

分散処理システムのエージェント指向アーキテクチャ

藤田 茂[†] 菅原 研次[†]
木下 哲男^{††} 白鳥 則郎^{†††}

本論文では、最初に従来の分散処理システムのサービス機能を実現する種々の計算機プロセスをエージェント化して構成されるエージェント指向分散処理システム ADIPS (Agent-based Distributed Information Processing System) の概念とそのアーキテクチャを提案する。次に、ADIPS の構築を支援する枠組み (ADIPS フレームワーク) を提案する。そして、目的とする機能をエージェントの協調によって自律的に構成するための拡張契約ネットプロトコルを提案する。ADIPS フレームワークの特徴は次の4つに集約される。(1) 利用者要求駆動で自律的にシステムが構成される。(2) 障害発生などのイベント駆動で自律的にシステムの再構成が行われる。(3) 自律的なシステムの構成/再構成のために、エージェントは設計者・運用技術者の知識を利用する。(4) エージェント化により既存プロセスの系統的再利用ができる。さらに、本論文では、TCP/IP ネットワーク環境で試作された ADIPS フレームワークを用いたエージェント指向テレビ会議システムの試作と実験により、提案した枠組みの有効性を確認した。

Agent-based Architecture of Distributed Information Processing Systems

SHIGERU FUJITA,[†] KENJI SUGAWARA,^{††} TETSUO KINOSHITA^{††}
and NORIO SHIRATORI^{†††}

In this paper, first, we propose a new concept of Agent-based Distributed Information Processing System (ADIPS) and its architecture, which is constructed by agentification technology of various kinds of computational processes which provide services of conventional distributed information processing systems. Second, an ADIPS Framework which provides facilities of constructing ADIPS is proposed. Third, Extended Contract Net Protocol (ECNP) is proposed to provide a protocol for cooperation to construct and reconstruct organizations of ADIPS agents. The advantages of ADIPS Framework are (1) Autonomous construction of systems driven by user's requirements, (2) Autonomous construction of systems driven by events such as system troubles, (3) Heuristics of designers and operators is used by agents to construct and reconstruct organizations of agents to deal with tasks, (4) agentification makes reusability of existing processes more efficient. Finally, a prototype of the ADIPS Framework is developed in the distributed environment of TCP/IP network protocol, and an experimental system of a TV conference system based on the ADIPS architecture is developed to validate the ADIPS Framework.

1. はじめに

近年のネットワーク技術やマルチメディア技術の発達により、ネットワークにより接続されたデジタル化社会の到来が期待されている。ところが、オフィス、学校、家庭などで分散システムを日常的に利用する状況

を想定すると、規格化された定型的サービスに制限される従来の分散システムでは、多様な利用者要求に柔軟に対処することは困難となりつつある。現在、多様化/高度化する利用者要求に対応する利用者指向の分散システムや情報通信システムのアーキテクチャ^{1),2)}、設計技術^{3),4)}、あるいは運用技術⁵⁾などに関する研究開発が活発に進められているが、現状では、様々な利用者要求に対して、要求駆動で動的にサービスを構成するための効果的な手法は確立されていない。

一方、上述した課題は、所与の問題 (利用者要求) を動的に解決する (サービス機能の生成と提供) システムを構成する問題としてとらえられる。こうした、いわゆる問題解決システムを動的に構成するための技

[†] 千葉工業大学情報工学科

Department of Computer Science, Chiba Institute of Technology

^{††} 沖電気工業 (株) マルチメディア研究所

Multimedia Lab., Oki Electric Industry Co., Ltd.

^{†††} 東北大学電気通信研究所

Research Institute of Electrical Communication, Tohoku University

術は、これまで分散人工知能の分野を中心として研究開発が進められ、その実現形態としてマルチエージェントシステムの枠組みが提案されている^{6),7)}。

近年、こうしたエージェント技術は様々な分野で応用が試みられつつあるが⁸⁾、現状では、システム設計開発に対する応用はまだきわめて少なく、エージェント指向システムの設計方法論や設計支援環境を確立することも今後の重要な課題として残されている⁹⁾。

本論文では、エージェント技術に基づいて、従来の分散処理システムのサービス機能を実現する、種々の計算機プロセスを自律的なエージェントとして定式化（エージェント化）することによって構成されるエージェント指向分散情報処理システム（Agent-based Distributed Information Processing System: ADIPS）の概念を提案し、ADIPSにおけるエージェント指向システム設計を系統的に支援する新たな枠組み（ADIPS フレームワーク）を提案する。ADIPS フレームワークでは、利用者指向分散システムの実現に適した次のような特徴が得られる。すなわち、(1) 利用者要求駆動で自律的にシステムが構成される、(2) イベント駆動で自律的にシステムの再構成が行われる、(3) 自律的なシステム構成/再構成のために、エージェントは設計者・運用技術者の知識を利用する、(4) エージェント化により既存プロセスの系統的な再利用ができる。

これまで、エージェントのモデルは Telescript¹⁰⁾ などのように情報処理作業を利用者に代わって実行する比較的単純な構造のエージェントから、自律性、協調性、学習能力などの高度な能力を持つ知的エージェントに至るまで、様々なエージェントモデルが提案されている。本論文で提案するエージェントのモデルは、能力としてはこれらの中間に位置し、分散処理システムを動的に構成/再構成するための自律性と協調性を実現することを目的として設計されている。すなわち本論文は、高度な問題解決能力を有するエージェントのモデルを提案することが目的ではなく、既存の計算機プロセスをエージェント化することにより「分散環境で自律的、協調的に作業を実行するソフトウェアモジュールの集団」を構成することに主眼がおかれている。すなわち、ADIPS フレームワークでは、自律的、協調的エージェントを用いて、機能再利用、自律的システム構成/再構成の面で、これまでの分散処理システムにない機能を実現することができた。

以下、2章で、ADIPS とその構成要素となるエージェント（ADIPS エージェント）のアーキテクチャを提案する。そして、3章では、ADIPS フレームワークの設計思想を述べ、その基本的枠組みを提案する。

次に、4章で、ADIPS エージェントによって分散処理システムの構成/再構成を行う際に用いられる拡張契約ネットプロトコルを提案し、5章で、ADIPS エージェントに格納される知識ベースについて説明する。さらに、6章では、Unix 環境上で試作した ADIPS フレームワークの概要と、ADIPS フレームワークを利用して構築されたやわらかいテレビ会議システムの試作例を示し、ADIPS フレームワークの有効性を示す。

