

マルチエージェントに基づく マルチメディア処理コンポーネントの動的構成方式

高橋 晶子[†] 菅沼 拓夫^{††} 木下 哲男[†]

利用者指向サービスの実現に向けて提案されたやわらかいネットワーク層において、様々なマルチメディアデータの処理機能群の組合せによって動的にマルチメディア通信サービスを提供するマルチメディア処理コンポーネントの動的構成方式について述べる。やわらかいネットワーク層では、利用者要求や資源状況、マルチメディアデータの特性などに応じ必要十分なマルチメディア処理コンポーネント群を動的に構成する必要がある。この問題に対し(1)マルチメディア処理コンポーネント自体と、それに関連する知識から構成されるメディアエレメントエージェント(MEA)のアーキテクチャ、および(2)QoSを考慮したエージェント組織構成のためのプロトコルQ-CNP、の2点を提案し、上記問題を解決する手法を与える。

Dynamic Construction Scheme of Multimedia Processing Components Based on Multiagent Framework

AKIKO TAKAHASHI,[†] TAKUO SUGANUMA^{††} and TETSUO KINOSHITA[†]

In this paper, we describe a dynamic construction scheme for multimedia processing components in Flexible Network Layer (FNL). The organization of multimedia processing components provides multimedia communication service dynamically by combining various data processing functions. In FNL, it is required to construct the necessary and sufficient group of the multimedia processing components based on status of user requirement, platform/network environment, and characteristics of multimedia data. To resolve this problem, we propose (1) an architecture of Media Element Agent (MEA) that consists of a multimedia processing component itself and related knowledge concerning the component, and, (2) the Q-CNP that is a protocol to construct the agent organization considering the QoS.

1. はじめに

インターネットの高速化・高機能化にともない、様々な機能を持ったマルチメディア型ネットワークアプリケーションが登場してきている。従来のネットワークアプリケーションは、利用者の要求や環境の状況の変化に適応して安定したサービスを提供することが困難であるという問題がある。そこで我々は、利用者指向の安定した通信サービスを提供するためのネットワークアーキテクチャとして、論理ネットワーク層(Logical Network Layer: LN層)とアプリケーション層(Application Layer: AP層)の間に、利用者要

求や計算機・ネットワークの資源状況に基づき、流量制御、経路制御、QoS制御などの自動調整を行うミドルウェアとして、やわらかいネットワーク層(Flexible Network Layer: FN層)を新たに導入する動的ネットワークアーキテクチャ^{1),2)}を提案している。

現在、FN層内の各機能ユニットの詳細設計および実装を進めている。本研究ではその中で、AP層に対してマルチメディア通信サービスを直接的に提供する上位層インタフェースの役割を持つミドルウェアサービスユニット(Middleware Service Unit: MSU)に焦点を当てる。MSUでは、様々なマルチメディアデータの処理機能群(以降、メディア処理コンポーネントと略記)の組合せによってマルチメディア通信サービスを提供する。そのためには、利用者要求、資源状況、提供されるQoS、メディアデータの特性などに応じ必要十分かつ動的にメディア処理コンポーネントを構成する枠組みが必要となる。

本論文では、上記の枠組みをエージェント指向コン

[†] 東北大学情報シナジーセンター/情報科学研究科

Information Synergy Center/Graduate School of Information Science, Tohoku University

^{††} 東北大学電気通信研究所/情報科学研究科

Research Institute of Electrical Communication/Graduate School of Information Science, Tohoku University

ピューティング技術³⁾を適用して実現する。すなわち、メディア処理コンポーネントのエージェント化により実現されるメディアエレメントエージェント (Media Element Agent: MEA), および, MEA 間でのエージェント間協調プロトコルを提案する。具体的には, まず, メディア処理コンポーネントを, コンポーネントの設計者, 開発者, 利用者の持つ知識に基づいて管理と制御をする MEA のアーキテクチャを設計し, 利用者要求に基づいて協調処理を行う MEA 間での協調プロトコルを与える。次に, QoS を考慮したエージェント組織構成を行うために, 従来のコントラクトネットプロトコル (Contract Net Protocol: CNP)⁴⁾の枠組みを拡張した QoS-based CNP (Q-CNP) を提案する。MEA は, MEA の持つ知識と Q-CNP を利用して相互に連携し, 環境の変化に動的に適応するマルチエージェントシステムを構成して, QoS を考慮したマルチメディア通信サービスを提供する。

以下, 2 章では, MSU の機能要件および本論文で取り扱う課題について述べる。3 章では, エージェント概念を適用して MSU を設計する。4 章では, 3 章の設計に基づく実装について述べ, 5 章では, MSU におけるエージェント動作の具体例を示し提案手法の効果について述べる。

2. MSU の機能定義

2.1 MSU の機能要件

MSU では, マルチメディアデータの獲得・変換・送受信・再生処理 (以降, メディア処理と略記) によって計算機やプラットフォーム, ネットワークの資源を大量に消費する傾向があるため, 実行時の資源状況の影響を受けやすい。また, AP 層に対して直接的にサービスを提供するため, 利用者要求と提供サービスとの差異を的確に反映した制御を行う必要がある。さらに, 利用者はネットワーク上に分散するため, 異種環境への対応, 分散システム全体としての利用者要求の充足, システム全体としての安定動作なども考慮する必要がある。以上の特色から, MSU では以下の機能要件を満たす必要がある。

- (R1) 計算機・ネットワーク資源が十分に利用できる場合には, 搭載されたメディア処理機能は設計仕様どおりに動作できる。
- (R2) サービス開始時に利用者要求, 資源状況を反映して動的に MSU 内の機能が生成できる。
- (R3) サービス提供中の状況変化に対して, 自律的に対応できる。
- (R4) メディア処理やその制御のための情報交換が,

多くの異種プラットフォーム上で実行できる。
(R5) 新たに開発された機能を容易に組み込める。

2.2 技術課題

2.1 節で述べた機能要件を満たすシステムとして, Java Media Framework (JMF)⁵⁾などに代表されるメディア処理のためのコンポーネントフレームワークがある。

