

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

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開発のための枠組みである ADIPS フレームワークにおけるエージェントは、基本機能要素を実現するプリミティブエージェントと、これらを組み合わせてサブシステムを実現するための組織構成エージェントに分類される。本論文では、組織構成エージェントの領域知識の記述形式を提案し、その処理系を与える。さらに、提案した手法を利用したアプリケーションの試作を通して、エージェント指向分散処理システム設計を系統的に行うことができ、エージェント設計者の負担を大きく軽減できることを確認し、提案した手法の有効性を示す。

Domain Knowledge Description Forms for Organizational Agents of Agent-based Distributed Information Processing System, ADIPS

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

Agents defined in the ADIPS framework are classified into primitive agents and organizational agents. The primitive agents realize functional components for distributed processing systems, and the organizational agents compose an organization of agents to realize an distributed processing system, using primitive agents. This paper propose an effective method of knowledge description and its processing mechanism for organizational agents. Using the proposed method, prototype systems have been developed. We confirm that systematic design and development of agent-based distributed systems can be attained, and the burden of the designers can also be reduced.

1. はじめに

これまでに、利用者要求の多様化/高度化に対応するための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)}などの研究が進められているが、利用者要求駆動で動的にサービスを提供するための分散処理システムを効果的に開発する手法は確立されていない。この問題に対して、“エージェント”^{5)~9)}の概念を用いた自律的な部品から構成される分散処理システムのアーキテクチャ:ADIPSが提案されている¹⁰⁾。

ADIPS フレームワークでは、分散処理システムを

開発するための機能要素を実現するプリミティブエージェントと、これらを組み合わせてサブシステムを実現する組織構成エージェントの2種類のエージェントの組織として分散処理システムを動的に構成することができる。

ところが、ADIPS フレームワークの実用化を目指すうえで、以下の機能が提供されていないためエージェントを開発する際の設計者の負担が大きいことが問題になっていた。すなわち、

- (1) 組織構成エージェントの領域知識を記述する指針が与えられておらず、その分析とモデル化が個々の設計者の視点に基づいて行われていたため、系統的に知識記述を行うことが困難であった。
- (2) 領域知識の記述方式が ADIPS フレームワークの実装言語に依存していたので、知識を記述したり、その内容を把握したりすることが困難であった。
- (3) 領域知識の記述形式が統一されていなかったので、エージェント組織の要素となるエージェントを領

[†] 千葉工業大学情報工学科
Department of Computer Science, Chiba Institute of Technology

^{††} 東北大学電気通信研究所
Research Institute of Electrical Communication, Tohoku University

域知識を用いて効率的に検索することが困難であった。

そこで本論文では、上記の問題に対して次の解決策を与える。

- (1) 分散処理システムの構成要素となる組織構成エージェントの役割や機能を分析し、その領域知識を6種類に分類し、その記述内容と相互関係を明確化した。
- (2) フレームワークの実装言語に依存しない、領域知識記述形式とその処理機構を与えた。
- (3) 6種類の領域知識ごとに標準的な知識記述テンプレートを導入し、領域知識の記述形式を統一した。

これによって、領域知識記述の指針が与えられ、設計対象となる分散処理システムごとに、新たに領域知識記述形式を定義する作業が削減され、分類された領域知識の内容を系統的に獲得、記述できるようになり、エージェント設計者の負担が大きく軽減される。

以下、2章ではADIPSフレームワークの概要を述べる。3章では、我々が提案してきたADIPSフレームワークで定義されているエージェントの領域知識の分類に基づき領域知識の表現形式を定義し、その処理機構を提案する。4章では、3章で定義された領域知識ごとの知識記述テンプレートを提案する。5章では、提案手法を用いた組織構成エージェントの設計例を示し、エージェント指向分散処理システム設計における提案手法の有効性を示す。

2. ADIPS フレームワーク

2.1 ADIPS フレームワークの概要

ADIPS フレームワークの基本的アーキテクチャは、文献10)で提案されている。これは図1に示すように、クラスエージェントを格納するエージェントリポジトリ(以下、リポジトリと略す)と、利用者にサービスを提供するインスタンスエージェントのためのエージェント動作環境(以下、動作環境と略す)から構成される。

クラス/インスタンスエージェントは、分散処理システムの基本機能要素を実現するプリミティブエージェントと、これらを組み合わせてサブシステムを実現するための組織構成エージェントに分類される。

2.2 プリミティブエージェント

プリミティブエージェントは、分散処理システムの機能を実現する計算機プロセスに対して、これを管理するタスク処理機構、協調機構と領域知識ベース機能を付加して構成される¹⁰⁾。計算機プロセスは、既存プロセスを再利用する場合と、新規に作成する場合とに

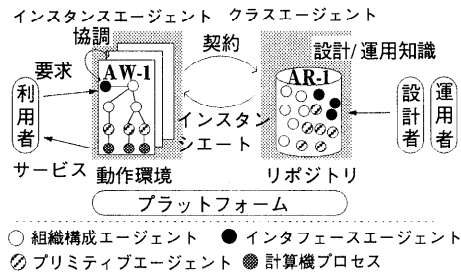


図1 ADIPS アーキテクチャ

Fig. 1 ADIPS architecture.

