Management of Multimedia Contents for Distributed e-Learning System

Kazunari Meguro, Daisuke Yamamoto, Shinichi Motomura, Toshihiko Sasama, Takao Kawamura and Kazunori Sugahara

Abstract—We have developed a distributed asynchronous Web based training system. In order to improve the scalability and robustness of this system, all contents and functions are realized on mobile agents. These agents are distributed to computers, and they can use a Peer to Peer network that modified Content-Addressable Network. In the proposed system, only text data can be included in a exercise. To make our proposed system more useful, the mechanism that it not only adapts to multimedia data but also it doesn't influence the user's learning even if the size of exercise becomes large is necessary.

Keywords—e-Learning, multimedia, Mobile Agent.

I. INTRODUCTION

Asynchronous Web-Based Training systems (hereafter we abbreviate as WBT) are very popular in e-Learning systems. A WBT allows a learner to complete the WBT in his own time and schedule, without live interaction with an instructor. Although a large number of studies have been made on asynchronous WBT [1] [2] [3], all of them are based on the client/server model. In the client/server model, a server machine offers all functions and manages all data. The client/server model has an advantage of easy construction and maintenance. However, the client/server systems generally lack scalability and robustness.

There is a Peer to Peer (hereafter we abbreviate as P2P) model to supplement the disadvantage of client/server model. There is a feature in the system based on the P2P model that each computer works as a client of a server. The feature can distribute the load to a node. Moreover, the function of the entire system doesn't stop even if some nodes break down.

We have proposed and implemented a distributed e-Learning system based on P2P architecture [4] [5]. The proposed e-Learning system has two distinguishing features. Firstly, it is based on P2P architecture. Every user's computer (hereafter we refer to such a computer as a node) plays the role of a client and a server. When a node joins in the system, the part of contents is given from a joined node, and it has the responsibility to manage the contents and of sending appropriate contents to the requesting nodes. Secondly, each exercise in the system is not only but also an agent so that it has functions, such as scoring user's answers, giving the correct answers, and showing some related information without human instruction.

K. Meguro is with Ph.D. student in Tottori University.
D. Yamamoto is with M.Sc. student in Tottori University, S. Motomura is Information Media Center Tottori University, T. Sasama, T. Kawamura and K. Sugahara are with the Department of Information and Knowledge Engineering Tottori University.

In the proposed system, an agent is made by an exercise, and is distributed to nodes in the system. When a user learns an exercise, copies of agents which hold the exercise is send to learner's node. However, in the proposed system, only text data can be included in an exercise. Audio data and video data (hereafter we refer to these data as an Audio-Video data) cannot be included in an exercise. In this paper, the method of making the Audio-Video data correspond to the exercise is described. There are two issues which should be considered to introduce the method. When an agent holds all of multimedia data in exercise, the size of multimedia data becomes large, the size of agent becomes large. Consequently, time until the agent is sent to learner's node increases, and a start of the learning is late. As a resolution of the first issue, the method of dividing the agent which holds the text data and the agent which holds the Audio-Video data is introduced. Introducing the method, when a learner learns exercise with contents of large data, he can start learning by comparatively short time. The second issue is time till being played the Audio-Video data. When an agent holds an Audio-Video data, the Audio-Video data is not played till migration to a node which demanded learning is completed. Consequently, the larger the size of the Audio-Video data becomes, the later the data is played. As a resolution of the second issue, the agent which holds the Audio-Video data is divided into a constant size, and the divided agents are distributed to each node. When a learner requests the Audio-Video data, the divided Audio-Video data are played by streaming. Time till being played the Audio-Video data can shorten by this method.

This paper is organized in 6 sections. The proposed e-Learning system is described in Section 2. We describe the design of adaptation to the Audio-Video data in Section 3. The design of the streaming play of the Audio-Video data is described in Section 4. In Section 5, we describe the experimental result of design describing in Section 3 and Section 4.

II. PROPOSED E-LEARNING SYSTEM

A. Components of system

In order for the proposed system to be considered as a distributed WBT system, functions must be distributed among all nodes. Mobile agent technology is adopted to achieve this goal.

There are following agents and user interface programs on each node. These agents have implemented in the mobile agent framework that we have developed [10].

