

LL-006

分散型 e-Learning システムにおける学習コンテンツ管理機能 Management Scheme of Learning Materials on a Distributed e-Learning System

倉持 一生[†] 本村 真一[†] 川村 尚生[‡] 菅原 一孔[‡]
Kazuo KURAMOCHI Shinichi MOTOMURA Takao KAWAMURA Kazunori SUGAHARA

1. はじめに

e-Learning システムとは、コンピュータを利用した学習の総称であり、本稿では、このうち非同期 Web Based Training (以後、WBT と呼ぶ) と呼ばれるものを対象とする。WBT において、学習者はネットワークを通じて取得できる学習コンテンツを用いて、好きな時間に学習を行うことができる。

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、サーバがすべてのサービスを提供する。このモデルのシステムは、構築やデータの管理、更新が行いやすいという利点がある。しかしシステムを利用する学習者が増加するにつれ、サーバへの負荷集中による応答性や安定性の低下が問題となる。

最近、これらの問題を解決する P2P システムの研究がさかんになっている。P2P システムは、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) で構成される。したがって、P2P システムでは、サービスを多数のノードに分散させることで負荷分散を図ることができる。また万が一、いくつかのノードが故障してもシステム全体が使用不可能になることはない。

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て構築することで、コンテンツと機能を複数のノードに分散することを実現している。本稿では、提案システムにおける学習コンテンツの追加、削除、更新を実現する学習コンテンツ管理機能について述べる。

クライアントサーバモデルによる WBT システムでは、提供される学習コンテンツは単にデータとしてすべてサーバが保持する。そのため学習コンテンツの追加、削除、更新はサーバ上に配置されるデータを直接操作することで容易に実現できる。これに対し提案システムでは、学習コンテンツはエージェントによって保持、提供される。エージェントは、ネットワーク上の各ノードに分散配置され、ノードの参加や離脱に伴い、ノード間を移動する。そのため、学習コンテンツが存在するノードは定まっておらず、動的に変わる。そこで、提案システムにおいても従来の WBT と同様の学習コンテンツ管理を実現するため、学習コンテンツ管理の依頼ノード内の操作に加え、他ノード内のエージェントと協調した次の操作を併せて行う。

1. 対象エージェントとその管理ノードの特定、およびエージェントの取得からなる前操作
2. システムへのエージェントの登録からなる後操作

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 章では学習コンテンツ管理機能の概要と構成方法を、第 4 章では学習コンテンツ管理機能の実装を示す。最後に第 5 章で結論を述べ、本稿をまとめる。

2. 分散型 e-Learning システム

2.1 概要

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多数存在する。提案システムにおける学習コンテンツは問題、正解、解説および問題の解答履歴で構成されている。学習者は特定の学習コンテンツではなく、学習したいカテゴリを指定することで、そのカテゴリに属するいずれかの学習コンテンツを得ることができる。また学習中の問題について、答案の採点および正解の表示を求めることができる。

システムに接続している間、ノードはいくつかのカテゴリを担当する。すなわち、該当カテゴリに属するすべての学習コンテンツを保持しており、他のノードからの求めに応じて適切な学習コンテンツを送り出す責任を負う。ここでは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者が解いている問題のカテゴリとは無関係である点が重要である。

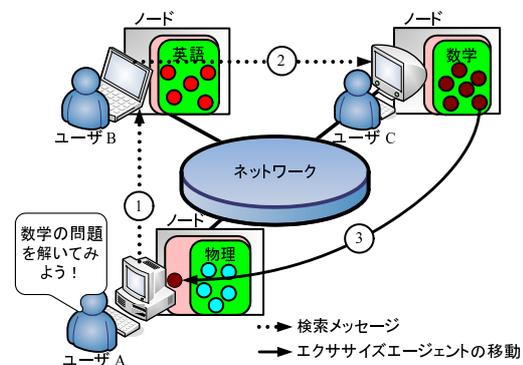


図 1: 学習コンテンツの移動

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り、この例においては、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。

学習コンテンツと同様に、学習者ごとの学習履歴などのユーザ情報も各ノードに分散される。ユーザはグループに分けられ、グループごとに各ノードが分担してユーザ情報を保持する。各ノードでの分担について、グルー

