

# ADIPS フレームワークのためのエージェント開発支援環境

原 英樹<sup>†</sup> 藤田 茂<sup>††</sup> 菅原 研次<sup>†</sup>  
木下 哲男<sup>†††</sup> 白鳥 則郎<sup>†††</sup>

分散システムをエージェント指向アーキテクチャで実現するための枠組みである ADIPS フレームワークの実用化のためには、大量のエージェントが開発され、それらがリポジトリに蓄積されていることが重要である。本論文では、これを支援する方式の提案を行い、そのための開発支援環境の設計と実装について述べる。本開発支援環境は、(A) 分散システム開発時に行う機能分割を支援する機能、(B) 分散システムを構成するエージェントの開発時にその知識記述を支援する機能、および、(C) 開発したエージェントの動作確認を支援する機能、をエージェントの開発者に提供する。

## Tools for Development of Agents for ADIPS Framework

HIDEKI HARA,<sup>†</sup> SHIGERU FUJITA,<sup>††</sup> KENJI SUGAWARA,<sup>†</sup>  
TETSUO KINOSHITA<sup>†††</sup> and NORIO SHIRATORI<sup>†††</sup>

Using ADIPS framework practically, which provides a methodology to develop distributed systems based on an agent-oriented architecture, it is important to develop many agents and store them into a repository. In this paper, we propose a support method for these activities and describe a design and implementation of tools which support these activities. Our tools help developers to (A) design a constructive specification of distributed systems, (B) describe domain knowledge of a designer to develop an agent, and (C) test the distributed systems which consist of agents.

### 1. はじめに

我々はこれまでエージェント指向分散システムを開発するための ADIPS (Agent-based Distributed Information Processing System) フレームワークの提案と、これに基づく実行環境の設計と開発を行ってきた<sup>1)</sup>。また本フレームワークを応用したシステムとしてやわらかい TV 会議システムなどを開発し、実用的フレームワークとしての妥当性を検証してきた<sup>5)</sup>。その結果、ADIPS フレームワークに基づいて開発した分散システムには、利用者要求駆動でシステムを構成できるという利点と<sup>1)~3)</sup>、環境の変化に対して機能や性能を柔軟に変化させる適応能力を持つという利点が

ある<sup>4)</sup>ことを確認することができた。

このような ADIPS フレームワークの利点を実現するためには多数のエージェントを蓄積し再利用することが必要である。そのためにはエージェント開発者の負担を軽減する開発支援機能を提供することが重要である。本論文では、ADIPS フレームワークのためのエージェント開発支援環境に必要な、以下の支援機能を実現する方式について述べることを目的とする。

- (A) 機能分割プロセス支援機能：設計対象である分散システムの機能を機能分割する際に、リポジトリに蓄積された既存エージェントを可能な限り再利用できるように機能分割する作業を支援する機能。
- (B) 知識記述支援機能：知的部品であるエージェントの知識を記述する際に、より容易にエージェントが開発できるように、記述作業を支援する機能。
- (C) 動作確認支援機能：開発したエージェントをリポジトリに蓄積する際に、このエージェントが既存のエージェントと協調して、目的とするマルチエージェントシステムとしての機能を実現することを確認する作業を支援する機能。

以下、2 章では ADIPS フレームワークの概要を述

<sup>†</sup> 千葉工業大学情報ネットワーク学科  
Department of Network Science, Chiba Institute of Technology

<sup>††</sup> 千葉工業大学情報工学科  
Department of Computer Science, Chiba Institute of Technology

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

べる。3章では、ADIPS フレームワークにおいて再利用部品となるエージェントの開発とリポジトリへの蓄積に必要な技術である、前述した(A), (B), (C)の3つの支援機能について述べる。これらの支援機能に関しては、4章で支援機能(A)で述べた“部品再利用指向の機能分割を支援する機能”の設計について詳細に説明する。5章では支援機能(B)の“エージェントの知識記述を支援する機能”と、支援機能(C)の“機能の実現を確認する作業を支援する機能”の設計について詳細に説明する。6章では上記の章で述べた設計に基づいて実装した開発支援環境の実装方式について述べる。7章では、本提案によるエージェント開発支援環境の有効性の議論を行う。

## 2. ADIPS フレームワークの概要

ADIPS フレームワークとは、利用者要求駆動によるシステムの構成と、障害発生などのイベント駆動によるシステムの再構成とを可能にするための、エージェント指向分散処理システムを実現する枠組みである<sup>1)</sup>。

図1にADIPSフレームワークの構成を示す。ADIPSフレームワークに基づいて開発される分散処理システムは、ADIPS動作環境上に組織構成されたプリミティブエージェントと組織構成エージェントの2種類のインスタンスエージェントと呼ばれるエージェントから構成されるマルチエージェントシステムである。この2種類のエージェントは、図1の中のリポジトリ内に格納されたクラスエージェントから生成される。クラスエージェントは、分散処理システムの利用者の要求に応じて、クラスエージェントの持つシステム設計に関する領域知識とエージェント間の組織構成プロトコル<sup>1)</sup>を利用して、利用者要求を実現する分散システムのためのエージェントの階層的組織を自動的に設計する。すなわちクラスエージェントは定型的設計<sup>12)</sup>に関する設計者の知識をエージェントとして分散的に格納しており、リポジトリはこれらの知識を用いたエージェントによるマルチエージェントシステムの設計プロセスの場を形成している。これらの設計作業の結果として、利用者へサービスを行うための機能を有するインスタンスエージェントから構成されるマルチエージェントシステムとしての分散システムが動作環境に構成される。

ADIPSフレームワークの利点として、次の2点があげられる。

### ●利用者要求に基づくシステムの動的構成

ネットワークには様々な目的と利用スキル水準を持つ多数の利用者が存在する。このような多様な利用

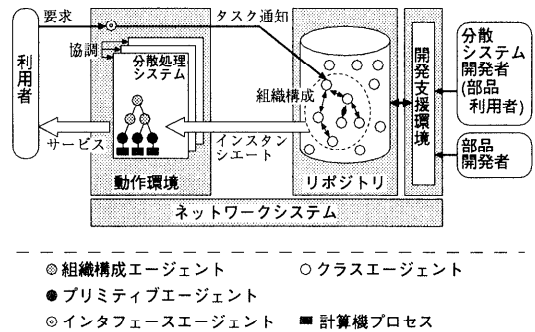


図1 ADIPS フレームワーク

Fig.1 ADIPS framework.

