

数値計算言語を統合したオブジェクト指向言語による制御系設計

Control System Design using Object Oriented Language Integrated with Numerical Computation Language

古賀 雅伸 (九工大)

松永 隆徳 (九工大)

Masanobu KOGA, Kyushu Institute of Technology, Kawazu 680-4, Iizuka-shi, Fukuoka
Takanori MATSUNAGA, Kyushu Institute of Technology

In this paper, we propose a new method to develop a numerical computation program in Java by using the annotation for Numerical Computation. We develop the annotation that define a numerical computation part in Java, and this processing system that execute annotation part. By the annotation for numerical computation, the user can easily develop the numeric calculation program in Java only by the knowledge of mathematics.

Key Words: Object Oriented, Numerical Computation Language, Control System Design

1. はじめに

近年、ソフトウェア開発の大規模化・複雑化が急速に進んでいる。これは行列や複素数計算など科学技術計算を扱う数値計算分野のプログラム開発においても同様である。

従来、数値計算を扱うプログラム開発方法は、大きく分けて2つに分類できる。ひとつは、Java 言語など広く普及している言語を用いた開発。もうひとつは、Matlab や MaTX[1]などの数値計算言語 NCL(Numerical Computation Language)を用いた開発である。前者は、オブジェクト指向言語による再利用性の高いプログラム開発が可能であるが、数式部分の可読性・保守性が低下してしまう。一方、後者は可読性・保守性に優れた数式表現が可能であるが、再利用性や拡張性が低いプログラムとなってしまう。

これまで、NCL 用のプログラムを C 言語や Java 言語用に変換する処理系や、Java 上から数値計算エンジンを呼び出す方法[2]が提案されてきたが、Java 上での数式表現の可読性・保守性の低さを解消するまでには至っていない。

そこで、本研究では、Java 言語と NCL を組み合わせた方法として、Java 言語のアノテーション技術を用いた数値計算言語の利用に関する研究を行っている。具体的には、Java のアノテーション技術を用いて、Java プログラム中に数値計算言語を埋め込むことを目的としている。本研究では、Java プログラム中に数値計算言語を表現するための数値計算用アノテーションを開発し、そのアノテーションを処理するためのアノテーションプロセッサを開発する。これにより、利用者は数学的知識だけで Java 上で数値計算プログラムを開発することが可能となる。

なお、本研究で扱う数値計算言語は、科学や工学に必要な数値および数式計算をサポートする記述性に優れたプログラミング言語である MaTX を用いる。

2. 関連研究

2.1. インライン・アセンブラ インライン・アセンブラとは、C言語などで記述されたプログラムの一部に、アセンブラコードを埋め込むための技術である。これによって、最

小限のアセンブリ言語の知識で、マイクロプロセッサの性能を極限まで引き出して実行速度を高めたり、メモリ使用量を極限まで減らしたりなどの実行効率を上げることが可能となる。

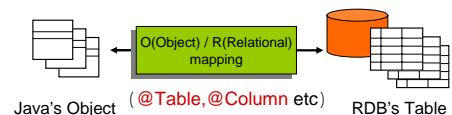
2.2. アノテーション アノテーションとは、あるデータに対して関連する情報(メタデータ)を注釈として付与することを意味し、プログラムに開発者の意図を付加する技術である。Java標準では、対象となるメソッドがスーパークラスのメソッドをオーバーライドしていることを示す@Overrideなどのアノテーションが用意されている。

アノテーション技術の応用として、アノテーションは異なるデータモデルを結び付ける技術として利用されている。Javaオブジェクトからリレーショナルデータベース(RDB)のテーブルを利用することができるO/R(Object/Relational)マッピングツールやJavaからXMLを扱うことができるJava/XMLバインディングツールがそれにあたる(Fig.1)。

3. 数値計算アノテーションの概要

本研究で開発した数値計算アノテーションを「@MaTX」と表現し、これをプログラム中に付加することで Java プログラムにおける数値計算部分であることを明示する。

• Java and RDB (Relational Database)



• Java and XML

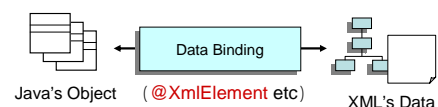


Fig.1: The example of annotation

3.1. プログラム上での定義 数値計算アノテーションを利用するには、フィールド宣言時に以下のようにアノテーションの宣言と値の定義が必要である。

```
@MaTX("アノテーションの値")
フィールドの型 フィールド名;
```

