

# モデルベーステストに基づくテスト実行スクリプト生成手法の提案

丹野 治門†

張 暁晶†

†NTT ソフトウェアイノベーションセンタ

## 1 はじめに

ソフトウェアテストはソフトウェアの品質を確保する上で重要である。しかしながら、現状、テストは大部分が手動で行われており、とても労力がかかるという問題がある。テストはテスト設計とテスト実施に分けられる。テスト設計とはソフトウェア仕様に基づいたテストケースの網羅的な洗い出しや、各テストケースにおけるテストデータ作成などの作業であり、テスト実施は各テストケースを実際に実行し、動作結果を確認し合否判定を行う作業である。本研究では、特に手作業だと労力のかかるテスト実施の効率化に着目した。

テスト実施の労力を減らすための既存方法として、テスト自動実行ツール [1] が入力とするスクリプトを作成し、テスト実行を自動で行う方法があるが、この方法には以下のような問題点がある。

問題 (1): キャプチャ&リプレイ機能の補助により、多少効率化できるものの、初回のスクリプト作成には労力がかかる。

問題 (2): 仕様や設計が変更された時の、スクリプトのメンテナンスにも労力がかかる。スクリプト修正箇所のサジェストなどにより、メンテナンスのコストを小さくする試み [2] もあるが、全てが自動化されているわけではないため、やはり手動のコストはかかってしまう。

本研究の目的は、上述したような問題を解決し、ソフトウェア仕様に基づいた網羅的なテスト実施の労力を削減することである。本研究では、図 1 に示すような業務システムのフロントエンドである Web アプリケーションの、単一画面遷移の動作を確認するためのテストにおけるテスト実施をスコープとする。

## 2 提案手法

本研究では、モデルベーステストの考え方に基づき、ソフトウェアの設計書からテスト実行スクリプトを自動生成する手法を提案する。

本手法は、筆者らが過去に提案した設計書からテストケースを自動生成する機構 [3] を拡張して実現した。図 2 に示すように、本手法では、ソフトウェア設計書及びその内部モデルである設計モデル、テストモデルを拡張しており、テストスクリプト生成に必要な物理情報 (HTML における画面要素の ID 等) を保持できるようにし、各テストケースと対応づける点を特徴とし

図 1: 対象とする Web アプリケーションの画面の一部

ている。この特徴により、以下のように既存方法の問題点を解決することができる。

問題 (1) の解決: ソフトウェア設計工程の成果物である設計書のみからテスト実行スクリプトを生成できるため、初回のスクリプト作成の労力がかからない。

問題 (2) の解決: 仕様変更があり設計が修正された場合でも、設計書からスクリプトを再生成するだけで、設計書と整合性の取れた最新のスクリプトが得られる。

これらにより、提案手法を用いることで効率よくテスト実施を行うことが可能となる。

図 1 を題材とし、提案手法の入力となる設計書 (図 2 の (a)) の例を図 3 に示す。図のように、画面設計書では、外部設計における論理情報 (画面要素) と、詳細設計相当の物理情報 (HTML における物理 ID) の対応を記述可能にし、これらの情報からそれぞれのテストケースに対応したスクリプト生成 (図 2 の (b)) を行うことで、実際にブラウザ上で実行可能なスクリプトが生成可能となる。図 3 の設計書からのスクリプト生成結果の出力例として、キーワード駆動の考え方に基づく GUI テスト自動実行フレームワーク Open2Test[4] が入力とするスクリプトを図 4 に示す。スクリプト本体 (図 4 の (ア) Test Script) では操作対象のオブジェクトに対してどのようなアクションを行うかを示しており、画面要素とその ID の対応表は別ファイル (図 4 の (イ) Object Repository) となっている。例えば、図 4 (ア) の 1 行目はログイン ID という画面要素に対応するブラウザ上の TextBox へ「Tanaka」というテストデータを入力することを示す。

