LANGUAGE TECHNOLOGY IN AIR TRAFFIC CONTROL

Margret Dora Ragnarsdottir, Helga Waage, Hex Software, Reykjavik, Iceland.
Ebba Thora Hvannberg, University of Iceland, Reykjavik, Iceland.

Abstract

Voice communication is a volatile part of Air Traffic Control (ATC). According to research, on average one miscommunication happens every hour per radio frequency where there is frequent communication such as in TRACON. ICAO puts great emphasis on improving communication in ATC. This paper proposes that a language technology system (LTS) can make communication between controller and pilot more reliable and efficient, thus improving safety in aviation. An LTS can for example detect readback errors. It can also directly feed data from the voice recognizer on XML (eXtensible Markup Language) form into a flight data processing system or interact with it as we show. By interviewing air traffic controllers and studying the literature, we identified several examples of use of language technology in ATC. As an example we take a system to support controllers in their work by making the LTS give warnings when discrepancy is found in the communication between controller and pilot. This system is not meant to control the airspace autonomously. Latest advances in language technology have enabled the development of such a system. The functionality of the proposed LTS is described using scenarios and sequence diagrams. A demonstration conversational agent using Hex Technology was implemented. A Wizard of Oz usability test was administered to seven controllers. Their attitude to the agent was positive and indicates that there is reason for further research. The performance and error logs of the agent and voice server were analyzed and give guidance on further development of a fully functioning language technology system for air traffic control.

Environment

In ATC communication, information flows between the controller and pilot to make sure that the pilot receives the instructions from the controller and conforms to them. It is imperative to the safety of the aircraft and other aircraft in the airspace that the information entered into the flight management system and the flight data processor is the same. The communication is most frequently direct between controller and pilot as described in Figure 1. There are two persons and two systems that all need to be synchronized, through three links of communication:

A. Communication from the controller as he enters information into the Flight Data Processing System (FDPS).
B. Radio communication between the controller and pilot.
C. Communication from the pilot as she enters information into the Flight Management System (FMS) of the aircraft.

Each link in the communication is a safety hazard for it opens up the possibility of the information being corrupted. Things are furthermore complicated in the oceanic environment where the communication is mediated through high frequency (HF) radio. For HF communication special radio stations are in place and the controller/pilot communication go through a third party radio operator. This adds links to the communication. Continuing with the alphabet ordering from above, the communication links in this scenario are as shown in Figure 1:

D. Communication between Air Traffic Controller and radio operator. The messages are most frequently sent between them as text messages via a computer system (as shown) or, rarely, vocally via radio/phone (not shown).
E. Communication between radio operator and his workstation as he enters and receives information.
F. Communication between radio operator and pilot via HF radio.

Above we have described the environment we wish to analyze. The study presented in this paper was the subject of the first authors MSc thesis at the University of Iceland, spring 2003 [13].

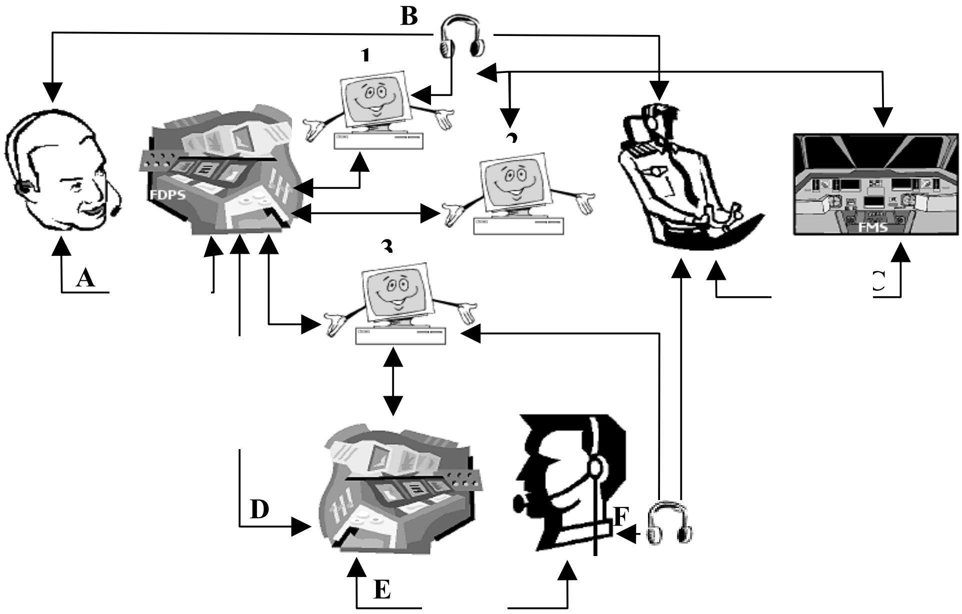

Figure 1. Air Traffic Controller/Pilot communication

Proposed Use of Language Technology

In any system that is intended for use, the people who will use the system must be made a part of the design from the beginning [20]. We are fully committed to User-Centered-Design and with that in mind we sought out the opinion and valuable help from professional Air Traffic Controllers right from the outset. The following eight functionalities for an LTS were identified through interviews with controllers and extensive literature research (no particular order):

