1N-5

上流工程の UML クラス図を入力としたソフトウェアの保守性測定メトリクススイート

津田 直彦† 鷲崎 弘宜‡ 深澤 良彰‡
早稲田大学基幹理工学部情報理工学科† 早稲田大学‡

1 はじめに

ソフトウェア開発の早い段階からソフトウェアの保守のしやすさを評価できることが望ましい。品質の測定は蓄積された過去の資産に基づいて行われる場合が多い。蓄積された資産(データセット)をもとに、品質測定のための指標を統計処理で推定して決定している。しかし、上流工程では資産の蓄積が十分でない現状がある。

我々は、定量測定がしやすい UML 設計クラス図を対象とした品質測定手法を提案する。

本論文は以下の 2 点を Research Question とする。

RQ1：単一の UML クラス図において、構造的特徴に基づくメトリクスの測定値はどのようにばらついているのか？

RQ2：測定値のばらつきから、データセットなしに保守性を評価できるか？

上述の Research Questions に応えるために、我々は文献調査や経験に基づいて、保守性に影響すると推測される構造的特徴を定量測定するメトリクス群からなるスイートを GQM 法により構築した。さらに、66 個の実際のクラス図を対象として、測定結果のばらつきと、熟練者による定性評価を照らし合わせることを通じて、提案メトリクススイートの有用性を示す。

2 背景

2.1. UML クラス図

UML クラス図は、分析・設計領域の静的な構造を表現するために用いられる標準化された表記法である。実際には現場や設計ツールの違いにより、略記法や構造情報の表現に違いがある。

2.2. 関連研究

志水らは、設計モデルの記述内容の詳細さ(詳細度)によって、測定値の傾向が異なるメトリクスが存在することを示している[1]。そして、品質評価基準の推定には、詳細度を揃えたデータセットを用いる必要があることを述べている。

Lange らは、利用目的によって、同種のモデルでも抽象度が異なることを指摘した[2]。

Cruz は、ソースコードとモデルとでは、同じ内容のメトリクスでも得られる尺度が異なることを指摘し、尺度別の正規化が必要と述べた[3]。

2.3. 従来手法の設計図品質測定の課題

従来手法では、測定値を評価するにはデータセットが必要であった。

図 2-1の簡易なクラス図を例に、サブクラス数によって品質評価する場合を考える。まず、図 2-1と同じ分野かつ同じ抽象度で記述されたクラス図をデータセットとして蓄積する。次に、データセットに統計処理を施しサブクラス数の最適値を推定する。そして、図 2-1の各クラスの測定値と最適値を比較し、近似しているクラスには高い評価を与える。

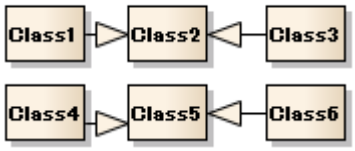

図 2-1 UML クラス図の例

我々はこのような従来手法は設計モデル対しては困難であると考える。2.2節から、従来手法では抽象度が揃ったデータセットが必要と考えられる。しかし、2.1節のように、設計工程の成果物は抽象度が揃えられていない場合が多い。適切な資産が蓄積されるためには、複数の開発が統一された形式で管理されている必要がある。

3 測定値のばらつきに着目した保守性評価の提案

我々は、開発分野固有の特徴の推定によらない手法として、単一の UML クラス図から問題箇所を検出する枠組みを提案する。

我々は GQM 法[4]を利用し、ソフトウェアの保守性とクラス図の属性との対応関係を求めるメトリクススイートを作成した。この際、過去の GQM 法を利用した研究[1][5]を参考にした。図 3-1のように、保守性が高いことを Goal とした場合に、何を調べればよいかを Question としてまとめた枠組みを提供する。また図 3-2のように、各 Question を何の Metrics 測定値によって評価すべきかを提供する。

Metrics Suite Measuring Maintainability for a UML Class Diagram in Upper Process

†Naohiko Tsuda, Department of Computer Science and Engineering, Fundamental Science of Engineering. Waseda University
‡Hironori Washizaki and Yoshiaki Fukazawa, Waseda University

Goal	SubGoal	Question
保守性が高い	解析性が高い	モジュール性を考慮した要素分割がされているか
		抽象化が適切か
		階層化が適切か
	変更性が高い	誤解なく扱えるか
		構造の説明が十分か
		構造が簡潔か
	安定性が高い	変更が外部に与える影響が大きすぎないか
		変更が外部から受ける影響が大きすぎないか
	試験性が高い	規模が大きすぎないか
		結合度が大きすぎないか

