マルチメディア通信と分散処理　82－2
（１９９７．４．２４）

# ADIPSフレームワークを用いた エージェント指向システムの実装

藤田茂[†]　嵯峨毅[††]　菅原研次[†]　木下哲男[†††]　白鳥則郎[†††]

[†]千葉工業大学情報工学科　[††]株式会社サイエンティア　[†††]東北大学電気通信研究所

概要

本稿では，分散システムを構成する要素である計算機プロセスを，その計算機プロセスの設計知識/運用知識を用いて自律的に動作するエージェントにより管理/制御するADIPSフレームワークの概要を述べ，このADIPSフレームワークでのエージェント間プロトコルについて述べる．ADIPSフレームワークは現在C++，Tclによる実装と，Javaによる2種類の実装が行われており，この実装についての報告を行う．またC++，TclによるADIPSフレームワークの実験例として，2者間でのビデオ会議システムをADIPSフレームワークを用いて構築した．

# An Implementation of Agent oriented Distributed System based on ADIPS framework

Shigeru FUJITA[†], Takeshi SAGA[††], Kenji SUGAWARA[†],
Tetsuo KINOSHITA[†††], Norio SHIRATORI[†††]

[†] Department of Computer Science, Chiba Instibute of Technology
[††] Scientia Co., Ltd.
[†††] Research Institute of Electrical Communication, Tohoku University

Abstract

In this paper, first, we discuss an ADIPS: Agent-based Distributed Information Processing System framework and its protocols among agents which control and manage computational processes which is a part of distributed systems. There are two experimental systems, the first one is implemented by C++ and tcl. Next, another one is implemented by Java. An example system of the video conference system was developed by ADIPS framework with C++ and tcl.

# 1 はじめに

高性能パーソナルコンピュータ/ワークステーションの普及と，ネットワーク技術の発達に伴って，様々な利用者要求を充足するためのプログラム (アプリケーションプログラム) が開発利用されてきている．このため，アプリケーションプログラムが動作するために要求する計算機資源は増加する一方であり，複数のアプリケーションが利用するネットワークへの資源要求も増加することとなり，ネットワーク資源要求の競合発生/資源確保が困難であるという問題がある．ネットワーク資源確保のためプロトコルとして，RSVP[1] のような資源確保プロトコルが提案/研究されているが，資源確保が行えない状況下でも，柔軟に利用者要求に対応する必要がある．

また，ワークステーションのように複数の計算機プロセス/利用者が存在する環境では，複数のアプリケーションにより利用者要求を充足する場合がある．このような場合，複数のアプリケーション間で計算機資源の利用状態の調節が必要であるが，現在の OS が提供している資源確保を中心とする手法では実現が困難であり，アプリケーション間の情報交換のための手法も提供されていない．

我々はこれまでに，“やわらかいネットワーク”[2] 実現のための一手法として，分散システムを利用者要求に基づいて構成する ADIPS フレームワークを提案してきた [4]．この ADIPS フレームワークでは，利用者要求に基づいて分散システムを構築する，他の分散システムとの間で計算機資源に関する交渉を行うという二つの機能を定義利用することで，利用者要求をできるかぎり充足し，上述の 2 つの問題点を解決している．

2節では，ADIPS フレームワークの概略を述べ，3節では，エージェント間のプロトコルで用いられるプロトコルについて述べる．4節では，C++, tcl を用いた試作環境とこの試作環境を用いて実験したテレビ会議システムについて述べる．5節では本稿のまとめと，今後の課題について述べる．

