

## DESIGN AND IMPLEMENTATION OF OPEN-CAMPUS EVENT SYSTEM WITH VOICE INTERACTION AGENT

TAKAHIRO UCHIYA

*Nagoya Institute of Technology, CREST JST, Japan*  
*t-uchiya@nitech.ac.jp*

MASAKI YOSHIDA

*Nagoya Institute of Technology, Japan*  
*yoshida@uchiya.nitech.ac.jp*

DAISUKE YAMAMOTO

*Nagoya Institute of Technology, CREST JST, Japan*  
*yamamoto.daisuke@nitech.ac.jp*

RYOTA NISHIMURA

*Keio University, Japan*  
*nishimura@ae.keio.ac.jp*

ICHI TAKUMI

*Nagoya Institute of Technology, CREST JST, Japan*  
*takumi@nitech.ac.jp*

Many universities are positively performing open-campus activities as events that provide campus information for recruitment of high school students. However, open-campus activities present some problems. We propose a system to hold an open-campus event with a spoken dialogue toolkit that functions on smartphones. This method can reduce printing costs by displaying a browser with information that would otherwise be published in the brochure. Furthermore, by guiding the course with a voice interaction agent expressed using a 3D character model, GPS, Google Maps, and walking-path estimation system, no guide person is necessary. After designing and implementing an open-campus event system using these functions, we evaluated the proposed system through experimentation.

*Key words:* Open-campus event system, Agent-based campus guide, Voice interaction agent

### 1 Introduction

In connection with a decrease in birth-rates in recent years, recruiting of students is becoming difficult at many universities. Therefore, many universities are positively performing open-campus activities providing campus information as events for student recruiting and admissions for junior high school or high school students.

The university directly presents attractive aspects of a university to visitors by carrying out open-campus activities. Additionally, the visitor can concretely experience aspects of campus life such as the university atmosphere instead of merely reading a presentation in a pamphlet. To use this benefit to the greatest degree, a technique is sought for using information technology to facilitate open-campus events.

Open-campus activities are performed in general according to the following flow.

- (1) Using the internet, visitors make reservations to participate.
- (2) Visitors gather for open-campus activities at the specified time.
- (3) Visitors are divided into specified groups.
- (4) A guide person presents the area to visitors. Visitors can see the results of several studies being conducted at the laboratory through the demonstration.
- (5) After the visit, visitors respond to a questionnaire survey.

However, the following problems exist in relation to open-campus events performed now.

P1. Costs of printing pamphlets, etc.

For open-campus events, pamphlet printing costs must be paid to distribute a pamphlet.

P2. Time and effort to prepare open-campus event personnel

Performing an open-campus event necessitates decisions related to the pamphlet contents and sufficient lead time for planning the open-campus event. Furthermore, burdens are imposed on people who manage the whole open-campus event and those who take charge of guidance on the day of the open-campus event.

P3. Time restrictions of visitors for open-campus activities

During this open-campus event, because the time and date are specified, visitors cannot visit the university except on a certain date.

Therefore, some system is necessary for the resolution of these problems. To resolve these problems related to an open campus, we propose an open-campus event system by voice interaction, which requires no person to act as a guide. In this system, the voice interaction agent plays a virtual role of a guide, and offers route guidance and guidance services at a visitation place. For this research, in addition to a voice dialogue function, the system carries out position estimation at outdoors and indoors, supports conversations among visitors, a questionnaire automatic collection, etc. to support the open-campus event completely.

The contents of this paper are the following. In chapter 2, we describe a voice interaction toolkit used in this research. In chapter 3, we explain the design and implementation of the proposed system. In chapter 4, this report describes the experiment and presents discussion of the effectiveness of proposed system. Chapter 5 presents a summary of this research.

## **2 Voice Interaction Toolkit**

### *2.1 Spoken Dialogue System*

In earlier research, to facilitate the use of a spoken dialogue system, a fully open-source toolkit for voice interaction systems called MMDAgent [1] was developed at our university (Nagoya Institute of Technology) using speech processing technology. A practical example exists: digital signage has been operated in front of the main gate of our university since 2011 [2], allowing anyone to interact with a three-dimensional (3D) character named ‘Meichan’ using a microphone, speaker, and big screen (Fig.

