Multimodal HALEF: An Open-Source Modular Web-Based Multimodal Dialog Framework

Zhou Yu[‡†], Vikram Ramanarayanan[†], Robert Mundkowsky[†], Patrick Lange[†], Alexei Ivanov[†], Alan W Black[‡] and David Suendermann-Oeft[†]

Abstract We present an open-source web-based multimodal dialog framework, "Multimodal HALEF", that integrates video conferencing and telephony abilities into the existing HALEF cloud-based dialog framework via the FreeSWITCH video telephony server. Due to its distributed and cloud-based architecture, Multimodal HALEF allows researchers to collect video and speech data from participants interacting with the dialog system outside of traditional lab settings, therefore largely reducing cost and labor incurred during the traditional audio-visual data collection process. The framework is equipped with a set of tools including a web-based user survey template, a speech transcription, annotation and rating portal, a web visual processing server that performs head tracking, and a database that logs full-call audio and video recordings as well as other call-specific information. We present observations from an initial data collection based on an job interview application. Finally we report on some future plans for development of the framework.

1 Introduction and Related Work

Previously, many end-to-end spoken dialog systems (SDSs) used close-talk microphones or handheld telephones to gather speech input [3] [22] in order to improve automatic speech recognition (ASR) performance of the system. However, this limits the accessibility of the system. Recently, the performance of ASR systems has improved drastically even in noisy conditions [5]. In turn, spoken dialog systems are now becoming increasingly deployable in open spaces [2]. They can also inter-

[†] Education Testing Service (ETS) R&D, San Francisco, CA, USA, Princeton, NJ, USA.
e-mail: { vramanarayanan, suendermann-oeft, rmundkowsky, plange, aivanou}@ets.org
[‡] Carnegie Mellon University, Pittsburgh, PA, USA. e-mail: {zhouyu,awb}@cs.cmu.edu

The first author performed the work while she was interning with Education Testing Service R&D in San Francisco, CA.

act with users remotely through the web without specific microphone requirements [8], thus reducing the cost and effort involved in collecting interactive data.

Recently, multimodal sensing technologies such as face recognition, head tracking, etc. have also improved. Those technologies are now robust enough to tolerate a fair amount of noise in the visual and acoustic background [4] [10]. So it is now possible to incorporate these technologies into spoken dialog systems to make the system aware of the physical context, which in turn will result in more natural and effective conversations [19].

Multimodal information has been proven to be useful in dialog system design in driving both low level mechanics such as turn taking as well as high level adaptive strategies such as user attention regulation. Sciutti et al. [16] used gaze as an implicit signal for turn taking in a robotic teaching context. In [20], a direction-giving robot used conversational strategies such as pause and restarts to regulate user's attention. Kousidis et al. [6] used situated incremental speech synthesis that accommodates users' cognitive load in a in-car navigation task, which improved user experience but the task performance stays the same. However most multimodal systems suffer from not enough data for model training or evaluation, and they are not easy to access, since most of them require you to be physically co-present with the system. The community has been struggling with limited publicly available data for a long time. We propose a web-based multimodal dialog system, Multimodal HALEF, to tackle this issue. It integrates the video-enabled Freeswitch telephony server with an open-source distributed dialog system, HALEF. Multimodal HALEF records the remote user's video interaction and streams it to its servers by accessing the remote user's camera via a web browser. The source code is available at `https://sourceforge.net/projects/halef/`.

2 Foundational Frameworks

In this section, we describe prior frameworks that the Multimodal HALEF framework extends and builds upon. Figure 1 schematically depicts the overall architecture of the Multimodal HALEF framework.

The HALEF dialog framework leverages different open-source components to form an SDS framework that is modular and industry-standard-compliant: Asterisk, a SIP (Session Initiation Protocol) and PSTN (Public Switched Telephone Network) compatible telephony server [18]; JVoiceXML, an open-source voice browser that can process SIP traffic [14] via a voice browser interface called Zanzibar [12]; Cairo, an MRCP (Media Resource Control Protocol) speech server, which allows the voice browser to initiate SIP and RTP (Real-time Transport Protocol) connections between the speech server and the telephony server [12]; the Sphinx automatic speech recognizer [7] and the Kaldi ASR system; Festival [17] and Mary [15] 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

