Development of a Distributed Pair Exercise System for Network Construction with a Dialogue Support Function

Yuichiro Tateiwa
Nagoya Institute of Technology
Aichi-ken, Japan
tateiwa@nitech.ac.jp

Yoshiaki Ooka
Nagoya Institute of Technology
Aichi-ken, Japan
ooka@moss.elcom.nitech.ac.jp

Yonghwan Kim
Nagoya Institute of Technology
Aichi-ken, Japan
kim@nitech.ac.jp

Yoshiaki Katayama
Nagoya Institute of Technology
Aichi-ken, Japan
katayama@nitech.ac.jp

***Abstract*—This Research to Practice Full Paper presents a distributed pair exercise system for network construction with a dialogue support function. In a certain type of network construction exercise, each of the two learners pairs up and constructs networks. One member of a pair, called driver, operates network components, while another member of the pair, called navigator, checks the driver's work. Several exercise systems realize environments where learners in distant places carry out network construction exercises using virtual machines. However, there are two problems associated with such network construction exercise systems: 1) it is difficult for the members of a pair to edit a common network at the same time and 2) it is difficult for the members of a pair to share attention areas efficiently. Attention areas are areas that a member would like the partner to look at. We propose an exercise system that has GUIs for editing common networks at the same time, selecting attention areas intuitively, and visualizing attention areas. The experiment confirmed that subjects conveyed attention areas more easily, faster, and more accurately using the proposed method than with plain text (traditional method).**

***Index Terms*—network construction, pair exercise, chat, virtual machine**

I. Introduction

In a certain type of network construction exercise, each of the two learners pairs up and constructs networks. One of the members of a pair, called **driver**, connects cables and sets up parameters of devices, while the other member of the pair, called **navigator**, observes the driver's work and reviews their own network. In addition, they have dialogues: the driver asks questions about the state of the network to the navigator, and the navigator gives advice to solve problems. We call such exercises **pair exercises** in this paper. In recent years, much attention has been paid to distance learning through which e-learning systems have been developed. In [1], researchers developed an e-learning system where learners in distant places are able to carry out pair exercises (say **distributed pair exercises**), and found that the exercises raise the learning effect and reduce the burden on instructors.

Systems [1]– [4] can provide remotely-editable networks and communication tools between learners of a pair. Using the systems, it is possible to carry out distributed pair exercises. However, because the systems serve a text chat feature for communications between users, it is difficult for the members of a pair to share **attention areas** efficiently; attention areas are areas that a member would like the partner to look at. For example, generating and understanding the expression "a character located at the fifth row from the top and the tenth column from the left on the console of node A" wastes time.

We propose a distributed pair exercise system for network construction and present how to realize this system in the paper. Furthermore, in experiments, we discuss how efficiently the members of a pair can share attention areas. The system has the following features.

1) The members of a pair can edit a common network consisting of virtual machines at the same time through the Internet. Furthermore, a navigator is able to watch the operations by the driver in real time.
2) Members can select attention areas intuitively using a computer mouse. These members can share messages including hyperlinks for the areas using text chat. GUIs can visualize the attention areas based on hyperlinks. Such hyperlinks are called **attention area links**.

The contributions of the paper confirm that visualizing attention areas have a positive impact on dialogues and how one could realize an exercise system that hosts such dialogues.

II. Related work

VNLab [5], SAIL [6], and the system in [7] provide remotely-editable networks with learners. However, the systems cannot accomplish our objectives because they have no features for communication between the learners.

The system in [1] realizes an environment for pair exercises. The server machine implements networks using virtual

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

machines as nodes. The client machine provides user interfaces for consoles of virtual machines and text chat between members. The system has a feature to share input commands in consoles between the members in real time. By combining Remote Desktop, Microsoft Treminal Services, Remote Assistance, and VMWare Workstaion, RLES [2] provides an environment where remote learners construct networks. Furthermore, a navigator can watch the driver's work in real time through Remote Assistance windows. NetPowerLab provides an environment where multiple learners are able to construct networks through a division of labor [3], and serves software agents that play the role of the partners [4]. However, these systems cannot solve the problem mentioned before as it adopts text chat as the tool for dialogues between members.

