

UML を用いた試験仕様作成手法

6H-3

新井本 武士、田村 一賢、岡本 渉、平山 雅之
株式会社 東芝 研究開発センター システム技術ラボラトリ

1. はじめに

ソフトウェアシステムの品質を確保する手段としては、ソフトウェア開発プロセスの改善、設計における品質の作りこみ、テストによる品質の確保等が考えられる。中でもシステムが要求仕様どおりに動作することを確認するシステムテストは製品リリース前の最後の手段であり、品質を確保・保証するためには特に重要な作業である。

本報告では、UML をもとにシステムのテスト項目を効率的かつ高い詳細度で生成する手法について報告する。

2. システムテストにおける課題

近年、ソフトウェアシステムの大規模・複雑化によって、システムテストに以下のような課題が生じている。

(1) 試験仕様における網羅性

近年の大規模・複雑化したソフトウェアシステムにおいてはその機能構造を把握するのは容易ではない。そのため、システムの全機能・全操作を漏れなく網羅する試験仕様の作成が困難となってきている。

(2) 試験仕様作成・テスト作業におけるスキル依存

大規模・複雑化したシステムでは通常複数人によって試験仕様を作成する。この場合、試験仕様の質は担当者のスキルに大きく依存する。

上記課題の多くは、そもそもテスト項目を考える際に参考にする上流工程でのシステム仕様記述の曖昧さに起因している。このため、状態遷移図などの形式的仕様記述を用いて仕様を記述することで、上記課題を解決しようとするアプローチが少なくなかった。

しかし、実際の開発現場で、形式的仕様記述を使いこなし、対象製品の全ての仕様を整備して、これをもとにテスト項目を作成する作業は、あまりに煩雑であり、コスト的にも見合わないため十分には普及していない。

4. 手法概要

4.1 基本コンセプト

そこで我々は、比較的容易に導入が可能な手段で、

① 網羅性向上：対象とするシステムあるいは機能に対して漏れなくテストできるテスト項目を効率的に作成する。

② スキル依存性排除：経験の有無によらず、誰が利用しても適切なテスト項目を作成できる。

ことを目標とする試験仕様作成手法を開発した。

この手法の特徴は以下のとおりである。

特徴-1: 試験仕様作成の前提となる仕様記述に UML（アクティビティ図、ユースケース図）を導入する。これによって前提となる仕様記述自身の曖昧さを排除する。

特徴-2: UML により対象システムの機能の振る舞いや構

“Generating Test Cases Using UML diagrams”,

Takeshi Niimoto, Kazuyoshi Tamura, Wataru Okamoto, Masayuki Hirayama, TOSHIBA corp. R&D Center

e-mail:takeshi.niimoto@toshiba.co.jp

造を階層的に整理することで、テスト項目抽出の抜けや個人差を少なくする。

特徴-3: テストケース導出観点およびテストケース導出手順を明確にし、各テスト項目に関するテスト仕様作成段階でも抜けをなくし、個人差を少なくする。

開発した試験仕様作成手法は上記のような特徴をもつことによって、テスト対象となるシステムや機能について、必要な全てのテスト項目を抽出し、網羅性の高い試験仕様を作成することを可能としている。また同時に対象システムや機能に関する経験が必ずしも十分でなくとも、適切な試験仕様を作成することができ、テスト項目作成に関するスキル依存性を低く抑えることができる。

また、ユースケース図・アクティビティ図は UML の中でも比較的記述の容易な図であり、記述の制約も緩くしていることから、本手法は形式的手法と比べて低いコストで導入することも可能となる。

4.2 手法詳細

4.2.1 UML 記述作成

ユースケース図、アクティビティ図では、図中の要素を独立した図（サブ図）を用いて詳細に記述することによって階層構造を導入し、対象システムの機能構造の把握を容易にしている（図 2）。これによりユースケース、アクティビティを、見易さを損なうことなく詳細に記述し、仕様の漏れ・抜けを抑制することができる。その際、以下のガイドラインに沿って記述する。

- ・複数のユースケース内の共通する部分を独立させ、ユースケースをモジュール化する。
- ・複数の操作からなるアクティビティは、階層化してサブ図を描く。このとき、アクティビティが全て単一の操作と一対一に対応するようになるまで詳細化する。

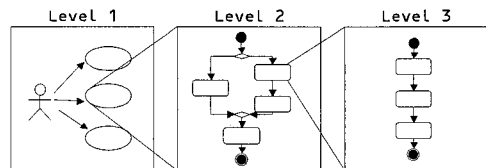


図 1：UML 記述の階層構造

4.2.2 試験仕様作成

テストケースの導出は以下の手順で行う：

(1) テスト項目抽出：UML 記述からアクティビティの階層構造を表形式に書き出すことにより視覚化、操作を抽出する

(2) テスト仕様作成：各操作について前提条件と期待される結果を網羅することによりテスト仕様を作成

(1) テスト項目抽出

図 2 に示したように、前節で作成した UML 記述は各ユースケースをルートとしたツリー構造をなしている。まず、これを表形式に書き出し、ツリー構造の葉にあたるアクティビティを抽出する。次に、それらを実現する操作を機能

操作仕様書を基に抽出し、アクティビティ階層表を作成する。表1に患者の検査情報を管理する病院情報システムの一部について作成したアクティビティ階層表の例を示す。ここで図1の3つダイアグラムの内、Level1の要素が列「ユースケース」に、Level2の要素が列「アクティビティ1」、Level3の要素が列「アクティビティ2」にそれぞれ列挙される。ここでアクティビティ2が個々のテスト対象の項目(テスト項目)となる。