# 2 ADIPS フレームワーク

ADIPS フレームワークでは，分散システムの構成要素はそれぞれ自律的なプログラムとして振る舞う“エージェント”である．このエージェントに対する知識記述を行うことで，分散システムの各構成要素を実現する．分散システムの構成要素の開発者，すなわちエージェントの知識記述者によって実現されたエージェントは，リポジトリと呼ばれるエージェント保管庫へエージェントを登録する．この登録されたエージェントは，クラスエージェントと呼ばれる．利用者から伝達される分散処理システム要求により，リポジトリ内部のクラスエージェントは記述された知識により，要求が実現できるか否かを判断し，要求を実現できる場合には，入札を行う．このとき，再帰的にリポジトリに存在する他のクラスエージェントに対して要求を伝達して利用者要求を充足する分散システムを構成する．この後，利用者からの落札通知により落札を受けたエージェントは具体的な処理を実現するために，ネットワーク上に存在するエージェントの実行環境上に自分自身のインスタンスを生成する．ここまでの過程で利用されるエージェント間のプロトコルは契約ネットプロトコル [3] を一部変更した拡張コントラクトプロトコルを用いている．この拡張契約ネットプロトコルについては，3.1節で述べる．

生成されたインスタンスエージェントは，利用者要求を充足するための処理を実現する．この処理は，エージェント自身の機能を利用する，外部の計算機プロセスを利用する，他のエージェントを利用することによって実現される．他のエージェントの利用については，3.3節で述べる交渉プロトコルをもって行う．

利用者要求に基づいて生成されたインスタンスエージェントは，計算機環境の変動 (ex, ネットワークの利用可能な帯域の増減，利用可能な CPU 時間の増減)，利用者要求の変化に対して利用者要求を充足するための動作を行う．この動作には，インスタンスエージェントが利用している他のエージェントの制御が含まれる．この利用とは，リポジトリ内部での拡張契約ネットプロトコルに基づいて，分散システムの構成が行われたときの関係，インスタンスエージェント間での拡張契約ネットプロトコルによる契約関係を示す．

この他，必要に応じて他のインスタンスエージェントに対して動作要求を伝達する，あるいは，他のインスタンスエージェントから動作要求を伝達される場合がある．この動作要求の伝達から，双方の合意が成立するまでの状態を，ADIPS フレームワー

クでは，交渉状態と呼んでいる．この交渉状態では，エージェントは自己が保持する利用者要求を充足することを優先するが，利用者要求充足に影響しない範囲で，他のエージェントの動作要求に基づいた動作をする．この動作要求の伝達により，複数の利用者要求を実現している複数の分散システム間で，ネットワーク資源の競合解消を実現する．また，交渉状態ではエージェントに知識記述があれば，その知識に基づいて，競合を解消するための代案の提示を行うことで，エージェント間の自律的な競合解消を実現する．

利用者要求に基づいて生成されたインスタンスエージェントの組織である，分散システムは，利用者要求の充足 (すなわち，利用者からの利用終了宣言) をもって，すべての動作を停止し，自身を動作環境から消滅させる．

図 1に ADIPS フレームワークの概念図を示す．

# 3 ADIPS エージェント間プロトコル

## 3.1 拡張契約ネットプロトコル

拡張契約ネットプロトコルは，契約ネットプロトコルがもつ分散問題解決の枠組を利用して，動的に分散処理システムを構成/再構成するためのプロトコルである．表 1にメッセージのヘッダと意味を，図 2に状態遷移を示す．

## 3.2 組織制御プロトコル

組織制御プロトコルは，拡張契約ネットプロトコルによって契約関係が成立した組織間での情報伝達/制御のためのプロトコルである．表 2にメッセージのヘッダと意味を示す．組織制御プロトコルは，拡張契約ネットプロトコルにより契約を結んだエージェント間での通信であり，一つのメッセージには，一つのメッセージが対応して終了する．`report`, `dissolve` には対応するメッセージがなく，`direction` に対しては，`acceptance-direction`, `refusal-direction` のいずれかのメッセージが対応する．