1). This signage provides campus information such as recent campus events, student club events, and teachers' room information.



Figure 1 Spoken dialogue system with a 3D character at the main gate of Nagoya Institute of Technology.

## 2.2 MMDAgent: Toolkit for building voice interaction systems

MMDAgent is a toolkit that includes advanced speech and graphics technology such as speech recognition, speech synthesis, 3D rendering, dialogue management, and a physics engine (Fig. 2). Voice interaction between user and system is especially useful. Therefore, in the proposed system, we use this MMDAgent.

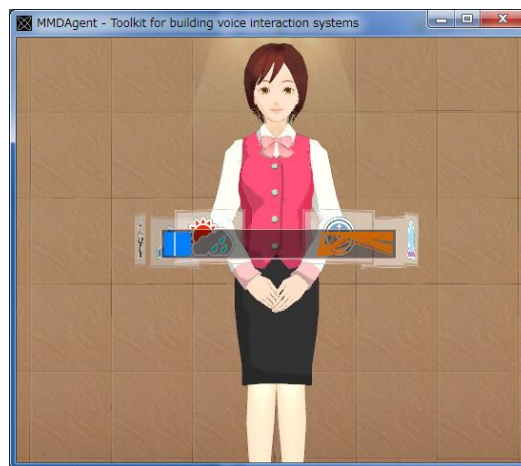


Figure 2 Interface of MMDAgent.

This system uses Julius [3] as a speech recognition engine, uses OpenJTalk [4] as a voice synthesis engine, uses the MikuMikuDance form [5] as a 3D model, and uses Bullet Physics [6] as a physical operation engine.

Julius has dictionaries of two kinds to raise the speech recognition rate. 60,000 words collected from the web are registered into the first system peculiar dictionary. In the second user dictionary, a system developer can use arbitrary words and phrases to make the words newly recognizable. By describing words and phrases to make it recognizable in this dictionary, a developer can tailor the MMDAgent to fit the environment.

MMDAgent can manage voice interaction scenarios based on the FST script. The FST script, which is based on the Finite State Transducer (FST) format as depicted in Fig. 3, can accommodate dialogue scenarios by triggering various events including speech keywords, sensor values, and timers. MMDAgent performs real-time voice recognition with a slight delay. For example, in Fig. 4, if the agent recognizes the utterance of “Hello”, the agent greets the person and says “Hello”.

```

1 10 RECOG_EVENT_STOP|Hello <eps>
1 10 RECOG_EVENT_STOP|Hi <eps>
10 11 <eps>| MOTION_ADD|mei|greet|greet.vmd
11 12 <eps> SYNTH_START|mei|normal|Hello
12 1 SYNTH_EVENT_STOP|mei <eps>
    
```

Figure 3 Sample FST script for MMDAgent.

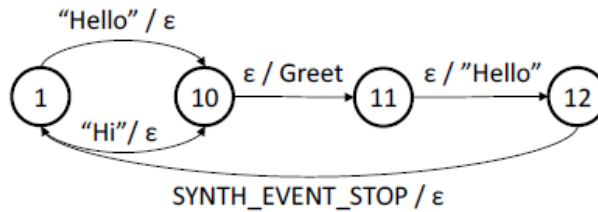


Figure 4 Sample behavior diagram of MMDAgent.

### 2.3 Smart Meichan

The MMDAgent operates only on a PC: we cannot make it operate during walking or outdoor situations. Therefore, in the proposed system, we use Smart Meichan (Fig. 5) [7,8], an MMDAgent transplanted to an Android OS smartphone. This is a standalone application that requires no servers for speech processing. Therefore, this system has high response without network delay.

Similarly to MMDAgent, Smart Meichan has a function by which we can set up the arbitrary contents of a dialogue in an fst file. This file can be rewritten arbitrarily. Therefore, we carry out open-campus events with voice interaction using Smart Meichan by describing the necessary dialogue for the campus guide in the fst file.

Figure 6 presents the Smart Meichan architecture. This system introduces the Java Bridge Module, which uses the Java Native Interface and PluginListener for Android OS. The merits of Java are its ease in developing network applications on Android OS.



