

異種エージェント相互運用機構の開発と評価 Development and Evaluation of Facilitator of Heterogeneous Agent

打矢 隆弘† 前村 貴秀‡ 木下 哲男§
Takahiro Uchiya Takahide Maemura Tetsuo Kinoshita

1. はじめに

特定の用途・目的に応じて独自開発された異なるエージェントプラットフォーム間を相互接続し、異種エージェントの相互運用性を確保することは、1.既存資産の有効活用、2.得意不得意分野に応じた処理の切分け、3.単一プラットフォームでは対処できない高度で複雑な問題の解決などを可能とする上で重要である。

これまで異種エージェントの相互運用に向けた仕組みがいくつか提案されているものの、拡張性の乏しさやプラットフォームの実装変更への対処が困難といった課題が残っていることから、現実問題に即した実用的な異種エージェントの相互運用の実現が課題となっている。

本研究では上記の課題の解決に向けて、異種エージェント相互運用機構(ファシリテータ)を提案する。本提案機構の特徴は以下の2点である。

特徴 1. 実用的な相互運用機能を網羅

特徴 2. プラットフォーム実装変更への追従性や、新規プラットフォームとの相互運用の際の拡張性を有す

本稿では、相互運用機構の概要と設計・実装及び、評価実験について述べる。そして、評価実験により、上記の特徴を有する相互運用機構の有効性を検証する。

2. 関連研究

異種エージェントの相互運用に焦点を当てた研究として、以下の3つが挙げられる。

Agentcities プロジェクト[1]における相互運用では、相互運用可能なプラットフォームが FIPA 準拠のものに限定されるため、多数の独自仕様プラットフォームとの相互運用ができず、既存資産の活用が困難である。

高橋ら[2]の研究では、相互運用のためのエージェントプラットフォームプロトコルを提案し、FIPA 準拠と独自仕様プラットフォームの相互運用を実現しているものの、対象プラットフォームの大幅な実装変更を必要とするため、プラットフォームの仕様変更やバージョンアップが発生する度、相互運用機構の対応も要求されるという問題がある。

須栗ら[3]の研究では、ゲートウェイエージェントを用いて FIPA 準拠と独自仕様プラットフォームの相互運用を実現しているが、プロトコルレベルやサービスレベルといったレベルでの相互運用には未対応であるため、現実的に即した柔軟な相互運用が困難である。

3. 異種エージェント相互運用機構の提案

3.1 概要

本研究では、それぞれ独立に開発・運用されるプラットフォームが混在した環境での異種エージェント相互運用の実現を目指し、その研究を実証的に進めるために、まず、データ処理に優れた FIPA 準拠 SAGE プラットフォーム[4]とエージェント組織化に優れた独自仕様 DASH プラットフォーム[5]を選定し、相互運用機構を提案し、その設計・実装を行う。

一般に、異種エージェントの相互運用実現手法としては、以下の3通りの手法が考えられる。

(手法 1)エージェントの動作環境であるプラットフォーム自体を修正して、異種エージェントの相互運用を実現する方法

(手法 2)異種エージェントの相互運用機能をプラグインとして開発し、エージェントプラットフォームにインストールして実現する方法

(手法 3)エージェントの相互運用機能をそのプラットフォーム上に動作するエージェントに実装する方法

本研究では、上記手法のメリット、デメリット、作業量、実現の難易度を考慮し、柔軟な異種エージェント相互運用機構の実現に適切と思われる(手法 3) 相互運用エージェントによる相互運用の仕組みを採用した。この手法では、プラットフォームの仕様変更・バージョンアップへの追従が容易であるという利点を有する。

3.2 設計と実装

提案する相互運用機構は、以下の3つの内部機構から構成される。このうち、(M1)、(M2)の機構については、上記で述べたとおり相互運用エージェントとして実現する。

(M1)通信機構

ソケットを利用したプラットフォーム間の通信処理を行う。通信路レベルの相互運用を実現する。

(M2)メッセージ変換機構

・メッセージ変換

メッセージ変換テーブルを用い、送信されたメッセージを送り先のプラットフォームのメッセージ形式に変換する。メッセージレベルの相互運用を実現する。

・プロトコル適用

異種エージェント間でrequest、query、request-when、proposeなどのプロトコルに従った会話を行うことを可能とする。プロトコルレベルの相互運用を実現する。

