

モバイルエージェントシステムのデバッグに関する研究の展望

東野 正幸^{†a)} 灘本 拓^{††b)} 高橋 健一^{††c)} 川村 尚生^{††d)}
菅原 一孔^{††e)}

A View of Debugging for Mobile Agent System

Masayuki HIGASHINO^{†a)}, Taku NADAMOTO^{††b)}, Kenichi TAKAHASHI^{††c)}, Takao KAWAMURA^{††d)}, and Kazunori SUGAHARA^{††e)}

1. はじめに

モバイルエージェントとはネットワークに接続された計算機間を移動できる自律的なソフトウェア部品である。複雑な分散システムをモバイルエージェント同士の協調により実現することで、人間が理解しやすい人間社会のようなモデルで設計可能となることから、モバイルエージェントは分散システムの構築技術として有用性が示されてきた。しかし、モバイルエージェントは、分散システムの設計が容易になる一方で、遠隔地の計算機へ移動して自律的に動作することから、何処で何をしているのかといった実際の動作の把握が難しく、システムのデバッグが難しいという問題がある。本稿では、モバイルエージェントのデバッグの困難性の分析し、その困難性を軽減するための動的なデバッグ手法について考察する。モバイルエージェントのデバッグの困難性が軽減されることで、より高度な分散システムの開発が可能になることが期待できる。

2. 研究背景

1993年7月に米ジェネラルマジック社が取得したモバイルエージェント技術に関する基本特許 (US Patent 5,603,031) [1] が2013年7月に失効し商業利用の目処が立ってきた。また、クラウドコンピューティングなどの、より大規模で柔軟な分散システムが要求されるようになり、モバイルエージェント技術の必要性はさらに高まっていくと考えられる。文献 [2] においてもモバイルエージェント技術が再び注目される可能性について示唆されている。

システム開発においてデバッグは必要不可欠である。しかし、複数の国内研究会によって毎年開催されているエージェントに関するワークショップである JAWS^(注1) や、エージェントに関する国際会議である AAMAS^(注2) においては、シミュレーションやアプリケーションに関する研究が多数発表されているが、デバッグに関する研究の発表は少ない。また、エージェントの動作を形式的に検証するモデル検査といった静的手法も提案されているが、システムが大規模化すると計算量が膨大になり検証が難しくなる問題がある。この場合、システムを動作させながらデバッグする動的手法が有効であり、オブジェクト指向言語や分散システムの分野では多数の知見が蓄積されている。しかし、計算機間を移動するエージェントを動的にデバッグする手法はほとんど研究が進んでいない。エージェントの標準仕様である MASIF [3] や FIPA [4], [5] においてもモバイルエージェントの移動に関連するデバッグについては十分に定義されていない。

モバイルエージェントはインスタンスが計算機間を自由に移動できるという点で従来の分散システムとは異なる。従来の分散システムでは、各々の計算機に固定されたインスタンス間の入出力を調べればデバッグが可能であった。しかし、モバイルエージェントは別の計算機に移動するため、デバッグするためには、どの計算機で、どのエージェントと、どのようなデータの入出力を行ない、どのようなデータの操作を行ったかを、ネットワークで接続された分散システム全体から見つけて調べる必要があり、このような要求は既存の分散システムには無く、モバイルエージェント特有の課題である。また、モバイルエージェントの研究分野では、これまでシミュレーションやアプリケーションといった応用に関する研究が主であり、そのデバッグ手法に関しては、まだ研究はあまり進んでいない。

本稿では、モバイルエージェントのデバッグの困難性を軽減するためにモバイルエージェントを動的にデバッグする手法について考察する。

[†] 鳥取大学 総合メディア基盤センター, 鳥取県鳥取市
Center for Information Infrastructure & Multimedia, Tottori University, 4-101, Koyama-Minami, Tottori 680-8550, Japan

^{††} 鳥取大学大学院 工学研究科 情報エレクトロニクス専攻, 鳥取県鳥取市
Department of Information and Electronics, Graduate School of Engineering, Tottori University, 4-101, Koyama-Minami, Tottori 680-8550, Japan

a) E-mail: higashino@tottori-u.ac.jp

b) E-mail: s102035@eecs.tottori-u.ac.jp

c) E-mail: takahashi@eecs.tottori-u.ac.jp

d) E-mail: kawamura@eecs.tottori-u.ac.jp

e) E-mail: sugahara@eecs.tottori-u.ac.jp

(注1): JAWS: Joint Agent Workshops & Symposium, accessed September 9, 2015, <http://jaws-web.org/>.

(注2): AAMAS: Autonomous Agents and Multiagent Systems, accessed September 9, 2015, <http://www.aamas-conference.org/>.

