DEVELOPMENT AND EVALUATION OF A CONTACT CENTER APPLICATION SYSTEM TO INTEGRATE MULTIPLE COMMUNICATION CHANNELS USING WEBRTC

Yusuke Shiga[1]and KazumasaTakami[2]

[1]Department of Information Systems Science, Faculty of Engineering,
Soka University, Tokyo, Japan
sugar_maestro@yahoo.co.jp
[2]Department of Information Systems Science, Faculty of Science and Engineering,
Soka University, Tokyo, Japan
k_takami@t.soka.ac.jp

ABSTRACT

WebRTC allows P2P communication between Web browsers. It has been attracting an interest in recent years and is beginning to be used in a wide range of fields. Progress in Internet technology is expected to diversify the means of communication between enterprises and customers from simple telephone calls and email to include easier and more convenient means, such as video calls and Web chats. We have developed an experimental application system that uses WebRTC to integrate a variety of task-specific communication tools, such as telephones, at a contact center with the aim of improving work efficiency there. Main functions implemented in this system include audio/video communication that involves an agreement procedure, setting up of FIFO-based inquiry channels, and visualization of access line congestion state. We have created test scenarios that simulated contact center tasks. Using these scenarios, we compared the experimental system and an existing system in terms of the response time, the degree of functional integration of tools, and usability, which is based on a system usability scale (SUS).

KEYWORDS

WebRCT, Contact Center,P2P Communication, Web Application, Usability, Communication Channel

1. INTRODUCTION

In recent years, there has been an interest in WebRTC (Web Real-Time Communication), a technology that allows a P2P (peer-to-peer) connection to be set up between Web browsers [1]. WebRTC has been released by Google as open source software. The World Wide Web Consortium (W3C) has standardized a WebRTC API for browsers [2] and the Internet Engineering Task Force (IETF) has standardized a related communication protocol [3]. WebRTC makes it possible to replace video chats between terminals implemented via servers using dedicated application programs with P2P connections between Web browsers. This new form of communication is expected to be used for a variety of purposes in the industry.

An example of a system that uses WebRTC is a browser-based system that remotely supports elderly people unaccustomed to the Internet in conducting electronic commerce (EC) [4]. The Web browsers of the inexperienced EC customer and the supporter exchange URLs and screen shots using WebRTC-enabled P2P communication so that they display the same image. The video captured by a camera and the voice picked up by a microphone on each side are also exchanged using P2P communication so that the supporter can recognize whom the other side is

DOI: 10.5121/ijcnc.2017.9301

communicating with and can provide detailed support information using audio and video communication. In addition, using HTML5 Canvas, the supporter can draw an image that indicates which part of the browser screen the customer should pay attention to. This drawing is synchronized on the customer's browser. In this way, the supporter can assist the EC customer effectively.

Although progress in Internet technology in recent years has been changing the way consumers communicate with each other, there have not been significant changes in the way consumers communicate with enterprises. Existing contact centers typically communicate with customers using phone calls and email. PCs are used for sending email. PCs are also used for recoding logs of interactions with customers besides sending emails and visiting Web sites. There is a demand for simplifying these tasks and improving work efficiency by integrating various uses of PCs. In addition, now that the widespread use of smartphones and SNSs has made digital communication the main mode of communication, particularly for younger consumers, it is desirable that diverse means of communication can be used for interactions with customers.

This paper proposes a WebRTC-based application system that integrates a variety of communication channels, including telephony, used at a contact center in order to improve work efficiency there. Section 2 describes tasks performed at a contact center and related studies. Section 3 presents our proposed application system and issues that must be addressed to implement the system. Section 4 proposes solutions to these issues. Section 5 describes an experimental system that implements the proposed application. Section 6 presents the method used to evaluate the proposed system. Section 7 reports on evaluation results. Section 8 summarizes this paper and presents future issues.

2.TASKS PERFORMED AT A CONTACT CENTER AND RELATED STUDIES

This section describes the current trend in contact center tasks, and WebRTC technology, which is central to the application proposed in this paper.

2.1. Means of Communication for Access to A Contact Center

Although progress in Internet technology in recent years has been changing the way consumers communicate with each other, there have not been significant changes in the way consumers communicate with enterprises. The main means of communication used for the latter are still telephony (86.1%), Web sites (79.8%) and email (51.0%) [5]. However, there is a demand for using digital means for communication with enterprises, such as Web chats (degree of expectation: 36.9%) and video calls (43.1%). This demand comes, in particular, from younger consumers, who are accustomed to digital communication thanks to their use of smartphones and SNSs [5]. While expectation for using the conventional means, such as telephony, Web sites and email, is lower, it is still significant. Therefore, enterprises are required to handle a wide variety of communication means [5].

Since younger Internet users are already using various types of applications, they are more or less accustomed to operations of Internet applications. However, in cases where the user is interacting with a single person, such as an operator at a contact center, it is confusing and troublesome to launch a number of applications and tools and to set the email address of the person with whom he or she is communicating. It is necessary to solve this problem by integrating these communication channels to a contact center and providing a unified user interface (UI), thereby eliminating the need to perform complicated operations on individual applications and tools.

2.2. WebRTC and Issues to be Solved for Applying It to Contact Center Tasks

WebRTC is an open framework that makes interactive and real-time communication possible using UDP/IP on a Web browser without requiring any plugin to be added to the browser [6]. WebRTC is supported by three Web browsers: Chrome, Firefox, and Opera. However, WebRTC may not always work even on these browsers depending on the OS used. For example, it does not work on Chrome running on iOS. WebRTC makes possible P2P communication of stream data captured by the camera/microphone of a terminal, text data and binary data [7, 8].

Using WebRTC, anyone can create a Web chat that runs on a browser simply by writing a dozen or so JavaScript/HTML lines and implementing a Web server that supports SSL. However, the following issues must be addressed if WebRTC is to be used on a browser at a contact center.

(1) Implementation of telephony functions
It is necessary to design a user interface (UI) for telephony functions at a contact center, such as call transfer, call holding, and dialing to a specific customer. This UI design must be implemented on the proposed application system.

(2) Implementation of telephone lines for use by customers for inquiries
In a call center, a private branch exchange (PBX) is used to enable multiple lines to be accessed with a single phone number. This ability must be implemented on the proposed application system.

(3) Operation logs
It is necessary to keep logs of tasks performed. Such logs may be used as evidence when dealing with complaints from customers, or as a data mining resource for improving quality of service. WebRTC ensures data confidentiality because it allows P2P communication of data to take place without passing through any server. Data are stored only in the client terminal unless deliberately designed otherwise. To keep logs, it is necessary to implement an ability to select data appropriate for the logs from among data exchanged using P2P communication and to store them in a database server.

