

複数ワークスペース間におけるマルチエージェント組織化手法

打矢 隆弘^{†1} 高橋 秀幸^{†2} Wenpeng Wei^{†2} 内匠 逸^{†1} 木下 哲男^{†2}

リポジトリ型フレームワーク DASH では、ネットワーク上に配置されているリポジトリを活用し、利用者要求に応じてリポジトリ内でエージェントの組織構成を実施し、形成された組織をワークスペースに実体化することで要求駆動の動的なシステム構成を実現している。しかし、分散環境上のワークスペースで生成された個々のエージェントに対して組織構成を行う手法が確立されていない。そこで本研究では、複数ワークスペース間におけるエージェント組織化手法を提案し、エージェントシステムの柔軟な運用を実現する。

Agent Organization Method among Multiple Agent Workplaces

TAKAHIRO UCHIYA^{†1} HIDEYUKI TAKAHASHI^{†2} WENPENG WEI^{†2}
ICHI TAKUMI^{†1} TETSUO KINOSHITA^{†2}

1. はじめに

近年、コンピュータの普及やネットワーク環境の整備に伴い、様々な社会的サービスがコンピュータやネットワークに依存するようになってきている。特に、社会的基盤を担うシステムなどでは、より高品質・高信頼なサービスを実現する必要があるため、動作環境に適して自律的に処理を行うことが可能な柔軟性を持つシステムが求められている。また、多くの利用者によって利用されるシステムでは、コンピュータの操作に不慣れな利用者でも容易に利用できるようにする必要があるため、利用者の状態などに応じて適切なサポートを行うことが可能な、利用者指向のシステムが求められている。こうした柔軟なシステムを実現するための手法として、エージェント指向コンピューティングが有効であるといわれている。エージェント指向コンピューティングとは、自律性・協調性を有するソフトウェアコンポーネントであるエージェントによってシステムを構築する手法である。複数のエージェントによって構成され、それらの協調動作を通して問題解決を行うシステムは「エージェントシステム」と呼ばれる。

エージェントシステムを実現するためには、エージェントの動作環境を提供するための仕組みであるエージェントフレームワークが必要不可欠である。一般的に、エージェントフレームワークはエージェントの自律的動作を実現するための推論機構・エージェント間の協調動作を実現するための通信機構などを提供するもので、エージェントの基本動作を保証するものとなっている。現在、特定の用途に応じて、DASH[1,2], SAGE[3], JADE[4], LEAP[5], OMAS[6] など数多くのエージェントフレームワークが提案され、そ

れらを用いて幅広い知的分散環境の適用が模索されている。本研究ではエージェントフレームワーク DASH に着目する。DASH では、リポジトリと呼ばれるエージェントの貯蓄・管理を行う機構を用い様々なエージェントを格納している。そして、利用者の要求に答えられるエージェント組織をリポジトリ内で動的に構成し、それをエージェント動作環境であるワークスペースに生成することでサービスを提供する。また、利用者の状態の変化に応じて、リポジトリを用いて組織の再構成を行うことにより、自律的に処理を行うことが可能であり、柔軟性を持つシステムとなっている。このように DASH ではリポジトリを用いて組織を構成するような仕組みが提供されている。しかし、リポジトリを用いずに組織を構成する仕組みについては提供されていないため、次の状況には対応出来ない。

- ・リポジトリの機能停止時
- ・リポジトリと接続できない環境
- ・ワークスペースエージェント同士の組織化

そこで本研究では、様々な環境下で柔軟に動作するシステムの実現を目的として、ワークスペース上に生成されたエージェントに着目し、複数のワークスペースに生成されたエージェントの組織構成手法を提案する。本手法では、DASH に対して新たに「ワークスペース用組織構成機構」を導入することにより、リポジトリを用いることができない環境下や、複数環境をまたいだワークスペースエージェントの協調処理が必要な場所で、組織構成を実行することが可能となる。

