

モバイルエージェントに基づく分散型 e-Learning システム Distributed e-Learning System Based on Mobile Agent

中谷 亮介[†]

本村 真一[†]

川村 尚生^{††}

菅原 一孔^{††}

Ryosuke NAKATANI[†]

Shinichi MOTOMURA[†]

Takao KAWAMURA^{††}

Kazunori SUGAHARA^{††}

[†] 鳥取大学 大学院 工学研究科

^{††} 鳥取大学 工学部

1 はじめに

近年インターネットの急速な普及に伴い、WBT (Web Based-Training) と呼ばれる、ウェブ技術を利用した e-Learning システムが種々提案されている。これら WBT は全て、クライアントサーバモデルで構築されている。クライアントサーバモデルとは全てのサービスをサーバが提供するものである。このモデルに基づくシステムは、構築しやすく、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、クライアント数が増加するに従いサーバにかかる負荷が増大し、応答時間の低下を招くといった欠点がある。また、サーバの故障によりシステム全体が使えなくなるという問題も抱えている。

クライアントサーバモデル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にサーバ機能が固定されず、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 (以下、ノードと呼ぶ) で構成されているのが特徴である。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可能である。また、いくつかのノードが故障してもシステム全体が使用不能にならない。

我々は、P2P モデルに基づく分散型 e-Learning システムを開発している [1][2]。このシステムでは、学習コンテンツをネットワーク上を移動できるモバイルエージェントとすることで、サービスをシステムに参加しているノードに分散させる。

2 分散型 e-Learning システムの概要

2.1 概要

本 e-Learning システムの問題は「数学/数学 II/微分と積分」、「数学/数学 II/図形と方程式」のようにカテゴリ単位で分類されている。ユーザは学習したい内容のカテゴリを指定することで、そのカテゴリに属する問題を学習する。ユーザの解答は後で述べるエクササイズエージェントにより採点される。ユーザは問題に関する解説も得ることができる。

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全ての問題を所持しており、他のノードからの要求に応じて問題を提供する責任を負う。ここで重要な点は、各ノードが担当しているカテゴリは、そのノード上でユーザが解いている問題のカテゴリとは無関係であるということである。図 1 は物理を担当しているノード

上のユーザが数学の問題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は未知であり、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。

本 e-Learning システムの初期起動時は全てのカテゴリは 1 つのノードに管理されている。新たなノードがシステムに参加すると初期ノードからカテゴリの一部を受け取り管理する。3 番目に参加するノードは既に参加している 2 つのノードのいずれかからカテゴリの一部を受け取って管理する。このように新たに参加するノードは既存のノードからカテゴリの一部を受け取り管理する。逆にシステムから離脱時には所持するカテゴリを他のノードへと受け渡す。カテゴリの授受には、そのカテゴリに属する問題の授受も含まれる。

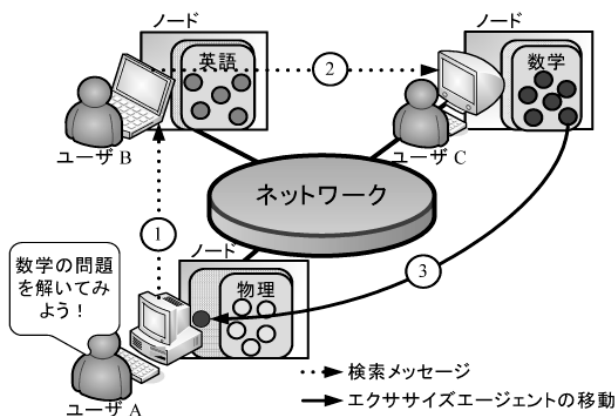


図 1: 分散型 e-Learning システムの概要

2.2 P2P ネットワーク

カテゴリの分散には分散ハッシュテーブルの一種である、Content-Addressable Network (CAN) [3] に基づく P2P ネットワークを用いる。CAN では (キー, 値) のペアを格納するための仮想座標空間がいくつかの領域に分割され、領域はノードに所持される。キー K_i はハッシュ関数により仮想座標空間にマッピングされる。この座標を求めることでキー K_i のペアとなる値 V_i を持つノードを検索できる。

