

やわらかいネットワーク層における メディアエレメントエージェントの設計

高橋 晶子[†], 菅沼 拓夫[†], 木下 哲男^{††}, 白鳥 則郎[†]

[†]東北大学情報科学研究科/電気通信研究所

E-mail: {akiko, suganuma, norio}@shiratori.riec.tohoku.ac.jp

^{††}東北大学情報シナジーセンター/情報科学研究科

E-mail: kino@ka.riec.tohoku.ac.jp

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ミドルウェアサービスユニットのエージェント指向設計について述べる。ミドルウェアサービスユニットでは、利用者要求や資源状況、マルチメディアデータの特性等に応じ必要十分かつ動的にマルチメディアデータの処理機能群を構成・再構成する必要がある。本稿では、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構成法について議論する。

A Design of Media Element Agent in Flexible Network Layer

Akiko Takahashi[†], Takuo Suganuma[†], Tetsuo Kinoshita^{††} and Norio Shiratori[†]

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

E-mail: {akiko, suganuma, norio}@shiratori.riec.tohoku.ac.jp

^{††}Information Synergy Center/ Graduate School of Information Science, Tohoku University

E-mail: kino@ka.riec.tohoku.ac.jp

We describe an agent-oriented design of Middleware Service Unit (MSU) in the Flexible Network Layer. MSU provides multimedia communication services by combination of functions for processing various multimedia data. In MSU, it is required to configure/reconfigure the necessary and sufficient functions for processing multimedia data dynamically, considering user requirements, resource statuses, and characteristics of the multimedia data. In this paper, we propose a design of a Media Element Agent that is composed by agentification of functions for processing multimedia data, and also propose scheme of multiagent-oriented construction of MSU by the agents' organization.

1. はじめに

IP ネットワークの普及に伴い、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。しかしながら、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(LN 層)とアプリケーション層(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(FN 層: Flexible Network Layer)を新たに導入する動的ネットワークアーキテクチャ[1]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・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(以降、MSU と略記)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が、利用者要求や資源状況、及びメディアデータの特性等に応じ必要十分かつ動的にマルチメディアデータの処理機能群を構成・再構成する枠組みが必要である。

本稿では、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 MSU のマルチエージェント指向の構成法について議論する。

2. 関連研究

本研究では、マルチメディアデータの処理機能をエージェントとして扱い、サービス提供時の QoS を考慮したエージェントの動的組織構成法を提案している。関連研究としては QoS アウェアなソフトウェアコンポーネントに関する研究 [2] がある。

文献 [2] においては、コンポーネント間に「契約」の概念を導入し、これに基づき、従来のコンポーネントの構成法の分類を行っている。文献 [2] では、きわめて高い信頼性が要求されるシステム環境下でのコンポーネントの利用・再利用において、コンポーネントの不完全性・信

用性・誤使用により生ずる様々な問題点について指摘している。従って、予めコンポーネントをどのような条件で組合せるかを規定すること、すなわちコンポーネント間での契約が重要であるとしている。また契約の発生するタイミングに応じ、契約を(L1)~(L4)の4つのレベルに分類している。

- (L1)プログラミング設計時に静的にコンポーネントの組合せ・パラメータ・例外処理等を決定。
- (L2)コンパイル時に事前・事後条件を用いてコンポーネント間交渉を行い、コンポーネントの組合せを決定。
- (L3)サービス提供開始時にコンポーネントの組合せを決定。環境変化や再構成には、サービス提供者が静的に対応。
- (L4)実行時に必要が生じた場合に交渉を行い変化。QoS を定量的に観測し、常に交渉が可能。環境変化や再構成には、自律的かつ動的に対応。

契約は(L1)から(L4)に向かって動的な性質が強くなる。理想的には(L4)の契約を実現することが望ましい。しかし、(L4)の実現には QoS の定量的評価が必要であり、評価基準の定義や、パフォーマンスに関する保証の面で実現が困難であるという問題がある。また、契約が可能な安定したコンポーネントを実現するには、一般のコンポーネントフレームワークに、契約に関する機能を組み込む必要がある。

