

モジュール間の依存関係に着目したソフトウェアプロダクト計測システムの作成

6 T-6

高谷暢之[†] 松本健一[†] 鳥居宏次^{††}

†大阪大学 ††奈良先端科学技術大学院大学

1. まえがき

ソフトウェアの複雑さを定量的に評価するための尺度が数多く提案されている⁽⁴⁾。プログラムの複雑さに関しては、モジュール間の依存関係に着目した評価尺度がいくつか提案されている⁽¹⁾⁽²⁾⁽³⁾。しかし、モジュール間の依存関係に着目した複数の尺度を、同一対象に対して同時に適用するためのシステムは開発されておらず、尺度の性能比較などは現状では困難である⁽⁴⁾⁽⁵⁾。

本稿では、モジュール間の依存関係に着目した複数の複雑さ尺度の同時利用が可能なソフトウェアプロダクト計測システムの概要と主な機能について述べる。本システムは、モジュール間の依存関係を評価するために必要となる基本的な値(基本値)の計測機能と、尺度に基づく評価機能を持つ。

2. モジュール間の複雑さの尺度

本システムで利用可能な3つの尺度について簡単にまとめる。ただし、計測対象プログラムPはn個のモジュール(m_1, m_2, \dots, m_n)から構成されているものとする。

(1) Information Flow⁽¹⁾

モジュール間で受渡しされる情報の流れに基づく尺度である。モジュール m_i の複雑さ($INFO_i$)は次式で定義される。

$$\begin{aligned} INFO_i &= L_i * (FI_i * FO_i)^2 \\ FI_i &= CM_i + GV_i \\ FO_i &= CM'_i + GV'_i \end{aligned}$$

- L_i : モジュール m_i の行数
- CM_i : モジュール m_i を呼出すモジュール数
- GV_i : モジュール m_i が参照する大域データ数
- CM'_i : モジュール m_i が呼出すモジュール数
- GV'_i : モジュール m_i が更新する大域データ数

(2) Design Stability⁽³⁾

プログラムの保守による潜在的な波及効果を測る尺度である。プログラムPの複雑さ(DS_P)、及びモジュール m_i の複雑さ(DS_i)は次式で定義される。

$$\begin{aligned} DS_P &= 1 / (1 + \sum_{i=1}^n DLRE_i) \\ DS_i &= 1 / (1 + DLRE_i) \\ DLRE_i &= TG_i + \sum_{j=1}^n TP_{ij} + \sum_{j=1}^n TP'_{ij} \end{aligned}$$

- TG_i : モジュール m_i に定義される大域変数の構成要素数
- TP_{ij} : モジュール m_j がモジュール m_i を呼出す時に、 m_i が返すパラメータの構成要素数(呼出さない場合は0)
- TP'_{ij} : モジュール m_i がモジュール m_j を呼出す時に、 m_i が渡すパラメータの構成要素数(呼出さない場合は0)

(3) Nested Interface Complexity⁽²⁾

モジュール間で受渡しされる情報と、受渡しが行われる文のネスティングレベルに基づく尺度である。モジュール m_i の複雑さ(NIC_i)は次式で定義される。

$$NIC_i = FCN_i + AVN_i + EVN_i + RVN_i$$

FCN_i (Function Call Number)

$$= \sum_{\text{モジュール呼出し} \in m_i} (\text{引数の数} + 1) * (\text{その呼出しの } NLP)$$

AVN_i (Accept Variable Number)

$$= \sum_{m_i \text{ への呼出し} \in P} (\text{引数の数} + 1) * (\text{その呼出しの } NLP)$$

EVN_i (External Variable Number)

$$= \sum_{\text{外部変数の出現} \in m_i} (\text{その出現での } NLP)$$

RVN_i (Return Variable Number)

$$\begin{aligned} &= \left(\sum_{\text{返り数の出現} \in m_i} (\text{その出現での } NLP) \right) * \left(\frac{m_i \text{ を呼んでいるモジュール数}}{m_i} \right) \\ &+ \sum_{\substack{\text{返り数の出現} \in m_i \text{ から} \\ \text{呼ばれているすべてのモジュール}}} (\text{その出現での } NLP) \\ NLP_i &(\text{Nesting Level Plus}) \\ &= (\text{ネスティングレベル} + 1) \end{aligned}$$

3. プロダクト計測システム

3.1 構成

システムの構成を図1に示す。システムは次の2つの部分から構成される⁽⁵⁾。

(1) 基本値計測部

モジュールの構成要素の特性のうち、モジュール間の依存関係に影響を与えようと考えられるものを計測する。入力にはC言語で記述されたプログラム、出力は計測結果のテーブル(基本値テーブル)である。入力として与えられたプログラムは、構文解析部において、モジュール単位で構文解析され、同時に、計測が行われる。計測結果は、テーブル作成部に送られ、基本値テーブルとしてまとめられる。

(2) 評価値算出部

基本値テーブルの情報を利用して、複雑さ尺度に基づく評価値を算出する。評価値の算出は、各尺度に対応する評価部(評価関数)によって行われる。基本テーブルのどのデータを利用し、どの評価部を用いるかは、入力として与えられるコントロールファイル(Control File)に記述されている。制御部が、このコントロールファイルを解釈し、基本テーブルからのデータの検索、評価部の起動を行う。

3.2 基本値

基本値計測部では、モジュール毎に次の4種類の値を計測する。

(1) モジュールの大きさ (Volume)

モジュールの大きさを表す値。

$$\text{Volume: } \langle loc, st, tkn, \dots \rangle$$

行数, 実行ステートメント数, トークン数, ...

