

# モバイルエージェントの検索機能の検討

東野 正幸<sup>1</sup> 灘本 拓<sup>2</sup> 高橋 健一<sup>2</sup> 川村 尚生<sup>2</sup> 菅原 一孔<sup>2</sup>

## 1. はじめに

モバイルエージェントとはノード間を移動可能な自律的なソフトウェアである。モバイルエージェント技術は分散システム的设计パラダイムとして通信の効率化や複雑な设计の簡素化など幅広い分野で利用されている [1]。モバイルエージェントの特徴はインスタンスの状態を継続したままノード間を移動できる移動性と移動性を用いたモバイルエージェントの動作を表現できる記述形式にある。これら特徴によりノード間の往復やノードの巡回といった表現が容易となり分散システム的设计の簡素化が期待できる。しかし、モバイルエージェントは遠隔ノードへ自律的に移動可能であるため動作中の分散システム内においてどのモバイルエージェントがどのノードでどのような処理を行っているのかといった実際の動作を把握することが難しくモバイルエージェントの動作確認やデバッグが難しいという問題がある。そこで本稿では分散システム内からモバイルエージェントを様々な条件で検索することでバグの原因の特定を支援する検索機能を検討する。

## 2. モバイルエージェントのモデル

モバイルエージェント技術を用いて構築されたシステムをモバイルエージェントシステムとよぶ。モバイルエージェントシステムのモデルを図 1 に示す。場所はエージェントが動作できる場所であり各々の場所は論理的または物理的に区別される。エージェントは場所内で動作できる。エージェントは他のエージェントとコミュニケーションを行うことができる。各々のエージェントは独立した状態を持っておりコミュニケーションにより双方の状態に変化を及ぼすことができる。またエージェントは場所間を移動できる。このモデルは OMG (Object Management Group) が策定したモバイルエージェントの仕様である MASIF [2] と FIPA が策定したエージェントシステムの抽象アーキテクチャの仕様である FIPA Abstract Architecture Specification [3] を包含している。

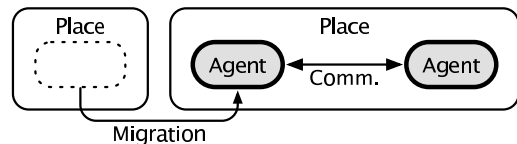


図 1 モバイルエージェントシステムのモデル

Fig. 1 The structure of a mobile agent system.

## 3. エージェントの検索に必要な情報

モバイルエージェントシステム的设计ではエージェントの振る舞いを分かり易く定義すべきであるが実装で発生したバグの原因を動作中のシステムから発見するためにはモバイルエージェントの柔軟性により生じる複雑性に対処する必要がある。一般的にシステムをデバッグする場合はプログラムの状態や状態変化を調査する手法が有効であることからモバイルエージェントシステムのデバッグに有効なモバイルエージェントの状態や状態変化を検索する手法を検討する。

**エージェントの一意識別子** エージェントはコミュニケーションによって状態変化を起こすことからデバッグする際にはエージェントの状態変化が起こった際のコミュニケーションに参加していたエージェントを特定する必要がある。しかしエージェントは複製を行ったり複製後に複製元と複製先が独立して動作する場合があるためデバッグを行う際にはこれらのエージェントのインスタンスを個別に識別できる必要がある。このためエージェントのインスタンスにはシステム内で一意に識別可能な一意識別子を付与する。

**場所の一意識別子** モバイルエージェントは場所間を移動しながらコミュニケーションを行うことからエージェントに状態変化が起こった際のエージェントが滞在していた場所を特定する必要がある。計算機に割り当てられる IP アドレスやポート番号は再利用される可能性があるため場所には再利用されない一意な識別子を付与する。

**状態変化の発生時刻** システムにおけるバグの観測は実時間と紐付けられて報告される場合が多い。このためバグが発生した際の物理時間とエージェントの状態変化

<sup>1</sup> 鳥取大学 産学・地域連携推進機構

<sup>2</sup> 鳥取大学大学院 工学研究科 情報エレクトロニクス専攻

をひも付けて記録しておくことは有効である．一般的に物理的な計算機は各々に物理時計を持っているがその物理時刻には差が存在する．このため、物理的に異なる計算機に設置されたプレースを横断して発生したエージェントの状態変化の順序を物理時刻を用いて正確に記録することは難しい．また、論理時計を用いて順序の正確性を担保する方法も考えられるが、デバッグの対象となるエージェントが存在するプレースの数が多い場合やプレース間の通信遅延が大きい場合に同期の処理によって生じるオーバーヘッドが大きくなる．そこで状態変化の記録はそれぞれのインスタンス毎に非同期で実施する．