JMF のようなコンポーネントフレームワークを利用することで, 2.1 節で述べた機能要件の (R1) (R4), (R5) はおおむね満足することができる。すなわち, JMF は十分な資源を持つ環境においては, マルチメディア通信を用いたコミュニケーションを行うための基本的なマルチメディア処理能力を提供できるので, (R1) は実現できる。また, 様々な異種プラットフォーム上で動作が可能となるように, コンポーネントの共通化とプラットフォームに合わせたコンポーネント提供が行われているため (R4) は満たされる。さらに, コンポーネントとして機能の部品化が行われており, 新機能の追加において他の機能との関連を強く考慮する必要がないため (R5) も満たされる。

これに対して, 動作時の動的で自律的な構成・調整を必要とする (R2) (R3) を実現することは困難である。これは, 各機能の組合せにおいて様々な制約条件や相互依存性が存在すること, また, 最適なパラメータの設定を動的に行うことが困難であることに起因する。本論文では, サービス開始時における MSU 内の機能構成 (R2) を実現する手法に焦点を絞り (R3) については別稿で述べる。

(R2) を満たすためには, 以下の 2 つの技術課題を解決する必要がある。

(P1) コンポーネントの仕様に関する記述が困難

それぞれのコンポーネント, および, それが提供するサービスの機能や QoS に関する様々な仕様を記述する手法が与えられていない。

(P2) 動的な環境における適切なパラメータの設定が困難

コンポーネントの仕様に関する記述が与えられた場合でも, コンポーネントの動作は計算機環境やネットワーク環境に依存するため, その仕様をすべて満足した動作を保障することは困難である。また, 状況が変化する環境において, 適切なパラメータ設定を動的に行うための手法も提供されていない。

2.3 エージェント指向コンピューティングの適用

本論文では, 2.1 節で述べた機能要件の中で (P1), (P2) を解決することによって JMF のような既存のコ

ンポーネントフレームワークの適用では解決できない (R2) を満たすために、エージェント指向コンピューティングを適用して MSU を設計する。

エージェントは、エージェント自体に組み込まれた機能や知識を用い、エージェントの外部環境を認識したり働きかけを行ったりして、与えられた問題を解決するソフトウェアコンポーネントである。本研究では、メディア処理コンポーネントをエージェントとして設計・実現する。エージェントは、対象コンポーネントの機能や性能に関する仕様を知識として保持する。すなわち、コンポーネントに関する知識、具体的には、3.1 節で述べるように、コンポーネントが実現する処理の種類や性質、コンポーネントの持つパラメータとそのパラメータ値の制約、パラメータ値の設定と外部環境の状況から推測される提供サービスの品質、コンポーネントの組み合わせ方によって生ずる依存関係の制約などを利用することにより、複雑な制約条件や相互依存を系統的に処理することが可能となり (P1) の解決が期待できる。

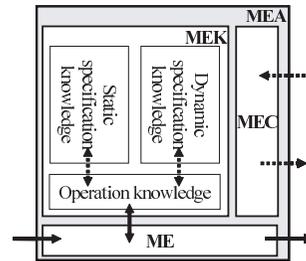
さらに、エージェントは、単体のエージェントでは解決できず、複数のエージェントが協調しなければ解決できないような複雑な問題を解決するために、動的にエージェントの集団を構成することができる。ある問題を解決するためのエージェントの集団をエージェント組織と呼ぶ。組織を構成する際に、エージェントの仕様知識が利用される。また、メディア処理は多くのコンポーネントの組合せによって構成されるので、全体として提供される QoS やそれぞれのコンポーネントの制約状態、コンポーネント間の相互依存性、パラメータ設定の制約条件なども考慮する必要がある。メディア処理コンポーネントのエージェント組織では、これらの条件がエージェント間の“契約”として形式化され、効果的かつ適応的なコンポーネント組織を構成することができる。これにより (P2) の解決が期待できる。

3. MSU のエージェント指向設計

3.1 MEA のアーキテクチャ設計

MSU におけるメディア処理機能を実現するプロセスそのものをメディアエレメント (ME) と呼び、ME をエージェント化したものをメディアエレメントエージェント (MEA) と呼ぶ。図 1 に MEA のアーキテクチャを示す。MEA は、ME、ME に関する知識 MEK、および、他のエージェントとの通信を行うモジュール MEC により構成される。

MEA ::= <ME, MEK, MEC>



MEA: Media Element Agent
 MEK: Knowledge of ME
 MEC: Communication module between agents
 →: Flow of multimedia data
: Flow of messages between agents
 →: Watch and control of ME
: Reference between knowledge

図 1 MEA のエージェントアーキテクチャ
 Fig.1 Agent architecture of MEA.

MEK は、静的仕様知識 (Static specification knowledge: S-MEA)、動的仕様知識 (Dynamic specification knowledge: D-MEA)、および動作知識 (Operation knowledge: A-MEA) により構成される。それぞれの知識の構成は以下のとおりである。

