

ネットワークミドルウェアの運用支援システムの設計

酒井 琢夫[†], 菅沼 拓夫[†], 木下 哲男^{††}, 白鳥 則郎[†]

[†]東北大学 電気通信研究所/情報科学研究科

^{††}東北大学 情報シナジーセンター

〒980-8577 仙台市青葉区片平 2-1-1

Tel: 022-217-5453

我々は、多様な利用者要求やネットワーク環境の変化に柔軟に対応することを目的としたネットワークミドルウェアを導入する新しいネットワークアーキテクチャの研究を行っている。このミドルウェアはマルチエージェントシステムとして構築され、エージェントの持つ自律性・協調性によって柔軟なミドルウェアの構成・運用が可能となる。しかしながら、エージェントが行う協調動作は各エージェントの知識に基づき自律的に行われるので、無駄な協調動作や不安定な挙動によってミドルウェアの運用に悪影響をおよぼすことがあることが指摘されている。そこで本論文では、エージェントが協調動作を行うために交換するメッセージに着目し、その特性とタイミングおよびメッセージ列を分析し、エージェント間の関係、協調動作の状態を検出することでミドルウェアの安定した運用に必要な情報を提供する、運用支援システムの機能を提案する。またその設計と適応例について述べる。

The Design of Operation Support System for Agent-based Network-Middleware

Takuo Sakai[†], Takuo Suganuma[†], Tetsuo Kinoshita^{††} and Norio Shiratori[†]

[†]Research Institute of Electrical Communication / Graduate School of Information Sciences
Tohoku University

^{††}Information Synergy Center, Tohoku University
2-1-1 Katahira, Aoba-ku, Sendai, 980-8577 Tel: +81-22-217-5454

With the static and fixed communication service which the conventional network architecture offers, it can respond to neither various user demands nor change of network environment. Then, we are studying the new architecture which introduces network middleware. This middleware is built by the multi-agent system, and flexible middleware composition and employment of it are attained by its autonomy and cooperativeness. However, cooperative actions which an agent performs are dependent on each agent's capability, are sometimes useless and unstable, and have had a bad influence on employment of middleware. In this paper, we focus on the message exchanged in order that an agent may perform cooperation action. The characteristic, the timing, and the sequence of the exchanged messages are analyzed, and the relation between agents and the function of an operation support system to detect the state of cooperation activities are proposed. Moreover a design and the example of adaptation are stated.

1. はじめに

従来のネットワークアプリケーションの多くは、TCP/UDP などが提供するトランスポート通信サービスを静的/固定的に用いて構成されて

いる。しかしながらトランスポート通信サービスを直接用いて利用する場合、多様な利用者要求やネットワーク環境の変化には追従することが困難である。この問題を解決するためには、

ソフトウェア工学的、あるいは利用者指向的な観点を考慮した新たなネットワークアーキテクチャを構成する必要がある。

我々は上記の問題を含む、IP ネットワークなどのサービス機能層である論理ネットワーク層とアプリケーション層の間の機能のギャップにより生じる様々な問題を解決するため、それらの中間にネットワークミドルウェアを導入する新たなネットワークアーキテクチャに関する研究を行っている[1]。ミドルウェアはアプリケーションから共通的に利用される、バイナリやテキストファイル、品質調整を伴う動画、静止画、音声メディアなどの送受信機能をコンポーネントとしてアプリケーションに提供できなければならない。加えて論理ネットワーク層の変動を柔軟に吸収するためには、性能的な調整、すなわちミドルウェアの機能構成を変更し大きな変動に適応することも求められる。そこで、ミドルウェアが持つべき機能を小規模な機能部品に分割し、状況に応じて組み合わせを変化させることによって、アプリケーションやネットワーク環境の違いに応じた機能調整を実現する。すなわち機能部品をエージェントとして実現し、マルチエージェントシステムでミドルウェアを構築することで、組織構成能力・再構成能力、協調的調整能力による、柔軟なミドルウェア構成・運用が可能となる。