Figure 5 Interface of Smart Meichan.

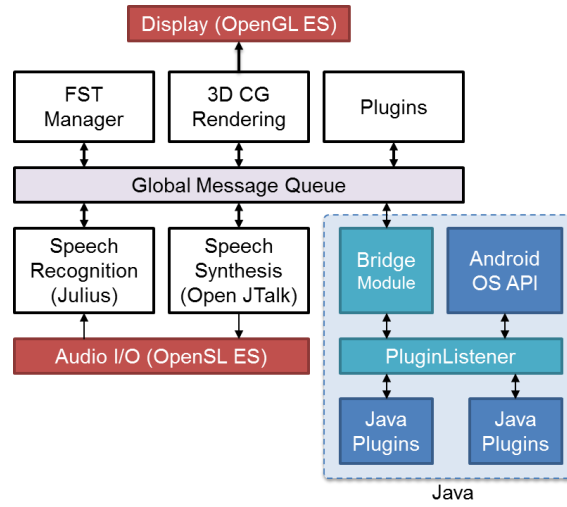


Figure 6 Architecture of Smart Meichan.

Smart Meichan has the following functions.

- Substitute fst

We can set up two or more substitute fst files in addition to the main fst file, which includes frequently used conversations.

- Re-reading of fst

Although Smart Meichan reads all the fst files at the time of starting, it can re-read arbitrary fst files during starting.

- Operation from Java source code

Smart Meichan can start an event such as inputting transition conditions from the source code, or reading out a specific character string.

Using these functions, the dialogue responses can be changed dynamically.

### 3 Design and implementation of proposed system

#### 3.1. Overview

To resolve these open-campus-related difficulties, we propose the open-campus event system using voice interaction. The proposed system shares information among smartphones using an information-sharing server (Fig. 7). Here, an information-sharing server can be used for information sharing or for log preservation.

#### 3.2. Smartphone Application (Voice Interaction Mode)

To ease the burden of the role of the guide of the conventional open-campus event, we conduct the open-campus event using Smart Meichan. As a merit of guidance by voice dialogue, the burden placed on open-campus event days can be mitigated (S2). Furthermore, the visitor can visit the university at any time (S3).

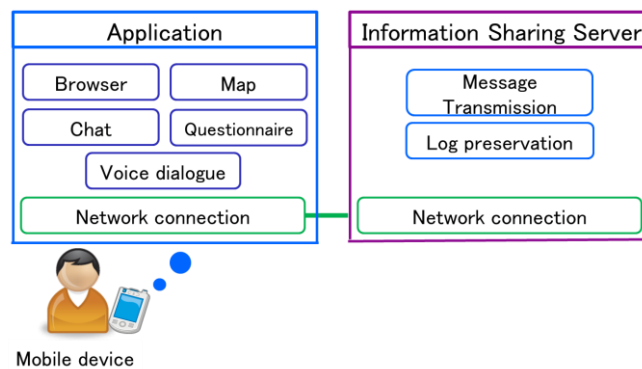


Figure 7 Overview of proposed system.

As a shortcoming of guidance by voice dialogue, some visitors might be troubled by an utterance that is not being recognized correctly. Moreover, the system cannot answer questions from a visitor flexibly because it is necessary to prepare the contents used in a voice dialogue beforehand.

This mode is equipped with the following features considering the benefits and the shortcomings above.

- Explanation and questions and answers with a sound:

Smart Meichan provides explanations at an explanation point. She responds with a sound to a question from a visitor after an explanation.

- Web-based information disclosure:

The system provides information about the university using web pages (S1). Moreover, by recording the demonstration as a demo video beforehand and by reproducing it on a smartphone, the demonstration is useful during bad weather and can realize cost reductions. Furthermore, the system changes web pages automatically according to the Smart Meichan's guidance, thereby easing a visitor's burden (Fig. 8 browser function).

- Routing assistance using position information:

A developer creates guide route information beforehand. The system displays it on a map, so that a visitor can pass to an area using the audio assist from Smart Meichan (Fig. 8 map function). The system uses Google Maps AndroidAPI v2 [9] as a display.

