

XMI を用いたオブジェクト指向設計モデル測定システムの構築

北野 貴稔†

野 中 誠††

東 基 衛†

早稲田大学大学院 理工学研究科†

早稲田大学理工学総合研究センター††

1. はじめに

現在，開発スタイルの主流は，Rational Unified Process や eXtreme Programming 等のラウンドトリップ型開発である．しかし，UML (Unified Modeling Language) 2.0 の登場によりモデルの表現力が増し，今後ラウンドトリップ型開発とモデル駆動型開発の混在型の開発スタイルが主流になっていくと考えられる．この開発スタイルの変化に伴い，今後上流工程におけるモデルの重要度が高まると思われる．従って，上流工程においてそのモデルを評価するためのメトリクスシステムが重要になるといえる．

現在，UML モデルをオブジェクト指向設計メトリクスで測定するシステムは幾つか存在する．しかしこれらの多くは CASE (Computer-Aided Software Engineering) ツールに依存しており，CASE ツールの多種多様なモデルの表現形式を直接的に測定することはできない．

本研究では CASE ツール非依存の XMI (XML Metadata Interchange) 形式の UML モデルをオブジェクト指向設計メトリクスで測定するシステムを構築することを目的とする．

2. 従来の UML メトリクスツール

これまで UML モデルをオブジェクト指向設計メトリクスで測定する研究は数多く行われている（例えば[1][2]）．これらの研究で開発されたメトリクスツールには次の3つの問題がある．

(1) 特定 CASE ツールへの依存

Rational Rose などの CASE ツールに依存しており，他の CASE ツールが出力するモデルの表現形式を直接的に測定することができない．

(2) 測定コンポーネントの新規追加の困難さ

新規のメトリクスを実現した測定コンポーネントを，測定システムを変更させることなく追加することができない．また，CASE ツール依存のプログラミング言語（Rose Script など）でなければ，測定コンポーネントを開発できない．

(3) 外部システムとの連携の問題

従来のシステムは，測定結果のデータを CSV (Comma Separated Value) 形式で出力するが，CSV 形式はデータの意味を表現しにくいいため外部システムとの連携が難しい．

3. システム化の要件とアプローチ

従来の UML メトリクスツールの問題を解決するために必要なシステムの要件およびアプローチは次のとおりである．

(1) CASE ツール非依存

CASE ツールは多様であるため，多種多様なモデルの表現形式から直接的に測定するためには，メトリクスシステムと CASE ツールは疎結合であることが望まれる．

本システムでは，CASE ツールが出力する XMI 形式の UML モデルを測定システムの入力として用い，測定システムに XMI 形式の UML モデルの解釈部を設けることで測定システムと CASE ツールとの疎結合を実現する．

(2) 測定コンポーネントの追加の容易さ

測定コンポーネントをシステムに容易に追加できるようにするため，システムが満たすべき必要条件は2つある．第一に，システムに変更を加えることなくメトリクスを追加できることである．第二に，測定コンポーネントの開発にメトリクスに関係のない知識を多く必要としないことである．

本システムでは，第一の要件を名前解決による動的クラスローディングを利用することで実現する．第二の要件は Relaxer を用いることで測定コンポーネントを Java と UML の知識のみで開発可能にすること，および測定コンポーネントの開発者がメトリクスの計算ロジック部分のみに集中できるように基本的サービスをフレームワークとして提供することで実現する．

(3) 測定結果の XML 出力

測定結果をプロセスやプロダクトの評価に用いるために，測定データを外部システムと連携する必要がある．その際に，測定データに意味を持たせ，連携をできるだけ容易にする必要がある．本システムでは，測定データのツール間交換仕様を Relax で規定し，測定データを XML 形式で出力可能にすることで実現する．

Object-Oriented Design Model Measurement System Using XML Metadata Interchange

† Takatoshi Kitano, †† Makoto Nonaka and † Motoei Azuma

† Graduate School of Science and Engineering, Waseda Univ.

†† Advanced Research Institute for Science and Engineering, Waseda Univ.

