3T-06 オブジェクト指向分散環境 OZ におけるセキュアプログラミング

西岡 利博　　塚本 享治
三菱総合研究所　　電子技術総合研究所

1 はじめに

オブジェクト指向分散環境 OZ は、未知のクラスのオブジェクトでも、ネットワーク上でやりとりして動作させられることが特長である。このようなシステムでは、ネットワークから輸入したオブジェクトによって、ホストシステムがダメージを受けないことと同時に、既存のオブジェクトにも適切な保護を与えることが重要である。OZ 上で採用されているセキュリティモデル[1]は、リモートメソッド起動 (RMI) におけるユーザ認証に基づくアクセス制御と、それを通して輸入されたオブジェクトから既存のオブジェクトを保護する機構とによって成り立っている。本稿では、このセキュリティモデル上で安全なプログラミングを行うためのプログラミングモデルについて述べる。

2 OZ のオブジェクトモデル

OZ では、オブジェクトは “セル” と呼ばれる単位で管理される。セルとは、ひとつの “グローバルオブジェクト” (GO) と0 個以上の “ローカルオブジェクト” (LO) からなるオブジェクトの集合である (1)。セルの外部からメソッドされ得るのは GO だけである。セルは

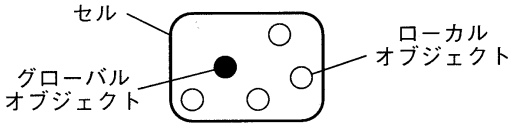

図 1: セル

分散の単位である。RMI の引数や返り値でオブジェクトを渡す場合、それらはディープコピーされるので、ひとつのオブジェクトが複数のセルに属すことはない。

3 OZ のセキュリティモデル

3.1 セル間セキュリティ

OZ では、すべてのセルにはそのオーナがおり、オーナの “ユーザ識別子” を取得する手段がある。セルの行う挙動は、そのオーナであるユーザの挙動と解釈する。

The secure programming model of OZ: an Object-oriented Distributed Systems Environment
Toshihiro Nishioka (Mitsubishi Research Institute, Inc.)
and Michiharu Tsukamoto (Electrotechnical Laboratory)

あるセルのすべてのメソッドを誰もが起動できると、セキュリティ上の問題を生じてしまう。そこで、そのメソッドを呼び出したユーザが誰であるかを特定し、そのユーザがそのメソッドを起動してよいかどうかを判断する必要がある。この、ユーザを特定する手続きを “認証” と呼ぶ。認証は OZ の実行機構によって実現される。

セルのプログラムは、認証によって得られた caller のユーザ識別子を取り出すことができ、それがそのメソッドの起動を許可されているユーザでない場合には、適切な例外を生じるなどして起動を拒否することができる。この、ユーザ識別子の検査は、一般的には、自身のオーナとの比較や、あらかじめ設定されたユーザ識別子の集合に含まれているかどうかの検査などである。この種のプログラミングの支援のために、ACL (Access Control List) ライブラリが提供される。

3.2 セル内セキュリティ

RMI によってセルの外部からもたらされたオブジェクトは、未知のクラスのインスタンスであるかもしれず、どのような挙動を示すか予測できない。しかし、セルの中にまで、セル間セキュリティのような認証に基づくアクセス制御を持ち込むと、効率上の問題がある。そこで、以下のような解決方法を提案した[2]。

危険なオブジェクトから重要なオブジェクトに対するメソッド起動を許さないために、オブジェクトを色分けする。最初からセル内に存在するオブジェクトは、安全であることを示すために、緑に塗る。緑オブジェクトが生成するオブジェクトは緑オブジェクトである。一方、RMI によってセルの外部からもたらされるオブジェクトは、赤く塗る。赤オブジェクトが生成するオブジェクトもまた赤オブジェクトである。

このように区別し、赤オブジェクトは緑オブジェクトのメソッドを起動できないものとすれば、重要なオブジェクトを保護できる。しかし、 OZ を含めて、一般のオブジェクト指向言語では、callee を特定することはできても caller を特定することはできない。そこで、OZ ではスレッドも色分けした。RMI によって発生したスレッドは、最初は緑であり、赤オブジェクトのメソッドを起動するところで、caller 側がスレッドを赤く塗り替える。スレッドの色を元に戻せるのは緑オブジェクトだけである。こうすることにより、緑オブジェクトは、自分のメソッドを起動したスレッドの色を確認すること

