

---

 テクニカルノート
 

---

# JavaMLを用いたクラス設計メトリクス測定ツールの開発とその利用

阿 萬 裕 久<sup>†</sup> 坂 井 一 憲<sup>†</sup>,  
山 田 宏 之<sup>†</sup> 野 田 松 太 郎<sup>†</sup>

オブジェクト指向設計・開発において、クラス設計は最も重要な要素である。そのためのメトリクスとして、“パブリックなインスタンスメソッドの数”、“クラス階層の深さ”といった10種のメトリクスが知られている。これまでにC++やSmalltalkを開発言語としたソフトウェア開発プロジェクトに対しては、各メトリクスの測定・解析が報告されている。本論文は、現在主流となりつつあるJava言語についてクラス設計メトリクスの測定を行い、その解析結果を報告することで現場での一助となることを目的としている。ここでは、Javaプログラムを解析が容易なJavaML形式へと変換し、それを用いてクラス設計メトリクスの測定を行うツールを開発している。そして、そのツールを用いてJava2 SDK 1.3.1付属の508個のJavaプログラムの測定を行い、対象クラスの9割が含まれている範囲とそうでない範囲との境界値を各メトリクスの閾値として抽出している。これを用いることで、Java言語で書かれているクラスの設計品質を客観的に評価するための一種の指標が提供でき、ソフトウェア設計・開発の一助とすることが可能である。

## A Class Design Metrics Collector Using JavaML and Its Application

HIROHISA AMAN,<sup>†</sup> KAZUNORI SAKAI,<sup>†</sup> HIROYUKI YAMADA<sup>†</sup>  
and MATU-TAROW NODA<sup>†</sup>

Class design is one of important factors of object oriented designs and developments. In order to assess class design, there are some metrics such as “number of public instance methods”, “depth of class inheritances”. Those metrics have been applied to analyze software written in C++ and Smalltalk. A software tool which measures class design metrics for Java is developed in this paper. Java classes are transformed into JavaML for the tool. Measurements are done for 508 Java classes included in Java2 SDK 1.3.1. Then some threshold values for those metrics have been derived from the measurement reports. These thresholds will be aid to class construction in terms of the class design quality.

### 1. はじめに

計算機の多様化にとまぬい、そこで稼動しているソフトウェアも多種多様なものとなってきている。そのため、高品質で再利用可能なソフトウェアの効率的な設計・開発技術が必要である。その代表的なものとして、現在ではオブジェクト指向技術が広く普及している。しかしながら、設計・開発過程のほとんどは人手で遂行されるため、ソフトウェアの品質は設計者・開発者の能力に依存するところが大きい。それゆえ、ソフトウェアの品質を客観的に評価・予測することが重

要である。そのための尺度のことをソフトウェアメトリクス、あるいは単にメトリクスという。

オブジェクト指向設計・開発において、クラス設計は最も重要な要素であるため、本研究ではクラス設計を客観的に評価するためのメトリクスに着目する。そのようなメトリクスとして、CKメトリクス<sup>1)</sup>や“パブリックなインスタンスメソッドの数”<sup>2)</sup>等のメトリクスが提案されている。これまでにC++やSmalltalkを開発言語としたソフトウェア開発プロジェクトに対しては、各メトリクスの測定・解析が報告されている<sup>2),3)</sup>。

そこで、本論文は現在主流となりつつあるJava言語についてクラス設計メトリクスの測定を行い、その解析結果の報告が現場での一助となることを目的とする。具体的には、Javaプログラムを解析が容易なJavaML<sup>6),7)</sup>形式へと変換し、それを用いてクラス設

---

<sup>†</sup> 愛媛大学工学部情報工学科  
Department of Computer Science, Faculty of Engineering, Ehime University  
現在、株式会社ジャステック  
Presently with JASTEC Co., Ltd.

計メトリクスの測定を行うツールを開発する．そして，多数の Java プログラムに対して測定実験を行い，各メトリクス値について解析する．

以下，2 章では測定するメトリクスについて紹介し，3 章で開発した測定ツールについて概説する．4 章で実験および結果を示し，5 章でまとめと今後の課題を述べる．

## 2. クラス設計メトリクス

設計メトリクスとは，ソフトウェアの設計・開発フェーズにおいて，クラス設計を客観的に評価するための一種の指標である．文献 1)，2) では，クラス設計のためのメトリクスがいくつか紹介されている．本章ではその中から，クラスのインタフェース情報を用いて測定可能な 10 個のクラス設計メトリクスをあげる．詳細については文献 1)，2) を参照されたい．

