

## 7 J-9 小規模開発におけるプログラムテスト支援ツール\*

石丸 靖啓 廣田 豊彦†

九州工業大学‡

## 1 はじめに

プログラムの品質を向上させるためには、組織的なテストを行なうことが重要であるが、特に一人または数人で全てを行なう小規模なプログラム開発においては、十分なテストが行なわれないことが多い。我々はそのような状況を改善することを目的として、ユニットテストを支援するためのツールを開発した。

本ツールの特徴としては

- モジュール構成の設計と並行してテスト設計が行える。
- テストに必要なスタブ/ドライバなどを管理しており、再テストが容易である。
- 文網羅率が自動的に測定される。
- モジュールを変更した場合に、再実行が必要なテストが自動抽出される。
- モジュール単位ならびにプロジェクト全体のテスト状況を示すレポートが生成される。

などがある。また、本ツールの開発に当たっては、ツール利用のために余分な労力をほとんど必要としないことを目的の一つとした。

## 2 ユニットテスト

テストには、ユニットテスト、インテグレーションテスト、システムテストなどがあるが、本研究では、その中のユニットテスト [1] に焦点を当てている。ユニットテストには以下のような問題点がある。

理想的な開発では、テストの設計はモジュール構成の設計の後に行なわれるべきである。ユニットテストはモジュールのコーディングの直後に設計されたり、全く設計されなかったりすることも多い。

また、ユニットテストには、テストドライバやスタブモジュールが作成され、使用される。これらはユニットテスト毎に一括して管理され、テスト実行時にはあまり意識しないで済むことが望ましい。

一般にモジュールが変更される時には、そのモジュールのみならず、そのモジュールを呼び出すモジュールや、そのモジュールから呼び出されるモジュールなども再テストされなければならないが、それらを人手で

チェックすると、テスト洩れが起こることもまれではない。

我々はこれらの問題点を解決する機能を持つツールを開発した。以下に本ツールについて説明する。

## 3 テスト支援ツール

本ツールには主に次の五つの機能がある。

- モジュール構成図の入力  
モジュール相互の呼び出し関係をモジュール構成図として入力する。本ツールはこの構成図を基礎としてさまざまな支援を行なう。
- テストの設計  
モジュール構成図上で、ユニットテストを設計する。一般にユニットテストはターゲットモジュールと、それに関連するテストドライバやスタブモジュールを指定するが、すでにコーディングが完了して全てのモジュールが存在する時には、ターゲットモジュールの指定のみでよい。
- テストの実行  
構成図上で、テストが設計されたモジュールを指定すると、そのモジュールに対するテストを実行することができる。実行中、文網羅率が記録されるが、これには Unix システムの C コンパイラの -d オプションと tcov コマンド [2] を使用している。
- テストの信頼性のチェック  
テストに用いられたモジュールのバージョン番号やタイムスタンプ、ファイルサイズが最新のテスト実行時と異なるとき、そのテストは信頼性のないテストと判定され、再テストを行なう必要があることを警告する。信頼性のチェックは、指定したモジュールに対するテスト又は全てのテストに行なうことができる。
- レポートの作成  
次の二種類のレポートを自動作成する。

1. モジュールレポート: 指定されたモジュールの次のような情報からなるレポートを作成する。

- モジュール情報

\*Program Testing Support Tool for Small-Scale Development

†Yasuhiro ISHIMARU Toyohiko HIROTA

‡Kyushu Institute of Technology

- バージョン情報
  - モジュールに対するテストの要約
  - 文網羅率
  - 実行されたテストの詳細
2. プロジェクトレポート: 次の情報を含んだプロジェクト全体のレポートを作成する.
- プロジェクト情報
  - モジュール情報の要約
  - 全テストの要約
  - 全テストの詳細
- あるモジュールのレポートを図1に示す.

以上の機能を実現するために、本ツールで用いるデータ構造は以下の通りである.