[†]鳥取大学大学院工学研究科
[‡]鳥取大学工学部

ブとユーザ情報との関係は、カテゴリと学習コンテンツの関係に対応する。

2.2 P2P ネットワーク

学習コンテンツを取得する際、所望のカテゴリをどのノードが担当しているか、すべてのノードに問い合わせたのでは効率が悪い。またすべてのカテゴリと担当するノードの対応を、全ノードが知っているとする、担当ノードの問い合わせは不要となる。しかしカテゴリを担当するノードが変われば、全ノードに伝えなければならず、効率が悪い。そこで提案システムでは、Content-Addressable Network (CAN) [3] をベースとした分散ハッシュテーブルを用い、学習コンテンツやユーザ情報の管理を行う。

提案システムでは、カテゴリ名とグループ名を、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とグループが対応しており、領域をそれぞれノードA~Hが分担する様子を示す。

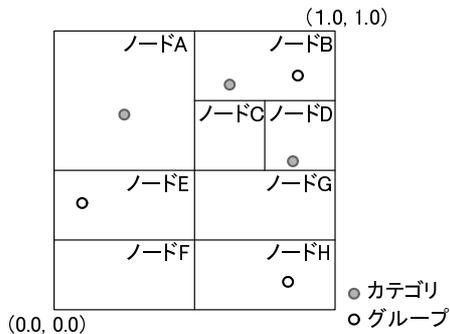


図2: ノード間の領域分割の様子

あるノードが新たに提案システムに参加するときには、すでに参加しているいずれかのノードから、領域の半分を受け取る。またノードがシステムから離脱するときには、参加する他のノードに領域を渡す。領域の授受には、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や、グループに属するユーザ情報の授受も含まれる。

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、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、その点に近い点を担当するノードに要求メッセージの転送を依頼する。依頼が転送されるうちに、カテゴリを担当するノードに依頼がたどり着く。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には、学習コンテンツを分散させるだけでは不十分で、学習コンテンツ内の問題に対する答案の採点機能、正解や解説

の表示機能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、ノード間を移動するモバイルエージェントにより提供する。加えて、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す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。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[4] 上に実現している。各ノードには次に示すエージェントおよびユーザインタフェースプログラムが存在する。

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。

グループエージェント (GA) ユーザのグループごとにユーザエージェントを管理する。またユーザのログイン認証を行う。

ユーザエージェント (UA) ユーザごとに存在し、ユーザの学習履歴や追加した学習コンテンツの属性情報を管理する。

インタフェースエージェント (IA) ユーザインタフェースプログラムと他のエージェント間の通信を実現する。

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プログラム

学習コンテンツ管理者用インタフェース (管理者 IF) 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プログラム。

3. 学習コンテンツ管理機能

3.1 概要

学習コンテンツ管理機能は、提案システムにおける学習コンテンツの追加、削除、更新の機能を、学習コンテンツ管理の権限を与えられたユーザである学習コンテンツ管理者（以後、管理者と呼ぶ）に提供する。なお、ここでの更新とは、すでに提案システム上で提供される学習コンテンツの内容をもとに、新たな学習コンテンツを作成することである。

提案システムにおける学習コンテンツは、ノード間を動的に移動するエージェントにより提供される。このため、第1章に示す前操作、後操作が必要となる。

3.2 前操作: EA とそのカテゴリ担当ノードの特定

この操作は、削除、更新に先立ち行われる。削除、更新では、対象の学習コンテンツを提供する EA を指定して処理が依頼される。提案システムにおけるすべてのエージェントはエージェント ID で一意に識別される。しか

し管理者にとっては、エージェント ID で対象を指定するよりも、学習コンテンツの属性情報から対象を選択するほうが容易である。そこで提案システムでは、EA のエージェント ID と学習コンテンツのカテゴリ名、タイトル、初期難易度および概要を項目とするリストを管理し、EA の指定に利用する。このリストは、どのノードからでも利用可能とするため、管理者がこれまで追加した学習コンテンツの属性情報のリストを、各管理者ごとに UA に管理させる。

UA とそれを管理する GA の関係は、EA と CA の関係と同様で、ともに分散ハッシュテーブルを用いて各ノードに分散される。管理者は、提案システムを利用する際に GA によりユーザ認証を受ける。その後、対応する UA が管理者のノードに移動して、学習コンテンツの属性情報リストの参照が可能になる。