3. Using WebRTC to Integrate Multiple Communication Channels

This section proposes an application system that uses WebRTC to integrate communication channels at a contact center in order to improve work efficiency. Specifically, as shown in Fig. 1, various communication functions at a call center, such as making phone calls, exchanging text data using email and clouds, file sharing, logging of interaction history, which is normally performed using a dedicated tool, are integrated into a single application system. It is necessary to address the following issues to implement this integration.

(1) Identifying application functions for handling various communication channels
We need to identify required functions related to communication channels used at a contact center, and to develop a use case for the experimental application system to be implemented.

(2) Developing an experimental application system
We will select an API for implementing WebRTC, a development language and a development environment, and develop an experimental application system in accordance with the functional requirements identified above.

(3) Enhancing usability

An application system that serves as a contact point with customers must satisfy a certain level of usability. We will design the experimental application system in such a way as to enhance its usability.

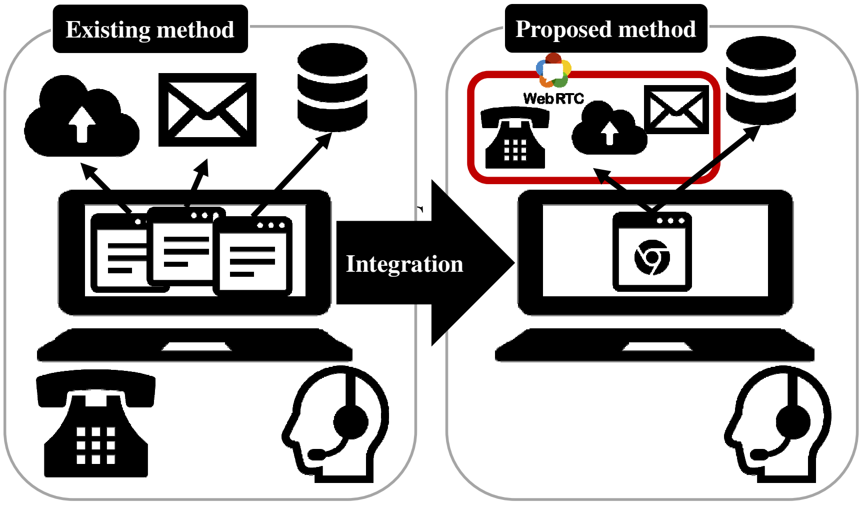


Figure 1. Use of WebRTC to integrate various functions

4. Proposed Application System

4.1. Functions of the Operator and the Customer and the Basic Business Flow

On the security principle of not giving the user more rights than are needed, we will limit the functions available to each type of user: operator and customer. Accordingly, the following conditions are imposed on two types of user:

Operator should:

(a) be able to check the number of customers in the waiting list and their sequences of arrival constantly.
(b) be able to initiate sending of a request for permission to communicate with a customer by clicking his/her name in a list.
(c) be able to transfer calls between operators (and operators alone) and to make multi-party calls.

Customer should:

(a) be able to check the number of customers in the waiting list and their sequences of arrival constantly.
(b) input his/her name, email address, and phone number when accessing a contact center.
(c) cannot initiate a call to an operator but wait for a call from an operator.
(d) do not start communication with an operator before explicitly agreeing to communicate in response to a communication request from the operator.

The application system needs to accept inquiries from customers by telephone. For this purpose, two different Web pages are provided: one accessed by the customer when making an inquiry and the other accessed by the operator when responding to a customer.

The basic flow of this application system is shown as an activity diagram in Fig. 2.First, the customer accesses the URL for inquiries, and waits for a response from an operator. The operator, who has already accessed a dedicated URL, starts communication with the customer located at the top of the waiting list and uses one of the communication channels provided by the application system to respond to the customer's request. When the request has been resolved, the customer disconnects the call and leaves the Web page. This completes the response flow for one customer.

Conventionally, different tools are required to handle different tasks, such as exchanging text/file data and sharing screen images. The proposed application system can handle all the tasks.

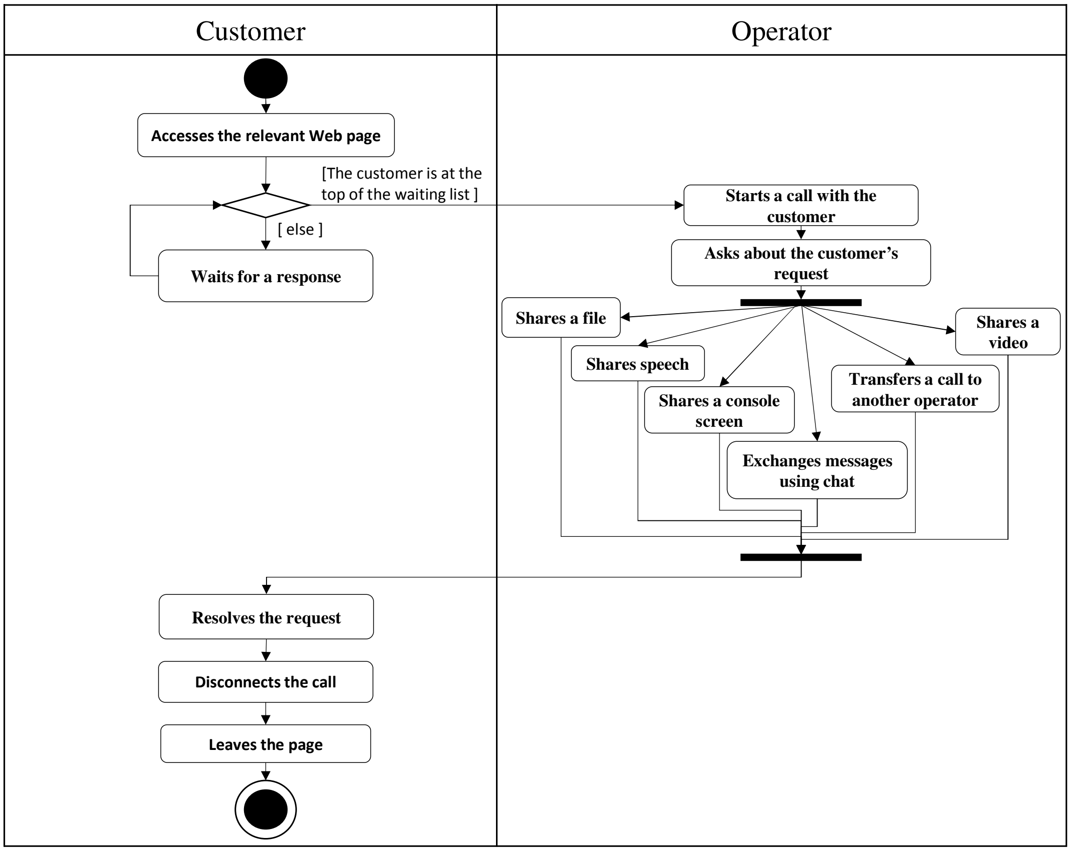

Customer
Operator
Accesses the relevant Web page
[The customer is at the top of the waiting list]
[else]
Waits for a response
Starts a call with the customer
Asks about the customer's request
Shares a file
Shares speech
Shares a console screen
Exchanges messages using chat
Transfers a call to another operator
Shares a video
Resolves the request
Disconnects the call
Leaves the page

Figure 2. Activity diagram

4.2. Functional Requirements and a Use Case

This section identifies the requirements for handling communication channels. The following functions are required to handle the tasks identified in Sec. 4.1.

