実用化に向けたモデル検査適用手法の開発

Development of Method for Practical Application of Model Checking

池田　信之　　今村　紀子　　高田　沙都子
■ IKEDA Nobuyuki　■ IMAMURA Noriko　■ TAKADA Satoko

近年，ソフトウェアは社会インフラシステムや組込みシステムとして人々の生活により深くかかわるようになってきており，今まで以上に安全性を確保することが重要になっている。モデル検査は，そのための有望な技術として期待が高まっているが，一方で，企業におけるモデル検査の適用はまだ事例が少なく，ノウハウの蓄積が十分ではない。このため，モデル検査を有効利用するには，既存の開発プロセスへの組込みや教育の実施，定着化の仕組み作りなど実用化のための課題が存在する。

東芝はこれらの課題への対策として，状態モデル表記法を組み合わせた仕様化プロセスによる導入コストの削減や，専任のモデル検査チームの編成による作業の効率化とノウハウの蓄積を行っている。

Guaranteeing the security of software is becoming increasingly important as the embedding of software in industrial infrastructure systems continues to expand. This has led to growing expectations on model checking technology as a means of solving this issue. However, there is a lack of sufficient documentation on the successful use of this technology in real cases. In addition, for the successful application of model checking technology, it must be integrated into current development processes, the appropriate training must be conducted, and schemes for its continued usage must be established.

In response to these requirements, Toshiba has combined several behavior models to reduce initial costs and formed a specialized model-checking team to improve efficiency and accumulate know-how.

1　まえがき

近年，ソフトウェアは様々な場所に取り入れられ，人々の生活に大きくかかわってきており，信頼性や安全性を高めることが重要視されている。しかし，ソフトウェアが複雑化してきたことでこれまで以上に誤りが混入しやすくなり，そのうえ，人の手では仕様を網羅した検証が困難になってきている。また，システムの設計段階で仕様の誤りが混入すると，試験工程で動作確認をするまで発見できず，この場合，仕様作成にまでさかのぼって対応する必要があるため，多大な修正コストがかかる（**図1**(a)）。

このような問題を解決する手段として，形式的手法が期待されている。特に組込みシステムなど装置制御を主な目的とするソフトウェアでは，上流工程で不具合を早期発見する方法として，形式的手法の一つであるモデル検査が着目されている。モデル検査を実装したツールも公開され，組込みシステム開発では適用を検討する段階に入ってきた。

なお，モデル検査は仕様の段階で不具合がないかを検証するものであり，決してテスト工程に代わるものではないが，システムの不具合を早期に発見することができれば詳細設計や実装を行う前に修正が可能であり，後戻りも最小限に抑えることができ，コスト削減の効果が期待できる（図1(b)）。今日，開発期間は短期化する傾向にあり，今後このような手法は更に有用となることが予想される。

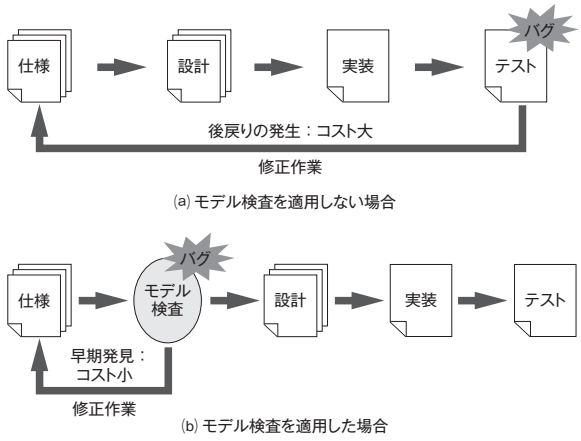

図1. 従来プロセスとモデル検査を適用したプロセスの修正コスト比較 — 従来工程では，テスト工程でバグが発見され後戻りのコストが大きかったが，モデル検査を適用すると，上流設計でバグを修正できるため修正コストが小さい。

