

テスト設計支援ツールの設計と実装

大村 裕[†] 渋谷 健介[†] 嶋田 大輔[†] 早川 祐志[†]

NEC 共通基盤ソフトウェア研究所[†]

1. はじめに

ソフトウェアの大規模化、複雑化に伴いソフトウェアの品質確保が非常に重要になっている。一般にソフトウェアの品質はテストによって確保される。開発者はソフトウェアの設計書をもとにテスト設計を行い、テスト環境の整備、実施を行う。

しかし、人手により設計書からテスト設計を行うため属人性を排除できず、テスト設計書の品質が一定ではないという課題があった。この課題に対処するために、モデルベーステストの研究開発が進められている[1]。モデルベーステストとはテスト対象のシステムをモデル化し、そのモデルからテストケースを導出する手法である。このモデルベーステストの支援ツールは様々なものが存在する[2]。しかし、これらツールは人手で設計書のモデル化を行うため、導入コストが高いという課題があった。そのため、導入コストの低いテストツールが求められる。

2. テスト設計支援ツール

本研究では上記の課題を解決するために、導入コストの低いテスト設計支援ツールを設計・実装した。本ツールは導入コスト低減のため、一定のフォーマットに従った設計書を入力として、テスト設計書を生成する。本ツールは業務用ウェブアプリケーションの画面入力テストと画面遷移テストを対象とした。テスト設計書をツールにより生成することで、品質を一定にでき、テスト項目の漏れを防止した。また、より導入コストを低くするため、社内の設計書フォーマットをなるべく変更せずにツールを導入できるように配慮した。

2-1. 業務用ウェブアプリケーションのテスト項目分析

我々は典型的な業務用ウェブアプリケーションのテスト項目を調査・分析し、画面入力テストと画面遷移テストが平均して全テストの約 6 割を占めることを確認した。

まず、テスト項目を以下のように分類した。

1. 画面入力テスト
2. 画面遷移テスト
3. ロジックテスト

4. 画面 UI 動作テスト

(ブラウザ内で完結する UI 動作のテスト)

5. 画面レイアウトテスト

6. その他

さらに、それぞれの分類項目のテストケース数が全体に対して占める割合を事例毎に以下に示す。

	1	2	3	4	5	6
事例 1	25.2%	41.5%	19.1%	11.2%	3.0%	0.0%
事例 2	28.3%	35.0%	16.5%	13.7%	4.7%	1.9%
事例 3	40.5%	24.9%	26.6%	2.5%	3.9%	1.6%
平均	31.3%	33.8%	20.7%	9.1%	3.9%	1.7%

表 1: テストケース数比率

上記結果にもとづき、本研究では全テストの 6 割を占める画面入力テストおよび画面遷移テストに着目し、テスト設計支援ツールを試作した。画面入力テストはブラウザにおける各項目のデータ入力のチェックが設計書通りであることを確認する。また、画面遷移テストはウェブページが遷移イベントに応じて、設計書通りに遷移することを確認する。

2-2. 画面入力テスト

本ツールでは設計書の画面入力項目の仕様を表形式で記述する。その仕様をもとに境界値分析を行い、画面入力テストケースを生成する。

ツール利用者は設計書の入力項目仕様にデータ型を指定する。本ツールでは 4 つのデータ型(整数、日付、リスト、文字列)を用意している。さらに各データ型に沿った仕様を設計書に記述する。一例として、文字列型の入力項目の仕様記述について説明する。

文字列型では入力項目の必須設定のほかに、文字列の最大長および最小長、文字種(半角、全角、混在)、文字列種(数字、英字、仮名、カタカナ、漢字)の指定を行う。これらの仕様は従来の設計書では自由記述されているが、本ツールでは表形式で記述を行う。本ツールはこの仕様から各仕様(必須、文字列の長さ、文字種、文字列種)の境界値分析を行い、それらを組み合わせてテストケースを生成する。

例えば、表 2 の仕様が記述されている設計書を入力とした場合、表 3 の 13 個のテストケースが出力される。表 2 ではこのユーザ ID を表す項目が記述必須の項目であり、長さが 8 固定で、半角の数字だけが入力として認められることを示している。表 3 は本ツールによって出力された

Design and Implementation of Test Case Generator

[†]OHMURA, Yuu, SHIMADA Daisuke, SHIBUYA Kensuke, HAYAKAWA Masashi

Common Platforms Software Research Laboratories, NEC Corporation(†)

