3J－6

ネットワーク上の多様な動作環境において自律的に移動できるMobile Agentの研究*

河原 正文†　砂原 秀樹†　湊 小太郎†
† 奈良先端科学技術大学院大学

1 はじめに

ノートパソコンなどの携帯端末の普及によって、高信頼でトラフィックを下げることなくネットワークにアクセスできるアプリケーションが要求されるようになってきた。その中で、Mobile Agentは、ソフトウェアオブジェクトがデータとともに、自律的に移動して情報収集を行う新しいソフトウェア技術として研究されるようになってきた。しかし、Mobile Agentがネットワークを移動するためには、各Agent専用のPlatformが移動先のホストに事前実装されていることが前提であり、従来の研究では、この制約の範囲内で動作するAgentの機能と移動方法の研究が中心であった。[1]

そこで、本研究ではインターネット上の任意のホストに、Mobile Agentの開発言語であるJava環境とともに、無理なくAgent Platformを実装するための手法の構築と、ホスト間をMobile Agentが自律的に移動するための移動先情報をネットワーク内に構築するアルゴリズムの設計を行なった。

2 Agent環境の実装方法

本研究では、Mobile Agentの開発言語として、Java(JDK1.1)を使用した。その理由は以下の通りである。

- Java Virtual Machine(JVM)によってホスト環境に依存しないプログラム開発が可能。
- RMI(遠隔メソッド呼び出し）など、Mobile Agentの仕様に適した機能がAPI化。
- インターネットに接続されている多くのホストマシンに実装されてきている。

そこで、Javaで新規に開発したAgent Platformを、JDKのインストール時、または更新時に、同時にホストに実装する手法を以下のように開発した。

- 次の仕様をもつインストール用プログラムの実行
  (1) Java環境とAgent Platformを同時に圧縮した新規のJDKファイル（例：jdk1.1.3new-solaris-sparc.bin)を解凍する。
  (2) JVM、Agent Platformなどの環境がホスト内に所定の構造で実装される。
  (3) 次の仕様のMobile Agent環境ブートプログラム(agentboot)をインストーラより起動する。
  - RMI registryの動作を開始（rmiregistry)。
  - Agent Platformのバックグラウンド動作開始。
  - Resident Agentのバックグラウンド動作開始。
- システムの再起動時に対応するため、OSの起動ファイルへagentbootを追加記述する。

3 Mobile Agentの構造

Mobile Agentの移動は、JavaのAPIにより機能提供されているRMIとObject Serializationによって実行している。動作の仕組みを図1に示した。前章の実装方法によって、インターネットに接続されているホストを、JVMとともに同一仕様のAgent Platformが動作する状態に置くことができる。Agent Platformが起動されたとき、すでにバックグラウンドのスレッドで動作しているrmiregistryは、Agent Platformオブジェクトがコールする Naming.rebind() メソッドによって、オブジェクト名:AgentPlatformでホストのNaming Registryに登録する。Mobile Agentは、移動先のurl(例：rmi://DestinationHost/AgentPlatform)を使い、Naming.lookup(url)メソッドによって、インターネット上の該当ホスト:DestinationHostと、その中のオブジェクト:Agent Platformを捜し出して呼び出しを行なう。呼び出しに対してオブジェクト参照が戻り値として返却され、RMIの経路ができる。この経路を通って、Mobile Agentオブジェクトは、Object Serialization機能によるバイト列へ自動変換され、Object Streamとして移動をする。移動後のMobile Agentは、移動先のAgent Platformによって管理される。[2][3][5][6]

4 Resident/Mobile Agentの協同による自律移動システム

JVMとAgent Platformのインストールと同時に、実装先のホスト名を、インターネット上にある特定の

*The Research of Autonomous Mobile Agent Migrating in the Heterogeneous Network Environment
†Nara Institute of Science and Technology,
Masafumi Kawahara, Hideki Sunahara, Kotaro Minato