## 3 評価

テスト実施の労力削減効果を確認するため、(1) スクリプトを全て手動で作成する方法、(2) 提案手法を用いてスクリプトを自動生成する方法、参考として (3) ス

Test Script Generation by Model Based Testing

†Haruto TANNO †Xiaoqing ZHANG

†Software Innovation Center, NTT

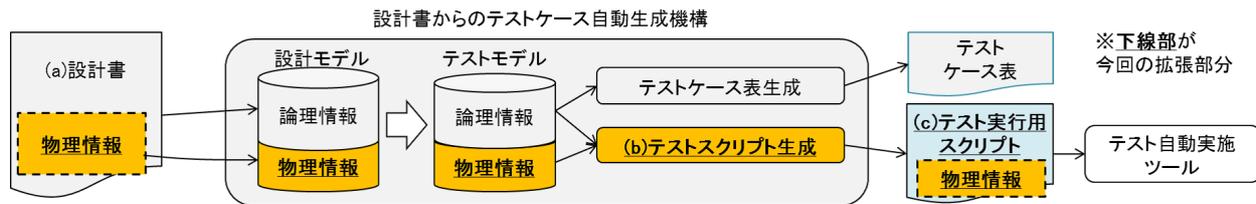


図 2: 提案手法の全体像

項番	物理ID	画面要素名	項目種別
1	userid	ログインID	テキストボックス
2	passid	パスワード	テキストボックス
...	...	...	...
5	country	国籍	リストボックス
...	...	...	...

図 3: 画面設計書の記述例 (抜粋)

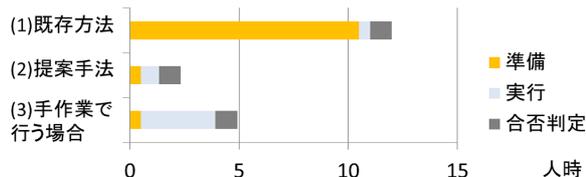


図 5: 評価結果

(ア) Test Script

実行	キーワード	操作対象オブジェクト	アクション
r	perform	Page0:ログインID	set:Tanaka
r	perform	Page0:パスワード	set:abcd
...	...	...	...
r	perform	Page0:国籍	select:日本
...	...	...	...

(イ) Object Repository

Object Name	Object Type	Object Path
Page0:ログインID	TextBox	id=userid
Page0:パスワード	TextBox	id=passid
...	...	...
Page0:国籍	ListBox	id=country
...	...	...

図 4: スクリプトの出力例 (抜粋)

クリプトを用いず全て手動でテスト実施を行う方法の3通りで行い、それぞれの方法でテスト実施にどれだけの労力(手動の時間)がかかるかを測定した。

テスト設計は、筆者らが過去に開発したテスト設計自動化ツール [3] により自動で行うことを前提とした。テスト自動実施ツールとしては、Open2Test を用いた。また、今回の評価では、図 1 に示したような単純な Web 画面 (画面要素は 13) をテスト対象として、テストケース全 1320 個のうち 60 個をランダムに選び、それらのテストケースを用いてテスト実施を行った。

評価結果を図 5 に示す。図において、準備とはテストを実際に行う前のテストスクリプト作成やデバッ

グ作業など、実行はテストの実行、そして、合否判定はテスト実行結果確認にそれぞれ要した手動の作業時間を人時単位で示している。提案手法によって、既存方法での準備に要する作業時間を 5% にまで削減し、テスト実施全体(準備, 実行, 合否判定)では 19% にまで作業時間を削減できた。これにより、提案手法では大きく労力を削減できていることが確認できた。

4 まとめ

本研究では、設計書からテスト実行スクリプト生成を行う手法を提案し、簡単な例を用いた評価によりもテスト実施の労力を削減できることを確認した。今後は、テストの合否判定など、より自動化の適用範囲を拡大し、テストの効率化を進めていきたい。

参考文献

[1] Selenium. <http://www.seleniumhq.org/>.  
 [2] Mark Grechanik, Qing Xie, and Chen Fu. Maintaining and evolving gui-directed test scripts. In *Proceedings of the 31st International Conference on Software Engineering, ICSE '09*, pp. 408–418, Washington, DC, USA, 2009. IEEE Computer Society.  
 [3] 丹野治門, 張曉晶, 田端啓一, 生沼守英, 村主一仁. ソフトウェアの品質確保と開発コスト削減を目指したテスト自動化技術. *NTT 技術ジャーナル*, Vol. 25, No. 10, pp. 19–22, oct 2013.  
 [4] Open2test. <http://www.open2test.org/>.