者要求に対し、ADIPS フレームワークではリポジトリに蓄積されたエージェントを用いて利用者要求駆動で柔軟にマルチエージェントシステムを構成し、様々な利用者の要求を満たすことが可能である<sup>1)</sup>。

### ●動作状況の変化への適応

ネットワーク環境では、利用者の増減や要求の変化に依存して、あるいは動作する計算機タスクの個数やサイズの変化などで、動作状況は変化する。ADIPSフレームワークでは、このような状況の変化などに対応するため、状況変化を監視するエージェントからの報告メッセージに対して、変化の対象となるマルチエージェントシステムが、リポジトリに蓄積されたエージェントと協調して機能・性能の調整や組織変更による機能の変化などを自的に行い、状況変化に適応する<sup>4)</sup>。

## 3. ADIPS フレームワークに基づくエージェント開発支援方式

ADIPS フレームワークは、リポジトリに蓄積されたエージェントを部品として再利用し、上記の利点を持つ分散システムをマルチエージェントシステムとして動的に構成するための枠組みである。以下、マルチエージェントシステムをMAS (Multi-Agent System) と略記する。本フレームワークは図1に示すように、(1)クラスエージェントを蓄積するリポジトリ、(2)インスタンスエージェントがサービスを実現する動作環境、(3)エージェントを開発するための開発支援環境、の3つの要素から構成される。(1)および(2)の実現法については文献1)~3)で発表している。本論文では(3)の実現法を提案し、その設計と実装について述べる。

まず初めに、ADIPSのエージェントを部品再利用に基づくソフトウェア開発方式の再利用部品と見なしたときのMASの開発の概要を図2を用いて説明する。MAS開発者(エージェント利用者)は、リポ

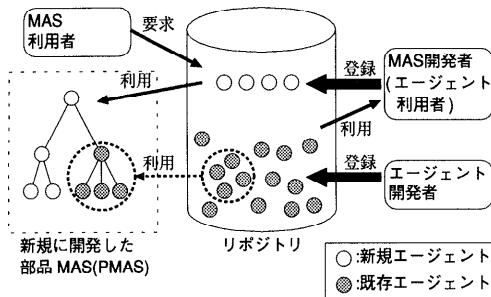


図2 ADIPS フレームワークを利用したマルチエージェントシステム (MAS) の開発

Fig.2 Development of Multi-Agent System (MAS) based on ADIPS framework.

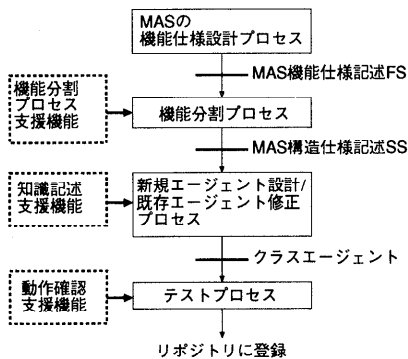


図3 MAS 開発プロセスと支援機能の概要

Fig.3 Overview of MAS development process and its support.

ジトリに格納されている既存エージェントを利用して目的とするMASの開発に必要なエージェントの組織(再利用する部品としてのMAS)の開発を行う。この部品として利用されるMASをPMAS(Parts of Multi-Agent System)と略記する。新規PMASの部品として開発された新規エージェントは、リポジトリに蓄積され他の分散システムを構成する既存エージェントとして再利用される。あるいは、既存エージェントの領域知識<sup>2),3)</sup>を追加、変更することにより、エージェントの機能を拡張し、これをリポジトリに蓄積し再利用する。このようにエージェントの機能を拡張することで、多様なエージェントの組織(新規PMAS)が構成可能になり、その結果、リポジトリはエージェント利用者に対して様々な機能を持つPMASを提供することが可能となる。

上記の開発のプロセスを図3に示す。図3の「機能分割プロセス」は、与えられたMAS機能仕様記述(Functional Specification: FS)からMASを構成するそれぞれのエージェントの選定・修正・新規開発を

行い、MAS構造仕様記述(Structural Specification: SS)を生成するプロセスである。ここで、FSとはMASを構成する個々のエージェントが持つ機能仕様(TASK知識)の集合であり、また、SSとは個々のエージェントが持つ構造仕様(PROC知識)の集合である。

ソフトウェア部品の再利用がソフトウェアの生産性の向上に有利である<sup>7)</sup>のと同様に、リポジトリに蓄積されている既存エージェントを可能な限り再利用し新規に開発するエージェントの個数を減らせば、MASの開発全体の負担を小さくできる。このためには、1章(A)項で述べた、“与えられたFSを構成するために利用可能な既存エージェントを検索し機能分割を支援する機能”が必要となる。

次の「新規エージェント設計/既存エージェント修正プロセス」では、MAS構造仕様記述SSに記述された部品としてのエージェントのうち、新規設計が必要なもの、あるいは既存エージェントの領域知識の修正/追加が必要なものの設計を行い、その知識記述を行う。また、これらを部品として利用する所与のFSを実現するクラスエージェントをFSとSSに基づいて設計し、知識記述を行う。このために、1章(B)項で述べた“エージェントの知識記述を支援する機能”が必要である。この結果、FSを実現するのに必要なすべてのエージェントの実体が作成される。

最後の「テストプロセス」は、作成されたクラスエージェントが目的とする機能仕様記述FSを実現することを確認するプロセスである。確認する項目は以下の2つである。

- 図3のプロセスの結果作成されたクラスエージェントおよび、その部品となる既存クラスエージェントが、リポジトリに投入されたFSを実現するエージェントの組織を構成すること。
- 構成されたクラスエージェントの組織から、図1に示すように、動作環境に生成されたインスタンスエージェントのMASが、所与の機能FSを実現すること。

このテストプロセスでは、1章(C)項で述べた“MASの動作確認を支援する機能”が必要となる。この支援機能により上記の2つの確認項目を満たしていないことが分かった場合には、機能分割プロセスや新規エージェント設計/既存エージェント修正プロセスに戻りそれぞれの作業を行う。

