RM-009

分散型 e-Learning システムのマルチメディアデータ管理

Management of Multimedia Data for Distributed e-Learning System

目黒 一成† 山本 大介† 本村 真一‡
Kazunari Meguro Daisuke Yamamoto Shinichi Motomura
笹間 俊彦† 川村 尚生† 菅原 一孔†
Toshihiko Sasama Takao Kawamura Kazunori Sugahara

１．はじめに

e-Learning とはコンピュータやネットワークを利用した学習形態のことであり，近年のネットワークの普及に伴い，Web-Based Training(以下，WBT)と呼ばれるネットワークを用いた e-Learning システムが普及しつつある．WBT では，学習者は自らの都合に合わせた学習が可能である．これまで WBT に関する研究が数多くなされているが[1, 2, 3]，それらは全てクライアントサーバモデルに基づいている． クライアントサーバモデルに基づく WBT では，サーバが全てのサービスを提供し，データを管理するため，構築や管理が容易である．しかし，クライアントサーバモデルは，利用者が増加するにつれサーバへの負荷が増加し，応答性や安定性が低下してしまう．

クライアントサーバモデルの欠点を補うものとして Peer to Peer(以下，P2P)モデルが提案されている．P2P モデルに基づく WBT では，システムに参加するコンピュータ(以下，ノード) が，クライアントとサーバ両方の機能を提供するといった特徴がある．この特徴により，システムに掛かる負担を参加ノードに分散することができ，あるノードが故障した場合であっても，システム全体が停止することが無い．更には，サーバのような高性能のコンピュータを必要としないため，クライアントサーバモデルと比較して，安価にシステムを構築できる．

我々はこれまで P2P モデルに基づくシステムを提案している[4, 5]．提案システムには 2 つの特徴がある．1 つ目の特徴は，P2P モデルに基づいているため，システムに参加するノードはクライアントとサーバの役割を果たすことである．あるノードがシステムに参加する際は，既にシステムに参加しているノードから学習コンテンツの一部を受け取る．そして受け取った学習コンテンツを管理し，また学習コンテンツを要求された場合には，要求された学習コンテンツを送信する役割を果たす．2 つ目の特徴として，提案システムにおける学習コンテンツはエージェントを用いて実現するため，学習結果の採点，正答の提示や正答に関連する情報の提示を行う機能を有する．提案システムにおいて，エージェントは学習コンテンツ単位で作成され，システム内のノードに配布される．学習者が学習コンテンツを学習する際には，学習コンテンツを保持するエージェントのコピーが学習者のノードへ送られる．

しかし，これまでの提案システムにおける学習コンテンツは，テキストデータのみを取り扱うことができ，画像，音声や動画を含んだマルチメディアデータに対応していなかった．そこで本稿では，提案システムの学習コンテンツにテキストデータだけでなく，画像，音声や動画を含んだマルチメディアデータに対応させた手法について述べる．

提案システムのマルチメディアデータ対応を行うにあたり，2 つ問題を解決しなければならない．1 つ目の問題は，マルチメディアデータのサイズにある．エージェントが学習コンテンツデータの中に全てのマルチメディアデータを保持する場合，マルチメディアデータのサイズ増大に伴ってエージェントのサイズも増大する．エージェントのサイズが増大することにより，エージェントが学習者のノードに送られるまでの時間が増加し，学習者が学習を開始できるまでの時間が遅くなる．この問題の解決方法として，テキストデータを保持するエージェントとそれ以外のマルチメディアデータ(以下，本稿ではテキストデータを含まないデータのことをマルチメディアデータと呼ぶ)を保持するエージェントとに分ける．この方法により，学習者が大きなデータを含む学習コンテンツを学習する場合であっても，比較的短い時間で学習を開始することが可能になる．2 つ目の問題は，学習コンテンツに含むデータがマルチメディアデータであった場合の再生開始時間にある． 1 つのエージェントがマルチメディアデータを保持する場合，エージェントが学習を要求したノードまで完全に移動するまでは再生が始まらない．つまり，エージェントが保有するマルチメディアデータのサイズ増大に伴い，これらデータの再生開始までの時間が遅くなる．この問題の解決方法として，マルチメディアデータを一定のサイズに分割し，エージェントが各分割マルチメディアデータをそれぞれ保持する．学習者がマルチメディアデータを要求した際には，分割されたデータをストリーミング方式によって再生する．この方法により，マルチメディアデータの再生開始時間は短縮される．

