

A-026

スクリプトインタフェースに基づく数値計算エンジンの再利用性向上

A Method for Improving Reusability of Numerical Computing Engine Based on Scripting Interface

古賀 雅伸[†]

Masanobu Koga

田中 俊行[†]

Toshiyuki Tanaka

1. はじめに

現在、様々な制御系のモデリングやシミュレーションのツールが利用されている。例えば、Simulink[1]やScicos[2]、Jamox[3]などである。これらは、SimulinkではMATLAB、ScicosではScilab、JamoxではMaTXEngine[4]というように内部ではそれぞれに対応した数値計算エンジンを利用している。

一般に、アプリケーションからエンジンを利用する際には、エンジン毎のインタフェースを通して利用する。アプリケーション側が対応すれば、複数のエンジンを切り替えて使用することもできる。しかし、それぞれのエンジン毎にインタフェースは異なるので、アプリケーションの開発者は各エンジンのインタフェースについての知識が必要となり、開発者にとっての負担が大きかった。

2006年12月に正式リリースされたJava SE 6では、Javaアプリケーションに関するこの問題を解決する一つの手法として、JSR223[5]が導入された。これはJava上でスクリプト言語を実行するためのAPIであり、スクリプトエンジンのインタフェースを統一するものである。

エンジンのインタフェースが統一されることにより、アプリケーション側では、図1のように、JSR223のインタフェースに関する情報だけで、複数のエンジンを切り替えて、利用できる。また、エンジンの汎用性が高くなるため、様々なアプリケーションで、エンジンを採用できる。現在、JSR223に対応したエンジンが徐々に報告されている。しかし、JSR223に対応した数値計算エンジンはまだ存在しない。

そこで、本研究では、数値計算エンジンMaTXEngineをJSR223の仕様に対応させることにより、スクリプトインタフェースの有効性を確認し、MaTXEngineの再利用性を高めることを目的とする。

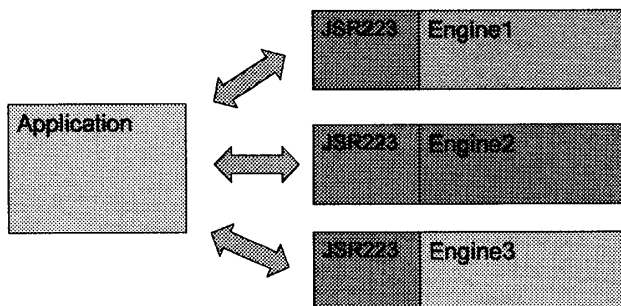


図1: エンジン切り替えの概念図

2. MaTXEngine

MaTXEngineは、数値計算言語MATX[6]の任意の命令を文単位で解釈、実行する、数値計算エンジンである。ユーザとの命令・結果のやりとりによって対話的に利用できる。

2.1 オブジェクトモデル

本研究では、MATXで記述された命令をインタプリトするため、MATXオブジェクトモデルで、ソースコードをオブジェクト化する。

2.2 エンジンの動作

MaTXEngineは、受け取った命令を、MATXオブジェクトモデルを利用してオブジェクトモデル化し、オブジェクトモデルの実行機能を動作させることで、自身の内部状態を更新する。この内部状態を参照・利用しながらユーザは命令を与える。これにより、ユーザは数値計算を進めることが可能となる。

3. JSR223への対応

MaTXEngineは先に述べたように、それ単体で使用できるのではなく、エンジンを使用するためのインタフェースが必要となる。そこで、JSR223に対応したMaTXEngineとJavaアプリケーションとのインタフェースであるMaTXDriverを作成した。このインタフェースはアダプタとなっている。それらの関係図を図2に示す。

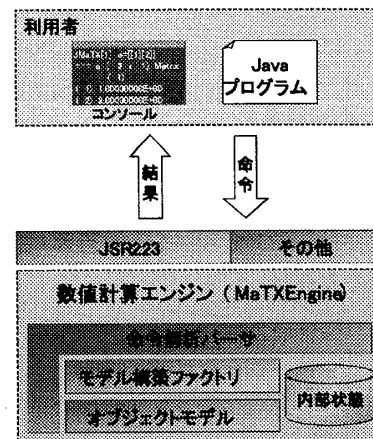


図2: アプリケーションとエンジンの関係

3.1 JSR223のAPI

JSR223のAPIを表1に示す。JSR223の仕様に対応するエンジンのインタフェースの作成には、最低限ScriptEngineインタフェースを実装しなければならない。このインタフェースを実装することでエンジンに送った命令を実行できる。また、オプションとしてInvocable、Compilableインタフェースを実装することで、

