

マルチエージェントシステム間を相互接続するゲートウェイ

須栗 裕樹† 児玉 英一郎‡ 宮崎 正俊‡

†株式会社コムテック ‡岩手県立大学ソフトウェア情報学部

1. はじめに

近年、エージェントに関する数多くの研究が行われており[1][2]、その研究成果の実社会への応用が期待されつつある。このエージェントという言葉にはさまざまな定義、分類があるが、本稿では、Yoav Shohamによって用いられた心的状態という概念[3]をベースに、「周りの状態と自分自身の持つ心的状態に基づき、プランを生成し実行するソフトウェア」をエージェントと呼んでいる。この心的状態は、通常、「信念」、「約束」、「選択」、「能力」などから構成されているが[3]、別な構成要素で定義した先行研究もある[1]。ここでは構成要素を限定せずに、これらを総称して心的状態と呼んでいる。また、複数のエージェントと、それらが協調してタスクを遂行するために必要なエージェント管理サービス（メッセージ転送や名前解決などを行う）を合わせたものをマルチエージェントシステムと呼ぶ。このようなマルチエージェントシステムには、FIPA 準拠のマルチエージェントシステムや従来からの KQML を使用したマルチエージェントシステムなど異なるアーキテクチャを持つものがある。

本稿では、これらの異なるアーキテクチャを持つマルチエージェントシステム間の相互接続性に関する問題点を明示し、その問題点を解決するための手法を提案する。

2. 相互接続の必要性

近年標準化が進んでいる FIPA 準拠のマルチエージェントシステムと、従来からの KQML を使用したマルチエージェントシステムとは、現状、相互接続することがで

きない。また、FIPA 準拠のマルチエージェントシステム間、KQML を使用したマルチエージェントシステム間など、同一アーキテクチャのマルチエージェントシステム間においてさえ、仕様のバージョンアップによる変更に対応していないなどの理由から、微妙に互換性の無いマルチエージェントシステムが数多く存在し、その相互接続は単純ではない。

このような状況では、マルチベンダ間でエージェントに基づくアプリケーションの開発や運用を行うことが難しい。エージェント技術の更なる普及のためには、異なるアーキテクチャを持つマルチエージェントシステム間で簡単に相互接続可能であることが望まれる。

異なるアーキテクチャを持つマルチエージェントシステムが相互接続できない主要因を以下に示す。

- エージェントの心的状態の構成要素が異なる。例えば、マルチエージェントシステム A のエージェントは心的状態の構成要素として「約束」を持つが、マルチエージェントシステム B のエージェントは持たないとしたら、単純に A のエージェントから B のエージェントに「約束」を取りつけることはできない。
- エージェントの使用するエージェント通信言語 (FIPA ACL, KQML など) が異なる。
- エージェント管理サービスの機能やそのインタフェースが異なる。このため、異なるマルチエージェントシステムにおけるエージェントの名前を解決し、メッセージを送信することができない。

3. 相互接続のための手法考察

この問題を解決するため、以下の(1)、(2)の手法について考察を行う。

Gateway for Interconnecting Multi-Agent Systems

† Hiroki SUGURI,

‡ Eiichiro KODAMA, ‡ Masatoshi MIYAZAKI

† COMTEC LIMITED.

‡ Faculty of Software and Information Science,
Iwate Prefectural University.

(1)すべてのマルチエージェントシステムで、心的状態の構成要素、エージェント通信言語、エージェント管理サービスを共通のものとする。

(2)心的状態の構成要素、エージェント通信言語、エージェント管理サービスを相互に変換するゲートウェイをマルチエージェントシステム間に設ける。

(1)と(2)に対する考察結果として、それぞれの利点、欠点を以下に示す。

(1)の利点：

各マルチエージェントシステム上で動作するアプリケーションの開発と販売において、開発コストを軽減でき、販売の市場が広がる。

(1)の欠点：

マルチエージェントシステムが動作する環境が多様であるため、機能のレベルを一意に決定できない。また、そのような共通仕様で実装することを強制するのは困難である。

(2)の利点：

異なる機能のマルチエージェントシステムを混在できる。また、既存の仕様や実装に手を加える必要がない。

(2)の欠点：

マルチエージェントシステムのスループットの低下や誤変換の可能性が考えられる。

以上を比較すると、(1)の手法は技術的問題以外の要素を含んでいるため、実現困難であると思われる。このため、本研究では(2)の手法による相互接続を提案する。

4. ゲートウェイによる相互接続

異なるアーキテクチャを持つマルチエージェントシステムの本提案手法による相互接続を図1に示す。図1から分かるように、本手法では、ゲートウェイで以下のことを行う。

- エージェントの持つ心的状態の構成要素の違いの吸収
- エージェント通信言語の翻訳
- エージェント管理サービスの相互接続

このため、各マルチエージェントシステム内のエージェントからの通信メッセージは、ゲートウェイにて変換され、他マルチ

エージェントシステム内のエージェントに転送される。

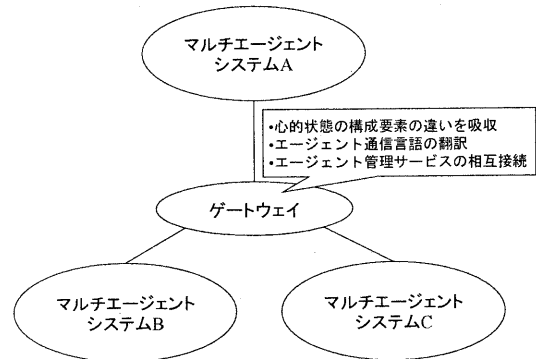


図1 ゲートウェイによる相互接続

5. おわりに

異なるアーキテクチャを持つマルチエージェントシステムを相互接続する必要性について述べ、そのための手法の提案を行った。

本提案手法では、ゲートウェイを用い、エージェントの心的状態、エージェント通信言語、エージェント管理サービスの違いを吸収している。これにより、異なるアーキテクチャを持つマルチエージェントシステム内のエージェント間での通信が可能となる。

今後は、本手法で提案したゲートウェイの実装を行い、その正当性、有効性の検証を行う予定である。

参考文献

- [1] 加藤貴司, 唐橋拓史, 木下哲男, 白鳥則郎: 心的状態に基づくエージェントコミュニケーションの機構について, 情報処理学会全国大会講演論文集, 第54回平成9年前期, No.2, pp.345-346, (1997).
- [2] 佐藤義顕, 沢村一: マルチエージェントシステムのための通信プロトコルとその効率分析, 情報処理学会全国大会講演論文集, 第55回平成9年後期, No.2, pp.370-371, (1997).
- [3] Yoav Shoham: Agent-oriented programming, Technical Report STAN-CS-90-1335, Computer Science Department, Stanford University, (1990).