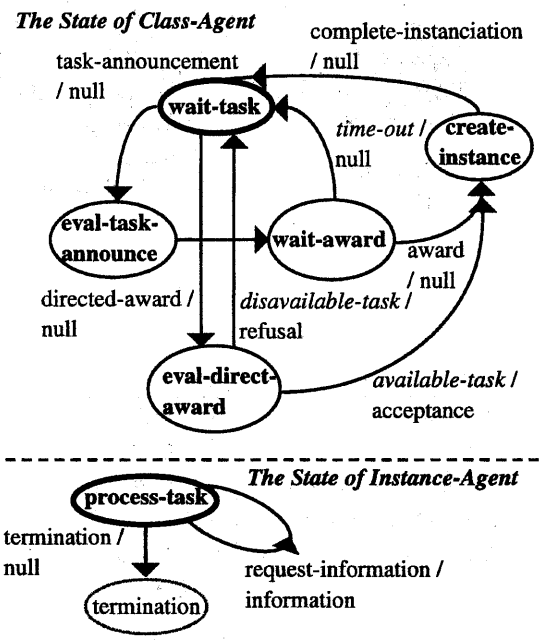


図 2 拡張契約ネットプロトコル状態遷移図

| メッセージヘッダ | 意味 |
|---|---|
| `task-announcement` | タスク通知 |
| `bid` | 入札通知 |
| `award` | 落札通知 |
| `directed-award` | 直接落札通知 |
| `acceptance` | 直接落札通知への受諾応答 |
| `refusal` | 直接落札通知への拒否応答 |
| `request-information` | 情報要求 |
| `information` | 情報伝達 |
| `termination` | 終了消滅指示 |

表 1 拡張契約ネットプロトコルメッセージヘッダ

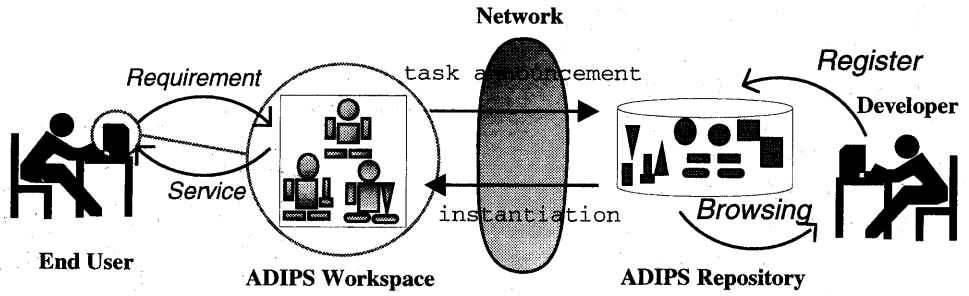


図 1　ADIPS フレームワークの概念図

| メッセージヘッダ | 意味 |
|---|---|
| report | 結果報告 |
| direction | 動作指示 |
| acceptance-direction | 動作指示受諾 |
| refusal-direction | 動作指示拒絶 |
| dissolve | 致命的エラーによる消滅通知 |

表 2　組織制御プロトコル

### 3.3　交渉プロトコル

交渉プロトコルは，動作環境上に存在するインスタンスエージェント間で他のエージェントに対して動作要求を行う，あるいは他のエージェントからの動作要求に対する応答を実現するために利用されるプロトコルである．表 3にメッセージのヘッダと意味を示す．交渉プロトコルのシーケンスは，request-action より開始され，このメッセージの受け手が，acceptance-request-action もしくは，refusal-request-action を応答することによって終了する．request-action に対して，counter-request-action が返答された場合，これに対しては，

- acceptance-request-action
- refusal-request-action
- counter-request-action

のいずれかによる応答を行う．

交渉状態の終了は

- acceptance-request-action
- refusal-request-action

のいずれかが，どちらか一方のエージェントにより交渉相手のエージェントに伝達されることによる．

| メッセージヘッダ | 意味 |
|---|---|
| request-action | 動作要求 |
| acceptance-request-action | 動作要求受諾 |
| refusal-request-action | 動作要求拒絶 |
| counter-request-action | 動作要求への代案提案 |

表 3　交渉プロトコルメッセージヘッダ

## 4　ADIPS フレームワークの試作

### 4.1　試作システム

これまで述べてきたADIPS フレームワークの定義に基づいて，ADIPS フレームワークの試作を，C++ および Tcl による実装，Java による実装の 2 通りについて行っている．

C++および tcl により実装が行われたシステムでは，そのサイズはリポジトリ部分の実装に関して，C++部分が，およそ 15000 行，tcl 部分がおよそ 5000 行である．また実行環境の実装に関して，C++部分が，およそ 17000 行，tcl 部分がおよそ 5000 行である．