IDEOL [8] provides a web-based environment for distributed pair programming and has a feature for linking lines of a source code to a dialogue. The feature helps members that are interested in lines of a source code to create a dialogue of lines and to be aware of the lines that are associated with a dialogue. However, the feature does not help members to compare multiple areas in network topologies and console outputs as well as pay attention to other areas in the middle of a dialogue. In contrast, attention area links allows one to combine multiple attention areas and embed multiple attention areas in one utterance. RIPPLE [9], Xeclip [10], saros [11], and sangam [12] are eclipse plugins that can share source codes between eclipse processes through networks in real time. In programming exercises described in these works, learners dialogue and share attention areas using text chat, not special tools for sharing attention areas.

III. Network construction pair exercise

A. Exercise summary

Target learners are those that have studied TCP/IP but have never constructed networks. The aim of the exercise is to accustom learners to designing networks that satisfy the following requirements: 1) IP address and subnet mask, 2) IP network in a segment, 3) default route, 4) static routing, and 5) server service (web page publishing). They must also be able to construct networks based on the designs.

Exercise problems consist of **communication examples** and **setting requirements**. The former is an example of the communication available in the learner's networks. The latter consists of nodes that must be installed in networks, values that must be set in nodes, and Ethernet cables that must be connected to the assigned nodes.

In the exercises, learners try to construct networks based on the exercise problems. They repeat the following steps by reading the problems and checking the network settings until the networks have been built to the specifications. In network construction, an instructor alternates the driver and the navigator in a pair at fixed intervals. The driver constructs networks, while the navigator observes the partner's work and reviews their own network. In addition, they have dialogues: the driver asks questions about construction and the navigator gives advice to improve the driver's work.

1) Network design: connection design, IP addresses design, and so on
2) Network implementation
   - Hardware setting
     - Machine: installation/uninstallation, boot/halt
     - Cable: connection/disconnection
   - Software setting: IP address assignment, route addition, and so on
3) Operation check: continuity check (ping), route check (traceroute), and so on

B. Dialogue between driver and navigator

For example, dialogues of the members of a pair occur with the following triggers.

- Driver
  - Implementation: a driver asks a question about a parameter at a specific position while inputting a command
  - Operation check: a driver asks a question about a part of command output while analyzing the output
- Navigator
  - Observation: a navigator points out an error part of the command that the driver is inputting
  - Operation check: a navigator points out an operation error according to a specific part of command output
  - Settings check: a navigator points out error values in command output

Here, we define three terms. **An utterance** is the information that a member of a pair conveys to its partner at once in a dialogue. **An attention area** is the information that a member of a pair would like the partner to look at. **An attention area group** is a group of attention areas.

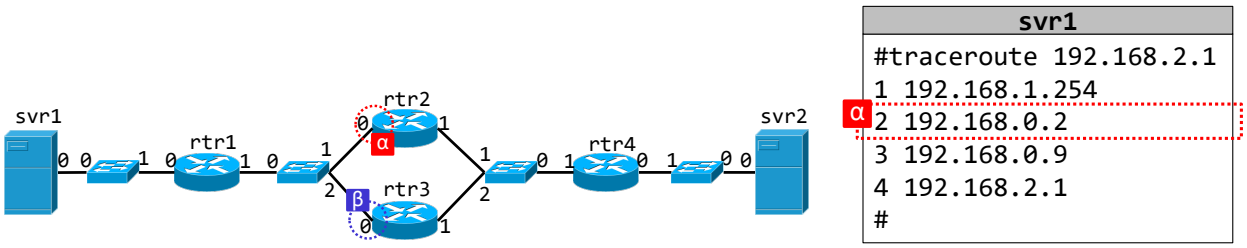
The members of a pair exchange utterances between themselves until the desired result is obtained. The first utterance in a dialogue includes one or more attention areas. The other utterances can include attention areas. The following four items are the types of attention areas in our model.

- Device
- Ethernet port
- Cable
- Characters in a rectangle in console output

C. Example of utterance

A pair is constructing a network where svr1 and svr2 exchange data through a specific route. Figure 1a denotes the current network topology. Figure 1b displays the current console output of svr1. Table I shows IP addresses of the devices.

When the navigator of the pair notices an error in the current communication route with traceroute, he/she points out that "data must pass through port β instead of port α " as shown in Figure 1. Note that α is an attention area group that shows the current route (port 0 in rtr2) and its ground ("2 192.168.0.2"). Also, β is an attention area that shows a specific route (port 0 in rtr3).



