

ソースコード埋め込み型カバレッジツールについて

桐井 隆志[†] 三好 辰弥[†] 岸上 諭[†] 大里 立夫[†] 曾根原 勝[†]

(株)日立製作所 情報・通信グループ 生産技術本部[†]

1. 序論

1.1 背景

信頼性の高いシステム開発のためには、C1 カバレッジの100%達成が必要条件である[1]。特に日立では、社会インフラを担うミッションクリティカルな業務システム開発を多数手がけている。そのため、十分なテストが行われているかを判断するための指標が必要である。

このように、精度の高いテストカバレッジ取得は非常に重要な課題であり、高精度のテストカバレッジを効率的に取得できるツールが必要とされていた。

1.2 テストカバレッジの考え方

表 1 テストカバレッジの定義

項番	種類	加 ^レ ジ ^ツ [%]
1	C0 加 ^レ ジ ^ツ	実行済みステップ数 / 総ステップ数
2	C1 加 ^レ ジ ^ツ	実行済み分岐ステップ数 / 総分岐ステップ数
3	RC0 加 ^レ ジ ^ツ	実行済み修正ステップ数 / 総修正ステップ数

表 1 に示す C0, C1 は、広く知られている[1]。また、我々はテスト工数削減の観点から、修正部分に対する差分カバレッジとして RC0 を表 1 のように定義している。

この差分カバレッジの考え方は、以下 2 点の問題意識に基づくものである。僅かでもプログラム修正を行うと、修正前に蓄積したテストカバレッジが信頼性の観点で意味を失う。C0, C1 を 100% とするようなテストを修正の度に繰り返し行うことは、作業負荷が高い。そこで RC0 カバレッジを部分的に適用すれば、信頼性を保ちつつテスト工数を削減できると考える[2]。

1.3 市販テストカバレッジツールの問題点

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はこれまで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支援するものは日立製 COBOL コンパイラ以外には見当たらなかった。また ABAP/4 に関してテストカバレッジの取得を支援する製品は、同様に見当たらなかった[3]。

1.4 対策

市販テストカバレッジツールについて調査した結果、1.3 に示す問題点があった。そのため、高精度かつ高品質の要求を満たさないと判断し、独自のツールを開発することとした。

そこで、我々にとって主要な開発言語である Java, C/C++, Visual Basic (以下 VB) について、C0 に加えて C1, RC0 カバレッジまで取得できる方式を検討した。また、ABAP/4 でも C0, C1, RC0 カバレッジ取得方式を検討した。

2. テストカバレッジ取得方式の検討と実装

我々が開発したソースコード埋め込み型カバレッジツールについて、テストカバレッジ取得方式の概要と特に C1 カバレッジ取得方式について報告する。

2.1 テストカバレッジ取得方式の概要

(1) テストカバレッジ取得方式の検討

例えば日立製 COBOL コンパイラなど、C1, RC0 カバレッジ取得機能を内在させているものがある[4]。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。

そこで我々は、コンパイラまた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とした。本報告ではこれ以降、この方式をソースコード埋め込み型と呼ぶ

(2) ソースコード埋め込み型の機能概要

ソースコード埋め込み型として、表 2 に示す 3 つの機能を検討し、ツールに実装した。

表 2 ソースコード埋め込み型の機能構成

項番	機能名	機能概要
1	ソース解析・埋め込み	加 ^レ ジ ^ツ 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	加 ^レ ジ ^ツ 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、加 ^レ ジ ^ツ 情報を取得する。
3	加 ^レ ジ ^ツ 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 の加 ^レ ジ ^ツ 情報を表示する。

また、上記機能の動作概要を次の図 1 に示す[3]。

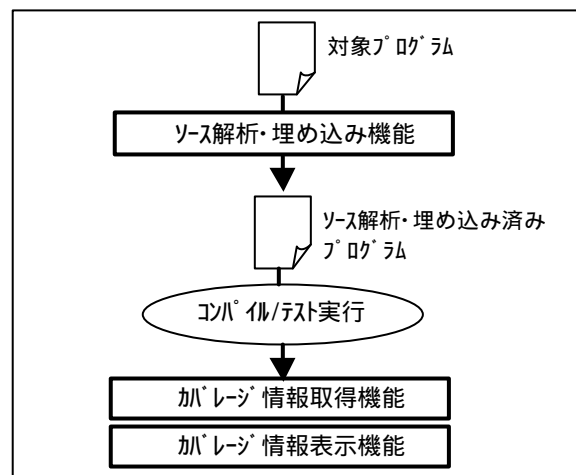


図 1 ソースコード埋め込み型の機能動作

On Measurement of the Test Coverage of Software Development

[†]Takashi Kirii, Tatsuya Miyoshi, Satoshi Kishigami, Tatsuo Oosato, Masaru Sonehara

