

分散型 e-Learning システムへの相互援助のための ユーザ間通信機能の実装について

Implementation of Interactive Communication Function for Mutual Assists on the Distributed e-Learning System

本村 真一[†]中谷 亮介[†]川村 尚生[‡]菅原 一孔[‡]

Shinichi MOTOMURA

Ryosuke NAKATANI

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る。近年、ネットワークの急速な普及により WBT(Web Based-Training) 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。

これら WBT はすべてクライアント/サーバモデルで構築されている。クライアント/サーバモデルとはすべ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う欠点がある。また、サーバ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, 3]。このシステムでは、学習コンテンツをネットワーク上で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。

本稿では、分散型 e-Learning システムの相互援助機能、すなわち、ユーザ同士が不明点について対話できる機能について報告する。

既存の e-Learning システムにも相互援助機能を持つものが存在している [4, 5]。これらのシステムでは、サーバに設置された、どのユーザも読み書きできる掲示板を通じて質問に答えあう。

しかし、我々は掲示板方式を採用しない。掲示板のために集中型サーバを導入することは望ましくないし、かといって掲示板を分散システムとして効率よく実現することは困難と考えられるからである。さらに、もしも効率のよい実現が可能であったとしても、掲示板方式は、ユーザが自分に答えることのできる質問がないかと掲示板を能動的に見ることが前提となっており、質問に答える能力を持つユーザが掲示板を見ていなければ、いつまでも回答が得られないという欠点を持つ。

そこで、本稿では、質問するユーザとそれに答えるユーザが直接通信する方式を提案する。質問に答えるユーザ候補はエージェントによって選び出される。

本稿は以下のように構成されている。第2章で我々が開発している分散型 e-Learning システムの概要を説明する。第3章では分散型 e-Learning システムへのユーザ間通信機能の実装について論じ、第4章では実装したユーザ間通信機能の実行例を示す。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 分散型 e-Learning システムの概要

本 e-Learning システムの学習コンテンツは「数学/数学 I/2 次関数/2 次方程式」、「数学/数学 A/数と式/因数分解」のようにカテゴリ単位で分類されている。また、学習コンテンツは問題、解答、解説で構成されている。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。ユーザの答えは後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を得ることもできる。

学習コンテンツは、システムを同時に使用しているユーザのコンピュータ、すなわちノードに、カテゴリ単位で分散され管理される。初期状態ではすべてのカテゴリは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って管理する。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に参加しているノードからカテゴリの一部を受け取って管理する。逆にシステムから離脱する場合には、管理するカテゴリを他のノードに受け渡す。

学習コンテンツの要求メッセージは、その学習コンテンツを管理するノードまで運ばれ、要求されたカテゴリに属する学習コンテンツは要求者の元へと移動する。図1はその様子を示したものである。

2.2 分散型 e-Learning システムの構成要素

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[6]を用いた。ただし、ユーザインタフェースは Squeak[7]上に実装した。