- (1) 1 つのクラスに含まれるパブリックインスタンスメソッドの数：PIM
- (2) 1 つのクラスに含まれるインスタンスメソッドの数：NIM
- (3) 1 つのクラスに含まれるインスタンス変数の数：NIV
- (4) 1 つのクラスに含まれるクラスメソッドの数：NCM
- (5) 1 つのクラスに含まれるクラス変数の数：NCV
- (6) クラス階層のネストレベル：HNL
- (7) サブクラスによってオーバーライドされるメソッドの数：NMO
- (8) サブクラスによって継承されたメソッドの数：NMI
- (9) サブクラスによって追加されたメソッドの数：NMA
- (10) 特殊化指数：SIX (次式参照)

$$SIX = \frac{NMO \times HNL}{\text{メソッドの総数}}$$

## 3. 測定ツール概説

本論文では，前章で紹介した 10 個のクラス設計メトリクスに基づいて Java プログラムを測定し，それらを解析する．本章ではそのために開発した測定ツールの概要について説明する．本論文ではこのツールを CDMC と呼ぶ．なお，CDMC とは “Class Design Metrics Collector” の意味である．

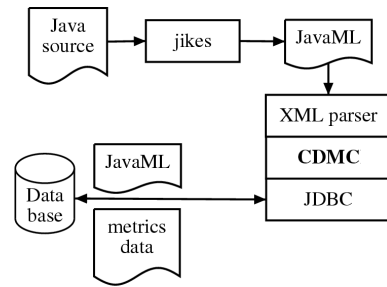


図 1 測定ツールを含めたシステム全体の概要図  
Fig. 1 Overview of the data collecting system.

測定ツール CDMC を含めたシステム全体の概要図を図 1 に示す．このシステムでは，

- (1) Java ソースファイルをコンパイラ Jikes<sup>5)</sup> を用いて JavaML<sup>6),7)</sup> (XML の一種．詳細については後述する) に変換する．
- (2) CDMC は XML パーサを介して JavaML を読み込み，各メトリクスの測定を行う．
- (3) その後，CDMC は JDBC を用いて JavaML および測定結果をデータベースへ格納する．また逆に，CDMC は必要に応じてデータベースから JavaML や測定結果を獲得することもできる．

### 3.1 Java ソースプログラムから JavaML への変換

本ツールの目的は，Java 言語で書かれたクラス宣言を用いてそのクラスの各メトリクスの値を測定することにある．そのためには Java プログラムの構文解析が必要となる．そこで，いったん Java プログラムを解析しやすい別の形式に変換する．そのような形式として，本研究では JavaML を用いる．

JavaML とは XML の一種であり，Java ソース・コード用のマークアップ言語である．この JavaML を用いる利点は，プログラムの分析を XML の分析へと置換できることにある．つまり，既存の豊富な XML ツールを使って Java プログラムの分析を容易に実現できる．詳細については文献 6)，7) を参照されたい．

### 3.2 その他

その他の仕様として，XML パーサには Xerces-J 1.4.4<sup>8)</sup> を使用した．データベースには PostgreSQL 7.0<sup>9)</sup> を，JDBC としては PostgreSQL 7.0 付属の jdbc7.0-1.2.jar をそれぞれ使用した．

## 4. 実験および結果

ここでは 2 章で紹介した 10 個のクラス設計メトリクスに基づいて多数の Java プログラムの測定を行い，それらの分布について調べる．測定ツールには 3 章

クラス階層において，最上位クラスから対象クラスまでの継承木の深さを意味する．

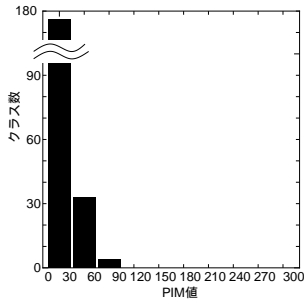


図 2 モデルクラスに対する PIM 値のヒストグラム

Fig.2 Histogram of PIM values for model classes.

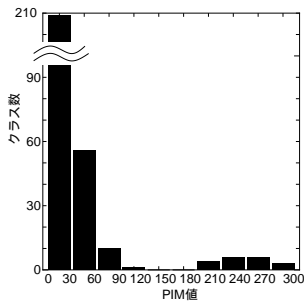


図 3 UI クラスに対する PIM 値のヒストグラム

Fig.3 Histogram of PIM values for UI classes.

で概説した CDMC を用いる。

#### 4.1 実験

本実験での測定対象とする Java ソースファイルには、Java2 SDK Version 1.3.1 に付属しているファイルの中から `java.awt`, `java.lang`, `java.util` 等 24 のパッケージに含まれている 508 個のクラスを用いた。

実験手順としては、以下の (1), (2) を繰り返すものとした。

- (1) Jikes を用いて、実験対象となる Java ソースファイルから JavaML を生成する。
- (2) 生成された JavaML に対し、測定ツール CDMC を用いてメトリクス値の測定を行う。

#### 4.2 結果

結果の一部として“1つのクラスに含まれるパブリックインスタンスメソッドの数：PIM”のヒストグラムを図 2, 図 3 に示す。ここでは文献 2) に従って、測定結果をモデルクラスとユーザインタフェース (UI) クラスとに分類している。

PIM 以外のメトリクスについても PIM 同様に右がりの分布が確認された。

そこで、これらを Java プログラムに対する 1 つの設計指標として利用できるよう、メトリクス値に対して閾値を設ける。今回は、対象クラスの 9 割が含まれている範囲とそうでない範囲との境界値をそのよう

表 1 各メトリクスにおける閾値  
Table 1 Thresholds for each of metrics.