(a) Network topology

(b) Console window of svr1

Fig. 1. State of incorrect network

TABLE I
ASSIGNED IP ADDRESSES

ID	Port	IP address
svr1	0	192.168.1.1
rtr1	0	192.168.1.254
rtr2	0	192.168.0.2
rtr3	0	192.168.0.3
rtr4	1	192.168.0.9
svr2	0	192.168.2.1

IV. SYSTEM REQUIREMENTS

We define system requirements that provide the environment for the exercises described in Section III-A.

1) A pair can carry out the exercises described in Section III-A.
2) A member of a pair can see that the partner is editing their own network in real time.
3) All the members of a pair can edit a common network at the same time.
4) A pair whose members are at distant places can carry out the exercises.
5) All the members of a pair can share attention areas.
6) The members of a pair can dialogue through text chat.
7) A pair can embed attention areas in messages easily.
8) A pair can select attention areas from network components intuitively.
9) A pair can select attention areas from characters in console output intuitively.
10) A pair can compose an attention area group whose elements are network components and characters in console output.
11) The system can visualize attention areas that a pair can select from a dialogue.

V. PROPOSED SYSTEM

In the proposed system in Figure 2, a pair carries out distributed pair exercises through chat clients and network editors. Network manager implements networks constructed by a pair using virtual machine User-mode Linux [13]. We call such a network **virtual device network (VDN)**. Network editor is a GUI where a pair edits a common VDN through the Internet. It consists of console windows of devices in the network and a GUI where a member edits network topology

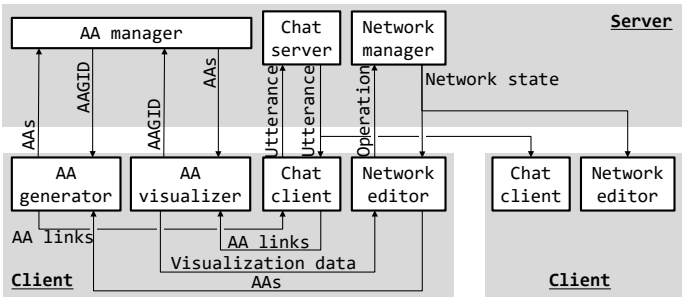


Fig. 2. System overview

(e.g., installing devices). Chat server and Chat clients enable members to convey utterances (plain texts and attention areas). AA manager, AA generator, and AA visualizer encode/decode attention areas in the chat system.

Utterance consists of user ID uid, time t, and message msg, where a member with uid sends msg at t. The message consists of attention area links and natural language. An attention area link is a hyperlink that points to an attention area group (i.e., one or more attention areas) and can be embedded in a message (Figure 3). An attention area of a network component consists of device ID did that points to a network device and Ethernet port name ep in the device. An attention area of characters in console output is characters in a rectangle, and consists of console ID cid, a row number of the left top corner tr, a column number of the left top corner tc, a row number of the right bottom corner br, and a column number of the right bottom corner bc. Attention area manager (AA manager) stores attention areas (AAs) using data structure of a set of tuples (attention area group ID $aagid$, did, ep, cid, tr, tc, br, bc).

To implement system requirements 1 and 2, Network editor receives operations from a member and then sends them to Network manager. Network manager reflects them to VDN and sends the state of VDN (i.e., network configuration and

```
<a href="AAGID">string</a>
```

Fig. 3. Attention area link

console output) to Network editors of a pair. Network editors update its own view according to the state.

To implement system requirement 3, the system can handle operations from a pair simultaneously by realizing system requirements 1 and 2 using a multithreading technique.

To implement system requirement 4, each member in a pair uses a different client machine; the client machines and the server machine exchange data using network communications.

To implement system requirement 5, AA manager stores AAs, and attention area visualizer (AA visualizer) obtains AAs that correspond to attention area group ID (AAGID) from AA manager.

To implement system requirement 6, Chat client provides the member with a text entry field and a text output field.

To implement system requirement 7, attention area generator (AA generator) generates AA links that corresponds to AAs selected by a member and then sends them to Chat client. AA visualizer receives AA links from Chat client and then generates Visualization data for the attention areas that corresponds to AAGID in Attention area link and finally sends it to Chat client.

To implement system requirements 8 and 9, Network editor provides a member with a user interface where the member can select attention areas and send the selected attention areas (AAs) to attention area generator (AA generator).

To implement system requirement 10, AA manager can store multiple attention areas (AAs) by the same $aagid$. Furthermore, AA area manager sends all attention areas (AAs) whose $aagid$ is equal to attention area group ID (AAGID) received from AA visualizers.