各ノードには下記に示すエージェント及びユーザインタフェースが存在する。

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

[†]鳥取大学大学院工学研究科

[‡]鳥取大学工学部

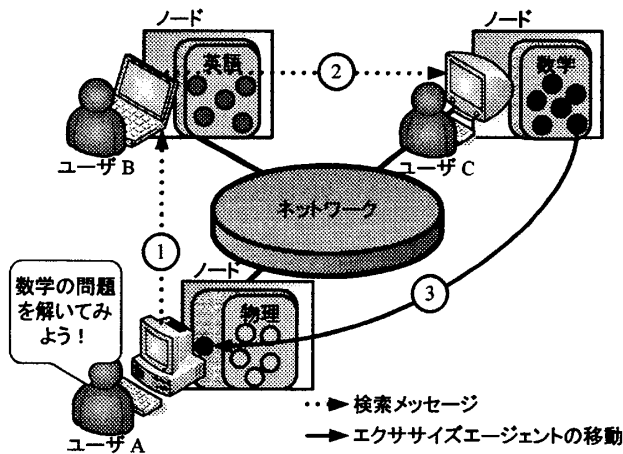


図1: 学習コンテンツの移動

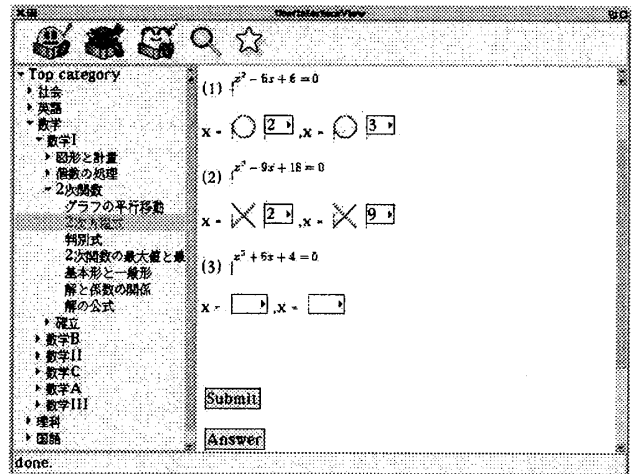


図2: ユーザインタフェース

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

インタフェースエージェント 各エージェントとユーザインタフェースとの仲介を行う

ユーザインタフェース ユーザが e-Learning システムを利用するためのインタフェース

ノードエージェント, インタフェースエージェント及びユーザインタフェースは各ノードに1つだけ存在し, そのノード上に留まる. 一方, カテゴリエージェント及びエクササイズ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の学習コンテンツを管理する. また, 隣り合うノードの隣接ノード情報を管理する. 隣接ノード情報とは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である.

カテゴリエージェントは学習コンテンツを保有するエクササイズエージェントの管理を行う. ユーザが学習コンテンツを要求すると, 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派遣を行う.

エクササイズエージェントは学習コンテンツの要求者の元へと移動して, 問題を提示し, ユーザの答案を採点し, 解説を表示する.

インタフェースエージェントはユーザインタフェースと各エージェント間のメッセージの仲介を行う.

ユーザインタフェースはエクササイズエージェントからインタフェースエージェントを介して問題を受け取り表示する. ユーザの答案はインタフェースエージェントを介してエクササイズエージェントへと送られ, エクササイズエージェントにより採点される. その結果は図2のように表示される.

3. ユーザ間通信機能

3.1 ユーザ間通信機能の設計

あるユーザが質問した場合 (以後, 質問ユーザと呼ぶ), 現在システムを使用しているユーザの中から, その質問に答えられそうなユーザ (以後, 回答可能ユーザ) を選んで回答を依頼する. ここで, 本システムは分散型であるため, どのようなユーザが現在参加しているかを把握していない. したがって, 適切な回答可能ユーザを選ぶためには,

1. どのような基準で, 回答可能だと判断するか,
2. そのユーザはどのようにして探すか,

という2つの問いに答えなければならない.

ここでは, 1 番目の問いについては,

1. 要求のなされた学習コンテンツを解いている,
2. その得点が 80 点以上である,

という2つの条件を満たすユーザを回答可能と推定することにした. 2 番目の問いについては, 図3に示す, 回答可能ユーザ検索アルゴリズムを開発した.

アルゴリズムの実行中, 待ち行列 Q には調査すべきノードのアドレスが格納され, 集合 S には同じノードを調べないように, 既に調べたノードのアドレスを保持する.

図4に示す例を用いて図3のアルゴリズムを説明する. 質問ユーザのノードを図4のノード A (以下 A, その他のノードも同様) とする. A の隣接ノードは B, D, E, G の4つである. まず, A のアドレスを S に入れ, B, D, E, G のアドレスを Q に格納する.