Comparison of modification costs of two development processes

2　モデル検査の概要

2.1　手法の位置づけ

モデル検査手法は，形式的手法と呼ばれる技術の一つであ

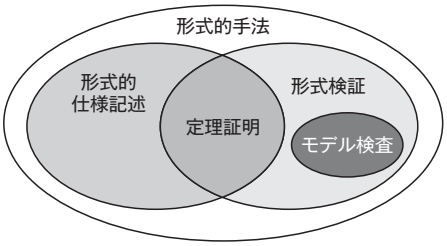

図2. モデル検査の位置づけ — モデル検査は，形式的手法のうち形式検証に含まれる技術である。

Position of model checking

る（**図2**）。形式的手法には，形式的仕様記述と形式検証という二つの大きな技術カテゴリーが存在する。形式的仕様記述はシステムの満たすべき仕様を数式により厳密に記述する技術であり，形式検証はシステムの性質を数学的に証明する技術である。形式検証には，数式で表現した仕様モデルから推論を積み重ねて証明する定理証明手法と，状態モデルなどのふるまいモデルを用いて性質を検査するモデル検査手法がある。定理証明手法は厳密解が得られるが，適用できる対象が限定的であり，証明に時間がかかる。一方，モデル検査手法は検査結果の完全性は保証されないものの，適用できる範囲が広く，検査時間が比較的かからないという利点がある。

モデル検査手法では，検査対象となるシステムが取りうるすべての状態を網羅的に探索することで，状態遷移系に対する不具合の有無を検証することができる。また，“到達性解析”や“進行性解析”が可能である。到達性解析では，状態空間にデッドロックや飢餓状態が存在するか否かを発見することができる。一方，進行性解析では，意図どおりの遷移がどの実行列でも存在しているかどうかを検証することができる。

2.2 作業フロー

モデル検査を実施する作業フローを**図3**に示す。モデル検査の入力は，システムのふるまいやシステムが動作する環境を記述した状態モデル（仕様）と，システムが満たすべき性質を定義した検査条件の二つである。これら二つを作成することで，モデル検査ツールによる検査や，必要に応じて動作シミュレーションが実行できる。もし検査の結果，状態モデルに問題があり検査条件を満たさない場合，モデル検査ツールから“反例”として，検査条件を満たさないシステム動作シーケンスの一例が出力される。検査を繰り返し，反例が出なくなるまで状態モデルを修正することで，最終的に正しい状態モデルが得られる。

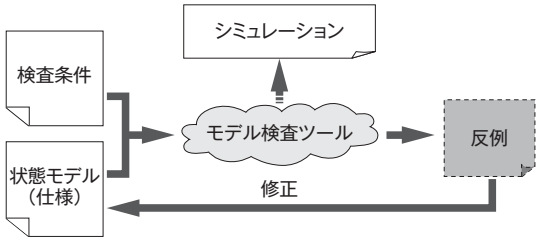

図3. モデル検査の作業フロー — 検査条件と状態モデルを作成した後，モデル検査を実施する。

Workflow of model checking

3 モデル検査適用の課題

モデル検査を実際の開発で利用しようとする場合，いくつかの課題に直面する。それらは，“検査技術自体の難しさ”に起因する課題(A)と，“従来開発とのすり合せの難しさ”に起因する課題(B)の二つに大きく分類することができる（**図4**）。

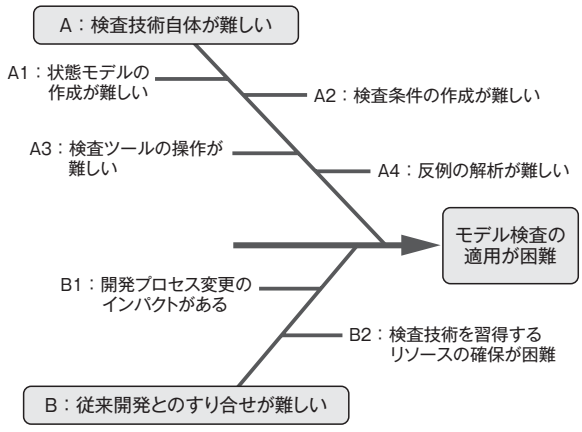

図4. モデル検査適用における課題の分析 — 課題は，検査技術自体の難しさに起因するものと，従来開発とのすり合せの難しさに起因するものに分けられる。

Tree analysis of issues involved in application of model checking

3.1 検査技術自体の難しさ

(1) 状態モデルの作成が難しい（A1）　モデル検査では，状態モデルをツールで機械的に検査するため，検査対象システムのふるまいをもれなく正確に記述しなければならない。このため，従来の開発で人間系での作業を前提として状態モデルを作成していた場合，精度の向上が必要となることが多い。また，検査ツールが独自の記述言語を持つ場合が多く，その習得が必要となる。

(2) 検査条件の作成が難しい（A2）　検査条件は時相論理に基づく論理式やオートマトンを用いて記述する必要がある。また，状態モデルと同様，ツール独自の記述言語の習得が必要となる。

(3) 検査ツールの操作が難しい（A3）　各検査ツールの特性に合わせた実行時パラメータの調整と，状態モデルや検査条件の最適化のノウハウが必要となる。また，検査ツールごとに操作方法が異なるため，ツールが変更さ