MEK ::= <S-MEK, D-MEK, A-MEK>

S-MEK ::= <ID, OSPEC, SQOS, FUNC, CONST, DEP>

ID: ME の基本情報

OSPEC: ME の入出力形式などの外部仕様

SQOS: ME が提供可能な QoS

FUNC: ME のパラメータなどの機能仕様

CONST: 環境などによる機能の制約

DEP: 他の MEA との依存性

D-MEK ::= <PARAM, QOS, THRES, STAT, RELS>

PARAM: 指定されている ME のパラメータ

QOS: ME により提供されている QoS

THRES: PARAM の変更基準となる閾値

STAT: ME 内の観測可能な状態変数

RELS: 他の MEA との協調状態

A-MEK ::= <ANA, AC, REL, CONT, COOP>

ANA: ME の状態分析

AC: エージェント間プロトコル処理

REL: 契約締結などのエージェント間の関連付け

CONT: ME の制御

COOP: 他の MEA との協調動作

ここで、本論文で特に重要な働きをする S-MEK の知識テンプレートを図 2 に示す。このようなテンプレートに基づき、各 MEA の知識を記述する。S-MEK

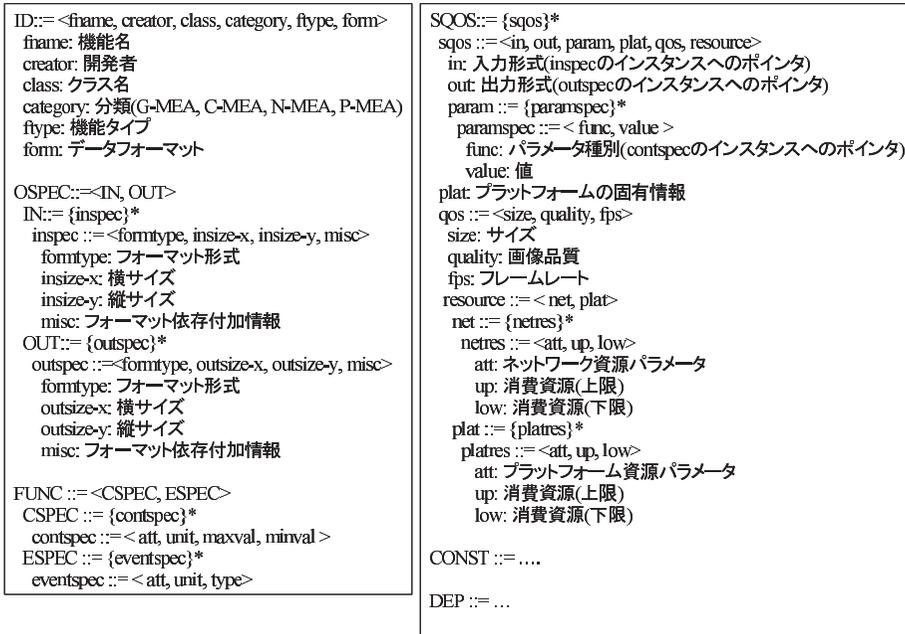


図 2 S-MEK の知識テンプレート
Fig. 2 Knowledge template of S-MEK.

は、対象とする ME の基本機能などの静的な仕様を記述したフレーム型知識であり、この知識を、エージェントの動作を決定するルール型知識である A-MEK が参照・利用する。具体的には、ルール型知識の条件部で、S-MEK の内容と、MEC を介して他のエージェントから送られてきたメッセージの内容を比較する。そして、両者が一致する場合、当該ルールのアクション部に記述された動作系列を実行することによって、各エージェントの振舞いが決定される。

3.2 MSU のエージェントフレームワーク設計

本論文では、3.1 節で述べた知識、および本節で述べるエージェントフレームワークに基づいて構成されたエージェント組織が、3.3 節で述べる動作を行うことで、実行時の利用者要求や資源状況を反映した MEA のエージェント組織を動的に構成する。図 3 に MSU のエージェントフレームワークを示す。

MEA は、データのキャプチャを行う G-MEA、データの変換を行う C-MEA、ネットワーク上でデータの送受信を行う N-MEA、および、データの再生を行う P-MEA に分類される。また、MEA 群の組織を管理するための Manager エージェント（以降、Manager と略記）がサービス要求ごとに生成される。

各々の MEA はエージェントリポジトリ（Agent Repository：以降、リポジトリと略記）と呼ばれる機構の中に、個々の ME のクラスファイルとそれぞれ

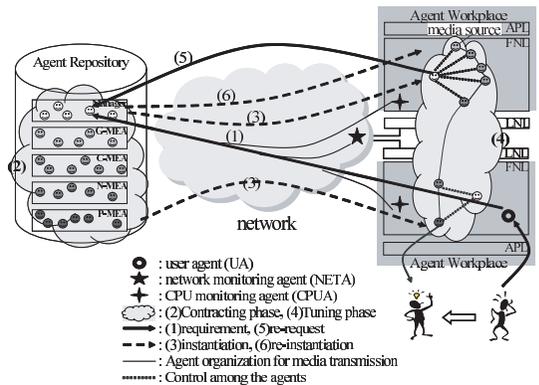


図 3 MSU のエージェントフレームワーク
Fig. 3 Agent framework for MSU.

に対応する知識（MEK）に基づいて定義されたエージェントとして、機能ごとに分けて格納されている。また、リポジトリはエージェントワークスペース（Agent Workplace：リポジトリからインスタンス化されたエージェントが実際に協調して動作し、利用者にサービス提供を行う機構。以降、ワークスペースと略記）とネットワークを介して接続されている。ワークスペースには、ユーザインタフェースを保持し利用者要求を獲得するユーザエージェント（User Agent: UA）、ネットワークの空き帯域などに関する情報を保持するネットワーク監視エージェント（NETwork

monitoring Agent: NETA), CPU 利用率などの情報を獲得する CPU 監視エージェント (CPU monitoring Agent: CPUTA) が存在する。実際に利用者要求が生じ、サービスが開始されるまでのエージェントの動作は以下のとおりである。

- (a) 利用者 (受信側) は, 自分の要求を受信側のワークスペースの UA に伝える。受信側の UA は利用者要求をリポジトリ内の MSU の組織を構成する役割をする Manager に送信する (1)。
- (b) Manager は送信側の UA からメディアデータに関する情報と NETA や CPUTA からの資源情報を得る (1)。
- (c) Manager は利用者要求を満たすサービスを提供可能な G-MEA, C-MEA, N-MEA, P-MEA とその組合せを適切に選択する (Contracting phase: 契約フェーズ (2))。
- (d) エージェントがインスタンス化され送信側各々のワークスペースにエージェント組織が生成される (3)。
- (e) ワークスペースでは, 提供される QoS や資源状況を詳細に監視しながら, インスタンス化されたエージェントが処理を開始する。このとき, 契約条件を満たすように, パラメータの調整が行われる (Tuning phase: 調整フェーズ (4))。
- (f) 契約条件が満たされた場合は (g) へ。契約条件が満たされていない場合は, 代わりに組織を要求するためにワークスペースの Manager がリポジトリに再要求 (re-request) を送信 (5) し, (d) へ移行する。
- (g) MEA の組織構成が終了する。

