

テスト技法を適用する際に支障となる事項とその解決策

Obstacles in applying the test technique and their Solutions

主査 : 奥村 有紀子 (有限会社デバッグ研究所)
副主査 : 堀田 文明 (有限会社デバッグ研究所)
副主査 : 秋山 浩一 (富士ゼロックス株式会社)
リーダー : 細野 隆章 (キヤノンファインテック株式会社)
研究員 : 窪田 邦夫 (カルソニックカンセイ株式会社)
 吉田 亮平 (株式会社東京ビジネスソリューション)
 北川 尚理 (株式会社ユニケソフトウェアリサーチ)
 川田 修

概要

研究員の所属する組織では、ソフトウェアの機能が増大し複雑になっているにも関わらず、開発期間が短縮されテスト期間が十分にとれない状況である。そのため効率的にテストを行うための手段であるテスト技法を学び実業務への適用を試みる。しかし、テスト技法を学んでもなかなか実業務では適切に適用できず、作成したテストケースは人によりバラつきが発生する。

本研究では、実業務において適切にテスト技法を適用する際に、支障となる事項と除去方法をまとめ、テストケース作成における属人性を排除する条件を抽出した。

Abstract

In the organization of the researcher, it is a situation that does not take enough test period despite the fact that complicated function of software increases, development time is shortened. To seek to apply to the actual operations to learn test technique is a means for performing the test efficiently therefore. However, it can not be properly applied in the real business quite even learn testing techniques, variation is born in a test case that was created by people. In this study, when applying the test methods as appropriate in actual operations, and the combined removal methods and what the problem was extracted conditions to eliminate Zokujin resistance in the test case creation.

1. はじめに

研究員の所属する組織では、ユーザーの多種多様な要求にこたえるべくソフトウェアの機能が増大し複雑化している。その一方で、開発期間は短縮されテスト期間が十分に取れない。このような現状から効率よく欠陥を見つけるために、テスト技法の導入を試みている。欠陥とは要求された機能をコンポーネントまたはシステムに果たせなくする、コンポーネントまたはシステムの中の不備である[1]。しかし研究員からは、テスト技法導入のため技法を学んでも実業務で適用しようとする教科書通りにはいかず適切に適用することができないという声が挙がった。そのような状態で作成されたテストケースは、人によりバラつきが発生し、実行しても欠陥が発見できない恐れがある。このような背景から、本グループでは例題に対してテスト技法を適用し、実業務でテスト技法を適用する際に支障となる事項を抽出した。

本論文の構成は以下の通りである。2章では研究課題を述べる。3章ではテスト技法の調査と検証結果について説明する。4章ではまとめと考察を述べ、5章では今後の課題を述べる。

2. 研究課題

2.1. 前提条件

本研究を進めるにあたり以下を前提条件とした。

- ・ 対象者はテストの実務経験があり、テスト技法については概念や教科書程度の知識レベルでは適用方法を理解しているが、実業務では適用したことがないこととする。
- ・ 想定するテストレベル、テスト技法は研究員の業務内容から判断し、テストレベルは統合テストとする。統合テストのうち、機能間連携テストに目的を絞り、組合せのテスト技法である決定表テスト、状態遷移テストの2つを研究対象とする。機能間連携テストとは、例を挙げると機能 A と機能 B を組み合わせた際の動作テストとする。

2.2 現状と課題

研究員は、SQiP 研究会の分科会活動を通して同値分割、境界値分析を始めとする様々なテスト技法を学んだ。しかし実業務で技法適用を試みると、テキストに記載されていないが、技法適用時には必要となる情報を見つけ出すことが難しいケースが存在することが判明した。たとえば、状態変更とアクションの前後関係、論理構造がテストに影響するので、技法適用時にはそれらのことを考慮しなければならない。このようなことはテスト技法の入門書や初心者向けのセミナーに記載されているレベルのテスト技法の知識では考えつかない。この状態で無理矢理テスト技法を適用したとしても、見当違いなテストケースを作成することになりかねず、欠陥を見つけられないテストとなってしまふ。そのため、技法適用時に必要となる情報を見つけ出す力が必要となってくるが、同じテスト技法の知識を持った人であっても、経験・スキルが異なるとテスト技法を適用して作成されるテストケースにバラつきが発生する可能性がある。このような人によるバラつきを抑えるためには何が必要かを考える必要がある。