しかしながら、ミドルウェアとして最適な機能を提供するためにエージェントが行う協調動

作は、各エージェントの知識や自律性に依存しており、無駄な協調動作や不安定な挙動をとる場合がある。そのため、ミドルウェアの運用において処理性能や安定性の面で悪影響をおよぼしていることが指摘されている。これらの問題点を解決するには、エージェント間の依存関係や協調動作を観測し、その結果に基づきエージェントの協調動作を効率化するための運用支援システムが必要である。

本研究では、エージェントが協調動作を行うために交換するメッセージに着目し、その特性とタイミングおよびメッセージ列を分析し、エージェント間の関係、協調動作の状態を検出する手法、およびその情報を用いてエージェントの効率的な協調動作を支援する枠組みを提案する。また上記提案を実現するための運用支援システムの設計と実装を行っている。本論文では、そのうちのメッセージが持つ情報を分析しエージェント間の関係、協調動作の状態を検出する機能の設計と適応例について述べる。

以下、まず2章で、エージェント指向技術を用いたネットワークミドルウェアの概要を述べる。次に3章で、本研究の提案である運用支援システムの概要および、そのうちの検出機能の提案を行い、プロトタイプシステムの設計について述べる。4章でこのシステムの適応例と実験について述べる。

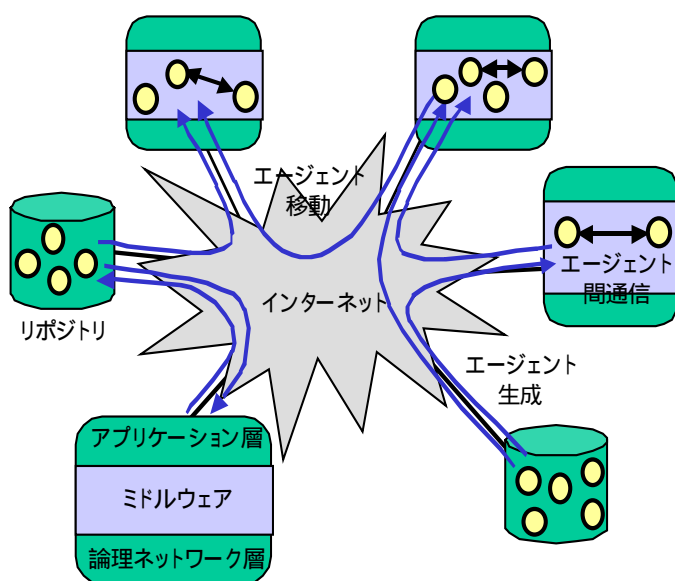


図1. エージェント指向ネットワークミドルウェア環境モデル

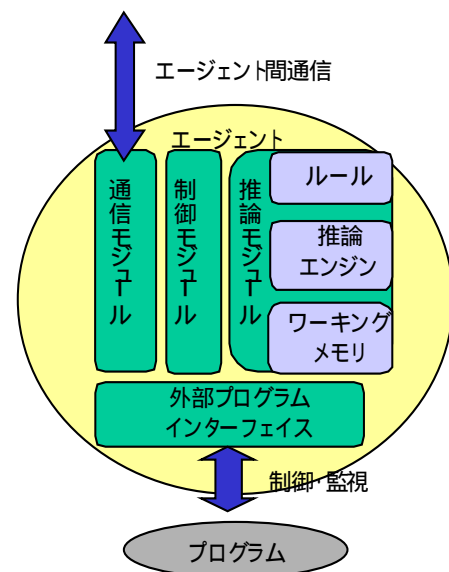


図2. エージェントアーキテクチャ

2. エージェント指向ネットワークミドルウェア

2.1 エージェント指向ネットワークミドルウェアの概要

本論文で対象としている環境モデルを図1に示す。この環境モデルでは、インターネットなどの広域分散ネットワークにおいてエージェント指向ネットワークミドルウェアを実装した多数のコンピュータが存在しており、これらのミドルウェアは各々の論理ネットワーク層とアプリケーションに応じた機能を構成し提供している。ミドルウェアの機能を構成しているマルチエージェントシステムは、ネットワーク上に存在するいくつかのリポジトリに格納されているエージェント群がアプリケーションの要求に応じて組織を構成し生成され、ミドルウェアとして動作を開始する。