2. ADIPS と ADIPS エージェント

2.1 分散システム設計と ADIPS 構想

分散情報処理システム DIPS (Distributed Information Processing System) は、ネットワークや OS などで構成されるプラットフォーム上に分散され、相互に通信を行うことによって協調的に情報処理を実行する多数の計算機プロセスから構成される。一般に、計算機プロセスの集団は複数のサブシステムを構成し、これらのサブシステムが階層的に構造化されることにより DIPS が構成される。従来、こうした DIPS の構築、運用、保守においては、次のような問題が指摘されている。

- DIPS は、多数の利用者が共有・利用することから、多種多様な利用者要求を充足するシステムとなっている必要がある。しかしながら、すべての利用者要求を包含する DIPS を構築することは、システムの過度の大規模、処理効率の低下、あるいは、運用コストの増大などを招く要因となる。

- DIPS は、ネットワーク環境で分散的に動作する多数の計算機プロセスと、それらの複雑な相互関係に基づいて構成される。こうしたシステムでは、分散配置された計算機やそれらを接続するネットワークの動作状況の変化に起因する様々な影響を受け、システムの動作効率の低下や障害を引き起こす。

- 新しい処理要求の発生やネットワーク環境の要素の版更新などにより、DIPS のサービス機能や性能に関する変更が必要となる。しかしながら、一般に、大規模で複雑なシステムである DIPS の保守のコストやその期間は多大なものとなる。

本研究では、DIPS とその動作環境との相互作用の結果として生起するこれらの問題を解決するために、DIPS の設計モデルとしてエージェント指向アーキテクチャを導入する。すなわち、DIPS を構成する計算機プロセスを、自律的、協調的エージェントとして定式化（エージェント化）し、これらのエージェントの協調動作に基づいて、DIPS の動作環境で生起する様々な変化に対して自律的に適応する機能を実現する。こ

れをエージェント指向分散情報処理システム (Agent-based DIPS: ADIPS) と呼び、本論文では、その設計開発を効果的に支援する環境を提案する。本章では、その基礎として、設計目標となる ADIPS のアーキテクチャ、および、ADIPS の構成要素となるエージェントのアーキテクチャを提案する。

2.2 ADIPS のアーキテクチャ

DIPS は、利用者にサービス機能を提供する実体である計算機プロセスの集合体であり、次のようなプロセスのグラフとして定式化される。

$$DIPS = \langle CP, REL-CP \rangle.$$

ここで、CP は計算機プロセスの集合である。本論文では、CP に含まれる計算機プロセスをベースプロセス、また、DIPS を構成するサブシステムをベースサブシステムと呼ぶ。一方、REL-CP は、ベースプロセス/ベースサブシステム相互間でのデータや制御権の受け渡しなどの関係の集合である。したがって、前節で述べた問題の原因となる DIPS とその動作環境との相互作用から生じる種々の変化は、最終的に CP や REL-CP の変化としてとらえられる。したがって、これらの変化に対処するためには、DIPS 自身がベースプロセスやベースサブシステムの動作状況を管理・監視し、何らかの不都合な変化が観測された場合には、その状況に応じて CP と REL-CP を自律的に再構成して変化を吸収する機構が必要となる。

そこで、本論文では、DIPS のベースプロセスを自律的なエージェント (ADIPS エージェントと呼ぶ) として定式化し、利用者要求に即した REL-CP をエージェントの組織として構成・管理する機構を導入する。これをエージェント型管理機構 AMM (Agent-based Management Mechanism) と呼び、この AMM を備えた DIPS を ADIPS と呼ぶ。すなわち、ADIPS アーキテクチャは、

$$ADIPS = \langle AMM, DIPS \rangle$$

で与えられる。ここで、AMM は、3 種類の ADIPS エージェントからなるエージェントを節集合とすることにより、次のように形式的に定義される。

$$AMM = \langle PA, OA, IFA, REL-AG \rangle.$$

図 1 は、ADIPS アーキテクチャを模式的に表現したものである。以下、ADIPS アーキテクチャの中核となる AMM の構成について説明する。

まず、PA は、DIPS のベースプロセス $cp \in CP$ を直接的に監視・制御する ADIPS エージェントの集合であり、その要素はプリミティブエージェントと呼ばれる。プリミティブエージェントは、ベースプロセスを ADIPS エージェントとして定式化する際の基本的

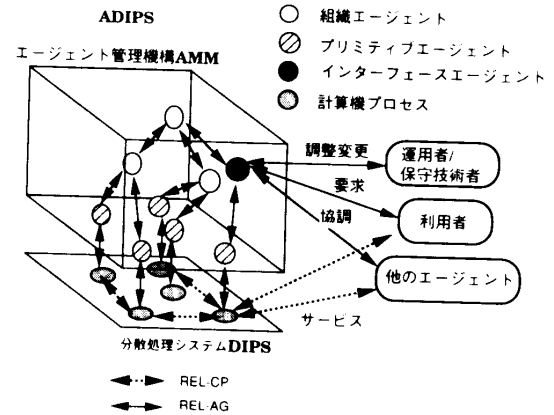


図 1 ADIPS アーキテクチャ
Fig. 1 ADIPS architecture.

手段を与えるものである。次に、OA は、DIPS のサブシステムを管理・制御する ADIPS エージェントの集合で、これらは組織エージェントと呼ばれる。各組織エージェントは、DIPS のサブシステムの構造に対応して階層的に組織化される。また、IFA は、利用者、運用者、保守技術者、あるいは他のエージェントシステムなどからなる DIPS 外部の動作環境との対話や相互作用 (協調) を行う組織エージェントの集合であり、これらは特にインターフェースエージェントと呼ばれる。さらに、REL-AG は、ADIPS エージェント相互間で締結された契約関係を表すもので、これによって DIPS のサービスを実現する ADIPS エージェント組織の構造が与えられる。