図 3-1　GQM 法により作成した保守性評価枠組み

Goal	SubG	Ques	SubQuestion	Metrics
保守性が高いか	解析性が高いか	抽象化が適切か	責務の量が適切か	メソッド数
				属性数
				子の数
				子孫の数
				自身を誘導可能な関連数
				自身が誘導可能な関連数
			要素の凝集性が高いか	LCOM
			共通部分をくくり出しているか	継承した属性数の割合
				継承したメソッド数の割合

図 3-2　解析性に関わる Metrics 一部抜粋

このメトリクスのスイートに基づいて、単一のクラス図から測定値を集め、測定値のばらつき度合いによって保守性を評価する。

2.3節の例を考える。図 2-1では、サブクラス数の分布をとると、Class2 と Class5 の偏差値が高く測定される。

4　評価

4.1. 評価実験

ET ロボコン 2010 年東京地区大会に提出された 66 個の設計モデルに対して、本手法を適用し、クラスに対するメトリクス測定値のばらつき度合いをモデルごとに以下の式で求めた。

$$Z\text{スコア} = \frac{\text{測定値} - \text{平均値}}{\text{標準偏差}} \quad (4.1)$$

$$\text{逸脱度合い} = \begin{cases} =0 & (0 \le |Z\text{スコア}| < 1) \\ =1 & (1 \le |Z\text{スコア}| < 2) \\ =2 & (2 \le |Z\text{スコア}| < 3) \\ =3 & (3 \le |Z\text{スコア}|) \end{cases} \quad (4.2)$$

$$\text{ばらつき度合い} = \text{逸脱度合いの総計} \quad (4.3)$$

平均値と標準偏差は同じパッケージ内で算出し、逸脱度合いはメトリクスごとに総計した。

各メトリクスのばらつき度合いと、設計モデルの審査員評価項目のうち設計モデルと関連が強い項目の合計点との関係を調べた。

4.2. 考察

図 4-1は、「審査員評価」と「クラスが持つ関連の数のばらつき度合い」の散布図である。

図 4-1からは得点が高いほど、ばらつき度合いが大きい傾向がわかる。設計時にクラスが担う責務を特定していない場合、パッケージ内には関連性の弱い要素が混在しやすく、モデルの評価が下がりやすい。また、このような場合、測定値の標準偏差が大きくなりやすいため Z スコアが小さくなり、ばらつき度合いが小さく算出される。

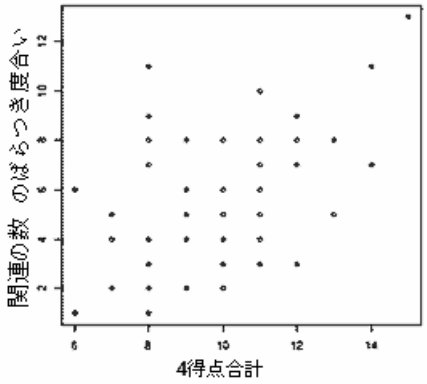

図 4-1　「関連の数」のばらつき度合いと審査員評価

RQ1：図 4-1の例では、ばらつき度合いが小さい方が、品質低下が疑わしいという結果を得た。このことから、単純に測定値のばらつきを抑えることが保守性の向上に繋がらないメトリクスが存在することがわかった。

RQ2：ばらつき度合いの定量化は可能と考える。しかし、保守性評価にはデータセットの代わりとして、「ばらついている方が良いか悪いか」という定性的な知見が分野ごとに必要となる。

5　おわりに

本手法の自動測定でレビュー活動を支援することで、評価の抜け漏れ解消と早期の問題発見ができ、開発コストの削減が期待できる。また、実際の保守活動と測定値のばらつきとの関連について、今後の研究が期待される。

6　参考文献

[1] 志水理哉ほか, ″設計段階における抽象度を考慮したソフトウェアの保守性評価枠組み″, 第 18 回 ソフトウェア工学の基礎ワークショップ FOSE 2011 in 浅虫温泉, 2011.

[2] Chiristian F. J. Lange, Michel R, ″Managing Model Quality in UML-based Software Development″, Proc. Of the 13th Int. Workshop on Software Technology and Engineering Practice, 2005.

[3] Ana Erika Camargo Cruz, ″Exploratory Study of a UML Metrics for Fault Prediction″, In Proceedings of International Conference on Software EngineeringICSE'10, Vol2, pp.361-364, 2010.

[4] Victor R.Basili, David M.Weiss, ″A Methodology for Collecting Valid Software Engineering Data″, EngineeringData, IEEE Trans. on SoftwEng., Vol.SE-10, No.6, pp.728-838, 1984.

[5] 株式会社オージス総研, Adqua, http://www.ogis-ri.co.jp/product/b-08-000001A6.html