図2にエージェントのアーキテクチャを示す。本論文においてのエージェントは通信モジュール、制御モジュール、推論モジュールおよび外部プログラムとのインターフェイスから構成される。通信モジュールは、他のエージェントとメッセージを交換する通信機能を提供する。推論モジュールは、ルールおよびワーキングメモリ内の状態を基に推論エンジンを使って推論が行われる。外部プログラムインターフェイスは、様々な資源（デバイス）を直接制御する機能を提供する。制御モジュールは、上記3つの機能を制御する機能である。エージェントは、これ

らの機能によって協調動作を行い、資源などの状況に応じて自律的にミドルウェアで提供している機能を調整する。

2.2 問題分析

以上の環境モデルにおいて動作するマルチエージェントシステムには以下の問題がある。

(P1) 反対な性質の協調動作を繰り返す振動現象が起こる

(P2) 連続的な状況変化に対し柔軟な協調動作ができない

(P3) 制御不能な状況の変化に対し協調動作による調整を行う

(P1)はたとえば、軽微なパラメータ調整などの要求が発生した場合、資源獲得において獲得と開放を繰り返す振動が起こる問題である。これは、1セットの協調動作の間の関連性を認識する知識をエージェントが保持していない場合に自分（エージェント）が振動していることを認識できないために起こる。ここで、1セットの協調動作とは、様々な外因を監視するエージェント、例えば利用者要求の監視エージェントや資源状況の監視エージェントなどから発生する要求メッセージまたは状況変化メッセージから始まり、発生した要求や状況変化に基づき関連するエージェントが協調し要求や状況変化に対し最適な調整を行い安定状態になるまでのエージェントの動作系列を示す。