2.3 仮説とアプローチ

現状と課題を踏まえ、技法を適用して作成するテストケースのバラつきを抑えるためには何が必要なのかを研究員で検討を行った。その結果、バラつきの要因となる問題点を抽出し、その解決策を技法適用時の追加情報としてまとめ、技法とともにこの情報を適用することで、誰でも同じようなテストケースを作成できるのではないかと考えるに至った。

これらの追加情報はテスト技法の入門書や初心者向けのセミナーからでは得ることが出来ないため、実際にテスト技法を適用した結果から洗い出す必要がある。そこで本研究では、テスト対象とする簡易モデル（システムや機能）に対してテスト技法を適用する。その結果から必要と考えられる情報、条件等を抽出し、技法ごとにまとめ、その情報もあわせて利用することで、技法を適用して作成されたテストケースに対してバラつきを抑えることができるかの考察を行う。

3. テスト技法の調査と検証

3.1 状態遷移テスト

3.1.1 技法の解説

状態遷移テストは、テスト対象とするシステムのふるまいを状態遷移図と呼ばれるモデルや状態遷移表で表現し、実際の動作が図や表の記述通りであるかを検証するテスト技法である。

状態遷移図や表の記述は、システムの現在の状況を表す「状態」、ある状態から他の状態への移行を表す「遷移」、遷移が発生するためのトリガとなる入力を表す「イベント」を組み合わせて表現する。

この技法の適用対象はある特定のテストレベルに限ることは無く、何をシステムとして

捉えるかによってユニットテスト（単体テスト）レベルから受け入れテストレベルまで全てのテストレベルにおいて適用することが可能である。

状態遷移図はテスト技法の一種であると同時に、システムのふるまいと共に外部からの入力との繋がりを図式化して表現することが可能な設計手法でもあるため、設計段階で作成されている場合は自然言語による仕様記述のあいまいさに起因したバグの混入を防止するためのツールとしても有効である。

3.1.2 技法適用する上での問題点の抽出

状態遷移テストを適用する方法について簡単なテスト対象を想定し、そこに技法を適用することで、技法適用時に考慮すべき事項を抽出し、考察を行う。テスト対象は付録1【状態遷移テスト】の1-1.状態遷移テストに対するテスト対象の要求仕様、および図1、図2に示した歩行者用信号と車道側の信号に対する制御を使用した。

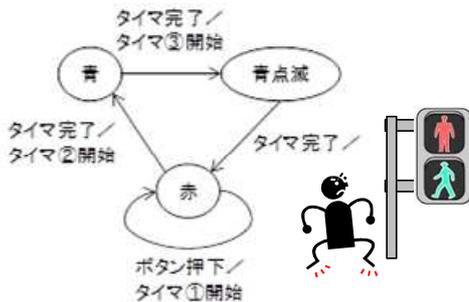


図1. 歩行者用信号の状態遷移図

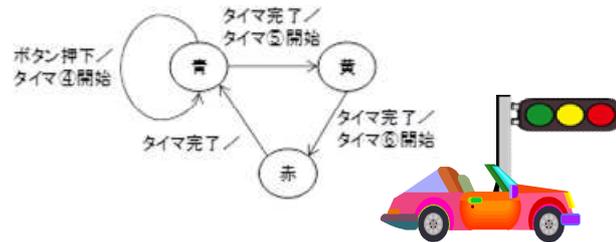


図2. 車道用信号の状態遷移図

車道用および歩行者用の各々の制御がコンポーネントとして実装され、車道用信号、歩行者用信号のテストを各々単独で実施後に統合テストを実施すると仮定する。これをテスト対象として状態遷移テスト技法適用の検証を実施した。

車道用信号で赤、青、黄の3種類の状態、歩行者用信号で赤、青、青点減の3種類の状態を取り得るため、組合せとしては9種類が存在することとなる。ただし、信号の状態として実際にはあり得ない組合せを除くと5種類の状態があり、それらの状態にイベントを割り付け、付図4の状態遷移図を作成する。また、状態遷移表で表現すると表1のようになる。表1に記載しているタイマ から の種類については付録1の1-1.状態遷移テストに対するテスト対象の要求仕様に記載している。

