Web-based Multimodal Multi-domain Spoken Dialogue System

Ridong Jiang, Rafael E. Banchs, Seokhwan Kim, Kheng Hui Yeo, Arthur Niswar, Haizhou Li

Abstract. In this paper we describe a web-based spoken dialogue architecture, which is well-suited to the development of cloud enabled multimodal multi-domain dialogue systems. Different from conventional client/server architecture, the system leverages the latest web technology for low-latency and persistent communication. The dialogue service is configured by means of our generic spoken dialogue platform, namely APOLLO. This research work addresses the related issues on multiple dialogue services management, multimodal input and output as well as how to maintain dialogue states when working with stateless web interface. Several dialogue services ranging from different domains to different languages were implemented. Experiments on these systems showed that the systems are responsive and dialogues can be carried out through multimodal web interface with right state management.

1 Introduction

Spoken dialogue system is an intuitive, natural and flexible mean of communication between human and machine [1]. In recent years, spoken dialogue system has been widely used in a variety of applications: voice-operated cell phones, car navigation systems, gaming, education, healthcare and talking agents, etc. Successful commercial applications such as Apple Siri [2], Google Now and Samsung S Voice have also emerged and attracted human attention. In the mean time, speech interface, as an efficient and inexpensive mean of input and output, it becomes more and more popular for information query. This is especially true in the era when World Wide Web and mobile technology are experiencing explosive growth. Traditional dialogue system based on local desktop is shifting its landscape to providing web based dialogue service for a wide range of remote users.

Many efforts have been made to develop web based dialogue applications [3-7]. VoiceXML is one of the technologies which enable interactive access to the web through the telephone or a voice-driven browser and it is commonly used by many organizations from the speech, telecommunications and information technology industries. However, VoiceXML is not convenient for customization and enhancement (e.g. database support), lacks features on programming logic control, modeling of returning dialogue flow, etc. [8][9]. SpeechBuilder employs a we b-

R. Jiang (✉) • R.E. Banchs • S. Kim • K.H. Yeo • A. Niswar • H. Li

Institute for Infocomm Research, 1 Fusionopolis Way, #21-01, Connexis, Singapore 138632
E-mail: {rjiang, rembanchs, kims, yeokh, aniswar, hli}@i2r.a-star.edu.sg

Proceedings of 5th International Workshop on Spoken Dialog Systems
Napa, January 17-20, 2014

based interface and implemented via a number of Perl CGI scripts [10]. The semantic frame is converted to the CGI parameter representation. Dialogue state variable is exchanged with CGI application on every turn. This approach mixed the dialogue task representation with conventional CGI parameters which need to be sent using HTTP.

In this paper, we propose an architecture which leverages the latest web technology and our configurable spoken dialogue platform to deliver various dialogue services by means of multimodal web interface. This paper is organized as follows. First, the system architecture and proposed approach are presented in Section 2. It is followed by the description of multimodal input and output. Next dialogue service configuration is presented in section 4. Finally conclusion is drawn in Section 5.

2 Web-based Spoken Dialogue Architecture

The web-based multimodal spoken dialogue system can be broadly divided into following parts: a web server, a proxy server, dialogue services and web browsers as shown in Fig. 1.

2.1 Web Server

The web server is purely providing the web interface for dialogue input and output. In the mean time, this web page specifies the dialogue service to which the client web browser will subscribe. When this web page is opened through a client web browser, the web browser will directly connect to the proxy server as shown in Fig. 1. Once connected, a dialogue service identifier (DSID) will be automatically sent to the proxy server to inform the proxy what dialogue service the user is subscribing to. The DSID is a unique identifier for a dialogue service. Based on the DSID, the proxy will be able to direct the user to the right dialogue service.

2.2 Proxy Server

Proxy server bridges all web browsers with correspondent dialogue services through DSID. There are two servers running in this Proxy. The first server is websocket server. It is the server for all web browsers for dialogue services. Websocket is a protocol providing low-latency, bi-directional, full-duplex communication channels over a single TCP connection [11]. It is one of the HTML5 features which support by many latest web browsers such as Google Chrome, Windows internet explorer, Mozilla Firefox, etc. The connection is persistent and

the server can initiate communication with the browser. Another server is TCP server. It is the server for all dialogue engines which provide different dialogue services. When every dialogue engine starts, it will automatically connect to this TCP server. Once connected, a DSID will be sent to this server to register its service. To bridge the connection between a web browser and a dialogue service, the proxy server will internally match them by DSID. In the mean time, for every connection, a unique client identifier (UCID) which is composed of client IP address and socket number is assigned by the server. This is to facilitate the matching from dialogue service to web client. The UCID is necessary because one dialogue service may be connected by multiple web clients.

