

## エージェント開発環境 ADEPT における通信基盤の試作

3 J-5

城島貴弘

朝倉敬喜

sirosima@hml.cl.nec.co.jp asakura@hml.cl.nec.co.jp

NEC ヒューマンメディア研究所

## 1. はじめに

現在、分散通信分野においては、オブジェクト指向を使用した通信技術として、OMG による CORBA や Microsoft による DCOM が提唱されている。

しかし、これらの技術を使用してエージェントを作成する場合、その規格の複雑さにより本来エージェント開発者にとって主要ではない通信部分の作成に多大な労力を割かざるを得ない。また、前記の技術を使用したエージェントを動作させるには、これら技術の動作環境を導入、設定しなければならず、その維持、管理を行う必要が生じる。この作業は、エージェントを普及させる上での障害となり、その普及を困難なものとする。

よって、エージェント開発者がより簡易に使用でき、動作環境の導入、維持のコストが少ないエージェント通信基盤を作成する必要がある。そこで我々は、現在制作中であるエージェント開発環境 ADEPT において、この諸問題を解決したエージェント通信基盤を試作した。本稿ではこの ADEPT 通信基盤において考慮した問題とその解決方法について述べる。

## 2. エージェント通信基盤の要件

我々は、エージェント通信基盤を作成する上で以下の点に留意した。

- 1) 位置独立性
- 2) 単独実行性

位置独立性とは、エージェント間でメッセージを送受信する際、その宛先指定にホスト名を含めないということである。

宛先指定にエージェントが動作しているホスト名が含まれていると、エージェントが他のホスト上に移動した場合、エージェントの参照名が変わってしまい、他のエージェントからは新しいエージェントが出現したかのように見えてしまう。これはモバイルエージェントなど、ホスト間を移動するエージェントを考慮した際に重大な欠点となる。

単独実行性とは、エージェントが通信を行う際、エージェント本体以外に通信用のモジュールを必要としないということである。

これは、1章で述べたように、通信に別のモジュールを利用すると、そのモジュールの導入、管理が必要となり、エージェントという柔軟なソフトウェアには適さない。通信基盤に単独実行性を持たせることにより、エージェントの運用を柔軟かつ容易に行うことができ、エージェント

を導入する際の手間も減少する。

以下、上記の性質を持たせるため、ADEPT 通信基盤で用いた手法について説明する。

## 3. ADEPT 通信基盤による実現

## 3.1 位置独立性の実現手法

ADEPT 通信基盤では、位置独立性を持たせるため、通信の際の宛先指定にエージェントの動作するホスト名を含まず、任意の識別子でエージェントを参照できるようにしている。このため、指定された識別子から宛先のエージェントが動作しているホストを検索するための変換テーブル(エージェント管理表)を保持する。この変換テーブルは、エージェントの識別子と実際のネットワーク上の資源(ホストの IP アドレス、ポート番号など)との対照表であり、通信の際に指定された宛先エージェントの識別子をネットワーク上の資源に変換するために使用される。

## 3.2 単独実行性の実現手法

前述の識別子をネットワーク資源に変換する方法は、DNS などでも使用されている一般的な手法である。しかし、従来の手法では、DNS サーバといったサーバを別に用意する必要があり、単独実行性を持たせることができない。

そこで、ADEPT 通信基盤では、先に述べたエージェント管理表を全てのエージェントに持たせることにより、サーバの必要性をなくしている。しかし、これにより以下の問題が生じる。

1. どのようにしてエージェント管理表を取得するか
2. ネットワーク上にエージェントが存在しない場合、どのようにエージェント管理表を作成するか

以下、それぞれを解決するため、ADEPT 通信基盤で用いている手法について説明する。

## A. エージェント管理表の取得手順

各エージェントは以下の新規登録作業を行うことにより、エージェント管理表を得る。

- i) 他のエージェントに対して、登録要求メッセージをブロードキャストで送信
- ii) 他のエージェントから登録完了メッセージをブロードキャストにて受信
- iii) 他のエージェントに対して、エージェント管理表要求メッセージをブロードキャストで送信
- iv) 他のエージェントから、現在のエージェント管理表をブロードキャストにて受信

ネットワーク上の他のエージェントは、i)で送信された登録要求メッセージにより、新たに起動したエージェントをエージェント管理表に登録し、iii)により、エージェン

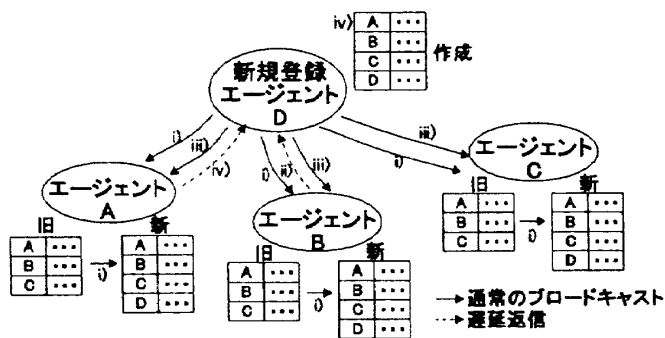


図1 新規エージェントの登録

ト全体で共通なエージェント管理表を返信する。(図1参照)