分けられ、領域知識ベースにはこれらの計算機プロセスの機能仕様、運用・調整のための方法・手順などに関する記述が格納されている。プリミティブエージェントはこれらの知識を用いて、組織構成エージェントとの契約に基づいて動作環境上に生成され、分散処理システムの構成要素として動作する。本プリミティブエージェントの知識記述テンプレートについては、稿をあらためて報告したい。

2.3 組織構成エージェント

一方、ADIPS フレームワークにおける組織構成エージェント^{*}の役割は、プリミティブエージェントを用いて、分散処理システムまたは、分散処理システムの構成要素であるサブシステムを構成することである。本論文では、3章において、この組織構成エージェントの知識表現モデルを定式化し、これに基づく知識記述テンプレートを4章で提案する。

3. 組織構成エージェントの領域知識表現形式と処理系

3.1 組織構成エージェントの領域知識表現モデル

ADIPS フレームワークにおけるエージェントの構成モデルを図2に示す。この構成モデルは、ADIPS フレームワークで提案してきたエージェントの構成モデル¹⁰⁾を今回提案する知識記述テンプレートを処理するために、領域知識処理を行う知識ベースの内部構造を詳細化したモデルである。

組織構成エージェントが制御するエージェントを、そのメンバエージェントと呼ぶ。本論文で対象とする組織構成エージェントに特有の基本的領域知識の一覧はADIPS フレームワークで提案されており¹⁰⁾これを表1に示す。

また、図3にエージェントの動作に関する知識の典型的な利用法を示す。図3について、以下で説明する。

^{*} 文献10)では、組織エージェントと呼称したが、本論文では組織構成エージェントと呼ぶことにする。

表1 タスク領域知識ベース DK の内容—組織構成エージェント

Table 1 Contents of task domain knowledge-base DK—Organizational agent.

知識名	知識の内容	エージェントの種別	制御する内部機構
TASK	機能、性能、仕様、資源などエージェントの能力表現の知識	クラス	CM
PROC	与えられた機能や性能を実現するための機能分割・詳細化の知識	クラス	CM
BID	入札作成、TPM 初期化知識	クラス/インスタンス	CM
RECOV	TPM から制御/監視されているエージェント (メンバエージェント) からのレポートを分析し、回復手段を計画実行する知識	インスタンス	TPM
CHANGE	RECOV 知識では処理不能な変化を分析し組織構成エージェントの動作を決定する知識	インスタンス	CM
INTER	交渉、依頼の手順計画作成、評価知識、被動作要求時の処理知識	インスタンス	CM

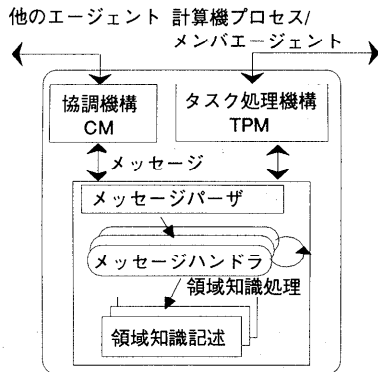


図2 エージェントの構成

Fig. 2 Architecture of agent.

(A) 組織構成

利用者要求は、動作環境上に存在する利用者エージェントにより、“タスク通知”に変換され、リポジトリにブロードキャストされる。この“タスク通知”を受信することにより、エージェントは次の動作を開始する。

“タスク通知”に対して、エージェントの実現可能なタスクの仕様を記述する TASK 知識によりタスク実行が可能であると判断され、そのタスクの仕様を実現するための手続き、方法を記述する PROC 知識により組織構成が行われ、構成された組織の情報を元に、要求されたタスクを実現するために必要な条件や、その条件下で提供可能な機能や性能を提示するための知識を記述する BID 知識により入札が作成されている。

(B) 環境変化に対するメンバエージェントのパラメータ調節による対応

利用者要求や動作状況の変更は、それらを監視するメンバエージェントから“レポート通知”により伝達され、エージェントがそれらの状況の変化に対応する方法を記述した知識である RECOV 知識を用いて、TPM から制御されているメンバエージェントに対してパラメータ調節などの動作指示を行う。

RECOV 知識は、メンバエージェントより伝達され

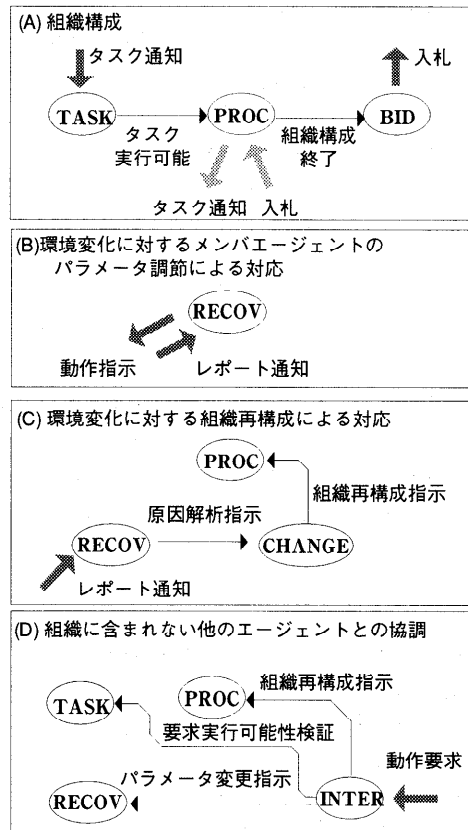


図3 組織構成エージェントの知識間の関係

Fig. 3 Relation of the classes of knowledge of organizational agent.