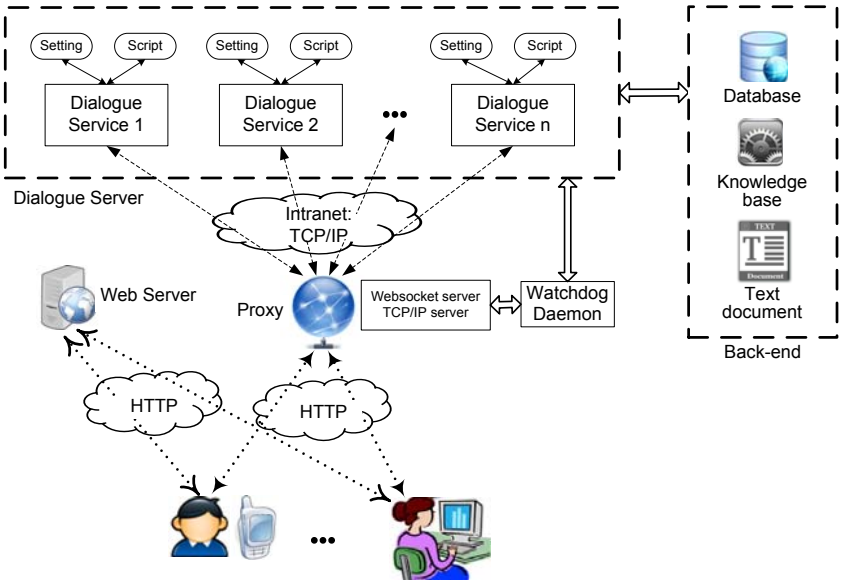

Fig.1. System architecture of web-based spoken dialogue system

2.3 Dialogue Server

Dialogue server refers to the machines running different dialogue services. Physically it can be one or multiple machines. One dialogue server allows running multiple instances of the same APOLLO dialogue system. Each instance is configured to work on particular dialogue tasks. Hence one dialogue instance running on the dialogue server is considered as one dialogue service. Every dialogue instance comes with its own XML setting file and a XML service script file. The dialogue engine is fully driven by the XML script file. The script file specifies dialogue states, dialogue flow control as well as low level communication between dialogue components such as natural language understanding and text-to-speech.

3 Multimodal Input and Output

The web interface is composed of following basic elements for input and output: a) Animated avatar, b) Text input field with "submit" button, c) Text output field, d) Icon for voice recording, e) Inline frame for embedding output web document. Fig. 2 shows an example dialogue interaction with multimodal input and output. The input can be text or speech. Currently, we use Google Voice to transcribe voice to text when Chrome browser is used. For other browsers, e.g. Internet Explorer, text can be directly input into the web interface by typing. Vision is also supported in the web interface. Sitepal avatar [12] is used in the web interface for face tracking and text to speech. The face position is translated into screen coordinates which are used to control the Avatar eye gaze. By incorporating the face tracking function, the avatar will be able to turn its head to follow human's movement so that more natural communication with eye contact can be achieved.

Multimodal outputs include response from the dialogue, which is rendered in both text display and speech, avatar's synchronized lip movement, as well as related web page for the answer.

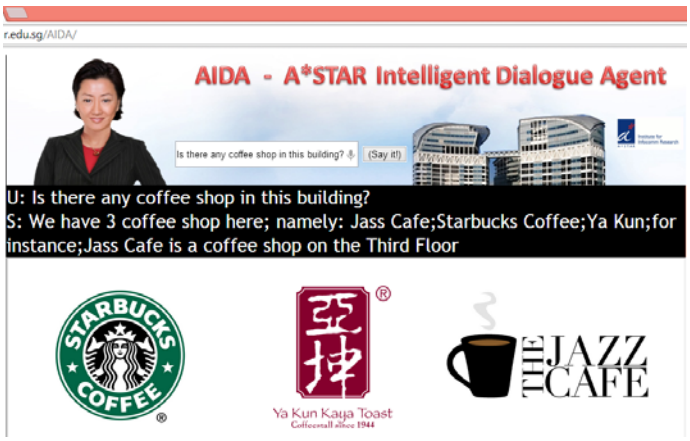

Fig.2 example dialogue interaction with multimodal input and output

4 Dialogue Service

4.1 Maintaining Dialogue State

Web-based system has to address the challenge of multiple parallel users. This is especially true when the interaction has to deal with dialogue states and contexts such as flight booking. In this framework, client UCID is used by the dialogue

service to keep track the user's interaction as well as to identify the current dialogue user in the following turns of interaction.