以上のようなリポジトリを中心としたエージェントフレームワークを構成することにより, 以下の利点が生ずる。

- エージェントをリポジトリに格納することにより, ME や MEK を集中管理することができる。
- 要求発生と同時にリポジトリからワークスペースにインスタンス化する仕組みと組み合わせることで, つねに最新の ME とそれに関する知識 MEK を利用することが可能となる。
- ワークスペースでの動作後に MEA をリポジトリに戻すことで, 動作結果を知識に反映させることが可能となり, エージェントの動作を環境に適応させることが容易になる。

3.3 MSU のエージェント動作設計

MSU エージェント組織は, コントラクトネットプロトコル (CNP) の枠組みを, QoS を考慮した組織

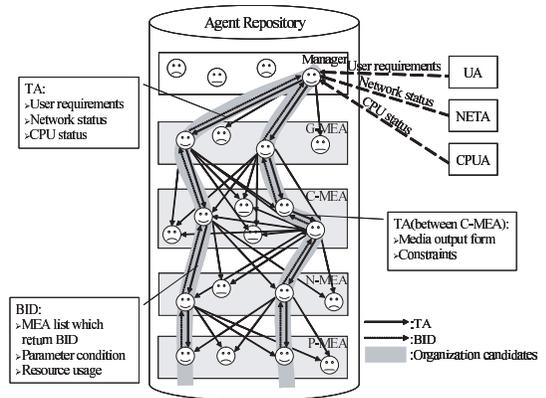


図 4 Q-CNP における契約フェーズのエージェント動作
Fig. 4 Behavior of MEA in Contracting phase of Q-CNP.

構成に適合するように拡張した QoS-based CNP (Q-CNP) を利用して MEA 間の協調動作により構成される。従来の CNP では, プラットホームやネットワーク状況に依存して契約条件が不安定となるため, QoS を考慮したメディア処理コンポーネントの組織構成には適していない。これは, リポジトリ内で締結された契約条件に基づき, ワークスペースにおいてエージェントが条件を満たして動作することが保障できないためである。そこで, 本研究では, この問題を解決するために, 従来の CNP における契約フェーズを拡張するとともに, インスタンス化後に契約条件の微調整を行う調整フェーズを新たに導入した。図 4 に拡張された契約フェーズにおけるエージェント動作を示す。その処理の流れは以下のとおりである。

- (1) 利用者からのサービス要求に従って, UA は利用者要求を Manager に伝える。
- (2) Manager は NETA と CPUTA それぞれから, ネットワーク状況 (空き帯域など) や CPU 状況 (利用率など) に関する情報を得る。
- (3) エージェント組織を構成するために, Manager は利用者要求やネットワーク状況, CPU 状況を含んだタスク通知 (TA) を生成する。この情報は, MEA 間の契約条件として評価される。TA は, メディアデータをキャプチャするためのエージェント組織の構成のためにすべての G-MEA に送信される。
- (4) メディアデータの変換をするエージェント組織を構成するために, Manager からの TA を受けた G-MEA はすべての C-MEA に対し TA を送信する。TA を受信した C-MEA は, 自身の S-MEK の OSPEC 知識を参照することで, TA を送信してきた C-MEA の出力するメディアデータ形式

を受け取ることができるか判断し、できる場合は OSPEC の OUT 知識を参照して、自身の出力するメディアデータ形式の情報を含めた TA を他のすべての C-MEA に対して送信する。この TA の送信は、契約条件を満足するまで、繰り返し行われる。ただし、TA の送信を所定の回数（現在は、経験的に 5 回に設定している）だけ行っても契約条件が満たされなかった場合は、TA の送信を打ち切る。

- (5) 契約条件を満たすメディアデータの形式が見つかったら、メディアデータをネットワークを介して送受信するためのエージェント組織を構成するために C-MEA から N-MEA に TA が送信される。
- (6) メディアデータを再生するためのエージェント組織を構成するために、TA を受け取ることのできた N-MEA はすべての P-MEA に TA を送信する。
- (7) (3) ~ (6) の手順で TA が伝播された結果、P-MEA -> N-MEA -> C-MEA -> G-MEA -> Manager の順に入札通知 (BID) が返される。BID には、選択された MEA のパラメータ条件や、MEA によって消費される資源の情報などが、S-MEK の SQOS 知識に基づいて生成され、加えられる。すなわち、特定の条件下で各 MEA が提供可能な QoS と、その QoS を提供する際に使用される資源量が sqos のリストとしてエージェント間を伝播し、Manager に集積される。
- (8) BID を受け取った Manager は、TA の条件との差異が最小の契約条件を提示した BID を返したエージェント組織を選択し、これに対して落札通知 (AWARD) を送信する。Manager は、次のエージェント組織の候補の再インスタンスを迅速に行うために、ワークスペースから組織決定通知 (complete) を受け取るまでの間、選択されなかったエージェント組織の候補を保持しておく。

契約フェーズが終了すると、ワークスペース上に仮のエージェント組織がインスタンス化され、Q-CNP は調整フェーズへと移行する。図 5 に調整フェーズのエージェント動作を示す。調整フェーズの流れは以下のとおりである。

- (a) ワークスペースの Manager (aManager) は、CPUA や NETA、P-MEA から資源状況や提供 QoS についての報告 (report) を受ける。
- (b) 契約条件が満たされていないことが観測されたら、aManager は調整する必要がある MEA にパラメータの調整の動作要求 (request-action) を

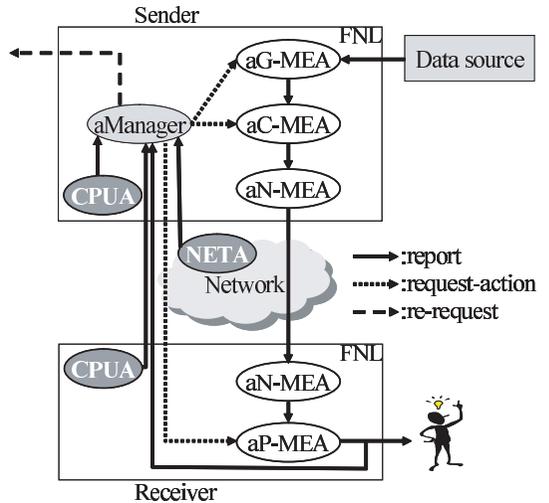


図 5 Q-CNP における調整フェーズのエージェント動作
Fig. 5 Behavior of MEA in Tuning phase of Q-CNP.