Java により実装が行われたシステムでは，そのサイズはリポジトリ部分/実行環境の実装に関して，それぞれ，およそ 6000 行である．

図3にエージェント開発時に利用可能な，エージェント記述エディタの様子を示す．このエディタではエージェントに対して知識記述を行う際に重要となる，他のエージェント記述者によって定義されている機能を表現する語彙の参照，エージェントの検索を行うことにより，エージェントの知識記述者を支援している．

図4にエージェントの記述例を示す．

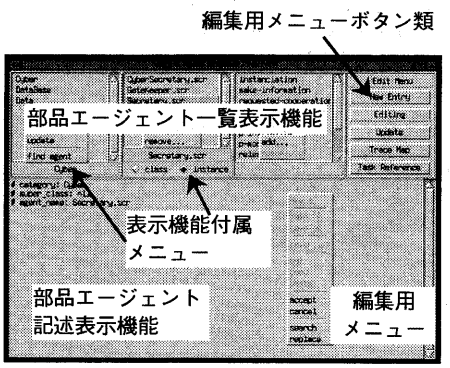


図3　エージェント記述エディタ画面例

## 4.2　ビデオ会議システム

C++, Tcl により実装された，ADIPS フレームワークの実験として，2者間でのビデオ会議システムを行った．このときのエージェントの構成を図5に示す．このとき，ネットワークトラフィックの増加によるパケットロストによって，利用者要求から著しく逸脱する．これに対して，それぞれの利用者要求を充足するために，エージェントが動作し，組織再構成を行った．これと同様の実験を Java により実装された ADIPS フレームワークにより行う予定である．

# 5　まとめ

本稿では，利用者要求に基づいて分散システムを構成し，動作環境の変化に対しても利用者要求を充足し続けることを目的とした ADIPS フレームワークについて述べた．この ADIPS フレームワークにより実現されたシステムは，利用者要求を充足する

```
( specification
    ( identification
          agent-name Organizer
          agent-description
          $ Organizer tries to make Muler and LSist get organized. But
when LSist
          reports the existence of the html-file, Organizer releases
this organization
          and make another one.$
    )

    ( organization-temporary
          ( o-eval-task
             /* Accept Directed-Award from AWS immediately. */
             task-specification (cond (== 1 1))
             id 0
          )

          ( make-organization
             /* The first attempt to establish the organization. */
             /* Muler and LSist will get organized. */
             id 0
                   task-announcement (msg
                       (departure &A_General(MyHost)&)
                       (arrival AAR)
                       (performative Task-Announcement)
                       (from &A_General(MyName)&)
                       (to Muler)
                       (reply-with &DK_MakeReplyWith&)
                       (expiration-time 30)
                       (contents ((require-knowledge ((== 1 1)))))
                   )
                   task-announcement (msg
                       (departure &A_General(MyHost)&)
                       (arrival AAR)
                       (performative Task-Announcement)
                       (from &A_General(MyName)&)
                       (to LSist)
                       (reply-with &DK_MakeReplyWith&)
                       (expiration-time 30)
                       (contents ((require-knowledge ((== 1 1)))))
                   )

                   evaluate-ack (proc Organize)
                   bid-msg (msg
                       (departure &A_General(MyHost)&)
                       (arrival &A_TA_M(Departure)&)
                       (performative Bid)
                       (from &A_General(MyName)&)
                       (to &A_TA_M(From)&)
                       (reply-with &DK_MakeReplyWith&)
                       (in-reply-to &A_TA_M(Reply-with)&)
                       (contents ((regular-contents)))
                   )

             /* Beautifier and LSist */
             id 1
                   task-announcement (msg
                       (departure &A_General(MyHost)&)
                       (arrival AAR)
                       (performative Task-Announcement)
                       (from &A_General(MyName)&)
                       (to Beautifier)
                       (reply-with &DK_MakeReplyWith&)
                       (expiration-time 30)
                       (contents ((require-knowledge ((== 1 1)))))
                   )
                   task-announcement (msg
                       (departure &A_General(MyHost)&)
                       (arrival AAR)
                       (performative Task-Announcement)
                       (from &A_General(MyName)&)
                       (to LSist)
                       (reply-with &DK_MakeReplyWith&)
                       (expiration-time 30)
                       (contents ((require-knowledge ((== 1 1)))))
                   )

                   evaluate-ack (proc Organize)
                   bid-msg (msg
                       (departure &A_General(MyHost)&)
                       (arrival &A_TA_M(Departure)&)
                       (performative Bid)
                       (from &A_General(MyName)&)
                       (to &A_TA_M(From)&)
                       (reply-with &DK_MakeReplyWith&)
                       (in-reply-to &A_TA_M(Reply-with)&)
                       (contents ((regular-contents)))
                   )
          )
以下省略
```

