

既存開発プロセスへのモデルベーステスト手法の適用

張 暁 晶^{†1} 丹 野 治 門^{†1} 星 野 隆^{†1}

テスト項目やテストデータの自動生成を行うモデルベーステスト手法を、従来の開発プロセスとの親和性を高める方法を提案し、開発現場への適用を容易にする方法について議論する。

How to Integrate Model-based Testing Technology into Traditional Development Process

XIAOJING ZHANG,^{†1} HARUTO TANNO^{†1}
and TAKASHI HOSHINO^{†1}

The authors would like to discuss on how to integrate the model-based testing techniques such as test case extraction and test data generation into traditional development process.

1. はじめに

抽象モデルに基づいてテスト工程成果物を自動生成するモデルベーステスト（以降、MBT）手法は、ソフトウェア開発のテスト工程のコスト削減に寄与できると期待されている。しかし、既に開発標準が規定されている現場に新しい手法を導入するには困難を伴う。本ワークショップにおいては、既存 MBT 手法を拡張することで従来の開発プロセスとの親和性を高める方法を提案し、適用を容易にする方法について議論したい。

2. モデルベーステストを適用する際の障壁

MBT を用いたテスト項目抽出手法は、組み込みシステムのドメインで報告されている少数の利用事例を除き、開発現場への導入はまだあまりなされていない。多くの開発現場では、既に「開発標準」が規定されている。そうした現場で MBT の既存手法の導入を試みても、2つの問題が壁となり、スムーズな導入ができない。

2.1 抽象モデルの記述によるオーバーヘッドコストがかかる

第一の障壁は、開発現場で抽象モデルを記述することの困難さである。既存の MBT 手法の入力に着目すると、有限状態機械（FSM）やラベル付き遷移システム（LTS）、UML 振る舞いモデル等の抽象モデルの

記述を前提としている。一方、既存開発プロセスにおいて、設計工程の成果物として、外部設計書、内部設計書、プログラム設計書など各種の設計書を作成し、またテスト工程の成果物として、テスト項目表などを作成することが定められている。多くの開発現場において、これらの設計文書やテスト文書は、MS Office 文書の形式で、一定のテンプレートに準拠した形で記述される。抽象モデルの記述は、こうした従来の執筆作業を代替するものではないため、開発案件ごとの技術者が記法を学び、新たな成果物を作成することとなる。特に開発案件が改造開発である場合、過去の開発で作成された設計文書を再利用して加工できないため、母体を含めたシステム全体のモデルを新規に作成するとしたらさらに大きなコストとなる。

2.2 従来の手作業によるテスト項目抽出を代替できるかが不明

第二の障壁は、従来の人手による抽出をカバーできるのかという不安である。既存の MBT 手法に対する評価は、バグ検出能力やモデル網羅率、ソースコード網羅率など、「テストした結果」に着目した評価指標がよく用いられる。一方、開発プロセスでは仕様を網羅的にテストすることが求められており、ピンポイントでバグを検出できればいいという訳ではない。そのため、従来手作業で抽出するテスト項目のバリエーションに対して、うちどの部分が自動抽出されたテスト項目で代替可能なのか、その部分が開発案件では多いのか少ないのか、が知りたい。不足部分があれば、手動もしくは他の自動化手法でテスト項目を補うよう

^{†1} NTT ソフトウェアイノベーションセンター
NTT Software Innovation Center

にできる。そうすると結果的に、テスト項目抽出にかかるコストが自動化によりどの程度削減できるかという予測につながる。既存評価では、開発プロセス観点で「もともと人間がやっていた作業を代替可能か」という評価が欠けているため、特定の開発案件に対して特定の手法を導入できるかの判断が困難である。