図3のプロセスの中で必要となる支援機能を図4に示す。図4の(A)機能分割プロセス支援機能は、本論文で提案する部品再利用に基づくMAS開発には重

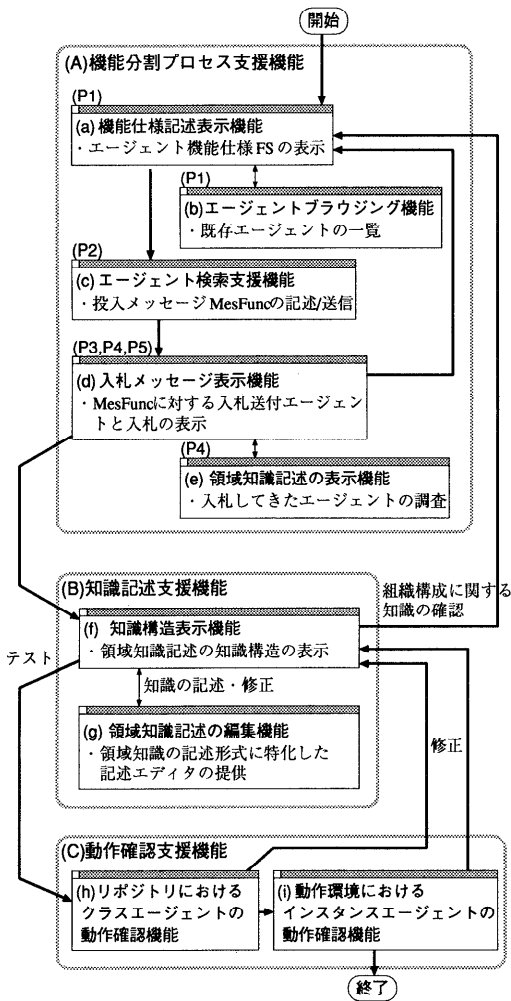


図4 開発支援環境の設計

Fig. 4 Design of support tool for agent development.

要な支援機能である。また、図4の一部を構成する(B)知識記述支援機能、(C)動作確認支援機能についても、ADIPSフレームワークにより分散システム開発の負担を軽減するために必要な支援機能である<sup>6)</sup>。4, 5, 6章ではそれらも含めて、図4に示すADIPSフレームワークのための開発支援環境の実装方式の説明を行う。

#### 4. 機能分割プロセス支援機能

##### 4.1 機能分割プロセスの支援方式

部品再利用を指向したシステム開発法では、部分機能がリポジトリに蓄積されている部品で実現できるように目的とする機能の機能分割を行うことが、その開発効率の向上につながる。また、これをADIPSフレームワークで行う場合には、部分機能を実現する部

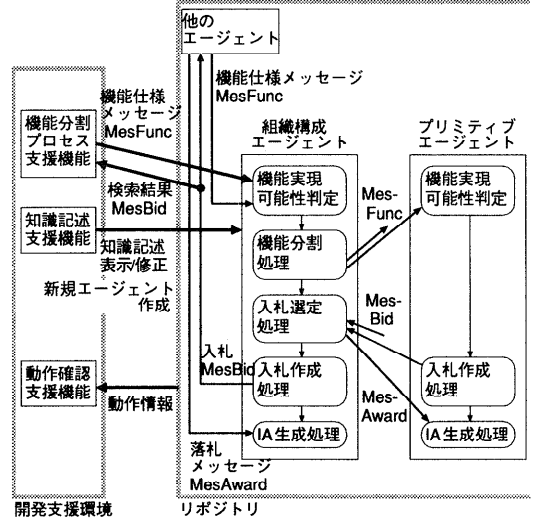


図5 クラスエージェント間のエージェント検索プロトコル  
Fig. 5 Protocol for component retrieval used by class agent.

品として可能な限り既存の組織構成エージェントを指定することで、過去の設計結果を再利用することが可能となり、さらなる開発効率の向上につながる。その理由は図2に示したように、部品として指定された組織構成エージェントは、自動的にその機能を実現する部品としてのマルチエージェントシステム (PMAS) を構成するからである。このような機能分割を行うための支援方式の説明のために、まずクラスエージェントが必要な部品を検索する方法について説明する。

ADIPSフレームワークに基づくエージェントは、その領域知識ベースの中に、与えられた機能仕様 FS の実現性の判定、入札作成、機能分割、入札選定などの処理のための知識を格納している<sup>2),3)</sup>。図5に示すように、リポジトリに機能仕様メッセージ *MesFunc* (契約ネットプロトコル<sup>13)</sup>のタスク通知に相当) が投入されたとする。このとき、各エージェントは機能実現可能性判定知識を用いて処理可能と判定したときのみ、入札作成知識を用いて *MesFunc* に対する動作条件を作成し、投入元に返送 (入札) する。投入元は、入札選定知識を用いて、入札してきたエージェントのうち、最もその機能実現に適当なものを1つ選び出し、それに落札条件を示したメッセージを送付する。

このプロセスは、設計者が部品を検索し選定するプロセスを、エージェントに格納されたエージェントの利用に関する設計知識と組織構成プロトコルによりシミュレートしている。すなわち、エージェントは与えられた機能 *F* を実現するために部分機能を検索し選定するプロセスを、そのエージェントを開発した設

計者の知識を利用して、自動的に実行している。本論文で述べる機能分割プロセス支援方式は、クラスエージェントの持つ上記の機能を対話的に利用することにより、エージェント利用者のエージェント検索と選定の作業の負担を軽減する。以下にその方式を述べる。

エージェント利用者は図3で機能分割プロセスに与えられた機能仕様記述  $FS$  に対して、それを実現するためのいくつかの部分機能の候補の集合  $\{f_1, \dots, f_n\}$  に分割し、その各々を図4の(c)エージェント検索支援機能の一部であるメッセージ記述支援機能を利用して記述し、リポジトリに機能仕様メッセージとして投入する。リポジトリ内のエージェントは各々の機能仕様メッセージについて、これを実現するための条件や、そのときの提供可能な機能の品質を領域知識を用いて確認することで、実現可能性の判定を行う。たとえば、図5の組織構成エージェントの場合は、実現可能と判断した機能をいくつかの部分機能に分割して、これらの部分機能を実現するエージェントを検索するため、リポジトリにそれぞれの機能仕様メッセージを自動的に投入する。この過程は再帰的に繰り返され、最終的にプリミティブエージェントと呼ばれる機能仕様を直接実現できるエージェントにそれぞれ分割された部分機能の機能仕様メッセージが伝播される。この過程でエージェントが計算した機能を実現するための条件や、提供可能な機能の品質の情報を記述した入札記述が、再帰的にプロセスを戻っていく過程で積算され結合される。これにより得られた部品候補の機能  $\{f_1, \dots, f_n\}$  に対してこれらを実現するエージェントと、その実現の条件や提供する機能の品質などの情報がエージェント利用者に提供される。