本研究では、FN 層における MSU を構成するマルチメディアデータの処理機能を、知的コンポーネントであるエージェントとして構築し、サービス提供時の QoS を考慮した MSU の動的構成法について提案を行う。従って、本研究は、文献 [2] で提案された契約の分類のうち (L4) を実現するための具体的な設計指針を与えるものである。

3. MSU の機能定義

3.1 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・ネットワーク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する必要があり。更に、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等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)機能的・性能的に十分な MSU 内のメディア処理機能を提供できる。
- (R2)実行時の利用者要求、資源状況を反映して動的に MSU 内の機能を組立てられる。
- (R3)実行時の状況変化に対し、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プラットフォームに依存せずに行える。
- (R5)新たに開発された機能を容易に組み込む。

3.2 既存フレームワークとの比較

3.1 で述べた機能要件を満たすシステムとして、Java Media Framework (JMF)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。例えば、JMF では Fig.1 に示される機能構成によって以下の手順でメディア処理が行われる。

- (1) 送信側の Device がデータを取り込む。
- (2) 送信側は受信側の要求に合うデータ形式に変換するために、いくつかの機能 (Processor) の静的な組合せにより変換処理を行う。
- (3) 受信側の要求に合ったデータに変換した後、適切な通信プロトコル処理機能(ここでは RTP Manager)を経由して、ネットワークを介して受信側にデータを送出する。
- (4) 受信側では、Player を通して受信側の Device から再生する。

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、3.1 で述べた機能要件の (R1), (R4), (R5)は満足することができる。すなわち、JMF のそれぞれの機能はコンポーネント化されており、それらを静的に組合せるこ

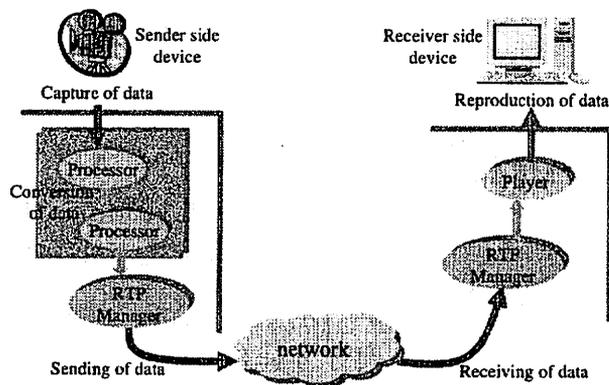


Fig.1 Media processing in JMF

とで、要求された機能の構成が可能であるので、(R1)は満たされる。また、プラットフォームに依存しないコンポーネントで構成されているため、(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無いため、(R5)は満たされる。

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)を満たすことは困難である。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、最適なパラメータの設定が困難であることに起因する。ここで、制約条件や相互依存性、パラメータ設定の制約などの問題を解決するために、それらの情報を集中管理する機構を導入する解決策が考えられる。しかしながらこの方法では、新規機能の追加時のメンテナンス等の問題が発生し、コンポーネント方式そのものの利点である新しい機能の組み込みに対する容易性を満たせなくなる。従って、たとえコンポーネントフレームワークを導入したとしても、(R5)の実現は困難となる。

4. MSU のエージェント指向設計

4.1 エージェント指向の適用

前章で述べた機能要件に関して、既存のコンポーネントフレームワークでは解決できない機能要件を満たすために、本研究ではエージェント指向に基づく MSU の設計を提案する。

エージェントは、自身が持っている知識を用いることによって、自律的に外部環境の認識や外部への働きかけなどを行い、与えられた問題に対して他のエージェントと協調して解決することができるソフトウェアコンポーネントである。これを用いることで機能要件を満たした MSU の実現が可能となると考えられる。そこで、メディア処理機能そのものに加えて、その機能・性能などに関する仕様知識と、仕様知識に応じて自身の振る舞いの制御や他のエージェント・環境への働きかけを行う動作知識を持たせたエージェントとして、メディア処理機能群を設計する。また、メディア処理は、いくつかの機能を組合せる必要があるが、これをエージェントの協調能力を活用することで効果的に行う。

複数のエージェントが協調して問題解決を行う方法の一つに、契約ネットプロトコルがある。契約ネットプロトコルでは、以下の手順でタスク分割を行い、エージェントの組織化を行

