KICS The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express 5 (2019) 47–51

www.elsevier.com/locate/icte

MMDAE: Dialog scenario editor for MMDAgent on the web browser

Ryota Nishimura[a,*], Daisuke Yamamoto[b], Takahiro Uchiya[b], Ichi Takumi[b]

[a] *Department of Technology, Industrial and Social Science, Tokushima University, Tokushima, Japan*
[b] *Department of Computer Science, Nagoya Institute of Technology, Nagoya, Japan*

Received 7 February 2018; accepted 21 March 2018
Available online 12 April 2018

Abstract

We have developed MMDAgent (a fully open-source toolkit for voice interaction systems), which runs on a variety of platforms such as personal computers and smartphones. From this, the editing environment of the dialog scenario also needs to be operated on various platforms. So, we develop a scenario editor that is implemented on a Web browser. The purpose of this paper also includes making it easy to edit the scenario. Experiments were conducted for subjects using the proposed scenario editor. It was found that our proposed system provides better readability of a scenario and allows easier editing.

Keywords: Spoken dialog system; Scenario editor; Web browser; MMDAgent

1. Introduction

Various techniques of speech processing have recently been developed. Among these, techniques of speech recognition and speech synthesis are widely used. Spoken dialog systems (SDSs) integrating these technologies have also been developed. Pioneer of the spoken dialog system is a VOYAGER of MIT. It was developed in the early 1990s. ATIS project [1] has been carried out in the early 1990s led by DARPA in the United States. In recent years, 'GalateaToolkit' a toolkit of spoken dialog systems [2] and 'SCHEMA' robot that can interact multiplayer [3] has been developed. Commercial systems, such as Siri (Apple Inc.) [4], have appeared and gained popularity. However, SDSs are not yet widely used.

Therefore, for anyone to be able to use an SDS easily, we constructed a fully open-source toolkit for voice interaction systems (MMDAgent [5]) using speech processing technology. As a practical example, digital signage has been set up in front of the main gate of a university (Nagoya Institute of Technology) [6], allowing anyone to interact with a life-size three-dimensional (3D) character named 'Mei-chan' (Fig. 1). SDS software has been developed for the personal computer (PC; running Windows, Mac OS, or Linux), and it has also been ported to Android so as to work on any smartphone [7]. Additionally, an SDS using the video communication function of Skype (Voice over Internet Protocol) has been developed [8].

The MMDAgent toolkit includes software for speech recognition, speech synthesis, character drawing as part of 3D computer graphics and dialog management to meet the requirements of an SDS. An environment to build an SDS can easily be created using this toolkit. However, even to create the environment, expert knowledge of the spoken dialog is necessary in constructing an SDS. Novice users find it difficult to build a dialog system without such knowledge. In the construction of the SDS, it is necessary to edit the dialog scenario of the contents of the conversation. However, in the current editing environment, it is difficult to read in a complex dialog scenario, and editing is thus difficult even for the expert user. Therefore, in this paper, we develop a dialog scenario editor to improve the editing environment.

* Corresponding author.
E-mail address: ryota@nishimura.name (R. Nishimura).
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2018.03.002

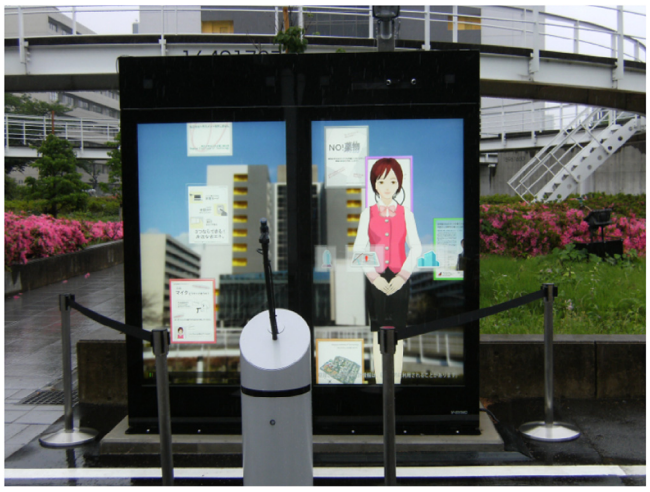

Fig. 1. Life-size 3D character “Mei-chan” at the main gate of a university.

2. Dialog scenario

As a method of describing a spoken dialog, VoiceXML (VXML) is well known [9]. Interaction between human -computer has been described in the XML format. VoiceXML is a kind of standard XML format of the W3C. XISL have also been developed [10], it is possible to describe the interaction using any modality.

