INTERSPEECH 2016: Show & Tell Contribution
September 8–12, 2016, San Francisco, USA

A Multimodal Dialogue System For Air Traffic Control Trainees Based On Discrete-event Simulation

Luboš Šmídl, Adam Chýlek, Jan Švec

Department of Cybernetics,
University of West Bohemia, Pilsen, Czech Republic
[smidl,chylek,honzas]@kky.zcu.cz

Abstract

In this paper we present a multimodal dialogue system designed as a learning tool for air traffic control officer trainees (ATCO). It was developed using our discrete-event simulation dialogue management framework with cloud-based speech recognition and text-to-speech systems. Our system mimics pilots in an air traffic communication, allowing the ATCOs to practice a control of a virtual airspace using spoken commands from air traffic control English phraseology.

Index Terms: dialogue system, dialogue management, spoken language understanding, air traffic control

1. Introduction

The dialogue system is designed for an interactive air traffic control officer training. Such training comprises ATCO-to-pilot communication lessons, where the ATCO controls a virtual airspace in order to learn rules and phrases [1] that apply to an air control environment. Nowadays, the ATCO communicates with several human pseudo-pilots that respond to commands or initiate the communication based on a time plan created by an instructor. Pseudo-pilots process ATCO's commands into an input for virtual aircraft that is visible on both pseudo-pilot's and trainee's radar screens. [1] Each of the pseudo-pilots often handles more than one aircraft. This results in confusions when an ATCO hears the same voice from different aircraft. ATCO trainees can also get used to a noise-free environment during the training and have to face additional stress when confronted with noises in real radiotelephony communication.

Our goal is to provide means of communication for ATCO trainees, mimicking a real air traffic communication with pilots without the actual need for human pseudo-pilots. This is achieved by a dialogue system with automatic speech recognition (ASR), spoken language understanding (SLU), text-to-speech synthesis (TTS) capability and graphical user interface (GUI) (Fig. 1). The SLU uses semantic entity detection [2] to extract semantic meaning from ATCO's utterance. Such entities are *commands, flight levels, communication frequencies, headings, clearances*, etc. Additional noises and the variance in TTS voice accents brings the training closer to reality than the noise-free environment with human pseudo-pilots.

The dialogue system is multimodal, although there is an emphasis on the spoken aspect of the training. The ATCOs can interact with a GUI (Fig. 1) in their browser in order to view simulated radar screen, pause and resume the simulation and view additional information about aircraft and the goals of the exercise. The instructors can interactively change flight scripts and other properties of simulated aircraft. After the training session, there is an evaluation phase where the ATCO and the instructor can go through a replay and statistics of that session.

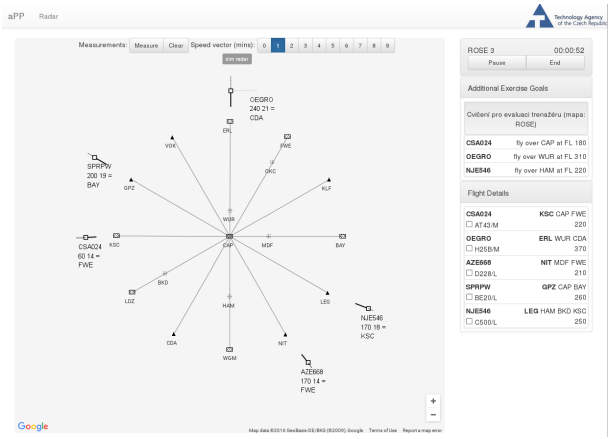

Figure 1: The graphical user interface presented to the ATCO

2. System structure

Our system consists of 3 parts (Fig. 2): the dialogue management system (automatic pseudo-pilot, aPP), the TTS and ASR server (SpeechCloud) and the virtual airspace simulator (ATG).

2.1. Dialogue management framework

Our dialogue framework uses process-based discrete-event simulation framework consisting of simulation processes, time, events and resources. It is using a slightly modified version of SimPy [3], a framework written in Python with simulation processes defined as generator functions.

The execution of the processes can be paused with a *yield* statement. When the coroutine exits, the framework gains control, updates its state (e.g. simulation time, a set of events) and then activates another coroutine based on current event set and simulation time. The coroutine in a paused state naturally cannot react to any events until the framework activates it again.

The simulation framework's decision is based solely on the events the coroutines yielded. These events can be conditioned by either time, method call or availability of a resource.

The acquisition of the simulation resources is also governed by the simulation framework. This means that the coroutine requesting particular resources has to yield a request event causing it to pause and return the control to the framework. The framework checks whether the resource is available (i.e. the ca-

[1] An interactive demo is available at `https://itblp.zcu.cz/` and a video of a short session can be found at `https://youtu.be/01Ev0tle288`.

pacity of the resource is not depleted) and if so, activates the requesting coroutine. If the resource is not available, the coroutine stays paused until used coroutines release the resource.