CAN では仮想座標空間の次元数を規定していないが、本システムの P2P ネットワークでは 2 次元としている。カテゴリはカテゴリ名をキーとして仮想座標空間上にマッピングされ、この座標を含む領域を担当するノードに所持される。その様子を図 2 に示す。こ

の例では「物理」は座標(0.4, 0.4)にマッピングされ、その座標を含む領域を担当するノードAが所持し、「英語」及び「数学」はそれぞれ座標(0.6, 0.1)、座標(0.9, 0.8)にマッピングされそれぞれノードB及びノードCが所持している。ノードの参加・離脱により、各領域を担当するノードは変化する可能性があるが、ノードはカテゴリ名をキーとすることで、そのカテゴリが配置されたノードを検索できる。

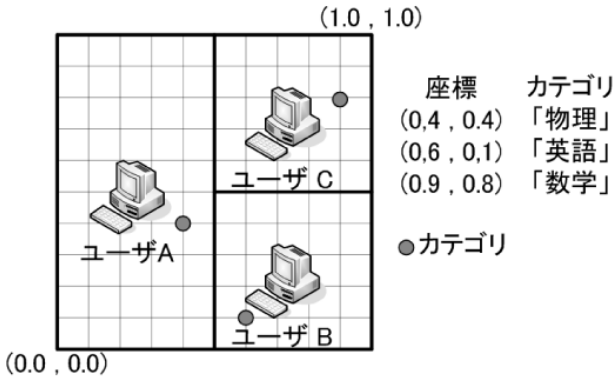


図 2: カテゴリのマッピング

2.3 モバイルエージェントとしての特徴

WBTを分散システムとして実現するには、問題を分散させるだけでは不十分で、そ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。従って、本システムでは問題を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とすることで実現する。

2.4 ユーザの分散管理

本システムではユーザが任意のコンピュータを利用した場合でも以前の学習履歴を参照できるように、学習記録をシステム内のいずれかのノードに持たせ、任意のノードから取得できるようにする。ある学習記録が誰のものかを把握するためにユーザ管理を行う必要がある。一般にユーザ管理を必要とするシステムでは、特定のサーバがその役割を担う。しかし、本システムは分散型のため、特定のサーバを必要としないユーザ管理を行う。また、ユーザ管理を行うノードに負荷が集中しないよう、複数のノードでユーザ管理を行う。

本システムではユーザを n 個のグループに分け、グループ単位でユーザ管理を行う。グループには「0」から「 $n-1$ 」までの番号が割り当てられる。グループはその番号をキーとして仮想座標空間上にマッピングされ、その座標を含む領域を担当するノードに所持される。これはカテゴリの分散と同様の手法で各ノードに配置される。

ユーザがどのグループに属するかは、そのユーザ名により決定される。ハッシュ関数によりユーザ名からハッシュ値を得る。ただし、このハッシュ関数はキーを仮想座標空間にマッピングするハッシュ関数とは別

である。得られたハッシュ値をグループの数(n)で割り剰余を計算する。剰余に対応する番号のグループにそのユーザは属する。

2.5 分散バックアップ

本システムではシステムに参加するノードがカテゴリを分け合い所持することで問題を分散させている。そのため、コンピュータが突然シャットダウンしてしまうなど、ノードに障害が起こった場合、そのノードが所持するカテゴリの問題は失われてしまう。