2.3 ADIPS エージェントのアーキテクチャ

プリミティブエージェント pa のアーキテクチャは、ベースプロセス cp を ADIPS エージェントとして定式化する 3 種類の機構に基づいて次のように定義される。

$$pa = \langle CM, TPM, DK, cp \rangle.$$

ここで、CM は協調機構 (Cooperation Mechanism: CM) と呼ばれ、 pa が組織エージェント oa との協調によって ADIPS エージェントの組織を動的に構成/再構成するための協調プロトコルとその処理機能を提供する。この CM は 4 章で提案する拡張契約ネットプロトコルを利用する。また、TPM はタスク処理機構 (Task Processing Mechanism: TPM) と呼ばれ、ベースプロセスの監視・制御・実行を担当する機構である。さらに、DK は領域知識知識 (Domain Knowledge: DK) と呼ばれる知識ベースであり、ベースプロセス cp が提供するサービス機能や他のプロセスとの関係などの設計仕様知識、およびそれらの運用方法を記述した運用知識など、エージェント pa に固有の機

能や特性に関する情報を格納・管理している。CMとTPMは、DKでの推論を参照し、拡張契約ネットプロトコルを用いることにより、paが属するエージェント組織における自己の役割を認識し、処理すべきタスクに応じたエージェント組織のメンバとして動作する。

一方、組織エージェントoaは、

$$oa = (CM, TPM, DK, MA)$$

と定義される。ここで、MAはメンバーエージェントと呼ばれるADIPSエージェントの集合であり、oaのサブタスクを処理するために締結された契約関係に基づくエージェント組織を表す。MAの要素はプリミティブエージェントか組織エージェントであり、これらのエージェントの関係が、DIPSのサブシステム階層に対応したメンバーエージェントとして構造化されることで、ADIPSエージェントの組織が定義される。上記の定義から明らかなように、MAを除けば、組織エージェントはプリミティブエージェントと同一のアーキテクチャを持つが、TPMに埋め込まれる機能はプリミティブエージェントと異なる点に注意されたい。すなわち、組織エージェントのTPMは、メンバーエージェントとして組織化されたADIPSエージェント群の制御・監視を行う機能を持ち、これによって、ADIPSの動作状況における種々の変化に対して自律的に対応することができる。

2.4 ADIPS エージェントによる ADIPS の構成と再構成

ADIPSの構成/再構成の設計プロセスが駆動される要因は次の2つである。(1) 外部から与えられた利用者要求に対処するために、その要求を受理したインタフェースエージェントIFAによってベースプロセスの構成/再構成の要求が発せられる場合。こうした要求は、それに対応できるAMM内の組織エージェントに送られ、その組織エージェントが、自分のメンバーエージェント群の構成や再構成の処理を試みる。(2) DIPSの稼働中に、あるADIPSエージェントにおいて非定型タスクが発生し、このタスクを処理するためにベースサブシステムを再構成する必要が生じた場合。非定型タスクを検出したエージェントは、(1)と同様に、これを処理することのできるエージェントに対して非定型タスクを送ることで、エージェント組織の構成や再構成が行われる。

メンバーエージェントに組織エージェントが含まれる場合には、そのメンバーエージェントに対して、構成/再構成の処理が再帰的に適用され、最終的には、プリミティブエージェントのレベルでベースプロセスの構成/再構成がなされる。

このように、種々の知識に基づいて自律的に処理を実行するADIPSエージェントにより、様々な利用者要求や動作環境の変化に動的に対応できるADIPSが実現される。

このような動的な組織構成における問題点として、局所的な情報に基づいて適応動作を行うとき、適応動作の遅延により連続的にエージェントの組織が再構成を繰り返す事態が発生することが考えられる。これに対してADIPSでは組織エージェントが、下部組織の再構成を行ったときの動作履歴を記憶しておき、組織再構成が必要となるイベントが発生したときにこの履歴を参照することにより、組織再構成の振動状態の検出を行い、これに基づいてDKに記述されている対象領域における振動状態回避の経験的知識を用いることで、振動状態の発生に対処している。

しかしながら、より適切な動作制御を行ううえで、エージェント動作やシステム動作状況変化に関する予測などが必要と考えられる。こうしたエージェント組織の動的再構成メカニズムの高度化は今後の課題である。

3. ADIPS フレームワーク

3.1 ADIPS フレームワークの設計構想

2章で提案したADIPS、およびその構成要素となるADIPSエージェントの設計開発を効果的に支援する環境としてADIPSフレームワークを提案する。まず、ADIPSフレームワークにおける重要な概念を以下に列挙する。

(1) エージェント化と協調機構

ADIPSは、対象とするDIPSを構成するベースプロセスやベースサブシステムに対して協調機構を付加する操作(エージェント化)により実現される。

(2) 拡張契約ネットプロトコルとエージェント組織の構成/再構成

ADIPSでは、エージェント管理機構AMMのADIPSエージェント組織によってDIPSが制御され、利用者へのサービスはベースプロセスの集団によって提供される。AMMは、拡張契約ネットプロトコルに基づくADIPSエージェントの協調動作によって構成される。また、DIPSの動作環境の変化に適応するためにAMMが再構成され、その結果がDIPSの再構成に反映される。

(3) 領域知識と設計プロセス

ADIPSエージェントには、エージェント化されたベースプロセスあるいはベースサブシステムの設計仕様・運用・保守に関する領域知識が格納される。これ

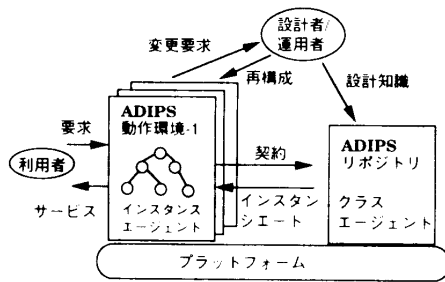


図2 ADIPS フレームワーク

Fig. 2 ADIPS framework.

