

移動エージェント相互運用の実現： 3S-3 インカネーションエージェントによる相互運用の実現

長 健太* 長谷川 哲夫* 大須賀 昭彦* 糸野 文洋** 中島 震***
株式会社東芝* 株式会社三菱総合研究所** NEC***

1 はじめに

移動エージェント基盤間の相互運用を実現することで、移動エージェントは、様々な移動エージェント基盤がカバーする、より広い範囲のネットワークを移動することが可能となる。より多くの情報やサービスにアクセスし、各移動エージェント基盤の持つ特性を利用することで、ネットワークを利用した高度なシステムを構築することができる。

しかし、現状では、それら移動エージェント基盤は、異なる実装方式、異なるエージェント表記法を用いており、あるエージェントを異種の移動エージェント基盤へ移動させることは難しい。

本稿では、インカネーションエージェントによる相互運用の実現について述べる。インカネーションエージェントは、エージェントの持つタスクを抽出し、それを異種移動エージェント基盤上のエージェントに受け渡すことにより、エージェントの移動と同等の動作を実現する。まずインカネーションエージェントの機能概要を述べ、交換タスクへの変換方法、およびエージェントのライフサイクル情報の管理方式について述べる。

2 インカネーションエージェントの機能概要

インカネーションエージェントは、実装方式や表記法の異なる移動エージェント基盤間での、論理的なエージェントの移動を実現する。インカネーションエージェントは、エージェントが異種移動エージェント基盤間を移動する際に必要な、処理の意味的変換と実行状態の継続を行う。

エージェントを移動させるには、移動先でもエージェントの意図する動作が正しく行われる仕組みを用意する必要がある。各移動エージェント基盤は、実行状態、内部状態の管理方法や、備えている機能特徴がそれぞれ異なる。そのため、移動エージェントを異種移動エージェント基盤に移動させるためには、エージェントの行う処理内容を移動先基盤での機能特徴に適合させる必要がある。また、各移動エージェント基盤ごとに異なる、エージェントの実行状態や内部状態の管理方法の整合性を取る必要がある。

インカネーションエージェントは、エージェントから抽出したタスクの交換タスクへの変換、およびライフサイクル情報の管理によって、これらの動作を実現する。

インカネーションエージェントは、交換タスクを相互運用ミドルウェア間でやりとりすることで、エージェン

トの論理的な移動を実現する。本稿では、この論理的な移動をインカネーションと呼ぶ。

相互運用ミドルウェアは、以下の動作によって交換タスクを生成する。(図1)

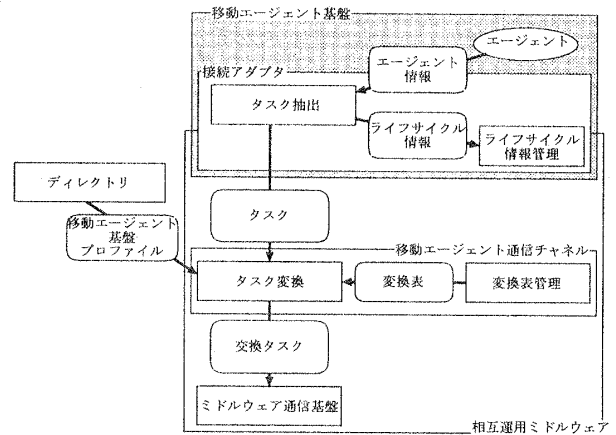


図1: ミドルウェアによる交換タスクの生成

1. エージェントからのインカネーション要求をうけて、接続アダプタが、インカネーションさせるタスクを抽出する。
2. 移動エージェント通信チャンネルが、変換表を用いて、タスクを交換タスクに変換する。
3. ミドルウェア通信基盤が、交換タスクをインカネーション先の移動エージェント基盤へ移送する。

3 交換タスクへの変換

交換タスクは、インカネーションエージェントが扱う、エージェントの意図を移動エージェント基盤に依存しない形式で表現したものである。交換タスクは、FIPA[1]で規定されているACL (Agent Communication Language) で表記され、FIPAのエージェントモビリティプロトコルに従って移送される。

交換タスクへの変換は、移動エージェント通信チャンネルで行われる。移動エージェント通信チャンネル内には、各移動エージェント基盤ごとの変換プログラムが登録される。変換プログラムは、変換表管理機能から取得した変換表を用いて、タスクを交換タスクに変換する。

移動エージェント基盤が備えている機能は、各基盤ごとに大きく異なるため、エージェントの持つタスクをインカネーション先基盤の機能に合わせて適合させる必要がある。その際、インカネーション元の移動エージェント基盤上で提供されている機能と同等の機能に変換するだけでなく、インカネーション先の移動エージェント基盤の特徴を活かした形に変換されることが望ましい。移動エージェント基盤プロファイルをディレクトリから取得

Interoperability Middleware for Mobile Agents: Incarnation Agent

Kenta Cho, Tetsuo Hasegawa, Akihiko Ohsuga, Fumihiko Kumeno, Shin Nakajima

Toshiba Corporation, Mitsubishi Research Institute, Inc., NEC Corporation

し、インカネーション先の移動エージェント基盤の機能に合わせた変換表を取得するように、変換プログラムの実装方法を工夫することで、このようなインカネーションを実現することができる。

