高品質なソフトウェアを効率よく開発できるモデルベーステスト技術

Model-Based Testing Method for Development of High-Quality Software with High Efficiency

太田　暁率　　進　　博正　　渡邊　竜明
■ OHTA Akinori　■ SHIN Hiromasa　■ WATANABE Tatsuaki

高品質なソフトウェアを効率よく開発するため，ソフトウェア開発の上流工程から，動作モデルを用いる仕様検討と，動作モデルを用いてテスト設計を進めるモデルベーステストの導入が進められている。しかし，適用対象が複雑になるとテストケースが膨大になるため，テストケースを選択する技術が求められている。

東芝は，独自のテストケース選択方式を搭載したモデルベーステスト技術を開発した。この技術は，動作モデルから得られた状態遷移系を同値関係により縮約してテストケースを選択し，縮約前の状態遷移系上で復元することで同値な状態間の遷移関係を網羅した状態遷移テストを生成する。この技術を自動車のクルーズ制御システムに適用し，その有効性を確認した。

For the efficient development of high-quality software, both specification review based on a behavioral model and test design utilizing a model-based testing method are being introduced at the early stages of development. However, since it is difficult in practice to cover all test cases of complex software, a technology for the selection of test cases has become essential.

Toshiba has developed a model-based testing method incorporating our original technology for the flexible selection of test cases. This method makes it possible to realize state transition testing generated from a behavioral model that covers all equivalent classes corresponding to the typical behavior of software. We have applied this method to the test design of an automobile cruise control system and confirmed its effectiveness.

1　まえがき

高品質なソフトウェアを効率よく開発するためには，開発の上流工程での仕様検討を十分に行い，下流工程の後戻りを減らすフロントローディング設計が有効である[1]。上流工程での仕様検討を深めるには，従来の文書中心の仕様検討では限界があり，モデルを用いた仕様検討が必要と考えられる。ソフトウェアの設計に用いるモデルは様々だが，開発対象のふるまいをとらえる動作モデルは，上流での仕様検討だけでなく下流の検査工程で用いるテストの設計にも非常に役だつ。

東芝は，ソフトウェア開発の上流工程から，動作モデルを用いて仕様検討とテスト設計を進めるモデルベーステストの導入を進めている（**図1**）。

しかし，適用対象が複雑になるとモデルベーステストで生成するテストケースは膨大になるので，テストケースを選択する技術が不可欠となる。当社は，独自のテストケース選択方式を搭載したモデルベーステスト技術を開発した[2]。ここでは，この技術の概要と組込みシステムへの適用例などについて述べる。

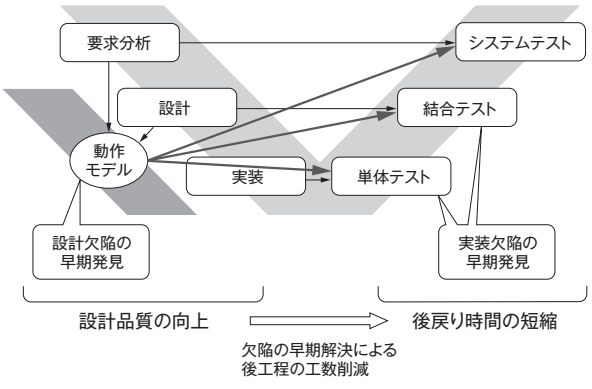

図1．動作モデルとソフトウェア開発工程 ― ソフトウェアの開発で，開発対象のふるまいをとらえる動作モデルは，上流での仕様検討だけでなく下流の検査工程で用いるテストの設計にも非常に役だつ。

Behavioral model and software development process

2　モデルベーステストの基本

モデルベーステスト[3]とは，検査対象の特定の性質を表現するモデルを用いてテストケースを設計するテスト技法の総称である。テストの設計にモデルを用いる利点は，モデルの構造を介してカバレッジ基準を明確化できる点や，カバレッジ基準を満たすテストケースを系統的に設計できる点にある。なかでも検査対象のふるまいを表現する動作モデルを用いると，上流工程での形式検証を利用して仕様検討を深めたり，実行可能なテストケースの生成に活用できる利点がある。

