DISTRIBUTED E-LEARNING SYSTEM USING P2P TECHNOLOGY

Shinichi Motomura, Ryosuke Nakatani, Takao Kawamura, Kazunori Sugahara
Tottori University
4–101, Koyama-Minami, Tottori 680–8552, JAPAN

Keywords: P2P, e-Learning, Mobile Agent, Distributed hash table.

Abstract: In this paper, we present a novel framework for asynchronous Web-based training. The proposed system has two distinguishing features. Firstly, it is based on P2P architecture for scalability and robustness. Secondly, all contents in the system are not only data but also agents so that they can mark user's answers, can tell the correct answers, and can show some extra information without human instruction. We also present a prototype implementation of the proposed system on Maglog. Maglog is a Prolog-based framework for building mobile multi-agent systems we have developed. Our system has methods for communications between learners and for recordings of their score without server computers. The user interface program of the proposed system is built on Squeak. Performance simulations demonstrate the effectiveness of the proposed system.

1 INTRODUCTION

The term e-learning covers a wide set of applications and processes, such as Web-based training (hereafter we abbreviate as WBT), computer-based training, virtual classrooms, and digital collaboration. We are concerned with asynchronous WBT that allows the learner to complete the WBT on his own time and schedule, without live interaction with the instructor.

Although a large number of studies have been made on asynchronous WBT(Helic et al., 2003; Homma and Aoki, 2003), all of them are based on the client/server model. The client/server systems generally lack scalability and robustness. In the recent years, P2P research has grown exponentially. In this paper, we present a novel system for asynchronous WBT.

We had proposed the system(Kawamura and Sugahara, 2005) has two distinguishing features. Firstly, it was based on P2P architecture and every user's computer played the role of a client and a server. Namely, while a user used the proposed e-Learning system, his/her computer (hereafter we refer to such a computer as a node) was a part of the system. It received some number of contents from another node when it joined the system and had responsibility to send appropriate contents to requesting nodes. Secondly, each content in the system was not only data but also an agent so that it could mark user's answers, tell the correct answers, and show some extra information without human instruction.

In this paper, methods for communications between learners and for recordings of their score in the proposed e-learning system are proposed. The methods are implemented in our proposed e-Learning system. Many of e-learning systems including our proposed e-Learning system are designed for learner's self-educations. In such systems, learners want to ask for someone's advices when they cannot solve their exercises. To satisfy such demand many of ordinary WBT systems have BBS (Bulletin Board System) functions. However BBS functions are effective in concentrated WBT systems, they require WBT servers and as a result they are not well suited for our proposed e-learning system. The method proposed in this paper is for completely distributed systems and does not require server computers. When one learner needs someone's advices, the system tries to find good scored person of the exercises. When time of the good scored person permits, the learner can start communication with him/her and can get some good advices. The method for recording learners scores in our proposed e-learning system is also proposed in this paper. In the ordinal system, this function is also implemented with server computers. We tried to realize the function by using P2P architectures and mobile agents

Motomura S., Nakatani R., Kawamura T. and Sugahara K. (2006).
DISTRIBUTED E-LEARNING SYSTEM USING P2P TECHNOLOGY.
In *Proceedings of WEBIST 2006 - Second International Conference on Web Information Systems and Technologies - Internet Technology / Web Interface and Applications*, pages 250-255
DOI: 10.5220/0001251202500255

technologies in order not to reduce the advantages of our proposed e-learning system.

In this paper,we present a prototype of the proposed system on Maglog (Motomura et al., 2006) that is a Prolog-based framework for building mobile agent systems we have developed.

2 PROPOSED SYSTEM

2.1 Overview

As mentioned in the previous section, we focus on asynchronous WBT, that is to say, a user can connect to the proposed e-Learning system anytime and anywhere he/she wants. Once connection is established, the user can obtain exercises one after another through specifying categories of the required exercises. User's answers for each exercise are marked as correct or incorrect right away. Extra information may be provided for each answer, which can be viewed when the correct answer is shown.

While a user uses the proposed e-Learning system, his/her computer is a part of the system. Namely, it receives some number of categories and exercises in them from another node when it joins the system and has responsibility to send appropriate exercises to requesting nodes.