この問題はカテゴリのバックアップ、すなわちそのカテゴリに属する問題のバックアップをとることで解消できる。当然、カテゴリのバックアップはオリジナルのカテゴリとは別のノードが所持する必要がある。カテゴリのバックアップはオリジナルのカテゴリがマッピングされた座標の点対称となる座標にマッピングさせる。これによりカテゴリのバックアップとオリジナルのカテゴリが同じノードに所持されることはなくなる。図3はその様子を示したものである。ノードAの所持する「物理」のバックアップは点対称である座標(0.6, 0.6)にマッピングされ、この座標を含む領域を担当するノードCに配置される。

またノードに障害が生じた場合には、そのノードが所持するカテゴリのバックアップをもとに失われたカテゴリの復元が可能となる。

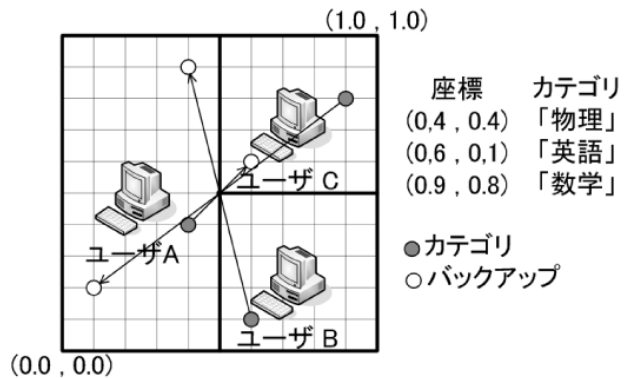


図 3: バックアップのマッピング

3 分散型 e-Learning システムの実装

分散型 e-Learning システムの実装に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][5]を用いた。本システムは次に示すエージェントおよびユーザ・アプリケーションインタフェースで構成される。なお、ユーザ・アプリケーションインタフェースについては次章以降で詳しく述べる。

ノードエージェント 領域情報の管理及び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 エクササイズエージェントの管理及び派遣を行う。

エクササイズエージェント ユーザへの問題提供及び採点、解答解説の提供を行う。

グループエージェント ユーザ管理を行う。

ユーザエージェント ユーザの学習記録を保持する。

ノードエージェントは各ノードに1つだけ存在し、そのノード上に留まる。一方、他のエージェントは複数存在しノード間を移動する。

ノードエージェントは自身の管理領域に含まれるカテゴリを管理する。また、隣り合うノードのアドレス及びそのノードの管理領域の情報も管理する。

カテゴリエージェントは問題及び、それに関する解答・解説を保有するエクササイズエージェントの管理を行う。ユーザが問題を要求すると、カテゴリエージェントはユーザの元へエクササイズエージェントのコピーを派遣する。エクササイズエージェントは要求者の元へと移動して、問題を提示し、ユーザの解答を採点、解答・解説を表示する。

ユーザエージェントはユーザごとに存在し、ユーザの学習記録を保持する。ユーザがシステムに参加した際には、そのユーザの元へと移動し、学習記録を受け渡す。グループエージェントはユーザの管理を行い、管理ユーザのユーザエージェントも管理する。

なお、これらエージェントは全て Maglog の提供するフィールドを介することで通信を行う。

4 ユーザ・アプリケーションインタフェースの設計

前章まではシステムの内側について述べた。システムの外側にはユーザ、アプリケーションなどがある。ユーザがシステムと対話したい。システムがアプリケーションと対話したい。または、ユーザ同士で対話したい、といった要求がある。これらを実現するためのユーザ・アプリケーションインタフェースを開発する。

ユーザ・アプリケーションインタフェースは以下の機能を有する。

- ユーザとエージェント間の通信機能
- エクササイズエージェントとアプリケーション間の通信機能
- ユーザ同士の通信機能

エクササイズエージェントは問題を持ち、そしてユーザの解答を採点、解答・解説を提供する機能を持っているが、直接ユーザと対話することはできない。そこで、ユーザ・アプリケーションインタフェースがユーザの要求をフィールドを介して、エクササイズエージェントに伝え、逆にエクササイズエージェントからの返答をユーザに伝える。