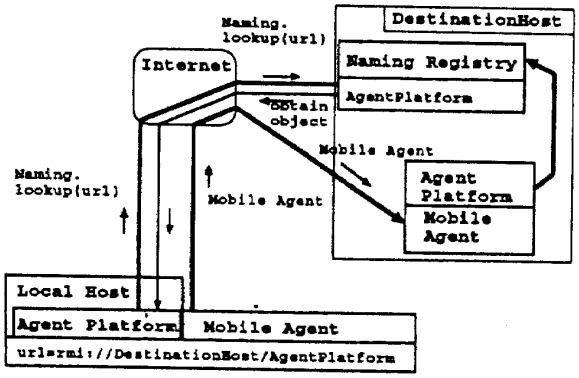

図 1: Mobile Agent の移動の仕組み

ホスト (ホスト名:SourceHost) に対して、RMI によって送信する機能をもつ Resident Agent を実装する。

ローカルホスト上の Resident Agent は、Agent Platform と同様にバックグラウンド動作を行ない、定期的に SourceHost 上の Resident Agent に対して、Naming.lookup(url) メソッドを実行して呼び出しを行なう。SourceHost は特定のホストであるので、url=rmi://SourceHost/ResidentAgent のようにホスト名とオブジェクト名を定数化できる。SourceHost との RMI 経路ができると、ホスト名を送付する。ホスト名は、InetAddress オブジェクトのもつ getLocalHost()、getHostName() メソッドによって入手する。SourceHost 側では、Agent Platform が登録されているホスト名を入手し、Hashtable 構造の移動先リストに更新登録する (図 2)。Hashtable の要素のホスト名 (key) と url(value) は、String オブジェクトとして登録している。[4][6]

key (host name)	value (url)
SourceHost	rmi://SourceHost/AgentPlatform
DestinationHost1	rmi://DestinationHost1/AgentPlatform
DestinationHost2	rmi://DestinationHost2/AgentPlatform
DestinationHost3	rmi://DestinationHost3/AgentPlatform
DestinationHost4	rmi://DestinationHost4/AgentPlatform

図 2: 移動先リスト (Hashtable 構造)

ローカルホストの Mobile Agent は、Resident Agent によって作成された移動先リストを、SourceHost から入手し、オブジェクトとして内部にカプセル化して出発する。Mobile Agent は、移動先のホストにおいても、最新の移動先リストを SourceHost から入手できるので、移動経路の情報を動的に更新しながら、インターネット内での自律的移動を実現できる。(図 3)

5 実装と評価

Java の要素技術である、RMI、Object Serialization を応用して新規開発した Mobile Agent を実装し動作を確認した。また、Agent 環境の実装方法によって、ネットワークに接続されているホストへの JDK の実装または更新時に、Agent Platform 等の Agent 動作環境を同

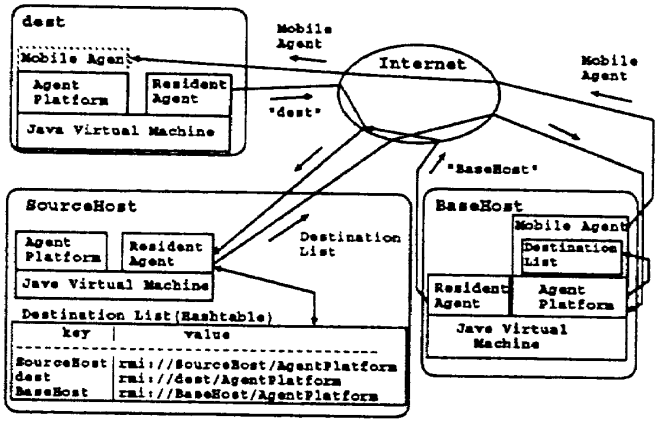

図 3: Resident/Mobile Agent 協同システム

時に実装し稼働させることができるようになった。これによって、インターネット上の任意のホストに、同一仕様の Mobile Agent の環境を実装する技術的手法を確立できた。さらに、「Resident/Mobile Agent 協同システム」のアルゴリズムの設計と基本機能の実装を行ない、ネットワーク内のホスト上に、RMI による Mobile Agent の移動先リスト (Hashtable) の登録と読み出し機能を実現した。これにより、インターネット上を自律的に移動できる Mobile Agent システムを開発できる環境を提示することができた。

6 まとめ

今後の課題として、「Resident/Mobile Agent 協同システム」のインターネット上での実装評価を続けて改良を行なうことが必要である。また、応用研究として、電子透かし情報をもつデータ探索の技術開発を行ない、ホームページなどでの個人データの不正使用を発見する機能をもつ Mobile Agent の開発が考えられる。

参考文献

[1] H.S.Nwana:"Software Agents: An Overview", The Kowledge Engineering Review Vol 11(3)(5/9/1996).

[2] Michael Morrison, et al: "Java 1.1 UNLEASHED, 3rd edition", Sams Net(1997).

[3] D.Groner, K.C.Hopson, H.Prabandham, T.Sundsted 著, スリーエーシステムズ訳 (武藤健志監修) : "Java API スーパーバイブル 1 (標準クラスライブラリ編) ", SE 社 (1997).

[4] N.Nagaratnam, B.Maso, A.Srinivasan 著, スリーエーシステムズ訳 (武藤健志監修) : "Java API スーパーバイブル 2 (ネットワーキング、AWT 編) ", SE 社 (1997).

[5] Scott Oaks, Henrry Wong 著, 戸松豊和監訳, 西村利浩訳 : "JAVA スレッドプログラミング", O'REILLY(1997).

[6] M. and C. Hughes, M. Shoffner, M. Winslow: "JAVA Network Programming", MANNING, Printice Hall (1997).