3. 研究課題

3.1 検索クエリ

モバイルエージェントを動的にデバッグするには、バグの原因となるモバイルエージェントを分散システム内から見つけ出す仕組みが必要となる。しかし、モバイルエージェントは自律性を持っているため、エラーが顕在化せずに動作し続ける場合がある。また、それらは分散システム内で拡散して動作する。このため、バグの可能性のあるモバイルエージェントを見つけるには、様々な検索条件を組合せて多角的に絞り込んでいくアプローチが必要になると考えられる。このため、バグの原因の可能性のあるモバイルエージェントの絞り込みには、どのような検索パラメータ及びその組み合わせが、どのようなバグ原因の絞り込みに有効かを明らかにする必要がある。

3.2 検索アルゴリズム

モバイルエージェントは計算機間を移動するため、分散システム内からバグの可能性のあるモバイルエージェントを見つけて出すには、分散ハッシュテーブル (DHT) などの分散システムにおける検索技術が必要となる。しかし、モバイルエージェントのように計算機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、DHT では性能が低下する問題がある [6]。このため、検索対象であるモバイルエージェントの動作の特徴に応じて適切な検索アルゴリズムを切替えたり組合せたりすることで、検索コストの低い検索方式を検討する必要がある。また、それぞれの計算機がエージェントを保持したままネットワークから切断及び再接続することも想定されるため、耐チャーン (churn) 性も要求される。

3.3 動作履歴の記録形式と追跡方式

一般に、モバイルエージェントはメッセージ交換により協調動作が実現されているため、モバイルエージェントの動作履歴を確認するためには、メッセージがどのように交換されたのかを記録する必要がある。しかし、モバイルエージェントはメッセージの送信・受信・生成・削除・加工・複製などを行い、さらに、モバイルエージェント自体もシステム内で生成・削除・加工・複製される。さらに、計算機もネットワークから参加・離脱する。このように、変化が激しく揮発性が有りしかも粗結合しているメッセージの関連性を追跡可能な形式で分散システムに記録し追跡する手法が必要である。

3.4 ステップ実行機構

モバイルエージェントの動作を詳細に調査する場合、一般的なソフトウェアと同様にステップ実行が有効であると考えられる。しかし、既存の一般的なデバッガはインスタンスが計算機間を移動することを想定していないためステップ実行中にエージェントの移動が生じた場合にエージェントを見失ってしまう場合が考えられる。このため、デバッグ中に移動しても見失うこと無く追従しながらデバッグ可能なステップ実行機構の検討が必要である。

3.5 動的な修正方法

モバイルエージェントは動作中に自身を複製したり自身が持つプログラムコードに対して追加・削除などの操作を行う場合がある。このため、バグ原因に関連するモバイルエージェントを修正するためには、モバイルエージェントが持つプログラムコードがどのように頒布されたかを追跡して置換する必要がある。モバイルエージェントシステムにおけるプログラムコードの授受操作を記録し、バグ原因が除去されたプログラムコードへ動的に置換する方法の検討が必要である。

また、大規模な分散システムではシステム全体の停止と再開は高いコストが要求される。このため、バグ原因に関連するモバイルエージェントを局所的に捕捉して修正を適用する機構が必要となる。また、分散システム内のモバイルエージェントに頒布されたプログラムコードを追跡して動的に置換する機構の検討も必要である。

3.6 開発環境の実現

前述した課題を解決したモバイルエージェントの開発環境を実現し、応用システムにおけるシステムの有効性の検証が必要である。また、MASIF [3] や FIPA [4], [5] といったエージェントの標準化仕様の拡張および適用を検討し、標準的な仕様に基づいたモバイルエージェントの開発環境への適用についても検討が必要である。

4. おわりに

システム開発において充実したデバッグ環境は必要不可欠であり、特に副作用の制御が重要となるモバイルエージェント技術においてデバッグ環境の提供は実用化のために重要である。モバイルエージェントのデバッグの困難性を軽減することで、モバイルエージェント技術に基づく分散システムのデバッグに必要なコストを削減できる。また、モバイルエージェント技術を実用化に近づけることで大規模で柔軟性が要求される分散システムをさらに発展させることへの貢献にも期待できると考えている。

謝辞 本研究は JSPS 科研費 15K15982 の助成を受けたものである。

文 献

- [1] General Magic, Inc., "System and method for distributed computation based upon the movement, execution, and interaction of processes in a network," Nov. 1997. US Patent 5,603,031.
- [2] N. Fukuta, "A survey and analysis of mobile agent technologies and their application," IPSJ SIG Technical Report, vol.2011-ICS-164, no.5, pp.1-2, 2011.
- [3] Object Management Group, Inc., "Mobile agent system interoperability facilities specification," 1997.
- [4] Foundation for Intelligent Physical Agents, "FIPA agent management specification (SC00023K)," 2004.
- [5] Foundation for Intelligent Physical Agents, "FIPA abstract architecture specification (SC00001L)," 2002.
- [6] A. Hayakawa, M. Asahara, K. Kono, and T. Kojima, "A strategy for efficient update propagation on peer-to-peer based content distribution networks," IPSJ Online Transactions, vol.3, pp.110-124, 2010.