

B-013

YAMLを用いたJavaプログラムコードの リファクタリングの提案*

菊池 大輔[†]茨城工業高等専門学校[‡]産業技術システムデザイン工学専攻[§]小飼 敬[¶]

茨城工業高等専門学校

電子情報工学科^{||}滝沢 陽三^{**}

茨城工業高等専門学校

電子情報工学科

1 はじめに

現在のソフトウェア開発は多種多様化しており、どのような変更にも素早く応じること、つまり、適応的であることが求められている。

その中で、処理内容はそのままにコードを書き換え、保守性と再利用性を高めるリファクタリングはとても効果的である。[1]

しかし、リファクタリングは、その適用箇所を見つけることが難しい。また、適用箇所が分かったとしても、リファクタリングを実際に行う段階において、ヒューマンエラーが容易に混入してしまう。

そこで、リファクタリングについてのこれらの問題を解決するために、リファクタリングを補助するためのツールが必要だと考えた。

本研究では、データシリアライズ用フォーマットであるYAML(ヤムル)を用いたリファクタリング方法を提案し、それを用いたツールの開発を目指した。

2 YAML

YAML(YAML Ain't Markup Language)とは、構造化データやオブジェクトを文字列にシリアライズするためのテキストデータ形式の一種であり、Ruby言語を用いることで、XMLとの相互変換が可能である。[2]

インデントを使って階層構造を表現するため、XMLと比較して可読性により。例えば、チェスのプレイヤーのデータをXMLとYAMLに表現したものをそれぞれ以下に示す。[3]

XMLの場合：

```
<players>
  <player id="kramnik"
    name="Vladimir Kramnik"
    rating="2700"
    status="GM" />
  <player id="fritz"
    name="Deep Fritz"
    rating="2700"
    status="Computer" />
  <player id="mertzt"
    name="David Mertz"
    rating="1400"
    status="Amateur" />
</players>
```

YAMLの場合：

```
players:
  Vladimir Kramnik: &kramnik
    rating: 2700
    status: GM
  Deep Fritz: &fritz
    rating: 2700
    status: Computer
  David Mertz: &mertz
    rating: 1400
    status: Amateur
```

3 JYaml

JYamlとは、JavaでYAML形式のデータを用いるためのライブラリであり、JavaのインスタンスオブジェクトをYAML形式にシリアライズすることと、逆にYAML形式のデータからインスタンスオブジェクトへデシリアライズすることをサポートする。[4]

*Proposal for Refactoring Java Program Code with YAML

†Daisuke Kikuchi

‡Ibaraki National College of Technology

§Advanced Course for Information

¶Kei Kogai

||Electronic and Computer Engineering

**Yozo Takizawa

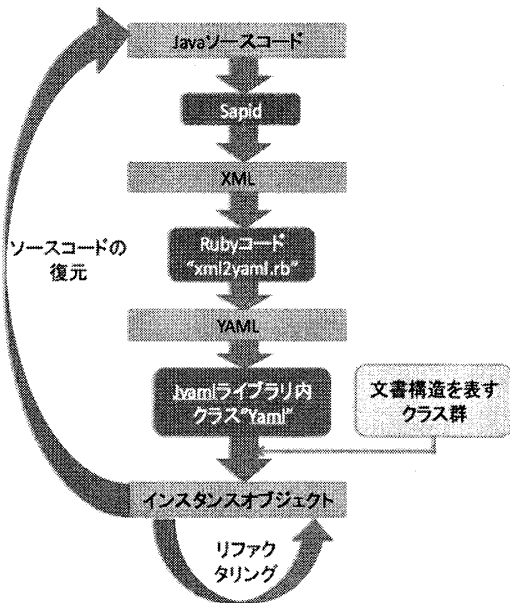


図 1: システム構成

Fig.1: System configuration

4 ツールの開発

4.1 概要

本システムでは、Java ソースコードの文書構造を YAML 形式データに変換し、それを Java のインスタンスオブジェクトへデシリアライズし、そのオブジェクトに対し操作を行うことによって、ソースコードに手を触れることなく、リファクタリングを行う。

4.2 構成

図 1 に本開発ツールのシステム構成を示す。

CASE ツールプラットフォームである Sapid を用いて Java ソースコードを解析し、XML 形式の解析結果を出力させ、その結果を Ruby 言語で作られた XML から YAML に変換する変換器で YAML 形式に変換し、その後 YAML 形式のデータを JYaml を用いてインスタンスオブジェクト化する。

インスタンスオブジェクト化する際、元のソースコードのインポートやクラス、フィールド、メソッドなどの構造を表すためのクラスを前もって用意する必要がある。リファクタリング自体は、インスタンスオブジェクトを操作することによって実現される。操作のためのメソッドを用意しておき、それらを用いて、リファクタリング適用箇所の特定、実際のリファクタリング、Java のソースコードへ復元することが可能となる。

5 課題・発展

5.1 課題

本システムの課題として以下 2 つの機能を実装することで、ツールとして使用可能となる。

- リファクタリング方法のテンプレートをある程度作成しておき、そのためのメソッドを用意しておく。
- 1 つのツールで、Sapid、XML から YAML への変換器を呼び出せるようにする。

5.2 発展

YAML を用いるために今回は JYaml ライブラリを用いたが、YAML に対する知名度はまだまだ低く、YAML を用いるためのライブラリは数が少ない。YAML を用いるためのライブラリが多くなってくれば、このツールよりも効率的な操作を実現することが可能であると考えられる。

6 まとめ

リファクタリング操作において、ヒューマンエラーが混入してしまうことは、生産性の低下につながる。本研究では、ツールによって箇所の特定、実際の操作を行うことで、確実なリファクタリングを行うことができると考えられる。

参考文献

- [1] マーチン・ファウラー：リファクタリング - プログラミングの体質改善テクニック、ピアソン・エデュケーション (2000)
- [2] The Official YAML Web Site
<http://yaml.org/>
- [3] XML の論考：YAML は XML に改良を加える
<http://www.ibm.com/developerworks/jp/xml/library,matters23/jp>
- [4] JYaml - Yaml library for the Java language
<http://jyaml.sourceforge.net/>