A distributed cloud-based dialog system for conversational application development

Vikram Ramanarayanan[†], David Suendermann-Oeft[†], Alexei V. Ivanov[†] and Keelan Evanini[‡]
Educational Testing Service R&D
[†] 90 New Montgomery St, # 1500, San Francisco, CA
[‡] 600 Rosedale Road, Princeton, NJ
<vramanarayanan,suendermann-oeft,aivanou,kevanini>@ets.org

Abstract

We have previously presented HALEF–an open-source spoken dialog system–that supports telephonic interfaces and has a distributed architecture. In this paper, we extend this infrastructure to be cloud-based, and thus truly distributed and scalable. This cloud-based spoken dialog system can be accessed both via telephone interfaces as well as through web clients with WebRTC/HTML5 integration, allowing in-browser access to potentially multimodal dialog applications. We demonstrate the versatility of the system with three conversation applications in the educational domain.

1 The HALEF spoken dialog system

The HALEF (Help Assistant–Language-Enabled and Free) framework leverages different open-source components to form a spoken dialog system (SDS) framework that is modular and industry-standard-compliant: Asterisk, a SIP- (Session Initiation Protocol), WebRTC- (Web Real-Time Communication) and PSTN- (Public Switched Telephone Network) compatible telephony server (van Meggelen et al., 2009); JVoiceXML, an open-source voice browser that can process SIP traffic (Schnelle-Walka et al., 2013) via a voice browser interface called Zanzibar (Prylipko et al., 2011); Cairo, an MRCP (Media Resource Control Protocol) speech server, which allows the voice browser to initiate SIP or RTP (Real-time Transport Protocol) connections from/to the telephony server (Prylipko et al., 2011); the Kaldi (Povey et al., 2011) and Sphinx-4 (Lamere et al., 2003) automatic speech recognizers; Festival (Taylor et al., 1998) and Mary (Schröder and Trouvain, 2003)–text-to-speech synthesis engines; and an Apache Tomcat-based web server that can host dynamic VoiceXML (VXML) pages and serve media files such as grammars and audio files to the voice browser. HALEF includes support for popular grammar formats, including JSGF (Java Speech Grammar Format), SRGS (Speech Recognition Grammar Specification), ARPA (Advanced Research Projects Agency) and WFST (Weighted Finite State Transducers). Figure 1 schematically depicts the main components of the HALEF system. Note that unlike a typical SDS, which consists of sequentially-connected modules for speech recognition, language understanding, dialog management, language generation and speech synthesis, in HALEF some of these are grouped together forming independent blocks which are hosted on different virtual machines in a distributed architecture. In our particular case, each module is hosted on a separate server on the Amazon Elastic Compute Cloud (EC2)[1]. This migration to a cloud-based distributed computing environment allows us to scale up applications easily and economically. Further, added integration and compatibility with the WebRTC standard[2] allows us to access HALEF from within a web browser, thus allowing us to design and develop multimodal dialog interfaces (that potentially can include audio, video and text, among other modalities. For further details on the individual blocks as well as design choices, please refer to (Mehrez et al., 2013; Suendermann-Oeft et al., 2015; Ramanarayanan et al., submitted). In this framework, one can serve different back-end applications as standalone web services on a separate server. Incorporating the appropriate start URL of the web service in the VXML input code that the voice browser interprets will then allow the voice browser to trigger the web application at the appropriate point in the callflow. The web services in our case typically take as input any valid

[1] http://aws.amazon.com/ec2/
[2] http://www.w3.org/TR/webrtc/

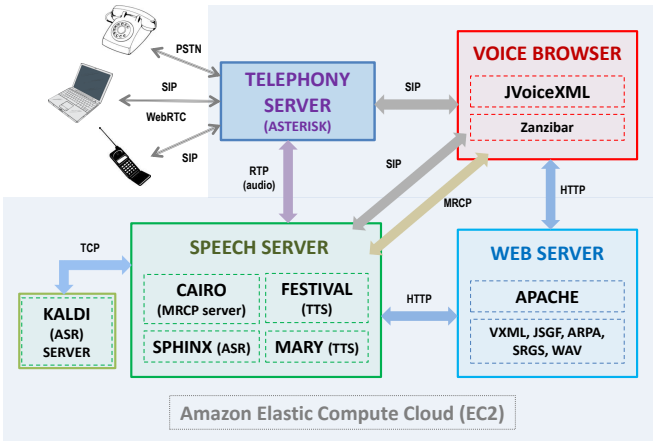

Figure 1: System architecture of the cloud-based HALEF spoken dialog system depicting the various modular open-source components.

HTTP-based GET or POST request and output a VXML page that the voice browser can process next. In the next section, we describe a software toolkit that can dynamically generate a sequence of VXML pages from a dialog flow specification.

