

モジュール結合度に基づくプログラム疲労度の測定法の一考察 ～疲労予測式の定式化～

A Consideration of the Method of Program Fatigue Measurement based on the Module Coupling

○坂本裕司* 中島直哉* 武内 惇** 藤本 洋**

Hiroshi Sakamoto Naoya Nakajima Atsushi Takeuchi Hiroshi Fujimoto

* 日本大学大学院工学研究科 〒963-8642 福島県郡山市田村町徳定字中河原 1

** 日本大学工学部 〒963-8642 福島県郡山市田村町徳定字中河原 1

1. はじめに

プログラムに仕様変更要求があった場合、現在のプログラムを変更するか、新規にプログラムを作成するかプログラムの変更可否判断（以下、変更可否判断）はプログラム開発者の経験によるため、プログラムの変更作業（以下、変更作業）工数が予定を超えるという問題が起こることがある。プログラムの変更作業を行う前に、現在のプログラムの変更の難しさの状態を算定して、その結果から現在のプログラムを変更するか、新規にプログラムを作成するかを判断する手法が求められている。

これまでに、プログラムの変更の難しさの状態を「プログラム疲労度」という量で表し、①処理手続きの変更によるモジュール(C 言語の関数プログラムに相当する)内の処理手続き表現の複雑化がプログラム疲労度を増大させる仕組みと、②プログラム疲労度を測定する方法を提案した^[1]。

本稿では、C 言語で記述されたプログラムを対象に、モジュール間の結合の方法がプログラム疲労度を増大させる仕組みと、プログラム疲労度を測定する方法を考察する。

2. プログラム疲労度の定義と測定法の考え方

2.1 プログラム疲労度の定義

プログラムの変更可否判断は変更作業に要する工数に基づいて行われるため、プログラムの変更の難しさの状態(プログラム疲労度)を変更作業に要する工数に基づいて表現する。すなわち、プログラム疲労度は、以下のように変更行数 1 行当たりの変更作業工数を用いて定義する^[2]。

プログラム疲労度 = 変更作業工数 / 変更作業行数

変更作業工数：変更作業を行った工数

変更作業行数：変更作業を行ったプログラムの行数

2.2 プログラム疲労度の測定法の考え方

変更作業を開始する前に、変更可否判定を行えるようにするため、変更作業によるプログラム疲労度が増大する量を予測する必要がある。このため、これまで実施した変更作業の結果に基づいて、変更作業によるプログラムの処理手続きの表現法や、モジュール間の結合法の変更内容を表すパラメータで構成する疲労予測式を定式化する。本稿では疲労予測式を用いて変更作業を行う前にプログラム疲労度を算出することをプログラム疲労度の測定と呼ぶ。

2.3 疲労予測式の定式化

モジュール内の処理の変更要求は処理手続きの場合分けを増やすためテスト作業の工数を増大させることに注目し、ネスト数と条件式の数をパラメータとするプログラム疲労度を算定する疲労予測式を作成した。

本稿では、あるモジュール内の処理の変更が他のモジュールの処理に影響を与える仕組みを分析し、以下の手順に

従って疲労予測式を定式化する。

手順1：モジュールの結合度の分類に基づくモジュールの変更が他のモジュールに与える影響の大きさの検討

手順2：外部変数の変更に伴う変更作業工数の増大要因の分析

手順3：外部変数値の変更回数と参照回数に注目した疲労予測式の定式化

3. モジュール結合度に基づく疲労予測式の定式化

3.1 モジュールの結合度の分類に基づくモジュールの変更が他のモジュールに与える影響の大きさの検討

モジュール結合の仕方によって、呼び出し元のモジュールの処理手続きを変更したとき、①その影響が呼び出し先モジュールにどのように伝わるか、②その影響を検討する範囲はどこかを分析し、検討する範囲の大きさに基づいて変更作業の大きさを検討する。同様に、呼び出し先モジュールの処理手続きを変更したとき、呼び出されたモジュールに与える影響についても同様に検討する(表1)。結合の仕方はマイヤーズの結合度の分類^[3]に基づく。

検討の結果から処理手続きの変更に伴うプログラム疲労度は外部変数によって連結される外部結合や共通結合のモジュール間で増大しやすいと考えられる。

3.2 外部変数の変更に伴う変更作業工数の増大要因の分析

表2に、①外部変数の値の変更と参照場所の変更の2つの変更形式に分けて、それぞれの変更を行うことにより発生する可能性のある問題と、②問題の発生しないことの確認作業ならびに問題の発生する場合に行う対策の作業からなる問題対策作業、③問題対策作業の工数を増大させる要因を分析した結果を示す。分析の結果から、外部変数の値の変更と参照場所の変更が、外部変数の変更を伴う変更作業工数の増大要因であると考えられる。