う。

- (1) タスク通知：他のエージェントにタスクを割り当てるマネージャエージェントが、他のエージェントに対してタスクの処理要求(タスク通知)を行う。
- (2) 入札：タスク通知を受け取ったエージェントは、提示されたタスクを自分が解決できる場合に、入札のメッセージをマネージャに返送する。
- (3) 落札：マネージャは受け取った入札メッセージから判断し、最も適切な入札メッセージを返したエージェントに対して落札のメッセージを送る。落札を受け取ったエージェントは、コントラクタとなってタスクを実行する。

上記の契約ネットプロトコルを MSU に適用することによって、メディア処理機能間でタスク通知・入札・落札を行い、その組合せを自律的かつ適切に決定することができる。

また、メディア処理のための機能そのものだけでなく、その機能に関する様々な仕様を知識として持たせてエージェントとして設計することで、各々の機能における制約条件や機能間の相互依存性、パラメータ設定の制約などを、エージェント自身で解決することが可能となる。以上により、前章で述べた機能要件を満たすことができる。

4.2 MSU のエージェント構成設計

MSU におけるメディア処理の機能そのものをメディアエレメント(ME)、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。MSU では、実行時に利用者要求や資源状況に基づき動的に MEA のエージェント組織を構成し、エージェントの協調性によって機能の調整や組織再構成を行うことで、前章で述べた機能要件を満たす。

MEA は、データの獲得を行う 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、及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 Agent Repository(以降、リポジトリと略記)と呼ばれるエージェントを保持する機構の中に、それぞれの機能ごとのクラスに分けて格納される。また、リポジトリは 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

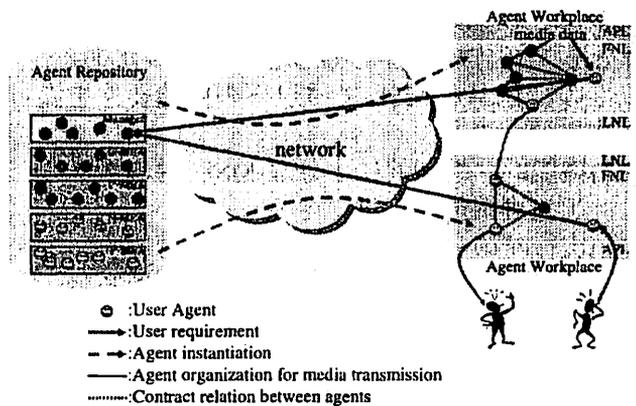


Fig.2 Configuration of agent organization of MSU

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 User Agent(以降、UA と略記)、ネットワークの空き帯域などに関する情報を保持する Network Monitor Agent(以降、NETA と略記)、CPU 利用率などの情報を獲得する CPU Monitor Agent(以降、CPUA と略記)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通りである(Fig.2)。

- (1) 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。
- (2) 受信側の UA は、リポジトリ中の Manager に利用者要求を伝える。
- (3) 利用者要求を受け取った Manager は、マルチメディアデータの発信元(送信側)の UA からデータに関する情報を得る。
- (4) 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。
- (5) エージェントがインスタンス化され送受信側各々のワークスペースでのエージェント組織の生成が行われ、ネットワークを介して送信側と受信側の ME が接続される。
- (6) マルチメディアデータの送信が行われ、利用者要求が満たされる。

4.3 MEA の知識設計

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>

MEK は Fig.3 に示す通り、静的仕様知識(Static specification knowledge: S-MEA)、動的仕

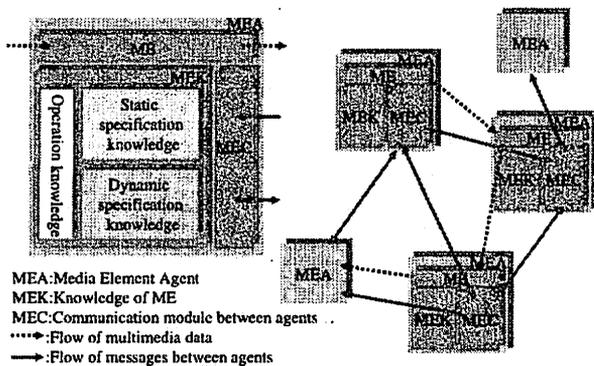


Fig.3 Architecture of MEA and communication between MEAs

様知識 (Dynamic specification knowledge : D-MEA), および動作知識 (Operation specification 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通りである。