3. 障壁を低減する方法

前節で述べた2つの障壁それぞれを低減する方法のコンセプトを述べる。具体的な拡張事例や評価事例については、筆者らの過去文献¹⁾で紹介している。

3.1 MBT 手法の入出力の拡張

第一の障壁を低減するために、利用者である一般的な開発者に対して「抽象モデル」を隠蔽できるよう、既存のテスト項目抽出手法を拡張する。「抽象モデル」を隠蔽し、手法の入力として、より簡便でなじみがある記法（シンタックスシュガー）を提供する。ここでシンタックスシュガーとしては、独自に定義するのではなく、既存開発プロセスで規定された工程成果物である設計文書のテンプレートに準拠する。テスト項目抽出は、設計文書から抽象モデルへ変換したのちに、既存のMBT手法を用いる。こうすることにより、利用者は余分な作業をすることなく、「○○工程では××書を作成」という既存開発プロセスで規定された通りの作業だけをすればよい。入力となる設計書の制定では、既存開発プロセスの設計書として、テンプレート化された帳票や図面を用いることにより、開発者から、なじみのない「モデル」あるいは「モデリング」を隠蔽し、通常的设计作業と同じような使用感で設計を行うことができる。設計文書によって、利用者から抽象モデルを隠蔽できる条件を述べる。Webアプリケーションのドメインを対象に考えると、画面遷移やビジネスロジックの呼び出し等MVCアーキテクチャに則った設計事項は定型的になっている。こうしたドメイン限定の暗黙知を利用することで、利用者に意識させずに内部で抽象モデルへの変換が可能となる。

3.2 MBT 手法に対する評価方法の拡充

第二の障壁を低減するために、MBTで自動生成したテスト項目と、従来手作業で作成したテスト項目の比較を通して、MBT手法に対する新しい評価観点を定義する。

評価に必要な資材を示す。まず評価用アプリケーションの開発過程で手作業で作成された、既存の設計書（以後、 M_d と呼ぶ）と、結合テストを対象とした既存のテスト項目（以後、 M_t と呼ぶ）を入手する。そして、 M_d を元に、拡張後のMBT手法が入力とする

設計書（以後、 A_d と呼ぶ）を作成する。これを使って自動生成したテスト項目（以後、 A_t ）を用意する。

評価観点1：従来の手作業を代替できるか

手動作成したテスト項目 M_t と自動生成したテスト項目 A_t の比較を行い、定量的尺度である手動代替率を定義する。 M_t の要素 m と A_t の要素 a の間に、テスト項目が一致することを示す二項関係 $=_R$ を定義する。目視でつき合わせを行い、一定のルールに従って一致するか否かを判定する。

「対応あり」を示す M_t の部分集合 M_t^R 、 A_t の部分集合 A_t^R を次のように定義する。

$$M_t^R = \{m \in M_t | \exists a \in A_t \text{ s.t. } m =_R a\}$$

$$A_t^R = \{a \in A_t | \exists m \in M_t \text{ s.t. } m =_R a\}$$

ここで、手動代替率 $|M_t^R| / |M_t|$ を定義する。これは手動作成するテスト項目の中で、自動生成によって代替されるテスト項目の割合を表す。同様に自動有効率 $|A_t^R| / |A_t|$ も定義する。これは自動生成のテスト項目の中で、従来手作業でも作成していたため経験的に有効と思われるテスト項目の割合を表す。

評価観点2：手法の導入にかかるコストはどの程度か

テスト項目抽出手法を利用するための作業工数を測定する。自動化によりテスト項目抽出自体の工数はほぼゼロとなるが、抽出の元情報である設計書 A_d を適切に作成する工数を指す。評価に用いたアプリケーションの規模に依存しないよう、単位当たり（例えば1Web画面分あたり）の設計書一式を作成するための工数を測定する。さらに、 A_d の作成工数 $< M_t^R$ の作成工数となれば、手法導入により工数削減の効果が期待できる。

4. おわりに

開発現場へのMBT手法の導入の障壁となっている問題を提起し、問題を解決するための拡張方法および評価観点を提案した。新しい評価観点により、MBT手法を適用できるか否かの判断がしやすくなったと考える。しかし、従来手作業のうち、MBTの自動生成に「完全に」任せておける領域、すなわち、自動抽出手法の能力の境界線を、利用者に対してどのようにして明確に示すかについて、今後さらに検討を深めたい。

参考文献

- 1) 張曉晶, 丹野治門, 星野隆. 設計書に基づくテスト項目抽出手法の提案および既存開発プロセスへの適用. 電子情報通信学会技術研究報告. SS, ソフトウェアサイエンス, Vol. 113, No. 269, pp. 167-172, oct 2013.