プロファイル内に、各移動エージェント基盤が実現可能な機能をケイパビリティ[2]として列挙しておく。タスクを、ケイパビリティの列として表現することで、交換タスクの抽象的なアクション表現を実現する。タスクからケイパビリティ列への変換を行う際には、インカネーション先の移動エージェント基盤が備えているケイパビリティへの変換を行う変換表を取得し、変換を行う。

インカネーション先の移動エージェント基盤が、エージェントアプリケーションに依存する抽象度の高いケイパビリティを備えており、タスクをそのケイパビリティで達成することができる場合は、交換タスクは抽象度の高いケイパビリティ列で表現される。抽象度の高いケイパビリティは、インカネーション先移動エージェント基盤で、その移動エージェント基盤に最適な動作に変換される。タスクが抽象度の高いケイパビリティで表現できない場合は、タスクを抽象度の低いエージェントの基本動作のケイパビリティに分解して、交換タスクを表現する。(図2)

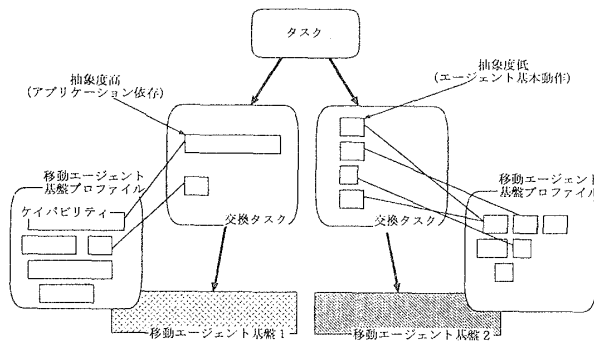


図2: ケイパビリティに合わせた変換

4 ライフサイクル情報の管理

エージェントの実行状態を管理する方法は、移動エージェント基盤ごとに異なるため、その管理情報をそのままインカネーション先の移動エージェント基盤に伝えるだけではエージェントの移動を実現することはできない。そのため、インカネーションエージェントでは、ライフサイクル情報の管理を行うことで、移動に必要な実行状態の継続を実現している。

ライフサイクル情報は、インカネーション元のエージェントの実行状態をミドルウェア内で一時的に管理し、そのエージェントに対して、サスペンドやレジューム、消滅、タスクの結果の反映などの動作を行うための情報である。

ライフサイクル情報を用いることで、エージェントが一時的に他の移動エージェント基盤に移動し、タスクを処理した後、移動元の移動エージェント基盤に戻ってくるといった動作などを擬似的に実現することができる。その場合のインカネーションは、以下のように実現される。(図3)

1. エージェントがインカネーションを開始する。
2. ライフサイクル情報を抽出する。ライフサイクル情報には、インカネーションを開始したエージェントに対して、サスペンドや結果の反映を行う際に必要な情報を含む。

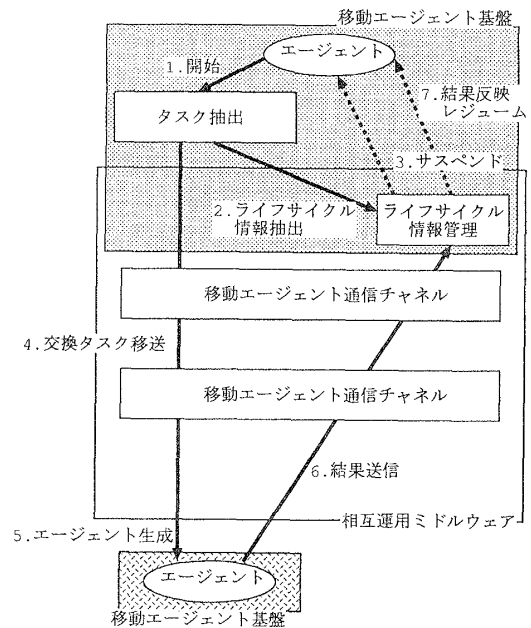


図3: インカネーションの動作の例

3. エージェントをサスペンドさせ、インカネーションの結果が返ってくるまで待たせる。
4. 交換タスクをインカネーション先のミドルウェアに移送する。
5. 交換タスクを移動先エージェント基盤固有のタスクに変換し、タスクを実行するエージェントを生成する。
6. タスクを実行したエージェントは、実行結果をミドルウェアを介して移送する。実行結果は交換タスクとしてインカネーション元のミドルウェアに渡され、移動エージェント通信チャンネルによって移動エージェント基盤固有の形式に変換される。
7. エージェントに、インカネーションの結果が、ライフサイクル情報に含まれる情報に従って反映される。エージェントはレジュームされ、動作を継続する。

これらの動作はミドルウェア内で自動的に行われるため、エージェント側からは移動エージェント基盤内での移動と同様の動作として扱うことができる。

5 おわりに

インカネーションエージェントを利用することで、実装方式などが異なる移動エージェント基盤上への論理的なエージェントの移動を実現することができた。

謝辞 本研究は、「IPA 次世代デジタル応用基盤技術開発事業」の一環として行われているものである。

参考文献

[1] Agent Management Support for Mobility, FIPA98 Specification, <http://www.fipa.org/>, 1998.
 [2] E. Kendall, M. Malkonn. The Layered Agent Patterns. Proceedings of the Conference on Pattern Languages of Programs (PLoP'96), 1996.