- MEK ::= <S-MEK, D-MEK, A-MEK>
- S-MEK ::= <OSPEC, SQOS, FUNC, CONST, DEP>
- OSPEC: ME の入出力形式等の外部仕様知識
- SQOS: ME が提供可能な QoS の知識
- FUNC: ME のパラメータ等の機能仕様知識
- CONST: 環境等による機能の制約知識
- DEP: 他の MEA との依存性に関する知識
- 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 との協調状態知識
- A-MEK ::= <ANA, AC, REL, ...>
- ANA: ME の状態を分析する知識
- AC: エージェント間プロトコル処理知識
- REL: 契約締結等のエージェント間の関連付けを行う知識

MEA は上記の知識をもとに、MEC で他のエージェントに送るメッセージを生成・送受信することで、エージェント間で協調動作し、利用者要求や資源状況に応じ、マルチメディアデータ通信の品質を適応的に調整する。

4.4 MSU のエージェント動作設計

エージェント組織の具体的な生成は、Fig.4 に示すように契約ネットプロトコルに基づく MEA 間の協調によって行う。以下に、エージェントの組織構成動作について述べる。

- (1) 利用者要求の発生に応じ、UA が MSU を構成する Manager に利用者要求を送る。
- (2) Manager は NETA と CPUA から、ネットワーク資源状況(空き帯域など)や CPU 資源状況(利用率など)に関する情報を得る。

- (3) Manager は、メディアの送受信を行うエージェント組織を構成するため、利用者要求・ネットワーク資源状況・CPU 資源状況(以降、契約条件と略記)を含むタスク要求通知メッセージ(TA)を生成し、データの獲得を行うエージェント組織を構成するために、全ての G-MEA に送信する。
- (4) TA を受信した G-MEA は、データの変換を行うエージェント組織を構成するために、全ての C-MEA に TA を送信する。
- (5) TA を受信した C-MEA は、さらに契約条件を満足するメディアデータの形式が現れるまで、全ての C-MEA に対して繰り返し TA を送信する。
- (6) 契約条件を満足するマルチメディアデータの形式が得られたら、データの送受信を行うエージェント組織を構成するために、全ての N-MEA に TA を送信する。
- (7) TA を受信した N-MEA は、メディアデータの再生を行うエージェント組織を構成するために、全ての P-MEA に TA を送信する。
- (8) (3)~(7)の手順で TA が伝搬された結果、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 やパラメータの条件、消費する資源の情報などが含まれる。
- (9) BID を受け取った Manager は、最適な BID を返したエージェント組織を選択し、選択したエージェント組織に落札通知(AWARD)を送信する。

以上の手順により、送受信側の双方を接続するエージェント組織が構成され、利用者要求を満たすマルチメディアデータの送受信が実現される。サービス提供中に利用者要求やネットワーク資源状況、CPU 資源状況の変動により

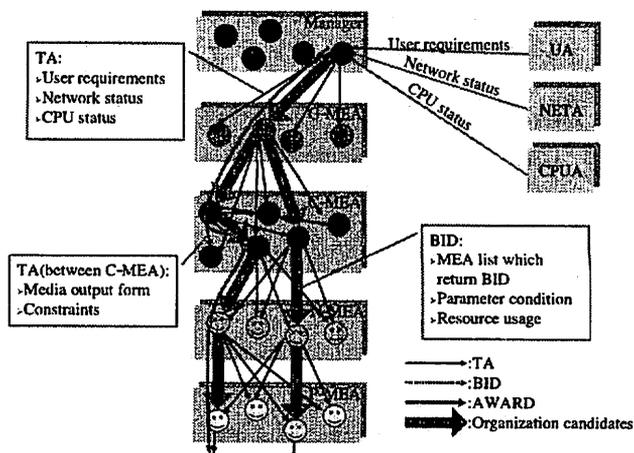


Fig.4 Behavior of MEA to configure organization

契約条件を満たせなくなった場合は、組織の再構成を行う。

5. 実装

提案システムの実装には、エージェントフレームワークとしてリポジトリ指向のマルチエージェントシステム基盤である ADIPS フレームワーク[3]を用いた。ADIPS フレームワークの最新バージョンである DASH1.1d[4]ではルール型知識表現が利用可能であるため、MEAの各知識をルール及びファクトとして実装する。また、DASH が Java オブジェクトのエージェント化をサポートしていることから、MEとしてはJMFを用いる。