本稿は 6 つの章で構成される．第 2 章では，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 章では，学習コンテンツをマルチメディアデータに対応させるための設計について示し，第 4 章ではマルチメディアデータのストリーミング再生の設計について示す．第 5 章では，第 3 章および第 4 章の内容を実装したシステムでの実験結果を示し，第 6 章で結論を述べ，本稿をまとめる．

†鳥取大学大学院 工学研究科情報エレクトロニクス専攻
‡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 分散型 e-Learning システムの構成要素

分散型 e-Learning システムを実現するには，e-Learning の機能をシステムに参加する全てのノードに分散させなければならない．我々は，モバイルエージェント技術を用いてこれを実現する．

提案システムでは，各ノード上に以下のエージェントとユーザインタフェースを実装する．エージェントは，我々が開発しているモバイルエージェントフレームワーク上で実装している[6]．

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，学習者への問題提供，採点実施および解答解説を提供する．

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．

ユーザエージェント(UA)：学習者ごとに存在し，学習者の学習履歴，ログイン名やパスワードなどの情報を管理する．

学習者用インタフェース：学習者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プログラム

2.2 P2P ネットワーク

提案システムにおいて，全ての学習コンテンツは，「数学/統計」や「英語/文法」などといったカテゴリごとに分類する．学習者は，必要とする学習コンテンツのカテゴリを特定した後に学習コンテンツを取得することができる．提案システム利用中の学習者は，提案システムの一部としての役割を担う．提案システムが起動する際，最初に参加するノードが全てのカテゴリを管理する．次に別のノードが提案システムに参加する際は，最初に参加したノードからカテゴリを分配される．システム内のカテゴリは，ノードがシステムに参加する場合または参加しているノードがシステムから離脱する場合に，システム内に参加しているノードに分配される．

Napster[7]，Gnutella[8]や Freenet[9]といった既存の P2P ファイル共有システムでは，各共有ファイルは特定のノードが管理する．従って，これらのシステムにおけるファイルは，最初に全ての参加ノードに分配される．一方で，提案システムのカテゴリファイルは，最初は特定のノードのみが保有する．新たなノードが提案システムに参加する際には，カテゴリファイルの位置情報だけでなく，カテゴリファイルそのものが新たに参加するノードに配布される．どのノードがどのカテゴリを管理しているかは，各ノードにとって未知であり，効率的な検索の方式を必要とするため，提案システムでは Content-Addressable Network（以下，CAN)[10]に基づく Distributed Hash Table(以下，DHT)を用い，学習コンテンツの管理を行う．

我々の P2P ネットワークは，カテゴリを保持するために，図 1 に示す[0.0, 1.0]の 2 次元座標空間を用いて構成される．図 1 中のノード C は，３番目にシステムへ参加するノードとして示している．ノード C がシステムに参加する以前には，ノード A とノード B が座標空間全体を半分に分割して管理する．その際，ノード A が「数学/統計」，「数学/幾何学」および「英語/リスニング」を管理し，ノード B が「英語/文法」，「歴史/地理」および「歴史/日本史」を管理する．ノード C が新たにシステムへ参加する際，ノード C にはランダムな座標点が与えられ，その座標点を管理するノードから管理領域の半分を与えられる．

例えば図 1 の場合，システムに参加したノード C はノード B から管理領域の半分を与えられ，与えられた管理領域にマッピングされた「歴史/日本史」の学習コンテンツを新たに管理する．その後，ノード C は自身の管理領域に隣接した領域を管理するノード(図 1 の例ではノード A およびノード B)の IP アドレスと管理領域の情報をルーティングテーブルとして登録する．システムに参加するノードが自身の隣接ノードを登録することで P2P ネットワークを構成する．