2. エージェントフレームワーク DASH

2.1 DASH

^{†1} 名古屋工業大学 大学院工学研究科

^{†2} 東北大学 電気通信研究所

エージェントシステムはエージェントシステムを開発・運用することを目的として構築されたエージェントフレームワークを用いて作成される。フレームワークには様々な種類が存在するが、本研究では DASH を利用する。

<特徴>

DASH はエージェントの蓄積・管理を行うリポジトリと呼ばれる機構を持つリポジトリ型エージェントフレームワークであり、我々の研究グループで開発・研究が進められている。リポジトリに様々なエージェントを一元的に集積することで、利用者の要求に合った組織を必要に応じて動的に構成する。また、構成された組織の不足機能などをリポジトリに要求し、組織を再構成して変動に対応する柔軟性を有する。DASH の構成を図 1 に示す。

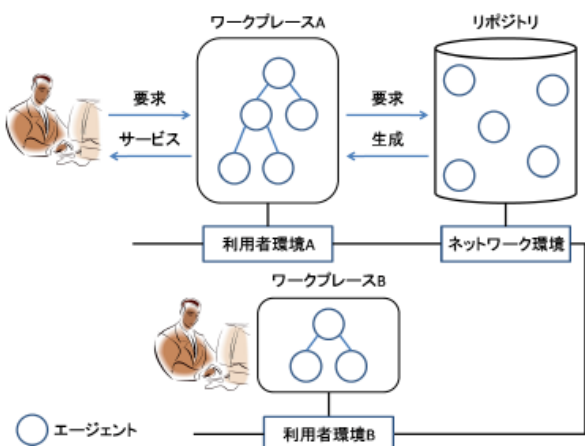


図 1: エージェントフレームワーク DASH

・リポジトリ

リポジトリは、蓄積されたエージェントを利用者の要求に即座に応答できる状態で保持すると共に、利用者の要求に応じてエージェント組織を生成する機構である。リポジトリでは、動作環境となるワークスペース上で動作するエージェントからの要求を受信し、それをリポジトリ内のエージェント群に配信する。そして、当該要求に対処可能と判断したエージェントを特定して、その複製をワークスペース上の新たなエージェントとして生成したり、特定されたエージェントがマネージャエージェントとなり、タスクを実行する他のエージェントをコントラクタとするエージェント組織を動的に生成し、その複製をワークスペース上に生成したりする。

・ワークスペース

ワークスペースは、リポジトリによって生成されたエージェントシステムを動作させる環境である。ワークスペース上では、エージェントシステムが他のエージェント群とメッセージ交換を行ったり、各エージェントに格納された動作知識に基づいて協調問題解決を実行したりする機能が提供されている。

・ネームサーバ

ネームサーバは DASH エージェントの情報を保持し、問い合わせに答える能力を持つ。

2.2 エージェントの組織化

DASH におけるエージェントの組織構成は、契約ネットプロトコルを基にした組織構成のための協調プロトコルである組織構成プロトコルを利用している。DASH では、組織構成を行う上でエージェント間に共通するルールをまとめた組織構成ルールセットを各エージェントにインクルードし、タスクについての詳細情報を追加で記述することで組織構成を行うことができる。組織構成プロトコルでは、エージェントは次の 2 つのいずれかの役割を担って動作する。

<マネージャエージェント>

解決すべきタスクを持つエージェントで、タスクを分割したサブタスクを他のエージェントに依頼して処理する。タスクの分割、サブタスクの割り当て、サブタスクの処理結果の結合などを担当する。