たレポート通知を解析し、メンバエージェントへの指示を出している。

(C) 環境変化に対する組織再構成による対応

環境変化がメンバエージェントから“レポート通知”により伝達され、RECOV 知識により分析された結果、メンバエージェントのパラメータ調節では対処することが不可能であることが報告されたとき、CHANGE 知識が、“原因解析指示”によりその原因を大域的に分析する。その結果メンバエージェントの再構成(変換、

追加)を行うことを計画し、その実現を行う PROC 知識に“組織再構成指示”を行う。

(D) 組織に含まれない他のエージェントとの協調 (B), (C) は (A) の組織構成によって作られた分散処理システム ADIPS の組織内のエージェントの協調による ADIPS 機能変更を実現している。しかし, (B), (C) では, 他の組織に属するエージェントとの協調は定義されていない。環境変化を分析する CHANGE 知識が他の組織のエージェントとの交渉が必要であると判断したとき, “交渉要求”を INTER 知識に伝達する。INTER 知識は協調機構 CM を通じて, 交渉相手の INTER 知識に“動作要求”を伝える。交渉の“動作要求”を受理した INTER 知識は, その判定を行うために, TASK, RECOV, PROC 知識に“要求実行可能性検証”指示等を出す。

以上組織構成エージェントの動作は, (A)~(D) で示したような, 領域知識を利用して実行される。

3.2 組織構成エージェントの領域知識処理系

タスク領域知識ベース DK は, 図 2 に示すエージェントの構成の中で, メッセージパーザ (以下, パーザと略す), メッセージハンドラ (以下, ハンドラと略す), 領域知識記述から構成される。

パーザは, 協調機構 CM, タスク処理機構 TPM から伝えられるメッセージを, 適切なハンドラに伝達し, またハンドラより伝えられるメッセージをそれぞれの処理機構へ伝達する。

ハンドラは, 表 1 に示した各知識名に対応した知

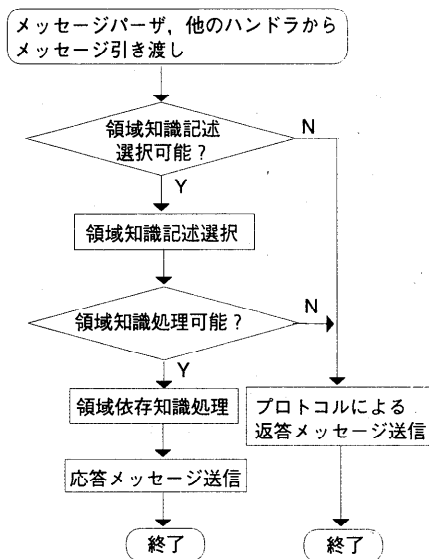


図 4 メッセージハンドラの知識処理手続き

Fig. 4 Procedure of knowledge processing in message handler.

識処理モジュールである。ハンドラは, パーザからのメッセージ引渡し, または他のハンドラからの, メッセージ, 領域知識の識別子の引渡しによって, 動作を開始する。

ハンドラは, 引き渡されたメッセージにより, 領域知識記述の動作条件が充足した場合, 4 章で述べる動作を開始する。動作条件が充足されない場合, エージェント間プロトコルに基づいて, 適切な応答メッセージを返送する。

たとえば, 直接落札によるタスク処理要求が行われ, これに対する領域知識の記述がない場合には, ハンドラの処理により, タスク処理要求拒絶メッセージが生成され, タスク処理要求の送信元へ返送される。

メッセージハンドラの知識記述処理手続きを図 4 に示す。

3.3 組織構成エージェントの領域知識記述形式

組織構成エージェントの領域知識を記述するためのエージェント知識表現形式 AGT を図 5 に示すように定義する。ただし, 大文字は非終端記号, 小文字は終端記号, 斜体は別に定義/記述される文字列, [] は 0 個または 1 個の出現, {} は 0 個以上の並びを表す。

プリミティブエージェントの記述である, PADS (Primitive Agent Definition Script) の詳細については稿をあらためて報告する。

OADS (Organizational Agent Definition Script)

ACT	:=	(primitive <i>agtid</i> PADS { PADS }).
		(organizational <i>agtid</i> OADS { OADS }).
OADS	:=	(CONT { RECV } { CHNG } { INTER }).
CONT	:=	TASK PROC BID { TASK } { PROC } { BID }.
TASK	:=	(task <i>id</i> MSG [INFO] TSCR).
PROC	:=	(proc <i>id</i> [INFO] PSCR).
BID	:=	(bid <i>id</i> [INFO] BSCR).
RECV	:=	(recov <i>id</i> MSG [INFO] RSCR).
CHNG	:=	(change <i>id</i> [INFO] CSCR).
INTER	:=	(inter <i>id</i> MSG [INFO] ISCR).
MSG	:=	(message MESH { MESH }).
INFO	:=	(req-info MESH { MESH }).
MESH	:=	(<i>pstext-name</i> WORD { WORD }).
WORD	:=	(ARG { ARG }).
ARG	:=	WORD STRG VARB.
STRG	:=	' <i>alphanumeric-string</i> '.
VARB	:=	? <i>alphanumeric-string</i>
		! ? <i>alphanumeric-string</i>
		! <i>alphanumeric-string</i> .