4.2.1. Essential Functions

(1) Audio/video communication.
Before the operator starts WebRTC-based *MediaConnection* with the customer, he/she needs to check that the customer agrees to that communication. This is done as follows. The operator clicks a customer name in the customer list assigned to that operator. This generates a *DataConnection*, and a communication request is sent to the customer to check whether the

customer agrees to communicate. On receiving this request, the customer client indicates the request using a pop-up window in the application system. When the customer agrees, the two parties begin to talk to each other. The flow of these interactions between the customer and the operator is shown in Fig. 3. Some functions supplementary to this audio communication are provided to the operator: a three-party call, in which the operator can talk with another operator regardless of whether he/she is in communication with the customer, call holding, and call transfer. In addition, by asking the customer to enter his/her contact information in advance, the operator can re-contact the customer when the telephone connection in use is unexpectedly disconnected.

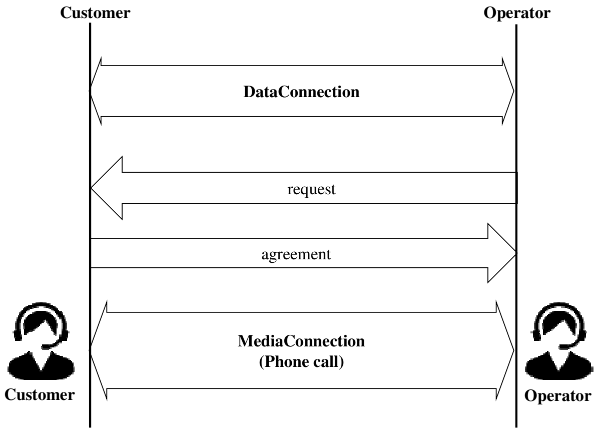


Figure 3. Communication flow including the agreement confirmation step

(2) Establishment of inquiry lines

Establishment of inquiry lines is based on FIFO as shown in Fig. 4. The customers are listed in the order in which they accessed the contact center. Operators respond to customers in the order on this list. This list is shown in the customer list window and is updated and redrawn on the window when a new customer accesses the center, when any customer in the list experiences a change in communication status, and when any customer in the list leaves the application system. Since customers can see the customer list, they can check how congested inquiry lines are.

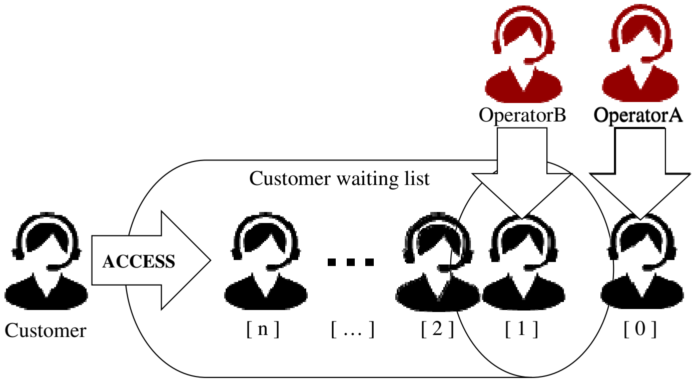


Figure 4. Customer waiting list and flow of responses to customers

(3) Automatic recording of operation logs

In order to reduce the workload of operators, a UI is provided to record operation logs either automatically or in an easy-to-use manner. There are two types of log: an activity history log, which records timestamps of call initiation and end, and a data history log, which records text data generated by speech recognition.

(4) Chat and file sharing

The operator should be able to send character strings to the customer while speaking with that customer so that the customer can easily receive information that is difficult to convey verbally. In addition, the operator should be able to send image files and document files so that he/she can share, with the customer, large documents or visual information that are difficult to convey verbally.

4.2.2. Optional Functions

(1) Screen sharing

The operator shares the console screen with the customer so that the customer can easily understand operational instructions given by the operator. The operator can give appropriate instructions because he/she can see operations executed by the customer.

(2) Visualization of the access line congestion state

The application system provides lines provided by conventional contact centers using a PBX. If a customer finds that his/her access line is busy but does not know how long he/she has to wait before a line becomes available, he/she may become frustrated. A way to reduce such stress is to visually present the congestion state of the center to the customer. Specifically, the application system discloses the number of customers in the waiting list and the particular customer's order in the list, as shown in Fig. 5. Operators respond to customers in the order in the waiting list. The customer can see his/her order in the waiting list anytime.