(P2)はたとえば、多数のエージェントから同

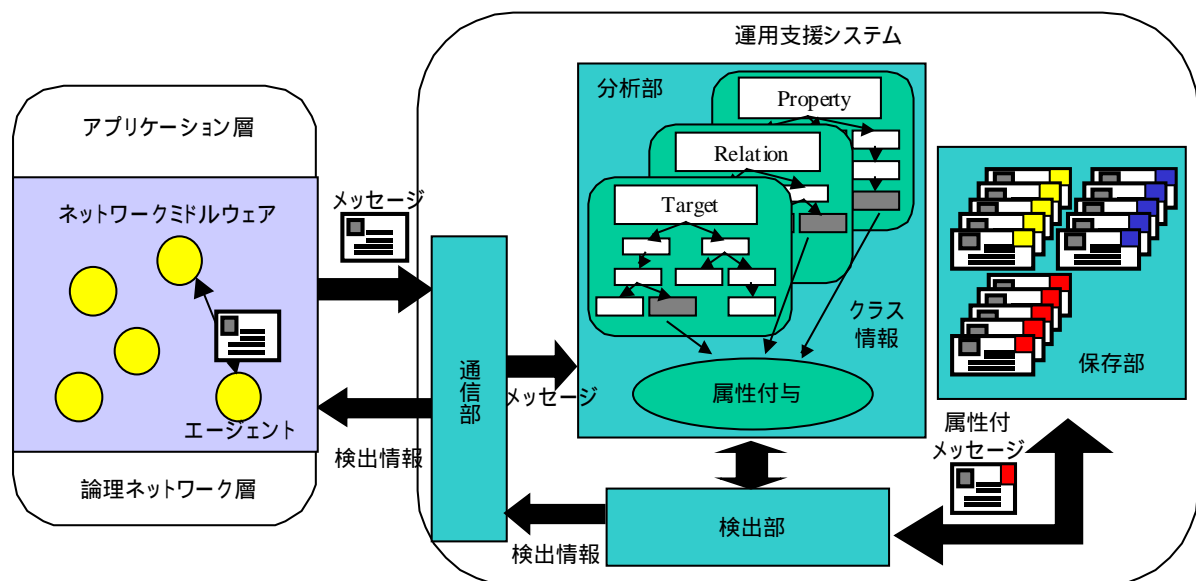


図3. ネットワークミドルウェア運用支援システムの検出機能のアーキテクチャ

じ資源に対し短期間に獲得要求が発生した場合に、現在行われている協調動作から順次継続して行われるため、発生した要求の関係を整理した後に対応することが出来ない問題である。これは、エージェントが状況変化ごとに協調動作による調整を行うため起こる。

(P3)はたとえば、メディア通信の通信先に起因するサービス品質の低下に対する要求メッセージが発生した場合に、直接的に関連性のうすい受信側の資源獲得などを目的とした協調動作を行ってしまう問題である。これは直接メッセージを交換していない相手の状態を、関係が有るものとしてエージェントが認識できないために生ずる。

これらに共通する原因として、直接メッセージを交換する通信の有無に関わらず、エージェントの協調動作を観測・分析し、分析した情報を用いた効率的な協調動作を支援するシステムが欠如していることが考えられる。

以上に述べた問題を解決するためには、エージェント間の関係、協調動作の状態を検出する機能および、検出した関係・状態を用いてエージェントの協調動作を効率的にする運用支援システムが必要である。

3. ネットワークミドルウェア運用支援システム

図3に運用支援システムの検出機能のアーキテクチャを示す。

検出機能では、エージェントが協調動作を行うために交換するメッセージに着目し、その特性とタイミングおよびメッセージ列を分析し、エージェント間の関係、協調動作の状態を検出することが可能である。

3.1 検出機能アーキテクチャ

検出機能アーキテクチャの各部の詳細を以下に示す。

通信部はネットワークを介し、監視したいエージェントが動作するミドルウェアのメッセージパスからエージェント間で交換されているメッセージを取得する。

分析部は個々のメッセージに対し取得した順にクラス分けを行う。クラス分けは、複数の項目で定義された分析クラス木を使い、木ごとに一致するクラスノードへマッピングする。このとき項目は、メッセージヘッダおよびメッセー

ジコンテンツなどの情報を基に構成する。ヘッダの中にはFrom・TO・Performativeなどの情報が収められている。分析されたメッセージには各項目でマッチしたクラスの属性を付与する。

検出部はあらかじめ定められた検出条件である検出条件式をもとに分析部でメッセージに付与した属性(式)とのマッチングを行う。検出は単独のメッセージに対して行われる場合と、タイムスタンプと保存部にあるメッセージログとを併せたパターンを検出条件として検出を行う場合とがある。

保存部は分析クラス木による属性(式)およびタイムスタンプを付けメッセージログとして保存する。

これらの機能によってエージェント間で交換されるメッセージの分析・クラス分けが可能となり、協調動作の状態やエージェント間の関係を検出することが可能となる。

3.2 分析クラス木

分析クラス木は複数のクラス木で構成される。主なものとして挙げられる3つのクラス木をさきほどの図3に示す。Targetクラス木は、そのメッセージが何に対して発生しているか特定する分析クラス木である。Relationクラス木は、そのメッセージがどのエージェント間で発生しているのかを特定する分析クラス木である。Propertyクラス木は、そのメッセージがどういった性質を持つものであるかを特定する分析クラス木である。これらの分析クラス木ではルー

| 検 出 条 件 式 |
|--|
| <pre> C O N D I T I O N 1 M e s s a g e C o n d i t i o n T a r g e t : c o n d i t i o n _ 1 R e l a t i o n : c o n d i t i o n _ 2 P r o p e r t y : c o n d i t i o n _ 3 . . . C o n t e x t : c o n d i t i o n _ X M e s s a g e P a t e r n C o n d i t i o n . . . A c t i o n . . . </pre> |
| <pre> </pre> |

