

B-32

テストカバレッジ取得方式の検討と適用 On Measurement of the Test Coverage of Software Development

三好 辰弥† 桐井 隆志† 森山 宣郎‡
Tatsuya Miyoshi Takashi Kirii Noburo Moriyama

降旗 由香里† 大里 立夫† 曾根原 勝† 原田 晃†
Yukari Furuhashi Tatsuo Oosato Masaru Sonehara Akira Harada

1. はじめに

1.1 背景

信頼性の高い情報システム構築のために、テストカバレッジ値 C0/C1 の 100% 達成が必要条件である[1]。

1.2 差分カバレッジの考え方

表1 差分カバレッジの考え方

項番	種類	テストカバレッジ [%]
1	C0 カバレッジ	実行済みステップ数 / 総ステップ数
2	C1 カバレッジ	実行済み分岐数 / 総分岐数
3	RC0 カバレッジ	実行済み修正ステップ数 / 総修正ステップ数
4	RC1 カバレッジ	実行済み修正分岐数 / 総修正分岐数

表1に示す C0/C1 は広く知られている[1]。ここでテストカバレッジ RC0/RC1 を、C0/C1 との差分カバレッジとして表1のように定義した。この差分カバレッジの考え方は、開発段階で小さなプログラム修正を行うと修正前に蓄積した C0/C1 カバレッジが信頼性の観点で意味を失い、また、C0/C1 を 100% とするようなテストを修正の都度繰り返し行うことは工数的に効率が悪いとの問題意識に基づく[2]。RC0/RC1 を部分的に使用すれば、信頼性を保ちつつ開発工数を絞ることができると考えられる。差分カバレッジとして2種類を定義したが、RC0 を使用するだけでも開発効率をある程度向上させることができると考えている。

1.3 従来のテストカバレッジツールの問題点

テストカバレッジの取得を支援する製品は、C0 カバレッジについてこれまでにいくつか知られている。しかし、我々の限られた調査の範囲では、C1 および RC0 カバレッジの取得をサポートしているものは日立製 COBOL コンパイラ以外には見当たらなかった。また ABAP/4 言語に関して、テストカバレッジの取得を支援する製品は、現時点で見当たらなかった。

このため Java, C/C++, Visual Basic について、C0 カバレッジに加えて C1/RC0 カバレッジまで取得できる方法を検討し適用を試みた。ABAP/4 についても、C0/C1/RC0 カバレッジの取得方法について検討し適用を試みた。

2. Java, C/C++, Visual Basic, ABAP/4 の C1/RC0 カバレッジ取得方式の検討と適用

2.1 C1/RC0 カバレッジ取得方式の検討

例えば日立製 COBOL コンパイラには、C1/RC0 カバレッジ取得機能を内在させているものがある[3]。しかし、他社製コンパイラ等には改造を加えることはできない。よって、該当言語におけるカバレッジ取得のために、コンパイラもしくはインタプリタが解釈できる手段を対象プログラムにソースコードとして(自動的に)埋め込む方式を採用することにした(本報告では、ソースコード埋め込み方式と呼ぶ)。

2.2 ソースコード埋め込み方式の検討と適用

ソースコード埋め込み方式として、表2に示す3つの機能を検討し、適用した。

表2 ソースコード埋め込み方式の機能構成

項番	機能名	機能概要
1	ソース解析・埋め込み	カバレッジ取得のため対象プログラムを解析し、テスト実行時に通過した分岐点では、通過したことをファイルに出力する処理を埋め込む。
2	カバレッジ情報取得	解析・埋め込み済みソースをテスト実行することで、カバレッジ情報を取得する。
3	カバレッジ情報表示	対象プログラムを構成しているモジュールに対して、C0/C1/RC0 のカバレッジ情報を表示する。

