

モバイルエージェントシステムのためのエージェント検索機能の提案

灘本 拓^{†a)} 高橋 健一[†] 東野 正幸^{††} 川村 尚生[†]
菅原 一孔[†]

Proposition of the agent search function for mobile agent system

Taku NADAMOTO^{†a)}, Kenichi TAKAHASHI[†], Masayuki HIGASHINO^{††}, Takao KAWAMURA[†], and Kazunori SUGAHARA[†]

1. はじめに

モバイルエージェントには分散処理における利点があり、分散処理システムの構築に有効な技術として、様々な研究が行われてきた [1]。しかし、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遠隔地のノードで自律的に移動、動作しているため、バグを引き起こしているエージェントを特定することが困難という問題がある。

このような問題解決のために形式手法を応用した手法として、Mobile Object-Z (MobiOZ) [2], LAM (Logical Agent Mobility) [3] がある。しかしこれらの手法では専用の仕様記述言語を用いる必要があるため新たなコストを要する。また、システムが大規模で複雑な場合、モデル検査のための計算量が膨大になり、静的なデバッグ手法である形式手法は適用が難しい場合がある。各ノードで移動履歴を追跡し、エージェントの現在地を特定し、ローカルでログを確認する手法として MiLog [4] が挙げられる。しかし、動作ログの確認は開発者が個々のエージェントに対して行う必要があるため、大量のエージェントが動作するシステムに用いることは難しい。

こうした問題を解決する手段として動的なデバッグ手法が有効である。そこで本研究では、動的なデバッグ手法としてモバイルエージェント向けの検索機能の提案を行う。

2. エージェント検索機能

モバイルエージェントシステムでは、エージェントの自律性やノード間の移動によってエージェントの動作が複雑に変化する。このため、システムに何らかの不具合が発生した時に、どのエージェントにその原因があるかを確かめることが困難となる。そこで、その原因を引き起こしたエージェントの発見を支援するための検索機能

を提案する。エージェントの検索機能では場所や動作、移動に関する情報を元にネットワーク上のノードに対して検索を行い、不具合の原因を引き起こしたエージェントの発見を支援する。

2.1 検索クエリ

システムに発生する不具合に関連する要因としては

- エージェントの異常な内部状態
- ノードの状態の不具合
- エージェントの間違った場所での動作

が考えられる。そこで、これらの要因を調べるための検索機能を検討する。

特定のエージェントの検索

モバイルエージェントはノード間を自律的に移動するため、開発者がエージェントの動作状況や場所を把握することが難しい。そのため、対象のエージェントを見つけるための検索が必要になる。そこで、各エージェントに対して一意な識別子を与え、そのエージェント ID を指定することで特定のエージェントを検索する。また、複数のエージェントを繰り返し検索する場合、検索の手間を省くため、複数のエージェントを種類ごとに識別化するためのタグを開発者がエージェントに付与することで、同じタグを持つエージェントをまとめて検索する。

状態による検索

開発者がバグに気づいた時、まずどのエージェントがバグの原因であるかを調べる必要がある。そこで、エージェントの振る舞いを元に検索することでエージェントの動作状況を把握し、バグの原因の絞込みを行う。エージェントの基本的な状態は様々なモバイルエージェントプラットフォームに対応できるように FIPA をベースに拡張し、検索に用いる状態を新たに定義する。プラットフォームやアプリケーションレベルでの振る舞いはユーザが定義する状態とし、開発者が任意で定義する。

ノードの検索

あるノードにバグが発生した場合、そのノードに異常な影響を及ぼしたエージェントを特定する必要がある。この時、そのノード上で動作していたエージェントを確認することでバグの原因となるエージェントの絞込みを行う必要がある。そこで、指定したノード上で動作してい

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

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

a) E-mail: s102035@ike.tottori-u.ac.jp

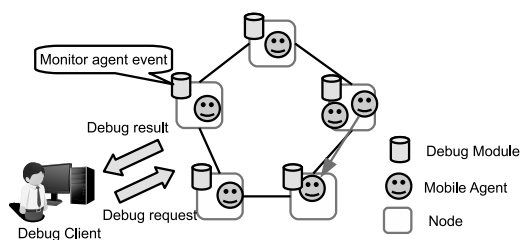


図 1 提案するデバグの構成
Fig.1 Structure of the proposed debugger

たエージェントのログを取得することで、そのノードに不具合を与えたエージェントの絞込みを行う。

移動経路の検索

モバイルエージェントシステムの代表的なバグにエージェントの移動によるものがある。これらはエージェントの移動経路、エージェントの移動頻度に異常な状態が現れるため、それらのエージェントの移動に関する情報を検索する必要がある。そこで、エージェントの移動先、移動元、エージェントが出て行った時間と到着した時間、移動回数のログをノードごとに記録し、それぞれの移動先と移動元のログをつなぎ合わせることで指定したエージェントの移動経路の検索を行う。