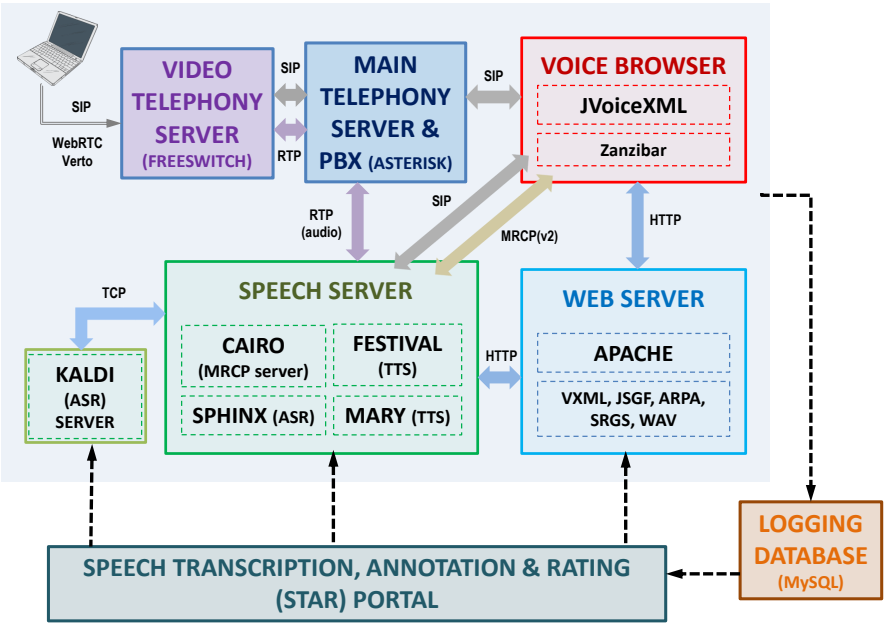

Fig. 1 System architecture of the HALEF multimodal dialog system depicting the various modular open-source components.

to the voice browser. 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. In addition, dialog designers 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. 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.

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. For further details on the individual blocks as well as design choices, please refer to [13] [9].

FreeSWITCH is a scalable open source cross-platform telephony framework designed to route and interconnect popular communication protocols using audio, video, text or any other form of media. It supports various communication technologies such as Skype, SIP and WebRTC. FreeSWITCH builds natively and runs standalone on several major operating systems, such as Linux, Windows and Mac

OS. FreeSWITCH has been previously integrated with a dialog system in [11] that allows access via a web browser. However, this system can only handle audio input. FreeSWITCH 1.6 is experimenter friendly. The experimenter can modify interaction settings, such as the number of people who can call in at any given time, whether to display the video of the user on the webpage, the resolution of the video, sampling rate of the audio, etc. FreeSWITCH 1.6 also allows users to choose between different I/O devices for recording. They can switch between different microphones and cameras connected to their computers by selecting appropriate options on the web-based graphical user interface.

3 Framework Integration

We integrated FreeSWITCH, specifically 1.6 Video version [2] including modules to support video with HALEF, so the dialog system have both video and audio as input from remote users through a webpage. Initially we planned to setup Asterisk to support video via WebRTC [3]. Unfortunately Asterisk does not support video yet. Thus we looked for other alternatives and found that FreeSWITCH released the FreeSWITCH 1.6 Video version which supports video as of May 2015.

We followed the FreeSWITCH documentation to set up FreeSWITCH 1.6 on Debian 8 (Jesse). We decicde to use Verto [4] for achieving the in-browser video access capability. Verto is a FreeSWITCH module included the default FreeSWITCH configuration. Using Verto, rather than SIP/RTP over WebRTC, offered a few advantages. First it has a working webpage based conference call demo that we easily modified to allow a caller to interact with HALEFs audio dialogs. Second it is capable of recording audio and video as it is streamed to/from the caller. The alternative, SIP/RTP over WebRTC has more web-based clients such as sip.js [5], sipml5 [6], and jssip [7], but these required more work to get them to be functional on Chrome and Firefox. The problems with these clients are likely rooted in the fact that WebRTC and Web Audio are continuously evolving, experimental APIs. Therefore we choose not use them for our current framework.