学習コンテンツ管理の処理は、EA を管理する CA が最終的に処理を行うことで実現する。すなわち、管理者からの依頼を CA へ届ける必要がある。しかしシステム全体を管理する存在はなく、各ノードはどのエージェントがどのノードに配置されているかはわからない。そこで学習コンテンツ管理において、P2P ネットワークを構築する分散ハッシュテーブルの機能を用いることで、対象カテゴリの担当ノードを検索しながら依頼を転送していく。あるカテゴリを担当するノード N を検索するアルゴリズムは以下のとおりである。

1. 管理者のノードを N に代入する。
2. 各ノードで共通のハッシュ関数に対象のカテゴリ名を与え、対象カテゴリに対応する二次元座標空間上の点を求める。この点を P に代入する。
3. もし、ノード N の管理領域が点 P を含むならば、終了する。
4. ノード N の隣接ノードのうち、点 P と管理する領域との距離が最も小さいノードを N に代入する。
5. 3 に戻る。

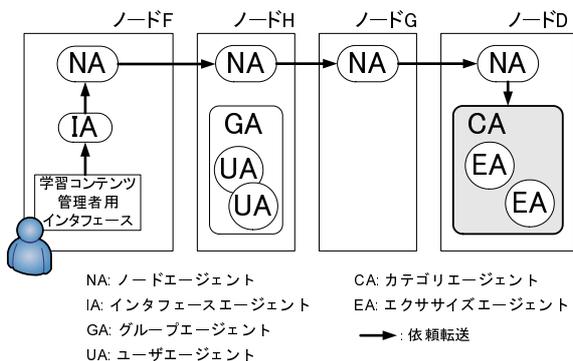


図 3: 依頼転送の様子

図 2 中のノード F からノード D 管理下の CA へ、依頼を転送する様子を図 3 に示す。提案システムにおいて、領域情報の管理とノード間の依頼の転送は、各ノードの

NA が担当する。図 3 では、ノード F 上の管理者 IF からの依頼が IA, NA を経由して、対象カテゴリの点に最も近いノード H の NA へ転送される。その後ノード H からノード G 上の NA を経由して、ノード D の NA に転送される様子が見える。対象ノードへの転送後、依頼は、NA からノード上の CA へ転送される。

3.3 前操作：EA の取得

この操作は、更新が必要となる。3.2 節に示すノード検索の手順によって更新の依頼を受け取った CA は、指定された EA に対し、管理者のノードに移動するよう依頼を送る。依頼を受け取った EA は、自身のコピーを作成し、コピーされた EA は管理者のノードに移動する。なおコピー作成元の EA は、コピー作成後もこれまで同様に学習の依頼に応じる。こうして移動した EA は、管理者 IF に自身の学習コンテンツの内容を提供する。

3.4 操作：EA の追加、更新、削除

この操作は、管理者のノードで呼び出される。追加の場合、管理者は管理者 IF を通じて新たに学習コンテンツの内容を作成する。これに対し更新の場合は、管理者は管理者 IF を通じて、移動してきた EA から受け取った内容をもとに学習コンテンツの内容を作成する。学習コンテンツの内容が準備された後は、追加、更新共に同様の処理を行う。まず、作成された内容を用いて EA を作成する。次に、3.5 節に示す手順により、作成した EA を提案システムに登録する。

削除の場合、管理者は管理者 IF を通じて対象の EA を指定し処理を依頼する。

3.5 後操作：EA の登録

この操作は、学習コンテンツの追加、削除、更新をシステムに反映させるため実行される。この操作では、これまでと同様、3.2 節に示すノードの検索を用いて達成される。対象カテゴリ担当ノードの検索後、追加、更新のため作成された EA は検索されたノードへ移動する。

提案システムにおいて、学習者へ提供される EA は CA に管理され、管理者が管理対象とする学習コンテンツの属性情報リストは UA に管理される。そのため、提案システムへの追加、削除、更新の反映は、次に示す処理で実現される。

CA へ追加登録・EA 削除 依頼が追加または更新の場合、CA は作成された EA を学習者への提供対象として追加登録する。また依頼が削除であれば、CA は指定された EA を管理対象から削除する。削除された EA はシステム上から消滅する。

属性情報リストへ項目追加・項目削除 依頼が追加または更新の場合、UA は作成された EA の属性情報項目を管理下の学習コンテンツの属性情報リストへ追加する。また依頼が削除であれば、UA は管理下の属性情報リストから対象の EA の項目を削除する。