はDIPSの開発時や運用時に技術者から獲得される知識で、これによって個々のADIPSエージェントの機能や役割が規定される。領域知識に基づいてADIPSエージェントの組織化の設計プロセスが実行される。

(4) リポジトリとクラスエージェント

ADIPSのAMMを構成するADIPSエージェントは、様々なDIPSの構築において共通的に利用される部品として共有データベースに格納される。こうした部品データベースはリポジトリと呼ばれ、そこには、各種のADIPSエージェントの原型(クラスエージェントと呼ぶ)が蓄積される。そして、サービスの構成要求やシステム動作環境の変化への対処要求が与えられると、適切なクラスエージェントから、実際にAMMの部品として動作するADIPSエージェント(インスタンスエージェントと呼ぶ)が生成される。クラスエージェントのアクセスやインスタンスエージェントの生成は、拡張契約ネットプロトコルに基づくADIPSエージェントの協調動作によって実行される。

上述した概念に基づいて設計されたADIPSフレームワークの構成イメージを図2に示す。すなわち、ADIPSフレームワークは、(1)ADIPS動作環境、(2)ADIPSリポジトリ、という2つのサブシステムから構成される。これらは、ネットワークやワークステーションからなるプラットフォーム上の分散システムとして実現される。以下、各々のサブシステムの機能について説明する。

3.2 ADIPS 動作環境

ADIPS動作環境(ADIPS Workplace: AW)は、ADIPSアーキテクチャのAMMとDIPSの実現基盤であり、利用者に提供されるDIPSのサービスは、AW上で稼働する多数のADIPSエージェント組織として実現される。AWのADIPSエージェントはインスタンスエージェントと呼ばれ、それらがADIPSアーキテクチャにおけるAMMの組織エージェントやプリミティブエージェントとして動作する。

ADIPSの利用者は、AWのインタフェースエージェ

ントIFAにアクセスし、IFAとの契約によって組織化されたインスタンスエージェント(組織エージェント/プリミティブエージェント)が提供するサービスを利用する。もし要求されたサービスを提供するインスタンスエージェントがAWに存在しない場合には、AWのエージェント組織を修正するか、ADIPSリポジトリ内のクラスエージェントと協調して新たなエージェント組織を構成することにより要求に対処する。一方、動作状況が変化してADIPSが正常な動作を維持することが困難となった場合には、同様に、ADIPSリポジトリとの協調動作により、現状のAWの再構成を行うか、変化した動作状況に即した新たなAWを自律的に構成してサービス維持を図る。

3.3 ADIPS リポジトリ

ADIPSリポジトリ(ADIPS Repository: AR)は、ADIPSのエージェント組織(AMM)を構成/再構成する際の部品となるADIPSエージェントを提供する機構であり、プラットフォーム上の複数のADIPS動作環境から利用可能なクラスエージェントの集合が動作する環境として構成される。AR上のクラスエージェントは、AW上のインスタンスエージェントからの利用要求に基づいて、要求されたサービスを提供するクラスエージェントの実体をAW上のインスタンスエージェントとして生成する。これをクラスエージェントの実体化と呼ぶ。

ARのクラスエージェントCAは次の構造を持つ。

$$CA = \langle CM, IM, DK \rangle.$$

ここで、IMは、エージェント実体化機構(Instantiation Mechanism: IM)であり、これによってクラスエージェントの実体(インスタンスエージェント)が生成される。

クラスエージェントのCMとDKは、インスタンスエージェントとして動作する組織エージェントやプリミティブエージェントと同一の構造を持つ。ただし、クラスエージェントの領域知識DKには、個別DKと共有DKという2種類の知識が格納される。すなわち、共有DKは、クラスエージェントの実体化に依存しない、そのクラスに共通する領域知識であり、必要に応じてインスタンスエージェントから参照・利用される。また、個別DKは、そのクラスエージェントから生成されるインスタンスエージェントに固有の領域知識であり、これは実体化されたインスタンスエージェントのDKに格納される。

3.4 ADIPS 動作環境とADIPSリポジトリの協調動作

2.4節で述べたADIPSの構成/再構成の設計プロセ

スは、AW と AR の協調動作によって実現される。

ADIPS の組織構成のプロセスでは、AW 上のインスタンスエージェント (たとえば、インタフェースエージェント) が、サービス実現に必要な ADIPS エージェントを得るために、AR に対して、あるタスクドメインに属するクラスエージェントの実体化要求を送る。AR では、その要求が処理可能であれば、クラスエージェントやそのメンバーエージェントを実体化し、AW 上に所与のサービスを実現する ADIPS エージェントの組織を構成する。

また、既存の ADIPS の組織再構成においては、利用者要求の変更や動作状況の変化を AW のインスタンスエージェントが検出し、そのインスタンスエージェントが、自分自身の領域知識を利用してエージェント組織の局所的な再構成を試みる。その処理が失敗した場合には、そのインスタンスエージェントが属するエージェント組織の上位の組織エージェントに再構成を依頼する。インスタンスエージェントの組織のみでの処理が不能な場合、上位エージェントは前述したエージェント組織構成プロセスと同様に、AR と協調して、組織の再構成を試みる。

このように、利用者要求や動作環境の変化に動的に対応する ADIPS を実現するうえで、ADIPS フレームワークに基づく AW と AR との協調動作が重要な役割を果たしている。そして、この協調動作を実現する協調プロトコルが、次章で定義する拡張契約ネットプロトコルである。

4. 拡張契約ネットプロトコル

拡張契約ネットプロトコル (Extended Contract Net Protocol: ECNP) は、契約ネットプロトコル¹¹⁾をベースにして定義されたもので、(1) 契約条件の変化による組織再構成、(2) 契約履行が不能になったときの組織再構成、に関するプロトコルを新たに追加することにより、環境の変化に対応して柔軟に分散処理システムを構成する協調プロトコルである。ECNP は、契約ネットプロトコルと同様に、エージェントの内部状態の状態遷移、および、タスク通知、入札、落札などの契約メタファに基づくメッセージパケットによって定義される。ADIPS エージェントの CM や TPM は、ECNP で定義した状態遷移を実行する機構であり、これらは知識ベース DK を参照することにより状態遷移を引き起こすイベントの分析とその処理を実行する。知識ベース DK については 5 章で述べる。

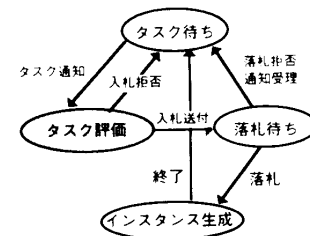


図3 クラスエージェントの CM の状態遷移
Fig. 3 State transition diagram of the CM of a class agent.