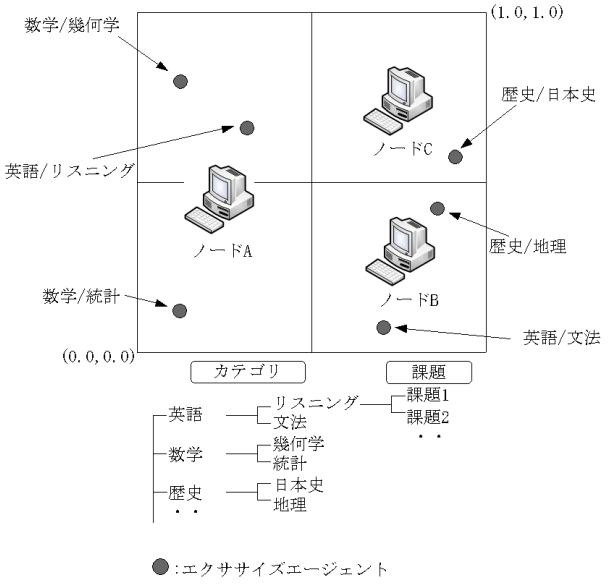

図 1 提案システムの P2P ネットワーク

3. 大容量マルチメディアデータへの適用

これまでの提案システムでは，テキストデータのみを学習コンテンツに含めることが可能であった．テキストデータ以外のマルチメディアデータを学習コンテンツに含めさせるため，FLV 形式，SWF 形式，MP3 形式のデータを新たに対応させる．しかし，学習コンテンツに音声および動画のデータ形式を対応させることにより，これまで同様に EA が全ての学習コンテンツを保持する場合，EA のサイズが増加することが想定される．EA のサイズが増加するにつれ，学習者のもとへ EA を派遣する際に掛かる時間も増加してしまう．それゆえ，新たにメディアエージェント(以下，MA)を提案システムに実装し，マルチメディアデータを EA から分離して管理させる．MA も EA 同様，提案システムに参加するノードによって管理される．MA は，MA が管理するマルチメディアデータを学習コンテンツに含む EA によって参照される．一般に，e-Learning システムでは，異なる学習コンテンツが同一のマルチメディアデータを利用する場合があるため，この場合には 1 つの MA を複数の EA が参照できるようにする．MA は以下の仕組みによって管理される．

MA の生成

EA が生成される際，学習コンテンツ内にマルチメディアデータを含むならば，以下手順に従い MA を生成する．図 2 に MA 生成の流れを示す．

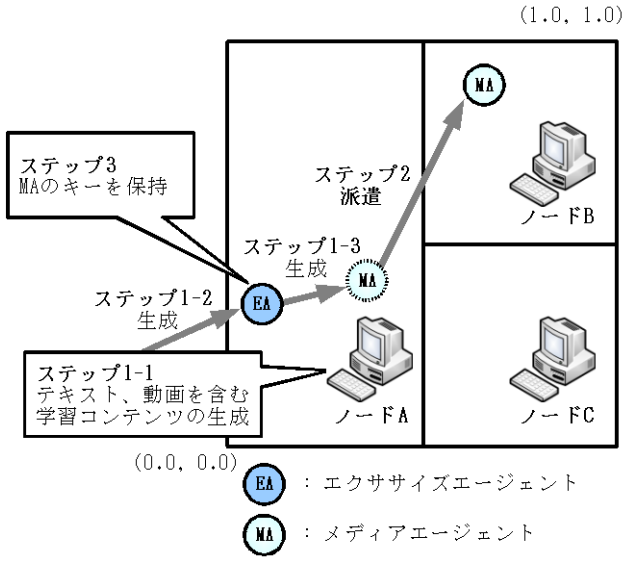

図 2 MA 生成時の処理

1. 生成された EA は，MA を生成する．この MA は 1 つのマルチメディアデータに対し 1 つ生成される．
2. 生成された MA は，マルチメディアデータ名をキーとして，DHT 上の座標にマッピングされる．
3. MA を生成した EA はマルチメディアデータ名のキーを管理する．これにより，EA は他のノードが管理する MA を要求することが可能となる．

MA の共有

MA は以下手順に従い共有される．

1. EA は MA を生成する前に，マルチメディアデータ名のキーが既にシステム内に存在するか調査する．
2. 上記 1 によって，マルチメディアデータのキー名が存在した場合，EA はマルチメディアデータのキーを保持する．
3. 上記 1 によって，マルチメディアデータ名のキーがシステム内に存在していなかった場合，新たに MA を生成し，そのキーを保持する．