図4　エージェント記述例 (一部)

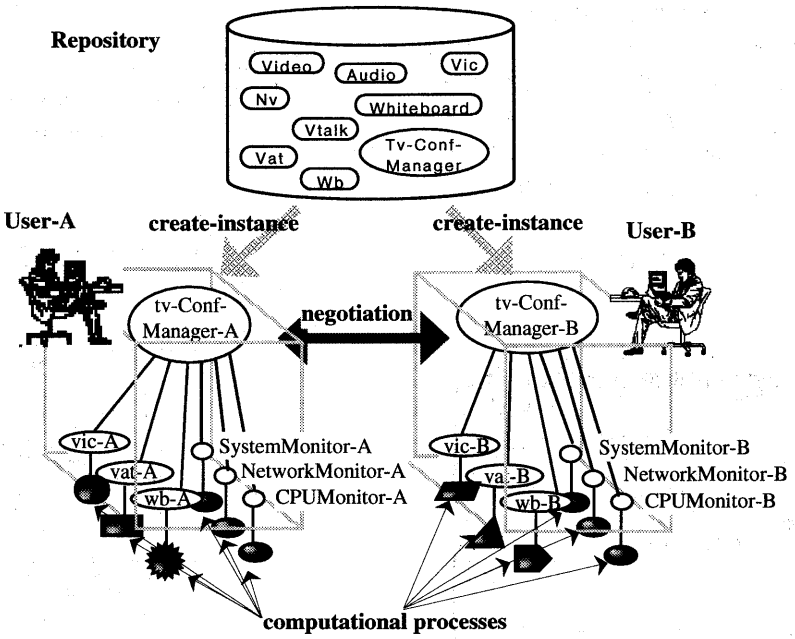


図 5　ADIPS フレームワークによるビデオ会議システム構成例

ために，他の利用者に利用されている，他の分散処理システムとの交渉を行い，分散処理システムのパラメータ変更，システム構成要素の変更，システム構成の変更を行う庫とにより目的を実現することを述べた．ADIPS フレームワークに基づいて設計/実装された環境により，目的の動作が行えることを確認した．現在，C++および Tcl によるものと，Java による二つの試験的実装を行っている．今後，エージェント間の協調プロトコルの設計を行い，これを導入したエージェント記述言語を定義する．

**謝辞**

日頃御討論頂く，東北大学電気通信研究所白鳥研究室，(株) サイエンティア，千葉工業大学情報工学科菅原研究室の皆様に感謝いたします．本研究の一部は沖電気工業による受託研究によるものです．

## 参考文献

[1] R. Braden, L.Zhang, and S. Berson. Resource reservation protocol(rsvp) - version 1 functional specification. draft-ietf-rsvp-spec.ps, 11 1996.

[2] N. Shiratori, K. Sugawara, T. Kinoshita, and G. Chakraborty. Flexible networks: Basic concepts and architecture. *IEICE Trans. Commun*, E77-B(11), 1994.

[3] R. G. Smith. The contract net protocol: Hight-level communication and control in a ditributed problem solver. *IEEE Trans. on Computers*, 29(12), 1980.

[4] 藤田, 菅原, 木下, and 白鳥. 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, 37(5), 1996.