

単体テストのスクリプト記述とカバレッジ評価によるテスト自動化ツール

5 C-6

木村 孝一 近藤 秀文 石原 孝一郎
拓殖大学 工学部

1. はじめに

質の高いソフトを作るためのテストが行われているが、テストは多くの工数を要し、また繰り返し行うことから、テストを自動化することにより、開発時間の大幅な削減が見込める。そこで本稿ではテスト自動化に関する調査として MARK L. MURPHY が紹介しているテスト手法[1]（以後 MARKSQT と呼ぶ）について述べ、それを改良し開発した自動テストツールの報告を行う。このツールは、内部処理（対話処理ではない）プログラムに関する単体テストから統合テストまでをスクリプト記述で、また画面上の操作でテストでき、テストの評価としてカバレッジを計測する。

2. 調査対象としたテストツール：“MARKSQT”

2. 1 “MARKSQT”の機能

“MARKSQT”は、C/C++ で書かれたプログラムを対象としたテストツールである。

2. 1. 1 テスト条件提示機能

この機能は関数のパラメータの型に着目し、型に対してテストを行うべき条件を登録しておく（カタログ）と、その関数のテスト条件を生成する。例えば、int 型に対する条件として、0・正の数・負の数という3つの条件をカタログとして登録しておく、int 型のパラメータを持つ関数に対して、この3通りのテストが必要ということを提示する。

2. 1. 2 スクリプト記述による単体テスト機能

関数、クラス単位で単体テストが行える。スクリプト形式で記述できる言語を提供している。テスト対象のプログラムをスクリプトで実行するためには、中間クラスを作り、テストドライバ（単体テスト機能本体）に組み込んでおかなければならない。テストは、スクリプトをテストドライバが解釈し実行することによって行う。テスト結果はログファイルとしてテキスト形式で出力する。（Figure 4 参照）

スクリプトの記述言語は、オブジェクト指向言語で基本的な計算・論理オペレータを扱え、出力、数値、文字列のためのクラスが組み込まれている。

スクリプトの例を次に示す。

スクリプト：

```
1:tmp val|
2:tmp.Assign SampleTest.New
3:val.Assign [tmp.max 1 2]
4:Log.Print "max(1,2) Result : " val
5:(val=2).IfTrueElse [Log.Print "OK"] [Log.Print "Error"]
```

出力結果（ログファイル）：

```
Max(1,2) Result : 2
OK
```

行番号：説明：

- 1: tmpとvalの2つの変数の宣言している。型は無い。
- 2: 変数tmpにクラスSampleTestを代入している。AssignはC言語の=にあたる。SampleTestが中間クラスにあたり、maxという関数を持っているとする。
- 3: 関数maxを実行し、結果を変数valへ代入する。
- 4: ログとして文字列と変数valの値を出力する。
- 5: 変数valの2なら“OK”、違うなら“ERROR”と出力する。

2. 1. 3 カバレッジ機能

分岐(if)、ループ(while, for, do_while)、ファンクションコールのカバレッジを計測できる。分岐カバレッジはtrueとfalseの両方を実行したか、ループカバレッジは0回ループ、1回ループ、複数回ループの3通りを実行したかを調べ、すべてを実行していなければ出力する。カバレッジを計測するためにマクロを使って、各命令にカバレッジ用のログを出力するよう変更している。カバレッジはこのログから求める。

3. 本研究で開発したツール

3. 1 ツールの機能と特徴

このツールは“MARKSQT”を元に改良を加えた Windows プログラムである。メイン画面 (Figure 1) は、MDI (Multi Document Interface) 形式のテキストエディタに、テストプロジェクトの情報を示すコントロールボックス（プロジェクトボックス）をつけたものである。テストプロジェクトとは、テストに関連するすべてのファイルを含めたものである。ファイルの管理を容易にするためにこのような形態にした。

このツールの機能を以下に示す。

3. 1. 1 プロジェクトボックスによるファイル情報の管理

プロジェクトボックスには 3 つのツリー表示のウィンドウがありタグで切り替えられるようになっている。3つのツリーは、スクリプトファイル、テスト対象のソースファイル、テストケースの情報を示している（Figure 1）。