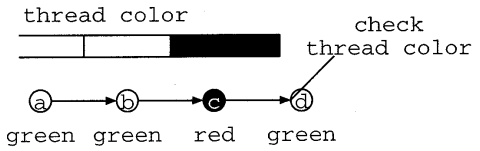

図 2: スレッドの色による危険なオブジェクトの識別

表 1: セル内外でのセキュリティ方針の差異

	セル間	セル内
行為者の識別	ユーザ認証	赤/緑
アクセス制御	プログラムで制御	メソッド起動不可

によって、自分を呼び出しているオブジェクトの色を確認できる。図2の例では、緑オブジェクト d は、緑オブジェクト b がスレッドの色を赤に変更してから赤オブジェクト c を呼び出しているので、スレッドの色を判定することによって赤オブジェクトから呼び出されていることが分かり、メソッド起動は拒否される。

オブジェクトやスレッドの色分けと、それに基づくアクセス制限は、エグゼキュータとコンパイラによって実現され、OZ のプログラマは記述しない。

表1に、セル間セキュリティとセル内セキュリティの差異をまとめた。

4 OZ のセキュリティプログラミングモデル

以上のセキュリティ上の特徴を持つ OZ では、一般にセルを次のように設計する。

4.1 メソッドの設計

セルのインタフェースである GO のメソッドは、次の二種類に分けて設計する。

特定のユーザのみが起動できるメソッド

必ずメソッドの先頭で起動者を確認する。

オーナだけが起動できるメソッドでは、その引数のオブジェクトの色を赤のままにしておくのか、緑に変更するのかを設計する。同様に、同じオーナの他のセルに対する RMI では、その帰り値のオブジェクトの色を変更するかどうかを設計する。

不特定多数が起動できるメソッド

その呼ばれ方によらず、セル自身の一貫性やサービス提供能力にダメージを与えられないようにする。

4.2 オブジェクトの設計

セル内のすべてのオブジェクトは、それが赤オブジェクトの可能性があるかどうかを明らかにしつつ設計しなければならない。赤かもしれないオブジェクトに対しては、以下のように設計する。

赤オブジェクトに対するメソッド起動は終了しないかもしれないのでタイムアウトをかける。

赤オブジェクトへのメソッド起動の引数としてオブジェクトを渡す場合は、通常は赤オブジェクトにアクセスされる必要があるので、破壊されても困らないオブジェクトに必要な情報を移し、赤オブジェクトに変更してから渡す。

4.3 セル構造の設計

赤オブジェクトが赤オブジェクトからメソッド起動される場合は、緑オブジェクトに対するような保護機構は働かないので、原則としてセルの内部で赤オブジェクトどうしが接触しないように設計する。

接触が必要な場合は、それらのオブジェクトの抽象インタフェースが自律性を持つように設計し、他の赤オブジェクトからのメソッド起動によって一貫性を欠いた状態に陥らない実装を可能とするようにする。

5 まとめ

未知のクラスのオブジェクトを受信して動作させるシステムでは、セキュリティ上の問題が生じるおそれがある。OZ では、RMI の際のユーザ認証に基づくアクセス制御と、RMI によって輸入したオブジェクトを識別する機能を提供することでこの問題を解決しようとしているが、本稿では、その際に必要なセル設計の指針を述べた。

本研究は、情報処理振興事業協会 (IPA) の「開放型基盤ソフトウェア研究開発評価事業」の一環として行われたものである。

参考文献

[1] 西岡, 塚本: “オブジェクト指向分散環境 OZ のセキュリティモデル”, SWoPP ’97, Aug. 1997.

[2] 濱崎, 西岡, 塚本: “オブジェクト指向分散環境 OZ のプロセス内セキュリティ”, 情報処理学会 第53回 全国大会, 1K-7, pp. 297–298, Mar. 1997.

[3] 西岡, 塚本: “オブジェクト交換を利用した分散サービス利用のためのフレームワーク”, 情報処理学会 研究報告 “システムソフトウェアとオペレーティングシステム”, Vol. 96, SWoPP 秋田 ’96, Aug. 1996.