数値計算アノテーションの値の定義では、行列の定義や数式表現を記述する。例えば、 2×2 の行列を定義するときは「@MaTX("[[1 2][3 4]]")」と記述する。また、数式を用いる場合は「@MaTX("A*B+C*D")」と表現する。

4. アノテーション処理システムの開発

本研究では、数値計算アノテーションを処理するためのアノテーション処理システムを開発する。本システムは、数値計算アノテーションの情報を Java で実行可能な状態にする目的から、アノテーション情報の Java コード化を行う。Fig.2 はその処理フローである。

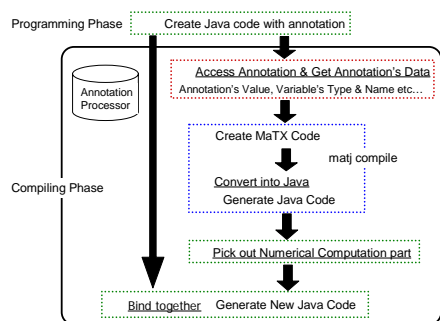


Fig.2: The processing flow of Annotation

まず、アノテーションの値として渡される行列等の定義や数式表現から MaTX プログラムを作成する。次に、MaTX プログラムを等価な Java プログラムに変換する処理系(matj コンパイラ)[3]を用いて、Java コード化を行う。そして、Java 用に変換したプログラムから数値計算アノテーションに関係する箇所を抽出する。最後に、アノテーションを付加した Java プログラムと、アノテーション情報を Java コード化したものを結合し、新しいプログラムとして生成する。

5. 例題

本研究で開発したアノテーションによる数値計算プログラムの開発方法を従来の開発方法と比較する。例として、(1)式に示すリカッティ方程式の解法のひとつである有本-ポッターの解法[4]を実装するプログラムを作成する。

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \quad (1)$$

Fig.3 がアノテーションを用いたプログラム実装、Fig.4 が従来のプログラム実装(Java 数値計算ライブラリ[5]を用いた方法)で表現したものである。今回、Fig.3 と Fig.4 について、プログラムの可読性・保守性の観点で評価を試みる。

```
@MaTX("[[0 0][0 1]]")
Matrix A = null;

@MaTX("[[1 1][0 1]]")
Matrix B = null;

@MaTX("[[1 1][1 1]]")
Matrix Q = null;

@MaTX("[[1 1][1 1]]")
Matrix R = null;

// The Arimoto-Potter Algorithm
@MaTX("[[A, -B*R~*B^T]-Q, -A^T]")
Matrix H = null;
```

Fig.3: Code with Annotation

```
// Define Matrix "A"
Matrix A = new RealMatrix(new double[][]{{0,0},{0,1}});

// Define Matrix "B"
Matrix B = new RealMatrix(new double[][]{{1,1},{0,1}});

// Define Matrix "Q"
Matrix Q = new RealMatrix(new double[][]{{1,1},{1,1}});

// Define Matrix "R"
Matrix R = new RealMatrix(new double[][]{{1,1},{1,1}});

// The Arimoto-Potter Algorithm
Matrix H = A.appendRight(B.unaryMinus()
    .multiply(R.inverse()).multiply(B.transpose()))
    .appendDown(Q.unaryMinus())
    .appendRight(A.unaryMinus().transpose());
```

Fig.4: Code with Class Library

結論として、アノテーションによって数学的表現で表現可能となったことから数値計算プログラムの可読性・保守性が向上した。

6. おわりに

本研究では、Java のアノテーション技術を用いた数値計算プログラムの開発方法を提案した。これにより、数学的表現と同等の記述で、Java で数値計算プログラムの開発が可能となった。本稿では、基本的な行列計算を例として紹介したが、今後は複雑な計算を必要とする制御系設計の場面において、その有効性を検証する。

文 献

- [1] 古賀雅伸, 制御・数値計算のための MaTX, 東京電機大学出版局, February 2000.
- [2] 松木毅, オブジェクトモデル化に基づく Java による数値計算エンジンの開発, 九州工業大学 修士論文, 2004.
- [3] 石田聡, 古賀雅伸. プログラミング言語のオブジェクトモデル化に基づく Java による数値計算言語の利用, 平成 14 年 電気学会 電子・情報・システム部門大会, 2002.
- [4] 小郷寛, 美多勉. システム制御理論入門. 実教出版株式会社. 1979.
- [5] 松木毅, Java 数値計算パッケージの開発と制御系設計への応用, 九州工業大学 卒業論文, 2001.

評価方法	従来方式	アノテーション方式
可読性	低い	高い
保守性	低い	高い