プログラム自動生成率と最適な単体テスト方式

7M－1

前田英行　横尾寛
日立公共システムエンジニアリング（株）

１．はじめに

我々は、ソフトウエア開発の生産性向上と品質向上を目的とし、プログラムの自動生成を実施してきた。

プログラムの自動生成率の向上にともない、単体のテスト方式をプログラム全体のテストから自動生成された部分を除いたテストさらに単体テストを省略する方式へ移行する目的で実験を行なった。

本論文では、所定の品質を確保した上で、テスト作業量を最少化するテスト方式とその適用条件を分析した結果を報告する。

２．プログラム自動生成の概要と生成率

図１に、プログラム自動生成の概要を示す。標準パターン、標準部品及び設計情報を元に、半完成のプログラムを合成する。これに必要に応じて追加、修正、削除(ユーザコーディングと呼ぶ)を加え、プログラムを完成させる。

ここで自動生成率を以下のように定義する。

自動生成率＝自動生成行数/完成ﾌﾟﾛｸﾞﾗﾑ行数

（自動生成行数:自動生成され完成プログラムに無修正で残った行数

完成プログラム行数＝自動生成行数＋ユーザコディング行数）

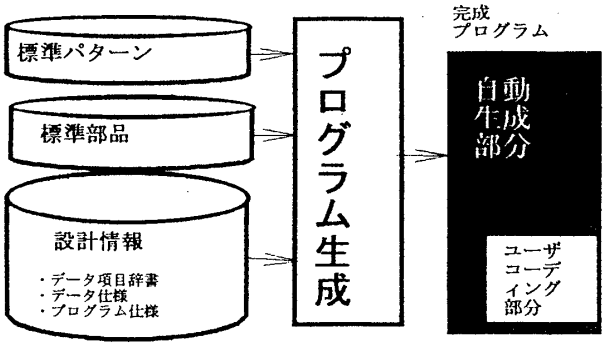

図１．プログラム自動生成の概要

図２に、あるターゲットシステムにおけるプログラム自動生成率の分布を示す。自動生成率が高いプログラム群がかなりの割合を占めていることがわかる。

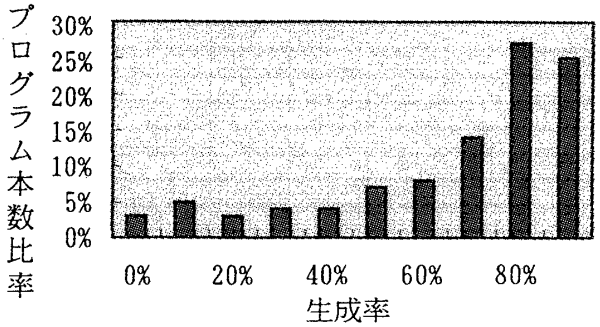

図２．プログラム自動生成率の分布

３．ソフトウエア開発工程と単体テスト

ソフトウエア開発工程を表１に示す。単体テストは内部部品であるプログラムを単体でテストするものである。これは、開発工程全体の中で、次工程である組合わせテストを実施するための品質を確保する工程と位置付けられる。

単体テスト終了時点の残留バグを目標値以下におさえ、かつ作業量の最少なテスト方式が、最適な単体テスト方式となる。

表１．ソフトウエア開発の工程

開発工程	作業内容
システム設計	システムの外部仕様を設計
プログラム設計	プログラムの内部仕様を設計
コーディング	ﾌﾟﾛｸﾞﾗﾑ自動生成、ﾕｰｻﾞｺｰﾃﾞｨﾝｸﾞ
単体テスト	プログラム単体の動作テスト
組合わせテスト	ﾌﾟﾛｸﾞﾗﾑを組合わせた機能（ｼﾞｮﾌﾞ）単位の動作テスト
システムテスト	システム全体の動作テスト

Program generation rate and a best way of program unit test
Hideyuki Maeda, Hirosi Yokoo
Hitachi Government and Public corporation system engineering, ltd.