4.2 Dialogue Framework

The dialogue system working behind the web is developed on top of our existing general purpose spoken dialogue framework named APOLLO. A plug-in, working as TCP socket client, was developed for communication with Proxy Server. In this dialogue framework, all spoken dialogue related components are component-based and reusable. The dialogue system is functionally divided into dialogue manager and a number of standard dialogue components or plug-ins. A programmable message centre is designed to facilitate the communication between internal components, as well as message routing to graphic user interface (GUI) or middleware interface for the communication with external module through TCP/IP. With the developed database plug-in, rule engine plug-in and information retrieval plug-in, the dialogue framework is able to access rich information from the backend (database, knowledge base) [13] to meet different requirements.

4.3 Configuration of Dialogue Service

APOLLO dialogue framework is configurable. Every dialogue service comes with two XML files: One script file defining dialogue logic and one setting file. Multiple dialogue services can be achieved on the same dialogue server by running multiple instances of the dialogue system with different setting file and script file. We have deployed several dialogue services for different domains and different languages. One typical application is a web-based dialogue agent for multiple domains which range from flight booking, corporate information FAQ, chatting, facility information query as well as the agent self information query. With an input of user's utterance, the agent will first identify the domain, and then the user's input will be directed to the right task processing engine and finally the answer will be formed and presented to the user through web interface.

5 Conclusions

In this paper we have presented an architecture which leverages the latest web technology and our configurable spoken dialogue platform to deliver various dialogue services by means of multimodal web interface. The proposed system separates web server from the dialogue services by direct persistent communication between web browser and dialogue proxy server in JSON format through HTTP.

Various dialogue services for different domains can be quickly configured by reusing existing dialogue components. Several dialogue services ranging from different domains to different languages were implemented. Experiments on these systems showed that the systems were responsive and dialogues can be carried out through multimodal web interface with right state management.

References

1. MICHAEL F. MCTEAR, "Spoken Dialogue Technology: Enabling the Conversational User Interface", *ACM Computing Surveys*, Vol. 34, No. 1, March 2002, pp. 90–169.
2. Jerome R. Bellegarda, "Spoken Language Understanding for Natural Interaction: the Siri Experience". Proceedings of the 4th International Workshop on Spoken Dialogue Systems, Ermenonville, France, Nov 28-30, 2012, pp.3-14.
3. Goddeau, David et al., Deploying Speech Applications over the Web, Digital Equipment Corporation, 4 pages,l997.
4. M. Fuchs, N. Tsourakis, and M. Rayner, "A scalable architecture for web deployment of spoken dialogue systems," in Proceedings of LREC 2012, Istanbul, Turkey, 2012.
5. Richard A. Frost, Ali Karaki, David A. Dufour, Josh Greig, Rahmatullah Hafiz, Yue Shi, Shawn Daichendt, Shahriar Chandon, Justin Barolak, and Randy J. Fortier, "MySpeechWeb: Software to Facilitate the Construction and Deployment of Speech Applications on the Web", ASSETS'08, October 13–15, 2008, Halifax, Nova Scotia, Canada.
6. Voice Extensible Markup Language (VoiceXML) 2.1, W3C Recommendation 19 June 2007: http://www.w3.org/TR/voicexml21/.
7. Kuansan Wang, "SALT: An XML Application for Web-based Multimodal Dialog Management". Proceedings of the 2nd workshop on NLP and XML - Volume 17, Association for Computational Linguistics Stroudsburg, PA, USA ©2002
8. S.W. Hamerich, Y. H. Wang, V. Schubert, V. Schless and S. Igel, "XML-Based Dialogue Descriptions in the GEMINI Project". Proceedings of the "Berliner XML-Tage 2003, Germany, pp. 404-412.
9. T. Heinroth and D. Denich, "Spoken Interaction within the Computed World: Evaluation of a Multitasking Adaptive Spoken Dialogue System", 35th Annual IEEE International Computer Software and Applications Conference (COMPSAC 2011), IEEE,2011.
10. James Glass and Eugene Weinstein, " SpeechBuilder: Facility Spoken Dialoguc System Dcvclopment." The 7th European Conference on Speech Communication and Technology, Aalborg, Denmark, September, 2001.
11. Internet Engineering Task Force (IETF), December 2011, The WebSocket Protocol: http://tools.ietf.org/html/rfc6455.
12. Sitepal – Create an animated talking character for your website: http://www.sitepal.com/home/.
13. R. Jiang, Y. K. Tan, D. K. Limbu, and H. Li, "Component pluggable dialogue framework and its application to social robots", in *Proc. Int'l Workshop on Spoken Language Dialog Systems*, 2012.