MA の削除

MA は学習コンテンツによって共有されるため，全ての学習コンテンツが参照しなくなった場合にのみ，削除することができる．MA は以下手順に従い削除される．

1. MA は自身を参照する EA の個数をカウンタとして保持する．EA が MA を生成する際，マルチメディアデータ名のキーが既にシステム内に存在する場合には，EA は自身が参照する MA のカウンタをインクリメントする．
2. EA がシステムから削除される際，この EA が参照する MA のカウンタをデクリメントする．
3. 上記 2 において，MA のカウンタが 0 となった場合は，MA をシステムから削除する．

MA の取得

提案システムにおいて学習者は学習者用インタフェースを用いて学習を行う．学習者が学習コンテンツを要求すると，MA は以下手順に従い提供される．図 3 に MA 取得時の流れを示す．

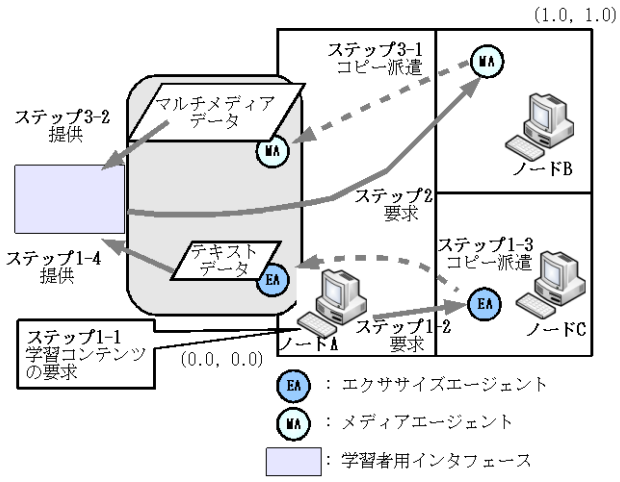

図 3:MA 取得時の処理

1. 初めに，要求された学習コンテンツに対応する EA のコピーが学習者のノードに派遣される．その後，EA が保持する学習コンテンツのテキストデータが学習者インタフェース上に表示される．
2. 学習者用インタフェースは，学習コンテンツに含まれるマルチメディアデータをバックグラウンドで順次要求する．その後，要求されたマルチメディアデータを保持する MA を検索する．
3. 検索によって見つかった MA の複製が要求元のノードへ派遣され，その MA が保持するマルチメディアデータが学習者用インタフェースに提供され，学習者に提示される．

提案システムでは，学習者のノードへ EA が派遣されていれば，そのノードにある学習者用インタフェースに学習コンテンツを表示することができる．一方，学習を開始する際は一般的に文章を読むことから始まる．従って，上記手順を用いることにより，EA はテキストデータのみを保持することが可能となり，学習コンテンツ内のマルチメディアデータのサイズが大容量になった場合であっても，学習者は一定の待ち時間で学習を開始することができる．

MA 取得が失敗した際の処理

実用的な e-Learning システムを実装する上では，失敗時の対策も必要となる．EA の取得は成功し，その後の MA の取得に失敗する場合である．学習コンテンツに解答するにあたり，マルチメディアデータの内容が必要な問題で会った場合，学習者は十分な情報を得ずに問題を解答することになる．この問題に対して，学習コンテンツの取得に失敗した場合には，学習履歴にコンテンツ取得失敗のフラグを記録することで解決を行う．

4. 分割マルチメディアデータのストリーミング再生

第 3 章では MA の仕組みについて述べた．しかし，この仕組みでは，マルチメディアデータが学習者のノードへ移動し終えるまで，これらのデータが再生されないといった問題がある．この問題を解決するため，我々はマルチメディアデータが学習者のノード上でストリーミング再生する手法を提案する．この手法を実現するには，以下の方法で MA を管理する．