<コントラクタエージェント>

マネージャから依頼されたタスクを引き受けて実行するエージェントである。実現可能なタスクの判断や引き受けたタスクを処理する機能を持つ。

組織構成は利用者の要求によって駆動し、要求を処理可能としたマネージャエージェントがまず特定される。その後の組織構成動作の流れを以下に示す。

1. タスク通知

マネージャエージェントがリポジトリ内のエージェントに対してタスク通知のメッセージを一括送信する。

2. 入札

コントラクタエージェントがタスクについて実現可能か否かを判断し、実現可能であればマネージャエージェントに対して入札メッセージを送信する。

3. 落札

マネージャエージェントは一定時間で入札を締め切り、最良の入札を行ったコントラクタエージェントに対して落札メッセージを送信する。

4. 生成

上記の過程を経て契約を結んだエージェントがエージェント組織のメンバとして決定され、各メンバの複製をワークスペース上に生成する。

2.3 エージェント開発支援環境 IDEA

エージェント開発支援環境 IDEA[7] は柔軟性・適応性を有する DASH エージェントを実装するためのインタラクティブなエージェント開発支援環境である。IDEA には「検索・取り込み」、「開発」、「動作シミュレート」、「登録」といった 4 種類の支援機構が存在し、開発者はエージェントシステムのプログラミングや動作テスト・デバッグを容易に実行できる。例えば、図 2 の動作シミュレータ画面を用

いて、DASH エージェントがリポジトリで組織化され、ワークスペースに生成される様子を確認できる。

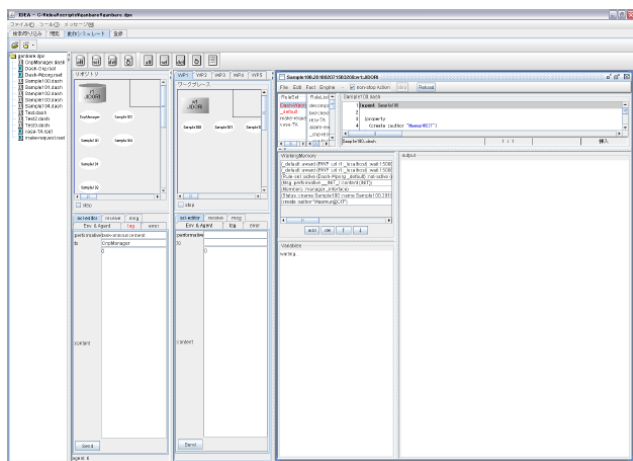


図 2: 開発支援環境 IDEA における動作シミュレート画面

3. ワークスペース用組織構成機構の提案

3.1 概要

本研究では DASH の動作環境となるワークスペース上で生成されたワークスペースエージェントを対象とする。このエージェントは現行 DASH において組織構成を行う動作知識を持っておらず、組織構成に必要なやり取りを行うことができない。そこでワークスペースエージェントがワークスペースをまたいだ組織構成を行うために、提案機構は

- ・エージェントに対し、ワークスペースをまたいだ組織構成を行うために必要な動作知識
- ・フレームワークに対し、必要な動作知識を実行するためのアクション

の 2 つを提供する。

以上によりワークスペース上で生成されたワークスペースエージェントは組織構成に必要な知識を有し、組織構成に必要なやり取りを行うことができる。

3.2 ワークスペース用組織構成機構の設計

ワークスペース用組織構成機構の設計について述べる。ワークスペース用組織構成機構は、

- ・ワークスペース用組織構成ルールセットの提供機能
- ・ルールセットの自動付与機能
- ・稼働ワークスペースに対するタスクブロードキャスト送信機能
- ・ネットワークを考慮した入札締切時間の自動調整機能を有する。以下にそれぞれについて説明する。

3.2.1 ワークスペース用組織構成ルールセットの提供機能

エージェントの組織構成は、組織構成に参加するエージェントが必要な動作知識を持ち、やり取りを行うことで実現している。そのため、ワークスペースでの組織構成に必要な動作知識を外部にワークスペース用組織構成ルールセ

ットとして記述し、これをエージェントがインクルードすることで必要な動作知識を保有し、組織構成を実行する形式を採用した。ワークスペース用組織構成ルールセットはマネージャエージェントが組織構成を行うタイミングから利用される。ルールセットに記述されている動作知識に基づく動作の流れを図 3 に示す。