The dialog scenario in MMDAgent is managed according to the finite-state transducer (FST) format. When the user creates a scenario, it is necessary to describe the state transition of the FST in a text file (Fig. 3). The FST format is a list of four values, namely the state number, transition state number, acceptance conditions and command, separated by spaces. In the example of Fig. 3, when an event (RECOG_STOP|Hello) is received in state 1, the system makes a transition to state 10 without any output (⟨eps⟩). It then outputs a command message (MOTION_ADD|mei|greet|greet.vmd), and the system makes a transition to state 11. Thus, the system controls the dialog by repeating the exchange of internal messages and state transitions. ⟨eps⟩ denotes the epsilon transition, which is a transition without any input or output.

In the scenario file, such as that in Fig. 3, the indenting of each item is performed manually using a space or tab. In some cases, indentations are not aligned and readability is thus poor. Additionally, the user must remember the commands to describe the scenario. Scenario editing in a text editor is therefore difficult even for the expert user.

```
 1 10 RECOG_STOP|Hello <eps>
 1 10 RECOG_STOP|Good morning <eps>
10 11 <eps> MOTION_ADD|mei|greet|greet.vmd
11 12 <eps> SYNTH_START|mei|normal|Good morning.
12 13 SYNTH_STOP|mei MOTION_ADD|mei|happy.vmd
13 14 <eps> SYNTH_START|mei|happy|It is a nice morning.
14  1 SYNTH_STOP|mei <eps>
```

Fig. 3. Example of a dialog scenario FST (Some notation is simplified).

3. Dialog scenario editor (MMDAE)

To improve the created environment of spoken dialog scenarios and thus solve the problems described above, we developed a scenario editor (Fig. 2). The scenario editor is named MMDAE (MMDAgent scenario Editor). Three features of MMDAE are discussed in the following.

3.1. 1: Completion of the input

To create a scenario, the input of four items is required, as shown in Fig. 3. It is difficult to type all the dialog scenarios (conditions and commands). An input complement function was thus implemented in the system. Furthermore, after the command is entered, the text area in which to enter arguments is displayed.

3.2. 2: Execution on various platforms

The MMDAgent can be run on a variety of platforms such as PCs (running Windows, Mac OS, or Linux) and smartphones (running Android). From this, the editing environment of the dialog scenario also needs to be operated on various platforms. So, we develop a scenario editor that is implemented on a Web browser. Furthermore, MMDAgent has the ability to share the dialog scenario on the Internet, and in this respect, the use of a Web browser is effective.

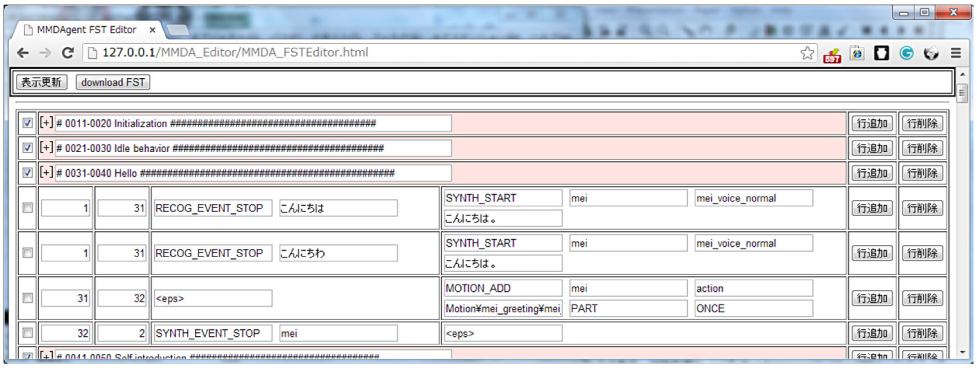

Fig. 2. Dialog scenario editor (MMDAE).

Fig. 4. Simple edit mode.

[dialog 1]
USER :Good morning
SYSTEM :Good morning

0031-0040 Hello
1 31 RECOG_STOP|Hello ...
31 32 <eps> ...
32 2 SYNTH_STOP|mei ...

Fig. 5. Example of an editing task.

3.3. 3: Changing the edit mode

To change the ease of editing the dialog scenario according to the user's knowledge and experience of using an SDS, it is possible to change the edit mode. Only a few items are displayed to the novice user. In this way, the user can edit the dialog scenario without knowledge of the scenario description method.

Expert users are presented many items and are provided an environment in which to create a scenario using the full functionality of MMDAgent. Furthermore, according to the hardware used to edit a scenario (e.g., a PC or smartphone), it is possible to select an appropriate display method. For example, a compact display is presented for the small screen of a smartphone, as shown in Fig. 4.