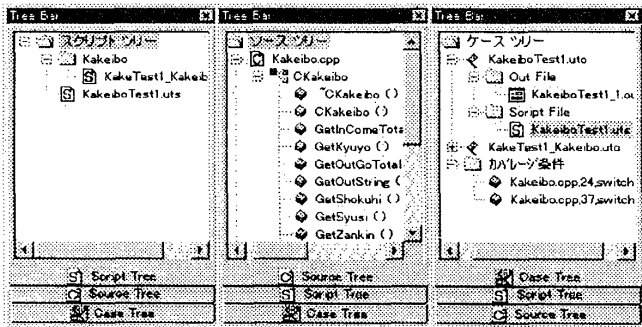


Figure 1 Project Box

3. 1. 2 テストスクリプトの自動作成機能

“MARKSQT”のテスト条件の提示機能を使い、条件を示すだけでなく、実際のテストスクリプトを作成する。操作はソースツリーから作成したい関数を選択し、“スクリプト作成”を実行するだけで良い。

3. 1. 3 単体テスト機能

テストの構造を Figure 2 に示す。“MARKSQT”では、スクリプトファイルが実行の単位だったが、このシステムではテストを実行する単位をテストケースとし、複数のスクリプトファイルをまとめて実行するようにした。このテストケースもマウス操作で作成できる。これにより簡単に複数のスクリプトファイルを実行できるようになった。またそれに伴いスクリプトファイルのテスト結果を加え、テスト結果の出力として、ログファイルだけでなくスクリプトの結果の一覧と、カバレッジ結果をまとめてアウトファイルとして出力する。

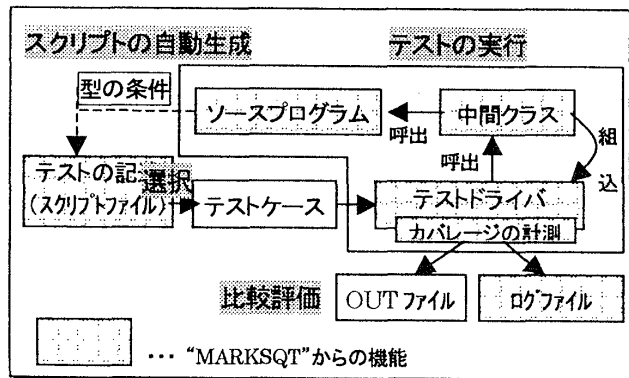


Figure 2 Structure of Test Process

3. 1. 4 カバレッジ機能

“MARKSQT”のカバレッジ機能を用いているので、記述の制限は変わらないが、次の点を改良した。

- switch のカバレッジの計測
- break 等でループを抜けても正しく計測する
- 実行されていない命令も出力する
- 実行されないカバレッジの種類を登録することにより、出力しなくする

3. 1. 5 テスト結果のグラフィカル表示

テスト結果として、テストの開始・終了時間、失敗数、ログファイル、カバレッジの度合いと、ソースファイルごとのカバレッジのリスト、スクリプトの結果のリスト、カバレッジ結果のリストを示す。また、2つのテスト結果を比べ、変化した箇所を示す機能と、複数のテスト結果から失敗数の推移とカバレッジの結果を示す機能追加した。

3. 2 ツールの評価

このツールは、“MARKSQT”と比べ次のように改善された。テスト情報のグラフィカル表示とマウス操作によるテスト実行が可能になり、テストの効率を向上できる。

- 1) ファイルの管理が容易。
- 2) テストスクリプトの自動作成。
- 3) 複数のファイルを実行するのが容易。
- 4) マウス操作による単体テストの実行
- 5) テストドライバの強化、機能の追加
- 6) カバレッジの機能の強化
- 7) テスト結果がグラフィカルに表示
- 8) テスト結果の比較、経過のグラフィカル表示機能

4. まとめ

本稿では、単体テスト、カバレッジ評価に関するテストツールにいて述べた。今後の予定として、もっと多くのプログラムに対してこのツールを用いたテストを実施し、有用性を検証する。

参考文献

- [1] Mark L. Murphy : C/C++ SOFTWARE QUALITY TOOLS , PRENTICE HALL (1996).
- [2] ボーリス・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).