- Conversations among visitors using a chat function:

By virtue of the chat function, visitors can converse with other visitors.

Voice dialogue mode performs open-campus activities according to the following flows.

Step 1.

A visitor starts this application at the open-campus event start point.

Step 2.

The visitor says "Start" to Smart Meichan.

Step 3.

Smart Meichan performs guidance to the next explanation point like the map function of Fig. 8.

Step 4.

The visitor says "Arrival" to Smart Meichan to signal that it has arrived at the explanation location.

Step 5.

This application starts explanations at that location when Step 4 and a current position are checked and have arrived at the explanation location (Fig. 8 browser function).

Step 6.

When a question is posed by a visitor, the system encourages the visitor to ask questions. When no question exists, the system encourages visitors to say "Next" and to start route guidance to the next explanation point. Then the system repeats Steps 3–6.

Step 7.

When the guidance is completed, the system displays a questionnaire screen that asks visitors to evaluate the open-campus event.

The following two points are described as problems that occur when using a voice dialogue.

- A sound is not recognized correctly.
- Addressing the device is embarrassing.

To resolve those shortcomings presented above, we created a button that performs the same operation as the time of speaking to Smart Meichan (Fig. 9). A button is displayed on a screen by a swiping operation. The present state is arranged as a murky display at the topmost part. Thereafter, only the buttons a visitor can select for guidance operation are displayed.



Figure 8 Interface of proposed system.



Figure 9 Direction by button operation.

### 3.3. Campus guidance information

To conduct the open-campus event in voice dialogue mode, it is necessary to prepare the course and the contents of guidance beforehand. Then, we use Google Map Myplace [10] as a tool that creates guidance information. A salient benefit of using Myplace is that it can be created if a web browser and a Google account exist. Moreover, a point with the easy operation method is also obtained. Myplace's information is downloadable in keyhole markup language (kml) form. In Myplace, a user can draw arbitrary pins and lines on the Google Map. Furthermore, users can produce a title and explanation to accompany each pin and line. We use these functions and the following conditions to create campus guidance information (Fig. 10).





Figure 10 Description of campus guide information.

- Behind the title of the pin of the beginning point of the campus guide, we show "(Start)".
- Behind the title of the pin of the ending point of the campus guide, we show "(Goal)".
- When the pin of the point which begins the campus guide and the pin of the point which finishes the campus guide are the same, we show "(Start, Goal)".
- The line title is taken as "the name of the pin used as the starting point, and the name of the pin used as a terminal point".
- A user inputs the text that Smart Meichan pronounces into the explanation input area of a pin and a line. When the pin of the point which begins the campus guide, and the pin of the point which finishes the campus guide are the same, a user inputs "(Start)" at the topmost part of the text about which Smart Meichan speaks at the time of a start. Moreover, a user inputs "(Goal)" at the topmost part of the text, which Smart Meichan speaks at the end.

For more flexible guidance in addition to the conditions described above, we added the following tags to a description of the explanation of a pin and a line.

- <URL: URL to display>

A function that changes the page currently displayed on the browser under explanation.

- <separator: Word of conditions that shift to the next explanation>

The function waits for a visitor's response between explanations. A user can set several conditional words. Using the "custom swipe pin" explained below, the system adds the words above to the menu displayed after a swipe operation in voice dialogue mode.

- <wait: Time >

This function stops Smart Meichan's operation according to the time (ms).

Furthermore, we defined special pins so that all information related to a guide route can be filled in only by Myplace (Fig. 11).

- default url pin: The pin sets up the URL of the screen, which is displayed in a browser when starting the proposed system (Fig. 11, upper left).
- dictionary pin: The pin registers words and phrases to make it newly recognizable (Fig. 11, upper right).
- question pin: Pin describes the expected question and its answer (Fig. 11, lower left).
- custom swipe pin: The function to add an item to the menu displayed by the swipe operation (Fig. 11, lower right).



Figure 11 Introduction of custom pin for campus guide.