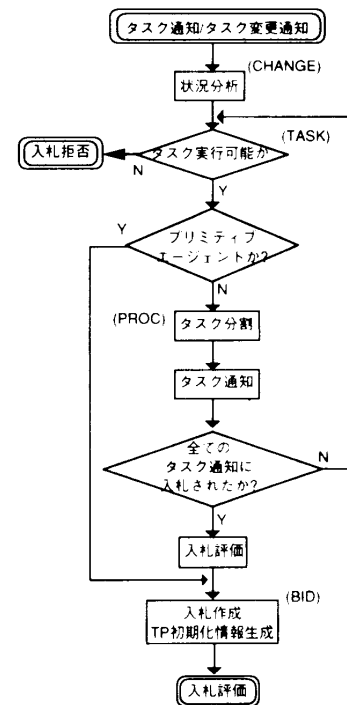


図4 タスク評価状態処理のフローチャート
Fig. 4 Flowchart of a process of the state of task evaluation.

4.1 クラスエージェントの協調機構 CM の状態遷移

クラスエージェントの状態遷移は図3に示す協調機構 CM の状態遷移により表される。タスク待ち状態にある CM はタスク通知に対して、タスク評価状態に遷移し、通知されたタスクが処理可能かどうかの判定を行う。

タスク評価状態の処理手続きを図4に示す。タスクが処理可能かどうかは、エージェントの能力を記述する DK の部分知識ベース TASK を参照することにより判定される。

通知されたタスクが処理可能と判定された場合、プリミティブエージェントでは、インスタンスエージェントを生成するために必要なベースプロセスの初期設定情報 (TP 初期化情報) を作成し、これを記憶した

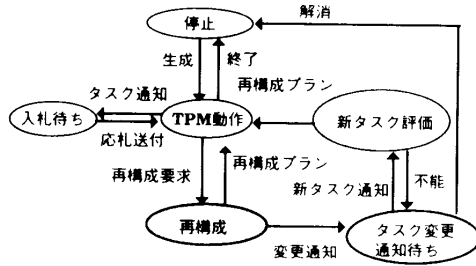


図5 インスタンスエージェントのCMの状態遷移

Fig. 5 State transition diagram of the CM of an instance agent.

状態でタスク通知を行ったエージェントに入札を行う。一方、組織エージェントでは、タスク分割プランに関するDKの部分知識ベースPROCを参照し、タスク分割プランにしたがってタスク通知を行う。

入札が行われた場合、その中で最も評価の高い入札の組合せを選定し、その組合せに基づいてTP初期化情報を生成、記憶する。これは落札が通知された後で、そのクラスエージェントがインスタンスエージェント（組織エージェント）をADIPS動作環境に生成する際に、そのメンバーエージェントに対して落札を発行するための情報として利用される。

通知されたタスクがその組織エージェントで処理不能の場合（入札拒否イベント）は、CMはタスク待ち状態に遷移する。

タスク通知に対して入札を行った後、クラスエージェントは落札待ち状態になる。落札が来ればインスタンス生成状態に遷移し、インスタンスエージェントを生成し、その後タスク待ち状態に遷移する。

4.2 インスタンスエージェントの協調機構CMの状態遷移

インスタンスエージェントの協調機構CMは、他のエージェントとの相互作用に関するプロトコルを実行する機構である。協調機構CMの状態遷移を図5に示す。

クラスエージェントに落札が通知されると、そのインスタンスエージェントがADIPS動作環境に生成される（生成イベント）。このとき、インスタンスエージェントのCMは停止状態からTPM動作状態に遷移する。TPM動作状態とは、エージェントの制御権をTPMに委譲した状態であり、エージェントが契約したタスクを処理している状態を表す。この間、新しい要求が発生したり契約条件を満たさない動作状況となり、メンバーエージェント以外の他のエージェントとの協調作業が必要になった場合には、制御権は再びCMに戻り、再構成状態に遷移する（再構成イベント発生）。

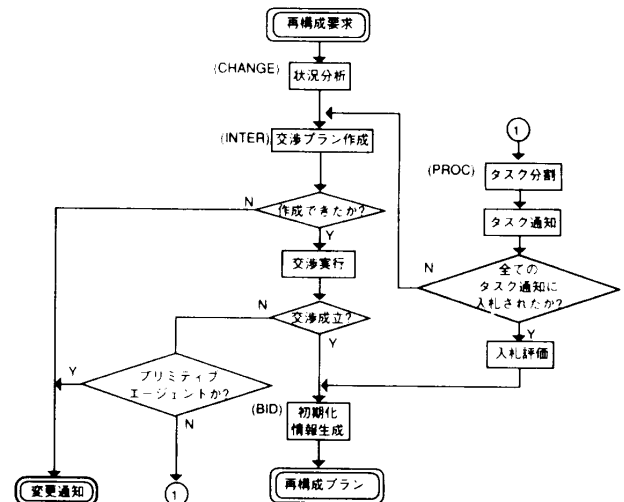


図6 再構成状態処理のフローチャート

Fig. 6 Flowchart of the process of the state of reconstruction.

再構成状態の動作フローを図6に示す。再構成状態では、契約条件を満たさなくなった原因を分析する。その原因としては、資源競合の発生、プラットフォームに障害や性能低下が発生、上位エージェントから送られるタスク仕様変更の通知、協調処理のプロセスにおける予測外の新規タスクの発生、などの事象が考えられる。これらに対して、CMは他のエージェントと協調して問題の解決を図る。

そのためDKの部分知識ベースINTERには、ECNPに含まれている他のエージェントとの交渉プロトコルに基づいて交渉を行うための知識が記述されている。同様に、DKには変化が生じた原因を分析するための部分知識ベースCHANGEが格納されている。この知識を用いた交渉の結果、ベースプロセスやメンバーエージェントが使用している資源が回復するなど、状況改善の効果が見られると、再びTPM動作状態に遷移する。

一方、上述した交渉により、そのエージェントの協調機構の問題解決能力では対処できないことが判明すると、そのエージェントでの局所的な回復処理を打ち切り、上位エージェントに変更通知を送り、上位エージェントに回復処理を依頼して、タスク変更通知待ち状態に遷移する。このとき、上位エージェントはこのエージェントと新たな条件での契約締結や現状の契約の解消について検討し、提示された新たな条件を満足すれば、再構成プランを記憶しTPM動作状態に遷移する。

また、インタフェースエージェントの場合、利用者などから新たなタスク要求を依頼されたとき、このタスクを処理するエージェント組織を構成するために、


```

message      = header addressee originator text trailer
text         = [context] | pstext
pstext       = task announcement
              |
              | ...
              | information-message
              |
              | reject-award
              | re-organization
              | stop-request
              | stop-release
              | dissolve
              | request-action
              | counter-request

reject-award = REJECT-AWARD [name] (reject-justification)*
re-organizaion = RE-ORG-MSG [name] (organization-specification)
stop-request  = STOP-REQUEST [name] (request-justification)*
stop-release  = STOP-RELEASE [name] (release-justification)*
dissolve      = DISSOLVE [name] (dissolve-justification)*
request-action = REQUEST-ACTION [name] (action-description)
counter-request = COUNTER-REQUEST [name] [message]
               (action-description)

```

図8 拡張コントラクトネットプロトコルのメッセージ形式
Fig.8 Extended contract net protocol specification.