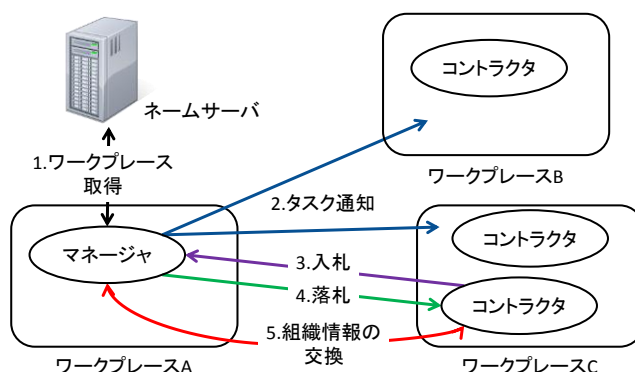


図 3: 複数ワークスペース間でのエージェントの組織化手順

<複数ワークスペース間におけるマルチエージェント組織化手法>

1. 稼働ワークスペースの取得

マネージャエージェントがネームサーバから通信可能範囲の稼働ワークスペースのリストを取得する。

2. タスク通知

1 で取得した通信可能範囲の稼働ワークスペースに対して、マネージャエージェントがタスク通知のメッセージをブロードキャスト送信する。

3. 入札

コントラクターエージェントがタスクについて実現可能か否かを判断し、実現可能であればマネージャエージェントに対して入札メッセージを送信する。

4. 落札

マネージャエージェントはネットワーク環境を考慮して調整された時間で入札を締め切り、最良の入札を行ったコントラクターエージェントに対して落札メッセージを送信する。

5. 組織情報の交換

組織構成を完了したメンバ同士で組織の親子関係、及び、タスク名、エージェント名、ワークスペース名の情報を交換することで組織化を完了とする。

3.2.2 ルールセットの自動付与機能

ワークスペースに生成されたワークスペースエージェントが組織構成ルールセットを持っていなかった場合、組織構成のやり取りを行うことが出来ない。本機能はワークスペースで生成されたワークスペースエージェントに対して自動的に組織構成ルールセットを付与することで、組織構成に必要な動作知識をエージェントに付与し、組織構成

のやり取りを行えるようにする。

3.2.3 稼働ワークスペースに対するタスクブロードキャスト送信機能

ワークスペース間でメッセージの送受信を行うに当たり、通信時間の削減を検討する必要がある。

従来手法では、1つのワークスペースに複数のワークスペースエージェントが存在する場合、そのワークスペースエージェント1体1体に対して同じタスク通知のメッセージを送信しなければならず、通信時間が増えてしまう。そこで、1体1体のワークスペースエージェントに対してタスク通知のメッセージを送信するのではなく、ワークスペースに対してタスク通知を送信し、そこからブロードキャストを行うことで通信時間の削減を図る(図4)。以下にメッセージの記述例を示す。

:arrival の値を all とすることで全ての環境に対してメッセージを送信することができる。

```
-----タスク通知の記述例-----  
(send :performative "task-announcement"  
:to _broadcast :arrival all  
:content ?dec:task :tid ?tc:id)
```

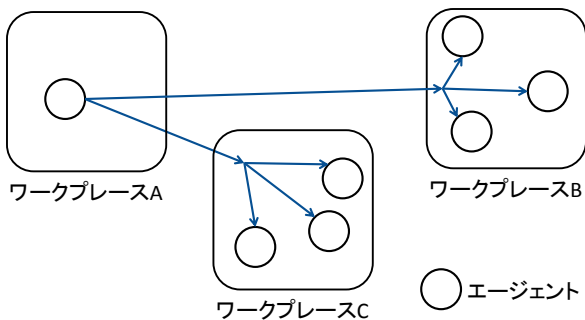


図4: 稼働ワークスペースに対するタスク通知
(ブロードキャスト通信)