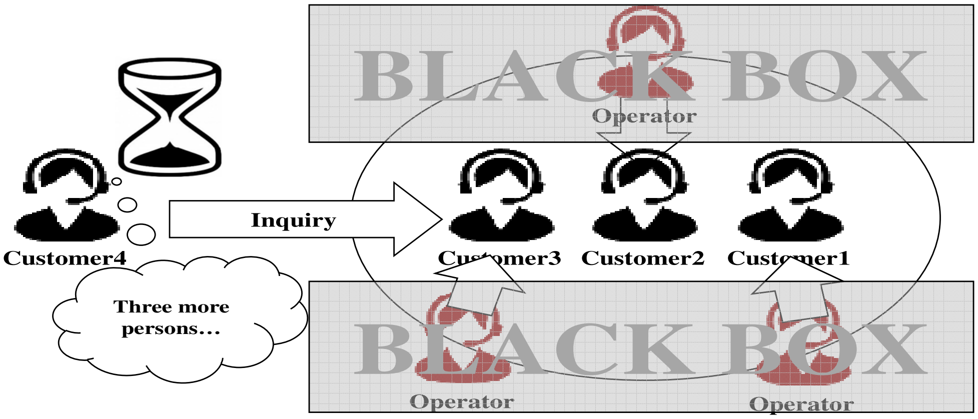


Figure 5. Visualization of the access line congestion state

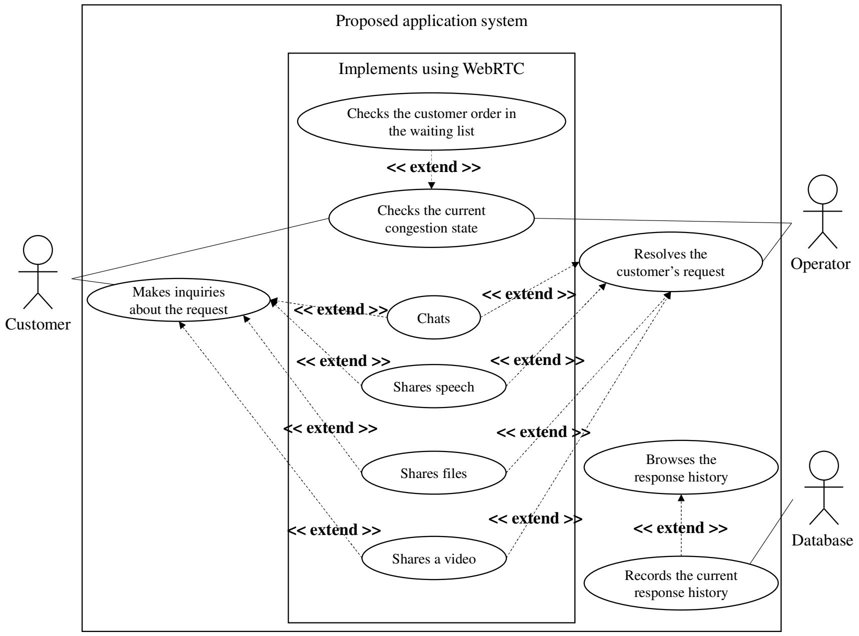

Proposed application system
Implements using WebRTC
Checks the customer order in the waiting list
<< extend >>
Checks the current congestion state
Resolves the customer's request
Makes inquiries about the request
Chats
Shares speech
Shares files
Shares a video
Browses the response history
Records the current response history
Customer
Operator
Database

Figure 6. Use case of the proposed application system

A use case of an experimental application system that is equipped with the above-mentioned functions is shown in Fig. 6.

5. Development of an Experimental System of the Proposed Application System

5.1. System Configuration

The configuration of the experimental application system developed is shown in Fig. 7. The Web server was implemented on *heroku*, which is a cloud application platform [9]. The server.js is an application execution file written in Node.js[10]. It manages page resources (HTML/JS/CSS) using a *connect* or *http* module, operates as a simple HTTP server, and accesses the database for the operation log using a *pg* (postgreSQL) module. Main program files in the experimental system are summarized in Table 1.

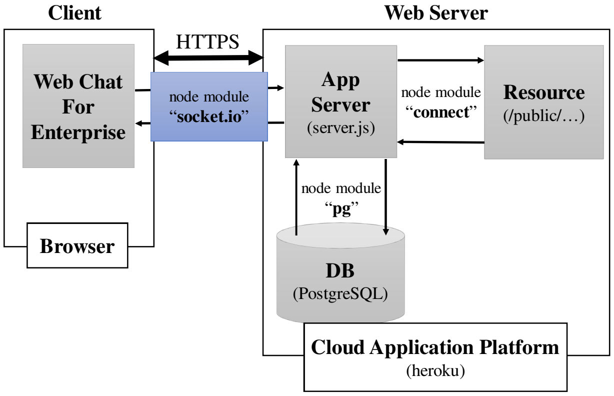


Figure 7. Configuration of the experimental system

Table 1. Main program files in the experimental system.

Application name	Web Chat For Enterprise		
System environment	Windows, Google Chrome Version 50 or higher		
Application type	Node.js application		
Written program files	Static resources	index.html	56 lines
		operator_.html	62 lines
		video.css	821 lines
		videochat.js	2965 lines
	Node.js application execution file	server.js	153 lines
Third-party plugin files	jQuery		jquery-1.12.2.min.js
	Speech recognition API*[1]	Resampler	libspeexdsp.js
			resampler.min.js
		MessagePack library	msgpack.codec.js
		Opus encoder	libopus.js
		Opus worker (worker thread to be run with WebWorkers)	libopus.worker.js
		Main unit of the speech recognitionclient library	speechrec.min.js
	Screen sharing API*[2]		screenshare.js
			screenshare.min.js
	Simple peer-to-peer with WebRTC*[3]		peer.js

(Note) SkyWay [11] is used as a WebRTC platform
*1: https://github.com/nttcom/SkyWay-SpeechRec
*2: https://github.com/nttcom/SkyWay-ScreenShare
*3: https://github.com/nttcom/peerjs

The HTML file, index.html, is a page for customers, and the HTML file, operator_.html, is a page for operators. All the processing needed for the operation of the application system is written in one file called videochat.js. All style sheets are put into one file called video.css. Both the index.html file and the operator_.html file read the same script file and the same style sheet file. The videochat.js file checks whether a particular client is accessing the page for the customer or the page for the operator, and operates accordingly.

As shown in Table 1, several plugin files provided by SkyWay [11] are used. SkyWay provides not only an SDK called PeerJS API, which is designed to facilitate implementation of P2P stream data communication and text binary communication using WebRTC, but also APIs for extended functions, such as TURN (traversal using relay around NAT), speech recognition, multi-party connections and screen sharing. In this way, SkyWay makes it easy to implement each function by providing an API for each of them.

5.2. Main Functions Implemented

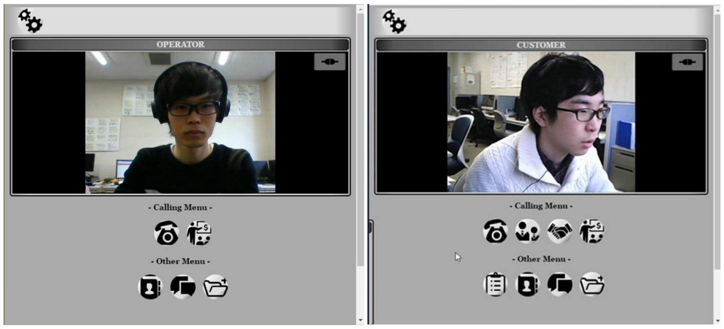
Main functions implemented in the experimental application system are shown in Table 2. An operator screen and a customer screen are shown in Fig. 8.By experimenting with a P2P communication between a customer and an operator, and a three-way communication among a customer and two operators (the operator who transfers a call and another who receives the transferred call), we confirmed that the implemented functions operated correctly. Details of the evaluation environment and conditions are described in Section 6. In the proposed system, the capacity of the Web application resources that a Web browser downloads from a Web server was very small, and thus did not affect the performance of communication with the Web server. Once downloaded, the functions in Web applications were executed by JavaScript codes on the browser. Therefore, the communication between a customer and an operator of the contact center was carried out on a P2P connection between the WebRTCs on the browsers of the two persons. The operator's PC was able to handle the processing load of this communication.

Table 2. Main functions implemented in the experimental application system.