[†]Hitachi, Ltd., Information & Telecommunication Systems

(3) ソースコード埋め込みの例

図 2 に、Java でのソースコード埋め込みの例を示す。

```
class HelloJava{
    public static void main(String args[]){
        テストカバレッジ取得用の処理
        System.out.println("Hello Java");
        テストカバレッジ取得用の処理
    }
}
```

図 2 ソースコード埋め込みの例 (Java)

上記以外にも C/C++, VB, ABAP/4 について、テスト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[3]。

2.2 C1 カバレッジ取得方式の検討と実装

(1) C1 カバレッジ取得上の問題点

Java, C/C++, VB, ABAP/4 では、例えば if-else の条件分岐において、else 節は省略可能である。このためソースコード埋め込み型では、テストカバレッジ取得用の処理を埋め込むために、キーワードとして else を利用できないといった問題があった(図 3)。

```
:
if (条件式) {
    テストカバレッジ取得用の処理
    条件式が true の処理
    (但し条件分岐を含まない)
    テストカバレッジ取得用の処理
}
:
```

図 3 2 分岐において else 節が省略されている場合は else 節に埋め込みできない例

```
:
if (条件式) {
    テストカバレッジ取得用の処理
    条件式が true の処理 . . . . .
    (但し条件分岐を含まない)
    テストカバレッジ取得用の処理
} else {
    テストカバレッジ取得用の処理
    条件式が false の処理 . . . . .
    (但し条件分岐を含まない)
    テストカバレッジ取得用の処理
}
:
```

図 4 2 分岐において else 節が省略されていない場合の埋め込み例

図 3 では else 節が省略されており、テストカバレッジ取得用の処理を埋め込むためのキーワードとして else を検出できない。このため、実際には 2 分岐において

true/false 両方の処理を通過していたとしても、C1 カバレッジは 50%にしかない。

図 4 では if-else で else 節が省略されておらず、条件式が true の処理 () と条件式が false の処理 () に対応するテストカバレッジ取得用の処理を正しく埋め込むことができ、true () と false () 両方の処理を通過すれば C1 カバレッジは 100%となる。

(2) 対策

ソースコード埋め込み型では、例えば if-else で else 節が省略されている場合(図 3)には、ソース解析・埋め込み機能において else 節の省略を検出し、自動的に処理のない空の else 節を挿入することとした。

(3) 評価

ソースコード埋め込み型カバレッジツール開発では、条件分岐で文法上は省略可能な分岐節について、入れ子構造を含む全ての場合の検出を満たすテストケースを作成しテストを実施した。

これにより、例えば if-else の else 節が省略されている場合も検出して空の else 節を自動挿入し、テストカバレッジ取得用の処理が正しく埋め込まれていることを確認した。その結果、全ての分岐ステップに対して実行済み分岐ステップ数を全て確認でき、C1 カバレッジ 100%の取得が可能となった。

3. 結論

Java, C/C++, VB, ABAP/4 の各言語について、C0 のみならず C1, RC0 カバレッジが取得可能なソースコード埋め込み型カバレッジツールを開発した。また、C1 カバレッジを正しく自動的に取得する方式を検討し、ツールに実装した。

ソースコード埋め込み型カバレッジツール適用で、テストカバレッジ取得の自動化と RC0 カバレッジによるテスト工数の削減ができる。これにより、追加工数を極力少なくした上で、業務システム開発における高い品質を確保することが可能となった。

4. 今後の課題

今後はより広範囲に信頼性の高い業務システムを構築するため、新技術への対応を行う。対象とする開発言語の拡大を図り、引き続きカバレッジツールの適用を推進していくことが課題と考える。

5. 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
差分テストデバッグ支援方法,特開平 6-103113,1994
- [3]三好辰弥,桐井隆志,森山宣郎,降旗由香里,大里立夫,曾根原勝,原田晃
テストカバレッジ取得方式の検討と適用
情報科学技術フォーラム講演論文集
pp. 151-152, Sep. 2002
- [4](株)日立製作所
COBOL85 テストデバッグ解説・手引書
6190-3-726-20p137,1999