモデル検査技術の理解を深める。

12. ライントレースロボットを使った実習

4 名のグループに 1 台のロボット（図 3）を与え、ロボットの状態遷移について、モデル化・モデル検査・実装のプロセスを体験してもらう。まず、開発対象ロボットに対する要求を満たすよう、ロボットの状態遷移を設計する。次に要求充足を検証するために検査項目を考える。設計した状態遷移を Promela に、検査項目を LTL 式に、それぞれ書き下し、Spin による検証を行う。最後に、検証済みの安全なモデルを実装に書き下し、実機の挙動を確認する。一連の作業はグループで相談しながら進めることとし、各自が気付いたことを共有する。

図 3　ライントレースロボット

13. 評価

演習課題レポート、プレゼン発表、出席日数を総合して評価する。

14. 教科書/参考書

- 吉岡ほか, “SPIN による設計モデル検証”, 近代科学社, 2008.
  本講座の教科書である。予習・復習に利用してほしい。

- 中島震, “SPIN モデル検査―検証モデリング技法”, 近代科学社, 2008.

- Mordechai Ben-Ari(著), 中島震, 谷津弘一, 野中哲, 足立太郎(訳), “SPIN モデル検査入門”, オーム社, 2010.

- B. Berard et al, “Systems and Software Verification: Model-Checking Techniques and Tools,” Springer Verlag, 2001.
  モデル検査手法を利用したソフトウェア検証について、入門から実践までの一通りが述べられており、この講義に最適である。

- E. M. Clarke e. al, “Model Checking,” MIT Press, 2000.
  モデル検査手法の原理を理論的に述べたものであり、上記教科書を補うのに最適である。

- G. J. Holzmann, “The SPIN Model Checker: Primer and Reference Manual,” Pearson Educational, 2003.
  モデル検査ツール SPIN に関する理論的背景とツールの利用法が述べられており、SPIN を用いた演習に最適である。