4. Evaluation

An experiment was performed to evaluate the performance of MMDAE. Subjects were 13 male bachelor's and master's students in their twenties.

The experiments were performed according to the following procedure.

1. Explanation of the experiment
2. Explanation of the interaction scenario (FST)
3. Pre-confirmation of questionnaire items
4. Experiment (scenario editing)
   - Editing using a text editor (Notepad)
   - Explanation of how to use MMDAE
   - Editing using MMDAE
5. questionnaire.

Notepad (the standard text editor in Windows) and MMDAE were compared in the evaluation of the editing environment of the dialog scenario. In the experiment, subjects appended the contents of the dialog in the scenario file. The order of use of editing tools may affect the experimental results, because the subject may become used to editing. Therefore, subjects were divided into two groups, and the order of the use of editing tools was different for each group. The duration of editing by each subject was recorded. Experiments were performed using a PC that is usually used by the subjects in the laboratory.

The task was presented to each subject in the form of Fig. 5. There were four types of tasks, and subjects used both MMDAE and Notepad for each task.

A questionnaire was conducted after the experiment. For the following items, the subjects gave a score on a five-point Likert scale. In the case of Q1, for example, a response that the subject found it difficult to edit is awarded 1 point, and a response that it was easy to edit is awarded 5 points.

- Easy to edit (Q1: Notepad, Q2: MMDAE)
- Readability (Q3: Notepad, Q4: MMDAE)
- Easy to understand the usage (Q5: Notepad, Q6: MMDAE)
- Convenience of function
  - Q7: input complement function
  - Q8: focus switched by TAB key
  - Q9: text area division in accordance with the number of arguments
- About MMDAgent
  - Q10: Were you aware of MMDAgent previously?
  - Q11: Have you used MMDAgent previously?
- For the following items, subjects wrote freely
  - Q12: What functions does the system require?
  - Q13: Other comments (good and bad points about the system).

5. Results

The results of subjective evaluation based on the five-point Likert scale and objective evaluation based on the editing duration are presented below.

5.1. Subjective evaluation

Experimental results are shown in Fig. 6. The questions Q1–11 are as listed in the previous section. Fig. 6 shows that, in comparing the two editing systems (i.e., Q1 vs. Q2, Q3 vs. Q4, and Q5 vs. Q6), MMDAE obtained higher marks for all items, and the difference between scores was more than 1 point. In

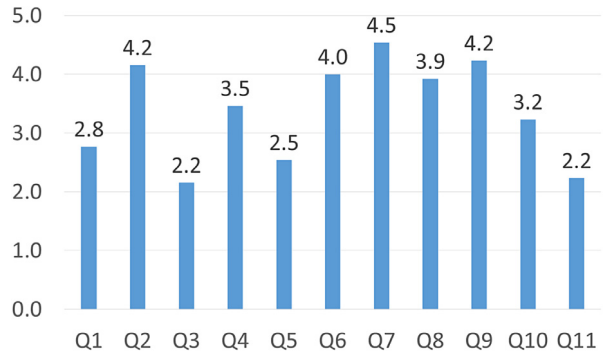

Fig. 6. Results of subjective evaluation (five-point scales).

Table 1
Edit time.

	Notepad	MMDAE
Average	4:50	3:58

particular, in terms of easy editing and easy understanding of use, MMDAE scored 4 points or more on average. Additionally, the functionality of MMDAE (Q7, Q8, and Q9) scored highly. According to these results, MMDAE is more suitable than Notepad for dialog editing.

5.2. Objective evaluation

The editing duration and efficiency were compared between editing systems. The results are given in Table 1. Whereas the editing duration using Notepad was 4 min 50 s on average, that using MMDAE was 3 min 58 s on average; i.e., the editing time was reduced on average by 52 s (or about 18% of the average duration using Notepad) when using MMDAE.

5.3. Opinions from the questionnaire

The subjects' opinions on MMDAE, in terms of their freely written answers to the questions in the survey, are listed below.

There were the following positive opinions.

- The efficiency of work was improved considerably by the input complement function of MMDAE.
- Readability using MMDAE is better than that using a text editor.
- Without any knowledge of the FST and with auto-completion of the conversation command, editing was easy.

According to these comments, the subjects felt that MMDAE had improved efficiency. In particular, there were many positive opinions on readability. The subjects in this experiment had no editing experience of the dialog scenario file prior to the experiment, and had no knowledge of commands of the FST. However, because the input was complemented by MMDAE, the user could learn the FST while editing.

There was however the following negative opinion.