The important point to note is that the categories a node has are independent of the categories in which the node's user are interested as shown in Fig.1. Figure 1 illustrates that user A's request is forwarded at first to the neighbor node, next forwarded to the node which has the requested category.

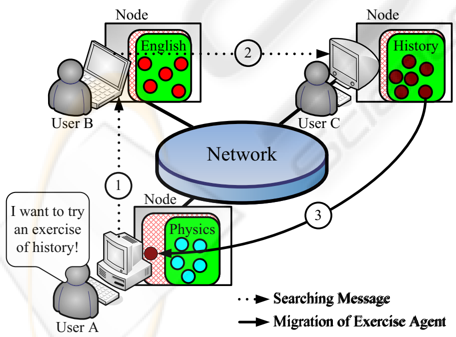

Figure 1: Proposed e-Learning system.

2.2 P2P Aspect

When the proposed system begins, one initial node has all categories in the system. When another node joins the system, it is received some number of categories from the initial node. The categories are distributed among all nodes in the system according as nodes join the system or leave the system.

We would like to emphasize that in existing P2P-based file sharing systems such as Napster(Napster, 1999) and Gnutella(Gnutella, 2000) each shared file is owned by a particular node. Accordingly, files are originally distributed among all nodes. On the other hand, the categories in the proposed system are originally concentrated. Consequently, when a new node joins the system, not only location information of a category but the category itself must be handed to the new node. Considering that, the P2P network of the proposed system can be constructed as a CAN(Ratnasamy et al., 2001).

A CAN has a virtual coordinate space that is used to store $(key, value)$ pairs. To store a pair (K_1, V_1), key K_1 is deterministically mapped onto a point P in the coordinate space using a uniform hash function. The corresponding $(key, value)$ pair is then stored at the node that owns the zone within which the point P lies. In the proposed system, we let each category be a key and let a set of exercises belonging to the category be the corresponding value.

Our P2P network is constructed with 2-dimensional coordinate space [0,1] $\times$ [0,1] to store exercise categories, as shown in Fig.2. The figure shows the situation that Node C is joining the system where Node A and Node B have already joined. Before Node C joins, Node A and Node B shared the whole coordinate space half and half. At that moment, Node A managed "Math/Geometry" category and Node B managed "Grammar" and "History" categories, respectively. When Node C joins the system, it is mapped on a certain coordinate space according to a random number and takes on corresponding categories from another node. For example, in the case of Fig.2, Node C takes on the "History" category from Node B and exercises move to Node C.

2.3 Mobile Agent Aspect

Generally, in addition to service to show an exercise, a WBT server provides services to mark the user's answers, tell the correct answers, and show some extra information about the exercise. Therefore, for the proposed system which can be considered a distributed WBT system, it is not enough that only exercises are distributed among all nodes. Functions to provide the above services also must be distributed among all nodes. We adopt mobile agent technology to achieve this goal. Namely, an exercise is not only data but also an agent so that it can mark user's answers, tell the correct answers, and show some extra information about the exercise.

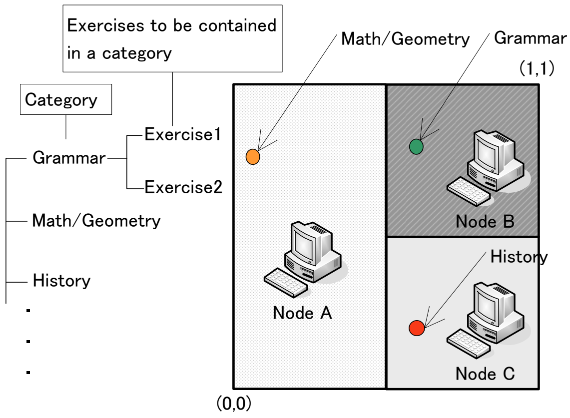

Figure 2: Peer to peer network.

In addition, mobile agent technology is applied to realize the migration of categories, that is, each category is also an agent in the proposed system.

