

モデル駆動開発を支援するための 拡張運用プロファイルベースドテスト法の適用方法

高木 智彦 河原林 春香 古川 善吾

香川大学工学部

1. はじめに

拡張運用プロファイルは、ユーザがソフトウェアをどのように利用するか、そしてソフトウェアがどのように振舞うかを表現した確率付き拡張有限状態機械である。確率付き有限状態機械として記述される従来の運用プロファイルと違って、状態や遷移の内部における複雑な動作（アクション）や、遷移の発火条件（ガード）、イベントに付随する入力（イベントパラメータ）、各遷移の実行に必要なエフォートなどを記述できる。拡張運用プロファイルに基づいて、ソフトウェアの信頼性を効果的に改善するためのテストケース、すなわち、開始状態から終了状態に至る状態遷移列として表されるテストの入力条件と期待結果を、UDC (usage distribution coverage) や N スイッチ網羅基準を使用して自動生成することが可能である。このテストケース設計法を拡張運用プロファイルベースドテスト法[1]という。一方、モデル駆動開発[2]は、形式モデルとして記述された仕様に基づいてソフトウェアを効果的に開発する方法であり、近年注目されている。形式モデルとして拡張有限状態機械が用いられることも多いため、拡張運用プロファイルベースドテスト法はモデル駆動開発との親和性が高いと考えられる。

本稿では、特に拡張有限状態機械を用いたモデル駆動開発を EFSMDD (extended finite state machine-driven development) と呼ぶこととし、EFSMDD を支援するための拡張運用プロファイルベースドテスト法の適用方法を検討する。まず、ソフトウェア開発への EFSMDD の導入の程度を表す EFSMDD レベルを定義する。次に、各 EFSMDD レベルにおける拡張運用プロファイルベースドテスト法の適用方法を提案する。

2. EFSMDD レベル

本節では以下の 5 つの EFSMDD レベルを定義する。EFSMDD に適しているソフトウェア、す

なわち、外部環境（ユーザや連携するシステムなど）と相互作用を行うソフトウェアを開発する場合は、EFSMDD レベルが高いほどソフトウェアの生産性が高くなることが期待できる。

レベル 1: 拡張有限状態機械を用いて、ソフトウェアの期待される振舞いを記述しない。技術者は、振舞いに関する仕様を体系的に分析、設計することなく、経験と勘に基づいて実装する。

レベル 2: 拡張有限状態機械を用いてソフトウェアの期待される振舞いを記述するものの、そこからソースコードを生成しない。技術者は、振舞いに関する仕様を体系的に分析、設計した上で、すべて手作業で実装する。

レベル 3: 拡張有限状態機械に基づいて、実行可能性を保証しない（すなわち、アクションやガード、イベントパラメータを考慮せず、本来実行不可能な状態遷移も起こし得る）スケルトンコードを生成する。技術者は、振舞いに関する仕様を体系的に分析、設計する。そして生成したスケルトンコードに基づいてプロトタイピングを行ったり、本番ソフトウェアの実装を行ったりする。

レベル 4: 拡張有限状態機械に基づいて、実行可能性を保証した半完成プログラムを生成する。技術者は、その半完成プログラムを実行することで、仕様の動作を即時にレビューしたり、半完成プログラムに手を加えてプロトタイピングを行ったり本番ソフトウェアを実装したりする。

レベル 5: 拡張有限状態機械から完全なプログラムを生成する。ここで「完全な」とは、生成されたプログラムに技術者が手を加える必要が基本的にない、という意味である。

3. 拡張運用プロファイルベースドテスト法の適用方法

本節では、各 EFSMDD レベルにおける拡張運用プロファイルベースドテスト法の適用方法を提案する。

レベル 1 では拡張運用プロファイルベースド

The Application Method of Extended Operational Profile-Based Testing Techniques for Supporting Model-Driven Development

Tomohiko Takagi, Haruka Kawarabayashi, Zengo Furukawa
Faculty of Engineering, Kagawa University, Takamatsu,
Kagawa 761-0396, Japan