Function name	Functional description	Icon	Operator	Customer
Audio/video call	Real-time audio/video call using a microphone and a camera	☎	Used	Used
Call with another operator	A video call simultaneously with one operator and one customer	👥	Used	
Call transfer	Transfer of a call with a customer to another operator	🤝	Used	
Call holding	Temporary holding of an ongoing call	◂● ▸	Used	
Checking of the line congestion state	Display and management of the IDs and names of the customers in the waiting list	📇	Used	Used
Message exchange	Input, transmission and reception of text data	💬	Used	Used
File sharing	Sharing of an upload area assigned to each account	📂	Used	Used
Desktop screen sharing*	Transmission of a desktop screen or a window	🧑‍🏫	Used	Used
Automatic storage of call logs	Automatic storage of the activity log and the history of message exchanges that contain details of interactions with customers in the database	📋	Used	

(Note) *: If the user wants to use the desktop screen sharing function, he/she needs to install in advance a Google Chrome Extension file that contains a Manifest file that was created separately specifically for this experiment

.



(a) Customer screen (b) Operator screen

Figure 8. Screens of the experimental system

6. Evaluation Method

The overall purpose of this evaluation is to examine whether the experimental application system integrates contact center functions with a high level of practicality. We have evaluated the application system from two aspects: work efficiency achieved by integrating communication channels, and usability.

6.1. Evaluation of Work Efficiency

Scenario1: The customer applies for joining a research group

The customer wants to join a research group. He has some questions and wants to know the necessary procedure. Therefore, he contacts the call center. An operator responds and answers his questions. When he asks about the procedure, the operator sends an electronic application form and asks him to fill in the necessary items in the form. The customer fills in the form right then and there following the instructions given by the operator, and sends back the form. This completes the application procedure.

Test purpose

We compared the work efficiency of a conventional call center (Pattern A) and that of a call center that employs the experimental application system (Pattern B).

Pattern A
The customer uses a mobile phone and a PC (for Google mail). The flow of tasks in this pattern is shown in Fig. 9.

Pattern B
The experimental application system is used. The flow of tasks in this pattern is shown in Fig. 10.

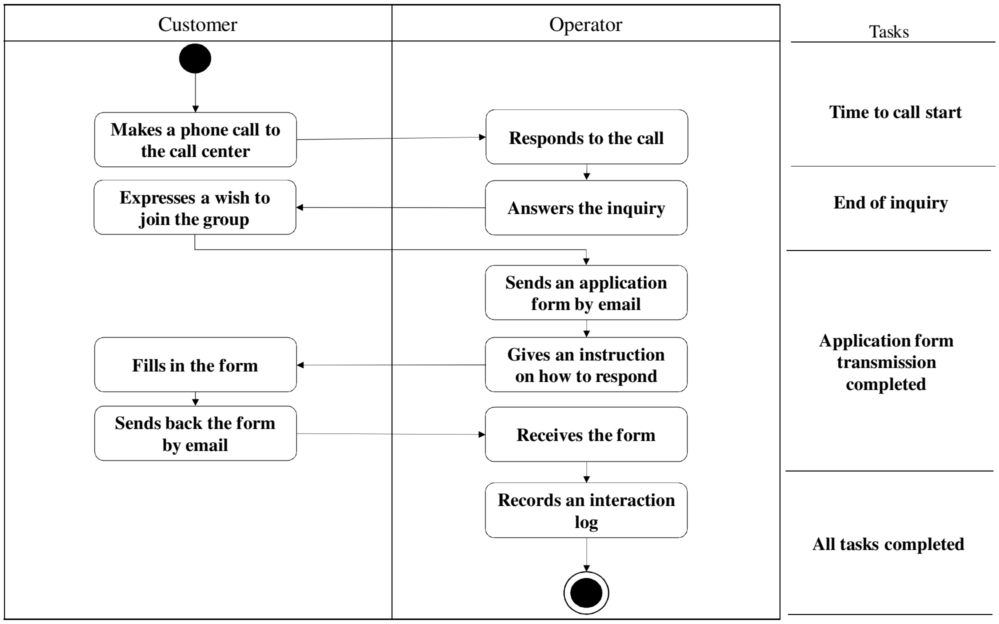

Customer
Operator
Tasks
Makes a phone call to the call center
Responds to the call
Time to call start
Expresses a wish to join the group
Answers the inquiry
End of inquiry
Sends an application form by email
Fills in the form
Gives an instruction on how to respond
Application form transmission completed
Sends back the form by email
Receives the form
Records an interaction log
All tasks completed

Figure 9. Flow of tasks of Scenario 1 in Pattern A

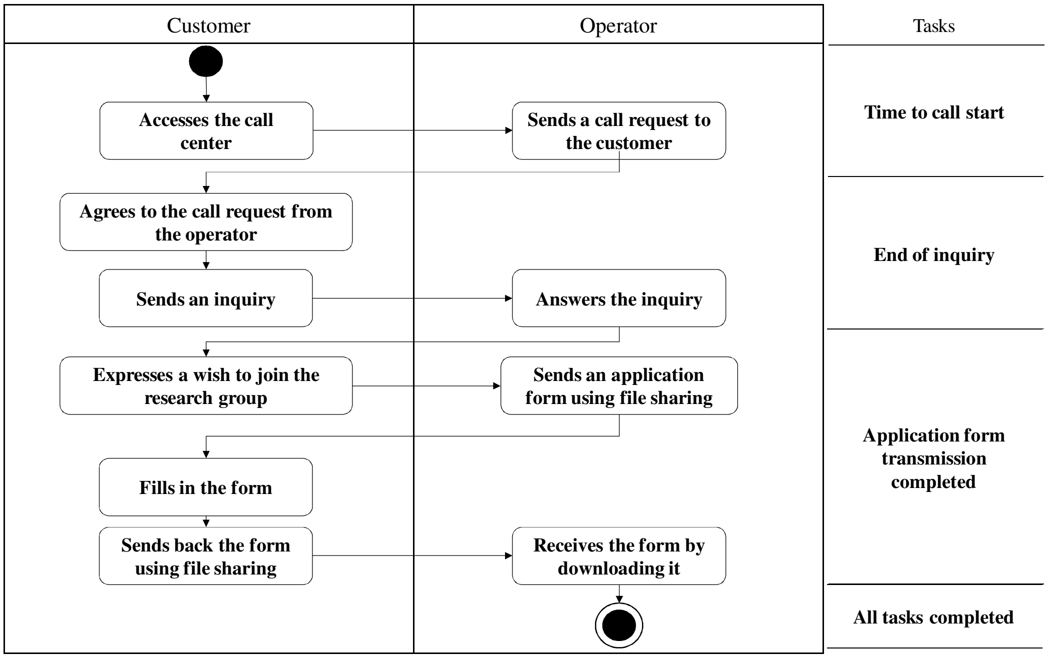

Customer
Operator
Tasks
Accesses the call center
Sends a call request to the customer
Time to call start
Agrees to the call request from the operator
Sends an inquiry
Answers the inquiry
End of inquiry
Expresses a wish to join the research group
Sends an application form using file sharing
Fills in the form
Application form transmission completed
Sends back the form using file sharing
Receives the form by downloading it
All tasks completed

Figure 10. Flow of tasks of Scenario 1 in Pattern B

Test method

The subjects shown in Table 3 executed the two patterns in Scenario 1. The time it took to complete all the tasks in Pattern A and that in Pattern B were compared to see how much work efficiency improved when the interactions between the customer and the operator were switched from Pattern 1 to Pattern 2. In addition, the subjects were asked to respond to a questionnaire survey that focused on usability.