図 4 . 検 出 条 件 式 の 例

トを除く全ノードにおいて対応するクラス（条件式）が存在する。

メッセージを取得するとクラス木ごとにクラスの検索が始まりそれぞれのルートノードから一致するノードへ順次遷移していくことで各クラス木に1つの終端クラスが特定できる。また、その経路からメッセージに付与する属性（式）を導き出せる。

このように複数のクラス木を用いることで、発生したメッセージを多角的かつ詳細に分析できる。

3.3 検出条件式

図4に検出条件式の概要として一例を示す。ひとつの検出条件であるCONDITIONには、メッセージ単体の属性の条件が記述されている Message Condition 部、タイムスタンプおよびメッセージ履歴とのパターンマッチング条件が記述されている Message Pattern Condition 部、および上記2つの条件が一致したときに実行される、すなわちメッセージまたはメッセージパターンが検出したことに対するアクションが記述されている Action 部から構成される。

検出条件は、検出したいエージェント間の関係、協調動作の状態に合わせて定義することで、無駄な協調動作や不安定な挙動を正確に検出することが出来る。

4. 実装と評価

4.1 ADIPS フレームワークによる実装

3章で提案した検出機能のプロトタイプシステムをリポジトリ型マルチエージェントフレームワークの事例である ADIPS フレームワーク [2][3]において実装した。この ADIPS フレームワークにおいても、2.2章で述べた問題点が指摘されている。

4.2 やわらかいビデオ会議システムへのプロトタイプシステムの適用

プロトタイプシステムによる評価実験用のアプリケーションとしてやわらかいビデオ会議システム (Flexible Video Conference System : 以下 FVCS と略) [4][5]を用いる。FVCS は ADIPS フレームワークによって提供されるマルチエージェントシステムの特徴である、自立性、協調性、組織構成能力などを利用し、利用者指向のサービスを提供するビデオ会議システムである。この FVCS において利用者要求を満たすために、柔軟なパラメータ調整、各種資源の獲得、広域分散ネットワークを介した通信が行われているため 2.2章で挙げている問題点が存在する。よって検出機能の有効性を評価に適していると考えられる。

図4に FVCS における分類・検出のための分析クラス木の例を示す。ここでは Target のクラス木を示す。

分析クラス木は、上位クラスほど抽象度が高く、下位クラスほどビデオ会議ドメインに特化

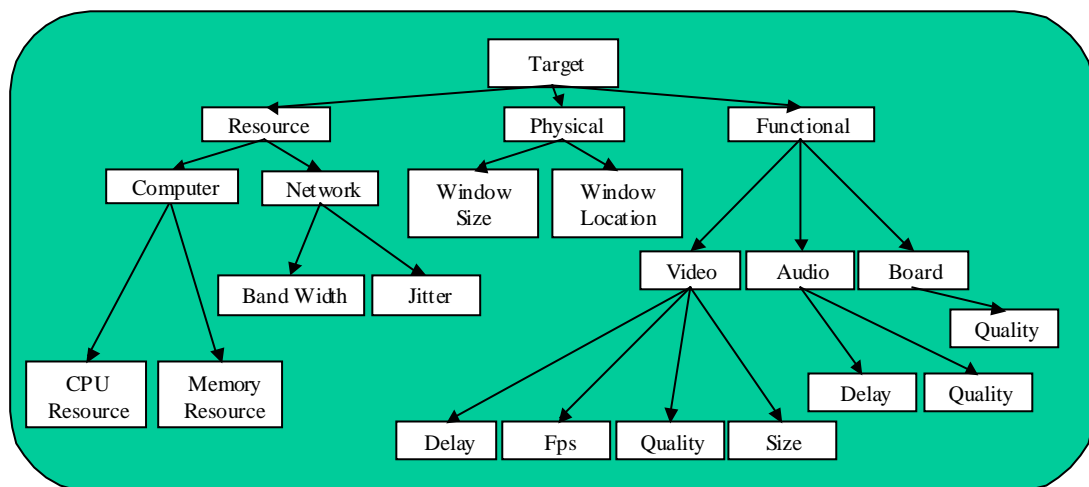


図5. やわらかいビデオ会議システムにおける分析クラス木の例 (Target)