分割した MA の生成

EA が生成される際，学習コンテンツにマルチメディアデータを含むならば，以下手順に従い，分割した MA を生成する．

1. EA はマルチメディアデータを分割し，それぞれを管理する MA(以下，分割 MA)を生成する．
2. 生成された分割 MA は，自身が管理するマルチメディアデータのキーに対応する DHT の座標空間上にそれぞれがマッピングされる．
3. 各分割 MA は，再生順序が自身の前後となる分割 MA のキーを保持する．
4. 分割 MA を生成した EA は，分割したマルチメディアデータのキーを保持する．

分割した MA の削除

分割 MA は学習コンテンツが共有する．したがって第 3 章で述べたように，全ての学習コンテンツが分割 MA を参照しなくなった際に分割 MA を削除することが可能となる．

分割した MA の取得

分割 MA は以下手順に従い，学習者のノードへ提供される．図 4 に MA が提供される際の手順を示す．

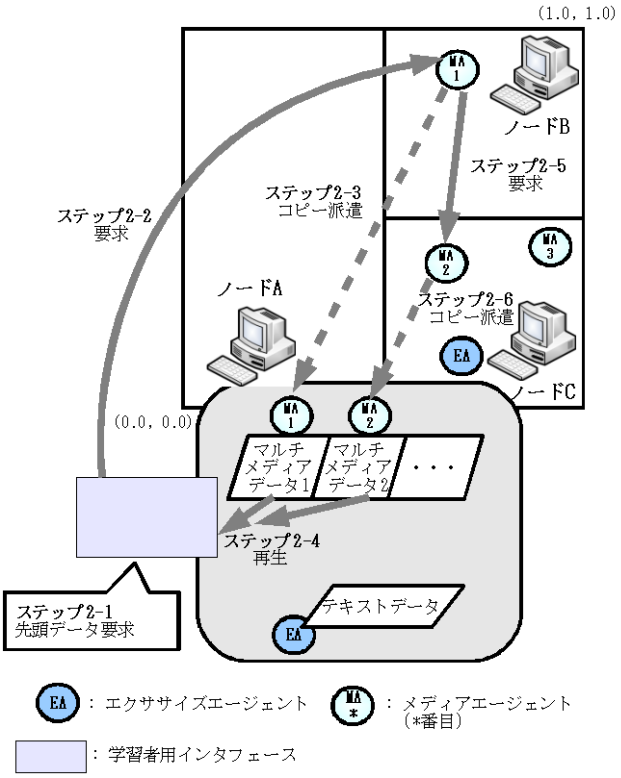

図 4：分割した MA 取得時の処理

1. 要求された学習コンテンツを管理する EA の複製を要求元のノードへ送信する．その後，学習者用インタフェースには EA が管理するテキストデータが表示される．
2. 上記 1 の後，学習者用インタフェースは，学習コンテンツ内のマルチメディアデータの先頭データをバックグラウンドで要求する．要求された分割データは，自身の次に続く分割データを要求する．
3. 要求された各分割データは，自身のコピーを要求元のノードへ送信する．学習者用インタフェース上では，分割されたマルチメディアデータの先頭データが届き次第，ストリーミング方式により再生が行われる．

この方法により，全ての分割 MA が要求元のノードへ移動しなくても，学習者はマルチメディアデータを再生することが可能となる．

5. 実験

この章では，本稿で述べた MA の動作検証結果について述べる．まず，EA からマルチメディアデータを分離した場合の動作検証を行い，次に，マルチメディアデータを保持する MA を分割した場合の動作検証を行った．

5.1 マルチメディアデータを EA から分離

提案システムに MA を実装した際の動作検証として，以下を確認する実験 1 を行った．

・マルチメディアデータを含む学習コンテンツを正しく取得できること
・MA が複数の学習コンテンツで共有できること
・マルチメディアデータの取得が失敗した際には，失敗した記録が行えること

表 1 および表 2 として，実験 1 での実験環境およびマシンスペックを示す．

表 1 実験 1 の実験環境

参加ノード数	5
カテゴリ数	20
学習コンテンツ数	60

表 2 実験 1 マシンスペック

CPU	Intel Pentium4 3.0GHz
メモリ	1GB
ネットワーク	1000BASE-T
OS	TurboLinux 11 Desktop

実験 1 の後，実験 2 として，全てのマルチメディアデータを EA が保持する場合と，マルチメディアデータは MA が保持する場合との応答時間の比較を行った．ここでの応答時間とは，学習者が学習コンテンツを要求してから学習インタフェース上に学習コンテンツが表示されるまでの時間をいう．