To implement system requirement 11, Chat client provides a pair with a user interface where the pair can select attention areas, and sends attention area links (AA links) that corresponds to the selected attention areas to AA visualizer. AA visualizer generates Visualization data according to AA links and sends it to Network editor of the pair. Network editor visualizes attention areas according to the data.

VI. Prototype System

A server machine provides VDN and realizes exchange of chat messages and share of network construction processes in a pair. Each client machine provides a member in a pair with a GUI in Figure 4. A member can edit a network with areas α ,β, and γ. Area γ displays a network topology that a pair shares. A member can compose a chat message with plain

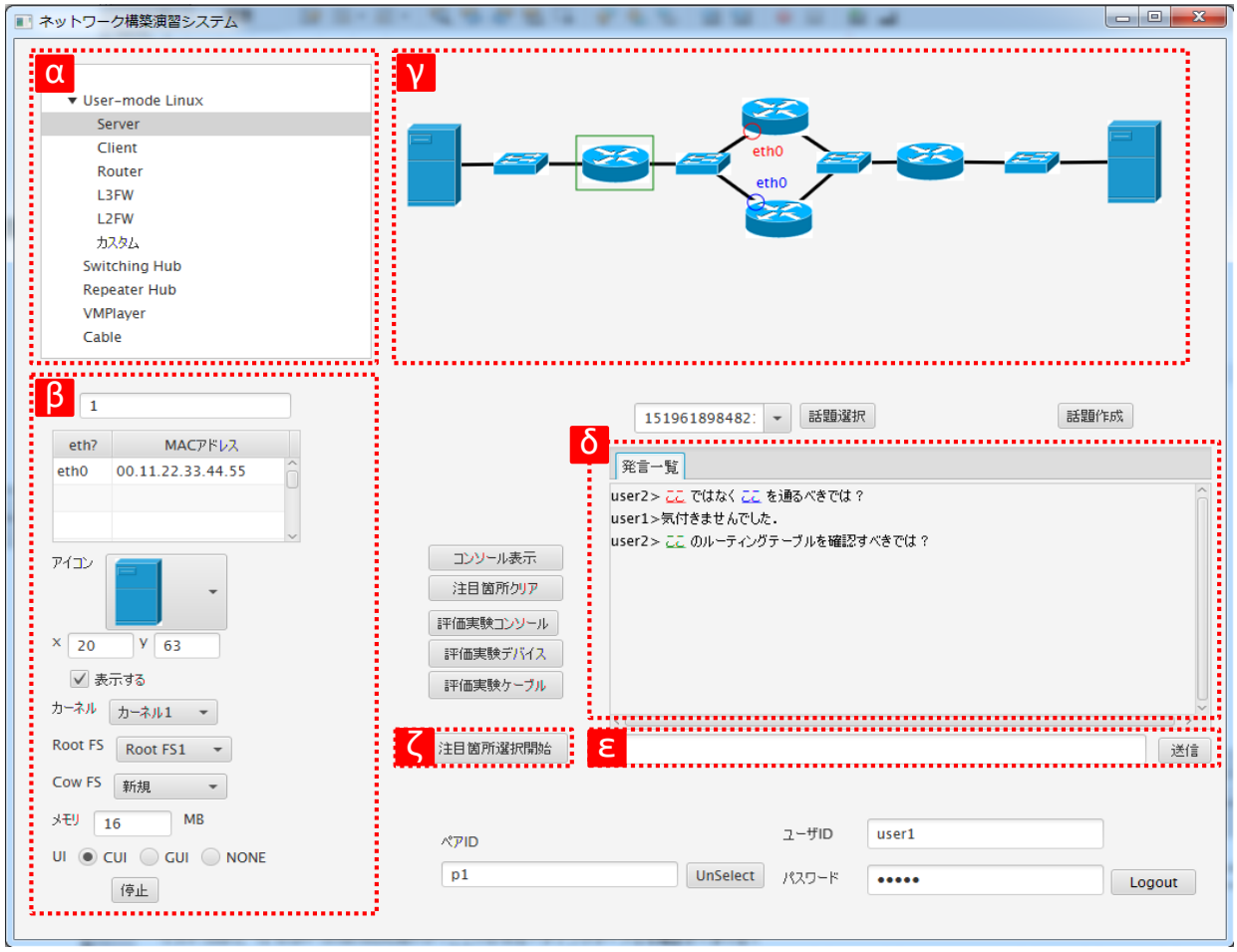


Fig. 4. Network editor

text and attention area links in area ϵ. Area δ displays chat messages.