また、上記機能の動作概要を図1に示す

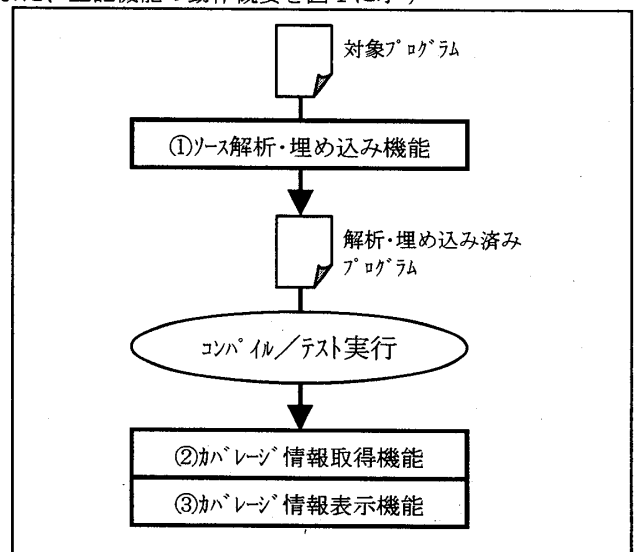


図1 ソースコード埋め込み方式の処理フロー

† (株) 日立製作所, Hitachi Ltd.

‡ 日立東北ソフトウェア(株), Hitachi Tohoku Software, Ltd.

表3 各開発言語に対する適用結果

2.3 Java, C/C++, Visual Basic, ABAP/4 への埋め込み
 図2に、Java へのカバレッジ取得の為の埋め込み例を示す。

```

Int ax = 0;
public void func01() {
    カバレッジ情報取得用の処理
    if (ax == 0) {
        カバレッジ情報取得用の処理
        System.out.println("True 通過");
        カバレッジ情報取得用の処理
    } else {
        カバレッジ情報取得用の処理
        System.out.println("False 通過");
        カバレッジ情報取得用の処理
    }
    カバレッジ情報取得用の処理
    return;
}
    
```

図2 カバレッジ取得のための埋め込み例

上記以外にも C/C++, Visual Basic, ABAP/4 にも、カバレッジ取得用の処理を (自動的に) 埋め込むことに成功した。

2.4 テストカバレッジ情報の取得

C0/C1/RC0 カバレッジ値の定義は、表1で述べたとおりである。では実際に、分母の総ステップ数・分子の実行済みステップ数等をどのようにして取得するのが焦点となる。

C0/C1 カバレッジの取得方法であるが、カバレッジ演算式の分子の計測は、図2に示すカバレッジ情報取得用の処理を実行したステップ数を元に計測できる。一方、分母の計測であるが、対象プログラムの解析を行う際、ソースコードの字句ごとに解析をし、一つの処理ごとに種別フラグを割り当てている。

次に RC0 カバレッジの計測方法であるが、対象プログラムの解析を行う際、ソースコードの字句ごとに解析し、修正前との比較を行い、処理ごとに修正フラグを割り当てている。

3. 結論

Java, C/C++, Visual Basic, ABAP/4 の各言語について C1/RC0 カバレッジの取得方法を検討し、ABAP/4 については C0 カバレッジについても検討適用した。

3.1 Java, C/C++, Visual Basic, ABAP/4 への適用結果

Java, C/C++, Visual Basic, ABAP/4 の各言語について、ソースコード埋め込み方式による C0/C1/RC0 カバレッジ取得方法を適用した結果を、表3に示す。

項番	対象言語	C0 取得		C1 取得		RC0 取得	
		日立	他社製品	日立	他社製品	日立	他社製品
1	Visual Basic	○	○	○	×	○	×
2	C/C++	○	○	○	×	○	×
3	ABAP/4	○	×	○	×	○	×
4	Java	○	○	○	×	○	×

○…カバレッジ取得実現
 ×…カバレッジ取得非実現

3.2 今後の課題

我々は、開発者の品質向上、テスト高速化の要望に応えるべく、今回のソースコード埋め込み方式によるカバレッジ取得を実現した。

今後は、信頼性の高い情報システムをより広範囲に構築するために、対象言語の拡大等が課題である。

4. 参考文献

- [1]保田勝通：ソフトウェア品質保証の考え方と実際、日科技連,1995
- [2]降旗由香里,大野治,森岡洋平((株)日立製作所)：差分テストバッグ支援方法、特開平 6-103113,1994
- [3](株)日立製作所：COBOL85 テストデバッグ、解説・手引書 6190-3-726-20p137,1999