We also developed a logging interface that helps users view log messages from the Tomcat server, speech server and voice browser in real time to facilitate debugging and understanding of how to improve the design of the item dialog flow. This web-based tool allows designers to observe in real time the output hypotheses generated by the speech recognition and natural language understanding modules at each dialog state, as well as hyperlinks to the grammars and speech audio files associated with that state. This allows even dialog flow designers with minimal spoken dialog experience to monitor and evaluate system performance while designing and deploying the application.

2 The OpenVXML dialog-authoring suite

Also integrated into the HALEF framework is OpenVXML (or Open Voice XML), an open-source software package[3] written in Java that allows designers to author dialog workflows using an easy-to-use graphical user interface, and is available as a plugin to the Eclipse Integrated Developer Environment[4]. OpenVXML allows designers to specify the dialog workflow as a flowchart, including details of specific grammar files to be used by the speech recognizer and text-to-speech prompts that need to be synthesized.

[3] https://github.com/OpenMethods/OpenVXML
[4] www.eclipse.org

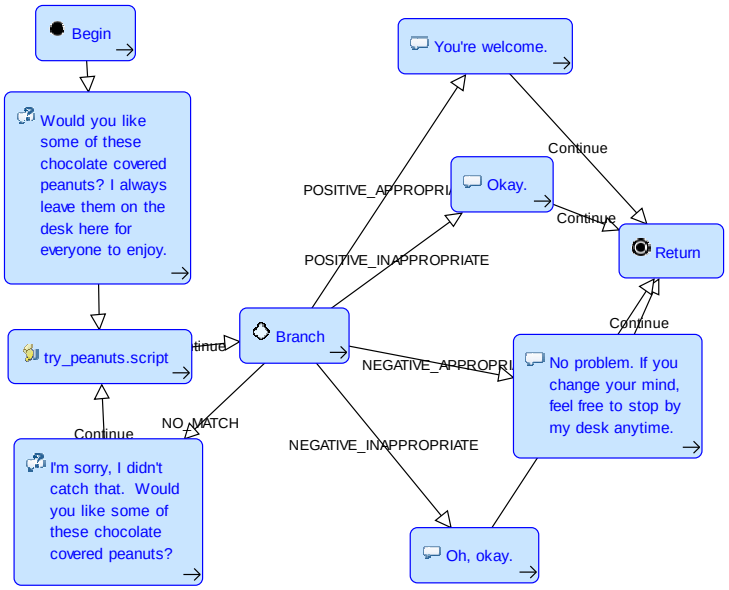

Figure 2: Example design of a workplace pragmatics-oriented application.

In addition, they can insert "script" blocks of Javascript code into the workflow that can be used to perform simple processing steps, such as basic natural language understanding on the outputs of the speech recognition, for example. The entire workflow can be exported to a Web Archive (or WAR) application, which can then be deployed on a web server running Apache Tomcat that serves Voice XML (or VXML) documents.

3 Applications

Figures 2 and 3 show example workflows of conversational items developed using OpenVXML. The caller dials into the system and then proceeds to answer a sequence of questions, which can be either be stored for later analysis (so no online recognition and natural language understanding needed), or processed in the following manner. Depending the semantic class of the callers' answer to each question (as determined by the output of the speech recognizer and the natural language understanding module), they are redirected to the appropriate branch of the dialog tree and the conversation continues until all such questions are answered. Notice that in the case of this simple example we are using rule-based grammars and dialog tree structures, though the system can also natively support more sophisticated statistical modules.

4 Conclusions

We have presented a prototype conversation-based application that leverages the open-source