$$S = \{A\}$$

$$Q = [B, D, E, G]$$

次に, Q 中の先頭ノードである B を取り出して, B のユーザが回答可能かどうかを調べる. いない場合, B の隣接ノードを Q に追加するが, A は集合 S に入ってい

```

var
  S : 既に回ったノードのアドレスの集合;
  Q : 調査すべきノードのアドレスの待ち行列;
  N : 隣接ノードのアドレスの集合;
  x : 隣接ノードのアドレス;
  y : 待ち行列Sから取り出したノードのアドレス;

begin
  S := 質問ノードのアドレス;
  質問ノードのノードエージェントからNを取得;
  Nの各要素xについて
  begin
    ENQUEUE(Q, x);
  end
  while (not EMPTY(Q)) do
  begin
    ENQUEUE(Q, y);
    yのノードへ移動;
    if ユーザが回答ユーザの条件を満たしている
      and ユーザが回答を了承 then
      return(ユーザ);
    ノードyのノードエージェントからNを取得;
    Nの各要素について
    begin
      if (not STORED(Q, x) and (not IN(S, x) then
        ENQUEUE(Q, x);
      end
    end
    S := UNION(S, y);
  end
  return(探索失敗);
end

ENQUEUE(Q, x) : 待ち行列Qの最後に要素xを格納する関数
DEQUEUE(Q, y) : 待ち行列Qの先頭から要素yを取り出す関数
STORED(Q, x) : 待ち行列Qに要素xが存在する時に真を返す関数
EMPTY(Q) : 待ち行列Qが空の時に真を返す関数
IN(S, x) : 集合Sに要素xが含まれている時に真を返す関数
UNION(S, T) : 集合SとTの共通集合を返す関数
  
```

K	J	F	
I	D	C	
H	G	A	B
L	E		

図 4: ノードの配置例

回答可能ユーザが回答を了承した場合、質問ユーザと回答ユーザはメッセージをリアルタイムでやりとりすることで対話する。このチャットによるユーザ間通信機能をユーザインタフェース上に実装した。

図 3: 回答可能ユーザ検索アルゴリズム

るので追加しない。また E は既に Q に格納されているので重ねては追加せず、C のみを追加する。そして調べ終わった B を集合 S に追加する。

$$S = \{A, B\}$$

$$Q = [D, E, G, C]$$

次に、Q 中の先頭ノードである D を取り出して、D 上に回答可能ユーザがいるかどうかを調べる。いない場合、F, I を Q に加え、D を S に追加する。

$$Q = [E, G, C, F, I]$$

$$S = \{A, B, D\}$$

この操作が、回答可能ユーザが見つかるか、Q が空になるまで繰り返される。

3.2 ユーザ間通信機能の実装

ユーザ間通信機能実現のため「メッセージエージェント」を実装した。メッセージエージェントは回答可能ユーザを捜し出し、ユーザ同士の仲介を行う。

メッセージエージェントはユーザが質問するとエクササイズエージェントにより生成され、前述のアルゴリズムに基づき各ノードを回って回答可能ユーザの検索を行う。回答可能ユーザか否かの判断はユーザの学習履歴を保持するインタフェースエージェントによって行われる。回答可能ユーザを検索の様子を図 5 に示す。