The following is a description of how video and audio data flow to/from the Multimodal HALEF system. First the human user goes to the Verto teleconference webpage we adapted in their browser and selects an extension to call. Different extensions are associated with different dialog system instances that have different task contents (e.g. task1: a generic job interview, task2: an interview for working at a pizza restaurant,...). This webpage application is written in HTML, CSS, and

[2] https://freeswitch.org/confluence/display/FREESWITCH/FreeSWITCH+1.6+Video

[3] http://www.webrtc.org

[4] https://freeswitch.org/confluence/display/FREESWITCH/mod_verto

[5] http://sipjs.com

[6] http://sipml5.org

[7] http://tryit.jssip.net

javascript which leverages experimental web browser APIs (WebRTC and Web Audio). The Web Audio API enable access to the audio and video devices (e.g. cameras, microphones, speakers, etc.) via the web browser. The Verto protocol, which leverages WebRTC, is then used via javascript to send video/audio to FreeSWITCH and receive audio from FreeSWITCH. Note that FreeSWITCH can also send video to the user, but HALEF currently only supports audio dialogs. The Verto teleconference webpage demo also uses the FreeSWITCH teleconference module. Overall, this is a teleconference that has two participants: HALEF and the user. When the call comes in from the user, HALEF starts the dialog with an audio prompt that flows out of HALEF system via Asterisk over SIP/RTP to FreeSWITCH. FreeSWITCH then sends the audio to the web browser over Verto. The user then gives a response to the system that flows over Verto to FreeSWITCH and then over SIP/RTP to HALEF's front end (Asterisk). During the teleconference, the users video and audio interactions are continuously streaming and being recorded in FreeSWITCH and HALEF audio prompts are being streamed to the user.

Within the HALEF system, once the interaction starts, Asterisk makes a SIP call to the Voice Browser (JVoiceXML) to get the specific dialog started based on the extension number dialed. The Voice Browser then gets various resource files from the Web Server via HTTP that will allow it to control the call flow or call logic. For example, if the user says "yes" or "no" to a question then the Voice Browser will tell HALEF the next audio prompt to send accordingly. The Voice Browser uses this information to tell the Speech Server a few things. First it tells the Speech Server to interact with Asterisk in regards to inbound and outbound audio (over SIP/RTP). Second it tells the Speech Server to send the transcriptions of the audio to itself. And finally the Voice Browser tells the Speech Server to receive text from it that will be synthesized to audio output to the user. The Voice Browser communicates with the Speech Server via MRCP and SIP.

There are other endpoints that are supported or likely can be supported with a little work by HALEF and/or Multimodal HALEF. An endpoint is defined as a device that lives at the network edge (e.g. a telephone or a client application that acts like a telephone) that usually one user uses. HALEF (non-Multimodal) already supports audio only dialogs with endpoints other than Verto. For example, we used PSTN (public switched telephone network) endpoints, aka land line telephones and cell phones, to place calls to HALEF. We did this by using a PSTN/SIP proxy such as ipKall [8]. We also used SIP over WebRTC, and SIP/WebRTC clients like sipml5, jssip, etc to connect to HALEF directly thru Asterisk as well as via webrtc2sip http://webrtc2sip.org/ to Asterisk. Note that we suggest using webrtc2sip, because it handles differences in implementations in Chrome and Firefox (in regards SRTP types and transcoding audio formats). As for Multimodal HALEF, we used Verto, but it is likely that with slightly modifications SIP/WebRTC clients could be supported.

[8] http://www.ipkall.com/

4 Supporting Modules

We introduced four supporting modules that assist researchers in conducting scientific research on interactive dialog systems and human behavior: a relational database that stores all call-log information, a survey webpage that collects users' feedback of the interaction and pushes the information to the database, a web-based call viewing and rating portal, and a remote video processing server.

4.1 Database

We use the open-source database MySQL for our data warehousing purposes. All modules in the Multimodal HALEF connect to the database and write their log messages to it. We then post process this information with stored procedures into easily accessible views. Metadata extracted from the logs include information about the interactions such as the duration of the interaction, the audio/video recording file names, the caller IP, etc. Additionally, we store participants' survey data and third-person rating information. All the modules connected to the database have been implemented such that all information will be available in the database as soon as the interaction, the survey, or the rating task is completed.

4.2 Participant Web-Survey

We created a survey webpage that participants were required to fill upon completion of their video call. The survey is embedded along with the experimental instructions for the participant. Once the participant finished the interaction, they will fill out the survey about their experience of the interaction and some demographic information of themselves. Once the participant clicks the submit button the information is pushed to the appropriate table in the MySQL database.