エクササイズエージェントはユーザの解答を採点する時、サポートアプリケーションを必要とする場合がある。例えば、エクササイズエージェントが持つ解答が「 $x^2 + x + y$ 」の時、「 $x + x^2 + y$ 」や「 $y + x + x^2$ 」なども正解である。しかし、可能性のある全ての解答を数式処理システムのサポートなしに正解とすること

は不可能である。そこで、ユーザ・アプリケーションインタフェースはエクササイズエージェントがサポートアプリケーションを利用して採点が行えるよう、両者間の通信を仲介する。

ユーザが問題の解答を理解するために、アドバイスを必要とする場合がある。既存の e-Learning システムでは BBS (Bulletin Board System) を利用しアドバイスの提供を行うが、本 e-Learning システムは分散型のため BBS は適していない。本システムではユーザ同士がチャットによる通信をすることでアドバイスの提供を行う。

5 ユーザ・アプリケーションインタフェースの実装

我々は、ユーザ・アプリケーションインタフェースを、ウェブブラウザ Firefox プラグイン [6] 及びインタフェースエージェントに分けて実装した。

5.1 Firefox プラグイン

ユーザは Firefox プラグインを利用することで問題を解く。図 4 はプラグインによって生成されたユーザインタフェースのメイン画面である。図 4 のカテゴリペインが示すよう、カテゴリはツリー構造で分けられる。ユーザはカテゴリを選択し、問題要求ボタンを押すことで、システム内の該当カテゴリを担当するノードからエクササイズエージェントが送られてくる。そして、問題ペインに問題が表示される。ユーザの解答は採点ボタンを押すことで、エクササイズエージェントへと伝えられ採点が行われる。また、解答・解説ボタンを押すことで解答・解説を得ることもできる。その様子を図 5 に示す。ユーザは問題に対するアドバイスを、チャットボタンを押すことで要求できる。

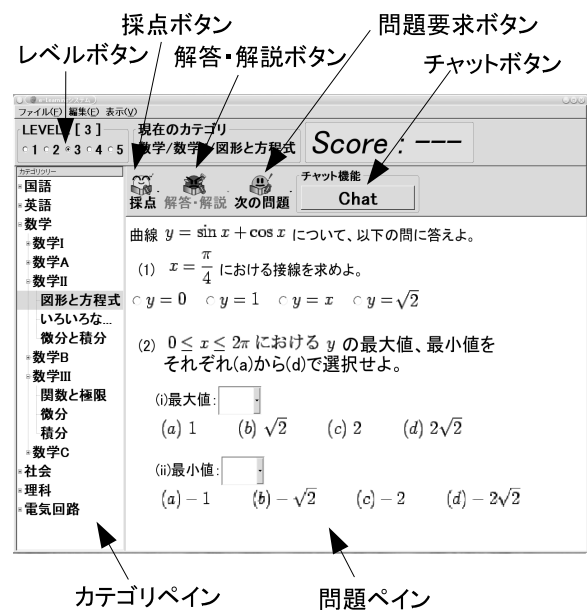


図 4: ユーザインタフェースのメイン画面

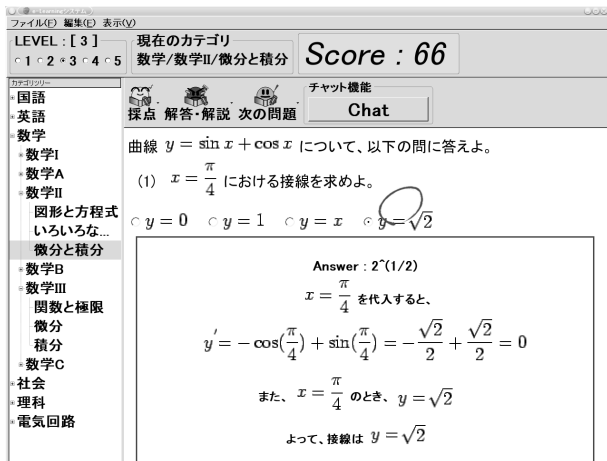


図 5: 採点, 解答・解説の表示

5.2 インタフェースエージェント

インタフェースエージェントは各エージェントと各プロセスとの接点となり, 次の通信機能を有する.

