システム分析結果を活用した論理網羅テストの方法

7K－5

藤井　諭
松江工業高等専門学校

１．はじめに

ソフトウェアは大規模化、複雑化しながら短納期化し、理解し易くかつ信頼性の高いブラックボックステストが求められている。本稿では、システムの要求定義で作成したＤＦＤから、ブラックボックスでのシステムテストのためのテストケースを設計する方法について述べる。

２．背景

従来よりＣＡＳＥによるソフトウェアの信頼性向上に取り組み、その一環として上流支援ツールの実プロジェクトへの適用も行ってきた。1) 経験的に、システムの分析・設計に構造化手法2) は初心者にも記述しやすく、また第三者にも理解し易い。

一方で下流工程においても、ソフトウェアは大規模化、複雑化の一方で短納期化し、より効率的なテスト方法が求められている。Ｍｙｅｒｓのテストケース設計のスペクトラムに当てはめると、仕様テストになるべく近いブラックボックステストの方法がより望ましい。すなわち仕様テストのためのより良いテストケースの設計方法が必要である。

３．方法

テストケースの設計のために、構造化分析によって作成したコンテキストダイアグラム（ＣＴＸ）、最上位のデータフローダイアグラム（ＤＦＤ０）とデータディクショナリ（ＤＤ）を用いる。まずＣＴＸのターミネータとシステムのやりとりに着目し、ＤＦＤ０まで使用して手順を設定し、分岐条件はＤＤのＯＲ条件から定める。これはＣＴＸやＤＦＤの平面情報に時間情報を加えることによる、立体情報への変換と言うこともできる。本方法では最上位のＤＦＤ０までしか使用しないが、システムの規模によってはＤＦＤ１以降を使うのはかまわない。しかしミニスペックまで降りてしまうと、実装方法とのかかわりが深くなりブラックボックステストと言えなくなるので、適度なレベルで打ち切る必要がある。

作成したテストケースでダイアグラムの塗り潰しを行う。塗り潰しによって、命令網羅度（Ｃ０）と判定条件網羅度（Ｃ１）を使った測定を行う。網羅度は一般にはホワイトボックステストで使用される。3) しかし本方法のように仕様定義のためのＣＴＸやＤＦＤを使う場合は実装方法に関与しないため、ブラックボックスでの論理網羅度と言える。モジュールテストにおいてＣ０は

Ｃ０＝実行ボックス数／全ボックス数

であるが、これをＤＦＤに投影する場合は、データフローやコントロールフローもボックスのひとつと見て定義する。また、分岐条件と分岐数は、ＤＤの中のＯＲ条件によって定義する。

Ｃ０＝通過フロー数／全フロー数　（1）

Ｃ１＝実行分岐数／全分岐数　（2）

４．実験

旅費精算システム4) を構造化分析し、ＣＴＸ、ＤＦＤ０およびＤＤの作成を行った。社員が出張終了後、旅費を精算する場合のＣＴＸの一部を図１に、ＤＦＤ０の一部を図２に示す。

［社員が旅費を精算する］テストケースの作成を行う手順の例を示す。

①ＣＴＸのターミネータ「社員」とシステム「旅費精算システム」との間で関連するデータフローとコントロールフローを抽出し、呼び出される順番を設

A Method of Logical Coverage Testing used System Analysis Documents
Satoru Fujii
Matsue National College of Technology
14-4 Nishi-ikuma,Matsue,Shimane 690,Japan