- **Exercise Agent:** Each Exercise Agent has questions and functions to score user's answers, give the correct answers, and show some related information about the exercise.
- **Category Agent:** Each Category Agent stands for a unit of a particular subject. It manages Exercise Agents in itself and sends them to the requesting node.
- **User Agent:** Each user has its own User Agent. A User Agent manages its user's information that includes login name, password, IP address of the user's computer, on-line/offline status, and log of studying or a list of created exercises.
- **Student Interface:** One student interface is on each node to which a user logs in as a student. It is a user interface program for studying.

B. P2P network

All exercises in the proposed system are classified into categories, such as "Math/Statistic", "English/Grammar", etc. A user can obtain exercises one after another by specifying categories of the required exercises. While a user uses the proposed e-Learning system, his/her node is to be a part of the system.

When the proposed system bootstraps, one initial node has all categories in the system. When another node joins the system, it receives a certain number of categories from the initial node. The categories are distributed among all nodes in the system accordingly as nodes join in the system or leave from the system.

In existing P2P-based file sharing systems, such as Napster [6], Gnutella [7], and Freenet [8], each shared file is owned by a particular node. Accordingly, files are originally distributed among all nodes. On the other hand, 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. That being the case, the P2P network of the proposed system can be constructed as a CAN [9].

Our P2P network is constructed with 2-dimensional coordinate space [0.0,1.0] × [0.0,1.0] to store categories, as shown in Figure 1. The figure shows the situation that node C has just joined the system as the third node. Before node C joins, node A and node B shared the whole coordinate space half and half. At that time, node A managed "Math/Geometry", "Math/Statistics", and "History/Rome" categories and node B managed "English/Grammar", "English/Reader" and "History/Japan" categories, respectively. When node C joins the system, we assume node C already knows the IP addresses of some nodes in the system and node C sends the join request to some node in the list. Then node C 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 Figure 1, node C takes on the "History/Japan" category from node B, then exercises in that category move to node C. After joining, node C gets a list of IP addresses of neighbor nodes in the coordinate space, such as node A and node B. Therefore, neighbor nodes can communicate with each other.
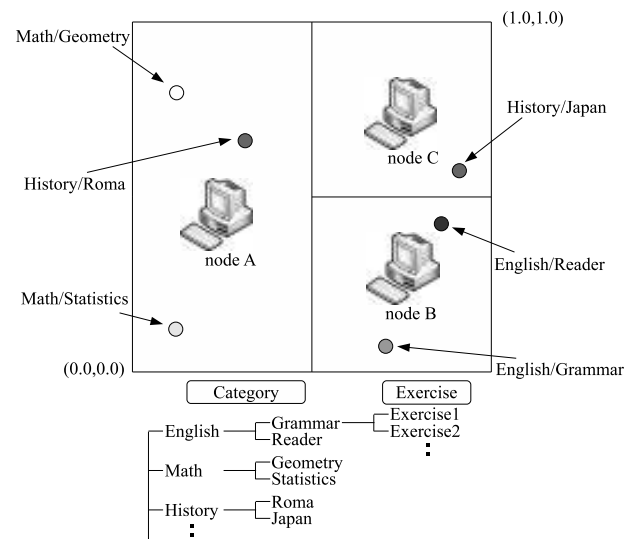

Fig. 1. P2P network of the proposed e-Learning system.

Because each joined nodes manage the part of the coordinate space, P2P network is constructed.

III. ADAPTATION TO LARGE CAPACITY MULTIMEDIA DATA

In the proposed system, only text data can be included in an exercise. To adapt to multimedia data such as the Audio-Video data, the format of Flash Video, Shockwave Flash and MPEG Audio Layer-3 can be newly treated in the system. In making exercise corresponding to these formats, when the EA holds the Audio-Video data as well as the proposed system, size of the EA becomes large and spends a lot of time on migration of the EA. Therefore, the Media Agent (Hereafter we refer to Media Agent as a MA) is newly implemented to divide the Audio-Video data from the EA. The MA holds the Audio-Video data, and is managed by nodes which join in a e-Learning system. A learner can use the Audio-Video data by letting the EA refer to the MA. In an e-Learning system, there is a case that different exercise uses the Audio-Video data. Therefore, the method by which two or more exercise can share the MA is introduced. The MA is managed by the following mechanisms.