Attention areas are selected according to the following steps.

1) Push button ζ to start selecting attention areas.
2) Drag on a console window to select characters in a rectangle area (Figure 5a) as an attention area. Click an icon in area γ and then select a part of the component in the dropdown list (Figure 5b) as an attention area.
3) Push button ζ again to finish selecting attention areas and add the attention area link to the message box in area ϵ. Figure 6 shows an example of an original utterance including hyperlinks.

When you click a hyperlink in area σ, the system highlights the attention area corresponding to the hyperlink (Figure 7). Hyperlink α in Figure 7a is highlighted like characters α in Figure 7b and like port α in Figure 7c.

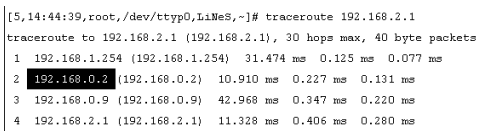
VII. Experiment

The purpose of the experiment is to clarify how efficiently a member of a pair conveys attention areas to the partner with the system. It compares the proposed method with the traditional method where a member expresses attention areas with plain text. The subjects are twenty students (ten senders and ten receivers) that constructed simple networks like the one in Figure 1a one year back. The flow of the experiment is as follows.

1) The subjects construct a network like the one in Figure 1a with the system to remember the knowledge/skill for network construction.
2) The subjects express and look at attention areas with both the methods using the system to get accustomed to it.
3) The experimenter evaluates the speed and accuracy in dialogues between the subjects using both the methods. The subjects answer questionnaires to evaluate the easiness of both the methods and operability of the system. The details regarding this are given in Section VII-A.

A. Evaluation

Each assignment consists of a few steps for reaching the situation where a member tends to generate an utterance,

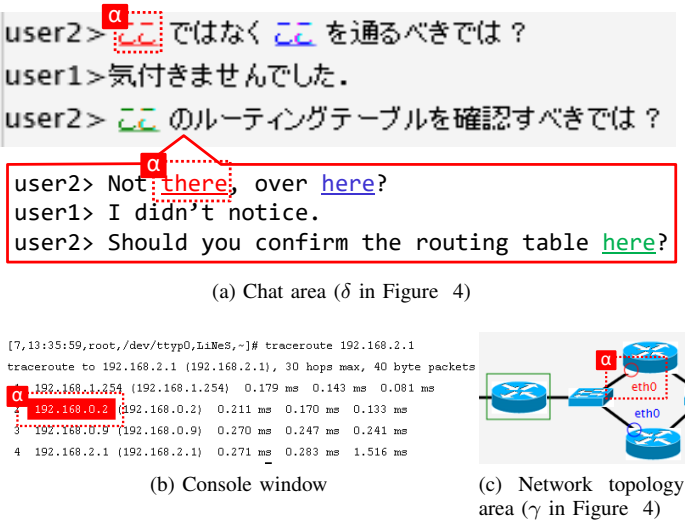


(a) Characters in console output (b) Network device

Fig. 5. Attention area selection

```
Not <a href="1517979440167730">there</a>,
 over <a href="1517979503616861">here</a>?
```

Fig. 6. Example of original utterance



(a) Chat area (δ in Figure 4)

(b) Console window

(c) Network topology area (γ in Figure 4)

Fig. 7. Attention area visualization

and a mission of expressing attention areas with either of the methods. The steps are concrete enough that all of the subjects complete the same network. The experimenter gives each pair 12 assignments that are based on six kinds of attention areas (Table II); the assignments are given in the order of 1P, 1T, 2T, 2P, 3P, 3T, 4T, 4P, 5P, 5T, 6T, and 6P, represented by "NM" where N is a kind of attention areas in Table II and M is a conveyance method (T: traditional method, P: proposed method). In each assignment, a sender constructs a network based on the steps and then he/she generates an utterance including attention areas; finally, he/she sends the utterance to the receiver. When a receiver receives an utterance, he/she starts to search for attention areas. When he/she is able to successfully search, he/she marks them in an answer sheet. After completing each answer sheet, a pair answers questionnaires to evaluate the effectiveness of the used method and operability of the system on a scale of 1 to 5. A sender evaluates how easily he/she expresses attention areas with the proposed method when compared to the traditional method. A receiver evaluates how easily he/she finds attention areas with the proposed method when compared to the traditional method.

