

エージェントによる異種分散環境における通信の支援について

4P-2

鈴木 満 唐橋 拓史 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

1 はじめに

インターネットの普及にともないネットワーク接続された異機種分散系上の計算機資源の共有、利用、管理や、その上でのユーザ間コミュニケーション等の実現のために、マルチエージェントシステムが提案されている。マルチエージェントシステムとは、複数のエージェントが互いにコミュニケーションを行い、協調動作する事で1つのシステムを構成するものである。エージェントとは自律的な計算主体であり、計算機資源利用に関する知識や制御機構を持つ。このようなエージェントを用いてシステムを構成することにより異機種間での差異を吸収し、また、柔軟なシステム運用が可能となる。

しかしながら、個々のエージェントは1つ以上のプロセスにより実現され、マルチエージェントシステムの実装にはマルチプロセスが前提となっている。そのため、P/C等マルチタスクをサポートしないOSによる小規模な計算機システムにおいては、その実装がそのままの形では困難なものとなっている。小規模計算機システム上の資源をマルチエージェントシステムから利用するためには、対象となる計算機システムやアプリケーションに特化したエージェントを用意することが必要であり、またその計算機資源の管理の一部を計算機システム外部から行なうための機構が必要となる。

本稿では、マルチエージェントシステム上のエージェントから小規模計算機システム上の資源を利用するために、GWA-ACAモデルを提案する。GWA-ACAモデルとは小規模計算機システム上にアプリケーションコントロール機構を設け、マルチエージェントシステム上からその計算機システム上の資源を利用できるようにするものである。また、本モデルをマルチエージェントシステムADIPS上に実装し、その有効性を示す。

2 GWA-ACAモデル

本章ではGWA-ACAモデルについて述べ、その構成要素であるGWAとACAのメカニズムについて議論する。

2.1 GWA-ACA

図1に我々が提案するGWA-ACAモデルの全体を示す。マルチエージェントシステム側にGateWay Agent

(GWA)を配置し、利用対象となる小規模計算機システム側にApplication Control Agent(ACA)を配置する。GWAとACAがコミュニケーションを行なうことにより情報を交換し、アプリケーションの利用を行なう。

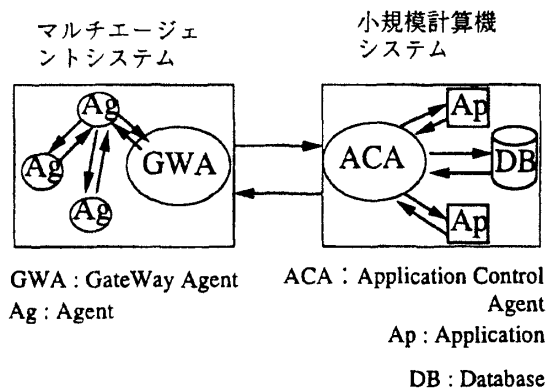


図1: GWA-ACA

GWAはマルチエージェントシステムと対象となる計算機システムとの接続を行う機構である。GWAは、マルチエージェントシステム側からのアプリケーション利用要求を取得し、そのためのアプリケーション管理、利用情報をACAに送信する。また、その利用結果をACAから取得し、マルチエージェントシステム側へ送信する。

ACAは、GWAを通してマルチエージェントシステム側から得られた情報をもとに、計算機システム上のアプリケーションを直接制御し、その利用を行なう。またアプリケーションからの情報を取得し、利用結果として必要な情報をGWAに送信する。

GWAとACAそれぞれのメカニズムについては、事項以降で詳しく述べる。

2.2 GWA

図2にGWAのモデルを示す。CMはマルチエージェントシステムにおける協調機構であり、他のエージェントとのコミュニケーションを行なう。DKはGWAの行動に関する知識及び、利用対象となる計算機システム上のアプリケーションに関する知識を持つ。CF/hはACAとの通信機構であり、メッセージ交換によるコミュニケーションを行なう。PMはGWAの動作エンジンであり、CM、CF/hからの入力に応じて、DKの知識に基づきCM、CF/hの制御を行なう。

Agent-based communication support between heterogeneous distributed computer environment

Mitsuru Suzuki, Takuji Karahashi, Tetsuo Kinoshita and Norio Shiratori

Research Institute of Electrical Communication/Graduate School of Information Sciences, Tohoku Univ.