図 5 エージェントの領域知識記述形式

Fig. 5 Domain knowledge description forms of organizational agent.

は組織構成エージェントの領域知識記述を示している。OADSに含まれる、TASK, PROC, BID, RECV, CHNG, INTERにより表現されている知識は、それぞれ表1および、図3に示した領域知識表現である。TASK, PROC, BID, RECOV, CHANGE, INTERに対応する。

agtid は、エージェント自身の識別子である。*id* はそれぞれの領域知識記述部分の識別子である。*pstext-name* は契約ネットプロトコル¹¹⁾、拡張契約ネットプロトコル¹⁰⁾のメッセージ名が記入される。

alphanumeric-string はそれぞれのエージェントが処理する、タスクごとの領域依存の宣言記述、処理実現のための手続き記述を行う。*?alphanumeric-string* はハンドラにより、他の領域知識記述、メッセージと置き換えられる変数を、*!alphanumeric-string* は置き換えられた変数を利用して、他のエージェントに伝達する項目を、*!alphanumeric-string* はハンドラ内の予約語を示している。

4. 知識記述テンプレート

契約ネットプロトコル¹¹⁾では、エージェント間のメッセージの形式は定義されているが、そのメッセージを処理するための知識記述方式は規定されておらず、応用領域ごとに処理知識/手続きを記述する形式と処理方式を設計する必要がある。たとえば、契約ネットプロトコルを応用したシステムの研究事例である分散センシングの研究¹²⁾においても、メッセージを処理するための知識処理法はその特定領域に限られており、一般領域に関する知識記述手法は具体的に与えられていない。

これに対して本論文では、ADIPSフレームワークで提供される組織構成エージェントに与えられるすべての種類のメッセージを処理し、構成部品としての機能を実現するための領域知識を6種類(TASK, PROC, BID, RECOV, CHANGE, INTER)に分類し、それぞれの種類の知識を記述するために必要な項目を網羅した記述テンプレートを定義した。これにより組織構成エージェントの開発プロセスにおける開発者の負担を軽減することが可能になる。以下それぞれの記述テンプレートの内容と、これが表現している領域知識の目的について述べる。

4.1 TASK 知識記述テンプレート

組織構成エージェントのTASK知識は、協調的問題解決組織の構成要素の機能仕様に関する知識である。このTASK知識記述テンプレートは、エージェント設計者の持つTASK知識を自分の機能の中で必要な

TSCR	:=	FUNC SPEC [PERF] [RESO] [REFU].
FUNC	:=	(function STRG).
SPEC	:=	(specification WORD).
PERF	:=	(performance WORD).
RESO	:=	(resources WORD).
REFU	:=	(send-refusal WORD).

図6 TASK 知識記述形式

Fig. 6 TASK-knowledge description form.

問題解決機構を持たないエージェントに対して、その問題解決機能を提供可能かどうかを判定するための項目とその記述形式を規定したものである。

このTASK知識記述テンプレートは、テンプレート自体の識別子を記述する*id*、エージェントが受諾可能でありうるメッセージを記述する*message*、他のエージェントに対する情報要求を行う知識を記述する*req-info*、機能仕様を表現するTSCR (Task SCRIPT) から構成される。このTSCRは、機能の識別子を示す*function*、実現する機能の仕様を記述する*specification*、実現する機能の能力/性能と資源との制約を記述する*performance*、機能実現のために必要な資源を記述する*resources*、直接落札に対して詳細な拒絶通知を生成する際の知識を記述する*send-refusal*により構成されている。

このTASK知識記述テンプレートの主要な部分であるTASK知識記述形式TSCR (Task SCRIPT)の定義を図6に示す。

4.2 PROC 知識記述テンプレート

組織構成エージェントのPROC知識は、TASK知識によって実行可能であると判断されたタスクを実行するために、タスク分割を行い、分割されたタスクを実行するエージェントを検索するための知識である。PROC知識記述テンプレートは、エージェント設計者の持つPROC知識を表現するために必要な項目とその記述項目を規定したものである。

id, *req-info*, および、組織構成/再構成を行う知識を表現するPSCR (Proc SCRIPT) から構成される。このPSCRは、問題分割を行うためのタスク通知、直接落札を記述する*task-announcement*, *directed-award*, タスク通知に対する入札を評価するための*eval-bid*, 直接落札に対して拒絶通知が送られたときの処理を記述する*rec-refusal*により構成されている。

このPROC知識記述テンプレートの主要な部分であるPROC知識記述形式PSCR (Proc SCRIPT)の定義を図7に示す。

4.3 BID 知識記述テンプレート

組織構成エージェントのBID知識は、TASK知識

```

PSCR := TDMS EVLW [ EXPW ].
TDMS := TMSG | DMSG | TMSG DMSG .
TMSG := ( task-announcement MMSG
         { MMSG } ).
DMSG := ( directed-award MMSG { MMSG } ).
EVLW := ( eval-bid WORD ).
EXPW := ( rec-refusal WORD ).

```

図7 PROC 知識記述形式

Fig. 7 PROC-knowledge description form.

```

BSCR := BIDW [ AWD ] INIT.
BIDW := ( make-bid WORD ).
AWD := ( award WORD ).
INIT := ( init-member WORD ).