4. 学習コンテンツ管理機能の実装

4.1 EA の構成

EA は学習コンテンツの内容を次の形式で表現する。

1. 問題や解説の表示レイアウトや答案採点用情報などからなる XML 文書
2. 問題で利用する画像データ

学習コンテンツを学習者に提供するため、EA は表示のための HTML 文書と採点のための正解などの値が必要である。これらデータは XML 文書から学習の度に生成することも可能である。しかしこの処理は毎回同じで、これらデータの大きさは十分小さい。そこで、EA の生成時に XML 文書から XSLT を用いて必要となる HTML 文書や処理値をあらかじめ生成し、EA に付与する。

4.2 学習コンテンツ管理者用インタフェース

管理者 IF は、学習コンテンツ管理機能で提供されるすべての機能をユーザに提供する。管理者 IF は、マルチプラットフォーム環境で動作し、操作方法がわかりやすい GUI アプリケーションとして提供することを狙い、Eclipse Rich Client Platform [5] により実装した。

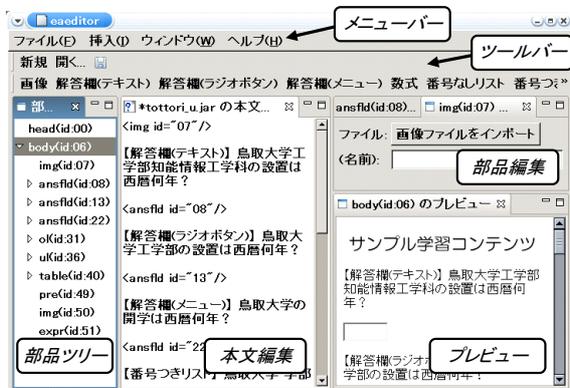


図 4: 学習コンテンツ作成の様子

管理者 IF 上で学習コンテンツを作成する様子を図 4 に示す。管理者 IF で作成される学習コンテンツは、問題の解答欄、表やリスト、画像、数式などの部品を利用できる。学習コンテンツの追加や削除などの機能は、提案システムへのログイン後にメニュー項目を選択することで実行できる。管理者 IF を通じて、削除対象の学習コンテンツを選択する様子を図 5 に示す。図 5 では、学習コンテンツの属性情報をもとに、削除対象を選択する様子がわかる。エージェントへの学習コンテンツ管理の依頼は、XML-RPC を通じて行う。

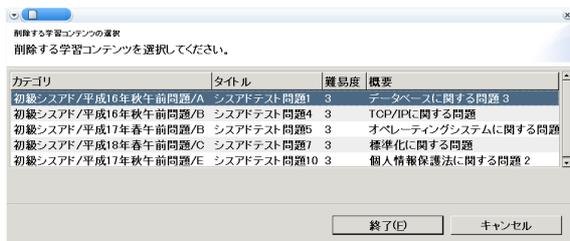


図 5: 学習コンテンツの属性情報リストから削除対象を選択する様子

5. おわりに

クライアントサーバモデルによる非同期 WBT システムでは、サーバへの負荷集中による応答性や安定性の低下が問題となる。この問題を解決するため、我々は P2P モデルによる分散型 e-Learning システムを提案している。本稿では、我々が提案する e-Learning システムにおける学習コンテンツ管理機能について述べた。

提案システムにおける学習コンテンツの追加、削除、更新にあたり、処理依頼ユーザのノード上での操作に加え、1) 対象エージェントとその担当ノードの特定およびエージェントの取得からなる前操作、2) システムへのエージェントの登録からなる後操作、を併せて行う。これにより、集中型 e-Learning システムと同様の学習コンテンツ管理機能を実現する。またこの機能を管理者 IF を通じて提供することで、ユーザは分散型の提案システムの構成を意識することなく、任意のノードから学習コンテンツの管理を行える。

ユーザの学習履歴はエージェントによって保持されている。現在提案システムでは、このようなデータはシステムの再起動によって初期状態に戻る。今後、提案システム実行中に保持されるデータをシステム再起動後に引き継ぐ手法の開発が必要である。

また、本稿で述べた学習コンテンツ管理機能を含む提案システムを大学などの機関で実際に運用し、提案システムの有効性を検証することが今後必要である。

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] 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).
- [5] McAffer, J. and Lemieux, J.-M.: *Eclipse Rich Client Platform: Designing, Coding, and Packaging Java™ Applications*, Addison-Wesley (2005).