2.4 Interactive Communication Function

A questioner is a learner who wants to ask for someone's advice. An answerer is a learner who has a good score and gives good advice. When a questioner wants to get some advice from an answerer interactively using online chat, however, a questioner cannot know what kind of learner uses the system. Therefore the system provides the function to find an answerer. In the system, answerers are defined that they already solved the request's exercise and its score was higher than 80. The algorithm to find an answerer is shown in Fig. 3. The queue Q contains addresses of nodes to inquire and the set S contains addresses of inquired node. The algorithm is explained using Fig. 4. It is assumed that the node of a questioner is A. Adjacent nodes of A are B, D, E, and G. First, the address of A is appended to S, and the addresses of B, D, E, and G are appended to Q.

$S = \{A\}$

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

Next, B which is the top of Q is retrieved and the learner on B is inquired whether he is an answerer. If he is not an answerer, adjacent nodes of B will be appended to Q, but A which is already contained in Q is not appended. And only C will be appended to Q because E already is contained in Q. After inquiring, B will be appended to S.

$S = \{A, B\}$

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

Next, D which is the top of Q is retrieved and the learner on D is inquired whether he is an answerer. If he is not an answerer, F and I will be appended to Q and D will be appended to S.

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

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

The above-mentioned operations are repeated until an answerer is found, or Q becomes empty.

```
var
    S : Set of addresses of inquired nodes;
    Q : Queue of addresses of nodes to inquire;
    N : Set of addresses of the adjacent nodes;
    x : Address of the adjacent node;
    y : Retrieved node address from S;
begin
    S := node address of questioner;
    To get N from node of questioner;
    To execute the following each x of N
    begin
        ENQUEUE(Q,x);
    end
    while (not EMPTY(Q)) do
    begin
        ENQUEUE(Q,y);
        To migrate the node assigned;
        if the learner on the node is an answerer
        and the answerer agrees to teach then
          return(y);
          To get N from y;
          To execute the following each address of N
          begin
            if (not STORED(Q,x)) and (not IN(S,x)) then
              ENQUEUE(Q,x);
          end
          S := UNION(S,y);
    end
    return(failure);
end

ENQUEUE(Q,x) : Appends x to the end of Q
DEQUEUE(Q,y) : Retrieves y from Q
STORED(Q,x) : Returns true if Q contains x
EMPTY(Q) : Returns true if Q contains no element
IN(S,x) : Returns true if S contains x
UNION(S,T) : Returns the union of S and T
```

Figure 3: The algorithm to find an answerer.

3 DESIGN AND IMPLEMENTATION

We have implemented a prototype of the proposed system on Maglog that is a Prolog-based framework for building mobile multi-agent systems we have developed.

Node Agent There is one node agent on each node. It manages the zone information of a CAN and forwards messages to the category agents in the node.

Category Agent Each category agent stands for a unit of a particular subject. It manages exercise

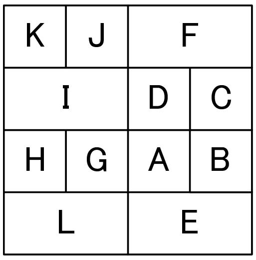

Figure 4: The example of arrangement of nodes.

agents in itself and sends them to the requesting node.

Exercise Agent Each exercise agent has a question and functions to mark user's answers, tell the correct answers, and show some extra information about the exercise. These data are formatted in HTML.

Messenger Agent A messenger agent is created when a questioner want to find an answerer. It goes round nodes to find the answerer.

Log Agent Each log agent contains learners' scores of the corresponding exercises. The number of them is identical of the number of learners. Teachers can get score of any learner by communicating with the corresponding log agent.

Interface Agent There is one interface agent on each node. It is an interface between the user interface program and other agents.

Agents communicate with other agents through 'field's provided by Maglog framework. A field is kind of a preemptive queue. Roughly speaking, the above-mentioned six kinds of agents execute a message dispatch loop. Each message to an agent is queued into the field owned by the agent. The user interface program also communicates with the interface agent through a field via XML-RPC(Winer, 1998).

As mentioned above, the user interface program of the proposed system has been developed through extending Scamper which is a simple web browser runs in Squeak(Ingalls et al., 1997).

Figures 5, 6, and 7 are screen-shots of the user interface program. By clicking the left button of a mouse on the category, a user can select it. After selection of the category, a user can obtain an exercise belonging to the category by clicking the left button of a mouse on one of buttons in the button pane. After a while an appropriate exercise agent comes from some node and the user can try the question. The user can require to mark his/her answer anytime by clicking the submit button. Figure 5 shows an example result of marking. Figure 6 shows the correct answers and extra information about the exercise that are shown by clicking the answer button. Figure 7 shows a questioner and an answerer communicate using online chat.