表1. 歩行者用信号と車道用信号を統合した状態遷移表

状態No.	状態		イベント	タイマ完了	
	歩行者用	車道用		歩行者用	車道用
1	赤	赤	-	タイマ 開始 No.3	No.4
2	青点減	赤	-	No.1	
3	青	赤	-	タイマ 開始 No.2	
4	赤	青	タイマ、タイマ 開始		タイマ 開始 No.5
5	赤	黄	-		タイマ 開始 No.1

上段:アクション
下段:遷移先の状態

今回のテスト対象においては歩行者用ボタン押下をトリガとして歩行者用信号、車道用信号の状態が変化し、また初期の状態に戻る制御となっており、起点から終点までを一つ

の流れで表現が可能となっている。イベントとしてボタン押下が発生した時の起点から終点までの状態の流れは付図5の通りである。また、時間軸に沿ってタイマと信号の状態をタイミングチャートで表すと付図6の通りとなる。この信号制御の事例ではボタン押下を起点として決まった順番通りに遷移が発生した後に最初の状態に戻っており、テストにおいてもその通りに遷移することを確認するが、それ以外にも状態遷移表を元にして各状態とイベントの組合せを網羅したテストを実施し、想定外のイベントで遷移しないことや想定外の遷移先に遷移しないことについても確認を行う必要がある。

一例としてボタン押下というイベントは、表1の状態 No. 4の状態以外のいずれのタイミングにおいても発生可能であり、他の全ての状態においてボタン押下が発生した場合に状態遷移に影響しないことのテストが必要である。また、実際のシステムにおいては突然の電源喪失やスイッチの故障等の外乱が発生することもあり得るため、それらを想定したテストも必要である。外乱の発生もボタンの押下と同様にどの状態においても発生することを考慮する必要がある。

信号制御をテスト対象として状態遷移表からテストケースを設計しようとした場合にどのような点がバラツキの原因となるかを検討し、テストの正確性・網羅性・効率性を考慮して抽出した結果を以下に列挙する。

- 1) テストベース¹の情報の不足・不完全
 - (1-1) 状態とイベントの抽出が不足しているまたは適切ではない
 - (1-2) イベント発生タイミングが表現されていない
 - (1-3) 状態変数の場所の説明がない
- 2) 状態遷移図・遷移表の記述が基準化されていない
 - (2-1) 状態遷移図の記述方法
 - (2-2) 状態遷移表の記述方法
 - (2-3) 遷移前状態×遷移後状態，遷移前状態×イベント等
- 3) テスト基準が定まっていない，適切ではない
 - (3-1) Nスイッチテスト²でN数の決定基準がない
 - (3-2) 状態とイベントの組合せについての確認基準がない

3.1.3 状態遷移テスト技法適用の支障事項考察

前項の-1)から-3)で述べたテスト技法適用の支障に対する解決策について考察し、以下にまとめる。

- 1) テストベースの情報の不足・不完全に対する解決策
 - (1-1) 状態とイベントを漏れなく洗い出す

機能を実現するために必要となる状態とイベントは仕様書には記述されているが、実際にはノイズの混入や予期しない電源の遮断等の想定外のイベントが発生することもある。そのような仕様書に記載されていないが実際には発生が予想されるイベントについても技法適用時に洗い出すことが必要である。その際、イベント抽出の着眼点をチーム内で共有し、担当者間によるイベント抽出のバラつきを防止すること。

状態は、実際のソフトウェアでは内部変数の値として保持する場合が多い。この時に仕様書で表現されている状態と内部変数の値との関係を確認しておくこと。
- (1-2) イベント発生タイミングを考慮する

信号の「ボタン押下とタイマ完了」をイベントとしていること、「ノイズやチャタ

¹ テストベースとはテスト対象となるコンポーネントやシステムの要件を推測できる全てのドキュメント類である。テストベースの情報を元にテスト設計を行う。

² NスイッチテストとはN + 1遷移の全ての有効なシーケンスのパスをテストする状態遷移テストの一形式である。

リングもイベントとするかどうか」等，テストを実施するために必要な情報が，仕様書には記載されていない場合もある．正しくテストを行うためには実際の構造や設計方針を参照すること．

