Java仮想機械の形式仕様とその検証

奥村 滋
言語設計学講座
北陸先端科学技術大学院大学 情報科学研究科

2000年2月15日

本研究では、Java仮想機械の一部であるバイトコード検証系の仕様を代数仕様記述言語CafeOBJにより記述した。

1 本研究の目的

Java言語のセキュリティーモデルは、バイトコード検証系、(アプレット)クラスローダ、Javaセキュリティーマネージャの3つによってセキュリティーを保っているといわれている。この中で、バイトコード検証系は、他の2つのものの土台となっており信頼性の求められるものである。しかし、バイトコード検証系の仕様が自然言語を用いた曖昧な形でしか記述されていないため、解釈に違いにより実装がうまくいっていないことがある。もし、バイトコード検証系が機能を果さないことになると、セキュリティー上の問題となるのは明らかである。

そこで本研究では、バイトコード検証系がどのようになっているかを自然言語で記述された仕様書に基いて、形式的に仕様を記述し、記述したものを基に、実際のコードの擬似的なものによってCafeOBJで実行し、その仕様について検討することが目的である。

2 本研究の背景と特色

Java言語の使用形態を考えてみると、ネットワークを介した使い方ができるためダウンロードしてきたJava言語のクラスファイルが悪意を持った者によって改竄されたクラスファイルであったり、欠陥のあるバイトコンパイラによって生成されたクラスファイルである可能性がある。そして、その改竄されたクラスファイルをそのままJava仮想機械上で実行してしまうと実行に破綻を生じる可能性がでてきてしまう。例として、コンピュー

ターのハードウェアやファイルシステムにダメージを与えるようなプログラムであったり、また、コンピューターをクラッシュさせたり、使えない状態にするクラスファイル。また、セキュリティー上で問題となるようなものとしてコンピューターについての情報やファイルに保存してあるようなデータを他のところに漏洩するような行為をするクラスファイルなどが考えられる。これらのクラスファイルを実行してしまうようなことは避けなくてはいけない。

このようなことにならないための一つの方法として、Java 仮想機械上で実行する前に読み込んだクラスファイルを検査するということをしている。検査の内容として、クラスファイルが最初の 4 バイトは決められたマジックナンバーを保持しているかどうか、また実際の実行をせずに実行をシミュレートして検査などをする。この実行をシミュレートする検査の部分を特にバイトコード検証系と呼ぶ。

3 Java 仮想機械とバイトコード検証系

バイトコード検証系で行っている検査内容は、クラスファイルに埋め込まれた実行列を読み実行をシミュレートすることである。このことで、オペランド・スタックがオーバーフローやアンダーフローをしないということや、使用、ストアするすべてのローカル変数は有効であるかどうか、そして、Java 仮想機械の命令に対する引数が有効な型であるかを保証している。この保証によってコンピューターをクラッシュさせたりするクラスファイルを Java 仮想機械上で実行しないようにすることができると予想できる。

しかし、バイトコード検証系を含め、Java 仮想機械の仕様は自然言語を用いた曖昧な形でしか記述されていない。そして、自然言語を用いたものによりバイトコード検証系を開発した場合、解釈の違いにより本来目的としていた仕様とは違うものが出来てしまう可能性がある。もし、本来目的としていた仕様とは違うものである場合、コンピューターをクラッシュさせるようなクラスファイルが検査を通ってしまう可能性が出てきてしまう。

そこで、自然言語で書かれた仕様を基に、代数仕様記述言語 CafeOBJ によってバイトコード検証系の仕様を記述する。また、形式的な手法で仕様を記述した上に、CafeOBJ は実行可能であるために、記述した仕様を用いて実際に実行することによりその仕様を実際に検査することができる。

4 研究成果と結論

クラスファイルを Java 仮想機械上で実行しても実行が破綻しないかどうかを検査するバイトコード検証系の仕様について代数仕様記述言語 CafeOBJ により記述した。この仕様を用いて、擬似的なバイトコードのを実際に CafeOBJ 上の実行環境により実際の検査をシミュレートした。しかし、形式的に記述した仕様によってシミュレートしただけでは、バイトコード検証系の検査を通ったクラスファイルを実行しても Java 仮想機械上の実行

が破綻しないという保証はできない。そこで、Java 仮想機械自体もモデル化をしてバイトコード検証系と同様に CafeOBJ によって記述し、バイトコード検証系を通ったものについては Java 仮想機械上で実行しても実行が破綻しないことを示す必要があると考える。

5　本論文の構成

本論文の構成は以下のようになっている。

第 2 章では、本研究で使用する代数仕様記述言語 CafeOBJ について説明を行なっている。

第 3 章では、形式的に仕様を記述したバイトコード検証系や Java 仮想機械についての説明を行なっている。

第 4 章では、バイトコード検証系についてより細い説明を行なっている

第 5 章では、バイトコード検証系をどのようにモデル化をし、どのように代数仕様記述言語 CafeOBJ によって記述したかを説明している。

第 6 章では、まとめと今後の課題、そして、バイトコード検証系を CafeOBJ により記述したものを付録としている。