表1 タスク領域知識 DK の部分知識ベースの内容

Table 1 Contents of sub-knowledge base of Task Domain Knowledge DK.

知識名	知識の内容	利用エージェント/内部機構
TASK	機能, 性能, 仕様資源など エージェントの能力に関する知識	クラスエージェントの CM
PROC	タスクの処理手順またはタ スク分割に関する知識	クラスエージェントの CM
BID	タスクの入札作成および TPM 初期化条件作成に関 する知識	クラスエージェントとインス タンスエージェントの CM
INTER	エージェント間の交渉や依 頼に関する手順を計画する ための知識	インスタンスエージェントの CM
CHANGE	発生した変化を分析し原因 を推定するための知識	インスタンスエージェントの CM
RECOV	TPM が直接制御している ベースプロセスやエージェ ントの障害を分析し, 回復 手段を生成する知識	インスタンスエージェントの TPM

クラスエージェントは, タスク評価状態 (図3) における動作フロー (図4) において, 知識ベース TASK, PROC, BID を参照することにより入札を実行する。インスタンスエージェントの協調機構 CM は, 再構成状態 (図5) における動作フローにおいて, 知識ベース INTER, CHANGE を参照することにより再構成プランを作成する。インスタンスエージェントのタスク処理機構 TPM は, 図7の TPM 変更処理において知識ベース RECOV を参照する。

ADIPS フレームワークでは, エージェントの設計と実装は, こうした DK の設計と実装に帰着され, CM と TPM は, これらの知識の解釈機構と見なすことができる。知識ベース DK の各知識表現モデルは, 各エージェントのタスクドメインに依存するが, これに

基づいて知識記述テンプレートが提供される。DK の部分知識ベースの標準的タスクドメインの知識表現モデルおよび知識記述テンプレートの詳細は稿を改めて報告したい。

6. ADIPS フレームワークおよび応用事例の試作

6.1 ADIPS 動作環境の実装と ADIPS リポジトリの試作

本論文で提案した ADIPS フレームワークの有効性を評価するために, ADIPS 動作環境と ADIPS リポジトリの試験的実装を行った。図9に示すような TCP/IP プロトコルで接続された UNIX ワークステーション上に分散した Smalltalk および知識ベース管理

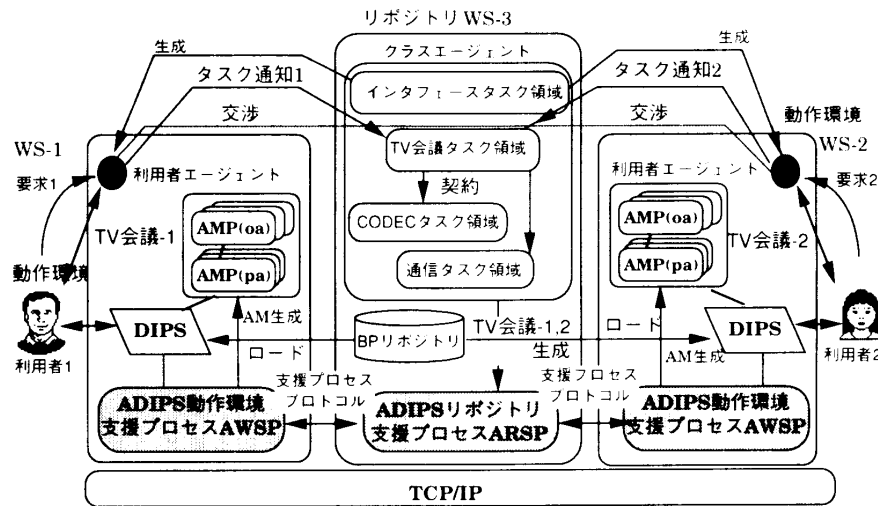


図9 ADIPS フレームワークの実装とテレビ会議システムへの応用

Fig. 9 Implementation of the ADIPS framework and application TV conference system.

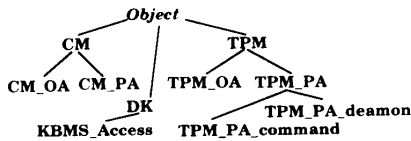


図10 Smalltalk でのクラス表現 (一部)

Fig. 10 A part of class hierarchy of ADIPS implementation on Smalltalk.

システム KBMS¹²⁾を用いて実装した。

図10にSmalltalk上に構築したADIPS動作環境およびリポジトリの実装に用いたクラス階層の一部を示す。ObjectはSmalltalkで定義されている基本クラスである。CM, TPM, DKはADIPSエージェントの構成要素の抽象クラス定義である。CM.OAは組織エージェントのCMを実現するクラスであり、CM.PAはプリミティブエージェントのCMを実現するクラスである。KBMS.Accessは知識ベースとして用いたKBMSとSmalltalkの接続とメッセージの変換を実現するクラスである。TPM.OAは組織エージェントのTPMを実現するクラスであり、TPM.PAはプリミティブエージェントのTPMを実現する抽象クラスである。プリミティブエージェントの場合には、対象とする計算機プロセスの種別・計算プロセスとの通信方式によりさらに下位クラスとして複数のクラスが定義される。

CM, TPMそれぞれのクラスは4章で述べた拡張コントラクトネットプロトコルを実装したメソッドを持ち、これにより、エージェントの自律性と協調プロトコルが実現される。領域知識ベースDKは、手続き型知識をメソッドとして持つが、フレームやルール型の知識は、下位クラスのKBMS.Accessにより管理