同じイベントを使用している場合でも，個々のモジュールが別の CPU により非同期で動作している場合や，通信によってイベントが通知される場合の通信タイミングの遅れ等も考えられるため，コンポーネントテストや統合テストにおいても対象となる機能の設計情報だけでなくシステム全体の設計情報も参照してテスト設計を行うこと．

(1-3) 状態変数の保持方法を確認する

状態変数をどのモジュールで保持しているかを確認しておく．今回の例では，統合テスト時に使用する状態変数が車道用信号と歩行者用信号の各々で保持されている場合や，それらの情報を元に全体の状態変数が別に存在することが考えられる．どの値を参照してテストすべきかを検討してテストを設計すること．

コンポーネントテストにおいて他のコンポーネントの状態を参照している場合は，テスト実行の際にドライバやスタブが必要となることがあるため，それらについてもテスト設計時に必要な情報として抽出しておくこと．

-2) 状態遷移図・遷移表の記述が基準化されていないことへの解決策

(2-1) 状態遷移図の記述方法

状態遷移図で多く使用される表記形式としてミーリ型やムーア型などがある．仕様書に状態遷移図を記載する場合は表記形式や状態遷移図内に記載する情報を事前にチーム内で決定し統一しておくこと．

ムーア型では状態とイベントで状態遷移を表現し，ミーリ型では他の状態へ遷移するためのイベントと共に実行されるアクションを併記するが，どちらの場合もアクション実行中にイベントが発生する可能性がある時はアクション実行中も一つの状態として取り扱うことが必要となる．状態遷移図の記載方法には，前述の 2 形式以外にハレル型という階層化された状態の表現が可能な表記方法もある．

(2-2) 状態遷移表の記述方法

状態遷移図と同様に表についても事前状態×事後状態，事前状態×イベント等記述方法を事前に定義しておくこと．

-3) テスト基準が定まっていない，適切ではないことへの解決策

(3-1) NスイッチテストでのN値の決定方法

NスイッチテストではN値が大きくなるほどに網羅性が上がると考えがちであるが必ずしもそうなるとは限らない．また，N値が大きくなるとテストケースの数は増加することが多く，N値を決定する際には注意が必要となる．

参考文献[2]では最初に状態遷移表により全ての状態とイベントの組合せをテストしてから1スイッチテストを実施し，その後2スイッチテストを実施すること，および，状態遷移テストで全ての遷移パターンをテストすることが困難な場合はユーザーの操作や外部の状況の変化を想定したシナリオを作成し，シナリオテストによって典型的なパターンを確認する等，他のテスト手法との併用を推奨している．

今回テスト対象とした信号機制御程度であれば単純な状態遷移であったが，実際にテスト対象となるシステムは複雑な遷移を持つことが多いため，[2]に従い，他のテスト手法を併用し，テストケースの増加を抑える工夫が必要と考える．

(3-2) 状態とイベントの組合せの網羅基準

状態とイベントの組合せについては，考えられる全ての組合せで確認すべきと考える．理由としては，仕様書に記載される状態遷移図では想定外のイベントの発生

や状態が表現されていないことが多く、その場合のテストの抜け、漏れを防止するためである。

個々の組合せの確認はNスイッチテストの項で述べた通り、Nスイッチテストの前に実施することで基本的な遷移の確認を早期に行うようにする。

3.2 決定表テスト

3.2.1 技法の解説

決定表テストは、組合せによるテスト設計技法の1つである。デシジョンテーブル（以下「決定表」と記述）にある入力と原因の組合せを実行するテストケースを設計する。決定表とは、入力(=原因)の組合せと出力(=結果)の関係を示す表である。