動作モデルを用いるモデルベーステスト技術の基本的な方式は，以下のとおりである。開発対象の動作モデルを与えると，動作モデルの状態空間を探索して状態遷移系を構成する

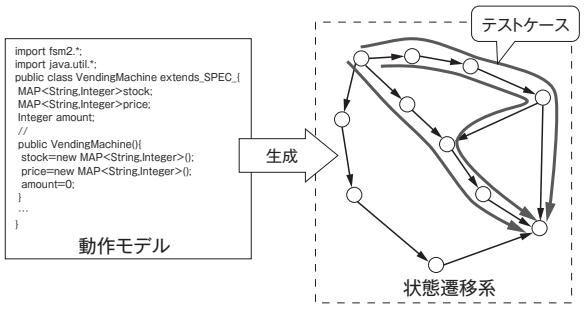

図2. 動作モデルと状態遷移テスト — 開発対象の動作モデルが与えられると，動作モデルの状態空間を探索して状態遷移系が構成され，テストをどれくらい実施すべきかというカバレッジ基準を満たす状態遷移テストが生成される。

State transition tests extracted from behavioral model

（**図2**）。構成した状態遷移系から，テストをどれくらい実施するべきかというカバレッジ基準を満たす状態遷移テストを生成する。ターゲットとするカバレッジ基準とは状態網羅や遷移網羅などであり，それぞれグラフアルゴリズムを用いて近似的に最小な遷移テストを生成できる。なお，状態遷移テストは制御系ソフトウェアと相性の良い方式と考えられている。

当社の方式の基本的な部分は，ほかのモデルベーステスト技術[4]と共通している。独自の部分は大規模な検査対象に適用するため，以下に述べる状態遷移系の生成と検査及びテストケースの選択機能を統合する部分にある。

状態遷移系は動作モデルの状態空間を調べて生成するが，現実の問題では多くの場合，状態空間を調べ尽くすことが不可能である。そこで，探索する状態空間を希望の範囲に制限するため，探索シナリオの制御機能や探索時の枝刈り制御機能を提供する。前者は，状態遷移を引き起こす事象の発生パターンを正規言語で指定する機能である。後者は，利用者がコールバック関数を記述することで，遷移先に応じた状態探索の継続有無を指定できる機能である。これらを組み合わせて，大規模な動作モデルから有限な状態遷移系を取り出す。

取り出した状態遷移系は，状態や遷移の検索機能を用いて検査できる。これは，状態や遷移を入力とし真偽値を返す関数を利用者が与えることで状態遷移系から希望の状態や遷移を検索する機能であり，いわゆる安全性などの検証を自動化できる。また，検索した状態間の最短の遷移パスなどを計算する機能を持っており，安全性の検証に失敗した場合は反例を表示する。

3　テストケースの選択

与えられた状態遷移系に対して状態遷移テストを求める問題は，状態をノードとみなし，遷移をエッジとみなした有向グラフ(注1)のパスを求める問題に帰着できる。例えば，状態又は遷移を網羅する状態遷移テストは，有向グラフの巡回セールスマン問題や中国人郵便配達問題に帰着でき，それぞれ近似解を求めるアルゴリズムを利用できる。しかし，これらのアルゴリズムは計算量が大きいので，有向グラフが数百万ノードまで大規模化すると安定的に解を求めるのが難しい。しかも，同じ規模の状態遷移系は応用上頻繁に現れるが，大規模な状態や遷移を網羅する状態遷移テストは類似のテストケースを多く含むので，網羅的な状態遷移テストの必要性は低い。そこで当社は，大規模な状態遷移系から状態遷移テストを柔軟に選択する技術を開発した[2]。

この技術は，状態遷移系に含まれる状態や遷移に関する同値関係を与え，同値な状態や遷移をグループ化することで縮約した状態遷移系を構成する（**図3**）。同値関係によって縮約した状態遷移系は，検査対象の代表的な挙動を示すと解釈でき，システムの代表的な挙動を確認するテストに対応する。縮約後の状態遷移系の遷移や状態を網羅する遷移パスを手がかりに，縮約前の状態遷移系上の遷移パスを復元する。