### 3.4. Information-sharing server

Using the proposed method, the system transmits the message sent from a smartphone to a suitable terminal. Because a log is saved, we use an information-sharing server. Using an information-sharing server, we realize transmission of URL for synchronizing a display page, position information, and the chat contents, which are features of the proposed system. Moreover, we disseminate information to developers by saving logs at an information-sharing server. We use a socket communication method for communication between this server and smartphones.

Information-sharing servers have the following features.

- Data sharing function: The system has a function of transmitting other messages and contents of a chat addressed to a smartphone.
- Log preservation function: Function to save data, such as questionnaire results, sent to information-sharing servers.
- Transmitting function of campus guide information: Preparation of the course and the contents of guidance must be done beforehand in voice dialogue mode. However, when this information is given in an application, it is necessary to update the application whenever it corrects campus guidance information, and to redistribute the information. Then an application acquires information from a server when it connects with an information-sharing server. This connection reduces the time and effort necessary to redistribute an application.

## 4 Experiment and evaluation

### 4.1. Experiment 1: Confirmation of effectiveness of voice interaction

#### Experiment purpose and environment

The emphasis of this experiment is effectiveness of voice interaction. We checked several items: interest in the proposed system, system usability, validity of the campus guide using voice dialogue, and the position information accuracy. Moreover, we confirm that the problems explained in Chapter 1 were resolved. We asked NIT personnel about the creation of contents and the course of the campus guide actually performed at Nagoya Institute of Technology using Myplace. An evaluation experiment was conducted using these data. To measure the accuracy of position information precisely, when a visitor arrives at the indoor destination, the system records the current position and the destination position to a server.



Figure 12 Campus guide course in experiment.

We conducted an experiment on November 28, 2013. Moreover, a mobile LTE router was used for network communication under the experiment. A campus guide course is presented in Fig. 12 to explain the campus research area and the route guide.

The evaluation method is a five-step evaluation that sets the minimum value as 1 and the peak value as 5. Moreover, comments are provided next to each item.

The subjects of this experiment were 10 university undergraduate students and graduate students. The experiment used a smartphone (XperiaZ1; Sony Corp.) and an earphone with a microphone. Moreover, outdoors, the system took position measurements using GPS. Indoors, the system took position measurements by position estimation using the smartphone's gyro sensor.

#### Experimental results and considerations

Table 1 presents results obtained from this experiment. We verified that the proposed system operated satisfactorily. We confirmed the following through experimentation.

- Cost reduction (S1)
- Reduction of the time and effort related to that day (S2)
- Time restrictions at the time of a visitor inspection were lost (S3).

Results demonstrated the effectiveness of the proposed system. Almost all questions elicited high values as responses. Many subjects evaluated this system as interesting and reported that its provided functions are useful. However, the voice recognition rate requires more improvements. We also enhanced the user interface for ease of operation.

Table 1 Result of experiment 1

Question	Average	Standard Deviation
Interest of proposed system	4.4	0.80
Superiority	3.4	0.66
Guide based on voice dialogue	3.7	0.78
Function of browser	3.6	1.36
Function of questionnaire	3.8	1.33
Touch interaction	3.6	1.22
Ease of operation	3.4	1.43
Voice recognition rate	3.0	1.55
Accuracy of position inference	3.6	1.50

#### *4.2. Experiment 2: Measurement of battery consumption*

##### Experiment purpose and environment

To ascertain whether the proposed system can be sufficient for practical use in terms of power consumption, we compared the proposed system with Google Map and my Smart Meichan. In the

proposed system, we activated Google Map, Smart Meichan, and all sensors for position estimation. The terminal we used for comparison was the Xperia Z1 (Sony Corp.), with battery size of 3000 mAh. We performed the acquisition of the battery balance using a battery-monitor3 [11]. Then we measured it during operations for 1 hr.

Experimental results and considerations

An experimentally obtained result is presented in Fig. 13. The consumed power from the battery of the proposed system was lower than the sum of the battery consumption amount of Google Map and Smart Meichan because the proposed system displays the Smart Meichan interface in a small display area. Figure 13 shows that proposed system consumes approximately 26% of the total battery power during 1 hr. Therefore, we can use this system for about 4 hr. This result shows that proposed system can be applied for practical use at an open campus event.