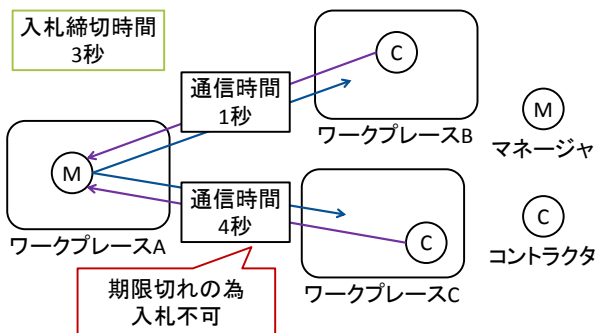


図5: 入札締切時間を一定時間にした場合

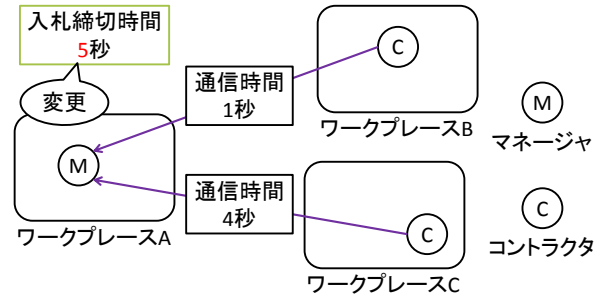


図6: ネットワークを考慮して入札締切時間を
設定した場合

3.2.4 ネットワークを考慮した入札締切時間の自動調整機能

リポジトリ内で行う組織化は単一の環境であるため、通信時間を考慮する必要はなく、入札の締め切りは一定時間で問題が無い。しかし、複数のワークスペースを対象とした組織化では環境が複数存在するため、入札締切時間を一定としてしまうと入札メッセージを送信できなかったにも関わらず、入札が締め切られてしまったため組織に参加できなくなる可能性がある(図5)。そこで、事前に通信時間を計測するエージェントをワークスペース上に生成しておき、他のワークスペースに生成されている同エージェントとの通信を行うことによって通信時間を測定する。その後、結果に基づいた入札締切時間を策定する。これによりネットワークを考慮した入札締切時間の自動調整を実現する(図6)。

4. 実験と評価

4.1 実験環境

実験に用いた計算機環境を以下に示す。

- CPU : Intel Core2 Duo E8500 3.16GHz
- メモリ : 4GB
- OS : Windows 7 Home Premium 64bit
- PC 台数 : 3 台
- ワークスペース数 : 1~5

4.2 実験内容

4.2.1 実験1: 提案機構の動作確認

提案機構を用い、ワークスペース上で生成されたワークスペースエージェントが組織構成に必要な動作知識を自動的にインクルードし、その動作知識を用いて組織構成を行えるかどうかを確認する。

4.2.2 実験2: タスク通知にかかる時間の比較

本機構で実装した稼働ワークスペースに対するタスクブロードキャスト通信から入札処理までにかかる時間(提案手法)と、従来の稼働エージェントに対するタスク通知から入札処理までにかかる時間(既存手法)の比較を行う。

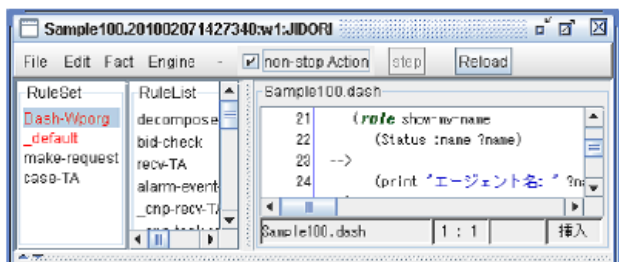


図 7: エージェントの持つルールセット

4.3 実験結果

4.3.1 実験 1

ワークスペース上にワークスペースエージェントが生成された時に、ワークスペース用組織構成ルールセットが自動的にインクルードされたことを確認できた(図 7)。次に、インクルードされた動作知識を用いた組織構成が行えるかどうか確認した。マネージャエージェントがタスク通知を行い、タスクを実現可能なコントラクターエージェントが入札を行い、マネージャエージェントが最良の入札を行ったコントラクターエージェントに対して落札を行い、互いに組織情報を交換することによって、組織構成が行われたことを確認できた。なお、動作確認には複数ワークスペースの同時状況確認が可能な改良版 IDEA(図 8)を利用した。