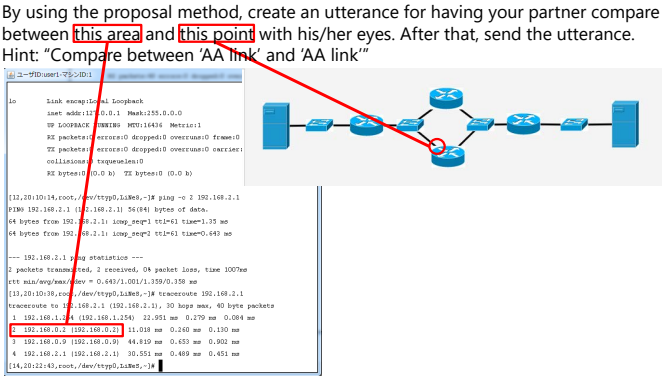
TABLE II
Six kinds of attention areas

No.	Summary
1	Two points in a network topology
2	A part of the output of command "ping"
3	Each IP address outputted by command "ifconfig" in two console windows
4	A route outputted by command "route"
5	An IP address in a route outputted by command "traceroute" and an Ethernet port related to the IP address in a network topology
6	Ethernet port information outputted by command "ifconfig" and an Ethernet port related to the information in a network topology

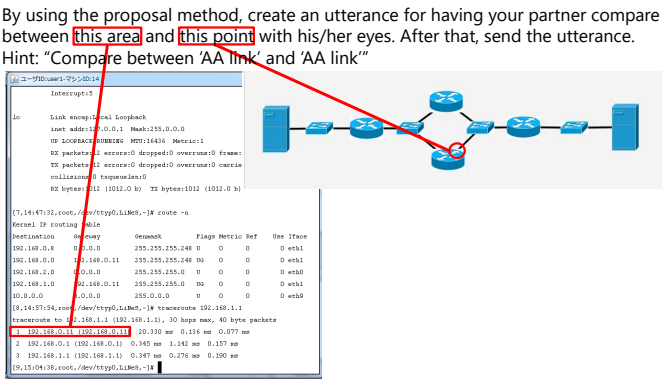
Figures 8a and 8b show missions for the proposed method and the traditional method respectively, which is based on attention area No.5 in Table II. The situations are occurred by that the senders execute command "traceroute 192.168.2.1" on the console window of machine 1 and command "traceroute 192.168.1.1" on the console window of machine 14 respectively. Figure 9 shows an answer sheet that displays all of the windows used to construct the network in the system.

The experimenter measures **generating time** and **searching time**. The former is the time from when a sender reads a mission to when the sender sends his/her utterance. The latter is the time from when a sender sends his/her utterance to when the receiver finishes marking. The experimenter evaluates the following three items.

- Success and failure of the mission: If all marks in an answer sheet correspond to all attention areas in the problem, the evaluation is "success."
- Adequacy of expression: If an expression made by a sender contains multiple meaning, the evaluation is "ambiguous."
- Adequacy of interpretation: If all marks in an answer sheet correspond to all attention areas in the utterance, the evaluation is "adequate." Note that this evaluation ignores the correspondence between the utterance and the assignment.



(a) For proposed method



(b) For traditional method

Fig. 8. Missions based on No.5 for all senders

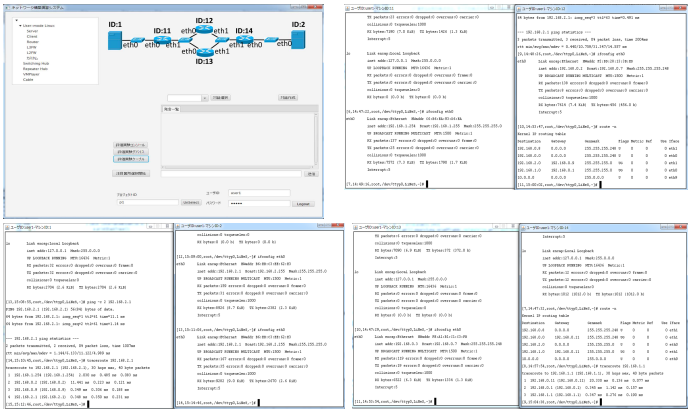


Fig. 9. Example of answersheet

B. Consideration

Figures 10 and 11 show the generating time and searching time, respectively. All the times of the proposed method are shorter than those of the traditional method. We found that senders spend more time to express attention areas using natural language, and receivers spend more time to interpret attention areas that are expressed using natural language.