- Many text boxes are displayed, which is confusing.

Opinions were often positive in terms of readability, but there was also the negative opinion that each user should be presented with a format that suits them. The present experiment was conducted using only the detailed edit mode, and the experiment should be repeated using the simple edit mode shown in Fig. 4. A user who is confused by the many text boxes in the detailed edit mode might find it easier to use the simple edit mode.

There were the following opinions.

- When the mouse cursor is over a button, a description of the button should be displayed.
- There should be a line copy function.

For the novice user of MMDAE, the function is displayed on each button or part of the display, and it is thus possible to edit the dialog scenario while learning how to use the system. Because this feature is important to beginners, it will be added to all edit modes of MMDAE in the future. Furthermore, the ability to copy one or more lines would be useful when describing a dialog scenario structured like a conversation scenario that has previously been described. This function will also be added.

6. Conclusion

To improve the environment for editing the scenario of an SDS, we developed a scenario editor (MMDAE). In an experiment, subjects preferred to edit a scenario in MMDAE than in Notepad, and the editing duration was about 18% less when using MMDAE. In terms of usability, MMDAE rated higher than Notepad in a survey of users.

In future work, from the results obtained in the subject questionnaire, we will add necessary functionality to the system. In addition to the items listed in the section "Objective evaluation", the automatic insertion of the FST number, checking of the transition destination, and a preview of the system response using speech synthesis will be implemented for the system.

Additionally, we want to add a function to enter the dialog scenario using speech recognition. Using this function, it will be possible to create a scenario while checking the behavior of the system. In using voices to edit the system response, prosodic information such as the speech rate, accent, and pitch can be input.

Acknowledgments

This study has been supported by the Core Research for Evolutional Science and Technology (CREST) (11102610) of the Japan Science and Technology Agency (JST), by JSPS KAKENHI Grant Number JP 25700009, and by the Strategic Information and Communications R&D Promotion Programme (SCOPE) (162309001) of Ministry of Internal Affairs and Communications (MIC), Japan.

Conflict of interest

The authors declare that there is no conflict of interest in this paper.

References

[1] P.J. Price, Evaluation of spoken language systems: the ATIS domain, in: Proc. DARPA Speech & Natural Language Workshop, 1990, pp. 91–95.

[2] S. Kawamoto, H. Shimodaira, T. Nitta, T. Nishimoto, S. Nakamura, K. Itou, S. Morishima, T. Yotsukura, A. Kai, A. Lee, Y. Yamashita, T. Kobayashi, K. Tokuda, K. Hirose, N. Minematsu, A. Yamada, Y. Den, T. Utsuro, S. Sagayama, Open-source software for developing anthropomorphic spoken dialog agent, in: Proc. of PRICAI-02, International Workshop on Lifelike Animated Agents, 2002, pp. 64–69.

[3] Y. Matsuyama, K. Hosoya, H. Taniyama, H. Tsuboi, S. Fujie, T. Kobayashi, SCHEMA: Multi-party interaction-oriented humanoid robot, in: ACM SIGGRAPH ASIA 2009 Art Gallery & Emerging Technologies: Adaptation, 2009, pp. 82–82.

[4] Apple Inc., Siri, http://www.apple.com/ios/siri/.

[5] A. Lee, K. Oura, K. Tokuda, MMDAgent - A fully open-source toolkit for voice interaction systems, in: Proc. of ICASSP 2013, 2013, pp. 8382–8385.

[6] K. Oura, D. Yamamoto, I. Takumi, A. Lee, K. Tokuda, On-campus, user-participatable, and voice-interactive digital signage, J. Jpn. Soc. Artif. Intell. 28 (1) (2013) 60–67.

[7] D. Yamamoto, K. Oura, R. Nishimura, T. Uchiya, A. Lee, I. Takumi, K. Tokuda, Voice interaction system with 3D-CG modeling for stand-alone smartphones, in: Proc. of International Conference on Human-Agent Interaction (IHAI) 2014, 2014, pp. 323–330.

[8] T. Uchiya, D. Yamamoto, M. Shibakawa, M. Yoshida, R. Nishimura, I. Takumi, Development of spoken dialogue service based on video call named 'Mobile Mei-chan', in: Proc. of JAWS2012, Interaction, 2012, pp. 1–3.

[9] VoiceXML Forum, VoiceXML: The standard application language for voice dialogues, http://www.voicexml.org/.

[10] K. Katsurada, Y. Nakamura, H. Yamada, T. Nitta, XISL:A language for describing multimodal interaction scenarios, in: Proc. of ICMI'03, 2003, pp. 281–284.