DASH エージェントは、スクリプト型プログラミング言語で記述されエージェント間の組織構成処理を表現する DASH/S 知識と、ルール型プログラミング言語で記述されエージェントの知的処理を表現する DASH/R 知識の 2 種類の知識を持つ。また計算機処理を行うベースプロセスと DASH/R を拡張する DASH/R 言語拡張プログラムは Java を用いて記述されるので、DASH エージェントは DASH/S, DASH/R, Java の 3 つの言語を使用するハイブリッド型エージェントである。DASH/S 知識はリポジトリ内で組織構成プロトコルによって、エージェントの組織構成・再構成を行う際に使用される。一方 DASH/R 知識と Java プログラムはワークスペース内でエージェントが動作する際に使用される。

Fig.5 に DASH/S の記述例を示す。DASH/S 知識はエージェントの機能仕様や必要な資源などを表現する Specification knowledge, タスク分割を表現する Decomposition knowledge, 組織構成プロトコルにおける入札・落札の判断とベースプロセスに関する情報を表現する Contract knowledge から構成される。

エージェントによるメディアデータ処理機

```
//Specification Knowledge
(specification spec() {
  (spec (
    (function Sample1)
  ) )
})
//Decomposition Knowledge
(:decompose decomp1 :args())
)
(decompose decomp1() {
  (task-announcement :alias member1:
    content (----

```

Fig.5 Sample template of knowledge of DASH/S

```
....
(decompose deco() {
  (task-announcement: alias YtoR1: content
    (:dw 540 :dh 480 :maxDataLength 460800
     :DataType VideoFormat :frameRate 30.0
     :yuvtype 2 :strideY 320 :strideUV 160
     :offsetY 0 :offsetU 76800 :offsetV 96000)
  )
  (task-announcement: alias YtoR2: content
    (:dw 160 :dh 120 :maxDataLength 28800
     :DataType VideoFormat :frameRate 25.0
     :yuvtype 2 :strideY 160 :strideUV 80
     :offsetY 0 :offsetU 19200 :offsetV 24000)
  )
  (task-announcement: alias YtoR3: content
    (:dw 176 :dh 144 :maxDataLength 38016
     :DataType VideoFormat :frameRate 30.0
     :yuvtype 2 :strideY 176 :strideUV 88
     :offsetY 0 :offsetU 25344 :offsetV 31680)
  )
  (task-announcement: alias YtoR4: content ...

```

Fig.6 Example of knowledge of MEA

群の組合せの構成・再構成を実現するためには、4.3 にて述べた知識を DASH/S によって記述すればよい。たとえば、Fig.6 の例に示すように ME の知識を Decomposition knowledge として記述することによって実現できる。

6. おわりに

本稿では、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構成法について述べた。

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、それらを用いたエージェント組織の構成・再構成実験及び評価を行うことによって、本システムの有効性の確認を行う。

参考文献

- [1] T. Sukanuma, T. Kinoshita, and N. Shiratori, "Flexible Network Layer in Dynamic Networking Architecture," Proc of The 1st International Workshop on Flexible Networking and Cooperative Distributed Agents (FNCD2000), pp.473-478, 2000
- [2] A. Beugnard, J. Jezequel, N. Plouzean, and D. Watkins, "Making Components Contract Aware" IEEE Computer, pp38-45, July 1999
- [3] S. Fujita, H. Hara, K. Sugawara, T. Kinoshita, and N. Shiratori, "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
- [4] DASH -Distributed Agent System based on Hybrid Architecture!-, [Online]. Available: <http://www.agent-town.com/dash/index.html>