される。KBMS.AccessはCM/TPMからのメッセージをKBMSのフレームに変更し、ソケットを通じてKBMS上の各エージェントの知識ベースに対してロードする。このフレームのロードとCM/TPMの状態により表1で定義された部分知識ベースのいずれかがアクティブになり、CM/TPMの状態を決定するフレームが生成され、KBMS.Accessを通じてSmalltalkのメッセージに変更される。このためにKBMSに対してはソケット通信を実現する通信プロセスを介してアクセスしている。

ADIPSリポジトリ(AR)は、図9に示すように1つのSmalltalk環境にクラスエージェントの集合として実装されている。ある1つのクラスエージェントは図10に述べたSmalltalkのCM, TPM, DKのサブクラスのインスタンスオブジェクトから構成され、ARに常駐している。

動作環境のインスタンスエージェントからのタスク通知がARに到着することにより、入札可能なクラスエージェントが4章に述べたプロトコルに基づいて落札した後、動作環境にインスタンスエージェントを生成する。

実装に要したクラスの総メソッド数は82である。またDKとして利用したKBMS上の知識記述の例を図11に示す。図11は、6.2節で述べる例題であるテレビ会議システムを構成する3種類のADIPSエージェントの領域知識DKにおけるフレーム型知識TASK記述の一部である。ここで、AvairableTaskOfTVConfというフレームは、#TVConf-oaエージェントにおけるテレビ会議システムに関するタスクの知識を記述したもので、たとえば、通信方式に関する属性connection-

```

#TVConf_pa
(defFrame AvailableTaskOfTVConf
  :super (Task)
  (connectionType UDP) (max_fps 30) (max_color 256)
  (min_color 1) (max_bandwidth 1024) (min_bandwidth 1)
  (video_coding (jpeg mpeg h261 nv ser cellb))
  (audio_quantize (4 8 16)) (audio_coding (pcm adpcm))
  ...)
)

#Movie_pa
(defFrame AvailableTaskOfMovie
  :super (Task)
  (connectionType UDP) (max_fps 30) (max_color 256)
  ...)
)

#Movie_send_pa(1)
(defFrame AvailableTaskOfMovieS1
  :super (Task)
  (connectionType UDP) (max_fps 30) (max_color 256)
  (min_color 16) (max_bandwidth 3072)
  ...)
)

#Movie_send_pa(2)
(defFrame AvailableTaskOfMovieS2
  :super (Task)
  (connectionType UDP) (max_fps 30) (max_color 4)
  (min_color 1) (max_bandwidth 512)
  ...)
)

```

図 11 TASK の記述例
Fig. 11 Example of 'TASK'.

Type, あるいは、動画変換方式の属性 video-coding などを明示的に指定することにより、タスクに関する仕様が宣言的に記述されている。

6.2 ADIPS 型テレビ会議システムの試作

ADIPS の能力を評価するための例題として、ADIPS フレームワークを用いてテレビ会議システムを構成した。テレビ会議システムは、参加者を表示する動画像、図面や文書を表示する静止画、音声といった各種の通信メディアを含む複合システムである。これらのメディアに要求される品質は利用者ごとに異なり、さらに同じ利用者でも、利用場面ごとにその QoS (Quality of Service) が変化する。さらに、通信回線は多数のアプリケーションによって共有されるため、回線の遅延時間などの動作状況が変化する。こうした変化に応じて、利用者が要求する QoS を維持するためには、たとえば、音声通信の品質を下げることによって映像品質を維持するなど、テレビ会議システムの部分機能間でのトレードオフが必要となる。こうしたトレードオフの結果に基づいて、機能・性能を動的に変更できるシステムを、ここでは ADIPS 型テレビ会議システムと呼ぶ。

図 9 に示す ADIPS 型テレビ会議システムの試作例では、利用者 1 のテレビ会議システムに対する要求 1 は「参加者の画像は可能な限りスムーズな動画を表示する」ことであり、利用者 2 の要求 2 は「参加者の画像は高画質の画像を表示する」ことである。

実験の初期段階では、利用者エージェント 1 と利用者エージェント 2 は、要求 1 と要求 2 に基づいて交渉を行い、そのときの回線状態が良好であったため、要求 1 と要求 2 を同時に満足するテレビ会議システムが双方の ADIPS 動作環境に生成された。このとき利用されていたプリミティブエージェントの一部が、図 11 で示した #Movie_send_pa (1) である。

その後、回線の状況が劣化し、要求 1 と要求 2 が同時に満足できない状況を設定すると、TV 会議-1 と TV 会議-2 の再構成が自律的に開始される。その結果、TV 会議-1 では、モノクロの低解像度ではあるがスムーズな動画を見ることができ、TV 会議-2 では、静止画ではあるがカラーで高解像度の高画質の画像を見ることができる構成となり、回線が悪化した状態でも、利用者 1 と利用者 2 の双方の要求を満足することが確認された。このとき利用者 1 の動画再生を実現したプリミティブエージェントの一部が図 11 で示した #Movie_send_pa (2) である。利用者 2 はパラメータ変更によって fps を下げた #Movie_send_pa (1) によって高画質の静止画を見ることができた。

6.3 評 価

(1) ADIPS フレームワークの機能面の評価

ADIPS フレームワークに基づいて動作支援プロセス AW とリポジトリ支援プロセス AR を Smalltalk 環境内にクラスとして実現した。この結果、

- 計算機プロセスのエージェント化が系統的に行えるようになり、従来困難であったプロセスレベル/サブシステムレベルでの機能再利用が容易となることが確認された。
- エージェント化により、ベースプロセスに対して種々の運用などの知識を付加することが可能となり、これらの知識を活用することにより、利用要求に応じた分散システムの柔軟な構成や自律的な再構成がこれまでの分散処理システムの構成法と比べて容易に行えることが確認された。

現在 ADIPS エージェントの機能として自律的・協調的な性質を主眼として設計を行っており、知的エージェントの持つ学習等の機能は付加していない。エージェントに学習能力を付加することにより、組織の自律的な能力の高度化が期待できるが、これは本研究の今後の課題とする。

(2) ADIPS フレームワークの実装面の評価

知識処理機構を備えたエージェントやエージェント指向システム (アーキテクチャ) の様々なモデルの提案が行われているが、実装環境の実現方式の提案と評価は多くない。本論文では複数のワークステーショ