(2) テストケース作成

上記で抽出されたテスト項目である各操作に対し、実行前に起こりうる状態とその実行結果を網羅することによってテストケースを作成する。その際、対象システムの特徴を考慮したテストケース導出観点を留意し、これを基にテストケースを導出する。テストケース導出観点とは、対象操作を実行する際のシステム/データの状態を網羅するための観点および基準を明示したものである。通常、以下の2つを対象システムに応じて詳細化し、明示する：

- (1) システムについて取得可能な状態
- (2) 対象操作が使用/定義を行うデータの状態

表2に具体的なテストケース導出観点を示す。また、表1で抽出した操作に対して、表2およびシステムの仕様書を基に作成した試験仕様書の例を表3に示す。

例えば、表1のアクティビティA(メニューから[オーダー]-[メニュー削除]を選択する)については、表2のテストケース導出観点B(表示状態)を適用し、これと仕様書の情報をもとに、表3の試験項目群Cに示されるような試験仕様を作成することができる。

5. 適用評価

提案手法を弊社で開発した情報管理システムのテストの一部に適用し、評価を行った。対象システムはVisualBasicによって開発されたWindowsアプリケーションで、規模は約100,000Stepsである。適用においては直接開発には関係していないテスト担当者が、仕様についてインタビューを行った上で本手法によってテスト仕様書を作成、テストを実施した。なお、本手法によるテストは通常のシステム試験後に再確認のテストとして行った。

5.1. 適用結果

(1) 不具合の検出状況

今回、適用部分では全10件の不具合が発見された。このうち2件を本手法で検出した。今回のテストは通常試験後の再確認テストであり、従来手法によって漏れた不具合を確実に発見したことになる。

(2) 詳細度の比較

従来手法は、自然言語で書かれた機能操作仕様書から試験仕様書を作成していた。このため機能構造の把握が難しく、テスト項目の粒度が大きくなる傾向があった。一方、本手法では対象機能の振る舞いを詳細に分析したアクティビティ図の個々のリーフレベルに対応したテスト項目が作成できた。このためテスト項目の粒度は、従来に比べてかなり詳細になっている。

5.2. 提案手法の効果

上記の結果から、提案手法には以下の効果があるものと考えられる：

(1) 網羅性の向上

従来手法よりも試験仕様の詳細度が高いことと、特徴-1,2に示した仕様記述へのUMLの利用およびテストケ

表1: アクティビティ階層表

ユースケース(Level1)	アクティビティ1(Level2)	アクティビティ2(Level3)	操作
ユースケース2	検査オーダーを削除する	2-1オーダーを指定する	(ユースケース3)
		2-2-1メニューから[オーダー]	メニューから[オーダー]
		-[オーダー削除]を選択する	-[オーダー削除]を選択する
		2-2-2削除を確認する	「はい」ボタンを押す
			①削除理由項目をクリックする
			②「確認」ボタンを押す
			削除理由項目をダブルクリックする
			「閉じる」ボタンを押す

表2: テストケース導出観点

観点	基準
ウィンドウ	どのウィンドウにおける操作なのか。 (例)・ウィンドウ名
画面状態	画面の表示状態。目に見える範囲で、画面の表示状態を網羅する。 (例)・表示モード ・表示しているデータの種類の ・遷移状態
データ状態	操作に係るデータの状態について、仕様上操作の結果に影響するデータの状態を網羅する。また、データの入力を要する場合は、件数、数値等の境界値に付いては試験を行うものとする。 (例)・操作対象となるデータ属性の値、件数 ・操作対象データが存在する/しない

表3: 試験仕様書

ユースケース	前提条件	期待される結果	アクティビティ(導出)
検査オーダーを削除する	検査オーダーを指定する		(ユースケース2)
	検査対象業務メイン画面において 一覧表示モード のとき	メニューから[オーダー] -[オーダー削除]を選択する	アクティビティ-2-1
	検査対象業務メイン画面において オーダー情報表示モード のとき	メニューから[オーダー] -[オーダー削除]を選択する	アクティビティ-2-1
	オーダー削除確認ウィンドウ において	「はい」ボタンを押す	アクティビティ-2-1
	オーダー削除確認ウィンドウ において	①理由項目をクリックする ②「確認」ボタンを押す	アクティビティ-2-2
	オーダー削除理由選択ウィ ンドウにおいて	「閉じる」ボタンを押す	アクティビティ-2-2
	オーダー削除理由選択ウィ ンドウにおいて	「閉じる」ボタンを押す	アクティビティ-2-2

ース導出手順の明確化の効果を考え合わせると、本手法の適用により試験項目の網羅性が向上したと考えられる。

(2) スキル依存の解消

今回の適用では試験仕様の作成、テスト作業ともにシステムの開発に直接関係のない人間が行った。これを従来のテスト仕様と比べても遜色がないことを確認している。これらを考慮すると、本手法の適用によりスキルに依存しないテスト仕様を作成することができたと考えられる。

4. まとめ

UMLのユースケース図・アクティビティ図を利用してシステム試験仕様を導出する手法を提案した。また、実際のソフトウェアシステムのテストに本手法を適用することによって、テスト項目が詳細化されること、担当者のスキルに依存しない試験仕様の作成、テスト作業の実行が可能であること等を確認した。

今後は上記の各効果の定量的な評価、再利用性向上における効果の確認、一部手法の自動化等を行う予定である。

参考文献:

- [1] 平山 雅之, 山本 徹也, 岡安 二郎 他: “機能モジュールに対する優先度に基づいた選択的ソフトテスト手法の提案”, 信学会 SS 研 01-5
- [2] グラディ・ブーチ: “UML ユーザーガイド”, ビアソン・エデュケーション, 2001