

シナリオテストにおけるテスト項目自動生成手法の提案

丹野 治門[†] 張 曉晶[†] 星野 隆[†]
 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、漏れやミスが発生してしまうという問題がある。このような問題を解決するため、筆者らは、業務システムの結合テストをスコープとし、これまで単一画面遷移を対象とし、ソフトウェア設計書からテスト項目を網羅的に自動生成する取り組みを行ってきた [1]。

本研究では、よりテスト項目の自動生成領域を拡大し、単一画面遷移より粒度の大きいテストである複数画面遷移を伴うテスト（以降、シナリオテストと呼ぶ）におけるテスト項目自動生成手法を提案する。本研究では、シナリオテストにおいて、設計書に基づき機能性に関する正常系、準正常系のテスト項目を網羅的に自動生成できるようにすることを目的とする。

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、設計工程の成果物である設計書から網羅性の高いテスト項目を自動生成するアプローチをとる。本アプローチの特徴は以下の 2 点である。

- 設計工程で必ず作成する成果物である **ユースケース記述**、**画面遷移図** という 2 種類の設計書を入力として、そこからテスト項目を自動生成するため、開発現場への導入障壁が低い。
- ユースケーステスト技法 [2]、経路網羅といった考え方に基づき、網羅性の高いテスト項目を生成できる。

以下、ログイン処理と検索処理から構成される簡単なアプリケーションを例として、提案手法が入力とする設計書と、テスト項目の生成方式について説明する。

項番	利用者アクション	システムアクション	他フローへのジャンプ
基本フロー			
(1)		「ログイン」画面を表示する。	
(2)	アカウント情報を入力し、「ログイン」ボタンを押下する。		代替フロー(1)
(3)		検索画面を表示する。	
(4)	検索条件を入力し、「検索」ボタンを押下する。		例外フロー(1)
(5)		入力された検索条件に基づき検索処理を行い、検索結果を「検索結果」画面に表示する。	
(6)	検索結果を確認する。		
(7)		<終了>	
代替フロー			
(1)		ログインに失敗した場合は、「ログインエラー」画面を表示する。	
(2)	ログイン画面へ戻るリンクをクリックする。		基本フロー(1)
例外フロー			
(1)		DB接続に失敗した場合は、「DB接続エラー」画面を表示する。	
(2)	「リトライする」ボタンを押下する		基本フロー(3)

図 1 ユースケース記述

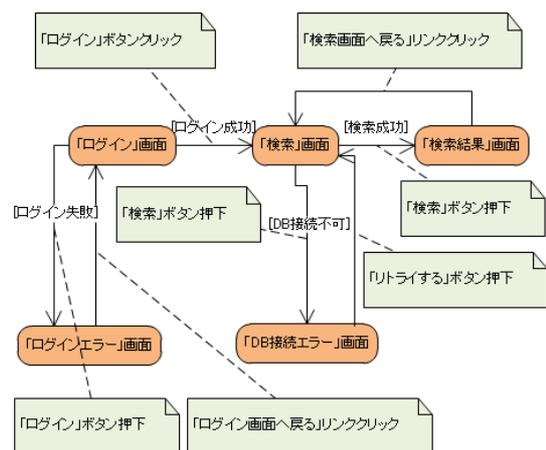


図 2 画面遷移図

提案手法が入力とするユースケース記述と画面遷移図を図 1, 2 にそれぞれ示す。ユースケース記述では、ログインを行い、次に検索を行い、最後に検索結果画面を確認し終了する、といった正常系のシナリオである基本フロー、そして、途中でログインに失敗した場合のエラー処理など準正常系のシナリオである代替/例外フローを、

Test Case Generation for Scenario Testing
[†] Software Innovation Center, Nippon Telegraph and Telephone Corporation

利用者のアクションとシステムのアクションに分けて記述される。画面遷移図では、アプリケーションの各画面と、各画面における利用者アクションとそのアクションに紐づく画面遷移の情報を記述される。

