

移動・分散プログラミング言語 Mobidget: 言語仕様

5U-1

藤田 悟*, Suresh Jagannathan†, Richard Kelsey†,
 小山和也*, James Philbin†, 山之内徹*

*NEC C&Cメディア研究所, †NEC Research Institute Inc.

e-mail: {satoru,kazuya,yamanouchi}@ccm.cl.nec.co.jp,
 {suresh,kelsey,philbin}@research.nj.nec.com

1 はじめに

Mobidget は、異機種分散コンピューティングに適したプログラム言語の作成を目的として研究を進めてきた。特に、通信帯域の狭い無線を用いたモバイル端末によるシステムから、高速 LAN のように広い通信帯域を持つシステムに至るまでの多様なネットワークに対応できるプログラム言語を提供することを主目的とし、移動と分散の両特性を持つ言語であることを特長とする。本稿では、Mobidget の概要として、基本モデルを中心に紹介する。

2 基本モデル

Mobidget の扱う世界は、ベースとエージェント、オブジェクトを中心に構成される。ベースは計算機上のローカルアドレス空間と計算資源を提供する。通常、ネットワークに結合した計算機上に複数のベースが提供されて、この複数ベースが提供するアドレス空間と計算資源が Mobidget の対象とする場となる。

エージェントは、上記の複数ベースからなる場の中に、独自のプロテクションドメインを形成する単位である。具体的には、独自のオブジェクト空間とスレッド、クラス管理機能等を持つ。エージェントとベースは多対多の関係にある。すなわち、ひとつのエージェントは複数ベースに分散して存在することができる。この時、各ベースに分散した部分エージェントをサブエージェントと呼ぶ。一方、ひとつのベース上には同時に複数のエージェント、あるいはサブエージェントが存在できる。言い方を変えると、各ベース上にはベース内に閉じた複数のサブエージェントが存在していて、ベース間にまたがった複数のサブエージェントを取りまとめ、ひとつのオブジェクト空間を管理する単位としてエージェントが存在するとも考えても良い。

オブジェクトは、エージェントの持つオブジェクト空間中に定義される。ひとつのエージェントが管理するオブジェクト空間内であれば、オブジェクトの存在するベースがローカルであるか、リモートであるかに関わらず、プログラム上から透明にアクセスできることが保証されている。一方、同じベース上にあるオブジェクトであっ

ても、異なるエージェントに管理されるオブジェクトであれば、エージェント間の通信プロトコルに従って、アクセスが制限される。すなわち、Remote インタフェースを実装したオブジェクトだけがエージェント間にインタフェースを公開し、他のエージェントが参照オブジェクトを保有したり、その参照オブジェクトに対するメソッドコールを行う許可を与える。

特別なクラスとして、エージェントとベースを抽象化するクラスが用意されている。プログラマは、後に述べるエージェントやオブジェクトを移動するプログラムを作成する時に、エージェントインスタンスへのメソッドコール、ベース指定のリモートメソッドコールなどを記述することができる。

典型的なベースとエージェント、オブジェクトの関係を図1に示す。ここでは、網掛けの四角は Remote インタフェースを実装したオブジェクトを表し、黒く塗りつぶされた四角は、Remote を実装していないエージェント内オブジェクトである。この図で、右側の計算機には、複数のベースが動いており、また、左上のエージェントは、複数のベース上に存在している。エージェント内であれば、ベース間の通信であっても、矢印(1)(2)のように、常にリモート参照として解決されることが保証され、値の一貫性を保つことができる。一方、エージェント間の通信は、Remote インタフェースが実装されていれば、リモート参照で、そうでなければオブジェクトのコピーで達成される。矢印(3)は、Remote を実装したオブジェクトへの参照であるので正当であるが、矢印(4)は、Remote を実装しないオブジェクトへの参照であり、実際には存在しない不当なアクセスである。

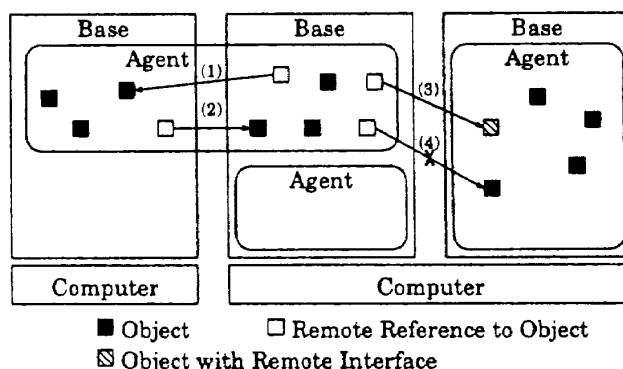


図1: Mobidget の基本構成要素

Mobidget: Mobile and distributed programming language
 Satoru Fujita*, Suresh Jagannathan†, Richard Kelsey†,

Kazuya Koyama*, James Philbin†, Toru Yamanouchi*

*NEC Corporation, †NEC Research Institute Inc.

