

1M-6

CAPEを入力とした
 テスティング自動化へのアプローチ

倉谷 正勝
 (富士通株式会社)

1. はじめに

ソフトウェア開発のテスト工程における、生産性・品質向上への取り組みは近年、様々な研究がなされている。ソースプログラムにもとづいたアプローチはパス解析等実用化されているが、プログラム機能仕様書(設計書)にもとづいたアプローチは具体的な成果をもたらしていない。従来のテスト作業の中で、テスト項目の選定、テストデータの作成は作成者の経験と勘に依存される場合が多いため、品質が左右され、工数もかかっていた。従って、ソフトウェア開発におけるテスト工程での生産性・品質向上への取り組みを考えると、プログラム機能仕様書にもとづいた

- ①実用的な手法・ツール
- ②作業の標準化

が必要となる。

以上の背景により今回、プログラム機能仕様書から完全な機能テストに近づけるための効果的なテスト項目及びテストデータを自動生成するツールの開発を行った。当論文では自動化における着眼点、ツールの機能及びその効果について述べる。なお、当ツールはプログラムの単体テストをサポートしている。

2. 自動化へのアプローチ

2.1 従来の問題点

プログラム機能仕様書は記述形式は標準化されているが、記述内容は作成者によって表現が曖昧になりがちであった。このような機能仕様書にもとづいたテスト項目選定は、テスト項目の漏れ・重複といった品質の問題を常にかかえていた。

2.2 「CAPE」*1の出現

先に述べた機能仕様書の問題を解決するために当社が提供している簡易言語「CAPE」に着目した。

「CAPE」とは当社が「ユーザアプリケーション開発のトータル生産性向上」を目的に提供している基幹業務開発ツール「CASE T」*2のプログラム設計情報を定義する簡易言語である。「CAPE」は標準提供されているユニットにより、プログラムの基本構造を定義し、必要があれば細かな点を日本語表記用語で表現した命令によって個別に追加定義できる。ユニ

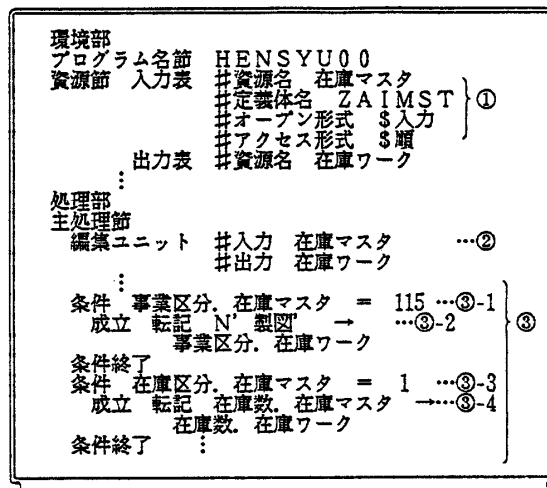


図1 CAPE記述例

ットとは事務処理の多くが標準的な処理の組み合わせであることに着目し、この個々の標準処理をパターン化したものである。「CAPE」は図1のCAPE記述例に示すように、使用するファイル・画面等の資源情報〔①〕や基本構造を示すユニット名〔②〕及び機能説明〔③〕等の表現が標準化されており、従来の曖昧な表現による問題点を解決している。

2.3 テスト項目の自動選定

「CAPE」を入力としたテスト項目の生成方法について述べる。

テスト項目を選定するには処理を制御する入力項目(ファイル・画面の項目)とその条件や値を抽出し、それらを組み合わせることによって起こる処理結果を1つのテストケースとする。図1のCAPE記述例と対応させてみると、

- ・処理を制御する入力項目 : ①の入力資源がもつ項目のうち、③-1、③-3のように条件文に使用されている項目
 - ・条件や値 : ③-1、③-3の条件文の比較記号(=、<等)の条件とその右にある値
 - ・条件の組み合わせ : ③の機能説明の条件文及び分岐制御のパスを解析した情報
 - ・処理結果 : ③-2、③-4のような1つの条件が成立または不成立の時に処理される結果
- 従って、「CAPE」にはテスト項目を選定するた

