ADIPSフレームワークにおける協調プロトコル実現手法

1H－4　藤田茂†　原英樹‡　菅原研次‡　木下哲男††　白鳥則郎††

千葉工業大学情報工学科†　千葉工業大学情報ネットワーク学科‡
東北大学電気通信研究所††

1　はじめに

情報ネットワーク技術の発達と普及に伴い情報空間を介して共同作業を実現することが容易になりCSCW等を用いたアプリケーションシステムが一般化している．これまでCSCW等では作業者間の協調を促進を対象としてきたが，ソフトウェアエージェント技術の発展に伴って，従来作業者が実行していた情報検索，情報処理等の作業をこれらソフトウェアエージェントに代行させることが期待されている．一方，情報ネットワークを実現する分散システム間の相互作用は複雑化高度化していることから，これら情報ネットワークの設計/保守作業が困難になっている[1].

我々はこれまでに自律的な計算機プロセスであるソフトウェアエージェントを構成要素とする分散処理システム，ADIPS: Agent-basedDistributed Information Processing Systemの研究開発を行なって来た[2]. ADIPSを実現するソフトウェアエージェント(以下，ADIPSエージェントと表記)は，分散システムを構成するモジュールの設計知識と，ADIPSで定義された分散システム組み上げのためのプロトコルに基づいて，利用者要求を充足する分散システムを構成し，計算機環境変化，利用者要求変化に対して必要な動作を実現する．

ADIPSは分散システムを自律的に構成し利用するという目的で設計されたため構成されたADIPS間の協調を実現することが困難であった．これはADIPS同士の協調実現が個別ADIPSを実現する設計知識記述に依存しているために，設計者の意図しない協調を実現することが，ADIPSエージェントの設計知識記述者にとって負担となっていたためである．

そこで本稿では異なる利用者要求を充足するために実現されたADIPS間の協調を実現するために，これまでad hocに実現されてきた協調プロトコルの形式的な定義を与えることを目的として，ADIPSフレームワークでのソフトウェアエージェント間協調プロトコル実現手法を提案する．

2　ADIPSフレームワーク

2.1　ADIPSフレームワークの特徴

ADIPSフレームワークは，利用者要求を受け取り目的の分散システムを構成する複数のADIPSエージェントの動作の場であるリポジトリと，構成された分散システムを制御監視するADIPSエージェントの動作の場である動作環境から構成される．ADIPSフレームワークでは，ある利用者要求を充足する分散システムは，これらの機能要素からなる複数のADIPSエージェントと計算機プロセスにより実現されるエージェント組織として，リポジトリ上で設計され動作環境上に動的に生成される．ADIPフレームワークの概念図を図1に示す．

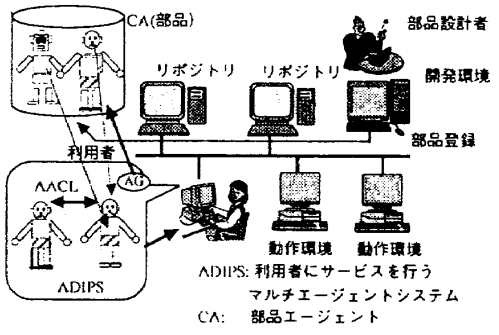

図1: ADIPSフレームワーク

ADIPSフレームワークに基づく分散システムは，構成要素であるADIPSエージェントが設計知識に基づいて動作することで，次の4つの特徴を持つ．(1)利用者要求駆動による自動システム構成．(2)障害発生などのイベント駆動による自律的システム再構成．(3)自律的システム構成・再構成を目的としたエージェン

A Method to Implement Cooperation Protocols on The ADIPS Framework.
Shigeru Fujita†, Hideki Hara‡, Kenji Sugawara‡, Tetsuo Kinoshita††, Norio Shiratori††
†Department of Computer Science, Chiba Institute of Technology
2-17-1, Tsudanuma, Narashino 2750016, JAPAN
‡Department of Network Science, Chiba Institute of Technology
2-17-1, Tsudanuma, Narashino 2750016
††Research Institute of Electrical Communication, Tohoku University
2-1-1, Katahira, Aoba-ku, Sendai 980-8577, JAPAN