- XML-RPC[7] を利用したフィールドアクセスによるエージェントとユーザ間の通信
- Java クラスを用いたエクササイズエージェントとサポートアプリケーション間の通信
- チャットによるユーザ間の通信

Firefox プラグインはユーザからの要求をメッセージに変換し, フィールドに書き込む. そのメッセージはインタフェースエージェントが受け取り, ノードエージェントもしくはエクササイズエージェントのいずれか適切なエージェントに伝える.

Maglog 上の全てのエージェントは Java で実装されている. このため, エージェントは Java オブジェクトを呼び出すことが可能である. インタフェースエージェントはエクササイズエージェントのサポートアプリケーション呼び出し要求を受ける. インタフェースエージェントは Java オブジェクト介する形でサポートアプリケーションと通信を行う. なお, 図 6 は数式処理システムを利用した採点例である. この例では「 $\cos(x) - \sin(x)$ 」, 「 $-\sin(x) + \cos(x)$ 」どちらも正しく採点されていることがわかる.

6 おわりに

本研究では, モバイルエージェントに基づく分散型 e-Learning システムの開発を行った. 問題をシステム利用者が分け合い管理し, 他のユーザの要求に応じて学習コンテンツを提供すること, ユーザの分散管理を行うことで負荷を分散させることができた. また, 問題のバックアップもとることで耐障害性を高めることができた.

そして, ユーザ・アプリケーションインタフェースを実装し, システム, ユーザ, アプリケーション間の対話を可能とした.

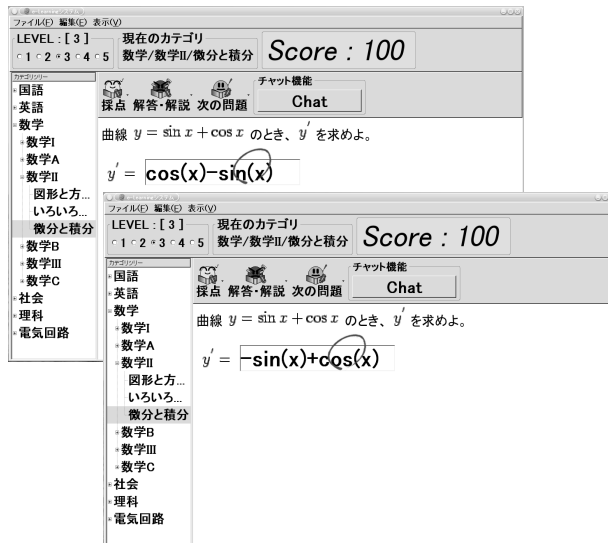


図 6: 数式処理システムを用いた採点例

なお, 本システムは本学科の演習室において学生たちの自習用として利用されている.

7 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. et al.: A Scalable Content-Addressable Network, *Proceedings of ACM SIGCOMM*, pp. 161-172 (2001). San Diego, CA.
- [4] Motomura, S., Kawamura, T. and Sugahara, K.: Maglog: A Mobile Agent Framework For Distributed Models, *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems*, pp. 414-420 (2005). Phoenix, Arizona, USA.
- [5] Motomura, S., Kawamura, T. and Sugahara, K.: A Logic-Based Mobile Agent Framework for WEB Applications, *Proceedings of the 2nd International Conference on Web Information Systems and Technologies*, pp. 121-126 (2006). Setubal, Portugal.
- [6] Corporation, M.: Rediscover the Web, <http://www.mozilla.com/firefox/>.
- [7] Winer, D.: XML-RPC Specification, <http://xmlrpc.com/spec> (1998).