

6E-2

品質保証ツールQTにおけるテスト
カバレッジ機能について

中田雄三, 伊東秀一, 沖洋一, 野木秀子

(日本コンピュータ研究所 システム開発部)

1 はじめに

最近、ソフトウェアの品質に対する関心が高まり、高級言語主体のテストカバレッジ機能を中心とした品質保証ツールが実用化されている。しかしながら実行速度やプログラムサイズの点から、又ハードウェアとの関連からアセンブラ言語を使用しているユーザは少なくない。ところがアセンブラ言語用の品質保証ツールはほとんど実用化されていない。QT (Quality assurance Tool) は、HP 64000 ロジック開発システムのもとで作動するアセンブラ言語用品質保証ツールである。本論文では、QTの機能と実現方式を中心にユーザの評価をまじえ、その有用性について述べる。

2 QTの機能

(1) テスト充分性の評価: QTは、HP 64000 ロジック開発システムの強力なテスト/デバッグ機能を使いながら、テストカバレッジ (テスト網羅率) を採取する。テストカバレッジは動的テストのテスト充分性を評価するもので、次の3つの尺度で測定する。

C0メジャ: モジュール内のテスト済み実行文の比率

C1メジャ: モジュール内のテスト済み分岐方向の比率

S1メジャ: すべての外部モジュールに対する呼び出しの実行比率

(2) テストケース作成の支援: モジュールの未テスト部分を抽出し、構造面からのテストケース作成を支援するため、次の項目を出力する。

(a) 実行文/分岐方向の実行回数の測定

(b) 未実行文/未実行分岐の抽出

(c) モジュールの実行回数の測定

(d) 未実行のモジュールの抽出

3 QTの実現方式

(1) 適用機種 HP 64000 ロジック開発システム

(2) 対象CPU モトローラ 6809、68000

インテル 8086、8088

(3) QTの構成

QTは次の4つのプログラムから成る。

(a) 構造解析プログラム

テストカバレッジを採取したいユーザプログラムを解析して必要な情報を構造ファイルに作成する。

(b) 監視プログラム

構造解析の結果を基に、ユーザプログラムの監視点 (カバレッジ情報を採取する位置へ、割り込み命令を挿入し、その後ユーザプログラムへ制御を渡し、その動きを監視しながら、テストカバレッジ情報を収集する。

(c) 監視終了プログラム

ユーザプログラム終了時、テストカバレッジ情報を構造ファイルへ蓄積する。

(d) カバレッジ情報編集出力プログラム

構造ファイルに蓄えられたテストカバレッジ情報を編集出力する。

The function of quality assurance tool
Yuzo NAKATA, Syuichi ITO, Yoichi OKI, Hideko
NOGI

CIJ Inc.

4 効果

- (1) テストカバレッジ (C0、C1、S1メジャ) により、テストの進捗度が客観的に把握できる。
- (2) 未実行文、未実行分岐方向がわかるため、テストケースを追加することによりテスト漏れがなくなる。
- (3) 実行回数の測定により、性能向上の指標になる。
Cメジャの出力例を図2に示す。

5 ユーザの評価

- (1) テスト時間の削減。
未実行部分をねらい打ちしたテストケースを追加していくことにより、無駄なテストがなくなりテスト時間の大幅な削減ができた。
(図1 不良の抽出推移 参照)
- (2) コーディング不良の抽出に効果大。
コーディング時に作り込んだ不良の抽出に効果がある。

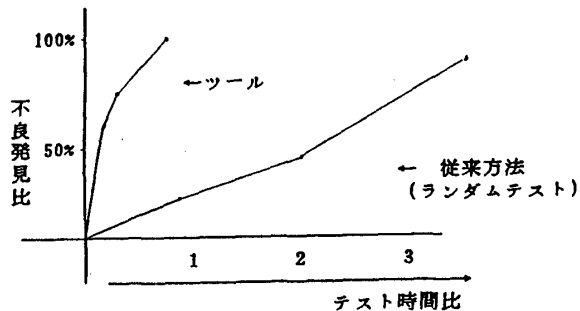


図1 不良の抽出推移

6 おわりに

今後は、表示のグラフ化、視覚化を図り、ユーザインタフェースやオペレーションについてさらに検討を加え、ソフトウェアの生産性向上、信頼性向上のためのよりよいツールとして発展させたい。

```

** (QT-6809 01-00) *****
*                               C Measure Report                               *
***** (( CIJ )) ***** } サイトルです。
                               テストカバレッジ検取モード (Audit と Trao
                               Delete の 2 種類がある) を表示します。
*** MODULE NAME : ( SHO_TEST0 ) ***   (( Audit Mode ))
                               モジュール (アセンブル単位) 名です。
                               Assembled Date & Time
                               アセンブル日時が表示されます。
LINE      COUNT      TRUE      FALSE      SOURCE LINE
CURRENT (TOTAL)  CURRENT (TOTAL)  CURRENT (TOTAL)
17
19      1 ( 1)
20      1 ( 1)
! 21      1 ( 1)      1 ( 1)
.
159      255 ( 255)
! 160      255 ( 255)      254 ( 254)      1 ( 1)
161      1 ( 1)
! 162      1 ( 1)      1 ( 1)
STACK START      EQU      $
LDS      #STACK
LDB      #0
JSR      MESSAGE_OUT
.
CMP X      #FREE_TOP
BHI      INIT_FREE1
STX      FREE
RTS
} 抽出コード。
} 抽出グラフです。
|-----|
| COVERAGE INFORMATION | CURRENT | TOTAL | |
|---|---|---|---|
| (1) | EXECUTABLE INSTRUCTIONS | 119 | 119 |
| (2) | EXECUTED INSTRUCTIONS | 92 | 104 |
| (3) | TOTAL BRANCH INSTRUCTIONS | 66 | 66 |
| (4) | TOTAL BRANCH DIRECTIONS | 85 | 85 |
| (5) | EXECUTED BRANCH DIRECTIONS | 58 | 65 |
|-----|-----|-----|
| (( C0 MEASURE )) = (2)/(1) | 77 % | 87 % |
| (( C1 MEASURE )) = (5)/(4) | 68 % | 76 % |
|-----|-----|-----|
* Measure_Graph * 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|-----|-----|-----|
| C0 | CURRENT | ***** |
| C0 | TOTAL | ***** |
|-----|-----|-----|
| C1 | CURRENT | ***** |
| C1 | TOTAL | ***** |
|-----|-----|-----|

```

図2 QTの出力例

7 参考文献

- 「1」 Miller, E. F: Program Testing: Art Meets Theory, IEEE Computer Vol. 10 No. 7, PP. 42-51, 1977
- 「2」 野木, 中田 テストカバレッジ指標を用いた系統的ソフトウェアテスト法の効果, 日科技連第2回品質管理シンポジウム