上記の検索に用いる検索パラメータとして ID, TAG, STATE, NODE, ROUTE を定義する。ID はエージェントごとに与えた一意な識別子を指定する。TAG は開発者がエージェントに与えたタグを指定する。STATE はエージェントの状態を指定する。NODE はエージェントが動作するノードを指定する。ROUTE は指定したエージェントの移動経路を検索する。

3. 実装

提案するデバグ環境は各ノードに設置するデバグモジュールと開発者の元で動作するデバグクライアントで構成される (図 1)。デバグモジュールは設置されたノードのエージェント実行環境からログを取得し、管理する。デバグクライアントはユーザインタフェースを持ち、開発者からの各デバグ操作を受け付ける。デバグモジュールとデバグクライアントは共に HTTP サーバ機能を保持しており、開発者の要求を受けたデバグクライアントからデバグモジュールに対して要求を送り、デバグ処理の結果をデバグモジュールからデバグクライアントに送ることで各デバグ処理を行う。

4. 利用シナリオ

実際に開発したシステムを通して検索機能が有効に働くことを考察する。分散ハッシュテーブル (以降 DHT とよぶ) を用いた検索アプリケーション開発時に発生したバグを例とする。DHT とは複数のノードで構成するネットワーク上にデータを均等に保存し、取得するシステムであり、ノードの参加・離脱やデータの保存・取得、ネットワークの修正をエージェントで実現している。

移動の間違えによるバグ 指定したデータを取得するエージェント (get_agent) が出力結果を表示しないバグが存在した。これをデバグが用いない場合、開発者は手探りでエージェントの場所と状態を調べ、原因を確かめるため移動処理ごとに停止処理を加え実行することでエージェントの移動経路を確認しなければならず、その結果エージェントが間違った順序で移動していることが分かった。これを検索機能を用いた場合で想定すると、デバグ対象のエージェント ID を検索することで、どのノードで動作しているかが容易に分かる。また、移動経路の検索を行うことで、エージェントの移動が間違っていることが分かりバグの原因特定までの時間を短縮できる。

その他、ノードが参加できないバグやデータが保存できないバグが存在した。また、利用者の旅行条件にあった情報をエージェントが各航空会社やホテル会社から収集するシステムとして旅行支援システムを開発した。このシステム開発時に発生したバグとして、データを保存するフィールドの初期化によるバグや収集したデータの一部が欠けているバグがあった。これらの開発時にどの検索クエリが役立つかを分析した結果を表 1 に示す。表 1 より、各検索パラメータを組み合わせて移動に伴うエージェントのバグの発見を支援できることがわかる。

表 1 バグと利用した検索クエリ
Table.1 Search query and usage scenarios

	ID, TAG	STATE	NODE	PATH
fail to get value	✓		✓	
fail to store value	✓		✓	✓
fail to join	✓	✓		
fail to initialize field				✓
lost a part of data			✓	✓

5. おわりに

本研究では、デバグ時においてバグの原因を発見する際の困難性を示し、それを解決するための検索機能を提案した。今後は提案機能の評価を行う。

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] Ali R. Hurson, Evens Jean, Machigar Ongtang, Xing Gao, Yu Jiao, and Thomas E. Potok. *Recent Advances in Mobile Agent-Oriented Applications*, pp. 106–139. John Wiley & Sons, Inc., 2010.
- [2] Kenji Taguchi and Jin Song Dong. Formally specifying and verifying mobile agents model checking mobility: the MobiOZ approach. *International Journal of Agent-Oriented Software Engineering*, pp. 449–474, 2008.
- [3] Dianxiang Xu, Jianwen Yin, Yi Deng, and Junhua Ding. A formal architectural model for logical agent mobility. *IEEE Trans. on Softw. Eng.*, Vol. 29, No. 1, pp. 31–45, 2003.
- [4] Yamaya Takafumi, Hattori Mitsuhiro, Ito Takayuki, and Shintani Toramatsu. A location transparency method for mobile agents and application of agents' distributed system, 2004.