

RM-006

# 分散型 e-Learning システムにおけるマルチメディアコンテンツの利用 Utilization of Multimedia Contents on a Distributed e-Learning System

山本 大介<sup>†</sup>  
Daisuke YAMAMOTO  
本村 真一<sup>‡</sup>  
Shinichi MOTOMURA

目黒 一成<sup>‡</sup>  
Kazunari MEGURO  
笹間 俊彦<sup>†</sup>  
Toshihiko SASAMA

川村 尚生<sup>†</sup>  
Takao KAWAMURA  
菅原 一孔<sup>†</sup>  
Kazunori SUGAHARA

## 1. はじめに

近年、ウェブを用いた Web Based Training (以後、WBT と呼ぶ) と呼ばれる e-Learning システムが広く普及している。WBT において、学習者は、好きな時間に自らのペースで学習できる。これまでの WBT は、クライアントサーバモデルで構築されており、特定のサーバがすべてのサービスの提供を行っていた。そのため、データの管理や更新が容易であるという反面、利用者が増加するとサーバに負荷が集中し、応答性や安定性が低下してしまう欠点があった。

この欠点への対処として、Peer to Peer (P2P) モデルが提案されている。P2P モデルに基づくシステムは、特定のコンピュータをサーバとするのではなく、クライアントとサーバのいずれとしても機能するコンピュータ (以後、ノードと呼ぶ) で構成されているのが特徴である。これにより、P2P システムでは、複数のノードに負荷を分散することができる。また、いくつかのノードが故障してもシステム全体が停止することはない。このような P2P モデルに基づき、我々は分散型 e-Learning システムを提案している [1, 2]。提案システムではモバイルエージェント技術を用いることで、コンテンツと機能を複数のノードに分散している。

本稿では、提案システムを、マルチメディアコンテンツへ対応させた手法について述べる。マルチメディアコンテンツとは、マルチメディアデータを含む学習コンテンツのこととし、マルチメディアデータとは動画データと音声データのこととする。提案システムでは、学習コンテンツはエクササイズエージェント (以後、EA と呼ぶ) と呼ばれるモバイルエージェントが保持している。そこで、データサイズの大きなマルチメディアデータをどう扱うかが問題となる。学習コンテンツ内のマルチメディアデータをそのまま保持すると、EA は巨大化し、ノード間の移動に時間がかかるようになり、結果的に、学習者が学習コンテンツを要求してから取得できるまでの時間 (以後、応答時間と呼ぶ) が延びてしまう。そこで、本研究では、マルチメディアデータを EA から分離し、分離したマルチメディアデータは、新しく作成したエージェントに保持させた。

本稿は次のように構成される。第2章で我々が提案する分散型 e-Learning システムの概要を説明する。第3章でマルチメディアコンテンツへの対応の概要を示し、第4章、第5章でさらに詳しく説明する。第6章では実験について示す。最後に第7章で結論を述べ、本稿をまとめる。

<sup>†</sup>鳥取大学大学院 工学研究科 情報エレクトロニクス専攻

<sup>‡</sup>鳥取大学 総合メディア基盤センター

## 2. 分散型 e-Learning システム

### 2.1 概要

提案システムは、利用者として教師と学習者を想定している。教師と学習者は、それぞれ専用のユーザインタフェースを用いて、学習コンテンツを利用する。教師が学習コンテンツを作成、管理し、学習者がそれを学習する。

学習コンテンツは、問題、解答、解説で構成されている。また、学習コンテンツは、「数学/数学 A/確率」、「英語/文法/仮定法」のようにカテゴリに分類されている。学習者は学習したいカテゴリを指定することで、そのカテゴリ内の学習コンテンツを学習することができる。

カテゴリは、e-Learning システムに参加している各ノードに分散して管理されている。各ノードは、自身が担当しているカテゴリに属する学習コンテンツを保持しており、他のノードからの要求に応じて、適切な学習コンテンツをそのノードへ送信する。ここで、各ノードが担当しているカテゴリは、そのノード上で学習している学習コンテンツのカテゴリとは無関係である点が重要である。