4.3 STAR Portal

We developed a third-person interaction rating portal dubbed the Speech Transcription Annotation and Rating (STAR) portal. It is mainly implemented using PHP and and the JavaScript framework jQuery. It has the advantage of accessing meta-data from the data warehouse and the audio/video data from the server as well. It provides a nice display of different information of the interaction. It also allows the experimenter to design rating questions that correspond to different components of the dialog framework or targeting the participant's performance, such as user speech input, system TTS, etc. It supports different types of questions, such as multiple

choice questions, open questions, etc. Thus the rating task can not only be simple perception feedback to the interaction, but also detailed human transcription of the entire dialog. The tool also lets the experimenter manage raters by assigning different interactions for different users for the rating task. All of the information will be stored in the database for further analysis. The webpage supports playing both audio and video recordings of the collected interaction data.

4.4 Visual Processing Service

We also set up a standalone Linux server for automatic head tracking via Cambridge Head Tracker [1]. It can track a user's head movement and also estimate the head pose (e.g. 15 degrees to the left) based on a video with a clear user face present. It supports live feature extractions. Currently we are still in the process of integrating it in the framework. It will take the video captured by FreeSWITCH server in real time as input and output the head tracking information to the webserver that hosts the VXML application (with dialog management instructions), thus making multimodal human behavior signals available in the dialog strategy selection module. This makes the dialog system aware of the user's behaviors so it can act accordingly. Previous literature suggests that "computer systems with capabilities to sense agreement and attention that are able to adapt or respond to these signals in an appropriate way will likely be perceived as more natural, efficacious and trustworthy" [19]. Visual information has also been shown to be critical in assessing the mental states of the users in other systems as well [21] [20]. So we included the visual processing service in our framework as well.

5 Example Application: Job Interview

The diagram in Figure 2 describes the dialog workflow of one example job interview application. This conversational item was created to assess participants' English conversational skills, pragmatics and ability to comprehend stimuli and respond appropriately to questions posed during naturalistic conversational settings. Among the things we would like to study in the long run with this type of application are: (i) how users signal engagement in conversing with a task-oriented dialog system; (ii) whether the ability to show engagement has an impact on potential hiring decisions; (iii) what aspects of the system prevent or contribute to third-person-perceived naturalness of the conversations.

We conducted a preliminary data collection in order to test and evaluate the system in anticipation of a much larger planned data collection using the Amazon Mechanical Turk crowd sourcing platform. The observations from this preliminary data collection follow in the sections below.