```

図8 BID 知識記述形式

Fig. 8 BID-knowledge description form.

により実現可能性を判定し PROC 知識によりその実現方法および条件を確定したタスクに対して、このタスクを要求した組織構成エージェントに対するタスク通知への入札を作成し、この入札時の条件を保持し、落札が行われた場合に、分割された部分タスクを実行する、各メンバエージェントへの落札通知、初期化情報を生成するための知識である。この BID 知識記述テンプレートは、これらの TASK 知識を表現するための項目と、その記述項目を規定したものである。

この BID 知識の主要な部分 BSCR (Bid SCRipt) は、入札を作成する知識を記述する **make-bid**、落札に対して問題分割の対象となったエージェントへの落札を生成するための知識を記述する **award**、インスタンス生成時に問題分割の対象となったエージェントに初期値を与える知識を記述する **init-member** により構成されている。

この BID 知識の主要な部分 BSCR (Bid SCRipt) の定義を図 8 に示す。

4.4 RECOV 知識記述テンプレート

組織構成エージェントの RECOV 知識はメンバエージェントから送られてくるメッセージである中間報告 interim-report, 最終報告 final-report, 消滅報告 dissolve を分析し、目的とする動作を組織全体で実現するために必要な動作を実現する知識であり、この RECOV 知識では、メンバエージェントの構成を変更するための知識が表現されない。RECOV 知識記述テンプレートは、組織構成エージェントでのメンバエージェントからのメッセージを処理するために必要な項目とその記述形式を規定したものである。

RECOV 知識による動作では、(1) メンバエージェントにパラメータ変更等の動作変更を要求する、(2) CHANGE 知識に対して分析した結果を付加して処理

```

RSCR := ( WORD ACMS ).
ACMS := RQACT | ( change id MMSG )
       | RQACT ( change id MMSG ).
RQACT := ( make-req WORD { GTACPT }
         { GTREFL } { GTCR } ).

```

RQACT 以降の詳細定義は INTER 知識での定義と同じ。

図9 RECOV 知識記述形式

Fig. 9 RECOV-knowledge description form.

を依頼する、のいずれかの動作を行う。

この RECOV 知識記述テンプレートの主要な部分である RSCR (Recov SCRipt) は、部分問題を担当しているメンバエージェントに対して動作要求を行うための知識を記述する **make-req**、問題解決組織の変更を含む決定を行うための判断を行う **change** により構成される。項目 **make-req** の詳細項目定義は後述する INTER 知識に同じである。

この RECOV 知識記述テンプレートの主要な部分である RECOV 知識記述形式 RSCR (Recov SCRipt) の定義を図 9、図 11 に示す。

4.5 CHANGE 知識記述テンプレート

CHANGE 知識は RECOV 知識により環境変化への対応が実現できなかった場合にメンバエージェントの構成を変更する、メンバ以外のエージェントとの協調により環境変化への対応を実現する知識である。CHANGE 知識記述テンプレートは、組織構成エージェントが他のエージェントとの協調により目的の処理を実行できるかの判断をするための項目とその記述形式を規定したものである。

この CHANGE 知識記述テンプレートの主要な部分である CSCR (Change SCRipt) は、メンバエージェントの一部入れ換えや、新規メンバエージェントの導入を実現するための組織再構成を実現するための知識である組織再構成知識 **re-organization**、より上位の組織構成エージェントに対して、環境変化と自らの状態の通知を行うための状態通知生成知識 **make-interim-report**、最終的な結果の通知生成知識 **make-final-report**、他のエージェントとの協調を実現するための交渉開始知識 **request-action**、および、エージェントが動作継続不能な状態に陥ったときの後処理と消滅報告の生成を実現する知識 **dissolve** により構成される。

この CHANGE 知識記述テンプレートの主要な部分である CHANGE 知識記述形式 CSCR (Change SCRipt) の定義を図 10 に示す。

4.6 INTER 知識記述テンプレート

組織構成エージェントの INTER 知識は組織構成エージェントが他の ADIPS を構成している組織構成

```

CSCR := ( analysis WORD JUNC { JUNC } ).
JUNC := REORG | REQA | IREP | FREP
      | DISL.
REORG := ( re-organization id [ WORD ]
          ( proc-id id ) ).
REQA  := ( request-action id [ WORD ]
          ( inter-id id ) ).
IREP  := ( make-interim-report id WORD ).
FREP  := ( make-final-report id WORD ).
DISL  := ( dissolve id
          { ( make-stop-release WORD ) }
          [ ( make-dissolve WORD ) ] ).

```

図 10 CHANGE 知識記述形式

Fig. 10 CHANGE-knowledge description form.