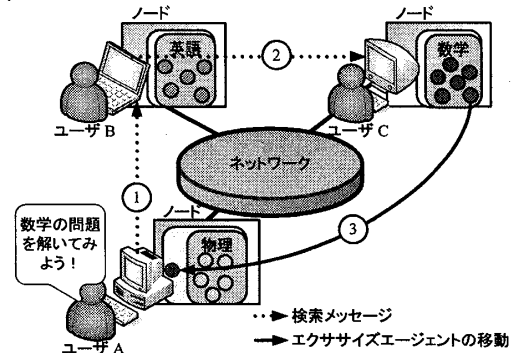


図 1: 学習コンテンツの移動

図 1 は、物理を担当しているノード上の学習者が数学の学習コンテンツを解こうとしている様子を示している。どのノードがどのカテゴリを担当しているかは各ノードにとって未知である。そのため、この例では、一度英語を担当しているノードに送られた要求が、数学を担当しているノードに転送されている。

### 2.2 P2P ネットワーク

どのノードがどのカテゴリを担当しているかは各ノードにとって未知であり、効率的な検索の方式が必要となる。そこで、提案システムでは、Content-Addressable Network (CAN) [3] に基づく分散ハッシュテーブルを用い、学習コンテンツの管理を行う。

提案システムでは、カテゴリ名をキーとして、全ノードで共通のハッシュ関数により二次元の直交座標空間上

の点に対応づける。システムに参加する各ノードは、図2に示すように座標空間を分担して担当する。図2では、二次元座標空間上にカテゴリが対応しており、領域をそれぞれノードA~Hが分担する様子を示す。

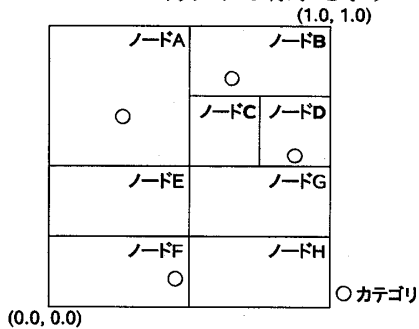


図2: ノード間の領域分割の様子

新たなノードがシステムに参加すると、すでに参加している他のノードのいずれかから、領域の半分を分け与えられる。逆に、あるノードがシステムから離脱する際には、そのノードが管理している領域を他の参加ノードに渡す。これらの領域の受け渡しには、その領域内の点に対応するカテゴリに属する学習コンテンツの受け渡しも含まれる。

提案システムに参加する各ノードは、自身の領域に接した領域を持つノード（以後、隣接ノードと呼ぶ）の情報のみを持つ。つまり、新たにノードが参加したときには、その情報は全ノードでなく、ごく一部のノードのみに伝えればよい。

学習コンテンツを要求するときは、まず、カテゴリを配置したときと同じハッシュ関数で、カテゴリ名から座標空間上の点を求める。求めた点が、自身の管理する領域に含まれていれば、目的の学習コンテンツは自身のノード上に存在することがわかる。自身の管理する領域に含まれていなければ、その点に近い領域を担当するノードに要求メッセージを転送する。メッセージを受け取ったノードは、最初のノードと同様に、目的の点が自身の管理する領域内か否かを確認し、領域内でなければ、さらにその点に近い領域を担当するノードに要求メッセージを転送する。これを繰り返すうちに、カテゴリを担当するノードに要求メッセージがたどり着く。このように、ノード数が多数の場合でも効率よく検索できる。

### 2.3 システムの構成要素

WBTシステムを分散システムとして実現するためには、学習コンテンツを分散させるだけでは不十分である。それに加えて、採点機能、正解や解説の表示機能なども分散させる必要がある。したがって、提案システムでは、学習コンテンツは単なるデータではなく、データに関連したプログラムを持ち、ノード間を移動して仕事を行うモバイルエージェントにより提供される。さらに、カテゴリもモバイルエージェントとすることで、カテゴリをノード間で受け渡しすることをカテゴリに対応するエージェントの移動で実現する。また同様にユーザ情報についてもエージェントが管理、提供する。さらにこれら学習コンテンツとユーザ情報については、管理しているノードに障害が発生した場合に備え、バックアップもシステム

に分散して配置しておく[4]。これにより、オリジナルが失われてしまっても、元のコンテンツを復元することができる。これらのエージェントは我々の開発しているモバイルエージェントフレームワーク Maglog[5] 上に実現している。システムを構成するエージェントの一例と、ユーザインタフェースを以下に示す。