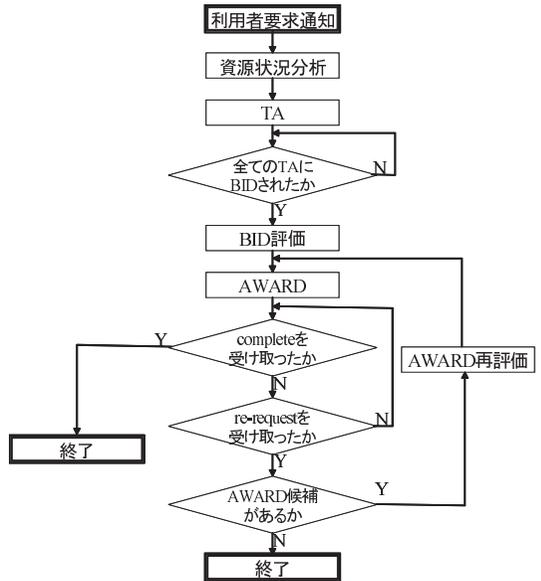


図 6 Manager の Q-CNP 処理のフローチャート
Fig. 6 Flow chart of Q-CNP processing of Manager.

- する。
 - (c) 契約の不履行の状況が改善されるまで (a), (b) は繰り返される。改善されない場合、すなわち (a), (b) の繰り返し処理が所定の時間を超過した場合、aManager は組織の再構成を行うためにリポジトリの Manager に再要求 (re-request) をする。
- 図 6 に Manager の Q-CNP 処理のフローチャートを示す。また、図 7 に Q-CNP 全体のプロトコルシーケンスをまとめる。

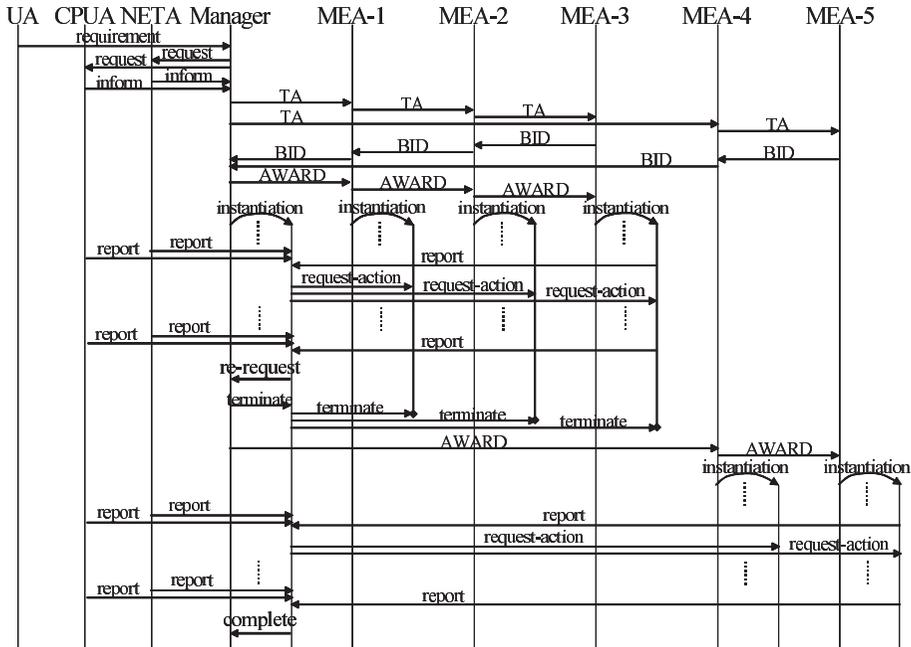


図 7 Q-CNP のプロトコルシーケンス
Fig. 7 Protocol sequence chart of Q-CNP.

4. 実 装

提案システムの実装には、マルチエージェントシステム基盤として ADIPS フレームワーク⁶⁾を用いた。ADIPS フレームワークを採用した理由は、本システムのフレームワークを実現するには ADIPS フレームワークの提供するリポトリ指向エージェントフレームワークが適すること、また ADIPS フレームワークのエージェントアーキテクチャが知識型であり、MEA のエージェントアーキテクチャを効果的に実現できることなどがあげられる。

ADIPS フレームワークにおけるエージェントは、制御対象となるオブジェクト部(ベースプロセスと呼ぶ)、およびベースプロセスに関する仕様知識・制御知識などを用いてベースプロセスを監視・制御する知識部から構成される。

ADIPS フレームワークの最新バージョンである DASH1.9.7h⁷⁾では知識部の記述にルール型知識表現が利用可能である。具体的には、DASH ではエージェントの知識記述をルール型プログラミング言語 DASH/R に基づいて行う。DASH/R を用いて、エージェント間協調やベースプロセス制御などの動作に関する知識をルールで、またベースプロセスの仕様やエージェントの状態などをファクトで、それぞれ記述し、エージェント内に知識として保持する。

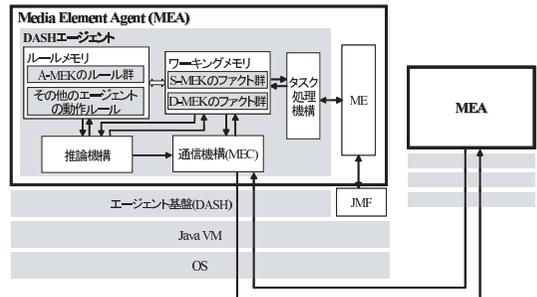


図 8 実装モデル
Fig. 8 Implementation model.