れるたびに習得が必要になる。

(4) 反例の解析が難しい (A4)　出力される反例は状態モデルの実行ログの形式となる場合が多いが，このログ量はしばしば膨大な量となるので，解析にはノウハウが必要となる。

3.2 従来開発とのすり合せの難しさ

(1) 開発プロセス変更のインパクトがある (B1)　モデル検査を設計時に実施するため，従来の開発プロセスと比較し，上流での作業コストが増加する。また，作成する状態モデルは設計仕様として利用できるが，従来作成していた仕様と種類や精度が異なるため，レビューの方法など，品質保証のための作業規定の見直しが必要になる。

(2) 検査技術を習得するリソースの確保が難しい (B2)
検査技術の難しさから，検査ツールを自在に使いこなせるようになるまでには一定の習得コストが必要になる。利用する検査ツールの特性や個人のスキルに依存するが，習得期間が数か月以上掛かる場合も珍しくない。

4 モデル検査適用のための施策

東芝は，前述の課題に対して検討を重ね，技術と体制の両面から施策を考案し実施してきた。以下，それらについて述べる。

4.1 技術面での施策（課題A1, B1に対応）

技術面では，状態モデル記述の困難さを低減し，効率よく正確な仕様を作成するため，複数の状態モデル表記法を組み合わせた仕様化プロセスを工夫した。利用した表記法は状態遷移図，状態遷移表，及びタイミングチャート（シーケンス図）であり，これらはそれぞれ**表1**に示すような特徴を持っている。

表1. 状態モデル表記法の比較
Trade-offs among state model diagrams

比較の観点	仕様表記の適正		
	状態遷移図	状態遷移表	タイミングチャート
人手による正確性の確認	★★	★	★★★
人手による網羅性の確認	★★	★★★	★

＊★の多いほうが，人手で仕様を確認するという用途に対して適正が相対的に優れていることを示す。

状態遷移表は網羅性の確認に優れており，モデル検査の直接の入力となる詳細な状態モデルを表記するのに適している。その理由は，状態遷移表では状態とイベントが列と行に記述されており，作成時に，遷移とアクションを直交したマスの中に一つずつ記述することになり，状態とイベントの組合せ漏れが抑制されるためである。

しかしながら，状態遷移表ではふるまいの表現が直感的でなく，ほかの二つの表記法と比べ，正しく仕様を書いているかを作業者自身で確認しにくい。このため，次のような，各表記法の長所を利用する手順を考案した（**図5**）。

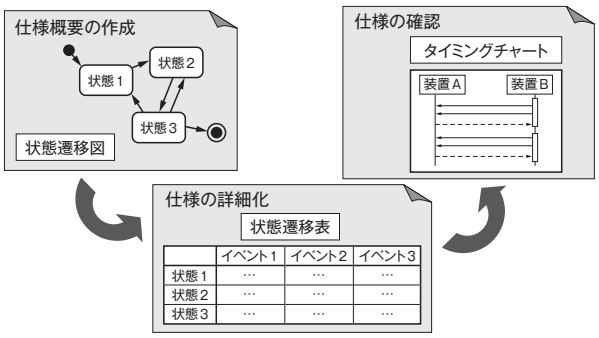

図5. 状態モデルの開発手順 — 状態遷移図，状態遷移表，タイミングチャートの順に仕様をまとめることで，仕様概要の作成，仕様の詳細化，仕様の確認を行う。
State model development process

(1) 状態遷移表より劣るが，ある程度網羅性に優れる状態遷移図を用いて仕様の概要を作成する。

(2) 仕様の概要を参考にしながら，状態遷移表により詳細な仕様を網羅的に作成する。

(3) 重要なシーケンスについては，正確性の検証に優れるタイミングチャートにより再度表記し，仕様を確認する。

4.2 体制面での施策（課題A1～A4, B2に対応）

体制面では，開発プロジェクト (PJ) とは独立に専任のモデル検査チームを編成し，モデル検査作業の効率化とノウハウ蓄積による作業品質の向上を図った（**図6**）。