#### 4.2 機能分割プロセスの定義

図4の(A)の機能分割プロセス支援機能は、リポジトリに蓄積されたエージェントを再利用しつつ、目的の機能  $F$  を実現するエージェントを開発するための機能分割作業を支援する。この支援機能を設計するために、図3内に示したエージェント利用者が行う機能分割プロセスを、以下のように  $P_1$  から  $P_5$  からなるプロセスとして説明する。

( $P_1$ ) 所与の機能仕様記述  $FS(F)$  を分析し、機能  $F$  を構成するための部分機能の分割の  $i$  番目の候補  $C(F, i) = \{f_1^i, f_2^i, \dots, f_n^i\}$  を定義する。次に、それらの関係  $R(C(F, i)) = \{(x, y, z) | x \in C(F, i), y \in C(F, i), z \in DataDoc\}$  を定義する。ただし、 $z$  は MAS の動作時に  $x$  が  $y$  に渡す情報を表す記述である。現状では  $DataDoc$  は領域依存の文書情報として定義されている。最後に、定義した  $C(F, i)$  と

$R(C(F, i))$  をもとに、部分機能  $f_1^i, \dots, f_n^i$  の機能仕様記述  $FS(f_1^i), \dots, FS(f_n^i)$  を記述する。

( $P_2$ ) 機能仕様記述  $FS(f_1^i), \dots, FS(f_n^i)$  のそれぞれをエージェント間の機能仕様メッセージ  $MesFunc(f_1^i), \dots, MesFunc(f_n^i)$  に変換し、これらリポジトリに投入する。

( $P_3$ )  $MesFunc(f_j^i)$  に対し入札を行った、再利用候補部品の集合  $RE(f_j^i) = \{a_1, \dots, a_m\}$  の中の最適なエージェント  $a_k$  を選択する。

( $P_4$ ) 分割の候補  $C(F, i)$  の要素である部分機能について、以下のように分類する。

- 再利用可能な既存エージェントが存在する部分機能
- 修正することにより再利用可能な既存エージェントが存在する部分機能
- 既存エージェントが存在せず新規開発が必要な部分機能

( $P_5$ )  $P_4$  の結果から、部分機能の分割の候補  $C(F, i)$  が適正と判断すれば、 $FS(F)$  の構造仕様  $SS(F) = (C(F, i), R(C(F, i)))$  を記述し、図3の次の開発プロセスへ進む。 $C(F, i)$  が適正でない判断すれば、プロセス  $P_1$  へ戻り、次の分割の候補  $C(F, i+1)$  に対して同様に行う。

#### 4.3 機能分割プロセス支援機能の設計

4.2節の各プロセスにおけるエージェント利用者の作業を支援する機能として、以下の機能を設計した。

(a) 所与の機能仕様記述  $FS$  を表示する機能 ( $P_1$  支援) : ADIPS フレームワークでは、エージェントの機能仕様を記述する形式がBNF形式で定義されている<sup>3)</sup>。これは、エージェントに格納され処理される知識記述の形式なので、このエージェントの機能仕様をエージェント利用者が読みやすくするための、インデント機能や記述属性名を強調するなどの表示機能を提供する。

(b) エージェントブラウジング機能 ( $P_2$  支援) : エージェント利用者が部分機能の分割の候補  $C$  を定義する作業において、カテゴリーに分類されたエージェントを順次ブラウジングし、そのエージェントが持つ機能を抽象的に表現している「機能仕様知識名」、「機能分割知識名」、「部品選択・入札知識名」に関する説明機能を提供する。これを手がかりに、(d)が提供する詳細な領域知識記述を調べていく機能を提供する。

(c) エージェント検索支援機能 ( $P_3$  支援) : 図5において、エージェント利用者による再利用可能なエージェントの検索を支援するため、機能仕様記述  $FS$

をエージェントが処理可能な形式に変換し、リポジトリに投入する機能を提供する。この投入されたメッセージに対して、実現可能と判断したエージェントは、その条件や性能を返送（入札）する。

(d) 入札メッセージ表示機能 ( $P_3, P_4, P_5$  支援) :

(a)と同様に、入札の条件などはエージェント間で定義されたメッセージ形式で送付されてくる。その内容を、(a)と同じようにエージェント利用者が理解しやすい形式で表示する。また一般に、1つの機能仕様メッセージに対して多くのエージェントが入札してくるが、これらをリストとして一覧できる機能を用意し、エージェント利用者がそれらの内容を見て、再利用するのに適した順序で並べる機能を提供する。

(e) エージェントの領域知識記述表示機能 ( $P_4$  支援) : エージェント利用者が再利用するエージェントの優先順位の設定や、修正の必要性の検討には、入札を行ったエージェントの知識記述を調べることができる機能を提供する必要がある。この機能については5章で述べる。

本節で支援の対象としている機能分割プロセスにおいて、所与の機能仕様記述  $FS(F)$  から部分機能の分割の候補  $C$  を推定する作業は、エージェント利用者の高度な経験知識を必要とする作業である。これについては、(a)から(e)の支援機能を用いて、 $P_1$  から  $P_5$  プロセスを繰り返すことにより、可能な限り新規開発するエージェントを少なくできる分割法を発見することが必要である。

## 5. 知識記述支援機能と動作確認支援機能

図3に示した「知識記述支援機能」および「動作確認支援機能」については、文献(6)ですでに報告した。図4に示すように、これらの機能を「機能分割プロセス支援機能」に加えて、全体でADIPSフレームワークの開発支援環境を構成する。この開発支援環境の効果を7章で評価するが、そのために本章ではこの2つの支援機能の概要を述べる。

知識記述支援機能は、ADIPSフレームワークにおけるエージェントの領域知識記述言語ADIPS/L<sup>(2),(3)</sup>の記述を支援する機能である。ADIPS/Lで記述する知識項目間には、図6に示したような組織構成プロトコルの性質を反映した複雑な関係が存在するため、この関係を整理して表示する支援機能が必要である。またADIPS/Lは知識項目ごとに記述形式の定義が異なるため、記述形式の複雑さを隠蔽して知識記述作業を誘導し、記述すべき知識の内容に集中できるような機

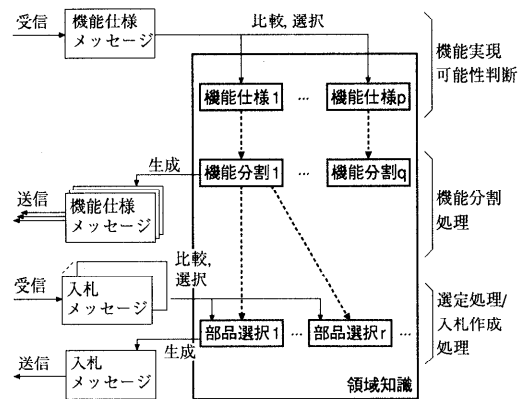


図6 領域知識の構造  
Fig. 6 Structure of agent domain knowledge.