- モジュール管理データ  
モジュール名, モジュール ID, バージョン履歴などからなる. モジュール構成図はこれのリンクで表される.
- バージョン履歴データ  
バージョン番号, タイムスタンプ, ファイルサイズからなる. テスト実行時にモジュールが更新されていれば, それを最新のバージョンとして登録する.
- テスト管理データ  
テスト名, テスト ID, 設計時刻, 使用されるモジュールなどからなる. テストを実行することにより, 実行回数や, 実行時のモジュールのバージョンも記録される.

本ツールを用いた場合の, テスト設計—テスト実行—テスト終了の手順を図2に示す. 二重枠のボックスが, 本ツールで支援している部分である.

4 おわりに

本ツールを用いて, 本ツール自身のテストを行ない, テスト支援機能を確認した. ただし, すでに前モジュールが完成した後にテストの設計を行っており, 今後新規プロジェクトにおいてモジュール設計時からの本ツールの適用を試してみる必要がある.

参考文献

[1] Glenford J.Meyers: "The Art of Software Testing", John Wiley & Sons, Inc., 1979. (長尾 真 監訳, 松尾 正信 訳 「ソフトウェア・テストの技法」 近代科学社 1980)

[2] W.B.Frakes, C.J.Fox, B.A.Nejmeh: "Software Engineering in the UNIX/C Environment", Prentice-Hall, Inc., 1990. (藤井 直人 監訳, 小川 晃男 訳 「UNIX/C ソフトウェアエンジニアリング」 トップラン 1991)

```

--- Module report ( list.cc ) : Sat Jan 23 05:02:48 1993 ---
1. The module information
Module name : list.cc
ID number   : 11
Date        : Wed Jan 13 02:17:31 1993
Size        : 2755 bytes
2. The version information
Ver. 1      : 2747 bytes, Tue Dec 29 16:48:02 1992
Ver. 2      : 2755 bytes, Wed Jan 13 02:17:31 1993
3. The summary of tests for this module
1) Test-11-1 Wed Jan 13 02:52:24 1993 executed 1 times, ver. 2.
4) Testo2    Thu Jan 14 14:54:11 1993 executed 3 times, ver. 2.
4. tcov's information
Top 10 Blocks
Line   Count
125    134954
54     58689
55     49652
17     7955
18     6989
22     6989
52     6217
58     6217
121    6171
128    6089
59     Basic blocks in this file
36     Basic blocks executed
61.02  Percent of the file executed
    
```

図 1: 作成されたレポート

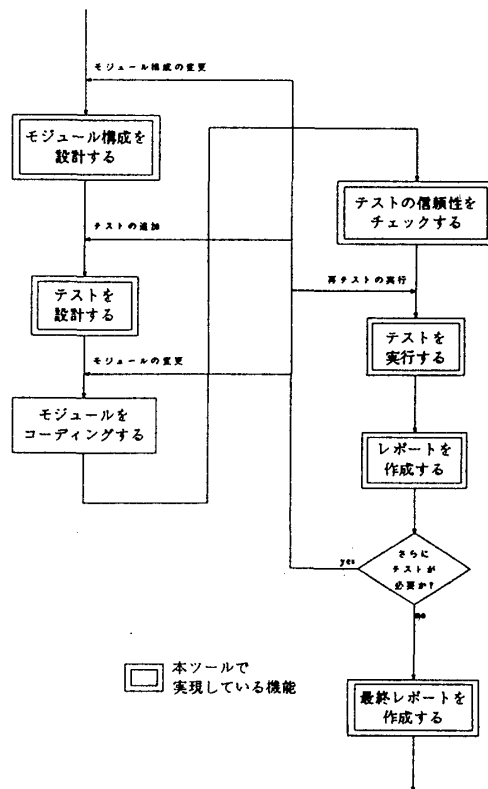


図 2: 本ツールを用いた時の手順