4．残留バグのモデル

単体テスト終了時の残留バグ数は以下のようにモデル化できる。

・残留バグ数＝コーディングでの作込みバグ数
－単体テストでの除去バグ数
＝(1-除去率)*作込みバグ数

である。ここで

・作込みバグ数＝自動生成部分のバグ数
＋ﾕｰｻﾞｺｰﾃﾞｨﾝｸﾞ部分のバグ数

とすると

・残留バグ密度＝残留バグ/完成ﾌﾟﾛｸﾞﾗﾑ行数
＝(1-j1)*t1*生成率
＋(1-j2)*t2*(1-生成率)...式①
(t1,t2はバグ作込み率でt1,t2 ≧0 、
j1,j2はバグ除去率で 1≧ j1,j2 ≧ 0)

となる。

5．単体テスト方式

従来の単体テスト方式は、自動生成率に無関係に、完成プログラム全体についてマシンテストを実施していた。

自動生成の品質及び生成率の向上によって、自動生成部分のバグ作込み率(t1)を充分低くできれば、式①においてt1≪t2 かつ t1≒0となる。これは式①の第１項の値の影響が少ない事を意味する。さらに式①の第２項において、生成率が向上し100% に近づくほど残留バグが減少することを意味する。

この事から、新しい単体テスト方式(案)として以下のテスト方式を採用し実験を行なった。

表2．新単体ﾃｽﾄ方式(自動生成率,品質が高い場合)

方式	内容	残留バグ予測	作業量
部分省略	自動生成部分のﾏｼﾝﾃｽﾄ省略	(1-j2)*t2*(1-生成率)	1-生成率
全体省略	ﾌﾟﾛｸﾞﾗﾑ全体のﾏｼﾝﾃｽﾄ省略	t2*(1-生成率)	0

注．作業量は従来のプログラム全体のマシンテストの作業量を１した場合の相対値である。

6．新単体テスト方式適用時の残留バグ密度

表２の新単体テスト方式を適用した結果を、図３に示す。これより以下の事が確認出来た。

①部分省略、全体省略方式ともに生成率の向上に伴い残留バグ密度が減少する。

②部分省略方式では生成率80%以上のプログラムの残留バグ密度が目標値以下。

③全体省略方式では生成率85%以上のプログラムの残留バグ密度が目標値以下。

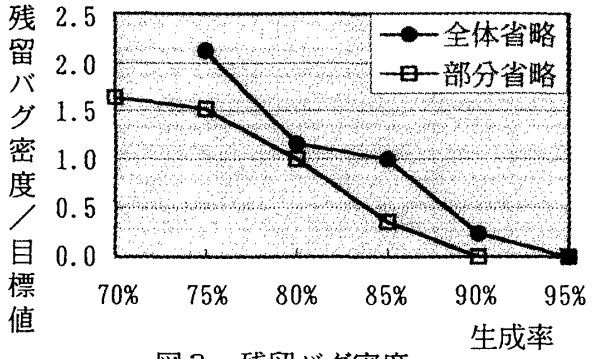

図３．残留バグ密度

7．おわりに

プログラム自動生成率の向上と自動生成部分の品質の向上に伴った、コスト効果の高いプログラム単体テスト方式を採用し適用実験を行なった。

今回の実験では、自動生成率が約80%以上のプログラムについては、部分省略方式の有効性が確認出来た。その場合の単体マシンテストの作業量は表２より従来方式の20%以下となる。さらに、自動生成率が約85%以上のプログラムについては単体のマシンテスト全体を省略しても、一定の品質が確保出来ることが確認できた。

この事から本方式が、ソフトウエア開発の生産性と品質の向上に寄与することが明確となった。

参考文献

(1)降旗,他:「EAGLE/P ディクショナリを用いたプログラムジェネレータの開発と適用効果」情報処理学会第43回全国大会 1K-1 (1991)

(2)森岡,他:「データ中心アプローチを応用した標準データ項目辞書の開発とその活用方法」日立評論 VOL.75 NO.11(1993-11)