仕様を表現する決定表は、規則間の関係性によって単適合決定表と多重適合決定表に分けられる。単適合決定表とは、条件の組合せのどの1つも、ただ1つの規則においてだけ満たされる決定表である。規則の優先順位によって1番から順に処理を実行し、成立すれば以降の規則を処理しない。多重適合決定表とは、条件の組合せのうち少なくとも1つが、2つ以上の規則によって満たされる決定表である[5]。決定表のルールは1番から順に最後まで処理を実行し、それぞれの規則(判定)に従い多重に処理を行う。プログラム論理に応じて、単適合決定表と多重適合決定表を適切に使い分けし、また、決定表間を連結することで、論理構造の見える化が実現できる。この見える化により合理的なテストが実現できる。なお、テストの決定表は仕様の決定表をベースに作成するが、仕様の決定表とは分けて作成し、単適合、多重適合の区別はない。

3.2.2 技法適用する上での問題点の抽出

決定表を適用する方法について簡単なテスト対象を想定し、そこに技法を適用することで、顕在化する考慮すべき事項について考察を行う。テスト対象は付録2の2-1.決定表テストに対するテスト対象の仕様に示したメニュー画面とした。

今回の仕様からテストケースを設計しようとした場合にどのような点が妨げになるかを議論した。テストの正確性・網羅性・効率性を考慮して選択した結果、次の項目が抽出できた。

- 1)テスト技法適用の前提となる仕様の表記が基準化されていない
 - (1-1)仕様が自然言語のみで記述されている。今回の仕様は比較的単純なケースだが、条件の組合せが多くなると仕様の抜け漏れを見つけることが難しい。
- 2)テスト範囲の選定基準がない
 - (2-1)仕様の決定表が記載されていないため論理構造が把握できない。そのためテスト範囲を適切に選定することができない。
- 3)単体テストの網羅基準が設定されていない
 - (3-1)単体テストではすべての条件の組合せを網羅することを基準にしておく必要がある。また、論理構造設計で適切な分割を実施しておかないと、一つの決定表に記入する条件の組合せや動作の組合せが多くなり、網羅が困難になる。
- 4)統合テストの網羅基準が設定されていない
 - (4-1)テスト範囲が広く、入力条件が多くなる統合テストではすべての条件を組み合わせるテストケース設計方法を採用すると、テストケース数が膨大となる。単体テストと補完し合いながら漏れないテストを実現する基準が必要である。

3.2.3 決定表テスト技法適用の支障事項考察

前項の-1)から-4)で述べたテスト技法適用の支障に対する解決策について考察し、以下にまとめる。

- 1)テスト技法適用の前提となる仕様の表記が基準化されていないことへの解決策

- (1-1)仕様策定時に条件の組合せと動作を洗い出し、仕様書に仕様の決定表を記載することによって、複雑な入力の場合をわかりやすく整理し、設計段階で仕様の抜け漏れを防止する。
- 2)テスト範囲の選定基準がないことへの解決策
 - (2-1)システムからコンポーネントまで、各階層での実装状態が決定表に記載されていることにより、各テストレベルでのテスト範囲を決定表より選定する。
- 3)単体テストの網羅基準が設定されていないことへの解決策
 - (3-1)統合テストのテストケース数の増加を防止する前提として、単体テストでは条件の組合せを網羅することが必要である。したがって、単適合決定表、多重適合決定表からのテスト決定表展開においては入力条件、および出力の組み合わせが網羅できるように基準を設定し、また、それぞれの仕様決定表からテスト決定表を展開するためのガイドを用意する。
- 4)統合テストの網羅基準が設定されていないことへの解決策
 - (4-1)統合テストでは広い範囲をテストするため入力条件が多くなる。統合テストは、単体テストの網羅基準を利用して組合せを削減すべきである。統合テストでは単体テストでカバーする個々の決定表の出力の組合せに注目した網羅基準を設定する。例えば判定網羅を基準に出力を組み合わせるなどである。

上記の解決策を、3.2.2で用いたテスト対象に対して決定表テストに適用した例を示す。権限別の決定表は今回の仕様から判断し単適合決定表とし、部署ごとの決定表は多重適合決定表とする。それぞれの決定表を表2と表3に示す。

表2．権限別の仕様の決定表

規則(単適合)		1	2	3
原因	管理者	Y	N	N
	承認者	N	Y	N
	一般	N	N	Y
結果	管理者メニュー	Y	N	N
	承認者メニュー	N	Y	N
	一般メニュー	N	N	Y

