

# オブジェクト指向によるプロトタイプ開発のための 工数見積りの検討

5ZC-10

上原 智, 水野 修, 菊野 亨  
大阪大学 大学院基礎工学研究科 情報数理系専攻  
E-mail: s-uehara@ics.es.osaka-u.ac.jp

## 1 まえがき

本研究ではオブジェクト指向開発によるプロトタイプ開発における工数見積りの方法についての提案を行う。従来から提案されてきているコスト見積り法では、必要となるデータが一般の開発現場で必ずしも揃わないという問題点がある。ここでは、オブジェクト指向プログラムの特性を活かした工数見積り法を与える。なお、本見積り法は現場への適用の容易さを考慮して幾つかのパラメータを含む評価式の形をとっている。更に、適用可能性を検討するために実プロジェクトのデータを利用した適用実験を行ったので、それについても紹介する。

## 2 オブジェクト指向の特性

オブジェクト指向プログラムにおける仕様変更作業の更新対象にはクラス、およびクラスに含まれる属性とメソッドがある。一方、更新作業は、1) 新規作成、2) 削除、3) 修正の3つの基本的な作業に分類でき、その内容によって更新に必要な労力は変化する。

更新対象であるクラス、属性、メソッドは型を持っている。その型によって、更新のための労力は変化する。ここでは型を次のように分類する [4]。

- 1) Void型: 返り値を持たない型。
- 2) 基本型: オブジェクト指向言語に元から組み込まれている型。
- 3) ライブラリ型: クラスライブラリに含まれている型。
- 4) 独自型: 開発対象のプログラムの中で新たに定義された型。

オブジェクト指向パラダイムが持つ特徴の中で、最も有効なものの一つがカプセル化による情報隠蔽である。情報隠蔽の度合いにより、考慮すべき有効範囲が変わるため、その更新に必要な労力も変化する。C++とJAVAにおける情報隠蔽を次の3種類に分ける。

- 1) private: この種類の属性やメソッドは他のどのクラスからも参照できない。
- 2) protected: この種類の属性やメソッドは継承されたクラス以外からは参照できない。
- 3) public: この種類の属性やメソッドは他のクラスから自由に参照できる。

<sup>0</sup>A Study of Effort Estimation for Updating Programs in Object-Oriented Prototyping Development  
Satoru Uehara, Osamu Mizuno, Tohru Kikuno  
Department of Informatics and Mathematical Science, Graduate School of Engineering Science, Osaka University

## 3 工数の見積り

プログラム  $P$  が仕様変更要求  $\sigma$  を受けてプログラム  $P'$  に更新される時、更新のための工数見積り式  $E(P, \sigma)$  を次のように定義する。

$$E(P, \sigma) = \sum_{i=1}^n E_{\text{requirement}}(R_i)$$

ここで、 $\sigma$  は  $n$  個の仕様変更  $R_1, \dots, R_n$  の集合を表す。仕様変更  $R_i$  に対処するための工数見積り式  $E_{\text{requirement}}(R_i)$  は次のように定義される。今ここでは、仕様変更  $R_i$  に対処するために  $m$  個のクラス  $C_1, \dots, C_m$  が更新されるものとする。

$$E_{\text{requirement}}(R_i) = \sum_{j=1}^m E_{\text{class}}(C_j)$$

次にクラス  $C_j$  を更新するための工数見積り式  $E_{\text{class}}(C_j)$  は次のように定義される。今ここでは、クラス  $C_j$  を更新する作業が  $p$  個の属性  $A_1, \dots, A_p$  と  $q$  個のメソッド  $M_1, \dots, M_q$  の更新から成るものと仮定する。

$$E_{\text{class}}(C_j) = \sum_{k=1}^p E_{\text{attribute}}(A_k) + \sum_{l=1}^q E_{\text{method}}(M_l)$$

メソッド  $M_l$  が更新される時の工数見積り式  $E_{\text{method}}(M_l)$ 、属性  $A_k$  が更新される時の工数見積り式  $E_{\text{attribute}}(A_k)$  は次のように定義される。

$$E_{\text{method}}(M_l) = \alpha \times w_{\text{upd}} \times w_{\text{type}} \times w_{\text{inf-h}} \times (1 + WCC(M_l) + WCM(M_l) + WPM(M_l))$$

$$E_{\text{attribute}}(A_k) = \beta \times w_{\text{upd}} \times w_{\text{type}} \times w_{\text{inf-h}}$$

ここで  $\alpha, \beta, w_{\text{upd}}, w_{\text{type}}, w_{\text{inf-h}}$  は2節で述べた特徴を表す定数である。また、 $WCC, WCM, WPM$  については4節で定義する。