Table III shows the average scores (five-point scale) of the following questions. Q1 and Q2 are used for senders, and Q3 and Q4 are used for receivers. We found that the proposed method is effective to convey attention areas because all the scores are close to 5.

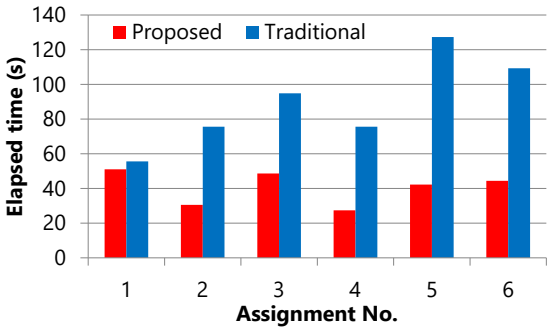


Fig. 10. Generating time

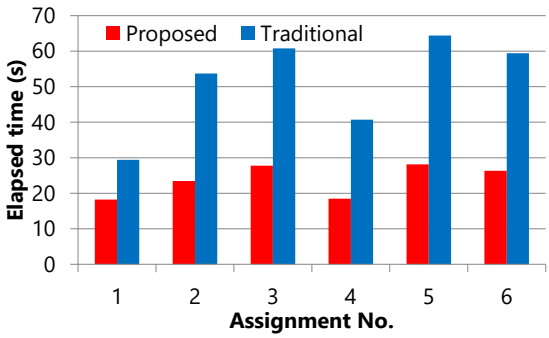


Fig. 11. Searching time

TABLE III
AVERAGE SCORES OF QUESTIONS

		Assignment No.					
		1	2	3	4	5	6
Ques. No.	Q1	4.6	4.3	4.5	4.4	4.7	4.7
	Q2	4.6	5.0	5.0	4.9	5.0	5.0
	Q3	4.9	5.0	5.0	4.9	5.0	5.0
	Q4	4.8	5.0	4.8	4.9	5.0	4.9

- Q1. How easily one can operate the system to generate utterances with the proposed method?
- Q2. How easily one can express attention areas with the proposed method when compared to the traditional method?
- Q3. How easily one can operate the system to search for utterances with the proposed method?
- Q4. How easily one can interpret attention areas that are expressed with the proposed method when compared to the traditional method?

Figure 12 shows the success rates (N=60) of conveying attention areas. The breakdown of the failure cases is as follows.

- Eighteen cases were caused by ambiguous expressions of senders; someone might obtain the correct information from the expressions. Although the receivers interpreted the expressions correctly, they were unable to obtain the correct information.
- Five cases were caused by incorrect expressions of senders. No one can obtain the correct information from the expressions using the correct interpretation.
- Six cases were caused by incorrect interpretation of receivers. We can obtain the correct information from the expressions only through the correct interpretation.

A failure case of conveyance caused by ambiguous expressions for attention areas is explained with the help of Figure 13. The assignment specifies area α as an attention area. The sender's expression is "the fourth line from the top in the output of command 'route -n'." Area β is the receiver's interpretation to the expression.

According to the experiments, we found that the senders could convey attention areas to the receivers more easily, faster, and more accurately with the proposed method than with the traditional method. Especially, it can be said that the proposed method is effective for conveying a part of command output quickly (Table II, Figures 10 and 11). We consider that it is hard to designate such areas by words, because command output consist of multiple lines and each line consists of multiple words.