したクラスが現れる。Target クラス木の下位クラスには資源に関する分類項目を表現する Resource クラス、機能性に関する分類項目を表現する Functional クラス、物理的な制約条件に関する分類項目を表現する Physical クラスがある。これらのクラスは、様々なドメインでのメッセージ分類に共通で利用できるクラスであり、これらのサブクラスを定義してゆくことで、特定のドメインに特化した分析クラス木を定義することが可能である。

FCVS の例では、Resource クラスのサブクラスとして、計算機資源に関する Computer Resource クラス、ネットワーク資源に関する Network Resource クラスを定義している。それらのサブクラスとしてさらに具体的なクラスが定義される。たとえば Computer Resource のサブクラスとして、CPU 資源に関する CPU Resource クラス、メモリ資源に関する Memory Resource クラスが定義されている。同様に Function クラスのサブクラスとして、映像機能、音声機能、共有ボードに対するクラスが定義され、さらにそれらの QoS 要求に関する具体的なサブクラスが定義されている。

4.3 やわらかいビデオ会議システムにおけるエージェントの状態検出

実験では、User-A と User-B が FCVS を利用して LAN を介しビデオ会議を行う。このとき、以下に挙げる 3 つの状態の検出を目的とする。

(C1) 周期的にネットワークの帯域の空きを変化させることで、状況に応じて起こる協調動作の振動の検出

(C2) User-A, User-B のそれぞれで画質変更を要求することで起こる連続した類似の協調動作の検出

(C3) User-B 側へ負荷かけることにより起こる画質の低下に対し、User-A 側で回復をするために行う無駄な協調動作の検出

以上の実験を行い、検出条件式の Action 項目で設定する検出情報を検証することで本研究での提案の有効性を示すことができる。

5. まとめ

エージェント指向ネットワークミドルウェアにおける利用者要求や環境変化に応じた適応型サービスを提供するようなアプリケーションでは、時々刻々と状況の変化が起こる。このよう

な環境においてマルチエージェントシステムによってサービスを提供するためには、必要以上に発生してしまうエージェントの協調動作を抑制し効率的なエージェント間協調を行う仕組みが不足している。

本研究では、エージェント指向ミドルウェアにおいて、エージェント間で交換されるメッセージを特性やタイミングを用いて分類し、その情報からエージェント間の関係を検出することにより効率的な協調動作を実現することを目的としている。この目的を実現するための検出機能の設計と、分析クラス木を用いた検出法について述べた。また、これらのやわらかいビデオ会議システムへの適用について述べた。

[参考文献]

- [1] T.Suganuma, T.Kinoshita, and N.shiratori, "Flexible network layer in dynamic networking architecture", Proc of The 1st International Workshop on Flexible Networking and Cooperative Distributed Agents (FNDA2000), 473-478, 2000
- [2] 藤田茂, 菅原研次, 木下哲男, 白鳥則郎, "分散処理システムのエージェント指向アーキテクチャ", 情報処理学会論文誌, Vol.37, No.5, pp.840-852, 1996.
- [3] S.Fujita, H.Hara, K.Sugawara, T.Kinoshita, and N.Shiratori, "Agent-Based Design Model of Adaptive Distributed Systems", Applied Intelligence, Vol.9, No.1, pp.57-70, 1998.
- [4] 菅沼拓夫, 藤田茂, 菅原研次, 木下哲男, 白鳥則郎, "マルチエージェントシステムに基づくやわらかいビデオ会議システムの設計と実装", 情報処理学会論文誌, Vol.38, No.6, pp.1214-1224, 1997
- [5] Suganuma T, etc, "Flexible Videoconference System based on ADIPS Framework", Proc of the 3rd International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM98), pp.83-100, 1998
- [6] Mark Klein, Stephen C-Y. Lu, "Conflict Resolution In cooperative Design", The International Journal for Artificial Intelligence In Engineering, Vol.4, No.4, pp.168-180, 1990