DASH/R に基づいて記述されるエージェントを DASH エージェントと呼ぶ。本提案システムでは、MEA を DASH エージェントとして実装する。図 8 に MEA の実装モデルを示す。本実装モデルでは、MEA の S-MEK と D-MEK をワーキングメモリ内のファクトとして、また A-MEK をルールメモリ内のルールとして、それぞれ実装する。

S-MEK と D-MEK はエージェント記述内の初期ファクト (initial_facts) として与える。S-MEK は静的な仕様記述であるので S-MEK を表現したファクトは基本的には知識の更新は行われない。一方、D-MEK を表現するファクトは ME の状態や他のエージェントとのインタラクションによって変化する。また、推論機構は、A-MEK を記述するルールに基づき、ワーキ

```
(ID (:frame H263Encoder :creator ibm
  :class com.ibm.media.codec.video.h263.NativeEncoder
  :category C-MEA :ftype Encoder :form h263))
(IN (inspec1 (:fomftype RGB :insize-x 640 :insize-y 480
  :misc (:datatype...)))
  (inspec2 (:fomftype RGB :insize-x 320 :insize-y 240
  :misc (:datatype...)))
  ...
(OUT (outspec1 ...)
  (outspec2 ...))
(CSPEC (contspec1 (:att size_x :unit pixel :maxvalue 640 :minvalue 120))
  (contspec2 (:att size_y :unit pixel :maxvalue 480 :minvalue 60))
  (contspec3 (:att quality :unit percent :maxvalue 100 :minvalue 0))
  ...
(SQOS (sqos1
  (:in inspec1 :out outspec1
  :param ((param1 :func contspec1 :value 320)
    (param2 :func contspec2 :value 240)
    (param3 :func contspec3 :value 80))
  :plat platspec1
  :qos (:size 320 :quality 80 :fps 16)
  :resource ((netres1 :att bandwidth :up 200 :low 150)
    (platres1 :att cpuusage :up 40 :low 20))
  ))
  (sqos2 ...)
  (sqos3 ...))
)
```

図 9 DASH/R に基づく H263Encoder エージェントの S-MEK の記述例

Fig. 9 Example of S-MEK description of H263Encoder Agent based on DASH/R.

```
(rule select-sqos-fact
  (req-sqos :in (:fomftype ?recfom :insize-x ?rec-insize-x :insize-y ?rec-insize-y))
  (ureq :size-x ?ur-size-x :size-y ?ur-size-y :smooth ?ur-smooth :quality ?ur-quality)
  (?sqos :out (:fomftype ?outfom :outsized-x ?out-outsized-x :outsized-y ?out-outsized-y)
  :qos (:size ?qos-size :quality ?qos-quality :fps ?qos-fps))
  (sqoslist ?sqos-list-body)
  (= ?recfom ?outfom)
  (= ?rec-insize-x ?out-outsized-x)
  (= ?rec-insize-y ?out-outsized-y)
  (> ?qos-size ?ur-size-x)
  (> ?qos-quality ?ur-quality)
  (> ?qos-fps ?ur-smooth)
  )
  (push ?sqos ?sqos-list-body)
)
```

図 10 DASH/R に基づく H263Encoder エージェントの A-MEK の記述例

Fig. 10 Example of A-MEK description of H263Encoder Agent based on DASH/R.

ングメモリ内のファクトとルールとの条件部との照合を行い、マッチしたルールのアクション部の記述に従って、ファクトの更新、通信機構を介した他のエージェントへのメッセージ送信、タスク処理機構を介した ME の制御などの動作を行う。また DASH では、基本プロトコル処理のための知識をルールセットの組み込みによって提供している。したがって、Q-CNP のプロトコル処理には、DASH に標準装備されている CNP のプロトコル処理のためのルールセットを改良して利用する。

また、DASH は Java オブジェクトをベースプロセスとして利用することを前提として設計されているため、Java との親和性が高い。このことから ME としては Java のマルチメディア処理拡張である JMF を用いる。すなわち、MEA は JMF のオブジェクトをベースプロセスとして利用する。

図 9 に、DASH/R に基づく JMF の H263Encoder コンポーネントの S-MEK の記述例を、また図 10 に同コンポーネントの A-MEK の記述例をそれぞれ示す。これらの知識の詳細については次章で述べる。

5. 適用例と評価

5.1 動画配信サービスへの適用

本節では、提案手法を、サーバ - クライアント間での動画配信サービスに適用した場合の MSU エージェ

ント組織の動的構成の事例を示す。

図 11 に、Q-CNP における契約フェーズの動作例を示す。リポジトリには、JMF のメディア処理コンポーネントを ME として保持する MEA が格納されている。ここで、ある利用者が遠隔のコンピュータに接続されたカメラの映像をネットワークを通じてリアルタイムに受信したいという要求を持ったとする。このとき、画像の品質に対する要求として、画像サイズが 300×200 pixel 以上、フレームレートが 15fps、画質が 60% を指定したとする。UA から Manager にこの要求が伝えられ、Manager は、NETA、送受信双方の CPUA に資源状況の報告を要求する。これらの情報を基に、Manager は TA を生成し、リポジトリ内に伝播させる。

TA の伝播時においては、主に、OSPEC 知識によって記述されている ME の入出力メディア形式が考慮される。すなわち個々の MEA では、TA によって記述された形式のメディアを受け取れる場合に限り、自身の出力可能な形式を記述した TA を伝播させる。この方式により、図 11 の organization-2 では、RawVideoBuffer -> RGBScaler -> RGBtoYUV -> H263Encoder という組織ができています。これは、G-MEA である RawVideoBuffer の出力が RGB であるのに対し、H263Encoder が YUV 形式しか入力として受け付けないという知識を持っていたため、RGBScaler で RGB を適切なサイズに変換したのち RGBtoYUV によって YUV に変換し、H263Encoder に渡すという組織が形成された例である。

一方 BID においては、ME が提供可能な QoS と使用する資源に関する情報が、各エージェントの S-MEK 内の SQOS 知識の要素 sqos のリストの形式で伝播され、Manager まで送られる。たとえば、H263Encoder は、RTPMultiplexer から受信した BID 内の sqos のリストから判断し、自身の S-MEK の SQOS 知識に基