3 移動・分散機能

Mobidiget の特徴は、プログラムの移動と分散にある。プログラムは、(サブ)エージェントとオブジェクトの移動命令を記述できる。以下に、移動と分散に関連した事項についてまとめる。

3.1 アンカードオブジェクトとピン属性

I/O ポートや外部の既存ソフトウェアへのアクセスのような計算機プロセス依存の実装をする場合、*Anchored* インタフェースを用いる。これを実装したクラスのインスタンスは、エージェントが移動する時にも移動せずに、元の場所に残り、エージェントからはリモート参照される。同様に、オブジェクトのピン属性は、一時的な移動不可を表す属性である。

3.2 エージェント移動

エージェント移動命令によって、原則として全てのオブジェクトやスレッドを移動することができる。例外は、前述のアンカードオブジェクトやピン属性がセットされたオブジェクトであり、移動の対象にならない。移動対象のエージェントから外部のエージェント内のオブジェクトへの参照関係、あるいは、外部のエージェントから移動対象のエージェント内のオブジェクトへの参照関係は、移動の前後で同様に保たれるように設計されている。

3.3 サブエージェント移動

上述のエージェント移動は、エージェント全てを目的地へと移動するのに対して、サブエージェント移動では、エージェントのうち、あるベースに存在するオブジェクトやスレッドだけを移動することができる。例えば、機能分散を意図して構築された分散型エージェントが、その分散性を崩すことなく、エージェントの一部だけを別のベースに移動するとき有効である。既に、サブエージェントが存在するベースに別のサブエージェントが移動した場合、2つのサブエージェントをマージして、ひとつのサブエージェントになる。

3.4 オブジェクト移動

オブジェクト移動命令によって、オブジェクト単位で他のベースに移動することができる。移動先に同一エージェントのサブエージェントが存在する場合には、オブジェクトは、そのサブエージェント内に移動する。サブエージェントが存在しない場合は、新たにサブエージェントを作成して、その中にオブジェクトを移動する。

3.5 リモートオブジェクト生成

ベースを指定したオブジェクト生成によって、リモートオブジェクトを生成できる。この場合、オブジェクト移動と同様に、指定先のベースに既に存在していたサブエージェント、あるいは、新たに生成したサブエージェントの中に新しいオブジェクトを生成する。

3.6 メソッドコール

Mobidiget は、プログラマに対して、以下の2つのメソッドコールモデルを提供している。

RPC モデル (remote procedure call model) では、メソッドを、オブジェクトの存在するベース上で実行する。この結果、自オブジェクトのフィールドアクセスが生じた時に、オブジェクトの位置は必ずローカルなので、高速なアクセスを行える。

インボーカーモデル (呼び出し側実行モデル) では、メソッドの呼び出し元のベースで、メソッド実行を行い、オブジェクトの存在するベースであることは保証されない。スレッドがベース間を頻りにジャンプしないというメリットを有する。

以上の2つのモデルを用いて、実行速度と分散性のトレードオフを加味したプログラミングが可能である。

また、インボーカーモデルのメソッドに '@[base]' を明示的に付けることで、実行ベースを指定することもできる。この場合、メソッドを呼び出し場所、実行が行われる場所、オブジェクトの存在場所が別々になることもあり得る。しかし、実行のセマンティクスに変化はない。

3.7 エージェント間通信

エージェント間通信には、*Remote* インタフェースを用い、リモートオブジェクトに対するインタフェースメソッドの実行を行う。まず、サーバ側プログラムは、上記のインタフェースを実装したクラスを定義し、メソッドを実装する。続いて、サーバエージェントを実行して、インスタンスを作成し、エージェント自身の持つレジストリ、または、ベースに用意されたレジストリエージェントにインスタンスを登録する。クライアントエージェントは、レジストリに必要なリモートオブジェクトの検索を依頼し、オブジェクトへの参照ポイントを得る。そして、クライアントエージェントは、そのリモートオブジェクト参照ポイントに対して、インスタンスメソッドコールを行い、エージェント間の通信を達成する。

引き数や戻り値にオブジェクトが指定された場合、それが *Remote* インタフェースを実装したオブジェクトであれば、オブジェクト参照ポイントを渡すことになる。一方、*Remote* インタフェースを実装していないオブジェクトが引き数等に現われた時は、オブジェクトの(深い)コピー渡しが行われる。

4 おわりに

以上、Mobidiget は、エージェントというプロテクションドメインを持ち、移動と分散に関する機能を合わせ持ったプログラミング言語である。実行中のスレッドを移動できること、分散機能が強化されていることなどで、既存の移動エージェント言語とは、異なる性格を持つ。今後、分散プログラムの記述容易化、分散オブジェクト配置による実行速度の向上などに対する効果について、評価を進める。