ノードエージェント (NA) 分散ハッシュテーブルの領域情報の管理とメッセージのルーティングを行う。

カテゴリエージェント (CA) カテゴリごとに学習コンテンツを管理する。

エクササイズエージェント (EA) 学習コンテンツごとに存在し、ユーザへの問題提供および採点、解答解説を提供する。

インタフェースエージェント (IA) ユーザインタフェースと他のエージェント間の通信の橋渡しをする。

学習者用インタフェース ユーザが提案システムで学習するためのユーザインタフェース。

教師用インタフェース ユーザが学習コンテンツ管理機能を利用するためのユーザインタフェース。

## 3. マルチメディアコンテンツへの対応の概要

### 3.1 新たに対応する学習コンテンツの種類

本研究では、新たに、FLV形式、SWF形式、MP3形式に対応した。FLV形式、SWF形式は、Flash上で再生できる動画形式であり、Flashの普及率の高さから選択した。また、MP3形式も、音声圧縮形式として普及率が高いため選択した。

### 3.2 実現方法

本研究で対応した動画や音声のデータは、データサイズが大きくなることが予想され、これらのデータをどう保持するかが問題となる。

まず、EAがテキストデータと共にマルチメディアデータも保持、提供する方法が考えられる。この場合、EAが常にマルチメディアデータを保持して移動することとなるため、マルチメディアデータのサイズが大きくなると、応答時間が延びてしまう。また、学習コンテンツに含まれるマルチメディアデータの全てを、学習者が利用するとは限らないため、非効率である。以上の理由から、本研究では、マルチメディアデータをEAから分離する方法を提案する。

メディアエージェント (MA) と呼ぶエージェントを新たに作成し、EAから分離したマルチメディアデータを保持させる。以下で、MAについて詳しく説明する。

## 4. メディアエージェント

### 4.1 概要

学習コンテンツを保持、提供するEAと同様に、MAはマルチメディアデータを保持、提供し、参加ノードにより分散して管理される。そして、マルチメディアコンテンツを保持するEAは、システムに存在するMAを参照することで利用する。また、学習コンテンツを保持するEAと同様、MAもバックアップをシステムに分散して配置する。

#### 4.2 メディアエージェントのシステムへの追加

EAがe-Learningシステムへ追加された際に、自身が保持する学習コンテンツにマルチメディアデータを含んでいた場合、EAはそのマルチメディアデータを保持するMAを作成し、e-Learningシステムへ追加する。一つのMAにつき一つのマルチメディアデータを保持する。よって、MAは、マルチメディアデータの数だけ作成されることとなる。作成されたMAは、自身のキーに対応する分散ハッシュテーブル上の位置に移動し、単独で配置される。そして、そのMAを作成したEAが、MAのキーを管理する。EAがMAを作成しシステムへ追加する様子を、図3に示す。

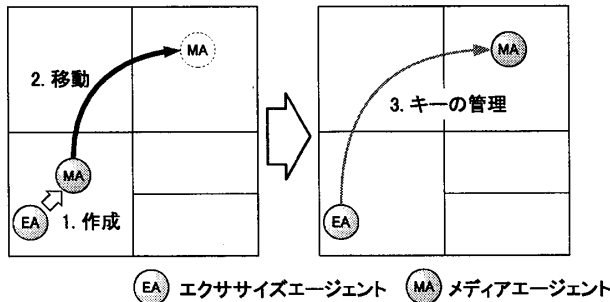


図3: メディアエージェントのシステムへの追加

#### 4.3 メディアエージェントの共有

別々の学習コンテンツにおいて、同じマルチメディアデータが必要となる場合がある。例えば、同じカテゴリ内の学習コンテンツであれば、解説の中の動画の一つとして、共通の動画を使用することが考えられる。このような場合に、毎回マルチメディアデータを作成することは非効率であるため、MAは複数の学習コンテンツで共有できるものとした。

EAがシステムへ追加され、MAを作成するときに、EAはまず、追加しようとしているMAがすでにシステム上に存在しないか確認する。このときにそのMAがシステム上に存在しなかった場合にだけ、EAはMAを作成する。つまり、学習コンテンツ作成者は、すでにシステム上に存在するMAが保持するマルチメディアデータを利用したいと考えた場合、そのMAのキーを学習コンテンツに含めるだけで利用することができる。