Table 3. Subjects in Scenario 1.

Item Role	Age	Sex	Frequency of using a PC	PC specification (OS / memory / CPU)
Customer	22	Male	Uses a PC daily	Windows8.1 / 8GB / Intel Core i5
Operator	22	Male	Uses a PC daily	Windows7 / 8GB / Intel Core i3

6.2. Evaluation of Usability

Scenario2: The customer accesses a contact center regarding server maintenance

The customer notices that the lighting states of the LEDs of the server bought from a vendor are unusual, and contacts the maintenance desk to find out what he should do. He first accesses the URL (Web chat for enterprise) of the maintenance desk, and starts a call with Operator A, who is in charge of discovering what the inquiry is about. The customer tells the operator that there is something wrong with the lighting states of the LEDs. Operator A determines that the inquiry is about a technical problem, and holds the call. He calls Operator B and tells him the history of the interactions with the customer. He asks Operator B to take over the inquiry. Operator A transfers the call with the customer to Operator B. After confirming that the call transfer has been completed correctly, Operator A ends the call with Operator B. Operator B starts a call with the customer and asks him to send detailed information about the problem including a video showing the lighting states of the LEDs. The customer takes a video of the lighting states of the LEDs with a camera (this video was an image printed on a sheet of paper in this test), and sends the video. Operator B identifies the fault from the positions of the lit LEDs and explains the fault to the customer. Operator B also gives information about what the customer should do. The flow of tasks in this test scenario is shown in Fig. 11.

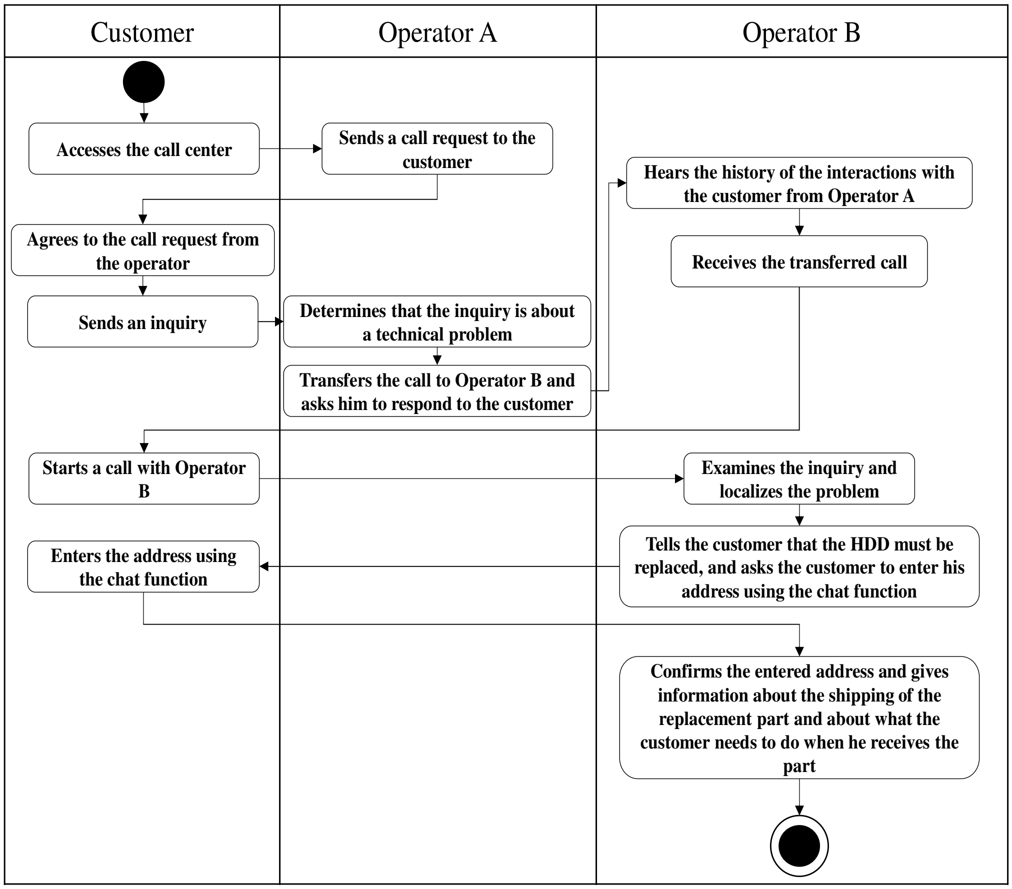

Customer
Operator A
Operator B
Accesses the call center
Sends a call request to the customer
Hears the history of the interactions with the customer from Operator A
Agrees to the call request from the operator
Receives the transferred call
Sends an inquiry
Determines that the inquiry is about a technical problem
Transfers the call to Operator B and asks him to respond to the customer
Starts a call with Operator B
Examines the inquiry and localizes the problem
Enters the address using the chat function
Tells the customer that the HDD must be replaced, and asks the customer to enter his address using the chat function
Confirms the entered address and gives information about the shipping of the replacement part and about what the customer needs to do when he receives the part

Figure 11. Flow of tasks in Scenario 2

Test purpose

We measured the usability of the application system.

Test method

The three subjects shown in Table 4 executed Scenario 2, which involved multiple communication channels. After the test, they responded to a questionnaire survey that focused on usability.

Table 4. Subjects in Scenario 2.

Item Role	Age	Sex	Frequency of using a PC	PC specification (OS / memory / CPU)
Customer	23	Male	Uses a PC daily	Windows8 / 5GB / Intel Core i5
Operator A	23	Male	Uses a PC daily	Windows7 / 3GB / Intel Pentium
Operator B	22	Male	Uses a PC daily	Windows8 / 5GB / Intel Core i5

7. EVALUATION RESULTS

7.1. Degree of Functional Integration of the Experimental Application System

The response time and the number of tools used in Scenario 1, which compared Pattern A (conventional contact center based on telephony and PC) and Pattern B (new contact center using the experimental application system), are shown in Figs. 12 and 13.