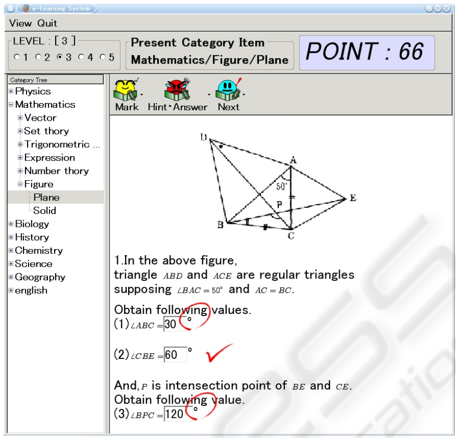

Figure 5: User's answers are marked as correct or incorrect by clicking the submit button.

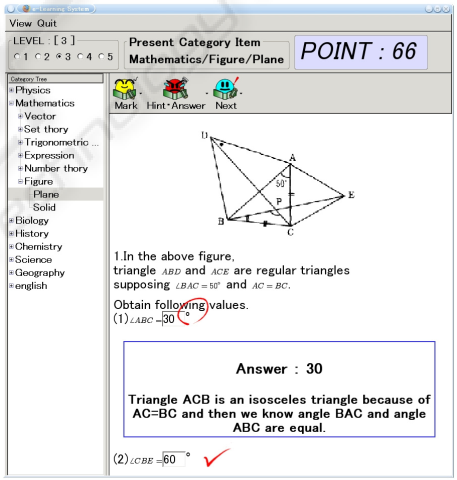

Figure 6: Correct answers and extra information are shown by clicking the answer button on the screen shown as Figure 5.

4 PERFORMANCE SIMULATION

This section presents performance simulations obtained from a prototype implementation of the proposed system described in the previous section.

The experimental environment consists of 8 PCs with Intel Pentium4 2.4GHz processor and 512MB of RAM, and all the PCs are running on GNU/Linux (kernel version is 2.2.26) operating system.

We measured the searching latency in the experimental environment under the conditions shown in

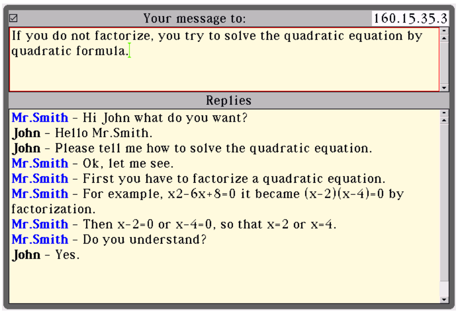

Figure 7: A questioner and an answerer communicate using online chat.

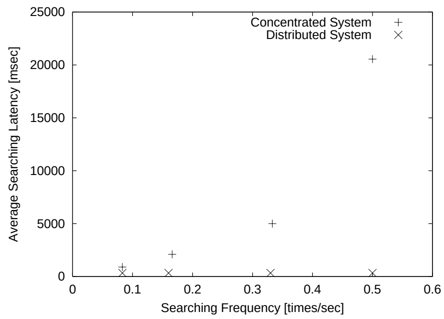

Figure 8: Comparison of concentrated and distributed systems in searching latency.

Table 1. All nodes send searching requests at the same time and exercises to be searched are selected randomly. We compared distributed and concentrated systems where these terms are defined as follows:

Distributed System Each node has one category.

Concentrated System One node has all categories and the rest nodes have no category.

It must be noted that the distributed system represents the proposed system in which all categories have ideal distribution. The concentrated system is equivalent to an ordinary WBT system.

Table 1: Experimental Conditions.

Number of Nodes	8
Number of Categories	8
Number of Exercises/Category	50
Searching Frequency [times/sec]	$\frac{1}{12}, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}$

Simulations are carried out with the time interval of 600 seconds. Each Simulation is repeated 10 times and the average of those is reported in Figure 8. Naturally, the higher searching frequency is, the larger searching latency is. Figure 8 shows that searching latency grows rapid in the concentrated system, while it grows slowly in the distributed system. In other words, the result suggests that the proposed e-Learning system has higher scalability than ordinary concentrated WBT systems have.

