

異種エージェントシステム間の相互運用

高橋 健一†

鍾 国強†

松野 大輔†

雨宮 聡史†

峯 恒憲‡

雨宮 真人‡

1. はじめに

インターネット上のサービスを組み合わせ利用するための枠組みとして、様々なエージェントシステムが提案されている。これらのエージェントシステムでは、一般にエージェントの活動を助ける環境 (Agent Platform, 以下, AP) が存在し、エージェント間のコミュニケーションはその上で実現される。しかし、これらの AP はそれぞれのエージェントシステムで独自に準備されているため、異種のエージェント間でコミュニケーションを実現することは難しい。このことを実現するための標準化団体として、FIPA [1] がある。FIPA では、Agent Communication Language (以下, FIPA ACL) や、Agent Platform などの仕様が定められているが、1) DF(Directory Facilitator)や AMS(Agent Management System)が必須であり、それらを準備していないエージェントシステムと相互運用することが難しい、2) FIPA ACL では、メッセージに送受信者の物理的なアドレスを指定するため、コミュニケーション相手の物理的位置を明確に意識しなければならない、などの問題点がある。そこで、我々は Agent Platform 間の共通プロトコルとして Agent Platform Protocol (以下, APP) を提案している [2]。エージェントシステム開発者は、AP に APP を実装することで、異種の AP との通信が行えるようになり、様々な場所で開発された様々なエージェントシステムを互いに利用することが可能となる。

そこで、本稿では、APP を 2 つのエージェントシステム (KODAMA[3], JADE[4]) に実装することで、異種のエージェントシステム間の相互運用を試みた結果について報告する。

2. Agent Platform Protocol

APP は、AP 間の通信プロトコルであり、次の状況を想定している。

- ・エージェントは物理的な情報を意識しない。物理的な情報は、AP で解決される。
- ・AP 毎にエージェントの管理方法は異なる。
- ・AP 毎に実現されている機能が異なる。

APP は、12 の method を定義したリクエスト/レスポンス指向のプロトコルで、次の機能を実現する。

- ・サーバを利用しない AP 間での peer-to-peer 通信：エージェントは、他から制限を受けることなく、各々が独立にコミュニケーションを行える。
- ・エージェントメッセージ (APP で利用されるメッセージと区別するため、本稿ではエージェント間で交換されるメッセージをエージェントメッセージと呼ぶ。) の配達：エージェントはコミュニケーション相手の物理的な情報やエージェントメッセージの配送経路を意識なくよい。
- ・AP 間でのエージェント情報の交換：移動したエージェントの情報を交換することで、エージェントの移動に

柔軟に対処できる。

AP は APP に従って目的のエージェントを探し出し、メッセージを配達するため、エージェントは互いの物理的な情報を意識することなくコミュニケーションを行うことができる。

2.1 AP Message

APP で用いられるメッセージを AP Message と呼び、method と header, body から構成され、この順で記述される。

AP Message = {method, header, body}

method はメッセージが表す要求、または、応答を表す。CONNECT¹, DELIVER²に分類される method は要求を表し、RESPONSE に分類される method は応答を表す。header は、method を解釈するために必要な情報で、

< header > = < header-name > : < value >

のように表現される。body は、各 method で必要とされる任意のデータである。例えば、DELIVER の場合、body は、エージェントメッセージとなる。

3. 異種エージェントシステム間の相互運用例

異種のエージェントシステム間での APP の相互運用例として、我々の研究室で研究・開発しているマルチエージェントシステム KODAMA (Kyushu university Open & Distributed Autonomous Multi-Agent) と FIPA 準拠のエージェントシステム JADE を利用した。本節では、まず、KODAMA と JADE について紹介し、これらの相互運用のための方法について紹介する。

3.1 KODAMA

KODAMA は、Java で開発されたソフトウェアエージェントシステムである。KODAMA におけるシステムは、コミュニティと呼ばれるエージェント群の階層的な組み合わせによって形成される。

KODAMA におけるエージェントは、AP として ACZ を利用する。ACZ は、論理的なエージェント空間を物理的な構造にマッピングする機構であり、エージェント間のコミュニケーションで必要とされる物理的通信機能を提供する。この機構により、エージェントは通信相手の物理的な位置を気にすることなく、自由にコミュニケーションを行うことができる。

3.2 JADE

JADE は、FIPA に準拠したエージェントシステムである。我々が JADE を相互運用の対象として選んだのは、1) ソースが公開され、2) KODAMA と同じ Java で開発されているので扱い易い、ためである。JADE における AP は、複数のエージェントコンテナから構成される。コンテナはエージェントの実行環境であり、各コンテナには、複数のエージェントが存在する。このため、JADE-AP は 3 つの方法を利用し、エージェントメッセージを配送する。

¹サーバを利用しない peer-to-peer ネットワークを構成するための method 群

²エージェントメッセージを配送するための method 群

†九州大学大学院システム情報科学府

‡九州大学大学院システム情報科学研究院

Case 1. 送信先が同一コンテナ上に存在する場合：イベントオブジェクトを利用してメッセージが送られる。

Case 2. 送信先が同一の JADE-AP 上であるが、異なるコンテナ上に存在する場合：Java RMI を利用してメッセージが送られる。

Case 3. 送信先が別の AP の場合：FIPA の基準に従って、IIOP (Internet Inter-ORB Protocol) を利用してメッセージが送られる。