3.3 外部変数値の変更回数と参照回数に注目した疲労予測式の定式化

変更可否判断を行うときには、詳細な処理手続きの検討に多くの工数をかけることができないため、変更作業を増大させる要因を疲労予測式のパラメータとしてカウントし、それらを用いて変更作業工数を算定できるようにする必要がある。

これまでに、モジュール内の変更作業を増大させる要因を示すパラメータとしてネスト数、条件式の数を設定した。モジュール間の変更作業を増大させる要因を示すパラメータとして外部変数を参照した回数と外部変数を変更した回数を設定する。疲労予測式は条件式の数、ネスト数、外部変数の参照回数、外部変数の変更回数を説明変数とする回

帰式で表現する (図1)。

$$P' = b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + e$$

P' : 目的変数 (プログラム疲労度の予測値)
 b_i : 偏回帰変数 x_i : 説明変数 ($i: 1 \sim 4$) e : 残差
 x_1 : 条件式の数 x_2 : ネスト数
 x_3 : 外部変数を参照した回数
 x_4 : 外部変数を変更した回数

図1. 疲労予測式

偏回帰係数は、変更作業工数と変更作業行数の実績値を基にして、回帰分析法を用いて算出する。

4. おわりに

本稿では、外部変数の使用法がモジュール間結合による

プログラム疲労度へ影響を与えていることに注目し、外部変数の変更回数と参照回数を用いてプログラム疲労度を表現する方法について提案した。

今後プログラムの変更実験を行い、疲労予測式を確立する。

【参考文献】

- [1]荒ほか: "Cプログラムの疲労度に関する研究", 平成14年度日本大学工学部学術研究報告会, 2001
- [2]室屋ほか: "Cプログラム疲労の疲労に関する一考察", 電気関係学会東北支部連合大会, 2004
- [3]Myers: "高信頼性ソフトウェア複合設計", 近代科学社, 1976
- [4]中島ほか: "モジュール構造を考慮したプログラム疲労度の測定法の高度化" 第50回日本大学工学部学術研究報告会, 2007

表1. モジュール結合度による影響の伝わり方と変更作業工数の関係^[4]

モジュール結合度	呼び出し	処理手続き変更の影響の伝わり方	処理手続きの変更の影響の検討範囲	変更作業工数
データ結合	元*1	入力パラメータのみによって伝わる	入力パラメータのみに注目して、呼び出される側の処理手続きの影響のみに注目して調べる	小
	先*2	戻り値のみによって伝わる	戻り値のみに注目して、呼び出される側の処理手続きの影響のみに注目して調べる	小
スタンプ結合	元	入力パラメータのメンバーによって伝わる	入力パラメータと入力パラメータのメンバーについて注目して、呼び出される側の処理手続きの影響のみに注目して調べる	小
	先	戻り値のみにより伝わる	戻り値のみに注目して、呼び出される側の処理手続きの影響のみに注目して調べる	小
制御結合	元	制御情報を伝える入力パラメータのみによって伝わる	制御情報を伝える入力パラメータのみに注目して、呼び出される側の処理への影響のみに注目して調べる	小
	先	戻り値のみにより伝わる	戻り値のみに注目して、呼び出される側の処理手続きの影響のみに注目して調べる	小
外部結合	元	外部変数のみによって伝わる	外部変数のみに注目して、呼び出される側のモジュールだけでなく当該外部変数を使用しているモジュールについても注目して調べる	大
	先	外部変数のみによって伝わる	外部変数を参照している箇所と、変更している箇所を調べる	大
共通結合	元	共通データ (C言語の場合は外部変数) によって伝わる	共通データと共通データのメンバーについても注目して、呼び出される側のモジュールだけでなく当該外部変数を使用しているモジュールについても注目して調べる	大
	先	共通データによって伝わる	共通データを参照している箇所と変更している箇所と、共通データのメンバーを調べる	大
内容結合	—	アセンブラプログラム以外は存在しないので、検討の対象から外す	—	—

*1 元: 呼び出し元 *2 先: 呼び出し先

表2. 外部変数の変更により増大する問題対策作業

外部変数の変更形式	外部変数の変更により発生する問題	問題対策作業	問題対策作業工数の増大要因
値の変更	変更後の処理で外部変数を参照するために処理の不具合が生じる可能性がある	①変更前の外部変数の値の確認 ②変更後の外部変数を参照している処理の分析 変更された外部変数を参照できない場合があるときは場合分けを行い、外部変数参照手続きを制御する	①プログラム上での外部変数の値の分析作業の増大
参照場所の変更	参照場所を変更したことにより外部変数の値が変わり、処理に不具合が生じる可能性がある	①変更した参照場所での外部変数の値の確認 ②参照している処理の分析 変更された外部変数を参照できない場合があるときは場合分けを行い、外部変数参照手続きを制御する	②テスト作業の増大