(M3)サービス検索機構

UDDI レジストリに基づくサービス検索を実現し、異種エージェントに対し、通信相手の場所やサービス内容を提供する。サービスレベルの相互運用を実現する。

† 東北大学電気通信研究所, Research Institute of Electrical Communication, Tohoku University

‡ 東北大学大学院情報科学研究科, Graduate School of Information Sciences, Tohoku University

§ 東北大学情報シナジーセンター, Information Synergy Center, Tohoku University

上述した設計に基づく相互運用機構を用いたメッセージ通信の例を図1に示す。

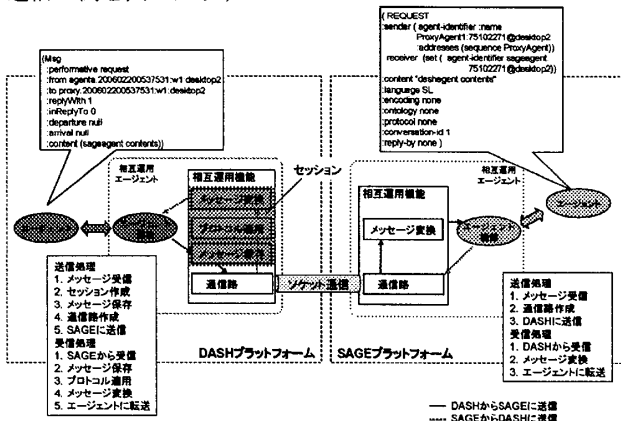


図1：相互運用機構を用いたメッセージ通信の例

4. 実験と評価

4.1 実験内容

評価のために以下の3つの実験を行った。

実験1：異種エージェント相互運用の特徴を活かしたアプリケーション（自動車故障診断システム）の試作と評価を通じて、相互運用機構の機能と動作を検証する。

実験2：相互運用対象を拡大した際の相互運用機構の再利用性を評価する。

実験3：同じサービスを他プラットフォームで再実現する場合と相互運用機構を使って実現する場合の作業量を比較し、相互運用機構の有効性を評価する。

4.2 実験結果と評価

実験1：相互運用機構の動作確認と評価

試作アプリケーション実行時の出力画面、出力ログから、通信路レベル、メッセージレベル、プロトコルレベル、サービスレベルの各レベルでの相互運用が実現されていることを確認し、実用的な相互運用機能が網羅されていることを確認・検証した。特に、プロトコルレベルやサービスレベルでの相互運用を実現している点は、既存手法と比較した場合の大きな特徴である。

実験2：相互運用対象拡大時の再利用性の検証

新規にFIPA準拠のJADEプラットフォーム[6]を運用対象とした場合の相互運用機構の修正にかかる作業工数を表1に示す。ここでは、DASHエージェント、SAGEエージェント用の相互運用機構の修正によりDASHエージェントとJADEエージェントの相互運用を実現する。

表1：相互運用対象拡大時の作業工数

	DASH	JADE
ソースコード総行数[n]	1250	1664
新規作成、修正行数[c]	68	82
流用行数[n-c]	1182	1582
ソースの再利用率 $(((n-c)/n) \times 100)$	94%	95%

表1に示すとおり、既存ソースの94%以上を再利用し少量の作業で効率的にDASH、JADEの相互運用を実現することができた。わずかな修正で相互運用対象の拡張が可能となることから、相互運用機構自体の再利用性、及び、新

規プラットフォームとの新たな相互運用における拡張性が確保できることを実証した。

実験3：相互運用機構の有効性の検証

DASHプラットフォームの温泉ホテル検索サービスをSAGEプラットフォームで利用する場合、そのサービスをSAGEプラットフォームで再度実装して実現する手法と、相互運用機構を使って実現する手法の作業量を比較した結果を表2に示す。

表2：作業工数の比較

	再度実装する方法	相互運用機構を使用する方法
ソースコード総行数[n]	418	406(DASH側+SAGE側)
新規作成、修正行数[c]	418[M1]	62[M2]
流用行数	0	344
作業量の削減率 $(((M1-M2)/M1) \times 100)$	—	86%

実験結果から、他のエージェントシステムのサービスを利用する場合、既存サービスを自分のプラットフォームで再度実装して実現する手法より、相互運用機構を使用する手法の方がより効率的であり、単純にソースコード数を作業工数と想定する場合、作業量の86%が削減できることが実証された。これにより、相互運用機構を導入することにより既存サービスを再利用する場合の作業量が大幅に削減可能であることが確認され、相互運用機構の有効性を確認できた。

5. おわりに

本研究では、異種エージェントの柔軟な相互運用を目的として、異種エージェント相互運用機構を提案した。そして、試作した相互運用機構の有効性を検証するために実証実験を行い、提案機構の有効性を確認した。

今後の予定として、多様なプラットフォームを交えた相互運用機構の動作検証を実施する予定である。また、実用面でのさらなる有効性を検証し、運用を通して異種エージェント相互運用のためのさらなる知見を獲得してゆきたい。

参考文献

[1] Agentcities project, <http://www.agentcities.org/>
 [2] 高橋健一, 鍾国強, 雨宮聡史, 峯恒憲, 雨宮真人, "エージェントプラットフォームを相互運用するためのメッセージ通信プロトコル", 電気学会論文誌 C, Vol.123, No.8, pp.1503-1510, (2003).
 [3] Hiroki Suguri, Eiichiro Kodama, Masatoshi Miyazaki, Isao Kaji, "Assuring Interoperability between Heterogeneous Multi-Agent Systems with a Gateway Agent", IEEE International Symposium on High Assurance Systems Engineering, pp.167-170, (2002).
 [4] SAGE - Scalable Fault Tolerant Agent Grooming Environment, <http://sage.niit.edu.pk/>
 [5] DASH - Distributed Agent System based on Hybrid architecture, <http://www.agenttown.com/dash/index.html>
 [6] JADE - Java Agent DEvelopment Framework, <http://jade.cselt.it/>