エージェントとの間で協調を実現するための知識であり、主要な知識は、他のエージェントへの要求を伝達する知識、他のエージェントからの要求を処理するための知識である。INTER 知識記述テンプレートはこの他のエージェントとの協調を処理するための INTER 知識の表現に必要な項目とその記述形式を規定したものである。

INTER 知識記述テンプレートは、メンバエージェント以外のエージェントに対して動作要求を行うための知識を記述する **make-req**、この組織構成エージェントに対して動作要求が行われたときの知識を記述する **make-ans**、この組織エージェントに対して情報要求がなされたときの知識を記述する **requested-info** より構成される。**make-req** は、動作要求が受諾されたときの知識記述 **get-acceptance**、動作要求が拒絶されたときの知識記述 **get-refusal**、動作要求に対して代案が提示されたときの知識記述 **get-cr**、代案を提示するための知識 **make-counter-request** に詳細定義される。**get-refusal** では、再度動作要求を行うための知識記述 **re-request-action** が定義される。**make-ans**、**get-cr** に対しては受諾/拒絶する知識記述 **acceptance**、**refusal** および、それぞれの応答メッセージを生成する知識記述 **make-acceptance**、**make-refusal** と、**make-counter-request** が定義される。

この INTER 知識記述テンプレートの主要な部分である INTER 知識記述形式 ISCR (Inter SCRipt) の定義を図 11 に示す。

5. 領域知識記述に基づく応用システムの試作

5.1 利用者要求に基づくシステム構成例

本章では、4 章で定義した、ADIPS フレームワークにおける組織構成エージェントを実現する 6 種類の領域知識を記述するための知識記述テンプレートが有

```

ISCR := RQACT | RQDACT | RQDINF.
RQACT := ( make-req WORD { GTACTP }
          { GTREFL } { GTCR } ).
RQDACT := ( make-ans WORD { ACPT }
           { REFL } { CRQACT } ).
RQDINF := ( requested-info MSG { MSG }
           WORD ).
GTACTP := ( get-acceptance WORD ).
GTREFL := ( get-refusal WORD
           [ ( re-request-action WORD ) ] ).
GTCR   := ( get-cr [ INFO ] WORD
           { ACPT } { REFL } { CRQACT } ).
ACPT   := ( acceptance id WORD
           ( make-acceptance WORD ) ).
REFL   := ( refusal id [ WORD ]
           ( make-refusal WORD ) ).
CRQACT := ( make-counter-request id [ INFO ]
           WORD ).

```

図 11 INTER 知識記述形式

Fig. 11 INTER-knowledge description form.

効であることを、具体的な分散処理システムを構成する組織構成エージェントを実現することにより示す。本例題は、“WWW-Browser を利用したい” という利用者要求に対して、リポジトリ上のクラスエージェントの知識記述テンプレートを用いて、記述された領域知識記述に基づいて動作環境上に目標とするシステムが生成されることを実験により確認した。

具体的には、利用者からの要求は利用者インタフェースから、利用者インタフェースエージェント (User-I/F-Agent) を通じて、リポジトリに伝達される。この過程で、利用者要求は拡張契約ネットプロトコルにおけるタスク通知に変換される。この場合は、WWW エージェントに対して直接落札によりタスクを依頼する。

WWW エージェントは、User-I/F-Agent からの直接落札が実行可能であることを、TASK 知識を用いて確認する。このとき用いられた図 6 に示した TASK 知識記述テンプレートによる TASK 知識の記述を図 12 に示す。

その後、PROC 知識を用いて Socket, HTML-handler, Graphic, Text, Window のそれぞれのエージェントに対して直接落札を用いてタスクを依頼する。このとき用いられた図 7 に示した PROC 知識記述テンプレートによる PROC 知識記述を図 13 に示す。

同様に再帰的に、GIF デコード機能を持ったプリミティブエージェント、テキスト表示機能を持ったプリミティブエージェントまでタスク依頼が分割して伝播され、利用者要求を充足するための WWW-Browser が動的に構成される。このとき、構成される WWW-Browser が画像形式、テキスト形式の表示のみを利用

```
(task WWW01
  (message
    (task-announcement
      (application 'WWW-Browser'))
    (directed-award
      (application 'WWW-Browser'))))
  (req-info (request-information
    :to !sender
    (network-if !?if-type)))
  (function 'WWW-Browser')
  (specification
    (if ((equal ?if-type '10Base)
      (proc-id proc-10Base))
      ((equal ?if-type 'tty')
        (proc-id proc-tty))
      (else (proc-id proc-other))))
  (resources (network-protocol ip)))
```

図 12 TASK 知識記述例

Fig. 12 Example of TASK-knowledge.

```
(proc proc-10Base
  (directed-award
    (directed-award :to Socket
      (function 'TCP'))
    (directed-award :to HTML-handler
      (function 'HTML2'))
    (directed-award :to Graphic
      (function 'Gif-Jpeg'))
    (directed-award :to Text
      (function 'html-text'))
    (directed-award :to Window
      (function 'Text-Graphic-color'))))
  (eval-bid
    (bid-id simple-WWW)))
```

図 13 PROC 知識記述例

Fig. 13 Example of PROC-knowledge.