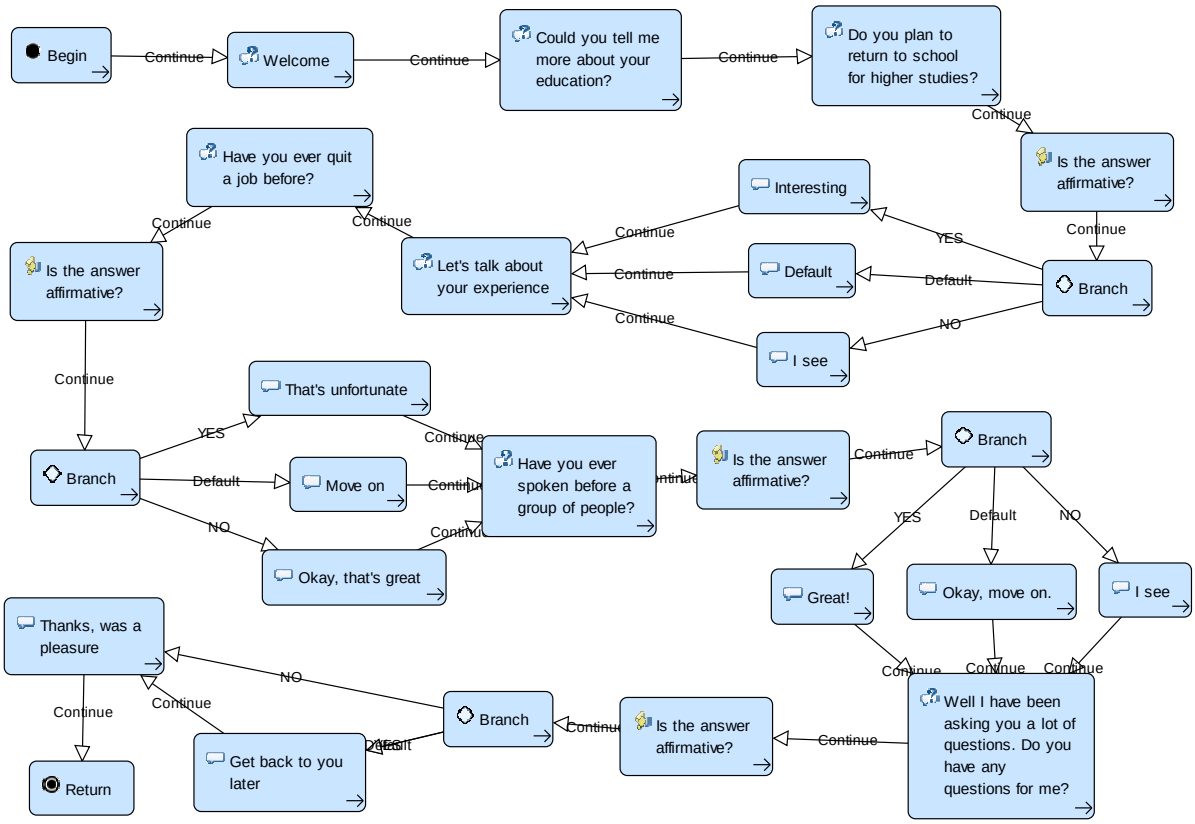

Figure 3: Example workflow design of an interview test application.

HALEF spoken dialog framework. HALEF can be accessed online at the following URL: `http://halef.org`. One can also call into HALEF for a demo of the interview item at the following US-based telephone number: `(206) 203-5276 (Extension:7749)`.

5 Acknowledgements

The authors would like to thank Lydia Rieck, Elizabeth Bredlau, Katie Vlasov, Eugene Tsuprun, Juliet Marlier, Phallis Vaughter, Nehal Sadek, and Veronika Laughlin for helpful input in designing the conversational items.

References

P. Lamere, P. Kwok, E. Gouvea, B. Raj, R. Singh, W. Walker, M. Warmuth, and P. Wolf. 2003. The CMU SPHINX-4 Speech Recognition System. In *Proc. of the ICASSP'03*, Hong Kong, China.

T. Mehrez, A. Abdelkawy, Y. Heikal, P. Lange, H. Nabil, and D. Suendermann-Oeft. 2013. Who Discovered the Electron Neutrino? A Telephony-Based Distributed Open-Source Standard-Compliant Spoken Dialog System for Question Answering. In *Proc. of the GSCL*, Darmstadt, Germany.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi speech recognition toolkit. In *Proc. of the ASRU*, Hawaii, USA.

D. Prylipko, D. Schnelle-Walka, S. Lord, and A. Wendemuth. 2011. Zanzibar OpenIVR: An Open-Source Framework for Development of Spoken Dialog Systems. In *Proc. of the TSD*, Pilsen, Czech Republic.

Vikram Ramanarayanan, David Suendermann-Oeft, Alexei Ivanov, Keelan Evanini, and Nehal Sadek. submitted. Toward an open-source spoken dialog framework for developing conversation-based educational applications. In *Interspeech 2015, Dresden, Germany*.

D. Schnelle-Walka, S. Radomski, and M. Mühlhäuser. 2013. JVoiceXML as a Modality Component in the W3C Multimodal Architecture. *Journal on Multimodal User Interfaces*.

Marc Schröder and Jürgen Trouvain. 2003. The german text-to-speech synthesis system mary: A tool for research, development and teaching. *International Journal of Speech Technology*, 6(4):365–377.

David Suendermann-Oeft, Vikram Ramanarayanan, Moritz Teckenbrock, Felix Neutatz, and Dennis Schmidt. 2015. Halef: an open-source standard-compliant telephony-based modular spoken dialog system–a review and an outlook. In *International Workshop on Spoken Dialog Systems (IWSDS) 2015, Busan, South Korea*.

P. Taylor, A. Black, and R. Caley. 1998. The Architecture of the Festival Speech Synthesis System. In *Proc. of the ESCA Workshop on Speech Synthesis*, Jenolan Caves, Australia.

J. van Meggelen, J. Smith, and L. Madsen. 2009. *Asterisk: The Future of Telephony*. O'Reilly, Sebastopol, USA.