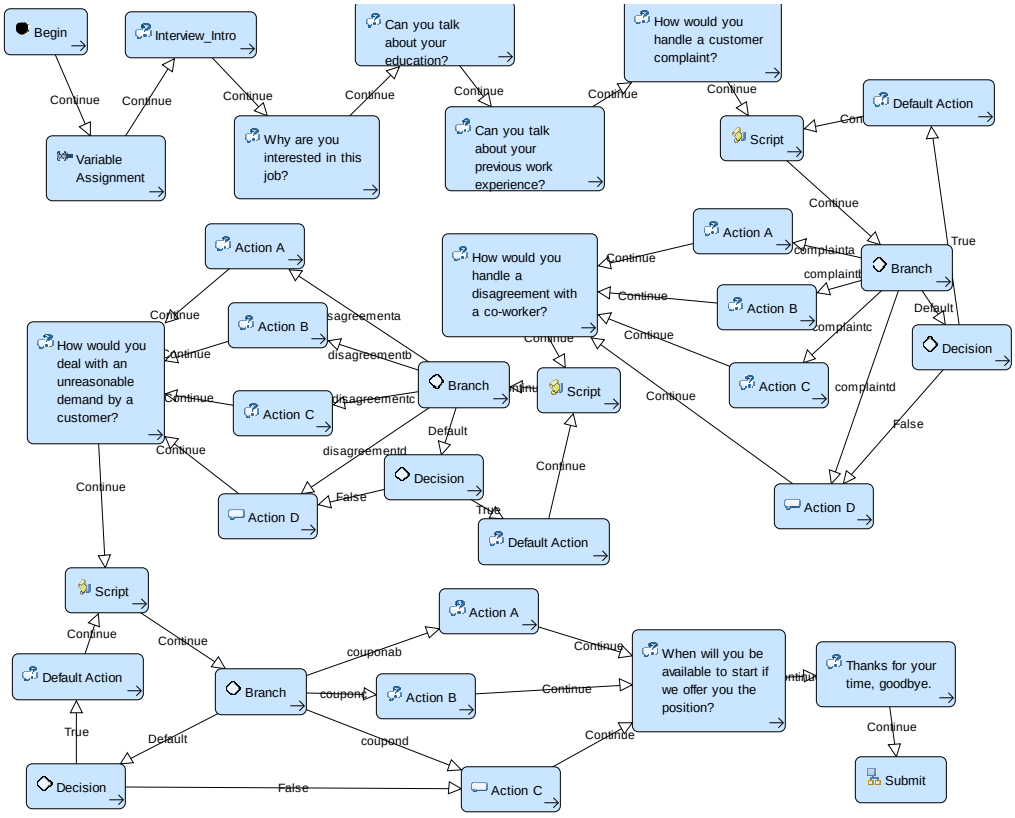

Fig. 2 Workflow of the job interview application

5.1 Preliminary Data Collection

5.1.1 Experimental Procedure

We recruited participants by sending them emails of a link to a webpage with detailed experimental instructions. The participants were given an example resume as a reference in the task in order to isolate the factor of different previous experience when they are talking to the interview application. The participants were instructed to access a webpage on their own computer and allow the webpage to use their local camera. Then they were asked to enter the specified extension and connected to the dialog system (every extension corresponds to a different dialog application). We instructed the user to select a uniform and low noise background when interacting with the system to reduce the possible errors of the head tracker and the ASR.

In the post-survey, we asked users a set of questions about the system itself and the experience of interacting with it. Sample questions include, “Please rate the delay of system responses in allowing you to carry on a natural conversation?”,“How

engaged did you feel while interacting with the system?", etc. In addition, we also had questions to elicit some background information from the participants, such as their English language nativeness, gender, age, etc. User background information is important in designing and performing statistical tests.

5.1.2 Data Statistics

There were 13 participants in total, six males and seven females, among whom eight were native English speakers and the other five, non-native English speakers. The participants' average age was 36. The average length for all the interactions was six minutes. Figure 3 shows a snapshot of a video recording. We asked participants to rate various aspects of their interactions with the system on a scale from 1 to 5, 1 being least satisfactory and 5 being most satisfactory. We also had an expert rate the recordings based on similar aspects. The results of the user and expert ratings are shown in Table 1.

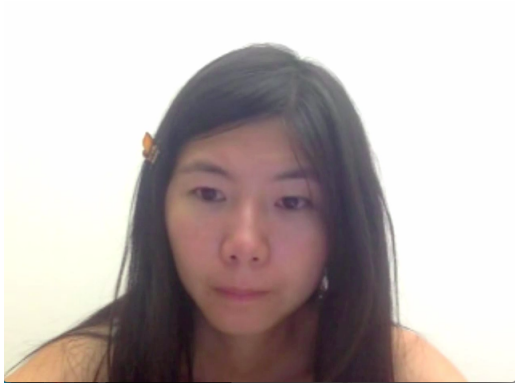

Fig. 3 A snapshot of a video recording

Table 1 User and Expert Ratings

User Ratings	Ratings Mean(SD)	Expert Ratings	Ratings Mean(SD)
Interaction completion	3.83(1.11)	Conversation Naturalness	3.62(0.51)
System intelligibility	4.61(0.77)	User cooperation	3.92(0.64)
User reported engagement	3.23(1.01)	User engagement	3.23(0.75)
System understanding degree	2.23(0.83)	Audio/Video quality	4.15(0.80)
Conversation latency	3.00(0.82)	System latency	3.92(0.28)
Overall user experience	3.00(0.82)	Overall user experience	3.62(0.50)

We used pre-recorded natural human speech for the system's voice output, thus the intelligibility of the system is very highly rated by the users. The self-reported

engagement and the expert user engagement rating is highly correlated (0.83 in Pearson correlation). We also found that the overall user experience and the conversation naturalness perceived by experts are both slightly correlated with the user's self-reported engagement during the interaction (0.50 and 0.49 respectively in Pearson correlation). This motivates us to develop systems that can improve user's engagement, which in turn may improve the overall user experience.

6 Conclusions and Future Work

We have designed and implemented an open-source web-based multimodal dialog framework, Multimodal HALEF, by integrating the existing dialog framework, HALEF, and a video conferencing framework, FreeSWITCH. The framework also includes a database, a rating portal, a survey and a video processing server. It allows an experimenter to collect video and audio data of users interacting with the dialog system outside of a controlled lab environment, and largely reduces the cost and labor in collecting audio-visual data. The framework was designed to facilitate scientific research on how humans interact with dialog systems, among other purposes. To this end, we also described and collected preliminary data on an example application that leverages the above framework and demonstrates its utility.