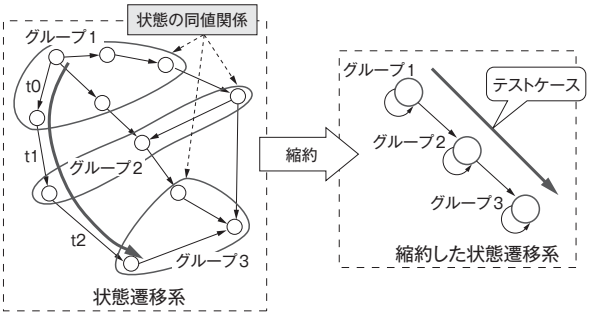

図3. 状態遷移系の縮約とテスト選択 — 状態遷移系に含まれる状態や遷移に関する同値関係を与えて，同値な状態や遷移をグループ化することで縮約した状態遷移系を構成して，テストケースを選択する。

Selection of test cases from contracted state transition system

状態や遷移の同値関係は，状態や遷移を定義域とする関数を与えることで指定する。状態遷移系を状態の有限集合S，遷移事象の有限集合E，遷移の有限集合$T \subseteq S \times E \times S$，及び初期状態$s0 \in S$から成る組$F=<S,T,E,s0>$で表記する。また，遷移$t \in T$の前後状態と遷移事象を，それぞれ$src(t)$，$dst(t)$，$evt(t)$で表記する。与えられた関数fにつき，状態s1とs2が$f(s1)=f(s2)$を満たすとき同値な状態とする。同様に，与えられた関数fとgにつき，遷移t1とt2が同値な前後状態を持ち遷移事象が同値なとき，つまり$f(src(t1))=f(src(t2))$，$f(dst(t1))=f(dst(t2))$と$g(evt(t1))=g(evt(t2))$を満たすとき同値な遷移とする。状態遷移系を同値分割する仕方は，与

(注1)　つながり方だけではなく，どちらからどちらにつながっているかをエッジに矢印をつけて表したグラフ。

える関数により柔軟に変更できる。

縮約後の状態遷移系の遷移パスから縮約前の状態遷移系の遷移パスを復元する手順では，後者の遷移パスは前者の遷移パス以上の長さとなるが，必ず縮約前の状態遷移系の上で実行可能な遷移パスとなることが保証される。この手順の概要は，次のとおりである。

前者の状態遷移系上の遷移パスに含まれる各遷移に対して，後者の状態遷移系の初期状態から順番に始めて，同値な遷移の中から状態が連続する遷移を選んでいく。もし同値な遷移の中に該当する遷移がなければ，希望の遷移先の状態に至る最短パスを求めて前者の1回の遷移に対応させる。また，希望の状態に至るパスも存在しない場合は，状態を初期状態にリセットしたうえで，希望の遷移と同値な遷移への最短パスを対応させる。これら対応する遷移を連結すると初期状態へのリセットが入る場合があるが，前者の状態遷移系の遷移パスを一度以上含んだ後者の状態遷移系の遷移パスが得られる。

4　組込みシステムへの適用

これまでモデルベーステストとテストケース選択の基本的な考え方を述べてきたが，この章では具体的に適用した事例として，自動車のクルーズ制御システムのモード遷移を実現する組込みソフトウェアについて述べる[5]。この事例は組込みソフトウェアの設計技術の比較によく用いられ，多数の実現例が存在する[6]。

4.1　動作モデルの検査例

事例のクルーズ制御システムは，自動車の速度を一定に保つ機能を実現する。制御ソフトウェアは，複数の制御モードを内部状態として備えており，制御パネルや運転ペダルの操作と走行状態センサの値に応じて内部状態を変える。制御モードには，例えば無効状態のオフモードや，準備状態のスタンバイモード，速度制御状態のクルーズモードなどがある。制御ソフトウェアの動作モデルはJava™(注2)言語で記述した[6]。ここでは，制御対象の車両の挙動も離散的な状態遷移系へ近似してモデル化している。