- a)  $\alpha$ : メソッド更新のための基本点。
- b)  $\beta$ : 属性更新のための基本点。
- c)  $w_{\text{upd}}$ : 更新作業の内容による労力の違いを表現する重み係数。
- d)  $w_{\text{type}}$ : 更新対象の型による労力の違いを表現する重み係数。
- e)  $w_{\text{inf-h}}$ : 情報隠蔽の度合いによる労力の違いを表現する重み係数。

## 4 基本メトリクス

これまでに提案されているオブジェクト指向メトリクス[1, 2, 3]は主にクラスを対象としている。ここでは、メソッドを対象としたメトリクスを定義する。

### 1) WCC(Weighted Coupling Classes)

メソッド  $M_i$  が参照している  $i$  番目のクラスの重み  $wC_i$  の和であり、次式で定義される。

$$WCC(M_i) = \sum_{i=1}^n wC_i$$

### 2) WCM(Weighted Coupling Members)

メソッド  $M_i$  が参照している  $i$  番目の属性の重み  $wA_i$  の和であり、次式で定義される。

$$WCM(M_i) = \sum_{i=1}^n wA_i$$

### 3) WPM(Weighted Parameters of Method)

メソッド  $M_i$  の  $i$  番目のパラメータの重み  $wP_i$  の和であり、次式で定義される。

$$WPM(M_i) = \sum_{i=1}^n wP_i$$

## 5 適用実験

実験対象としたプログラムは、ある企業でOMT[5]に基づいてプロトタイプ開発されたアプリケーションである。本アプリケーションはオブジェクト指向データベースに格納されたイメージデータを表示する機能を実現している。プログラムはJAVAで記述され、10個のバージョンを持っている。

実験を行うにあたり、3節において定義した重み係数を決定した。決定方法は、今回対象にしたプログラムの開発者へのインタビューによるものである。設定したサンプル値を表1に示す。

表1のサンプル値を見積り式に当てはめ、工数の見積りを行う。具体的には、対象プログラムの各バージョンに見積り式を適用し、次のバージョンに更新するための工数を計算した。図1に実際の工数と見積りの間の相関を示す。相関係数は0.68となった。この結果から、提案法によって実工数を見積ることが可能であるといえる。

## 6 まとめ

本報告では、オブジェクト指向プログラムへの機能変更に必要な労力の見積り法について議論した。必要な各基本作業量の重みは開発者へのインタビューで求めた。開発事例データを用いた適用実験から限定的ではあるが有効性を確認した。

今後の課題として、重み係数の理論的根拠の確立、提案法のより多くのプロジェクトへの適用がある。

表1: 重み係数の設定例  
(a) 属性 (基本点  $\alpha = 1.0$ )

$W_{upd}$	新規作成	削除	修正
	1.0	0.2	N/A
$W_{type}$	基本	ライブラリ	独自
	1.0	1.1	1.2
$W_{inf-t}$	Private	Protected	Public
	1.0	1.0	1.0

(b) メソッド (基本点  $\beta = 2.0$ )

$W_{upd}$	新規作成	削除	修正	
	1.0	0.2	0.8	
$W_{type}$	Void	基本	ライブラリ	独自
	1.0	1.0	1.2	1.5
$W_{inf-t}$	Private	Protected	Public	
	0.8	0.8	1.0	

(c) メトリクス WCC, WCM, WPM

	基本	ライブラリ	独自
wC	0.01	0.01	0.01
wA	0.01	0.01	0.01
wP	0.02	0.03	0.03

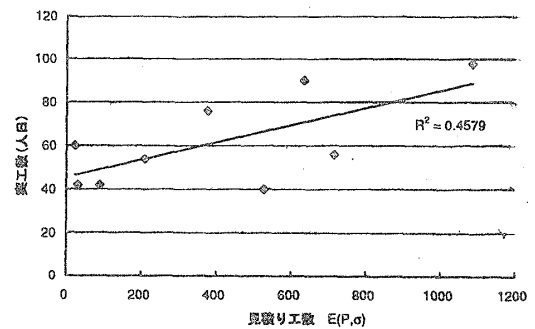


図1: 実工数と見積りの相関

## 参考文献

- [1] S. R. Chidamber and C. F. Kemerer: "A metrics suite for object oriented design", IEEE Trans. on SE, Vol.20, No.6, pp.476-493, 1994.
- [2] W. Li and S. Henry: "Object-oriented metrics that predict maintainability," Journal of Systems and Software, Vol.23, pp.111-122, 1993.
- [3] M. Lorenz and J. Kidd: *Object Oriented Software Metrics*, Prentice Hall, 1994.
- [4] B. Stroustrup: *The C++ Programming Language, 2nd edition*, Addison-Wesley, 1991.
- [5] J. Rumbaugh: *Object-Oriented Modeling and Design*, Prentice Hall, 1991.