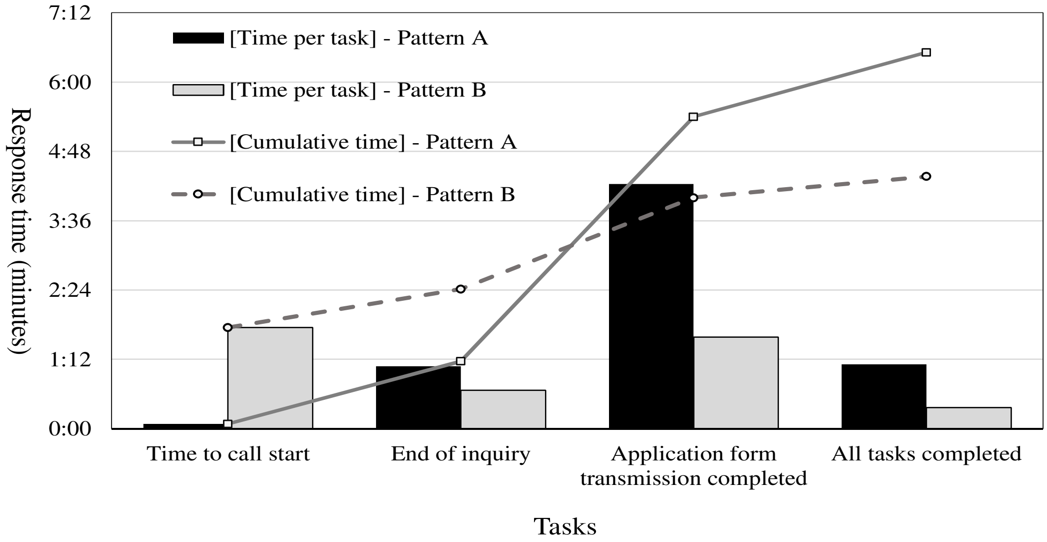


Figure 12. Relationship between response time and tasks in Patterns A and B

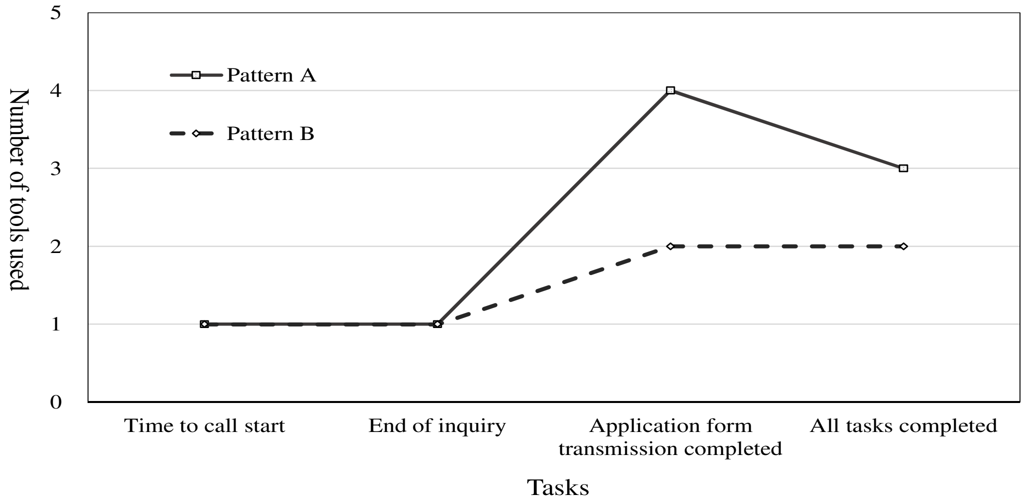


Figure 13. Relationship between the number of tools used and tasks in Patterns A and B

In Pattern B, the time to call start is long because it takes time to establish P2P communication using WebRTC and because thesubjects were unfamiliar with the operation of the experimental application system. In contrast, in Pattern A, it took only 5 seconds to start a phone call because anyone can do so without thinking. Even with this extra time needed for starting a phone call, Pattern B dramatically reduces the time needed in subsequent steps, resulting in an overall reduction of 43% of the time need for Pattern A.In Pattern A, the step of sending an application form involved simultaneous use of four tools: email, a login tool, a telephone and an application form Excel file. In Pattern B, this step is mostly covered by the experimental application system so that the number of tools involved was only two: use of the experimental application and sharing of an application form Excel file. The number of tools used never exceeds two. This result indicates that the experimental application system greatly enhances work efficiency.

7.2. Usability

The five subjects who participated in Scenarios 1 and 2 responded to a questionnaire survey, which quantifies the level of usability based on SUS (system usability scale) [12, 13]. The average SUS obtained in our test was 72. This value exceeds 68, which is said to be the normal average SUS, indicating that the level of usability of the experimental application system is sufficient. Table 5 shows the score for each question and the calculated SUS [12].

Table 5. SUS.

SUS Question*[1]	Operator in Scenario 1	Customer in Scenario 1	Operator A in Scenario 2	Operator B in Scenario 2	Customer in Scenario 2
Q1	5	3	4	4	4
Q2	4	2	2	3	3
Q3	3	4	5	3	4
Q4	2	2	1	2	3
Q5	4	4	5	5	4
Q6	2	2	1	2	2
Q7	4	5	4	4	5
Q8	2	2	1	3	1
Q9	2	3	5	3	3
Q10	4	3	2	3	1
SUS*[2]	60	70	90	65	75
Average	72				

(Note) *1: Questions defined for SUS evaluation.
*2:SUS was calculated by applying the score for each questionto the calculation method of Ref. [11].

We also asked the subjects to answer additional questions Aq1-Aq6 shown below. These questions are based on the usability items defined by Jakob Nielsen [14]. Some answers given by the subjects are shown in Tables 6 to 10.

(1)Learnability

Question Aq1: Did you think that you could operate this application system without reading the relevant manual? If you did or if you were actually able to operate it, specifically which operation do you think you can do?

Table 6. Answers to the question related to learnability.

Customer in Scenario 1	Operator B in Scenario 2	Customer in Scenario 2
Since the icon I should pay attention to is highlighted, I should be able to operate the system if the operator explains.	Since I only received transferred calls, I was able to handle most operations without reading the manual. (For, example, when I answered a call.)	I think that I can operate the system without reading the manual.

(2)Efficiency

Question Aq2: Which did you find more efficient, operation using the experimental application system or operation based on telephony and PC as at conventional call centers?

Table 7. Answers to the question related to efficiency.

Operator in Scenario 1	Customer in Scenario 1
The experimental system. Reason: It requires fewer actions, resulting in few errors. Consequently, the operation is more efficient in terms of both time and the number of actions.	The experimental system. Reason: The operation can complete using a PC alone.