2.2. Dialogue manager

The dialogue manager (DM) uses all the components of the discrete-event simulation. The simulation processes represent real-world, as well as virtual entities (e.g. ATCOs or aircraft in a virtual airspace). These processes are activated upon meeting corresponding simulation event's conditions. After activation, their inner state is changed reflecting the designed behaviour. Then another simulation event is generated and the process is suspended until the event is activated again (i.e. its conditions are satisfied). The change of the process' state needs to be immediate.

Each aircraft process waits for two kinds of events: one conditioned by an update of the radar and the other by user's utterance towards the aircraft. The virtual airspace is simulated on a standalone server (ATG) and the updates occur periodically, causing the processes to activate, change their state according to the radar data (parsed from an XML the server exposes) and pause, waiting on either of the two events to activate again. In reaction to the event of user's utterance, the process changes its state according to semantic entities present in the utterance and determines the actions that should be taken. For example, upon receiving a message with semantic entities {*callsign: CSA024*, *command: climb*, *flight level: 210*} the system generates a request for a response "climbing to flight level two one zero, CSA024" via TTS and sends a command to the ATG to simulate the climb of the aircraft to the flight level 210.

The ASR and TTS modules are represented also as simulation processes in the framework. These modules are cloud-based services that communicate with the DM via WebSockets by JSON messages. The simulation events of these processes are conditioned by receiving specific JSON messages from those systems. For example, the ASR process waits for events that succeed upon receiving messages that mark the beginning of the recognition, the end of the recognition or those that contain recognized utterances together with semantic entities from spoken language understanding subsystem. These processes also compete for a single simulation resource (with one simultaneous allocation permitted) in order to mimic the usage of a shared radio channel.

2.3. Automatic speech recognition

Our LVCSR system [4] uses an acoustic model with 8kHz sampling rate, three-state HMM models with 2000 states and 16 GMM per state. It was trained from 160 hours of pilot-to-ATCO communication mixed with 460 hours of LibriSpeech data [5]. A language model was created from a mixture of pilot-to-ATCO communication transcriptions, air traffic control phraseology and ICAO spelling alphabet.

The output of the ASR is a word lattice of hypotheses.

2.4. Spoken language understanding

The lattice from ASR is passed to a semantic entity detection algorithm [2] to create a sequence of n-best semantic entities. These entities are defined by expert-made grammars.

The dialogue manager parses this sequence using expert-made rules and determines the actions that should be taken. Such actions include TTS requests, commands for simulated aircraft or changes of the manager's internal state.

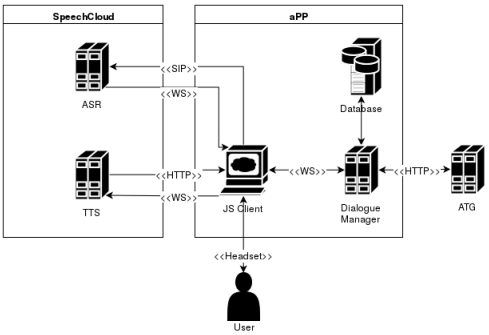

Figure 2: The structure of our system. ATG is a virtual airspace simulator and aPP is the dialogue management system.

2.5. Text to speech

The dialogue system uses a unit selection TTS with speech corpus of several voices created specifically for this task. Domain-specific source texts for the TTS corpus were generated [6] from the air traffic control phraseology and lists of air carriers, waypoints and ICAO spelling alphabet. The corpus consists of voices from native as well as non-native speakers with strong accents. Additional noises can be superimposed on the signal in order to emulate the radiotelephony communication channel and to add ambient noises present in a pilot's cabin.

3. Discussion

The system, as evaluated by 5 ATCOs with 5 to 42 years praxis, is sufficient for basic ATCO training. Complex scenarios with emergencies and other non-standard states are not suitable for our approach because they involve phrases outside of the air phraseology domain. The additional noises in TTS and the diversity of voices were considered to be an improvement over traditional training methods. The ASR accuracy is 91.40% and together with SLU module can process more than 95% utterances correctly.

4. Acknowledgements

This work was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports.

5. References

[1] C. Kaufhold, V. Gamidov, A. Kießling, K. Reinhard, and E. Nöth, "PATSY - it's all about pronunciation!" *Interspeech*, pp. 1068–1069, 2015.

[2] J. Švec and L. Šmídl, "Semantic Entity Detection in the Spoken Air Traffic Control Data," in *Speech and Computer*, 2014, vol. 8773, pp. 394–401.

[3] K. Muller and T. Vignaux, "Simpy: Simulating systems in python," *ONLamp. com Python Devcenter*, 2003.

[4] A. Pražák, J. V. Psutka, J. Hoidekr, J. Kanis, L. Müller, and J. Psutka, "Automatic online subtitling of the Czech parliament meetings," in *Text, Speech and Dialogue*. Springer, 2006, pp. 501–508.

[5] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

[6] M. Jůzová and D. Tihelka, "Minimum Text Corpus Selection for Limited Domain Speech Synthesis," in *Text, Speech and Dialogue*, 2014, pp. 398–407.