Fig. 12. Result of conveyance

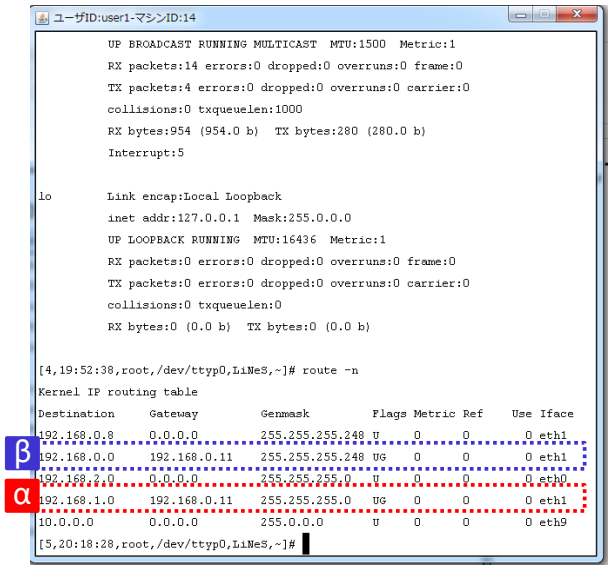


Fig. 13. Failure case of conveyance

VIII. CONCLUSION

In this paper, we presented a distributed pair exercise system for network construction with a dialogue support function. The members of a pair can edit networks consisting of virtual machines on a server machine through the Internet. They can select attention areas using a network editor intuitively and embed attention areas in utterances of the chat with hyperlinks. When clicking a hyperlink, they can observe the attention areas on a network editor. The experiments confirmed that the subjects conveyed attention areas more easily, faster, and more accurately with the proposed method than with plain text (traditional method). In the future, we will evaluate how efficiently the proposed method conveys attention areas in dialogues, not in utterances. This is because we reckon that the contexts of a dialogue have a positive effect for conveyance of attention areas, which will help improve the effectiveness of the traditional method.

ACKNOWLEDGMENT

This study was partially funded by the Grants-in-Aid for Scientific Research Foundation (17K01122) and public interest TATEMATSU foundation.

REFERENCES

[1] Chaknarin Kongcharoen, Wu-Yuin Hwang, "A study of pairprogramming configuration for learning computer networks," UMEDIA, pp. 369–375, 2015.
[2] Charles Border, "The Development and Deployment of a Multi-User,Remote Access Virtualization System for Networking,Security, and System Administration Classes," Proc. of SIGCSE'07, ACM, pp.576–580, 2007.
[3] Yuki Kitazawa, Youji Ochi, Shoji Mizobuchi, and Nobukazu Iguchi, "Investigation of IP Network Construction Learning System for Cooperative Practice using Cloud Computing (in japanese)," IEICE Technical Report, vol.112, no.66, pp.19–24, 2012.
[4] Nobukazu Iguchi, "Virtual IP Network Practice System with Software Agent," Complex, Intelligent, and Software Intensive Systems, Springer International Publishing, pp.711–720, 2018.

[5] M. Wannous, H. Nakano, "NVLab, a Networking Virtual Web-Based Laboratory that Implements Virtualization and Virtual Network Computing Technologies," IEEE Transactions on Learning Technologies, vol. 3, no. 2, pp. 129–138, 2010.

[6] Wayne C. Summers, Bhagyavati, and Carlos Martin, "Using a Virtual Lab to Teach an Online Information Assurance Program," Proceedings of the 2nd annual conference on Information security curriculum development, ACM, pp.84–87, 2005.

[7] Hilmi A. Lahoud and Xin Tang, "Information security labs in IDS/IPS for distance education," Proceedings of the 7th Conference on Information Technology Education, ACM, pp.47–52, 2006.

[8] Vu Nguyen, Hai Hoang Dang, Nguyen- Kha Do, and Dan- Thu Tran, "Enhancing team collaboration through integrating social interactions in a Web-based development environment," Computer Applications in Engineering Education, vol. 24, no. 4, pp. 529–545, 2016.

[9] Kristy Elizabeth Boyer, August A. Dwight, R. Taylor Fondren, Mladen A. Vouk, and James C. Lester, "A development environment for distributed synchronous collaborative programming," Proc. of the 13th annual conference on Innovation and technology in computer science education, pp.158–162, 2008.

[10] Despina Tsompanoudi, Maya Satratzemi, and Stelios Xinogalos, "Exploring the effects of collaboration scripts embedded in a distributed pair programming system," Proceedings of the 18th ACM conference on Innovation and technology in computer science education, pp.225–230, 2013.

[11] Stephan Salinger, Christopher Oezbek, Karl Beecher, and Julia Schenk, "Saros: an eclipse plug-in for distributed party programming," Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering, pp.48–55, 2010.

[12] Chih-Wei Ho, Somik Raha, Edward Gehringer, Laurie Williams, "Sangam: a distributed pair programming plug-in for Eclipse," Proceedings of the 2004 OOPSLA workshop on eclipse technology eXchange, pp.73–77, 2004.

[13] The User-mode Linux Kernel Home Page, http://user-mode-linux.sourceforge.net/.