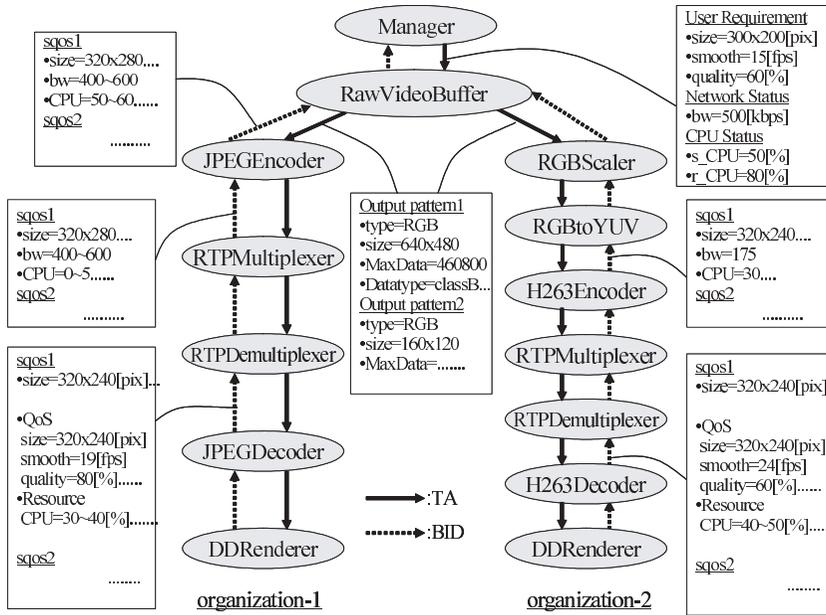


図 11 Q-CNP に基づくエージェント間協調の例 (契約フェーズ)

Fig. 11 An example of cooperation process among agents (Contracting phase).

づいて、与えられた条件下で自身の提供できる QoS を判定し、その情報を sqos のリストとして含めた BID を生成して、RGBtoYUV に送信する。図 10 に示したルール型知識である A-MEK 知識は、図 9 に示した S-MEK 知識に基づいて H263Encoder が自身の SQOS 知識から、条件に適合する sqos を探すための知識の例である。具体的には、図 10 に示す知識の 2 行目で、ファクトとして展開された RTPMultiplexer からの BID 中の sqos から、RTPMultiplexer が受信できるメディア形式に関する情報を取得している。また 3 行目では、利用者要求を取得している。さらに 4-5 行目で、自身の SQOS 知識のうち 1 つの sqos (たとえば図 9 の sqos1 ファクト) をバインドする。7-9 行目は、RTPMultiplexer が受信できるメディア形式と一致するかどうか、また 10-12 行目は、H263Encoder の出力が利用者要求を満たすかどうか、すなわち、要求値よりも提供 QoS 値が高いかどうかをチェックする条件である。これらの条件をすべて満たす場合にこのルールは発火し、14 行目のアクションを実行する。ここでは、条件にマッチした sqos をリストに追加している。

Manager は BID の情報に基づき AWARD を与える組織を決定する。具体的には、まず、BID に含まれる各 sqos の resource ファクトの bandwidth や cpuusage の値から、該当する組織が選択された場合の総資源消費量を計算する。次に、利用者要求として与えられた資源量の制約と比較して、資源に関する

制約が要求を満たしているかどうか、また QoS が実現されているかどうかを判断する。図 11 に示した例では、organization-1、organization-2 の 2 組織が候補となっている。両組織とも受信側で利用者に対して提供される QoS は利用者要求を満たしているものの、organization-1 が送信側の CPU 資源状況において契約条件を満たしていない。そこで、Manager は organization-2 を選択し AWARD を送信する。organization-1 は調整フェーズ失敗時の次候補として Manager によって保持される。

図 12 に、Q-CNP における調整フェーズの動作例を示す。図 12 では、契約フェーズによって決定された organization-2 がインスタンス化されている。ここで契約条件が満たされているかを確認し、必要であれば微調整を行う。この例では、(1) の report によって資源状況を確認したところ、送信側の CPU 資源状況が契約条件を満たしていない状況が確認される。この場合 (2) の request-action によって、H263Encoder に画像品質を所定の量 (この場合、20%程度) だけ低下させるよう指示を出す。H263Encoder は指示に従いパラメータを変更する (3)。再度 (4) の report により資源状況と利用者へ提供されている QoS を確認したところ、契約条件を満たしていることが確認されたため (5) の complete により組織決定をリポジトリの Manager に伝えた。これにより Manager は次候補である organization-1 をリリースする。

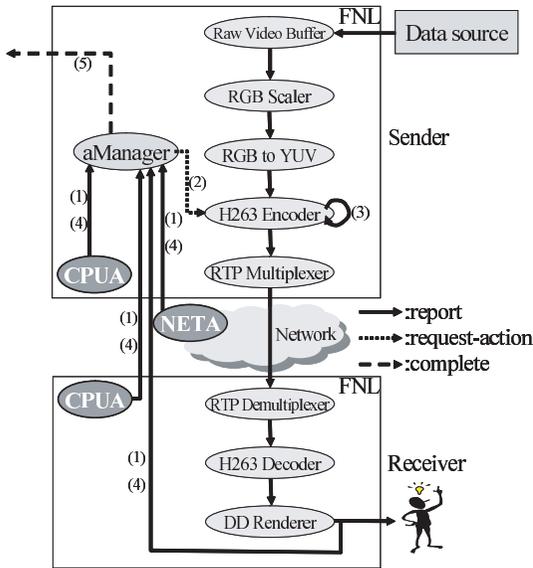


図 12 Q-CNP に基づくエージェント間協調の例 (調整フェーズ)
 Fig.12 An example of cooperation process among agents (Tuning phase).