テストケースのテスト観点を示している。実際には入力具体値も出力される。

	必須	最大長	最小長	文字種	文字列種
ユーザ ID	○	8	8	半角	数字

表 2: 画面入力設計書サンプル

テストケース	文字列長さ	文字種	文字列種
1	0		
2	7	半角	数字
3	7	半角	数字以外
4	7	全角	任意
5	7	混在	任意
6	8	半角	数字
7	8	半角	数字以外
8	8	全角	任意
9	8	混在	任意
10	9	半角	数字
11	9	半角	数字以外
12	9	全角	任意
13	9	混在	任意

表 3: テストケースサンプル

2-3. 画面遷移テスト

本ツールは設計書の画面遷移図の遷移イベント情報に Main、Sub という優先度情報を付加し、実際のシステム利用シーンに近い画面遷移テストケースを生成する。

ツール利用者は設計書の画面遷移仕様を本ツールで用意したフォーマットにより画面遷移図として記述する。図は開始点、終了点、画面、遷移イベント(Main, Sub)から構成される。遷移イベントの優先度情報はツール利用者が実際の利用シーンを考慮に入れて指定する。

本ツールはこの画面遷移図を入力としてすべての遷移イベントが必ず 1 回は通過するテストケースを生成する。ただし、単純にテストケースを生成すると実際のシステム利用シーンとは異なるテストケースを生成する可能性がある。例えば、図 1 の画面遷移図から何も考慮を入れずにテストケースを生成すると次のテストケースが生成される可能性がある。

S→P1→P2→P1→P2→P4→P3→P2→P3→E

このテストケースは実際のシステム利用シーンとは異なっている。さらにツール利用者にとって理解しやすいものではない。そこで、本ツールではツール利用者が記述する遷移イベントの優先度情報(Main, Sub)を利用して、1 つのテストケース内で Sub 遷移の数をなるべく減らしたテストケースを生成している。そのため、実際のシステム利用シーンに近く、ツール利用者にとって理解しやすいテストケースとなっている。例えば、図 1 の画面遷移情報を入力とした場合、本ツールは表 4 の 4 つテストケースを生成する。テストケース数は増えているが、1 つずつのテストケースが利用者にとって理解しやすくなっ

ている。

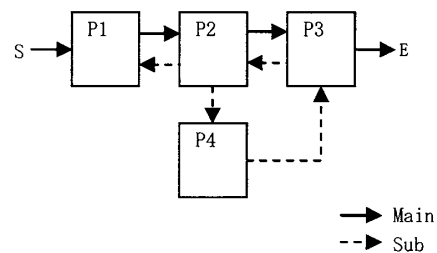


図 1: 画面遷移図

テストケース	画面遷移パス
1	S→P1→P2→P3→E
2	S→P1→P2→P3→P2→P3→E
3	S→P1→P2→P4→P3→E
4	S→P1→P2→P1→P2→P3→E

表 4: テストケースサンプル

2-4. 導入・評価

典型的な業務用ウェブアプリケーションに適用し、導入コストが低いことと本ツールにより生成されたテスト設計書の品質が一定であることを確認した。

まず、導入のために既存の設計書を書き換える必要があった。しかし、入力項目仕様は自由記述されていた仕様を表形式に書き換えるだけであり、画面遷移仕様は既存の画面遷移図の遷移イベント情報に Main または Sub の記述をするだけであるため、導入のためのコストは非常に少ないことを確認した。

また、既存のテスト設計書と本ツールにより生成されたテスト設計書を比較した。既存のテスト設計書は設計者のスキルによりテストケースの属人性が高く品質が一定でないのに対し、本ツールにより生成されたテスト設計書は属人性を排除したため、品質が一定で網羅性も高いことを確認した。

3. おわりに

ソフトウェアの品質確保のため、ソフトウェアの設計書からテスト設計書を生成するツールを試作、導入評価を行った。品質が一定なテスト設計書を得ることができ、導入コストが低いことを確認した。

参考文献

- [1] Utting, M. & Legeard, B. (2007) Practical Model Based Testing: A Tools Approach. Morgan Kaufmann, San Francisco.
- [2] Conformiq Qtronic™ Script Generation (SG) <http://www.conformiq.com/qtronic.php>