- **Creation of MA** When the EA is created, if the Audio-Video data is included in exercise, the MA is created by the following procedures. Figure 2 illustrates a procedure that the MA is created.
    1) The EA creates the MA in an e-Learning system. The created MA is created one per an Audio-Video data.
    2) The created MA is mapped a coordinate space on the Distributed Hash Table (hereafter we abbreviate as DHT) corresponding to the key of the Audio-Video data.
    3) Because the EA which creates the MA manages the key of the Audio-Video data, the EA can request the Audio-Video data in other node.
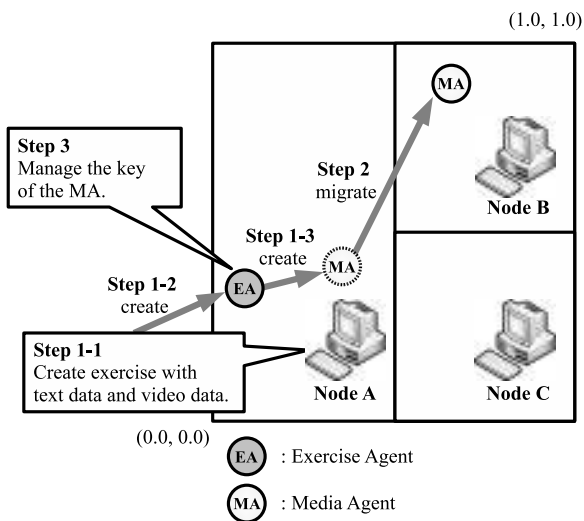
Fig. 2.   Procedure for creating Media Agent.

Fig. 3.   Procedure for obtaining Media Agent.

- **Sharing MA** The MA is shared by the following procedures.
  1) The EA investigates whether the key of the Audio-Video data already exists in the system before the EA creates the MA.
  2) If the key of the Audio-Video data exists, the EA includes the key in own exercise.
  3) If the key of the Audio-Video data doesn't exist, the EA creates the MA.
- **Deletion of MA** The MA is shared by exercises. Therefore, the MA can be deleted when it isn't referred from all exercise. The MA is deleted by the following procedures.
  1) The MA holds a counter for the number of the EA which refers to the MA oneself.
  2) When the EA creates the MA, if the key of the Audio-Video data already exists in the system, the counter of existing MA is incremented.
  3) When the EA is deleted from the system, the counter of MA which is referred by the MA is decremented.
- **Obtaining of MA** In the proposed system, a learner learns on the Student Interface. When the learner requests exercise, the MA is offered by the following procedures. Figure 3 illustrates a procedure that the MA is obtained.
  1) A copy of the EA corresponding to the requested exercise is sent to requesting node, and text data which is held by the EA is shown on the Student Interface.
  2) Then, the Student Interface requests the Audio-Video data included in exercise on a background sequentially, and a MA which holds the requested Audio-Video data is retrieved.
  3) A copy of the found MA migrates to the requesting node, and the Audio-Video data which is held by the MA is offered to the Student Interface. Then, the Audio-Video data is shown on the Student Interface. By using this method, even if the size of the Audio-Video data in exercise becomes large, the learner
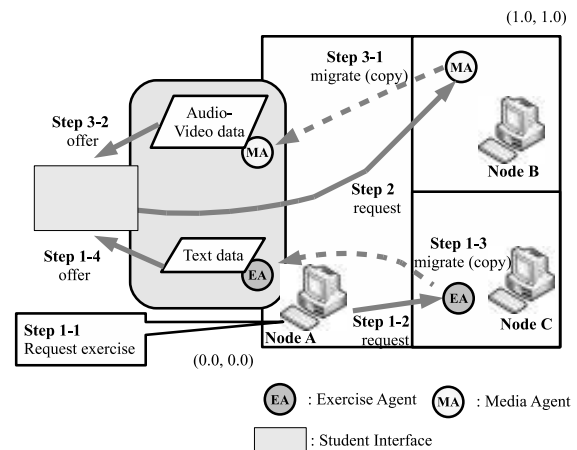
can begin learning by constant waiting time.
- **Processing when MA obtaining fails** In implementation of practical e-Leaning system, it is necessary to cope with a failed case. The case is that obtaining of contents from the MA is failed while obtaining of contents from the EA is succeeded. If the failed contents are necessary to solve an exercise, a learner will solve the exercise without obtaining enough information. As a resolution for this issue, if the failure occurs, a flag of the failure is put up in log of studying.