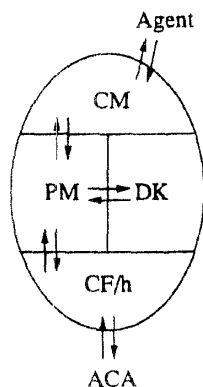


図 2: GateWay Agent

GWA の動作は、まず CM が他のエージェントからアプリケーション利用要求を受信し、PM にわたす。PM は DK を参照し、利用対象となるアプリケーションを選択する。この時、必要であれば ACA との通信により、必要なアプリケーションを決定する。その結果、CF/h を介して、利用するアプリケーション名とその利用手続きを ACA に送信する。また、ACA からアプリケーション利用結果を取得し、CM を介して要求元のエージェントに送信する。

2.3 ACA

図 3 に ACA のモデルを示す。AK は利用可能なアプリケーションとその制御に関する知識を持つ。CF/h は GWA との通信機構である。AC はアプリケーションの起動、制御、及び実行結果の取得を行なう。PM は ACA の動作エンジンであり、AC、CF/h をからの入力に応じ、AK の知識に基づき AC、CF/h の制御を行なう。

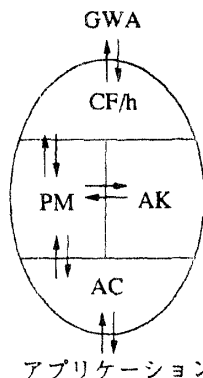


図 3: Application Control Agent

ACA の動作は、まず、CF/h を介して GWA からアプリケーション情報を受けとり、PM にわたす。PM はその情報と AK をもとに AC を制御し、アプリケーションの起動及び、利用を行なう。また、その結果取得された情報を CF/h を介して GWA に送信する。

3 実装、評価

本章では GWA-ACA モデルの実装について述べ、実装されたシステムを用い、モデルの評価を行なう。

3.1 実装環境

今回、提案したモデルに従い GWA 及び ACA の実装を行なった。実装環境としては、マルチエージェントシステムとして ADIPS (Agent-based Distributed Information Processing System)、対象となる計算機システムとしてマッキントッシュを用いた。ADIPS とは、分散処理システム上のサービスを複数の自律的なエージェントによって構成するシステムである。その特徴として、「利用者要求駆動で自律的にシステムを構成する」、「障害などのイベント駆動で自律的にシステムの再構成を行う」ことがあげられる。

GWA の実装は、ADIPS におけるエージェント記述言語である ADIPS/L を用いた。また CF/h の拡張部分を Tcl/Tk を用い記述した。マッキントッシュ側の ACA の枠組をマッキントッシュ上の Tcl/Tk を用いて記述し、CF/h、AC、AK、PM を Apple Script を用いて実装した。GWA、ACA 間の通信には、TCP/IP のソケットを利用した。

3.2 評価

GWA-ACA モデルの有効性を示すために、実装した GWA、ACA を用い、評価実験を行なった。今回想定した実験は、マッキントッシュ上のカード型データベースを ADIPS 環境下のデータベースの一つとして検索を行なうものである。実験の結果、他のデータベースと同様にデータの検索、引き出しを行なえることが確認された。以上により、本モデルを用いることでマルチエージェントシステムが実装できないような小規模計算機システム上の資源の有効的な利用が可能となると言える。

4 まとめ

本稿では、マルチエージェントシステムからの小規模計算機システム上の資源の有効利用の手法として、GWA-ACA モデルを提案し、実装、評価実験をおこなった。その結果、ADIPS からマッキントッシュ上のアプリケーション利用を行う事ができた。これによりマルチエージェントシステムが実装できない計算機システム上の資源利用における本モデルの有効性を示した。

今後、マッキントッシュ上から ADIPS 環境の資源の利用や、他の OS の資源の利用などを行なう予定である。

参考文献

- [1] Shiratori, N., Sugawara, K., Kinoshita, T., Goutam, C.: Flexible Networks: Basic Concepts and Architecture, *IEICE Trans. Commun.*, Vol. E77-B, No. 11, pp. 1287-1294 (1994)
- [2] 藤田茂, 菅原研次, 木下哲男, 白鳥則朗: 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).