[†]九州工業大学, KIT

表1に示したようなそれぞれの機能を持たせることもできる。

表1: Scripting API のインタフェース

インタフェース	説明
Bindings	キーと値のペアのマッピング
Compilable	事前コンパイル可能なメソッドをもつインタフェース。
Invocable	以前に登録された関数呼び出しをするメソッドを持つインタフェース。
ScriptContext	スクリプトエンジンとスコープ設定された Bindings などのオブジェクトとを接続。
ScriptEngine	仕様の全ての実装で完全に機能することが必要なメソッドを持つ基本インタフェース。
ScriptEngineFactory	各スクリプトエンジンの説明およびインスタンス化を行う。

3.2 MaTXDriver の機能

本研究で作成した MaTXDriver は主に以下の機能を持つ。

- MaTX の命令と実行
- 変数の定義と取得

また、MaTXDriver は Invocable インタフェースを実装しているため、以下のオプション機能も使用できる。

- Java アプリケーションからエンジン側の関数呼び出し

3.3 jrunscript での数値計算エンジンの利用

jrunscript とは Java SE 6 で標準提供されるコマンドラインスクリプトシェルであり、対話型モードとバッチモードの両方のスクリプト実行をサポートしているアプリケーションである。これはスクリプト言語に依存しないシェルであり、デフォルトでは、JavaScript エンジンの Rhino が使用される。

図3のように、エンジンのクラスパスとエンジン名を指定すると、指定したエンジンを使用した jrunscript が起動する。この例のように実行すれば、エンジンに MaTXEngine を用いた jrunscript が起動し、入力待ちの状態になる。

```
C:>jrunscript -cp matj-1.2.jar;matx.jar;
nfc-1.2.jar -l matx
matx>
```

図3: jrunscript の実行例

4. 性能評価

今回作成した MaTXDriver の有効性を確認するために、MaTXEngine を直接呼び出すアプリケーションである JMaTX[4] と比較する。実験環境を表2に示す。また、Java VM のバージョンは build1.6.0-b105,mixed mode である。

表2: 性能評価実験の実験環境

CPU	メモリ	OS
AthlonXP 2500+	1GB	WindowsXP SP2

4.1 性能測定実験

エンジンの性能を評価するために、以下の4つの問題での性能測定を行う。

- 実行列 (5000 * 5000) の加算
- 実行列 (1000 * 1000) の乗算
- 実行列 (1000 * 1000) の逆行列
- 実行列 (500 * 500) の固有値

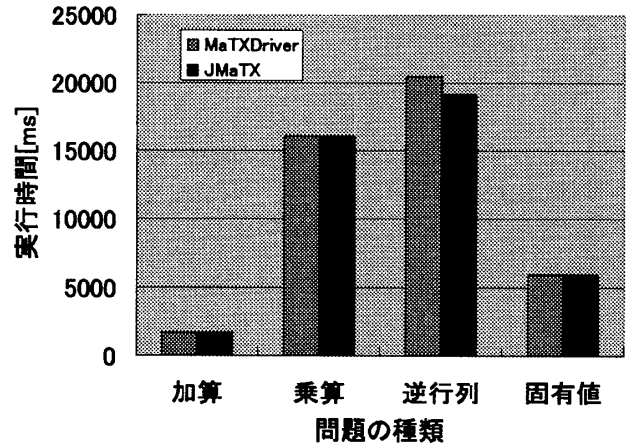


図4: エンジンの性能測定結果

MaTXDriver を通じて MaTXEngine を利用する場合でも、JMaTX との時間差はほとんどない。よって MaTXDriver は速度を落とすことなく有効に使えることがわかる。

5. まとめ

本研究では JSR223 の仕様に対応した数値計算エンジンのインタフェース MaTXDriver を開発した。これを用いることで Java アプリケーション側から簡単に数値計算エンジンを使用できる。また、JSR223 の仕様に対応したことにより、様々なアプリケーションが利用できる形となり、再利用性が高まり、スクリプトインタフェースの有効性が確認できた。今回は、Compilable インタフェースは実装していないが、今後、その実装を行うことにより、エンジン使用の際の高速化が期待できる。

参考文献

- [1] The MathWorks. Matlab. <http://www.mathworks.com/products/matlab/>.
- [2] Scilab. <http://www.scilab.org/>.
- [3] 松本 明紘. XML バインディングを用いた制御系のモデリング・シミュレーションプラットフォームの開発, 2007. 第51回システム制御情報学会研究発表報告会.
- [4] 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.
- [5] JSR. <http://jcp.org/en/jsr/all/>.
- [6] 古賀 雅伸. 制御・数値解析のための MaTX. 東京電機大学出版局, February 2000.