能が必要である。以上を考慮し設計した結果、知識記述支援機能として図4(B)に示した2つの機能で構成した。

(f) 知識構造表示機能：知識項目間の参照関係を可視化して表示し、処理順序の理解を容易にする。

(g) 領域知識記述の編集機能：知識項目ごとに異なる知識構造を持つ記述ウィンドウを提供する。各ウィンドウの構造は知識構造のみを表示してADIPS/Lの文法を隠蔽し、記述すべき項目に誘導する。これにより知識記述者は記述すべき知識の内容のみに集中して作業することが可能となる。その結果、従来のテキストエディタでの記述に比べ記述が容易になる。

一方、動作確認支援機能は、利用者要求駆動で自動的に組織構成と組織再構成を行うエージェントの動作のテストに必要な機能である。開発したエージェントが目的とする機能を実現していることを確認し、また、動作中に発生した障害に対して適切にエージェント組織の構成を変えることを確認するために、この機能は重要である。図4(C)に示したように、動作確認支援機能は次の2つの機能により設計した。

(h) リポジトリにおけるクラスエージェントの動作確認機能：知識記述の結果のクラスエージェントの動作を確認するためのビューを提供する。リポジトリ内のエージェントの動作とエージェント間メッセージの送受信の流れを視覚化することで、開発したクラスエージェントが適切に契約を行い分散システムを実現するインスタンスエージェントの階層的組織が構成されることを確認する手段を与える。これは2章で述べたADIPSの利点である、利用者要求に基づくシステムの動的構成を確認するうえで重要な機能である。

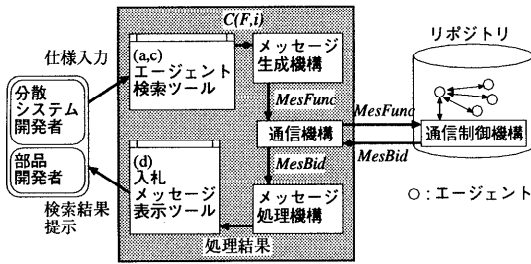


図7 機能分割プロセス支援機能の実装方式

Fig. 7 Implementation of tools for functional refinement process.

(i) 動作環境におけるインスタンスエージェントの動作確認機能: (h)と同様に、インスタンスエージェントの動作を確認するためのビューを提供する。動作環境上に生成されたインスタンスエージェントの階層的組織とエージェント間メッセージの送受信の流れを視覚化し、インスタンスエージェントが協調プロトコルを用いて適切に協調していることを確認する手段を与える。これは2章で述べたADIPSの利点である、動作状況の変化への適応を確認するうえで重要な機能である。

## 6. エージェント開発支援環境の実装

### 6.1 機能分割プロセス支援機能

4章と5章では、図4の支援機能(a)から(i)のそれぞれの支援機能の設計を行った。本章ではこれらの機能設計に基づくエージェント開発支援環境の実装の方式の詳細について述べる。エージェント開発支援環境はADIPSフレームワークと同様にJava言語を用いて実装した。また、支援機能(a)と(c)、支援機能(b)と(f)および、支援機能(e)と(g)は、その利用する状況を考慮して1つのツールとして実装している。

はじめに、1章(A)項で述べた機能分割プロセス支援機能の実装方式について述べる。これは図7に示すように、支援機能(a)と(c)を実現するエージェント検索ツール、支援機能(d)を実現する入札メッセージ表示ツール、メッセージの生成/解析機構および通信機構の各モジュールにより実現される。それぞれのモジュールはJavaのオブジェクトとして実装されており、モジュール間の通信は各モジュールで定義されたメソッドを呼び出すことで行う。以下、エージェント検索ツールと入札メッセージ表示ツールの実装方式を述べる。

#### (a, c) エージェント検索ツール

図8に示すエージェント検索ツールは、4.2節で述べた機能分割プロセス $P_1$ 、 $P_2$ を支援するツールである。

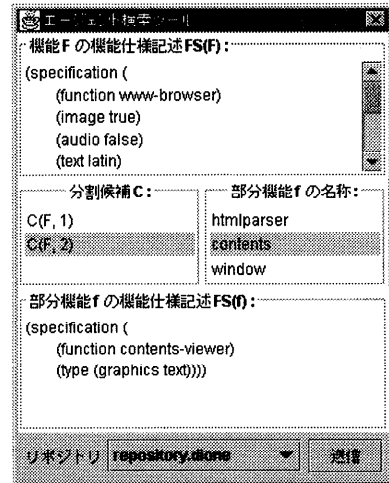


図8 エージェント検索ツール (支援機能 (a, c))

Fig. 8 Tool for component retrieval (supports (a, c)).

$P_1$ の目的は機能仕様記述 $FS(F)$ からの分割の候補 $C(F, i)$ を生成することである。エージェント利用者が入力した $FS(F)$ は、本ツールの「機能 $F$ の機能仕様記述 $FS(F)$ 」欄に整形されて表示される。この表示をもとに6.2節で詳述するエージェントブラウザを利用して $FS(F)$ の分割の候補 $C(F, i) = \{f_1^i, \dots, f_n^i\}$ を調べ、部品再利用に基づくエージェント開発方式の観点から最も適当な機能分割 $C$ を選定する。この選定を支援するためエージェント検索ツールには、分割候補名をリスト項目として追加する「分割候補 $C$ 」欄、部分機能 $f_j^i$ をリスト項目として追加する「部分機能 $f$ の名称」欄、部分機能 $f_j^i$ の仕様を記述する「部分機能 $f$ の機能仕様記述 $FS(f)$ 」欄のそれぞれを実装した。