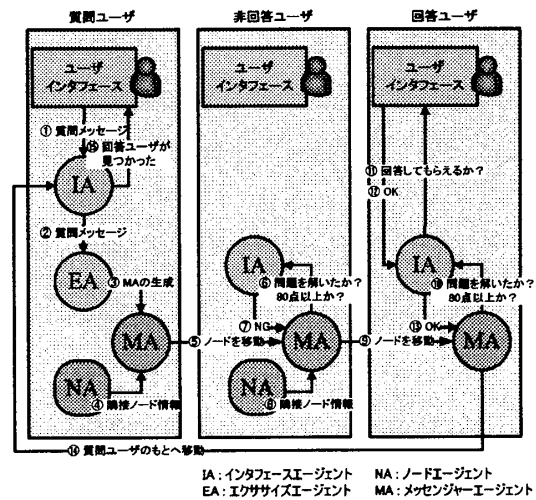


図 5: 回答可能ユーザの検索

4. ユーザ間通信機能の実行例

ユーザ間通信機能の実行例を示す。

1. メッセージの書き込み

質問ユーザは質問する際、図 6 に示すような質問メッセージを送る。このメッセージはメッセージエージェントにより運ばれ、回答可能ユーザに表示される。

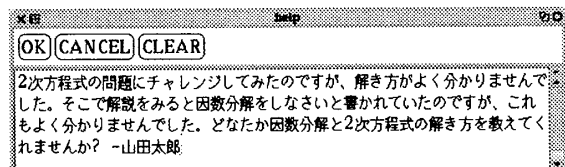


図 6: 質問メッセージの書き込み

2. メッセージの受け取り

回答可能ユーザへは図7に示すように、質問メッセージの冒頭が示される。回答を引き受ける場合は「YES ボタン」を押すことでチャットウィンドウが開かれ、ユーザ間通信が可能となる。

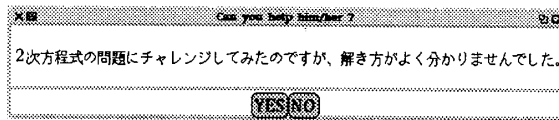


図7: 質問メッセージの受け取り

3. チャットによるユーザ間通信

図8は質問ユーザと回答可能ユーザとの間で行われているチャットの様子を示している。このウィンドウは両方のユーザに表示されている。

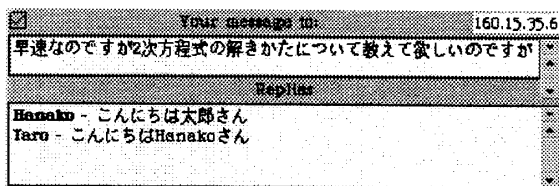


図8: ユーザ間のチャット

5. おわりに

我々が開発しているモバイルエージェント技術に基づく分散型 e-Learning システムに対して、相互援助のためのユーザ間通信機能を開発した。この機能は、既存の e-Learning システムに見られる掲示板方式ではなく、質問ユーザと回答可能ユーザが直接通信する方式をとっている。回答可能ユーザはエージェントが動的にシステム中を探索する。

ユーザ間通信機能の実装によりユーザ同士のコミュニケーションが可能となり、一人で学習を進める場合に比べ、より学習効率が高上がることが期待される。

現在のユーザ間通信機能では、質問された問題をすでに解いているユーザのみを探索対象に絞っているため、対象となるユーザ数が限られてしまう。今後、同一の問題を解いていなくても同じカテゴリに属する問題の成績がよいユーザを対象に加えるなど、探索対象のユーザを拡大することが必要であると考えている。また、回答に満足できない場合に、同じ質問を他のユーザに対して再度行う機能を追加することも今後の課題である。

参考文献

[1] Kawamura, T., Nakatani, R. and Sugahara, K.: P2P e-Learning System and Its Squeak-based User Interface, *Proceedings of the International Conference on Creating, Connecting and Collaborating through Computing*, pp. 59-65 (2005). Kyoto, Japan.

- [2] 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).
- [3] Kawamura, T., Kinoshita, S. and Sugahara, K.: A Mobile Agent-Based P2P e-Learning System, *Proceedings of the IASTED International Conference Parallel and Distributed Computing and Systems* (Gonzalez, T.(ed.)), pp. 873-877 (2004). MIT, Cambridge, USA.
- [4] NTT ソフトウェア: CALtutor.
<http://www.ntts.co.jp/ps/caltutor/index.html>.
- [5] イー・ステージ株式会社: Web 学習システム iCASS net.
http://www.e-stage.co.jp/products/icass_net.htm.
- [6] Kawamura, T., Kinoshita, S. and Sugahara, K.: Implementation of a Mobile Agent Framework on Java Environment, *Proceedings of the IASTED International Conference Parallel and Distributed Computing and Systems* (Gonzalez, T.(ed.)), pp. 589-593 (2004). MIT, Cambridge, USA.
- [7] Korienek, G., Wrensch, T., Doug Dechowans 菅原一孔, 鈴木元, 阿部和広 (訳): Squeak プログラミング入門 -オブジェクトランドへの招待-, (株) エスアイビー・アクセス (2004).