One limitation of the current system is that it only supports interactions with one user at time. In the future, we plan to port the system onto the Amazon Elastic Compute Cloud, thus making multiple simultaneous interactions possible. The participants of the initial data collection also provided some interesting feedback about the system which sheds light on potential future work. For example, one user suggested that having a virtual face/avatar would allow the system to signal turn taking patterns better. This is indeed along the lines of our planned future work involving multimodal synthesis and recognition (instead of simply speech recognition and synthesis, as is the case currently). In addition, we plan to incorporate and improve statistical models for automatic speech recognition, language understanding and dialog management to enhance system performance.

References

1. T. Baltrusaitis, P. Robinson, and L.-P. Morency. 3D constrained local model for rigid and non-rigid facial tracking. In *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2610–2617. IEEE, 2012.
2. D. Bohus, C. W. Saw, and E. Horvitz. Directions robot: in-the-wild experiences and lessons learned. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 637–644. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
3. M. Eskenazi, A. W. Black, A. Raux, and B. Langner. Let's go lab: a platform for evaluation of spoken dialog systems with real world users. In *Ninth Annual Conference of the International Speech Communication Association*, 2008.

4. X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using laplacianfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):328–340, 2005.
5. G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.
6. S. Kousidis, C. Kennington, T. Baumann, H. Buschmeier, S. Kopp, and D. Schlangen. A multimodal in-car dialogue system that tracks the driver's attention. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 26–33. ACM, 2014.
7. P. Lamere, P. Kwok, E. Gouvea, B. Raj, R. Singh, W. Walker, M. Warmuth, and P. Wolf. The cmu sphinx-4 speech recognition system. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*, volume 1, pages 2–5. Citeseer, 2003.
8. I. McGraw, C.-y. Lee, I. L. Hetherington, S. Seneff, and J. Glass. Collecting voices from the cloud. In *LREC*, 2010.
9. T. Mehrez, A. Abdelkawy, Y. Heikal, P. Lange, H. Nabil, and D. Suendermann-Oeft. Who discovered the electron neutrino? a telephony-based distributed open-source standard-compliant spoken dialog system for question answering. *Proc. of the GSCL, Darmstadt, Germany*, 2013.
10. L.-P. Morency, C. Sidner, C. Lee, and T. Darrell. Contextual recognition of head gestures. In *Proceedings of the 7th international conference on Multimodal interfaces*, pages 18–24. ACM, 2005.
11. A. Pappu and A. Rudnicky. Deploying speech interfaces to the masses. In *Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion*, pages 41–42. ACM, 2013.
12. D. Prylipko, D. Schnelle-Walka, S. Lord, and A. Wendemuth. Zanzibar openivr: an open-source framework for development of spoken dialog systems. In *Text, Speech and Dialogue*, pages 372–379. Springer, 2011.
13. V. Ramanarayanan, D. Suendermann-Oeft, A. V. Ivanov, and K. Evanini. A distributed cloud-based dialog system for conversational application development. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 432, 2015.
14. D. Schnelle-Walka, S. Radomski, and M. Mühlhäuser. Jvoicexml as a modality component in the w3c multimodal architecture. *Journal on Multimodal User Interfaces*, 7(3):183–194, 2013.
15. M. Schröder and J. Trouvain. The german text-to-speech synthesis system mary: A tool for research, development and teaching. *International Journal of Speech Technology*, 6(4):365–377, 2003.
16. A. Sciutti, L. Schillingmann, O. Palinko, Y. Nagai, and G. Sandini. A gaze-contingent dictating robot to study turn-taking. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, pages 137–138. ACM, 2015.
17. P. Taylor, A. W. Black, and R. Caley. The architecture of the festival speech synthesis system. 1998.
18. J. Van Meggelen, L. Madsen, and J. Smith. *Asterisk: the future of telephony*. " O'Reilly Media, Inc.", 2007.
19. A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing Journal*, 27(12):1743–1759, 2009.
20. Z. Yu, D. Bohus, and E. Horvitz. Incremental coordination: Attention-centric speech production in a physically situated conversational agent. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 402, 2015.
21. Z. Yu, D. Gerritsen, A. Ogan, A. W. Black, and J. Cassell. Automatic prediction of friendship via multi-model dyadic features. In *Proceedings of SIGDIAL*, pages 51–60, 2013.
22. V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J. Hazen, and L. Hetherington. Juplter: a telephone-based conversational interface for weather information. *Speech and Audio Processing, IEEE Transactions on*, 8(1):85–96, 2000.