トによる設計者知識利用．(4) エージェント化により既存分散システムを構成する計算機プロセス／ソフトウェアモジュールの系統的な再利用．

計算機プロセス／ソフトウェアモジュールのエージェント化とは既存分散処理システムの構成要素である計算機プロセス／ソフトウェアモジュールに対して，ソフトウェアエージェントによる制御監視機構を追加し，設計知識に基づく自律的動作を可能にすることを示す．

ADIPS エージェントのモデル図を図 2に示す．分散システム設計知識は領域知識ベース (Domain Knowledge-base:DK) に蓄積，利用される．分散システムを構成するためのADIPS構成プロトコル(ADIPS Assembly Protocol *) は，協調機構 (Cooperation Mechanism: CM) により処理され，DK 上の知識を利用して利用者要求を充足する分散システムを構成する．分散システムを実現する個別の計算機プロセスの制御監視は，タスク処理機構 (Task Processing Mechanism: TPM) により行われる．

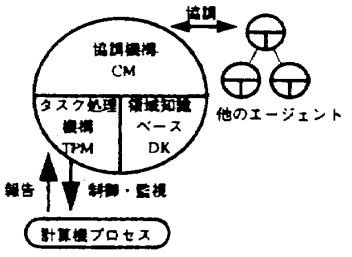

図 2: ADIPS エージェントモデル

2.2 協調プロトコル実現手法

エージェントに協調プロトコルを実現させる手法として文献 [3] で述べられているエージェントの心的状態を，ADIPS エージェントの領域知識ベース内で実現し，これまで定義されてきたADIPS 構成プロトコルの中で，協調を実現するために領域知識ベースの拡張を行う．本手法は既存の ADIPS エージェントの知識記述言語の拡張として実現され，ADIPS 構成プロトコルの中で定義されている情報通知 (tell) と問い合わせ (ask) メッセージの中に，協調を目的とした記述を行い，異なる目的のために構成された ADIPS 間での協調を実現する．

図 3に拡張された領域知識ベースの構造を示す．図中のプロダクションシステムを利用している箇所が本提案に基づいて設計されたADIPS エージェントの拡張である．

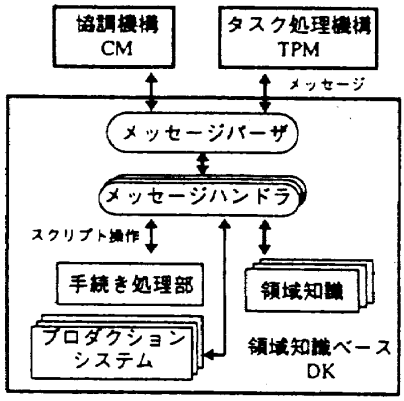

図 3: 拡張領域知識ベース

3 おわりに

本稿では ADIPS フレームワークに基づいて構成された分散システム間の協調プロトコルを定義するための実現手法として，ADIPS エージェントの領域知識ベースを拡張する手法について述べた．本手法は Java, JESS[4] を用いて実装が行われ，協調プロトコルの記述とその評価を行っている．

ADIPS 構成プロトコルは分散システム構成を目的としているために，領域知識ベースの拡張のみでは，分散システム間の協調を実現できない場合が生じる．例えば，分散システム間で明確な上下関係を動的に実現し一方の要求を必ず実現するようなエージェント組織を動的に構成する場合や，設計知識を記述している設計者間の背景知識の差異による競合を解消する場合である．このため，ADIPS 構成プロトコルそのものを実現している協調機能拡張を行う必要があり，領域知識ベースの拡張と同様に実装と評価を行う予定である．

参考文献

[1] 石田: "エージェントを考える", 人工知能学会誌, Vol.10, No.5, pp.663-667, 1995.

[2] 藤田, 菅原, 木下, 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, Vol.37, No.5, pp.840-852, 1996.

[3] Yoav Shoham. "Agent-oriented programming", Artificial Intelligence Vol. 60, pp. 51-92, 1993

[4] JESS, the Java Expert System Shell, http://herzberg.ca.sandia.gov/ jess/, (6/25/1998 現在)

*従来 ADIPS フレームワーク [2] では，"拡張契約ネットプロトコル" として定義していた．