一方、 $P_2$ の目的は $C(F, i)$ から機能仕様メッセージ $MesFunc$ に変換し、これをリポジトリへ投入することである。これを実現するために、 $P_1$ で記述した $C(F, i)$ を受け取り $MesFunc$ を生成するメッセージ生成機構および、 $MesFunc$ をリポジトリの通信制御機構に送信する通信機構を実装した。リポジトリの通信制御機構は、同一のリポジトリ内あるいはネットワークで接続された他のエージェント動作環境に存在するエージェント間の通信を仲介するモジュールである。開発支援環境とリポジトリはネットワークで接続された異なる計算機上で稼働するため、開発支援環境の通信機構とリポジトリの通信制御機構間の通信はJavaのRMI機能により実現している。

#### (d) 入札メッセージ表示ツール

図9に示す入札メッセージ表示ツールは、4.2節で述べた機能分割プロセス $P_3$ 、 $P_4$ 、 $P_5$ を支援するため

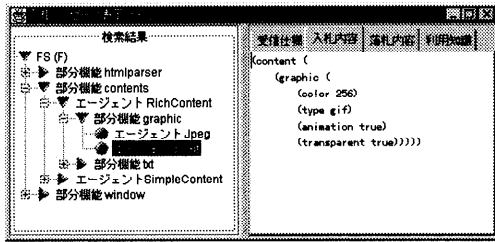


図9 入札メッセージ表示ツール (支援機能 (d))

Fig. 9 Tool for presenting bid message (supports (d)).

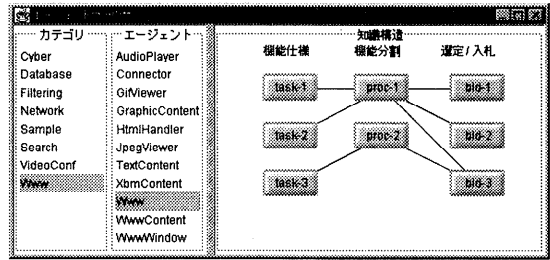


図11 エージェントブラウザ (支援機能 (b, f))

Fig. 11 Tool for browsing agents (supports (b, f)).

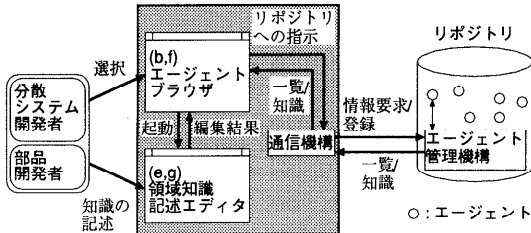


図10 知識記述支援機能の実装方式

Fig. 10 Implementation of tools for describing agent knowledge.

に以下の機能を実装した。

- $P_3$  の目的である最適なエージェントの選択の支援のための、入札メッセージの表示機能。
- $P_4$  の目的であるエージェントの再利用の可能性の判断のための、入札したエージェントの知識記述や機能仕様を表示する機能。
- $P_5$  の目的である構造仕様の記述のための、エージェント間の関係を木構造で表示する機能。

また、これらの機能の実現のためにメッセージ処理機構を実装した。

投入した *MesFunc* に対してエージェントから返答される入札メッセージ *MesBid* は、リポジトリの通信制御機構および開発支援環境の通信機構を介し、検索結果としてメッセージ処理機構に通知される。メッセージ処理機構は *MesBid* に記録されている機能分割の履歴を解析し木構造に構造化するモジュールである。入札メッセージ表示ツールはこの結果を図9に示したように表示することで、上記の  $P_3$  から  $P_5$  を支援する。

## 6.2 知識記述支援機能

本節では1章(B)項で述べた知識記述支援機能の実装方式について述べる。これは図10に示すように、支援機能(b)と(f)を実現するエージェントブラウザ、支援機能(e)と(g)を実現する領域知識記述エディタおよび、リポジトリのエージェント管理機構との通信を行う通信機構により実現される。

## (b, f) エージェントブラウザ

図11に示すエージェントブラウザは、4.2節で述べた機能分割プロセス  $P_1$  の一部と新規エージェント設計/既存エージェント修正プロセスを支援するツールである。このツールは、リポジトリ内のエージェントをカテゴリごとに分類したリストと、リストから選択したエージェントが持つ領域知識の構造のグラフ表示部からなる。

開発支援環境の通信機構とリポジトリのエージェント管理機構は、JavaのRMI機能を用いて通信を行う。エージェントブラウザは、起動時には現在のエージェントの一覧を要求してリスト表示部に表示し、また利用者によるエージェントの選択時には選択されたエージェントの知識を、通信機構を介してエージェント管理機構に対して要求する。取得したエージェントの知識はエージェントブラウザにより処理され、その知識構造がエージェントブラウザにグラフとして表示される。

機能分割プロセス  $P_1$  におけるエージェントブラウザの利用目的は、リポジトリ内の既存エージェントを見ることで機能分割の手がかりを得ることである。また、新規エージェントの設計や既存エージェントの修正の際にはエージェントの知識構造をグラフとして表示する機能により、知識の記述に重要である知識構造の把握を容易にする。

## (e, g) 領域知識記述エディタ

知識の記述項目ごとに特化した構造を持つエディタを提供する。例として、機能分割知識のための領域知識記述エディタを図12に示す。

領域知識記述エディタはエージェントブラウザから起動される際に、編集の対象となる知識記述の内容を受け取る。知識の内容の理解と記述の誘導のために、それぞれの知識項目ごとに定義された記述形式に基づいた形式でエージェントの知識の内容を提示し、また、記述すべき知識項目を提示して必須項目を強調するように実装した。



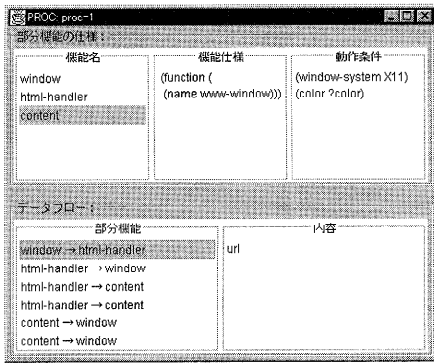


図 12 領域知識記述エディタの一部 (支援機能 (e, g))

Fig. 12 A part of editor for agent domain knowledge (supports (e, g)).