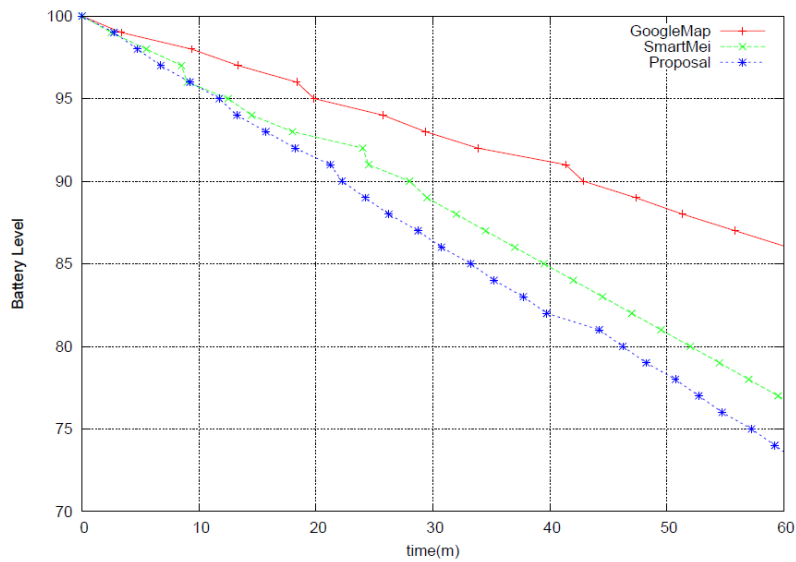


Figure 13 Result of experiment 2.

**5 Conclusion**

This paper described our proposed open-campus event system, which is expected to be useful to hold open-campus activities more efficiently. The proposed system resolves the following three problems: P1) Costs of printing pamphlets, etc.; P2) Time and effort to prepare open-campus event personnel; and P3) Time restrictions of visitors for open-campus activities.

The proposed voice interaction requires no person to act as a guide. In this system, the voice interaction agent plays the virtual role of a guide, offering route guidance and guidance services at an inspection place.

We conducted two experiments to evaluate the effectiveness of voice interaction and battery consumption. Results of the experiments demonstrate the validity of the proposed system. Future work will address improvement of the ease of operation. Useful help functions and tutorial functions are also needed.

## References

1. Lee, A., Oura, K. and Tokuda, K. MMDAgent: A Fully Open-Source Toolkit for Voice Interaction Systems. In Proc. IEEE ICASSP2013, 8382-8385, 2013.
2. Oura, K., Yamamoto, D., Takumi, I., Lee, A. and Tokuda, K. On-Campus, User-Participatable, and Voice-Interactive Digital Signage. Journal of Japanese Society for Artificial Intelligence, 28 (1), 60-67 (2013).
3. Lee, A. and Kawahara, T. Recent Development of Open-Source Speech Recognition Engine Julius. APSIPA, 2009, 131-137.
4. Open JTalk, <http://open-jtalk.sp.nitech.ac.jp/>
5. MikuMikuDance, <http://www.geocities.jp/higuchuu4/index.htm>
6. Bullet Physics, <http://bulletphysics.org/>
7. Yamamoto, D., Oura, K., Nishimura, R., Uchiya, T., Takumi I., Lee, A. and Tokuda, K. Development of a Voice Interaction Agent using 3D Modeling for Stand-alone Smartphones. IPSJ Interaction 2013, 675-680, 2013.
8. Yamamoto, D., Oura, K., Nishimura, R., Uchiya, T., Lee, A., Takumi, I. and Tokuda, K. Voice Interaction System with 3D-CG Virtual Agent for Stand-alone Smartphones. Proc. of the Second International Conference on Human Agent Interaction, ACM Digital Library, 2014, 320-330.
9. Google Maps Android API v2, <https://developers.google.com/maps/documentation/android/>
10. Google Map Myplace, <https://maps.google.co.jp/maps/myplaces>
11. Battery-monitor3, [https://play.google.com/store/apps/details?id=net.ohanasiya.android.battery\\_monitor3d](https://play.google.com/store/apps/details?id=net.ohanasiya.android.battery_monitor3d)