表 3 および表 4 として，実験 2 での実験環境およびマシンスペックを示す．

表 3 実験 1-2 の実験環境

参加ノード数	12
カテゴリ数	28
学習コンテンツ数	84

表 4 実験 1-2 のマシンスペック

CPU(ノード 1～8)	Intel Pentium4 2.53GHz
CPU(ノード 9～12)	Intel Pentium4 2.66GHz
メモリ(ノード 7, 8)	512MB
メモリ(上記以外)	256MB
ネットワーク	1000BASE-T
OS(ノード 1～9)	Debian GNU/Linux 4.0
ノード(上記以外)	Ubuntu 8.04 LTS

5.1.1 実験 1 の結果

1 つ目の確認項目に関して，マルチメディアデータを含む学習コンテンツを要求した際，学習者用インタフェース上に学習コンテンツが正しく表示された．次に，学習コンテンツが参照する MA の個数を確認したところ，1 つの EA が 5 つのマルチメディアデータを参照しているものが 9 つ存在した．しかし，MA の個数を確認したところ，生成された MA は 30 個であった．これにより，同一の MA が複数の学習コンテンツで共有されていることを確認できた．その後，ある学習コンテンツに参照されている MA を故意に削除し，学習コンテンツを要求した．この結果，学習者の学習履歴において，マルチメディアデータの取得に失敗した記録が正しくなされていることを確認した．

5.1.2 実験 2 の結果

図 5 に実験 2 の結果を示す．保持するマルチメディアデータのサイズに関わらず，応答時間は，EA がマルチメディアデータを保持するよりも，MA がマルチメディアデータを保持する方が速くなった．また，MA にマルチメディアデータを保持させることで，マルチメディアデータのサイズが大きくなった場合であっても，応答時間は EA がマルチメディアデータを保持した場合に比べて大幅な変化は無かった．すなわち，提案システムに MA を実装することにより，学習効率は向上するといえる．

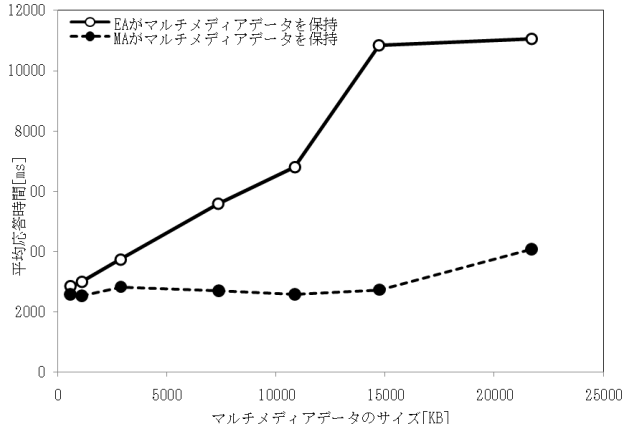

図 5 マルチメディアデータを EA が保持する場合と MA が保持する場合との応答時間の比較結果

5.2 複数の MA が分割したマルチメディアデータを保持

我々は実験 3 として，MA の分割サイズがマルチメディアデータの再生開始時間に与える影響についても調査した．表 5 に実験環境を示す．実験 3 は以下手順に従って行った．

1. 初めに，20 台のノードをシステムに参加させる．
2. 参加させたノードのうち，1 台のノードで学習者用インタフェースを起動する．その後，この学習者用インタフェース上でマルチメディアデータの取得を要求する．
3. マルチメディアデータの要求開始から，学習者用インタフェース上での再生が始まるまでの時間を計測する．
4. 上記 3 の計測は，計測する学習者用インタフェースおよび要求するマルチメディアデータをランダムに変更し，マルチメディアデータのデータサイズごとに 5 回ずつ行う．

実験に用いたマルチメディアデータのサイズは，17.2MB，34.8MB および 50.5MB とした．マルチメディアデータの分割サイズは，0MB(分割無し)，5MB，10MB および 20MB とした．