また、MAは、自身を参照するEAの数をカウンタとして保持しておく。EAがMAを作成する際、すでにシステム上に存在した場合には、そのMAのカウンタをインクリメントする。逆に、学習コンテンツの削除時に、EAをシステムから削除する際には、そのEAが参照しているMAのカウンタをデクリメントする。そして、このカウンタが0となり、どのEAからも参照されなくなった際に、MAはシステムから削除される。

### 5. 学習者用インタフェースにおけるマルチメディアコンテンツへの対応

#### 5.1 マルチメディアデータ取得の流れ

学習者が、学習コンテンツを要求すると、学習者用インタフェースは初め、IAに学習コンテンツ要求のメッセージを送る。IAは、要求された学習コンテンツのカ

テゴリに対応するCAをシステムから検索し、見つかったCAに対して学習コンテンツ要求のメッセージを送る。メッセージを受け取ったCAは、傘下のEAに、呼び出し元IAのいるノードへ移動するようメッセージを送る。そのEAは、IAのいるノードへ自身のコピーを派遣し、学習コンテンツのテキストデータを学習者用インタフェースに提供する。このとき学習者用インタフェースは、受け取ったテキストデータのみを表示する。

次に学習者用インタフェースは、バックグラウンドで、学習コンテンツに含まれるマルチメディアデータを順に要求する。マルチメディアデータが要求されると、学習コンテンツが要求されたときと同様にして、MAはIAによりシステムから検索され、自身のコピーをIAのいるノードへ派遣する。そして、保持するマルチメディアデータを提供する。最後に学習者用インタフェースがそれを表示する。

このようにして、最初はマルチメディアデータを表示せず、テキストデータのみを表示することで、学習コンテンツ内のマルチメディアデータの合計サイズが大きくなった場合にも、学習者は一定の待ち時間で学習に取り組むことができるようになる。マルチメディアデータ取得の流れを図4に示す。

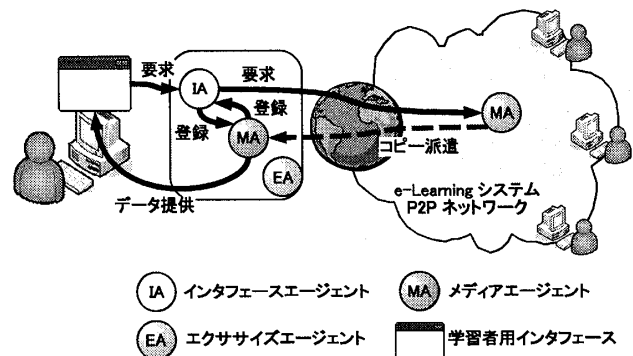


図4: マルチメディアデータ取得

#### 5.2 マルチメディアデータ取得失敗への対応

一つの学習コンテンツをテキストデータとマルチメディアデータとに分離することで、学習コンテンツの管理が複雑になるという欠点もある。学習者用インタフェースが、EAからのテキストデータの取得には成功したものの、マルチメディアデータの取得に失敗した場合、学習コンテンツの採点処理が問題となる。取得に失敗したマルチメディアデータが、学習コンテンツ内の問題を解くために必要な情報を含んでいた場合に、学習者は十分な情報を得られずに問題を解くことになる。さらに、教師は、学習者がマルチメディアデータの取得に失敗したことを知らずに学習履歴を参照することになる。

これを解決するため、学習者用インタフェースがマルチメディアデータの取得に失敗した際には、それと分かるフラグを立てることとする。教師用インタフェースと学習者用インタフェースは、学習履歴の表示時にこのフラグが立っていた場合には、マルチメディアデータの取得に失敗したことが分かる旨を表示する。これにより、教師や学習者は、後に学習履歴を参照したときに、マル

チメディアデータの取得に失敗した学習コンテンツを知ることができ、その後の処置を決定することができる。

## 6. 実験

### 6.1 実験内容

実験用に、マルチメディアデータを保持するEAを作成し、本研究で実装した、マルチメディアデータを分離するEAと、応答時間の比較を行う。

実験では、まず、一つのLANに含まれる12台のノードをシステムへ参加させ、学習コンテンツを分散させる。そして、あるノードから学習コンテンツの要求を行い、応答時間を記録する。ノード構成を変えずにこの記録を5回行い、その平均を実験結果とする。順にマルチメディアデータの大きな学習コンテンツを取得していき、応答時間の推移を確認する。実験に用いたPCの仕様を表1に示す。表において、括弧内の数字は実験に用いたノード番号を表す。