以上のように、Q-CNP の契約フェーズと調整フェーズによって、環境や資源状況に応じて、QoS を考慮したマルチメディア処理コンポーネントの組織形成を効果的かつ柔軟に行うことができる。

5.2 評価

適用例をもとに、提案手法の特徴と利点について述べる。

(1) 動的な環境における ME 選択とパラメータ決定
 JMF においては、ME の選択を自動的に行うアルゴリズムとして、“graph building algorithm”がある。このアルゴリズムでは、利用可能な ME の中から入出力形式のみを参照して接続可能な ME の組合せを検索し、最も接続経路の短いものを採用する。また、基本的に各マシン内の ME の構成を行うものであり、ネットワークを介した ME の接続を系統的にサポートしていない。

これに対し、本システムでは、入出力形式だけでなく、動的な環境における ME の QoS や使用する資源までを意識した ME の選択とパラメータの決定を可能としている。またネットワークを介した ME の接続もサポートしている。

これは、ネットワークも含めた資源に関する知識や QoS に関する知識を SQOS 知識としてエージェントが保持し、それを条件として契約を締結する仕組みをエージェントが備えていることにより実現可能となったためである。

一方、契約を締結する基本的な仕組みは、CNP である。一般に、CNP に基づく ME の組合せ決定や各 ME の動作パラメータ値決定は一種の制約充足問題と見なすことができる。しかし、ME の動作は計算機やネットワーク資源の影響を大きく受けるため、従来の制約充足アルゴリズムなどを用いて、これらを事前に決定することは困難である⁸⁾。すなわち、ME やそのパラメータが決定され、動作環境での動作が開始された後に、動作状況を反映した動的な微調整が必要となる。

これに対して Q-CNP では、この動的な微調整を実現するために、CNP に基づく契約フェーズを拡張し、新たに調整フェーズを導入した組織構成手法を提案した。この調整フェーズにおいて、インスタンス化された MEA の知識に基づく微調整が行われる。これにより動的な環境において、従来にない効果的かつ柔軟な ME 選択やパラメータ調整を系統的に行うことができ、2.2 節で述べた技術課題 (P2) が解決される。

(2) ME に関する知識の整理

ME 間の依存性などに関する情報は経験的かつ複雑であるため、それらを適切に記述・管理し、必要に応じて利用するための仕組みが必要となる。本システムでは、3.1 節で述べたように、ME の知識表現モデル (テンプレート) を与え、このモデルに従ってエージェント内に ME に関する知識を記述することで、系統的に関連知識を整理・記述することが可能となる。また、ME と独立に知識を記述することにより、知識の再利用を促進する効果も期待できる。これらは、2.2 節において述べた技術課題 (P1) に対する部分的な解を与えている。今後、この知識表現モデルに基づいて、MEA に特化した知識記述支援環境を整備することにより (P1) の大幅な改善も期待できる。

6. おわりに

本論文では、マルチメディアデータの処理機能群をエージェント化したメディアエレメントエージェントの設計、および、それらの組織によるミドルウェアサービスユニットのマルチエージェント指向の設計手法を提案した。

今後は、本設計に基づき様々な機能をエージェント化して Agent Repository に格納し、実環境におけるエージェント組織の構成・再構成に関する実験と評価を通して、提案手法の改良と応用事例の集積を図っていく予定である。

参 考 文 献

- 1) Shiratori, N., Suganuma, T., Sugiura, S., Chakraborty, G., Sugawara, K., Kinoshita, T. and Lee, E.S.: Framework of a flexible computer communication network, *Computer Communications*, Vol.19, pp.1268–1275 (1996).
- 2) Suganuma, T., Kinoshita, T. and Shiratori, N.: Flexible Network Layer in Dynamic Networking Architecture, *Proc. 1st International Workshop on Flexible Networking and Cooperative Distributed Agents (FNCD2000)*, pp.473–478 (2000).
- 3) Jennings, N.R.: An Agent-based Approach for Building Complex Software Systems, *Comm. ACM*, Vol.44, No.4, pp.35–41 (2001).
- 4) Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *IEEE Trans. Comput.*, Vol.C-29, No.12, pp.1104–1113 (1980).
- 5) JMF Home Page, [Online]. Available: <http://java.sun.com/products/java-media/jmf/>
- 6) Fujita, S., Hara, H., Sugawara, K., Kinoshita, T. and Shiratori, N.: Agent-based Design Model of Adaptive Distributed System, *The International Journal of Artificial Intelligence, Neural Network and Complex Problem-Solving Technologies*, Vol.9, No.1, pp.57–70 (1998).
- 7) DASH —Distributed Agent System based on Hybrid Architecture!, [Online]. Available: <http://www.agent-town.com/dash/index.html>
- 8) Beugnard, A., Jezequel, J., Plouzean, N. and Watkins, D.: Making Components Contract Aware, *IEEE Computer*, pp.38–45 (July 1999).

(平成 15 年 5 月 16 日受付)

(平成 15 年 12 月 2 日採録)



高橋 晶子 (学生会員)

1979 年生。2002 年東北大学工学部情報工学科卒業。現在、同大学大学院情報科学研究科博士前期課程在学中。第 10 回 DPS ワークショップ Young Researcher Award 受賞。マルチメディア、マルチエージェントシステムに興味を持つ。



菅沼 拓夫 (正会員)

1966 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修了。1997 年東北大学電気通信研究所助手。2003 年同助教授。やわらかいネットワーク、エージェント指向コンピューティング、エージェント型ネットワークミドルウェア等の研究開発に従事。The 8th JWCC Best Presentation Award, 情報処理学会第 54 回全国大会大会奨励賞等受賞。博士(工学)。IEEE, 電子情報通信学会各会員。



木下 哲男 (正会員)

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。知識情報処理技術の研究開発に従事。1996 年東北大学電気通信研究所助教授, 2001 年同大学情報シナジーセンター教授。知識工学, エージェント工学, 知識型設計支援システム, エージェント応用システム等の研究開発に従事。情報処理学会 1989 年度研究賞, 1996 年度論文賞, 電子情報通信学会 2001 年度業績賞等受賞, 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, IEEE, ACM, AAAI 各会員。