しているのは、“WWW で提供されるデータ形式としては、画像形式、テキスト形式が中心である”というエージェント開発者の経験的知識に基づいている。この結果、利用者に対して提供された WWW-Browser の画面を図 14 (左) に示す。

この実験の結果、組織構成エージェント記述テンプレートにより、動的な分散処理システム構成を実現する ADIPS フレームワークのためのエージェント記述が行えることを確認した。

5.2 状況に応じたシステム再構成例

状況の変化に対して自律的に適応するための、組織構成エージェントの知識記述が実現されているかを、上述した WWW-Browser の例を用いて実験により確認した。

5.1 節では、利用者要求が単に“WWW-Browser”というものであったので、WWW-Browser の機能としては、GIF 形式による画像形式データの表示とテキスト形式データの表示のみを実現していた。このとき、

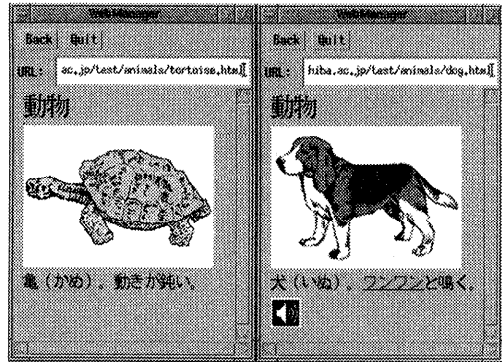


図 14 利用者要求に基づき構成された WWW-Browser の例 (左) 組織再構成による状況変化への対応例 (右)

Fig. 14 Example of WWW-Browser based on user requirement (left side).

Example of adaptation to change of situation by re-organization (right side).

```
(change new-agent-request
  (analysis
    (if ((equal (value (lmsg cause)) unknown)
      (re-organization new-agent01))
      (else (make-interim-report i-rep02))))
  (re-organization new-agent01
    (proc-id request-agt))
  (make-interim-report i-rep02 (i-repo unknown)))
```

図 15 CHANGE 知識記述例

Fig. 15 Example of CHANGE-knowledge.

HTML-handler は処理するエージェントが特定できないデータ形式として、サウンド形式を検出した。

このあと HTML-handler エージェントから、サウンド形式のデータが検出されたことが、WWW エージェントに伝達される。ただし、WWW エージェントの知識記述にはサウンド形式を処理するためのエージェントを特定するための知識がないために、最初に HTML-handler からのレポート通知を処理する RECOV 知識ではこの通知を処理することができない。このため WWW エージェントは、“形式不明なデータ”すなわち“処理可能なデータ形式に含まれないデータ形式の処理が要求された”として判断する。

この“形式不明なデータ”への対応を決定するために、図 10 に示した CHANGE 知識記述テンプレートによる WWW エージェントの CHANGE 知識の記述例を図 15 に示す。この CHANGE 知識により、WWW エージェントは“形式不明なデータ”であるサウンド形式のデータを処理するエージェントを利用するために WWW-Browser の組織再構成を行うことを決定する。

WWW エージェントはリポジトリに対して、必要

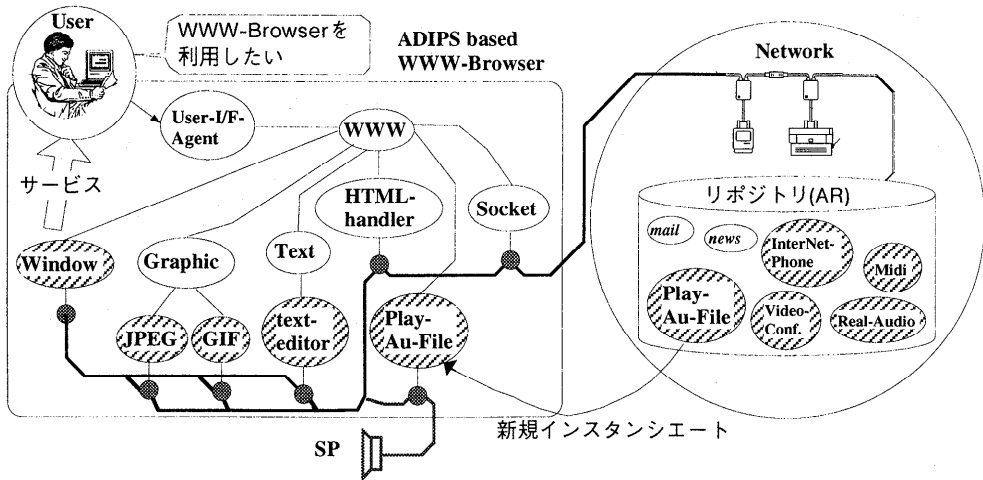


図 16 ADIPS による WWW-Browser
Fig. 16 WWW-Browser based on ADIPS.

な機能を有するエージェントを要求するためのタスク通知を発行する。

この WWW エージェントからのタスク通知に対して、リポジトリ内のクラスエージェントの中で、サウンド形式のデータを処理可能なエージェントが入札を行い、WWW エージェントが適切なエージェントを選び、落札を与える。

この結果、タスク通知において指定された AU 形式のサウンドファイルをデコードできるプリミティブエージェント Play-Au-File エージェントが図 16 に示すように、それまで動作していた WWW-Browser の内部にメンバエージェントとして追加され、図 14 (右) に示すように、サウンド形式のデータを利用者 に提供する WWW-Browser が新たに生成された。

この実験により、状況の変化に対応して分散処理システムの構成要素の変更を含む適応動作を実現する ADIPS フレームワークのための、組織構成エージェントを、組織構成エージェントの知識記述テンプレートにより記述できることを確認した。

5.3 評価

本論文では、提案した組織構成エージェントの知識記述テンプレートの評価を行うために、ワークステーション (Sun Ultra-Sparc) 上で動作する Java 言語動作環境 (Sun Soft JDK1.1.1) を用いて、ADIPS フレームワークを実現し、組織構成エージェントの知識記述テンプレートによりエージェントの知識記述を行った。この知識記述の対象は、4 章で述べた WWW-Browser である。この実験から、組織構成エージェントに要求される知識、すなわち、