IV. STREAMING PLAY OF DISTRIBUTING AUDIO-VIDEO DATA

In Section 3, the mechanism of MA is described. However, in the method described in Section 3, the Audio-Video data doesn't play till the MA finishes migrating to learner's node. As a resolution for this issue, we propose a method that the Audio-Video data is played by streaming on the learner's node. To realize the method, he MA is managed by the following mechanisms.

- **Creation of divided MAs** When the EA is created, if the Audio-Video data is included in exercise, the divided MAs are created by the following procedures.
  1) The EA creates the divided MAs in an e-Learning system.
  2) The created MAs are respectively mapped coordinate spaces on the Distributed Hash Table (hereafter we abbreviate as DHT) corresponding to the keys of divided Audio-Video data.
  3) Each divided MA has 2 keys of MAs which playing order becomes front and back of own.
  4) The EA which creates the MAs manages keys of divided Audio-Video data.

- **Deletion of divided MAs** The MAs are shared by exercises. Therefore, each MA can be deleted when it isn't referred from all exercise as same as described Section 3.

- **Obtaining of divided MAs** The divided MAs are offered by the following procedures. Figure 4 illustrates a procedure that the MA is obtained.
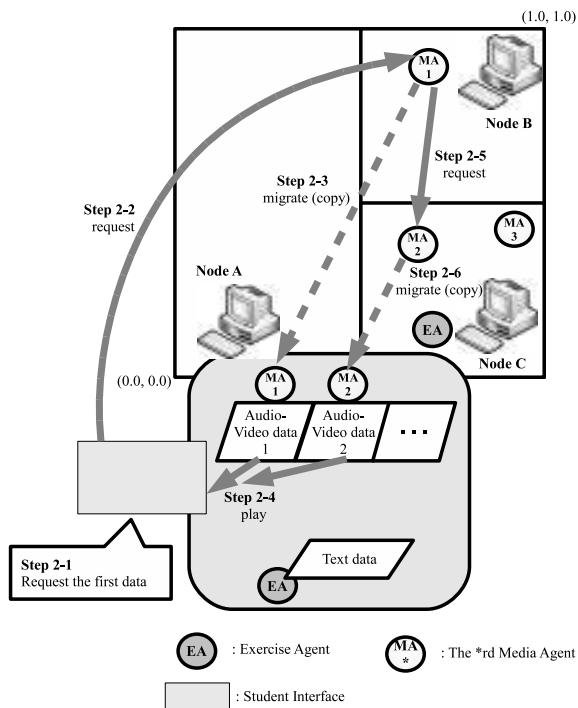

Fig. 4.   Procedure for obtaining divided Media Agent.

1) A copy of the EA corresponding to the requested exercise is sent to requesting node, and text data which is holds by the EA is shown on the Student Interface.
2) Then, the Student Interface requests the first data of divided Audio-Video data included in exercise on a background sequentially, and a MA which holds the first data of divided Audio-Video data is retrieved. Then, the found MA retrieves the next MA of own.
3) Copies of found MAs migrate to the requesting node. On the Student Interface, the Audio-Video data plays by streaming as soon as the first data of divided Audio-Video data is offered. By using this method, even if all MAs don't finish migrating, the learner can play the Audio-Video data.

V. EXPERIMENTS

This section presents the experimental results for confirming the operation and performance of the MA.

A. One MA holds an Audio-Video data

As an experiment 1, we confirm the following operation when the MA is implemented in the proposed system.

- Exercises including an Audio-Video data can be correctly obtained.
- A MA can be shared by some nodes.

- If obtaining the Audio-Video data failed, it can be confirmed not to obtain it.

Table I shows the experimental condition and Table II shows the machine specification in experiment 1.

TABLE I
EXPERIMENTAL CONDITION.

Number of node	5
Number of category	20
Number of exercise	60
Number of MA	30

TABLE II
MACHINE SPECIFICATION.