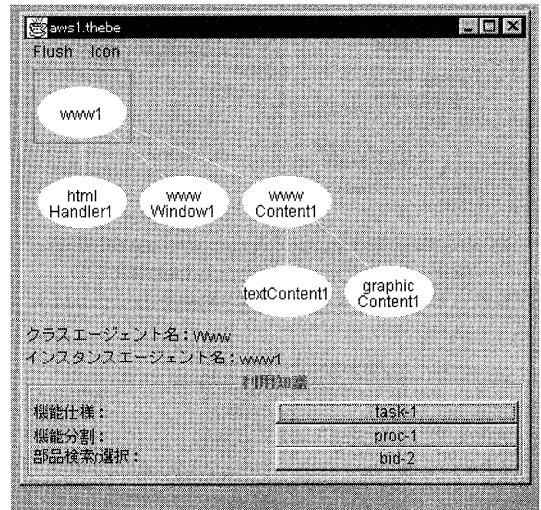


図 14 インスタンスエージェントの動作確認ツール (支援機能 (i))

Fig. 14 Instance agent viewer (supports (i)).

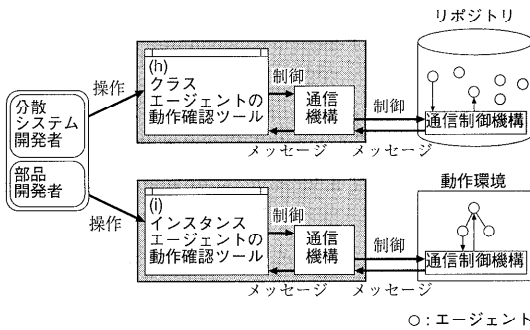


図 13 動作確認支援機能の実装方式

Fig. 13 Implementation of tools for checking agent behavior.

開発者が記述した内容は、知識の編集結果としてエージェントブラウザに渡され、エージェント記述言語 ADIPS/L に変換される。その後、通信機構によりリポジトリのエージェント管理機構に送信される。エージェント管理機構はこれを受け取り、これまでのエージェントを消去し、新しい知識を持つエージェントを生成する。

### 6.3 動作確認支援機能

本節では 1 章 (C) 項で述べた動作確認支援機能の実装方式について述べる。これは図 13 に示すように、支援機能 (h) を実現するクラスエージェントの動作確認ツール、支援機能 (i) を実現するインスタンスエージェントの動作確認ツールおよび、リポジトリ/動作環境の通信制御機構との通信を行う通信機構により実現される。

#### (h) クラスエージェントの動作確認ツール

開発したクラスエージェントが、投入された利用者要求に対して適切に組織を構成することを確認するツールである。リポジトリ内のクラスエージェントと、

クラスエージェント間のメッセージの流れを可視化し、エージェント利用者がクラスエージェントの動作を理解する作業を支援する。

動作確認ツールは起動されるとリポジトリの通信制御機構に対しエージェント間メッセージを動作確認ツールにも送信するように指示を出す。これによりすべてのエージェント間メッセージは、エージェントに送信される前に動作確認ツールの通信機構を介して動作確認ツールに送られるようになる。エージェント間メッセージは他のエージェントへの情報の通知時だけでなく、エージェントが生成あるいは消滅するときにも送信される。動作確認ツールはこれらのメッセージを受け取り、内容に応じてメッセージの移動やエージェントの生成、消滅などの様子をアニメーション表示する。1つのメッセージに関するアニメーション表示が終了すると、動作確認ツールはリポジトリの通信制御機構にそれを通知し、通信制御機構は動作を再開する。このようにアニメーションはエージェントの通信と同期して表示される。

#### (i) インスタンスエージェントの動作確認ツール

開発したクラスエージェントから利用者要求の投入により構成されるインスタンスエージェントの動作確認に用いる (図 14)。前述した支援ツール (h) と同様に、エージェントの動作の可視化を行うツールであり、メッセージの送信やエージェントの生成される順番がアニメーション機能により表示される。これにより、開発者は記述したエージェントが意図したと通りに組織を構成するかどうかを確認することが可能となる。これは、様々な組織構成を動的に行うエージェントの

知識記述を行ううえでの特徴的支援機能である。

## 7. 開発支援環境の効果

本論文で設計し実装した開発環境が、ADIPS フレームワークにおける分散システムの開発に与える効果について述べる。

### (1) 機能分割プロセスの負担軽減

現在、コンポーネントウェアなどのオブジェクト指向に基づく部品再利用方式が、従来の大規模ソフトウェア開発に関する問題点を解消する有効なアプローチとして期待されている<sup>8)</sup>。これまでの部品再利用は、部品庫に部品を登録するときの部品情報に関するキーワード検索、仕様に関する全文検索、あるいはハイパーテキストを利用した部品のブラウジングにより部品を探索している<sup>10),11)</sup>。これは CORBA などの分散ソフトウェアのオブジェクト指向開発基盤<sup>9)</sup>を用いた場合にも該当し、その基盤上で部品を可能な限り再利用するためには部品検索のプロセスは重要である。

これに対し、これまでの部品再利用方式にはない本方式の利点は、4章で述べた支援機能(a), (b), (c), (d), (e)により人間と部品(クラスエージェント)との対話的な部品検索を可能にしている点である。クラスエージェントは、協調機能と設計知識を利用して部品検索の過程を部品利用者に提示する。一方、部品利用者は、提示された内容から過去の設計事例や判断理由などの情報を得る。このように人間に対してエージェントが部品検索を支援することで、可能な限り部品を再利用できるように機能分割を行うための負担を軽減する。

### (2) 知識記述作業の負担軽減

ADIPS フレームワークが提供するエージェント指向システムは、2章で述べたように、利用者要求に基づくシステムの動的構成や動作状況の変化への適応などの利点を持ち、今後のやわらかいネットワーク社会を実現するための重要な技術である<sup>14),15)</sup>。ADIPS フレームワークにおけるこれらの利点を実現するためには、リポジトリに十分な量と種類のクラスエージェントが蓄積されていることが必要な条件となる。しかしながら、クラスエージェントの開発はたとえばテキストエディタなどの従来型のソフトウェア開発環境を利用して行ってきたため、エージェントの検索の適切な支援がないこと、知識記述の複雑さ、および作成したエージェントの動作確認の方法がないことが、目的とする機能を持つ部品を開発するエージェントの開発者の負担になっていた。

