

モバイルエージェントシステム AgentSphere の開発  
 —エージェント移動プロトコルの設計と実装—  
 Development of Strong Migration Mobile Agent System AgentSphere  
 - Design and Implementation of Agent Migration Protocol -

黒崎 信清<sup>†</sup> 甲斐 宗徳<sup>‡</sup>  
 Nobukiyo Kurosaki Munenori Kai

## 1. はじめに

モバイルエージェントシステム AgentSphere[1][2]では、一般的に普及している PC を利用して、簡単かつ低コストで並列分散処理を実現できるようにするため、エージェントと呼ばれるプログラムが自律的に AgentSphere と呼ばれるエージェントの活動領域間を移動し処理を行う。このシステムによって、インフラとしてのネットワークを準備するだけで、並列分散処理の知識を持たない人でも、簡単な操作・コーディングで並列分散処理を行って、高い処理能力と信頼性を持ったサービスを受けられるようにすることを目標としている。

現在の AgentSphere ではエージェントは通信先の相手に移動する処理をすると、相手に一方的に送って相手を受け取ったかどうかの確認をせずに移動処理を終了してしまう。もしも、あるエージェントが移動に失敗したらそのエージェントは途中で処理を終えてしまい、与えられた処理が終わらない可能性がある。そこで、エージェントが実際に移動する前に確認用通信をすることで事前に移動先へ移動できるかどうかの確認を取るとともに、実際にエージェントが移動した後に終了通知を出させることで移動が確実に完了したかどうかという問題に対応した。本研究ではエージェントが通信先へ移動するための手順を明確にし、その方法に従って通信を行うことによってエージェントの移動が正常に行われたことの保証を得ることを目的としている。

## 2. エージェント移動プロトコルの開発

### 2.1 エージェント移動プロトコルについて

通信プロトコルでは決めておかなければならない項目として次のようなものがあげられる。

- ・通信で送受信されたデータの形式。
  - ・送り側がどのようにして通信の開始を知らせるのか。
  - ・送信された情報が破損していた場合は再送信を要求するのか、そのまましておくのか。
- 以上の項目を踏まえてエージェントの移動を保証する通信プロトコルを定めることにした。ここではプロトコルの全体像を示す。
- 1: エージェントが移動命令を出す。
  - 2: ソケット通信を用いて送信側が受信側に通信要求を出す。
  - 3: 受信側が通信要求を受信する。
  - 4: 送信されてくるエージェントを収容できるかどうかの判定を行う。
  - 5: エージェントを収容可能なら送られてくるエージェントの情報を登録する。
  - 6: 受信側が送信側に応答を返す。
  - 7: 送信側が応答を受信する。

<sup>†</sup> 成蹊大学理工学研究科理工学専攻 Graduate School of Science and Technology, Seikei University

- 8: 送信側が受信側にエージェントを移動させる。
- 9: 受信側は送られてきたエージェントが事前に登録されていたエージェントか確認し、確認できたらエージェントの処理を再開させる。
- 10: 受信側がエージェントを受信完了したことを送信側に伝える。

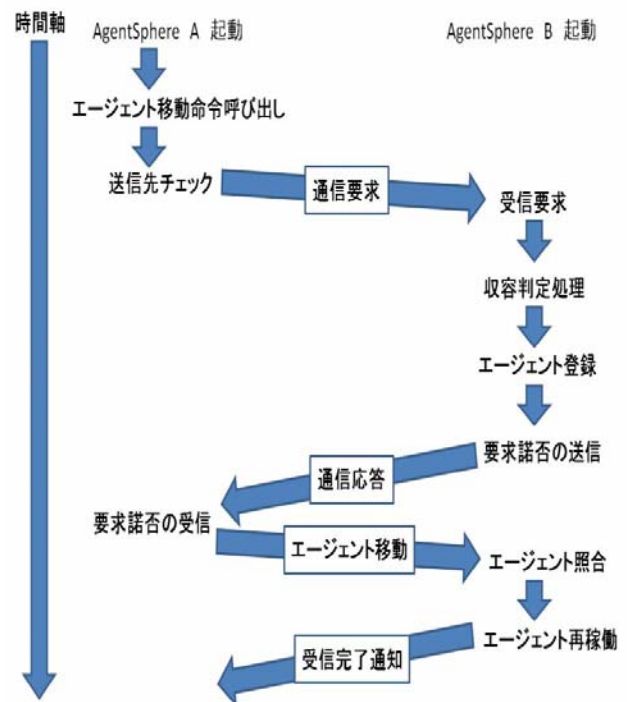


図1 エージェント移動プロトコル

### 2.2 確認用通信

現在、AgentSphere ではエージェントが移動すると移動先ですぐに再稼働するようになっている。移動先にエージェントが集中していても関係なく再稼働してしまうのは問題があるので、実際にエージェントを移動させる前に確認用の通信をするようにした。

#### 2.2.1 送信・受信スレッド

エージェントを移動させる前に AgentSphere 間で通信を行うために、専用の送信スレッドと受信スレッドをそれぞれ実装した。送信スレッドと受信スレッドはそれぞれ非同期で処理を行う。エージェントが移動命令を呼び出すと送信スレッドの中のキューに通信要求がタスクとして並べられる。キューに並べられる通信要求の内容は移動命令を出したエージェントの ID と送られてきたデータをどのように処理するか指定する通知語（文字列）で構成されている。送信スレッドはキューにタスクがあると宛先の AgentSphere の受信スレッドに向けて通信要求をソケット