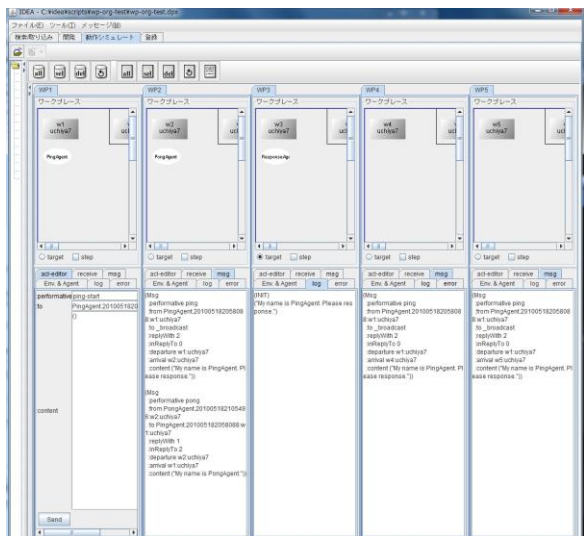


図 8: 改良版 IDEA の動作シミュレート画面

4.3.2 実験 2

従来手法と提案手法について、それぞれのタスク通知から入札処理までの時間の比較を行った。時間の単位を ms とし、結果を表 1 に示す。提案手法はエージェント数が増えるにつれて、従来手法よりも高い性能を示した。また、エージェント数が増えるにつれてメッセージの処理に時間がかかっており、ネットワークを考慮した入札締切時間を

設定する場合、通信時間だけではなく現在稼動しているエージェントの数に対しても考慮が必要であることが判明した。

表 1: 組織化処理(タスク通知～入札)の時間

	エージェント数			
	3	6	15	30
従来手法	383	639	1868	3726
提案手法	428	531	1029	1835

5. おわりに

本研究の目的は、ワークスペース上に生成されたワークスペースエージェントに着目し、複数のワークスペースに生成されたワークスペースエージェントの組織構成手法を提供することで、様々な環境下で柔軟に動作するシステムの実現であった。そこで本研究では、まず提案機構の実現に必要な 4 つの機能について設計を行った。評価実験では、本機構の 1 機能である稼動ワークスペースに対するタスクブロードキャスト通信が、従来手法に比べて効率の良い通信であることを確認し、その有効性を示した。

今後の課題を以下に示す。

- ・ネットワークと稼動エージェント数を考慮した入札締切時間の自動調整

参考文献

- 1) S. Fujita, H. Hara, K. Sugawara, T. Kinoshita, and N. Shiratori, "Agent-Based Design Model of Adaptive Distributed System," *Applied Intelligence*, vol. 9, no. 1, pp. 57-70 (1998).
- 2) DASH- Distributed Agent System based on Hybrid architecture, <http://www.agent-town.com/dash/>
- 3) SAGE -Scalable Fault Tolerant Agent Grooming Environment, <http://sage.niit.edu.pk/>
- 4) JADE -Java Agent DEvelopment Framework, <http://jade.cselt.it/>
- 5) LEAP -Lightweight Extensible Agent Platform, <http://leap.crm-paris.com/>
- 6) OMAS -Open Multi-Agent System-, <http://www.utc.fr/barthes/OMAS/>
- 7) Takahiro Uchiya, Takahide Maemura, Hideki Hara, Kenji Sugawara and Tetsuo Kinoshita, Interactive Design Method of Agent System for Symbiotic Computing, *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, Vol.3, No.1, pp.57-74, (2009).