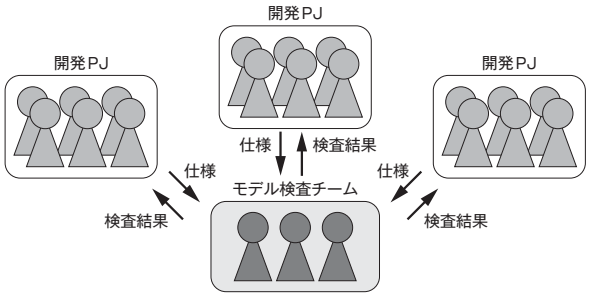

図6. モデル検査チームの役割 — 専任のモデル検査チームを編成し，複数の開発PJの支援を実施する。
Role of model checking team

モデル検査チームの役割は以下のとおりである。これらのうち，(1)～(3)は開発PJと協調して行うが，(4)は独自の役割である。

(1) 開発PJへの状態モデル・検査条件作成支援　状態モデルの作成支援として，記述ガイドラインの作成や事前教育の実施，実作業への協力を行う。また，開発PJから検査すべき条件をヒアリングし，必要に応じて過去の不具合などの情報から検査条件をまとめる。

(2) モデル検査の実施と報告　モデル検査ツールの入力

として状態モデル，検査条件を必要に応じて最適化したモデル検査独自の言語で入力し，モデル検査を実施する。また，検査結果を解析し，問題点を明確にした報告書を作成する。

(3) モデル修正案のレビューと再検査　開発PJにモデル検査結果を報告した後，開発PJからのモデルの修正案をレビューし，再度，検査を実施する。

(4) 最新技術の調査と適用方式の検討　モデル検査技術について最新動向を定期的に調査し，有望な技術について実開発への適用方法を検討する。

モデル検査チームを専任化するメリットは，チームに作業ノウハウが集中することにより，モデル検査作業のコストの低減や作業品質の向上が可能になることである。また，教育などの導入コストも削減できる。その反面，専任チームから現場への，モデル検査技術の移転が難しくなるというデメリットがある。しかし，以下の理由により，現時点ではモデル検査チームの専任化が最適と判断している。

(1) モデル検査の適用ノウハウを十分蓄積してから技術移管するほうが効率的

(2) モデル検査技術がまだ発展途上であり，今後，標準となる手法やツールが新たに現れる可能性があるため，モデル検査技術が成熟した時点で技術移管するのが効果的

5　モデル検査適用施策の評価と今後の改善

今回説明したモデル検査適用のための施策は，実際の開発PJに適用し，現在も評価を続けている。現時点で現場からは，施策に対する以下のような肯定的な意見と否定的な意見を得ている。

(1) 肯定的意見

(a) モデル検査の導入により，従来は発見が難しかった不具合の早期発見が可能

(b) 状態モデルの仕様化方法の改善により，モデル検査以前に仕様の漏れが防止でき，更に，設計レビューやテストケースでの利用が可能（副次的効果）

(2) 否定的意見

(a) 仕様の作成に掛かるコストの低減が必要

(b) モデル検査の実施に掛かるコストの低減が必要

肯定的意見については，施策の効果として当初から予想していたものである。一方，否定的意見については今後，次のような対策を実施していく。

(1) 仕様の入力を効率化する設計支援ツールの導入　状態遷移図や状態遷移表，タイミングチャートで相互にデータを利用するのに最適な設計支援ツールを選定し利用

(2) モデル検査を効率化するツールの導入　モデル検査作業でコストの掛かる状態モデルの入力や反例の解析に最適なツールを選定し利用，又は補助ツールを開発

6　あとがき

上流設計工程での品質改善のキー技術と目されるモデル検査の概要と課題，及び実用化に向けて当社が考案した施策とその評価について述べた。今後は，手法を更に洗練するとともに，関連する技術の整備と実績の拡大を図る。

文　献

(1) 産業技術総合研究所システム検証研究センター. 4日で学ぶモデル検査. 東京, 産業技術総合研究所 エヌ・ティー・エス, 2006, 174p.

(2) G. J. Holzmann. THE SPIN MODEL CHECKER. Boston, Addison-Wesley, 2003, 596p.

池田　信之　IKEDA Nobuyuki
ソフトウェア技術センター　ソフトウェア設計技術開発担当参事。ソフトウェア上流設計の技術開発とソフトウェア開発プロジェクトの支援業務に従事。情報処理学会会員。
Corporate Software Engineering Center

今村　紀子　IMAMURA Noriko
ソフトウェア技術センター　ソフトウェア設計技術開発担当主務。ソフトウェアの要求分析技術の開発に従事。
Corporate Software Engineering Center

高田　沙都子　TAKADA Satoko
ソフトウェア技術センター　ソフトウェア設計技術開発担当。ソフトウェア上流設計の技術開発とソフトウェア開発プロジェクトの支援業務に従事。
Corporate Software Engineering Center