メトリクス名	モデルクラス	UI クラス
PIM	33	54
NIM	41	60
NIV	8	14
NCM	8	5
NCV	7	27
HNL	5	4
NMO	8	6
NMI	17	41
NMA	29	25
SIX	0.6818	0.3077

な閾値として抽出した。結果として得られた各閾値は表 1 のとおりである。

表 1 の閾値を用いることで、Java 言語で書かれているクラスの設計品質を客観的に評価するための一種の指標を提供でき、ソフトウェア設計・開発の一助とすることが可能であると考えられる。

## 5. おわりに

本研究では、Java プログラムに対してクラス設計メトリクスの測定を行うツール“CDMC”を開発した。そして、それを用いて (1) PIM, (2) NIM, (3) NIV, (4) NCM, (5) NCV, (6) HNL, (7) NMO, (8) NMI, (9) NMA, (10) SIX の各メトリクスに基づいて Java 2 SDK 1.3.1 付属の 508 個の Java プログラムの測定を行い、それらの分布について調べた。そしてその結果から、対象クラスの 9 割が含まれている範囲とそうでない範囲との境界値を各メトリクスの閾値として抽出した。これを用いることで、Java 言語で書かれているクラスの設計品質を客観的に評価するための一種の指標を提供でき、ソフトウェア設計・開発の一助とすることが可能であると考えられる。

今後の課題として、開発現場での調査を行うことで、ここで示した閾値の有効性を検証することがあげられる。

## 参考文献

- 1) Chidamber, S.R. and Kemerer, C.F.: A Metrics Suite for Object Oriented Design, *IEEE Trans. Softw. Eng.*, Vol.20, No.6, pp.476-493 (1994).
- 2) Lorenz, M. and Kidd, J.: *Object-Oriented Software Metrics*, PTR Prentice Hall (1994). 宇治邦明 (監訳): オブジェクト指向ソフトウェアメトリクス, プレンティスホール出版 (1995).
- 3) Basili, V.R., Briand, L.C. and Melo, W.L.: A Validation of Object-Oriented Design Metrics

as Quality Indicators, *IEEE Trans. Soft. Eng.*, Vol.22, No.10, pp.751-761 (1996).

- 4) 山中祐介, 大畑文明, 井上克郎: プログラム解析情報の XML データベース化—提案と実現, コンピュータソフトウェア, Vol.19, No.1, pp.39-43 (2002).
- 5) <http://www-124.ibm.com/developerworks/opensource/jikes/>
- 6) Badros, G.J.: JavaML: A Markup Language for Java Source Code, *Proc. 9th International World Wide Web Conference* (2000). <http://www9.org/w9cdrom/342/342.html>
- 7) <http://www.cs.washington.edu/homes/gjb/JavaML/>
- 8) <http://xml.apache.org/xerces-j/>
- 9) <http://www.postgresql.org/>

(平成 14 年 3 月 25 日受付)

(平成 14 年 9 月 5 日採録)



阿萬 裕久 (正会員)

平成 13 年九州工業大学大学院工学研究科電気工学専攻(情報工学分野)博士後期課程修了。博士(工学)。同年愛媛大学工学部助手(情報工学科)。オブジェクト指向ソフトウェアメトリクスに関する研究に従事。電子情報通信学会, 日本ソフトウェア科学会, 日本ファジィ学会, IEEE 各会員。



坂井 一憲

平成 14 年愛媛大学工学部情報工学科卒業。同年株式会社ジャステック入社。在学中はクラス設計評価に関する研究に従事。



山田 宏之(正会員)

昭和 63 年大阪大学大学院工学研究科博士後期課程修了。工学博士。同年愛媛大学工学部助手(情報工学科)。同大学講師を経て, 平成 8 年より同大学助教授。ソフトウェアの要求獲得やソフトウェア保守に関する研究に従事。電子情報通信学会, 日本ソフトウェア科学会, 人工知能学会, IEEE, ACM 各会員。



野田松太郎(正会員)

昭和 44 年大阪市立大学大学院理学研究科博士課程修了。理学博士。昭和 45 年愛媛大学工学部助手(電子工学科)。同大学講師, 助教授を経て, 平成 3 年より同大学教授(情報工学科)。数式処理, 特に数値数式融合計算のアルゴリズムとシステム開発に関する研究に従事。最近では, Web コンピューティングやネットワークセキュリティに関する研究に興味を持っている。著書に『情報処理ハンドブック』(第 1 編 5 章「数値的方法」担当, 情報処理学会, オーム社), 『数式処理と数値計算—いかに結合させるか』(情報処理学会誌, 39 巻 2 号)等。その他論文・解説等多数。平成 4 年情報処理学会論文賞受賞。日本応用数理学会, 日本数式処理学会, ACM, IMACS 各会員。