CPU	Intel Pentium4 3.0 GHz
Memory	1GB
Network	1000BASE-T
OS	TurboLinux 11 Desktop

Then, as an experiment 2, the response times are compared when the EA holds all multimedia data and when the MA holds the Audio-Video data. The response time in this experiment means the time which takes to display after the learner request an exercise on the Student Interface.

Table III shows the experimental condition and Table IV shows the machine specification in experiment 1.

TABLE III
EXPERIMENTAL CONDITION.

Number of node	12
Number of category	28
Number of exercise	84
Number of MA	74

TABLE IV
MACHINE SPECIFICATION.

CPU (from node 1 to 8)	Intel Pentium4 2.53 GHz
CPU (from node 9 to 12)	Intel Pentium4 2.66 GHz
Memory (node 7, 8)	512MB
Memory (the others)	256MB
Network	1000BASE-T
OS	Debian GNU/Linux 4.0
OS	Ubuntu 8.04 LTS

1) Experiment 1: In the first confirmation item, when exercise including an Audio-Video data is requested, the Audio-Video data is displayed on Then, when the number of MA to which exercises refer is confirmed, it is confirmed that exercises referred to the MA more than the existing number. That is to say, MAs are shared by some nodes. In the third confirmation item, a MA which is referred from an exercise is deleted by force. Then, when a learner learns the exercise, the flag of the failure is put up in the log of studying.

2) Experiment 2: Figure 5 shows result of experiment2. Regardless of the size of multimedia data, the response time was earlier to hold the Audio-Video data in the MA than that of the EA. Moreover, in the case the MA holds the Audio-Video data, response time has hardly changed even if the size of data changes. That is to say, implementing the MA in the proposed system is useful from the point of learning performance.
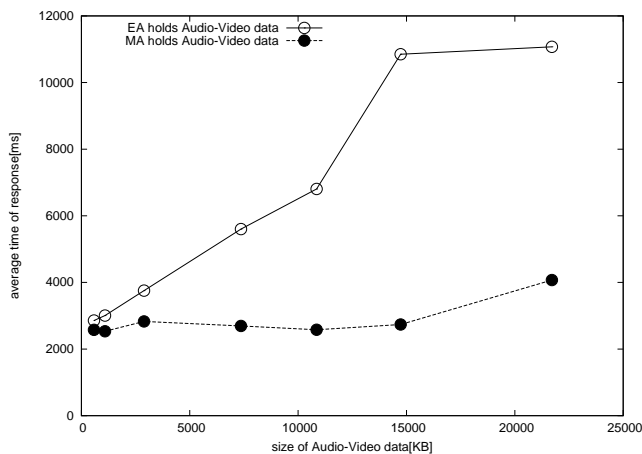
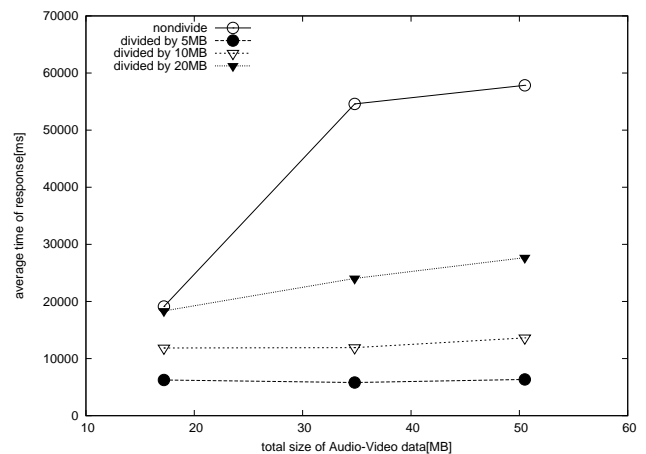Fig. 5. Comparison of response times when EA holds all multimedia data and when MA holds the Audio-Video data.

Fig. 6. Comparison of waiting time of playing by division size.

B. Some MAs hold divided Audio-Video data

We investigate how the waiting time of playing an Audio-Video data changes by division size of the data. Table V shows the machine specification in the experiment. This experiment is done according to the following procedures. The sizes of Audio-Video data are 17.2MB, 34.8MB and 50.5MB. The division size of Audio-Video data are 0MB (i.e., non divide), 5MB, 10MB and 20MB.