誤りを含んだ動作モデルからテストケースを生成しても意味がない。動作モデルの検証は状態や遷移の検索機能を用いて実施できる。例えば，検査すべき項目が**表1**に示すような形式で与えられる場合を考える。この表は各行が検査項目に対応しており，最初の列が事前条件p，次の列が遷移事象e，最後の列が事後条件qとなる。この検査項目は，事前条件pを満たす状態で遷移事象eが発生すると事後条件qを満たす状態へ遷移するという意味である。つまり，検査項目1は，動作モードがオフで車両速度が40 km/h以上のとき，クルーズスイッチをオンすると，動作モードがスタンバイに変化しクルーズランプが点灯するという意味になる。

表1．検査項目表の記述例
Example of table listing inspection items

検査項目No.	事前条件：p	遷移事象：e	事後条件：q
1	動作モードオフ 40 km/h ≦車速	クルーズスイッチオン	動作モードスタンバイ クルーズランプオン
2	…	…	…

与えられた動作モデルが検査項目遷移を満たすかどうかは，動作モデルから状態遷移系を生成して状態や遷移に関する検索機能を用いると系統的に検査できる。例えば，状態遷移系Fに含まれるすべての遷移t∈Tが，検査項目から生成した性質 $p(src(t)) \wedge e(evt(t)) \Rightarrow q(dst(t))$ を満たすか調べる。

4.2　遷移テストの選択例

状態遷移テストの選択方法を与えるため，状態や遷移の関数を指定する。例えば，状態sの関数 $f(s) = (p1(s), q1(s), \cdots)$ と遷移事象eの定数関数を用いる。この関数を用いると，検査項目に現れる事前条件と事後条件の充足の有無に基づいて分類された，状態の間の遷移系が得られる。この状態遷移系の遷移を網羅する状態遷移テストを用いると，検査項目を手短に確認するテストが得られると期待できる。縮約の有無によるテストの相違を**表2**に示す。

表2．テスト数の比較結果
Comparison of numbers of selected test cases with and without contraction

縮　約	状態数	遷移数	状態遷移テスト数	網羅率
なし	7,930	79,300	79,300以上	100 %
あり	12	65	256	100 %

動作モデルから生成した状態遷移系は状態数7,930，遷移数79,300の規模になった。各状態は制御モードの内部変数に加えて，5 km/hごとに離散化した車両速度と目標速度の組合せを含んでいる。“縮約なし”の場合，状態遷移テストは総遷移数以上の79,300以上となる。“縮約あり”の場合，検査項目の事前条件と事後条件を用いて状態数12と遷移数65の状態遷移系を構成でき，遷移関係を網羅する256個の状態遷移テストが得られた。ここで，表2の網羅率は，モデルベーステスト技術を用いて生成した状態遷移テストが表1の検査項目をテストする割合で，次の(1)式で表せる。

$$\text{網羅率} = \frac{\text{状態遷移テストが含む検査項目数}}{\text{すべての検査項目}} \qquad (1)$$

(注2)　Java及びその他のJavaを含む商標は，米国Sun Microsystems,Inc.の米国及びその他の国における登録商標又は商標。

縮約の有無にかかわらず，表1の検査項目を含む割合は同じ100％となる。したがって，表1の検査項目を一度以上テストすることが要件であれば，縮約ありのテストは縮約なしのテストに比べて，79,300÷256つまり約300倍効率が高い。

通常，動作モデルとテスト対象は詳細度が異なるので，モデルベーステストで生成した状態遷移テストは，開発者が情報を補わないと利用できない場合が多い。情報を手動で補う作業は，縮約ありで選択した状態遷移テスト以外では困難である。縮約の仕方を工夫すると多様な状態遷移テストを生成できる。

5　テスト技法の導入効果

ソフトウェア開発にモデルベーステストを導入すると，上流工程での仕様検討が深まり，下流工程でのテスト効率の向上や手戻りの減少で，コスト削減が期待できる。一方で，動作モデルを準備する工数が純粋なコスト増加となる。つまり，前者の効果が後者の効果を上回ると期待できれば，モデルベーステストを導入すべきである。