**エージェントの状態** エージェントの状態の表現方法はモバイルエージェントの実装方法によって相違する．MASIF[2]ではエージェントが持つ情報としてプロパティと呼ばれる連想配列が定義されている．そこでプロパティにエージェントの一意識別子、現在滞在しているプレースの一意識別子、実装に依存する情報等を格納し、プロパティに変更が生じることをエージェントの状態の変化とする．

#### 4. ログの記録方式と検索方式

特定のエージェントの移動履歴を検索する場合は各々のプレースにログを記録するよりもエージェント自身に記録させたログを参照する方が通信量は少なくなる．一方、各プレースにおけるエージェントの分布状況を検索する場合はエージェント自身にログを保持させるよりも各々のプレースに記録したログを参照する方が通信量が少なくなる．このように検索する対象によって最適なログの記録場所は異なる．そこでエージェントの状態変化に関するログはエージェント自身および滞在しているプレースの双方に記録する．

分散システムにおけるオブジェクトの検索技術としては分散ハッシュテーブルなどの構造化オーバーレイネットワークを用いたものやフラッシングなどの非構造化オーバーレイネットワークを用いたものが代表的である．しかしモバイルエージェントのようにプレース間を頻りに移動したり生成・消滅したりする対象を検索する場合、分散ハッシュテーブルなどの構造化オーバーレイネットワークでは性能が低下する可能性がある[4]．そこで各プレースに記録されたログをフラッシングにより検索する方式を主とし検索する対象によって構造化オーバーレイネットワークを用いた方が効率が良い場合はこれを用いるハイブリッド型の検索方式を検討する．

#### 5. デバッグにおける検索例

モバイルエージェント技術を用いて分散システムを開発する場合、一般的なプログラミングと同様にモバイルエー

ジェントの動作はデザインパターン[5]を用いて設計される場合が多い．モバイルエージェントではプレースを巡回移動するパターンや往復移動するパターンなどがよく用いられる．このようなエージェントがパターンどおりに動作しているか確認するためにはエージェントの移動履歴を検索して確認することが有効である．

モバイルエージェント技術を用いた負荷分散システムをデバッグする場合、システムの負荷に応じてエージェントが各々のプレースに正しく分散しているか確認する必要がある．この場合には各プレースにどのようなエージェントが分布しているかを検索して確認することが有効である．

あるエージェントに問題が生じた場合、その問題の原因はそのエージェント自身だけでなく、そのエージェントの複製元やコミュニケーションを行った相手のエージェントにも原因が存在する可能性がある．この場合、特定のエージェントの派生履歴やコミュニケーション履歴を検索して確認することがデバッグに有効である．

今後はこれらの検索例を用いて提案方式の有効性を評価していく予定である．

#### 6. おわりに

本稿ではモバイルエージェントシステムの動的なデバッグにおける困難性の軽減を目的としたエージェントの検索機能を検討した．モバイルエージェントシステムでは複数のエージェントが移動しながらコミュニケーションを行うためコミュニケーションにより生じたエージェントの状態変化はコミュニケーションを行ったエージェントの時刻と場所を紐付けて検索できる必要がある．次の課題として以下の検討項目が挙げられる．

- 移動を伴うエージェントの状態やコミュニケーションによって生じるエージェントの状態変化を検索するためのクエリ言語の開発
- エージェント間の複雑な関連性と揮発性に対してロバストなエージェントの検索クエリの伝搬方式の開発

#### 参考文献

- [1] Hurson, A. R., Jean, E., Ongtang, M., Gao, X., Jiao, Y. and Potok, T. E.: Recent advances in mobile agent-oriented applications, *Mobile Intelligence: Mobile Computing and Computational Intelligence*, pp. 106-139 (2010).
- [2] Object Management Group, Inc.: *Mobile Agent System Interoperability Facilities Specification* (1997).
- [3] Foundation for Intelligent Physical Agents: *FIPA Agent Management Specification (SC00023K)* (2004).
- [4] 一杉孝之, 村岡洋一: DHTにおけるデータの更新操作を考慮した検索方式の提案, 電子情報通信学会第19回データ工学ワークショップ(2008).
- [5] Lima, E. F. A., Machado, P. D. L., Sampaio, F. R. and Figueiredo, J. C. A.: An Approach to Modelling and Applying Mobile Agent Design Patterns, *ACM SIGSOFT Software Engineering Notes*, Vol. 29, pp. 1-8 (2004).