この時既存のエージェントは、i)及び iv)で新規登録中のエージェントに対して返信を行うが、返信メッセージの氾濫を防ぐため、IP Multicast における IGMP<sup>[1]</sup>と同様の処理を行う。これはメッセージを返信する際、乱数で決定された時間分だけ返信を遅らせるという方法である。返信処理を遅らせている間に、他のエージェントがブロードキャストにより同一のメッセージを返信すると、返信処理を中止する。また、遅延期間が過ぎた場合は、自らブロードキャストにて返信を行う(以下、この返信方法を遅延返信と呼ぶ)。

以上の手法を用いて、ネットワーク上の全てのエージェントで共通のエージェント管理表を管理できる。

B. 初期状態におけるエージェントの起動

ネットワーク上で最初に起動したエージェントは、i)及び iii)の処理を行っても返信が返されないため、i)において、ある一定期間の間(タイムアウト期間)返信を受信できなかった場合、ネットワーク上に他のエージェントが存在していないとして、自分だけが登録されたエージェント管理表を作成する。以下、このようなエージェントを初期エージェントと呼ぶ。

C. エージェント起動時における問題の解決

A 節及び B 節で述べた手法により、単独実行性を実現することができる。しかし、単にこの手法を実装しただけでは、エージェントの起動時に問題が発生する。本節ではこの問題について述べ、解決方法を示す。

a) 仮エージェント管理表の作成

ネットワーク上にエージェントが存在しないとき、複数のエージェントがほぼ同時に起動した場合、お互いがi)の返信待ちを行い、タイムアウト期間を過ぎて、両方とも自分自身しか登録されていないエージェント管理表を作成してしまう。

そこで各エージェントは、起動と同時に仮のエージェント管理表を作成し、iv)に至る前でもこの仮管理表を使用して登録作業を行えるようにする(図2)。iv)において、正規の管理表を受信した時、仮管理表は破棄される(図2-1)。また、初期エージェントの場合には、タイムアウト終了後、仮管理表を正規のエージェント管理表とすることによりエージェント管理表を得る(図2-2)。

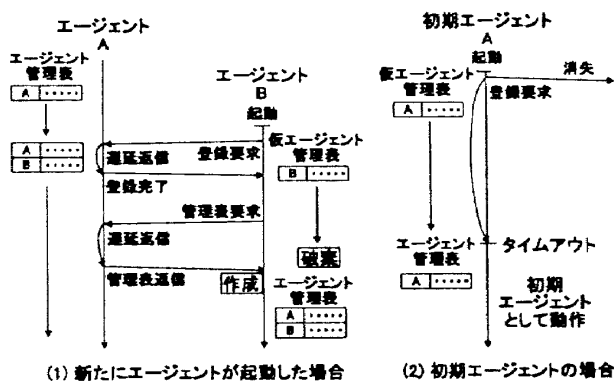


図2 仮エージェント管理表の使用

b) 待ち時間を含んだ登録完了メッセージの返信

初期エージェントが、仮管理表を使用して、登録完了メッセージを返信した場合、この登録完了メッセージを受信した新規登録エージェントは、すぐに iii)の処理に移ろうとする。しかし、初期エージェントは、i)のタイムアウト期間中のため、iv)の処理を行うことができない。そこで初期エージェントは、ii)で返信する登録完了メッセージ中に、自身のタイムアウトするまでの時間を含め、登録中のエージェントを初期エージェントがタイムアウトするまで待たせるようにする。

c) 遅延返信時における遅延時間の延長

すでに起動が完了したエージェント  $\alpha$  が存在するとき、二つ以上のエージェント(エージェントB,C)が同時に起動したとする。ここでBは、i)において登録要求メッセージを送信するが、このメッセージは、 $\alpha$ とCの両方で受信、処理される。この時、CもB同様登録処理中であるが、前記 a)に従い、遅延返信にて返信処理を行う。

ここで遅延処理における遅延時間の最大値を  $D_{MAX}$  とし、 $\alpha$ の遅延時間を  $D_\alpha$ 、Cの遅延時間を  $D_c$  ( $D_\alpha, D_c < D_{MAX}$ ) とすると  $D_\alpha < D_c$  の時には、 $\alpha$ が登録完了メッセージを返信するため、Bはすぐに iii)の処理に移れる。しかし、 $D_\alpha > D_c$  の場合、 $\alpha$ よりも先にCが前記 b)に従って、Cのタイムアウト時間が入った登録完了メッセージを返信し、Bは無駄なタイムアウト待ちを行う。

そこで、登録中のエージェントが返信する際には、通常の遅延時間に  $D_{MAX}$  を加える ( $D_c' = D_c + D_{MAX}$ )。これにより、 $D_\alpha < D_c'$  が保証でき、常に  $\alpha$  が優先的に返信することができる。

4. まとめ

エージェント間通信を実現する ADEPT 通信基盤において、1)位置独立性と 2)単独実行性の確保手段について説明した。現在 ADEPT 通信基盤は Java により実装され、前記の性質を満たすことにより、エージェント開発において柔軟性と利便性を実現している。

文献

[1] T. Maufer, C. Semeria: [Introduction to IP Multicast Routing], INTERNET-DRAFT, March 1997