- 拡張契約ネットプロトコルに基づいて、利用者環境上に分散処理システム/サブシステムを構成する知識、
- 状況変化を検知し、記述された設計/運用知識に基づいてサブシステムの構成要素を追加/変更する知識、

が 4 章で提案した知識記述テンプレートを用いて記述できること、および、これに基づいて、所定のエージェント動作が実現できることを実証した。

これまで、ADIPS フレームワークで提供するプロトコルおよび状態遷移¹⁰⁾に基づいて、応用システムごとに領域知識の表現モデルを設計し、同じく ADIPS フレームワークで提供するエージェントの領域知識を記述する基本的言語を用いてエージェントの実装を行ってきた¹³⁾。しかしながら、こうした手法は、具体的にエージェントを設計/実現する際に、それぞれの領域知識と記述形式を設計する作業が必要となるため、エージェントの開発者に対して大きな負担を与えるものであった。

本論文で提案した新たな手法は、組織構成エージェントの各領域知識を表現する際に利用される知識記述テンプレートである。エージェントを設計/実装する技術者はこのテンプレートの構造に従って組織エージェントの知識を系統的に抽出、記述することができ、リポジトリに存在するプリミティブエージェント/組織構成エージェントを組み合わせて新たな組織を生成する組織構成エージェントの設計が容易に行えることが実証された。すなわち、従来、ADIPS フレームワークに基づくエージェントの開発工程は、(1) エージェントに

記述する知識の獲得, (2) 知識記述形式の定義, (3) 領域知識記述, (4) エージェント間での知識記述形式の整合性の確認, (5) 知識の調整と実装の5つの過程を必要としていた。今回提案した手法により, このうち, (2) が削減され, (4) が容易に行えるようになった。その結果, ADIPS フレームワークに基づいて分散処理システムを構成するエージェント部品の開発過程, すなわち, 組織構成エージェントの設計および実装の過程が簡略化され, 設計者の負担を軽減することができた。

6. おわりに

エージェント指向分散処理システム開発のための枠組みである ADIPS フレームワークと, そのエージェントのための基本的知識表現モデルが提案¹⁰⁾されていた。しかし, 本フレームワークの実用化に関して, 組織構成エージェントの領域知識を記述する際の具体的手法が与えられておらず, エージェント開発者の負担が大ききことが問題となっていた。

この問題を解決するために, 本論文では領域知識を6種類に分類し, その記述形式と処理機構を提案した。さらに標準的な知識記述テンプレートを導入し, 知識記述を統一化する手法を提案した。これにより, 領域知識を系統的に獲得・記述する具体的な手段が与えられ, 領域知識記述形式を決定する作業が省略され, 組織構成エージェントの知識が系統的に設計でき, エージェント開発者の負担が大きく軽減されることを確認し, 提案手法の有効性を示した。

参考文献

- 1) Doyle, J. and McMahon, C.: Intelligenet 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) Shoham, Y.: Agent-oriented programming, *Artificial Intelligence*, Vol.60, No.1, pp.51-92 (1993).
- 6) Riecken, D.E.: Special Issue: Intelligent Agents, *CACM*, Vol.37, No.7 (1994).
- 7) Wooldridge, M. and Jennings, N.E.: *Intelligent Agents*, Springer-Verlag (1995).
- 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 (1997).
- 9) 石田 亨: 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).
- 10) 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).
- 11) Smith, R.G.: The Contract Net Protocol: Hight-Level Communication and Control in a Dtributed Problem Solver, *IEEE Trans. Comput.*, Vol.29, No.12, pp.1104-1113 (1980).
- 12) Davis, R. and Smith, R.G.: Negotiation as a Metaphor for Distributed Problem Solving, *Artificial Intelligence*, Vol.20, No.1, pp.63-109 (1983).
- 13) 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).

(平成9年5月19日受付)

(平成9年10月1日採録)



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。エージェント, 分散処理システム設計論に興味を持つ。平成7年度電子情報通信学会学術奨励賞, 平成8年度本会論文賞受賞。電子情報通信学会, 人工知能学会会員。



原 英樹

1970年生。1993年千葉工業大学情報工学科卒業。同大学大学院博士後期課程情報工学専攻在学中。ソフトウェア工学に興味を持つ。電子情報通信学会会員。

**菅原 研次 (正会員)**

1950年生。1980年東北大学大学院工学研究科博士課程中退。同年千葉工業大学電子工学科助手。現在同大学情報工学科教授。工学博士。エージェント指向コンピューティング、

CAI, サイバー社会に興味を持つ。1992年日本工業教育協会功績賞, 1994年本会山下記念賞, 平成8年度本会論文賞受賞。電子情報通信学会, 人工知能学会, IEEE, ACM各会員。

**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。平成8年東北大学助教授(電気通信研究所)。知識型設計支援システム, ヒューマンインタ

フェース, エージェント指向コンピューティングなどの研究開発に従事。平成元年度本会研究賞, 平成8年度本会論文賞受賞。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, 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)。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通信と分散処理研究会主査。1996年本会理事。本会25周年記念論文賞, 平成8年度本会論文賞受賞。IEEE, 電子情報通信学会, 人工知能学会各会員。