3.3 システム間の相互運用を実現する上での問題

KODAMA と JADE を相互利用するために、APP の実装とエージェントの構造上の違いについて考慮した。

3.3.1 APP の実装

KODAMA への APP の実装はすでに行われている [2] ので、前節で述べた JADE-AP の Case 2, 3 に関して、JADE に APP の実装を行った。この際、JADE への APP の実装は、APP がエージェントの仕組みに制限を与えないため、容易に実現できた。エージェントメッセージの伝達とエージェントが移動した時の対処方法について紹介する。

エージェントメッセージの伝達 エージェントメッセージは、DELIVER¹ と LINK² を利用して送られる。最初に、AP は目的のエージェントが存在する AP との間での接続の有無を調べ、接続がなければ、LINK を利用して通信経路の作成を行う。次に、エージェントメッセージをメッセージの body 部とし、目的のエージェントが存在する AP に届ける。

エージェントが移動した時の対処方法 AP はエージェントの位置を次の手順で発見する。

1. 2つの AP (AP1, AP2) がある時は、AP1 が AP2 に DELIVER を送るとする。この時、もし目的のエージェントが存在しなければ、AP2 は AP1 に FAILURE を送る。
2. FAILURE を受け取った AP は、目的のエージェントのアドレスを調べるため、AP2 に LOCATE_AGENT³ を送る。
3. AP2 は、目的のエージェントが存在するアドレスを探す。目的のエージェントが見つければ、AP1 にそのアドレスを返す。見つからなければ、FAILURE を返す。
4. 手順 3 で目的のエージェントのアドレスが見つかっていれば、AP1 はそのアドレスに対して DELIVER を送る。

このように、物理的な情報を AP が隠蔽するため、エージェントは物理的なネットワークを意識することなくコミュニケーションを行う。

3.3.2 エージェントの構造上の考慮

KODAMA におけるエージェントは、階層的なコミュニティ構造を形成する。この際、すべてのエージェントは階層上のいずれかの位置に割り当てられる。KODAMA では、このコミュニティ構造によって、エージェント間のコミュニケーションは制御される。この構造を考慮して、JADE に準備されていた DF を、KODAMA の最上位コミュニティの一員として割り当てた。逆に、DF は、目的のエージェントが JADE 側で見つからなかった場合、検索要求を KODAMA の最上位コミュニティのポータルエージェント

に送ることとした。これらによって、互いのエージェントが提供するサービスを発見、利用できるようになった。

3.4 KODAMA と JADE によるチャットシステム

APP を利用した、KODAMA と JADE の間の相互運用性能を確認するためにチャットシステムを実現した。本実験で実装したチャットシステムでは、各ユーザに ChatAgent が割り当てられる。各ユーザは、チャットを始めるときにチャットの相手を指定する。このことで ChatAgent は、指定されたチャットの相手を捜し出し、そのときチャットに参加しているメンバー一覧を受け取る。そして、受け取ったメンバー一覧に対してチャットへの参加を通知することでチャットへの参加が行われる。チャットへの書き込み、チャットからの脱退についても、同様に保持しているチャットメンバー一覧の相手に対して、書き込み、脱退通知を行うことで実現している。

実験では、ユーザ A, B に対する ChatAgent を KODAMA で、ユーザ C, D に対する ChatAgent を JADE で実現し、各 ChatAgent は、それぞれ異なる Java Virtual Machine 上 (KODAMA では異なる ACZ 上、JADE では異なるコンテナ上) で動作させた。この結果、以下のことが確認できた。

- ・ AP は APP を利用することで物理的な情報をエージェントから隠蔽したため、各エージェントは異なる AP が存在する環境においても、その違いを意識することなく互いにコミュニケーションを行うことができる。
- ・ 各エージェントシステムで実現されたサービス (エージェント) の発見方法はそれぞれ異なる。すなわち、サービスの発見は、それぞれのエージェントシステムに任せられなければならない。JADE で準備された DF を KODAMA の最上位コミュニティの一員として割り当て、KODAMA と JADE の間で互いに目的のサービスを発見する仕組みが提供されているため、目的のサービスを互いに利用することができる。

4. 終わりに

本稿では、異種エージェントシステム間での相互運用を可能とする APP の性能を確かめるために、KODAMA と JADE の間で、チャットシステムを実現した。チャットシステムで、各エージェントが異なる環境上に存在するサービス (エージェント) を発見し、その環境の違いを意識することなくコミュニケーションできることを確認した。

今後の課題として、さらに多くのエージェントシステム間での相互運用の実現があげられる。

参考文献

- [1] FIPA - Foundation for Intelligent Physical Agent Specification. <http://www.fipa.org>
- [2] 高橋健一, 鍾国強, 雨宮聡史, 峯恒憲, 雨宮真人: エージェント基盤のための通信プロトコル: Agent Platform Protocol, MACC2001, pp. 30-37, Nov. 2001
- [3] Guoqiang Zhong, Satoshi Amamiya, Ken'ichi Takahashi, Tsunenori Mine, Makoto Amamiya: The Design and Implementation of KODAMA System, IEICE Transactions INF. & SYST., Vol. E85-D, No. 04, 637-646, Apr. 2002
- [4] JADE Homepage. <http://sharon.cse.it/projects/jade>

¹ エージェントメッセージを配送する method

² AP間通信経路を作成する method

³ 物理的なエージェントアドレスを検索する method