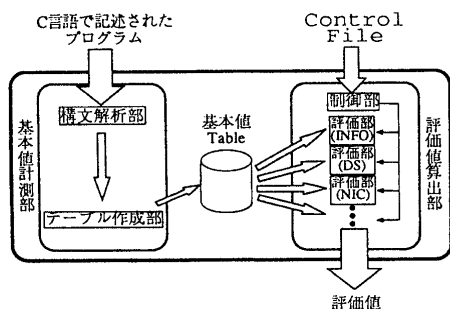


図1 システムの構成

(2) 制御構造 (Control Structure)

モジュール内での制御構造 (分岐 (Decision) や入れ子 (Nesting) など) に関する値。

Decision: $\langle dp, cyc, \dots \rangle$
分岐命令数, Cyclomatic 数, ...

Nesting: $\langle l, nl \rangle$
行番号, ネスティングレベル

(3) データ構造 (Data Structure)

モジュール内で用いられるデータ (大域変数 (Global) や局所変数 (Local) など) の構造に関する値。

Global: $\langle v, t, l \rangle$
変数名, 型, 宣言行番号

Local: $\langle v, t, l \rangle$
変数名, 型, 宣言行番号

(4) 他のモジュールとの依存関係 (Inter-Module Structure)

モジュール間の依存関係に関する値。次の3種類に分類される。

(4-1) 入出力 (Interface)

モジュール m_i と他のモジュールの間で受渡しされるパラメータ (Parameter) と戻り値 (Return-Value) に関する値。

Parameter: $\langle v, t, l \rangle$
名前, 型, 参照する行番号

Return-Value: $\langle t, l \rangle$
型, 戻り値を返す行番号

(4-2) 大域変数 (Global Variable)

モジュール m_i 内で参照する大域変数 (Refer) とモジュール m_i で定義され他のモジュールから参照される大域変数 (Referred) に関する値。

Refer: $\langle v, m_j, l, uf \rangle$
名前, 定義したモジュール, 参照する行番号, 更新フラグ

Referred: $\langle v, m_j, l \rangle$
名前, 参照するモジュール, 参照される行番号

(4-3) 呼出し (Call)

モジュール m_i からの呼出し (Call) とモジュール m_i に対する呼出し (Called) に関する値。

Call: $\langle l, m_j \rangle$
呼出しの行番号, m_i が呼出すモジュール

Called: $\langle m_j, l \rangle$
 m_i を呼び出すモジュール, 呼出される行番号

3.3 評価値の算出

評価値の算出においては、まず、評価に必要な情報が基本値テーブルから抜き出される。これは、3.1 で述べたように、本システムの

計測部が、特定の複雑さ尺度の定義に依存しない計測を行うためである。

尺度 NIC と、その評価に必要な基本値テーブルの情報の対応関係を図2に示す。図2では、例えば、NICのパラメータ FCN は、基本値 Call の組 (tuple) の構成要素 l と m_j をキーにして、Nesting の構成要素 nl の値、及び、 m_j の Parameter の組の数 (パラメータの数) を取り出すことにより算出できることが示されている。

このように、評価値算出部の各評価部は、図2に示したような関係と尺度の定義に基づいて評価値を算出する。

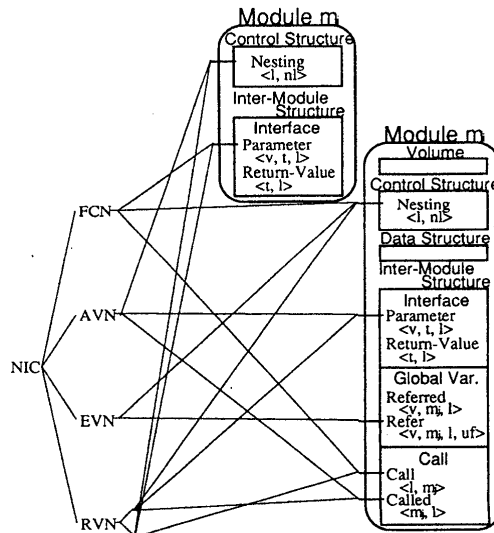


図2 尺度 NIC と基本値の対応

4. あとがき

現在開発中であるソフトウェアプロダクト計測システムの構成と主な機能について述べた。現在、実際のソフトウェア開発過程へのシステムの適用実験を計画中である。また、システムの出力となる複数の評価値を利用した、より効果的な複雑さ評価方法についても検討中である。

文 献

- (1) S. Henry and D. Kafura : "Software structure metrics based on information flow," IEEE Trans. Softw. Eng., SE-7,5, pp.510-519 (Sep. 1981).
- (2) 梁海述, 辻野嘉宏, 都倉信樹: "モジュール間依存度を考慮したプログラムの複雑度", 信学論 (D-I), J73-D-I, 11, pp.882-890 (Nov. 1990).
- (3) S. S. Yau and J. S. Collofello : "Design stability measures for software maintenance," IEEE Trans. Softw. Eng., SE-11,9, pp.849-856 (Sep. 1985).
- (4) H. F. Li and W. K. Cheung : "An empirical study of software metrics," IEEE Trans. Softw. Eng., SE-13, 6, pp.697-708 (Jun. 1987).
- (5) 高谷 暢之: "種々のプロダクトメトリクスの計測システムの作成", 大阪大学基礎工学部特別研究報告 (Mar. 1992).