- Twenty nodes are participated in a system.
- One of these nodes starts a Student Interface. Then an Audio-Video data is required from the Student Interface.
- A time from the requiring to the start of playing is measured.
- The measurement is repeated five times on the conditions that the Student Interface and the Audio-Video data are changed randomly.

TABLE V
MACHINE SPECIFICATION.

CPU	Intel Pentium4 3.00 GHz
Memory	1GB
Network	1000BASE-T
OS	TurboLinux 11 Desktop

Figure 6 shows result of experiment. As a result, the waiting time of playing an Audio-Video data shortens by dividing the data, and the smaller the division size becomes, the shorter the waiting time of playing the data becomes. The waiting time of playing the Audio-Video data shortening leads to the improvement of the learning efficiency.

VI. CONCLUSION

We have proposed the method of making multimedia data correspond to the exercise. As a result, it becomes possible to include not only text data but also audio data and video data in exercises. Because of the size of audio data and video data become larger comparing with text data, the MA which is newly implemented in the proposed system holds these data. Consequently, even if the size of audio data or video data

in an exercise data increases, it becomes possible to display the exercise which contains only text data at a little time. Moreover, we have proposed the method of playing divided audio data by streaming. As a result, when division size of an Audio-Video data is smaller than total size of it, a time till being played the Audio-Video becomes shorter than the method of holding a video data by one MA.

The division size of an Audio-Video data may not be too small. The reason is the playing of the previous data ends before the following data reaches learner's node, and the Audio-Video data doesn't play smoothly. In future work, it is necessary to investigate how much size of dividing an Audio-Video data is more effective to start earlier.

REFERENCES

[1] Nishita, T. and et el,: 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] Napster, http://www.napster.com (1999).
[7] Gnutella, http://welcome.to/gnutella/ (2000).
[8] 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).
[9] 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).
[10] 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).

Kazunari Meguro was born in 1977. He received his B.Eng. and M.Eng. degrees from Hiroshima city University, Japan, in 2002 and 2004, respectively. He is currently a Ph.D. student in Tottori University. His research interests multi-agent systems and distributed systems.

Daisuke Yamamoto was born in 1986. He received his B.Eng. degrees from Tottori University, Japan, in 2009. He is currently a M.Sc. student in Tottori University. His research interests multi-agent systems and distributed systems.

Shinichi Motomura was born in 1973. He received his B.Eng. and M.Eng. degrees in Computer Engineering from Toyohashi University of Technology, Japan, in 1995, 1997, respectively. From 2004 to 2008, he joined Tottori University as a research associate. In 2008, he received the D.Eng. degree from Tottori University, Japan. Since 2008 he had been in Tottori University as an associate professor in the Information Media Center. His research interests include multi-agent systems and distributed systems.

Toshihiko Sasama was born in 1972. He received his Ph.D. degree from Osaka University, Japan, in 2001. He is now an assistant professor in the Department of Information and Knowledge Engineering of Tottori University. His current research interests include mobile ad-hoc networks. Dr. Sasama is a member of IPSJ and IEICE.

Takao Kawamura was born in 1965. He obtained his B.Eng., M.Eng. and Ph.D. degrees in Computer Engineering from Kobe University, Japan, in 1988, 1990 and 2002, respectively. Since 1994 he had been in Tottori University as a research associate and has been in the same University as an associate professor in the Faculty of Engineering since 2003. His current research interests include mobile-agent systems and distributed systems. Dr. Kawamura is a member of IPSJ and IEICE.

Kazunori Sugahara received the B.Eng. degree from Yamanashi University, Japan, in 1979 and M.Eng. degree from Tokyo Institute of Technology, Japan, in 1981. In 1989, he received the D.Eng. degree from Kobe University, Japan. From 1981 to 1994, he was on the staff of the Department of Electronic Engineering, Kobe City College of Technology. In 1994, he joined Tottori University as an associate professor of the Department of Electrical and Electronic Engineering and he is a professor of the department of Information and Knowledge Engineering. His current interest lies in the fields of computer architectures and hardware realizations of image processing algorithms. Prof. Sugahara is a member of IEEE, IPSJ and IEICE.