4. 提案システム

4.1 システムアーキテクチャ

システムアーキテクチャを図 1 に示す。

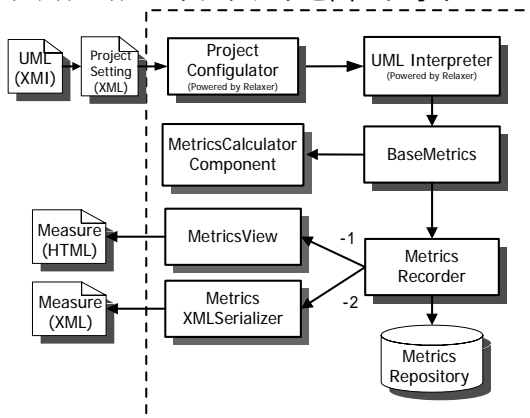


図 1 システムアーキテクチャ

以下にシステムの各構成要素について説明する。

Project Configurator

Project Setting を読み込み UML モデルのファイルパスを保持する。

UML Interpreter

Project Setting に指定された UML モデルの各要素を Relaxer オブジェクトに変換する機能を提供する。

BaseMetrics

名前指定によるクラスローディングの機能を利用し、測定コンポーネントを動的にロードする機能を提供する。また、全ての測定コンポーネントに共通する測定のための基盤サービスを提供する。

Metrics Calculator Component

測定コンポーネントの開発者が実装するコンポーネントである。メトリクスによる測定に必要な基盤サービスは BaseMetrics に実装されているため、測定コンポーネントの開発者は、メトリクスのアルゴリズムのみを実装すればよい。また、開発者は Metrics Calculator Component 中のクラスを継承することで、新規測定コンポーネントの追加を行うことが可能である。

Metrics Recorder

測定データを RDB (Relational Database) に永続化するための機能を提供する。

Metrics View

プロダクトの状態を可視化するサービスを提供する。

Metrics XML Serializer

規定したデータ交換仕様に基づき、測定データを XML 形式で出力する機能を提供する。

4.2 システムの動作解説

システムの動作は以下の通りである。

～ : XMI ファイルのロード

測定対象となる XMI 形式の UML モデルのファイルパスを Project Setting からロードし、UML Interpreter にそのファイルを渡す。

～ : メトリクスによる測定

UML Interpreter は UML モデルの要素を Relaxer オブジェクトとして保持する。BaseMetrics は、Metrics Calculator Component を用い測定を行う。

～ : 測定データの永続化

BaseMetrics は、測定結果のデータを Metrics Recorder パッケージに渡し、Metrics Repository に永続化する。

- 1 : 測定結果のデータの出力 (XML)

Metrics XML Serializer は、XML Serializer を用いて測定データを出力する。

- 2 : 測定結果のデータの出力 (HTML)

Metrics View は、Metrics Recorder から測定データを取得し、HTML 形式で表示する。

5. 評価と考察

本研究で提案したシステムアーキテクチャに基づき、C&K メトリクスを測定コンポーネントとして実装した。その結果、XMI 形式の UML モデルを C&K メトリクスで測定できること、C&K メトリクスを動的に追加できること、および測定結果のデータを XML 形式で表現できることを確認した。

このアーキテクチャにより、測定システムと CASE ツールの依存性の排除、測定コンポーネントの動的追加、および外部システムとの連携が可能になる。

6. おわりに

本稿では、CASE ツール非依存の XMI 形式の UML モデルをオブジェクト指向設計メトリクスで測定するシステムアーキテクチャを提案した。

本システムでは測定値の評価は対象外である。今後、プロジェクト管理システムとの統合により、メトリクスの全利用過程を対象としたい。

参考文献

- [1] Hyoseob Kim and Cornelia Boldyreff, Developing Software Metrics Applicable to UML Models, 6th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering, June 2002.
- [2] Mei-Huei Tang and Mei-Hwa Chen, Measuring OO Design Metrics from UML, 5th International Conference on the Unified Modeling Language, October 2002.