

Title	モバイルエージェントの動的部品交換方式の実験的検証
Author(s)	松原, 剛
Citation	
Issue Date	2000-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1325
Rights	
Description	Supervisor: 権藤 克彦, 情報科学研究科, 修士

モバイルエージェントの 動的部品交換方式の実験的検証

松原 剛

北陸先端科学技術大学院大学 情報科学研究科

2000 年 2 月 15 日

キーワード: モバイル, エージェント, 部品, 動的, 交換.

1 背景

本研究では MobileAgent が自分自身の保守を行える機構を提案及び実装する。

MobileAgent とはコンピュータ間を内部状態を持ったまま移動できるプログラムであり、比較的容易に分散アプリケーション組むことができるフレームワークを提供する。ここ数年、多くの MobileAgent システムが開発されてきた。

しかし、MobileAgent はネットワークを通じ、遠隔地のサーバへ移動して活動するという性質があるため、そのためオーナーがエージェントに直接手を加えることができない場所で活動することが多い。そのため、遠隔地で活動中のエージェントの保守が困難になり、様々な問題が起こる。

1. 例えば検索エンジンエージェントの例を考える。エージェントは各サーバを廻り歩き、HTML ファイルからデータベースを作り、オーナーに転送するといった機能を持つとする。ところが、まったく新しい Web サーバが生まれた場合、その新しいサーバには対応できなくなる可能性がある。通常のアプリケーションならば、オーナーがその場でアプリケーションに対しパッチを当てる等の保守を行えるが、モバイルエージェントの場合はオーナーが直接手を加えることができない場所で活動している場合があるので、保守が困難になる。
2. また、MobileAgent はネットワークに強く依存しネットワークコネクションを張るものも多い。例えばプロキシの動作を行うエージェントの場合、複数のクライアントから接続要求を受け、ネットワークコネクションを常に張っている状態になる場

合があり、その途中でアプリケーションを停止し、新しいプロトコルに対応させる保守を行うとネットワーク接続が切れてしまいサービスの質の低下に繋がるという問題が生まれた。

そこで、MobileAgent の保守を行う際に、オーナーの手を煩わせずに、また、ネットワークコネクションを切らずに行う要求が生まれた。

本研究では動的に自分の一部を組み変える保守を行い、上記の要求を満たす保守方式を提案する

近年の MobileAgent システムは Java ベースのものが主流になり、動的な保守が行い易い。Java ベースのものが主流になった理由の一つに Java にはクラスロード方式に柔軟性があることが挙げられる。ファイルのみならず、メモリ中のデータやネットワーク的に離れた場所にあるデータからクラスを生成することや、動的にクラスを定義することが可能である。この特徴を生かして MobileAgent の移動を実現しているシステムが多い。本研究でもこの動的クラスローディング機能を利用し、自己の一部を組替える機構を実現する。

2 目的

本研究では MobileAgent の一部をエージェント自体が動的に組替え保守を行う機構を提案及び実装する。

その際に特に、部品の配布方式に着目し、エージェントの移動性や独立性といった特色を生かした方法を考案する。また、本配布方式が有効であることを示すために、従来のアプリケーションに適用されている配布方法と比較検討する。

従来の部品配布方式には、エージェントに新しい部品を送りつける push 方式やエージェントに部品を取ってこさせる pull 方式がある。しかし、これらの方式にはネットワークコネクションの数が増えるといった問題や、部品を置いてあるサーバが非固定的な状態であった場合には配布が行われなくなるという問題がある。

そこで、本研究ではこれらの問題の影響をうけにくい方式を提案及び実装する。また、エージェントの動的部品交換に関しては実例に基づいた検証例が少ないので、実際に数種の例題について実装および検証し、どのような例に対してはどのような手法が効果的かという知見を得る。

3 比較交換方式

本研究ではエージェントの一部となる部品の配布方式に着目しており、エージェント同士がお互いの保持している部品を比較し、お互いに補完し合う方式を提案する。

- 1 移動後、同ランタイム上に存在するエージェントの部品リストを要求。同時に自分のリストも公開する。
- 2 自分の保持していない部品を持っている相手に部品の要求。
- 3 部品を受け取る。

図 1: 比較交換方式のシナリオ

4 実装

本研究では比較交換方式を使った動的保守のフレームワークを提案及び実装する。

また、その実装した環境上で実際に数種の例題を組み従来方式との比較を行う。

実験環境は佐藤一郎氏開発の AgentSpace を用いた。AgentSpace は Java1.1 ベースで実装されているため、本研究も必然と Java1.1 ベースの実行環境を必要とする。

Java にはクラスロード方式に柔軟性があり、比較的容易に組替え機能を実現できる。AgentSpace 自体もこの機能を使い実現されている。エージェントは AgentSpace の ClassLoader である AgentClassLoader を呼び出して行う。本研究では、この AgentClassLoader を拡張し、動的交換方式を実装した。

5 まとめ

本研究では、MobileAgent の自己管理能力を高める機構を実装し実験した。比較交換方式を使った保守は従来方式に比べ、以下の特徴を持つ。

- 1 ネットワークコネクションを張る必要性が少ない。
- 2 部品の置いてあるサーバの URL といった固定情報を持つ必要性が少ない。そのため、部品サーバとの接続が不安定になるネットワークポロジにも耐性を持つ。
- 3 ランタイムの数に対しエージェントの数が多い場合に効率的に部品の配布を行うことができる。