For other questions, we received the following responses related to the functional aspect:

- The initial input takes time. It would be better if an autocomplete feature were available so that previously input data are stored and reused.
- I find it inconvenient that a file cannot be uploaded with a drag and drop operation.
- It would be easier to track the call state if the relevant log is displayedat the edge of the screen.
- It is better to make *Calling Menu* and *Other Menu* easier to understand. Since there is some space available under the menu, some text, such as “Call completed,” can be added there.
- It would be better to strengthen associations between the response states and icons.

(3)Memorability

Question Aq3: Do you think that, if you have used this system once, you can use it smoothly the next time?

Table 8. Answers to the question related to memorability.

Operator A in Scenario 2	Operator B in Scenario 2	Customer in Scenario 2
Yes, I think so.	I think I will have to read the manual the next time because there are functions that I have not used.	I think I can because, once I have used it, I can tell what operation can be done with which button.

(4)Errors

Question Aq4: Did any errors or problems occur while you were operating the system? If they did, what were they?

Table 9. Answers to the question related to errors.

Operator A in Scenario 2	Operator B in Scenario 2	Customer in Scenario 2
There were many cases where the customer's response stopped.	While I was talking with a customer, I became unable to hear his voice.	No, I have not experienced any errors.

(5)Satisfaction

QuestionAq5: While you were operating this system, did you find anything pleasant or interesting? If you did, what was it?

Question Aq6: While you were operating this system, did you find anything unpleasant? If you did, what was it?

Table 10. Answers to the question related to satisfaction.

Question	Operator A in Scenario 2	Operator B in Scenario 2	Customer in Scenario 2
Aq5	I found it interesting that, after the other operator participated in the call and I left the room, the call continued.	Since I did not use many functions, I did not find anything interesting.	Since there were many buttons and functions, I wanted to try them.
Aq6	None.	None	Before I transferred a call to another operator, I became a little worried that the call was still connected or the screen was frozen.

The subjects' responses expressing satisfaction included "The operation was intuitive," and "The relevant site responded fast."

8. CONCLUSIONS

This paper has proposed to use WebRTC to improve use of communication channels at a contact center. We have developed an experimental application system and confirmed that the system sufficiently improves the work efficiency at a contact center.

Conventional contact center tasks rely on telephony and PC. This application system uses WebRTC-based real-time video, audio and data connections to integrate contact center tasks. We have confirmed that the system makes it possible to replace use of a telephone with use of a Web browser. This simplifies contact center operations.We have included additional functions, such as file sharing, text chats and screen image sharing, in order to expand the range of tasks to which the system can be applied. This has enabled a single WebRTC-based application system to handle both the essential and supplementary contact center operations. It has been easy to provide services that cannot be achieved with telephony alone. This has not only improved customer satisfaction but also will reduce the operational costs because no telephones are required.

We have converted telephone-based interactions with the customer into interactions using WebRTC-based connections. Since we have used UDP-based P2P communication, communications have been unstable many times. One of the future issues is to solve this problem. It is necessary to identify causes of unexpected disconnections, and implement the application system in such a way that, should a communication is disconnected, it will be reestablished immediately. Another issue with our approach is that not all Web browsers support WebRTC. This means that any system that relies on WebRTC cannot serve all customers. It is necessary to promote activities to encourage support for WebRTC by iOS and Internet Explorer.

REFERENCES

[1] Open Project,"WebRTC,".[Online]. Available: https://webrtc.org/.[Accessed: March 1, 2017].

[2] W3C,"WebRTC 1.0: Real-time Communication Between Browsers," Working Draft 24 November 2016. [Online]. Available:https://www.w3.org/TR/webrtc/. [Accessed: March 1, 2017].

[3] IETF, "Rtcweb Status Pages,"Real-Time Communication in WEB-browsers (Active WG). [Online]. Available:https://tools.ietf.org/wg/rtcweb/. [Accessed: March 1, 2017].

[4] N. Hongo, H. Yamamoto and K. Yamazaki, "Development and Evaluation of Browser Synchronization System for EC Refugees Using WebRCT," *The IEICE Transactions*, Vol. J97-B No.10, pp. 890-902, 2014.

[5] Transcosmosinc., "Investigation on actual situation of communication between consumers and enterprises 2016". [Online]. Available:http://www.trans-cosmos.co.jp/data/2016dec/. [Accessed: March 1, 2017].

[6] Alan B. Johnston and Diniel C. Burnett,*Webrtc: APIs and Rtcweb Protocols of the Html5 Real-Time Web*,Digital Codex LLC, 2012.

[7] R.Copeland and M. Copeland,"A Question of Quality – VoIP, WebRTC or VoLTE?,"reTHINK project. [Online]. Available:https://rethink-project.eu/publications/conferences-and-journals/. [Accessed:March 1, 2017].

[8] I. Tariq Javed, K. Toumi and N. Crespi, "Browser-to-Browser Authentication and Trust Relationships for WebRTC," IARIA, The Tenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM), October 9-13,2016.

[9] HEROKU, INC. "Heroku,". [Online]. Available:https://www.heroku.com/. [Accessed:March 1, 2017].

[10] Node.js Project, "Node.js,". [Online]. Available:https://nodejs.org/en/. [Accessed: March 1, 2017].

[11] NTT CommunicationsCorporation,"SkyWay,". [Online]. Available:http://nttcom.github.io/skyway/en/index.html. [Accessed: March 1, 2017].

[12] J. Brooke,"SUS-A quick and dirty usability scale," In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland (Eds.), Usability Evaluation in Industry. London: Taylor and Francis, pp. 189-194, 1996. [Online]. Available:http://hell.meiert.org/core/pdf/sus.pdf. [Accessed:March 1, 2017].

[13] A. Muddimer, S. Camille Peres and S. McLellan, "The Effect of Experience on SystemUsability Scale Ratings," *Journal of Usability Studies*, Vol. 7, Issue 2, pp. 56-67, February 2012. [Online]. Available:http://uxpajournal.org/the-effect-of-experience-on-system-usability-scale-ratings/. [Accessed:March 1, 2017].

[14] Jakob Nielsen, *Usability Engineering (Interactive Technologies)*,Morgan Kaufmann, 1994.

Authors

Yusuke Shiga

He received a bachelor's degree from Soka University, Japan in March 2017. He was researching on the web service techniques. He joined FFRI, Inc, Japan in 2017. Since April 2017, he has been engaged in development and study of cyber security system.

KazumasaTakami

He joined Musashino Electrical Communication Laboratories, Nippon Telegraph and Telephone Corporation (NTT), Musashino, Japan in 1979. Since April 2004, he has been a professor in Department of Information Systems Science, Faculty of Science and Engineering, Soka University. He is currently interested in Mobile Ad Hoc Network, Delay/Disruption-Tolerant Network (DTN), Inter-Vehicle Communication Network and Services Software. He is a member of the IEICE, IPSJ, DBSJ, and a senior member of the IEEE.