上述した 2 種類の設計情報を入力とし、提案手法では以下の 3 つの観点で網羅性の高いテスト項目を生成する。

No	観点	テスト項目	テスト手順	
		内容	対象	アクション
1	基本フロー	以下のパスにおいて正常に動作すること。 「ログイン」画面→「検索」画面→「検索結果」画面 また、各処理実施後の画面項目が正当であり、機能完了後のDB状態と遷移先画面が正しいこと。	「ログイン」画面 「検索」画面 「検索結果」画面	「ログイン」ボタン押下 「検索」ボタン押下
...
7	画面遷移図からの補完	以下のパスにおいて正常に動作すること。 「ログイン」画面→「検索」画面→「検索結果」画面→「検索」画面 また、各処理実施後の画面項目が正当であり、機能完了後のDB状態と遷移先画面が正しいこと。	「ログイン」画面 「検索」画面 「検索結果」画面 「検索」画面	「ログイン」ボタン押下 「検索」画面へ戻るリンクをクリック
...

図 3 テスト項目表

観点(ア)：利用者視点に基づいたテストを生成する。具体的には、ユースケーステスト技法の考え方に従い、ユースケース記述から「基本フローを順次実行するパス」、「それぞれの代替/例外フローを 1 回実行するパス」、「全ての代替/例外フローを実行するパス」、「それぞれの代替/例外フローを指定された回数実行するパス(今回は回数を 2 とした)」を抽出する。

観点(イ)：ユースケース記述からだけでは、作成されない利用者視点のテストを、ソフトウェアの構造に基づいた情報を補完することで生成する。具体的には、ユースケース記述と画面遷移図を用いたマッチングを行うことで、ユースケース記述からだけでは作成されないテストを補完する。例えば、「戻るリンク押下」等のテストがこれに相当する。

観点(ウ)：ソフトウェアの構造に基づいたテストを生成する。具体的には、どの画面でもシステムが終了するという観点のもと、経路網羅の考え方にに基づきテスト項目を抽出する。

これらの観点に基づき、図 3 に示すようなテスト項目表を出力する。各テスト項目には手順や確認内容が記述されており、テスト実施者はこのテスト項目表を用いながらテストを実施していくことができる。

3. 評価

提案手法によるテスト項目自動生成で、これまで手動で作成されていたテスト項目作成をどれだけカバーできるか、手動作成で漏れていて提案手法によって補強されたテスト項目があるかどうか、また、自動生成ではカバーできなかった項目がどれだけあるかという観点で評価を行った。

評価には、実開発案件であるシステム A (5 機能, 33 画面, テスト項目は 91 件) を用い、図 4 のように元々存在するシステム A の設計書を提案手法が入力とする設計書へ書き換え、そこからテスト項目自動生成を行い、この自動生成したテスト項目 104 件と、手動作成されたテスト項目 91 件との比較を行った。評価結果は図 5 のようになった。

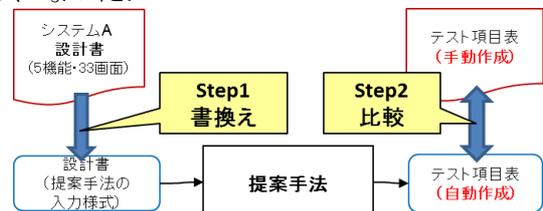


図 4 評価方法

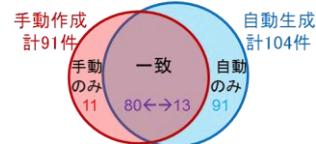


図 5 評価結果

シナリオテスト(正常系, 準正常系の画面遷移)については、87.9%(91 件中 80 件)の試験内容をカバーできた。自動生成できなかったテスト項目は 12.1%(91 件中 11 件)であり、これは、観点(イ)で、複数のユースケースを跨ぐ遷移を無視していることなどが原因であった。また、手動作成では漏れており提案手法で補強できたテスト項目は 91 件あった。これは元々、手動作成されたテスト項目が経路網羅に基づいた考えで作成されていなかったためである。

4. まとめ

本研究では、設計書に基づきシナリオテストのテスト項目を自動生成する手法を提案し、実開発案件を用いた評価を行った。今後は、開発現場のフィードバックを得ながら、使える技術を目指して手法を改良していきたい。

5. 参考文献

[1] 張 暁晶, 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, vol. 113, no. 269, SS2013-46, pp. 167-174, 2013 年 10 月
[2] ソフトウェア教科書 JSTQB Foundation 第 3 版