*1) Customer's Application Program Editor
 *2) Computer Aided Software Engineering Tools

めの情報がすべて存在し、それらを解析することによりテスト項目の生成が可能である。

2.4 ツール化の実現

「CAPE」には他にも以下のような特長があり、これらが今回のツール化実現の大きな鍵となった。

- ①機能単位にまとめられた命令や関数を使用した機能表現による、CAPE解析の機械化実現
- ②パターン化された処理構造及び、ファイル・画面の項目情報（属性・桁数等）の一括管理による、CAPE解析の効率化実現

3. 機能概要

ツールの仕組みがどのような機能で構成されているかについて述べる。

3.1 システム構成

システム構成を図2に示す。

当ツールはワークステーション上から入力した「CAPE」を機能仕様書として、ホスト上でテスト項目及びテストデータを生成するものである。

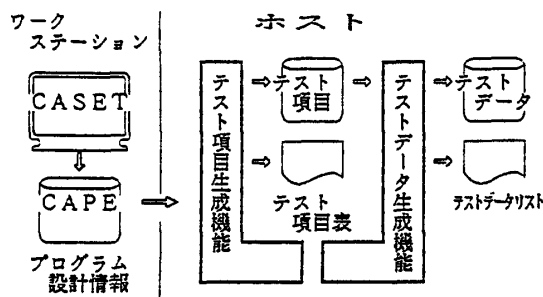


図2 システム構成

当ツールは大きく以下の二つの機能からなる。

- ①テスト項目生成機能
- ②テストデータ生成機能

3.2 テスト項目生成機能

テスト項目生成機能は、「CAPE」を入力として図3の例に示すテスト項目表を出力する。出力される情報はテストの入力項目の条件・値を「原因」、予想される処理結果を「結果」として出力している。「原因」欄の○印は条件の成立、×印は不成立を表し、それらの組み合わせによって起こる処理を「結果」欄の○印で表している。

条件内容		テストケース番号			
		1	2	3	4
原因	事業区分. 在庫マツ = 115	○	○	×	×
	在庫区分. 在庫マツ = 1	○	×	○	×
結果	転記 N'製図' → 事業区分...	○	○		
	転記 在庫数. 在庫マツ → ...	○		○	

図3 テスト項目表の例

3.3 テストデータ生成機能

テストデータ生成機能は、先に述べたテスト項目表の「原因」となる条件の成立・不成立に伴う値を組み合わせたデータを生成している。また、生成されたテストデータの確認をするために図4の例に示すテストデータリストの出力も可能である。図4の①は図3に示すテスト項目表のテストケース番号1に対応するデータ、②はテストケース番号3に対応するデータを各々生成したことがわかる。

テストデータリスト

資源名=在庫マスタ

REC-NO	テストケース番号	項目名	値	
0001	1	事業区分	115	└─①
0001	1	在庫区分	1	
0009	3	事業区分	000	└─②
0009	3	在庫区分	1	

図4 テストデータリストの例

利用者は、テスト対象のプログラム名及びテストを行いたいテストケース番号と出力先の物理ファイル名を入力するだけでデータの生成が可能である。

4. 効果

当ツールを適用することにより、次のような効果が期待される。

4.1 テスト品質の向上

- ①個人の能力に関係なく一定の質・量のテスト項目・データの生成が可能
- ②正確かつ見易いテスト項目表により、机上テスト（静的解析）時のテスト品質向上が可能
- ③作業の標準化が可能

4.2 作業工数・時間の削減

- ①作業の機械化により作業工数・時間の削減が可能
- ②テスト品質向上に伴い、後戻り作業等による工数の削減が可能

5. おわりに

当ツールはプロトタイプ版であり、今回は機能の確認に留めている。今後は機能拡張及び操作性や性能について強化を図っていきたい。併せて、当ツール適用時の作業方法についても適用評価を行いながら、確立していききたい。

【参考文献】

- (1) ソフトウェア・テストの技法
Glenford j.myers著 松尾正信訳
- (2) 日経エレクトロニクス 1982.3.15
「解説 手工業的手法からの脱皮を目指す
ソフトウェア・テスト」