表3．部署ごとの仕様の決定表

規則(多重適合)		1	2	3
原因	人事部	Y		
	経理部		Y	
	情報システム部			Y
結果	人事部メニュー	Y		
	経理部メニュー		Y	
	情報システム部メニュー			Y

次に、仕様の決定表からテストの決定表を作成した例を示す。仕様の決定表からテストの決定表を作成する場合は、単適合決定表と多重適合決定表の論理フローを利用して作成する。各判定はThenとelseの分岐であり、判定網羅でテストする。

表4．権限別のテストの決定表

テストケース		1	2	3
原因	管理者	Y	N	N
	承認者	N	Y	N
	一般	N	N	Y
結果	管理者メニュー	Y	N	N
	承認者メニュー	N	Y	N
	一般メニュー	N	N	Y

表5．部署ごとのテストの決定表

テストケース		1	2
原因	人事部	Y	N
	経理部	Y	N
	情報システム部	Y	N
結果	人事部メニュー	Y	N
	経理部メニュー	Y	N
	情報システム部メニュー	Y	N

最後に、表4の決定表と表5の決定表を結合する。なお、結合する際はテストケースの多い決定表に少ない方を移す。結合する際の基準は、判定網羅+個別動作とした。決定表を表6に示す。

表6. 「判定網羅 + 個別動作」基準で結合した決定表

テストケース		1	2	3	4	5	
原因	権限	管理者	Y	N	N	Y	N
		承認者	N	Y	N	N	Y
		一般	N	N	Y	N	N
	所属部署	人事部	Y	N	Y	N	N
		経理部	Y	N	N	Y	N
		情報システム部	Y	N	N	N	Y
結果	管理者メニュー	Y	N	N	Y	N	
	承認者メニュー	N	Y	N	N	Y	
	一般メニュー	N	N	Y	N	N	
	人事部メニュー	Y	N	Y	N	N	
	経理部メニュー	Y	N	N	Y	N	
	情報システム部メニュー	Y	N	N	N	Y	

4. テスト技法適用に対する支障の除去についてのまとめと考察

本研究では、テスト技法の入門書や初心者向けのセミナーを受講した段階で実業務への技法適用を試みた場合に支障となる事項を取り上げ、その解決策を検討・抽出した。

-1) テストベースとなる情報の正確性

(1-1) 状態遷移では、機能を実現するための状態、イベントのみではなく発生し得る状態、イベントを考慮して出し尽くしていること。内部変数や構造に基づいて定義されていること。

(1-2) 決定表では、仕様が自然言語だけでなく決定表で記載されていること。仕様の決定表が単適合決定表、多重適合決定表として作成されていること。

-2) 設計そのもののありかた

(2-1) 状態遷移では、状態遷移表を使用して全ての状態とイベントの組合せにおいてシステムのふるまいに問題が無いかを検討すること。アクション実行中など、イベントの発生タイミングについても検討を行うこと。

(2-2) 決定表では、適切な大きさに分割された論理構造であること。仕様の決定表と異なる順序で実装しないことが必要であり、さらには実装論理を追加した仕様の決定表が作成されると、正確かつ網羅度の高いテストが可能となる。

-3) 上記を実行するためには

(3-1) 状態遷移図や表、決定表は設計ツールとしても有用であるが設計時には実現したい機能に着目して記述することが多く、テストの観点を含めてテストベースとして十分であるかを関係者間で確認すること。

5. 今後の課題

本研究では、テスト対象に簡単なモデルを想定してテスト技法を適用したにすぎず、検証が不十分であった。そのため各研究員が所属する組織で実際にテスト技法を適用すること、また、技法適用に際し現状との工数を比較し、解決策の有効性を検証する必要がある。

6. 参考文献

- [1] ドロシー・グラハム ソフトウェアテストの基礎 株式会社ビー・エヌ・エヌ新社(2008-8)
- [2] 秋山浩一 ソフトウェア技法ドリル 株式会社日科技連出版社(2010-5)
- [3] リー・コーブランド はじめて学ぶソフトウェアのテスト技法 日経 BP 出版センター(2005-11)
- [4] ポーリス・バイザー ソフトウェアテスト技法 日経 BP 出版センター (1994-2)
- [5] JIS X 0125:1986 決定表