5 RELATED WORKS

A great deal of effort has been made on agent-based systems(Wong et al., 1997; Lange and Oshima, 1998; Tarau, 1999; Satoh, 2000). However, these technologies provide support for agent collaboration and communication but lack support for P2P technology. Therefore, there are few agent-based P2P applications. PeerDB(Ng et al., 2003) is one of them, however agent technology is only used to assist query processing while in the proposed e-Learning system it is used not only for interactiveness but also for migration of the functionality of the system.

Edutella is P2P network for exchanging information about learning objects(Nejdl et al., 2002). Edutella is based on RDF(Resource Description Framework), which is a framework for representing information in the Web. Consequently, Edutella does not intend to receive user's response. In contrast, that is one of main goal of the proposed system and it is achieved through agent technology.

6 CONCLUSION

Since existing asynchronous WBT systems are based on the client/server model, they have problems of scalability and robustness. The proposed e-Learning system solves these problems in decentralized manner through both P2P technology and mobile agent technology. The user interface program of the proposed system is built on Squeak so that it obtains much interactiveness and flexibility. Performance simulations suggest that the proposed e-Learning system has higher scalability than ordinary concentrated WBT systems have. More work needs to be done in the evaluation of the robustness of the proposed system. For example, whole services will be stopped if any one computer in the system is crushed, because there is no backup mechanism in our proposed e-Learning system. At that time, exercise agents located in the crushed computer can not be restored. Considering these points, the backup mechanism, especially for the distributed systems, is considered as future problems.

REFERENCES

Gnutella (2000). http://welcome.to/gnutella/.

Helic, D., Krottmaier, H., Maurer, H., and Scerbakov, N. (2003). Implementing project-based learning in wbt systems. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pages 2189–2196.

Homma, H. and Aoki, Y. (2003). Creation of wbt server on digital signal processing. In *Proceedings of 4th International Conference on Information Technology Based Higher Education and Training*. Marrakech, Morocco.

Ingalls, D., Kaehler, T., Maloney, J., Wallace, S., and Kay, A. (1997). Back to the future: The story of squeak, a practical smalltalk written in itself. In *Proceedings of ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*, pages 318–326.

Kawamura, T. and Sugahara, K. (2005). A mobile agent-based p2p e-learning system. *IPSJ Journal*, 46(1):222–225.

Lange, D. B. and Oshima, M. (1998). *Programming and Deploying Java Mobile Agents with Aglets*. Addison-Wesley.

Motomura, S., Kawamura, T., and Sugahara, K. (2006). Logic-based mobile agent framework with concept of field. *IPSJ Journal*, 47(4). will be published.

Napster (1999). http://www.napster.com/.

Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., and Risch, T. (2002). Edutella: A p2p networking infrastructure based on rdf. In *Proceedings of the Eleventh International Conference on World Wide Web*, pages 604–615. ACM Press.

Ng, W. S., Ooi, B. C., Tan, K.-L., and Zhou, A. (2003). Peerdb: A p2p-based system for distributed data sharing. In Dayal, U., Ramamritham, K., and Vijayaraman, T. M., editors, *Proceedings of the 19th International Conference on Data Engineering*, pages 633–644. IEEE Computer Society.

Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S. (2001). A scalable content-addressable network. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172. ACM Press.

Satoh, I. (2000). Mobilespaces: A framework for building adaptive distributed applications using a hierarchical mobile agent system. In *Proceedings of IEEE International Conference on Distributed Computing Systems*, pages 161–168. IEEE Press.

Tarau, P. (1999). Inference and computation mobility with jinni. In Apt, K., Marek, V., and Truszczynski, M., editors, *The Logic Programming Paradigm: a 25 Year Perspective*, pages 33–48. Springer.

Winer, D. (1998). Xml-rpc specification. http://xmlrcp.com/spec.

Wong, D., Paciorek, N., Walsh, T., and Dicelie, J. (1997). Concordia: An infrastructure for collaborating mobile agents. In *Proceedings of the First International Workshop on Mobile Agents*, volume 1219, pages 86–97. Springer-Verlag.