近年のフロントローディング設計やプロダクトライン開発の考え方は，モデルベーステストの導入とも相性が良い。フロントローディング設計の考え方によると，上流工程で品質が向上することで，下流工程での手戻りを抑制する効果があり，結果的に開発工数の削減効果を期待できる。従来の開発では，上流工程で見落とした不具合が下流工程のテスト工程で見つかり，不具合の修正のために設計変更や再テストを行うことで，開発期間が予期せず延びることがある。モデルベーステストの導入により，モデル作成のコストは余分に発生するが，上流工程での仕様検討は深まる。独立行政法人 情報処理推進機構（IPA）の調査報告[7]によると，ソフトウェアの不具合のうち37.4％が上流工程で発生し，90％がテスト工程で発見される。

モデルベーステストはプロダクトライン開発とも相性が良い。多くの製品開発は，既存製品を元に再利用して派生的に開発することが多い。IPAの調査報告[7]によると，大半のプロジェクトでは新規に開発するソフトウェアソースの行数は全体の50％以下である。派生品の開発時には，設計資料やコードだけでなく，モデルも再利用することができる。この場合は，全体の工数に占めるモデル作成のコストは新規開発時と比べて小さくなる。

6　あとがき

モデルベーステストの技法は，近年のフロントローディング設計やプロダクトライン開発の考え方とも相性が良く，ソフトウェアの高品質化や開発コストの削減に有効と思われる。この技術は，検査対象の状態空間を与えられた基準に基づき同値分割して，同値な状態間の遷移を網羅する状態遷移テストを生成する。同値分割の仕方は，状態や遷移の関数を与えることで柔軟に変更できるので，様々なシステムへの適用が考えられる。

ここでは，モデルベーステストと組み合わせるテストケースの選択技術と，具体的な組込みソフトウェアへの適用事例について述べた。より複雑な適用例として，フラッシュメモリ製品向け制御ソフトウェアのテスト設計へ適用した例もある。今後，ソフトウェア設計工程でのモデルベーステスト技術の導入を進め，組込みソフトウェアに対する高品質化や低コスト化への高い市場ニーズに応えていく。

文　献

(1) 池田義雄．フロントローディングによる上流設計力強化．東芝レビュー．**62**，9，2007，p.2－8.
(2) 進　博正，ほか．“商状態遷移系を用いたテストケース生成方法”．組込みシステムシンポジウム2008論文集．東京，2008-10，情報処理学会 組込みシステム研究会．2008，p.167－174.
(3) Utting, M.; Legeard, B. Practical Model-based Testing: A Tools Approach. San Francisco, CA, USA, Morgan Kaufmann Pub, 2006, 456p.
(4) Veanes, M., et al. "Model-Based Testing of Object-Oriented Reactive Systems with Spec Explorer". Lecture Notes in Computer Science. **4949**. Heidelberg, Springer Berlin, 2008, p.39－76.
(5) 太田暁率，ほか．“状態抽象化による組込みソフトの状態遷移テスト効率化実験”．第22回-軽井沢ワークショップ論文集．軽井沢，2009-04，電子情報通信学会．2009，p.246－251.
(6) サイバネットシステム（株）．Simulink/Stateflowサンプルモデル解説書−クルーズコントロール編．東京，MathWorks Japan，2004，32p.
(7) 経済産業省：独立行政法人 情報処理推進機構．“2008年版組込みソフトウェア産業実態調査報告書”．<http://sec.ipa.go.jp/reports/20080715.html>，（参照2009-06-30）.

太田　暁率　OHTA Akinori
研究開発センター システム技術ラボラトリー。
ソフトウェアの設計技術の研究・開発に従事。
System Engineering Lab.

進　　博正　SHIN Hiromasa
研究開発センター システム技術ラボラトリー主任研究員。
ソフトウェアの設計技術の研究・開発に従事。情報処理学会会員。
System Engineering Lab.

渡邊　竜明　WATANABE Tatsuaki
セミコンダクター社 システムLSI事業部 システム・ソフトウェア技術部主務。システム・ソフトウェアの開発に従事。
System LSI Div.