表 5 実験 3 のマシンスペック

CPU	Intel Pentium4 3.0GHz
メモリ	1GB
ネットワーク	1000BASE-T
OS	TurboLinux 11 Desktop

図 6 に実験 3 の実験結果を示す．実験の結果，マルチメディアデータの分割サイズが小さくなるほど，学習者用インタフェース上での再生時間が短くなることが示された．マルチメディアデータの再生開始時間が短くなることは，学習者が学習コンテンツ内のマルチメディアデータを取得するまでの時間が短くなるため，効率の良い学習を行うことが可能となる．

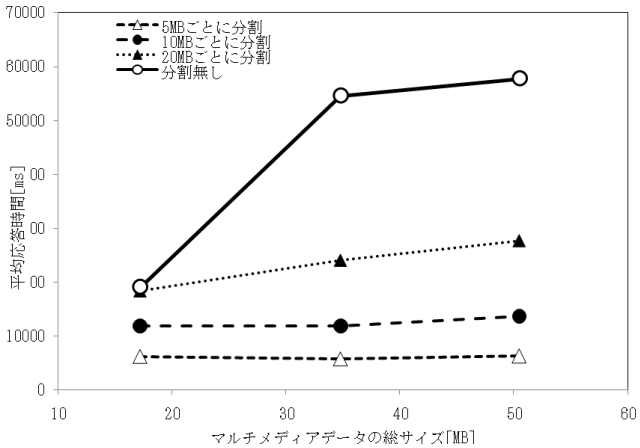

図 6 マルチメディアデータの分割サイズごとの再生開始時間の比較

6. おわりに

本稿では，我々が提案する分散型 e-Learning システムにマルチメディアデータを対応させる手法について述べた．結果として，テキストデータに加え，音声データおよび動画データを含めることが可能となった．音声データまたは動画データは，テキストデータに比べてデータサイズが大きくなることが想定されるため，これらのデータを保持するための MA を新たに実装した．この結果，学習コンテンツに音声や動画などの大容量データを含む場合であっても，一定の時間で学習コンテンツを学習者用インタフェースに表示することが可能となった．更に，マルチメディアデータを一定のサイズに分割させること

により，大容量のマルチメディアデータであっても再生開始までの時間を短縮させることが可能となった．

提案システムでは，学習者のノードが分割したマルチメディアデータを保有する際，そのノードのメモリ領域を使用する．ノードが大量の分割マルチメディアデータを保有することは，そのノードのパフォーマンス低下につながり，システム全体の効率を低下させることも考えられる．この問題を解決するため，ノードが自身のメモリ使用状況を管理し，動的に分割マルチメディアデータの取得および解放する仕組みについて今後の課題として調査し，実装する必要がある．

参考文献

[1]Nishita, T. and et al,: Development of a Web Based Training system and Courseware for Advanced Computer Graphics Courses Enhanced by Interactive Java Applets, *Proceedings of International Conference on Geometry and Graphics*, Vol. 2, pp. 123-128 (2002).

[2]Homma, H. and Aoki, Y.: Creation of WBT Server on Digital Signal Processing, *Proceedings of 4th International Conference on Information Technology Based Higher Education and Training*, (2003). Marrakech, Morocco.

[3]Helic, D., Krottmaier, H., Maurer, H. and Scerbakov, N.: Enabling Project-Based Learning in WBT Systems, *International Journal on E-Learning*, Vol. 4, No. 4, pp. 445-461 (2005). Norfolk, VA.

[4]Kawamura, T. and Sugahara, K.: A Mobile Agent-Based P2P e-Learning System, *IPSJ Journal*, Vol. 46, No. 1, pp. 222-225 (2005).

[5]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.

[6]Motomura, S., Kawamura, T. and Sugahara, K.: Logic-Based Mobile Agent Framework with a Concept of “Field”, *IPSJ Journal*, Vol. 47, No. 4, pp. 1230-1238 (2006).

[7]Napster, http://www.napster.com (1999).

[8]Gnutella, http://welcome.to/gnutella/ (2000).

[9]Clarke, I., Sandberg, O,. Wiley, B. and Hong, T. W.: Freenet: A Distributed Anonymous Information Storage and Retrieval System, http://freenetproject.org/papers/freenet.pdf (2000).

[10]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).