通信で送信する。一方、受信スレッドは送られてくる通信要求から通知語を読み取り、それに応じた処理を行った後、送信スレッドのキューに通信応答をタスクとして並べる。

### 2.2.2 処理を指定する通知語

これらの通知語は図 1 のエージェント移動以外の通信の際に送られるデータである。

処理を指定する通知語には応答を返すように指示する「reply」と移動許可を知らせる「access」、通信の失敗を知らせ、再送処理をする「again」、エージェントの受信完了を知らせる「finish」の 4 種類がある。通信の流れを具体的に説明すると、エージェントが移動命令を呼び出したら移動処理を行う前に相手に送りたいエージェントの ID と「reply」を送り受信要請を伝える。送り先ではこの通知語を受け取ると通信要請の内容を調べ、判定処理を行う。判定をクリアしたら、送られてくるエージェントの予約を登録し、通知語を「access」に変更して送り元に応答を返す。この応答を受信すると送り元でエージェントの移動処理を行う。エージェントが送り先に移動すると送り先ではそれが予約登録されたエージェントであることを確認し、「finish」を持たせた通知語を送り元に送ることで通信終了を伝える。

### 2.3 エージェントを収容できるかの判定処理

スケジューラなどによって最適な初期分散がなされたとしても、エージェントは自律的に移動するものなので AgentSphere 内の状況は常に変化していく。なにもせずにエージェントを移動させてしまうと JVM のメモリ不足によって JVM がダウンしてしまったり、エージェントが何か所の AgentSphere に集中してしまうという問題がある。

そのため、エージェントを移動させる前の通信で判定処理を行い、その判定をクリアしたエージェントのみが指定した AgentSphere へ移動できるように設計した。現状 AgentSphere 上で動かしているどのエージェントも簡単な仕事を行うだけの小規模なものなので、現段階での判定方法は判定時の JVM でのメモリ使用率を測定して、その値が一定の値を超えていないかどうかによってエージェントの収容が可能か不可能かを決めている。

この判定処理をクリアしたエージェントは指定した AgentSphere へ移動できるが、判定をクリアできなかった場合は指定先を変更するか今いる場所にとどまって処理を続けるか選んで処理を実行する。

今後、様々なエージェントが実行されるようになった場合、エージェントがそれぞれどれだけメモリを使用するか調べ、移動先の AgentSphere で収容しても使えるメモリ量をオーバーしないか確認する判定処理に変更する必要があると考えられる。

### 2.4 エージェントの受信確認

エージェントの移動を保証するにあたって、対象となるエージェントが移動先に確実に目的地に着いたかどうかの確認が必要である。そこで判定結果が収容可能だった場合受信したエージェント ID をリストに登録する。これによってエージェントがこれから移動してくることを予約することができる。そしてエージェントが他の AgentSphere から移動してくるとエージェントの受信を検出し、登録リストから予約されたエージェントを探して、見つかったらそ

の ID をリストから削除し、エージェントの移動完了を伝える。また、エージェント ID は AgentSphere で形成されたネットワークにおいて唯一にするために、UUID を使用している。

### 2.5 再送処理

送信スレッドが通信要求を出す際に通信に失敗した場合、改めて要求を送りなおす必要がある。通信失敗が考えられるのは通信のタイムアウトが発生した時や、データの読み込みに失敗してしまった場合があげられる。タイムアウト問題の解決策として、送信スレッドが通信要求の送信をしてから要求が返ってくるまでの許容時間を設定して、その時間を過ぎたら再度要求を出すことで問題の解決を試みた。また、受信したデータが何らかの原因で読み込めなかった場合は通信要求元の AgentSphere に再度通信要求を出してもらうことで解決を試みた。決まった回数再送処理しても応答が来ない場合は現在通信しているマシンから別のマシンに宛先を変更して送信させる。

### 2.6 受信完了通知

エージェントの受信が完了してエージェントが再稼働を始めた時点で送り元にエージェントが正常に受け入れられたことを通信することで、送り元はエージェントの移動が正常に行われたことを認識する。

## 3. おわりに

今回定めたプロトコルに従ってエージェントの移動が行われたことによって、エージェントの移動が正常に行われるという保証を得ることができるので、エージェントが移動したはずなのに実は移動に失敗していてエージェントが消滅していたという事態を防ぐことができるようになった。また、確認通信に失敗した際の再送処理を追加したことで、実際に移動する前に移動できないことを検出できるようになった。

今後の課題として、収容判定処理の改善があげられる。現段階での収容判定処理はどのエージェントも同じ大きさであり大きくないものであるという前提条件の上でのものなので、様々なエージェントに対応できるように改善すべきであると考えている。

### 謝辞

本研究は科研費（基盤研究(C)21500041）の助成を受けたものであることをここに記し、謝意を表します。

### 参考文献

- [1]赤井雄樹・横内 貴・若尾一晃・甲斐宗徳「強マイグレーションモバイルエージェントシステム AgentSphere の開発」,FIT2009, B-011, Sep.2009
- [2]鈴木幸祐・山口大祐・甲斐宗徳「モバイルエージェントシステム AgentSphere における強マイグレーション機構の改良」,FIT2011, B-030, Sep.2011