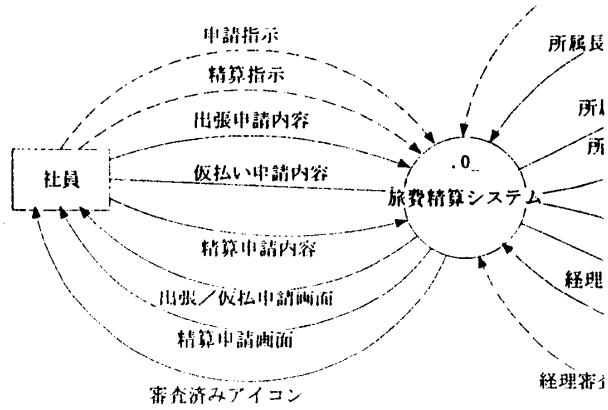

図1　コンテキストダイアグラム

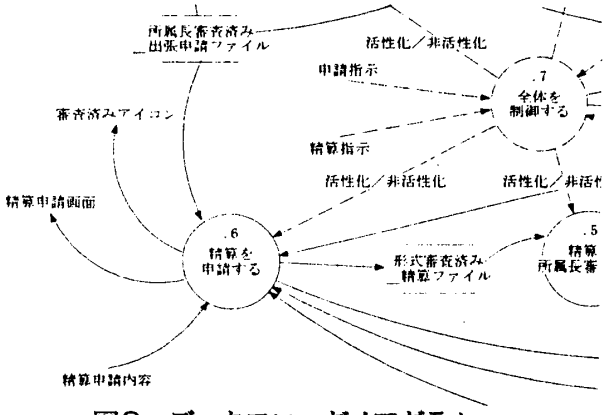

図2　データフローダイアグラム

定する。　審査済みアイコン→精算指示
→精算申請画面→精算申請内容→精算指示
②ＤＦＤ０と照合し、「審査済みアイコン」の発生元プロセス「精算を申請する」を活性化する。
③「審査済みアイコン」の発生要因はデータストア「所属長審査済み_出張申請ファイル」と「経理審査済み_仮払い申請ファイル」が考えられ、２つのテストケースに分けて、データストア「経理審査済み_仮払い申請ファイル」に特定する。
④ＤＤの精算申請画面＝［申請一覧｜伝票］　から、まず申請一覧を表示させる。
⑤プロセス「精算を申請する」で「社員」から送られた精算申請内容を、データストア「形式審査済み_精算ファイル」に書き込ませる。
⑥画面、アイコン等の後処理はＤＦＤ０のコントロールプロセスに従って記述する。

作成した全テストケースによってＣＴＸ、ＤＦＤ０およびＤＤの塗りつぶしを行う。ＤＤは［ＯＲ条件］で使った方を塗りつぶす。これを（１）式にあてはめる時、重複した通過フローは数えない。（２）式にあてはめる時、全分岐数はＤＤの全ての［ＯＲ条件］の和と考える。測定結果は次の通りである。

テストケース数＝１０	テストケース数＝１３
Ｃ０＝８８．２％	Ｃ０＝１００％
Ｃ１＝９５．１％	Ｃ１＝１００％

５．考察

Ｍｙｅｒｓの「機能テストの最小限の基準は、防衛的プログラミング用以外のすべての分岐を、少なくとも１度ずつすべての方向に実行すること」を基準とすると、テストケース数＝１０ではテストが不十分であることが明確である。しかしテストケース数＝１３でＣ０、Ｃ１＝１００％が得られ、機能テストの最小限の基準を満たす効率的な設計方法であると考える。ただしＤＤの複数の［ＯＲ条件］の組み合わせによっては、仕様テストの漏れが起こりうる。この漏れをチェックする方法としては、条件網羅度（Ｃ２）での測定が考えられる。

６．おわりに

要求仕様から最低限の基準を満たしたテストケースを作成して論理網羅テストを評価する方法について述べた。

今後の課題として、条件網羅度Ｃ２まで拡張したテストケースの生成法まで拡張して検討したい。また、状態遷移図などと組み合わせることによって、より効率的なテストケースの設計法への改良が考えられる。

参考文献

1)藤井他：「ソフトウェア開発支援システムＳＤＳＳにおけるＣＡＳＥ統合化」，情報処理学会論文誌 Vol.36 No.1 (1995)
2)Paul T. Ward：「Structured Development for Real-Time Systems」，Prentice Hall/Yourdon press (1985)
3) G.J.マイヤーズ（松尾正信訳）：「ソフトウェア・テストの技法」，近代科学社（1992）
4)シグマシステム：「オブジェクト指向技術検討部会報告書（WG2）」，（株）シグマシステム（1993）