テスト法を適用することはできない。適用を可能にするためには、まずモデリングツールの導入、モデリングすなわち拡張運用プロファイルの作成を行う技術者の育成および環境の整備を行う。これらは EFSMDD の導入を促進し、レベル2に上げることに繋がる。

レベル2では、拡張運用プロファイルに基づいて手作業でテストケースを設計する。仕様のレビューや、実装したプログラムのテストに使用することができる。ただし、拡張運用プロファイルベースドテスト法では本来テストケースは自動生成されるものであり、手作業によるテストケース設計ではコスト効果が低い。より効果的な拡張運用プロファイルベースドテスト法を可能とするために、モデリングツールのコード生成機能を有効にする必要がある。ここでいうコード生成機能とは、技術者が定義したコード生成規則にしたがって、拡張有限状態機械からソースコードやテストケースをはじめとした各種ドキュメント類を自動生成する機能のことである。これを有効にすることは EFSMDD の導入を促進し、レベル3に上げることに繋がる。

レベル3では、コード生成機能を有効にし、コード生成規則をカスタマイズする。すなわち、アクションやガード、イベントパラメータを考慮しない単純なグラフ探索アルゴリズムで拡張運用プロファイルを探索することで、実行可能性を保証しないテストケースを生成する。そしてテストケース生成後は、技術者が実行不可能なテストケースを手作業で除外する。また、抽象度が高いので、残ったテストケースを必要に応じて詳細化する。レベル2同様、仕様のレビューや、実装したプログラムのテストに使用することができる。加えて、コード生成規則のテストにも適用できる。テストケースを自動生成できる分コストを削減できるが、一方で、実行不可能なテストケースを除外する作業の結果、テストケースの網羅性が損なわれるという問題がある。より効果的な拡張運用プロファイルベースドテスト法を可能とするためには、コード生成規則を拡張し、アクションやガード、イベントパラメータを生成結果に反映できるようにする必要がある。これは EFSMDD の導入を促進し、レベル4に上げることに繋がる。

レベル4では、コード生成機能を用いて、アクションやガード、イベントパラメータを含む拡張有限状態機械の振舞いを実装したプログラム（半完成プログラム）を生成する。半完成プログラムは、イベントとイベントパラメータを入力すると、状態遷移および付随するアクシ

ョンを実行し、期待結果を出力する機能を持つので、これを用いて実行可能性を保証したテストケースを生成する[1]。イベントとイベントパラメータは、メタヒューリスティクスを応用した生成アルゴリズムによって、高い網羅性を実現するように生成される。最後に技術者は、生成されたテストケースを必要に応じて詳細化する。レベル3同様、仕様のレビューや、実装したプログラムのテスト、コード生成規則のテストに適用できる。テストケースを除外する必要がないので、網羅性を損なうことなくコストも削減できるが、依然として技術者がテストケースを詳細化しなければならないという問題がある。より効果的な拡張運用プロファイルベースドテスト法を可能とするためには、コード生成規則をさらに拡張する必要がある。これは EFSMDD を促進し、レベル5に上げることに繋がる。

レベル5では、コード生成機能により生成される完全なプログラムに対してイベントやイベントパラメータを入力する。仕様のレビューやコード生成規則のテストなどが可能である。

4. おわりに

EFSMDD を支援するための拡張運用プロファイルベースドテスト法の適用方法を提案した。本提案は、拡張運用プロファイルベースドテスト法でテストケース生成に用いるプログラムが、EFSMDD のプログラム自動生成と同様の技術で実現できるという点に着目したものである。

我々はレベル4のソフトウェア開発を想定したテストツールを開発し、試験適用を行っている。今後の研究では、レベル5への具体的な移行方法、およびレベル5における拡張運用プロファイルベースドテスト法の有効性について検討する予定である。

謝辞

本研究は JSPS 科研費 23700038 の助成を受けた。

参考文献

- [1] 高木智彦, 八重樫理人, 古川善吾, "拡張有限状態機械を用いた運用プロファイルベースドテストのテストケース生成手法とツール構成", 情報処理学会論文誌, Vol.54, No.2, 10 pages, Feb. 2013.
- [2] D. S. Frankel, "Model Driven Architecture: Applying MDA to Enterprise Computing", John Wiley & Sons, 2003.