表1: 実験用PC仕様

CPU (1~8)	Intel Pentium4 2.53 GHz
CPU (9~12)	Intel Pentium4 2.66 GHz
主記憶 (7,8以外)	512MB
主記憶 (7,8)	256MB
ネットワーク	100BASE-TX
OS (1~9)	Debian GNU/Linux 4.0
OS (10~12)	Ubuntu 8.04 LTS
Javaバージョン (1~9)	1.5.0
Javaバージョン (10~12)	1.6.0

### 6.2 実験結果

実験により得られた平均応答時間を、図5に示す。

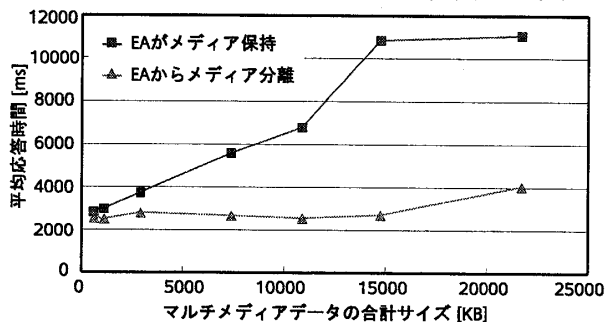


図5: 平均応答時間

EAがマルチメディアデータを保持した場合は、マルチメディアデータの合計サイズの増加につれて応答時間も増加しているのに対し、分離した場合には、応答時間がある一定の時間で抑えられていることがわかる。

### 6.3 考察

要求した学習コンテンツが、分散ハッシュテーブル上のどこに配置されているかによって応答時間が多少変化することはあるが、この実験結果によりおおよその傾向を確認できる。EAからマルチメディアデータを分離すると、学習コンテンツ内のマルチメディアデータのサイズに影響されることなく、学習者は、学習コンテンツを要求してから一定の時間で学習に取り組み始めることができる。

## 7. おわりに

本稿では、我々が提案する分散型e-Learningシステムのマルチメディアコンテンツへの対応について述べた。この対応によって、学習コンテンツ内に、静止画データに加えて動画データや音声データを含めることが可能となった。これらのデータは静止画データに比べてサイズの増加が予想されるため、マルチメディアデータは新たに作成したMAに保持させ、EAからは分離した。これにより、マルチメディアデータのサイズが増加しても一定の時間で学習コンテンツが表示され、学習者はすぐに問題に取り組みめるようになった。

現在、マルチメディアデータの再生は、それを保持しているMAの移動が完了するまで開始されない。しかし、今回対応したマルチメディアデータは動画データと音声データであり、一度に全てのデータが必要になることはなく、再生する部分のデータのみがあればよい。そこで今後の課題として、マルチメディアデータを断片に分割し、MAはその断片を保持することを考える。そして、要求があった際には、断片を保持するMAが学習者のノードへ集結し、すべての断片が揃うのを待たずに、集まった断片から再生していく。これにより、学習者はスムーズに学習を継続できるようになる。

## 参考文献

- [1] 川村尚生, 菅原一孔: モバイルエージェントに基づくP2P型e-Learningシステム, 情報処理学会論文誌, Vol. 46, No. 1, pp. 222-225 (2005).
- [2] Motomura, S., Nakatani, R., Kawamura, T. and Sugahara, K.: Distributed e-Learning System Using P2P Technology, *Proceedings of the 2nd International Conference on Web Information Systems and Technologies*, pp. 250-255 (2006). Setubal, Portugal.
- [3] Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Schenker, S.: A scalable content-addressable network, *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications*, ACM Press, pp. 161-172 (2001).
- [4] Kawamura, T., Sugahara, K., Kinoshita, S., Nakatani, R. and Motomura, S.: Backup and Recovery Scheme for Multi-Agent-based e-Learning System, *Proceedings of IEEE International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pp. 73-78 (2007). Waltham, Massachusetts, USA.
- [5] Motomura, S., Kawamura, T. and Sugahara, K.: Logic-Based Mobile Agent Framework with a Concept of "Field", *IPJS Journal*, Vol. 47, No. 4, pp. 1230-1238 (2006).