本論文で提供した開発支援環境は、5章で述べた知

識記述のための支援機能(f), (g)を提供する。これにより、記述すべき知識の項目と内容の記述形式を誘導し、記述した結果の知識構造を視覚的に分かりやすく表示する。そのためエージェントの開発者の知識記述作業の負担を軽減することが可能となる。

### (3) 動作確認作業の負担軽減

ADIPS フレームワークでは、再利用部品であるエージェント自身がその部分機能を実現する再利用部品であるエージェントを検索し選択する。しかしその過程は計算機内部で自動的に行われるため、その選択の妥当性を確認することが必要である。特に、エージェントの領域知識を修正した場合や、既存の部品と類似した新しい部品を部品庫に追加する場合に、正しく動作するかの確認が必要である。しかしながら、これまでエージェントの動作を分かりやすく表示するツールがなく、開発者は自動的に選択されたエージェントの知識記述を通常のエディタで読むことで動作確認を行っていた。

上記の作業負担を軽減するために、本開発支援環境は5章で述べた支援機能(h), (i)を提供する。これにより、作成したエージェントとそのエージェントが利用する部品としてのエージェントの動作をアニメーションなどの表示により視覚的に確認することが可能となり、再利用部品であるエージェントの動作確認作業の負担軽減が可能となる。

## 8. おわりに

本論文では、エージェント指向分散システムを実現する枠組みであるADIPS フレームワークのための、エージェントの開発環境の設計と実装について述べた。本エージェント開発支援方式の特長は、過去のエージェント開発者の設計知識に基づいて動作する既存エージェントが、その知識を利用して、新規エージェントの開発者に対して様々に支援する点にある。本開発支援環境はこの利点を生かすことを目標として設計し、実装を行った。

新しいエージェントの開発者である部品利用者は、本開発支援環境を利用してリポジトリの内部で行われる既存エージェントの協調作業を観察することにより、過去の設計知識を保持する既存エージェントを利用するための新規エージェントの開発における作業の負担が軽減される。また、この開発に用いた設計知識は新規エージェントとしてリポジトリに蓄積され、他のエージェント利用者に再利用される。以上により、本方式はこれまで提案してきたADIPS フレームワークを用いた分散処理システムの開発の負担を軽減し、

その実用化に貢献する。

謝辞 支援ツールの実装を担当していただいた、千葉工業大学大学院情報工学専攻〔現在、富士ソフトABC(株)〕の高橋誠康氏に感謝します。また、本研究の一部は東北大学電気通信研究所における共同プロジェクト研究の援助を受けています。

### 参考文献

- 1) 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).
- 2) 原 英樹, 藤田 茂, 菅原研次, 木下哲男, 白鳥則郎: 計算機プロセスのエージェント化のための知識記述方式, 電子情報通信学会論文誌, Vol.J81-D-I, No.5, pp.566-573 (1998).
- 3) 藤田 茂, 原 英樹, 菅原研次, 木下哲男, 白鳥則郎: エージェント指向分散処理システム ADIPS のための組織構成エージェントの領域知識記述形式, 情報処理学会論文誌, Vol.39, No.2, pp.188-198 (1998).
- 4) Fujita, S., Hara, H., Sugawara, K., Kinoshita, T. and Shiratori, N.: Agent-based Design Model of Adaptive Distributed Systems, *Applied Intelligence*, Vol.9, No.1, pp.57-70 (1998)
- 5) 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).
- 6) 原 英樹, 藤田 茂, 菅原研次, 木下哲男, 白鳥則郎: ADIPS フレームワークのための知識記述支援ツール, 情報処理学会論文誌, Vol.39, No.11, pp.3142-3145 (1998).
- 7) Biggerstaff, T.J. and Richter, C.: Reusability Framework, Assessment, and Directions, *IEEE Software*, Vol.4, No.2, pp.41-49 (1987).
- 8) 青山幹雄: コンポーネントウェア: 部品組立て型ソフトウェア開発技術, 情報処理, Vol.37, No.1, pp.71-79 (1996).
- 9) 佐藤広行, 大野邦夫: 共通オブジェクトリクエストブローカーアーキテクチャCORBA 1.1, 情報処理, Vol.35, No.9, pp.853-858 (1994).
- 10) Isakowitz, T. and Kauffman, R.J.: Supporting Search for Reusable Software Objects, *IEEE Trans. Softw. Eng.*, Vol.22, No.6, pp.407-423 (1996).
- 11) Mili, H., Mili, F. and Mili, A.: Reusing Software: Issues and Research Directions, *IEEE Trans. Softw. Eng.*, Vol.21, No.6, pp.528-562 (1995).
- 12) Brown, D.C. and Chandrasekaran, B.: *Design Problem Solving Knowledge Structures and*

- Control Strategies*, Morgan Kaufmann (1989).
- 13) Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *IEEE Trans. Comput.*, Vol.29, No.12, pp.1104-1113 (1980).
  - 14) 菅原研次: やわらかいネットワーク, 情報処理学会誌, Vol.36, No.9, pp.827-830 (1995).
  - 15) 白鳥則郎, 木下哲男, 菅原研次: 共生空間の実現へ向けて—ポスト・モダン分散システム, 電子情報通信学会誌, Vol.80, No.2, pp.165-168 (1997).

(平成 10 年 11 月 18 日受付)

(平成 11 年 9 月 2 日採録)

#### 原 英樹



1970 年生。1998 年千葉工業大学大学院博士後期課程情報工学専攻期間満了退学。同年同大学情報ネットワーク学科助手。博士(工学)。ソフトウェア再利用に興味を持つ。

#### 藤田 茂(正会員)



1968 年生。1997 年千葉工業大学大学院博士後期課程情報工学専攻期間満了退学。同年同大学情報工学科助手。博士(工学)。エージェント、分散処理システムに興味を持つ。

#### 菅原 研次(正会員)



1950 年生。1980 年東北大学大学院博士課程中退。同年千葉工業大学助手。現在同大学情報ネットワーク学科教授。工学博士。エージェント、サイバー社会に興味を持つ。

#### 木下 哲男(正会員)



1953 年生。1979 年東北大学大学院修士課程修了。同年沖電気工業(株)入社。1996 年東北大学助教授。工学博士。知識表現、知識型設計支援システムに興味を持つ。

#### 白鳥 則郎(正会員)



1946 年生。1977 年東北大学大学院博士課程修了。1984 年同大学助教授(電気通信研究所)。1990 年同大学教授(工学部情報工学科)。1993 年同大学教授(電気通信研究所)。