ン上で動作する Smalltalk と知識ベース管理システム KBMS を用いて、提案したフレームワークの実装を行い、既存の UNIX プロセスを協調させることにより、テレビ会議システムなどの分散アプリケーションを利用者要求駆動により自律的に構成できることを示した。

今回の実装では、組織構成/再構成を行うための時間が、例題の 17 個のクラスエージェントによる組織構成で 1 分程度の時間がかかる。したがって、さらにクラス数や知識数の増大することにより、組織構成の時間はきわめて増大することが予想される。この実装時の速度の問題は、今後の課題であり、現在 CM, TPM, DK の実装言語を C++ とすることにより、問題の解消を図る予定である。

(3) ADIPS フレームワークの応用面の評価

利用者指向のアプリケーションシステム構築のための有効な技術がこれまで、ほとんど提供されていなかった。しかし、本論文で提案した ADIPS フレームワークにより、エージェント分散アプリケーション向きのシステム構築基盤が与えられた。そして、本フレームワークを用いて試作されたテレビ会議システムにより、多様な利用者に対して、その利用者要求に基づいてインタフェースや機能を提供する利用者要求指向システムの実現可能性が検証された。なお、この試作システムは、利用者要求の変更や動作状況の変化に対して、自律的に機能を変更するやわらかいシステム^{2),3)}の実現例ともなっている。

7. 結 論

本論文では、利用者要求や動作状況の変化に適応するために、動的に分散処理システムを構成するための枠組み ADIPS フレームワークを提案し、これを実現するためのエージェント型アーキテクチャとその実現環境のモデルを与えた。動的にエージェントの組織を構成するために、エージェントが協調を行うためのプロトコルとして、契約ネットプロトコルを拡張した拡張契約ネットプロトコルを提案した。これらの手段により、ADIPS は、環境の様々な変化に対応して自律的に自己機能の変更を行うことができる。

提案した ADIPS フレームワークの評価を行うために、UNIX ワークステーションのネットワーク環境で、Smalltalk と知識ベース管理システム KBMS を用いて ADIPS 動作環境と ADIPS リポジトリを試作した。さらに、これを用いた応用システムとしてテレビ会議システムを試作した。その結果、多様な利用者要求に基づいて環境の変化に対応可能な分散アプリケーション

システムを構成することができ、ADIPS フレームワークの有効性が確認された。

参 考 文 献

- 1) Doyle, J.S. and McMahon, C.S.: Intelligent Network Concept, *IEEE Trans. Commun*, Vol.36, No.12, pp.1296-1301 (1988).
- 2) Shiratori, N., Sugawara, K., Kinoshita, T. and Chakraborty, G.: Flexible Networks: Basic Concepts and Architecture, *IEICE Trans. Commun*, Vol.E77-B, No.11, pp.1287-1294 (1994).
- 3) 竹中豊文 (編): 特集: フレキシブルネットワーク, 信学誌 Vol.77, No.4, pp.349-454 (1994).
- 4) Zitterbart, M., Stiller, B. and Tantawy, A.N.: A Model for Flexible High-Performance Communication Subsystems, *IEEE J. SAC*, Vol.11, No.4, pp.507-518 (1993).
- 5) Vogel, A., Kerherve, B., Bochmann, G. and Gecsei, J.: Distributed Multimedia and QOS: A Survey, *IEEE Multimedia Mag.2* (2), pp.10-19 (1995).
- 6) Bond, A.H.: *Readings in Distributed Artificial Intelligence*, Morgan Kaufman Pub. (1988).
- 7) Riecken, D. (Ed.): Special Issue: Intelligent Agents, *CACM*, Vol.37, No.7 (1994).
- 8) 村上国男: マルチエージェントシステムとその応用, 信学誌, Vol.78, No.6, pp.570-577 (1995).
- 9) Wooldridge, M.J. and Jennings, N. (Eds.): *Intelligent Agents*, Lecture Note in Artif. Intell. Vol.890, Springer-Verlag (1995).
- 10) General Magic, Magic Cap Concept, www.genmagic.com/MagicCapDocs/Concept/.
- 11) Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *IEEE Trans. Comp.*, Vol.C-29, No.12, pp.1104-1113 (1980).
- 12) KBMS-3 リファレンスマニュアル, NTT ソフトウェア (1993).

(平成 7 年 10 月 2 日受付)

(平成 8 年 3 月 12 日採録)

藤田 茂 (学生会員)



藤田 茂 (学生会員)

1968 年生。1992 年千葉工業大学情報工学科卒業。同大学院博士後期課程情報工学専攻在学中。エージェント、マルチエージェントに興味を持つ。平成 7 年度電子情報通信学会学術奨励賞受賞。電子情報通信学会会員。



木下 哲男 (正会員)

昭和 28 年生. 昭和 54 年東北大学大学院修士課程修了. 同年沖電気工業(株)入社. 現在, 同社研究開発本部マルチメディア研究所主任研究員. 知識表現, 知識型設計支援システム, ヒューマンインタフェース, 協調分散システムなどの研究開発に従事. 本会平成元年度研究賞受賞. 工学博士. 電子情報通信学会, 人工知能学会, 日本認知科学会, ACL, AAAI 各会員.



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 年同大助教授(電気通信研究所). 1990 年同大教授(工学部情報工学科). 1993 年同大教授(電気通信研究所). 情報通信システム, ソフトウェア開発環境, ヒューマンインタフェースの研究に従事. 1993 年本会マルチメディアと分散処理研究会主査. 本会 25 周年記念論文賞受賞. IEEE, 電子情報通信学会, 人工知能学会各会員.



菅原 研次 (正会員)

1950 年生. 1973 年東北大学工学部通信工学科卒業. 同年富士通株式会社入社. 1975 年東北大学大学院工学研究科修士課程入学. 1980 年同大学院博士課程中退. 同年千葉工業大学電子工学科助手. 現在千葉工業大学情報工学科教授. 工学博士. 分散人工知能, 知識型設計方法論, CAI, ソフトウェア再利用, やわらかいシステム, サイバー工学に興味を持つ. 1992 年日本工業教育協会功績賞, 1994 年情報処理学会山下記念研究賞受賞. 電子情報通信学会, 人工知能学会, ソフトウェア科学会, IEEE 各会員.