- Listening for information that is *missing,* for example making sure that there is readback and that *all* information is read back.
- Listening for information that is *wrong*.
- *Entering* information directly into flight data management systems.
- *Logging* the communication.
- Relaying the information between controller and pilot using the *appropriate mode of communication*, especially in airspace of aircraft with mixed equipage.
- Relaying the same information in text and voice for *redundancy*.
- Serving as a *backup* for data link communication.
- Assist with *training* – the LTS can simulate communication between controller and pilot (such as described in [18]).

Detailed analysis of voice communication in ATC shows that in a little less than 1% of transmissions there was some sort of miscommunication. Even though this does not sound like a real threat to air traffic this means that every hour at least one miscommunication is made on average on a single frequency where there is heavy communication such as in the tower. Note also that these are reported incidences. Most mistakes are caught on the fly and are never reported [6] [7]. Any effort to try to reduce these errors will improve safety in aviation.

Summarizing the functionalities above, language technology could support the communication by (numbers and characters shown in Figure 1):

1. Analyzing the communication and make sure that the there is no discrepancy. This enhances the **B** communication link.
2. Analyzing the communication and entering the information into the Flight Data Processing System and/or the Flight Management System. This eliminates the **A** and **C** communication links.
3. Taking over the communication between the Air Traffic Controller and pilot. This eliminates the **D** and enhances the **E** communication link.

Based on this analysis we designed and implemented a limited prototype that we administered in a Wizard of Oz usability test [2].

The purpose of the test was to demonstrate the functionality of a language technology system in ATC to professional air traffic controllers to get their reaction. This would indicate whether there is reason for further research. The limited scope we have selected for this paper is a system that checks the communication for discrepancy according to item 1 described above and shown in Figure 1. Furthermore, we decided to design the trial for the oceanic environment (OE) because:

1. The communication in the OE is very *formal and structured*.
2. The communication in the OE is *not time critical*.
3. *Recent developments* in the OE where communication is being rethought. This calls for new ideas to be brought into the discussion.

This field of ATC is chosen as an example only. If this system proved to be truly supportive in the OE we can consider applying it to the more time-critical fields such as approach and landing. But what do we mean by language technology how do we propose it is used in ATC?

Language Technology

Language technology deals with how computers can process and use language. This has many practical applications such as automatic translation, or natural language interaction with users. Natural Language Processing (NLP) is the technology of making computers 'understand' natural language. For it to be able to understand speech we use Automatic Speech Recognition (ASR). To make it speak we have Text-To-Speech (TTS) generators [12].

Language technology in ATC

Applying language technology in the ATC domain requires us to put certain requirements on the LTS beyond those that are used commercially in non-safety critical fields. In this case we must require the system to:

- *Be speaker independent*. The system must be able to understand all the people speaking in the airspace.
- *Be highly accurate and reliable.* It is important to set a level of acceptable recognition that must be achieved at minimum before any system using ASR in ATC can be used effectively. If the recognition is worse than human it will be more of a disturbance than support.
- *Be very responsive.* A system made for time critical situations like ATC cannot slow down the communication and has to deliver warnings as soon as discrepancies are found.
- *Listen for keywords*. Listen for keywords in the input will allow for more variation in input and increase performance. This is supported by the fact that ATC uses structured communication, especially in the OE. The vocabulary in ATC has been studied and a corpus collected [10] [19].
- *Use 'grammar' to aid recognition*. The structure of ATC communication makes it easy to device grammar to improve the recognition rate of the ASR. Grammar is used to give context in VoiceXML, (see chapter 'VoiceXML' below).

Some research has been done on how language technology can be made useful in the ATC domain. For example, Schaefer has researched using language technology in training of Air Traffic Controllers [15]. In 2002 an LTS was implemented in training simulators in the United States [18].

Below we will discuss how each of these requirements are achieved with language technology.

Natural Language Processing (NLP)

An LTS designed for this environment does not have to understand the meaning of utterances as a dictation system is forced to in order to function. It only needs to understand whether certain information is being conveyed or not, not what that information entails. It suffices thus that the system listens for keywords. Another argument to support keyword recognition is that even though the communication is formal it is often executed differently. This effectively means that in this environment a system that listens for the correct information but is not stuck in the formality of it is needed. A system that understands keywords [13].

The lowest level of NLP is lexical analysis. With a dictionary the computer can scan through text and make sure that every word in it is also in the dictionary. The lexical system does not understand the relationships between the words or the meaning of them and will allow the users to make grammatical errors. Not very sophisticated maybe but very useful for example in spell checking [11]. This is also enough to design an LTS that only need to understand keywords to be able to react sensibly to the user input.[9]. As mentioned above, the vocabulary of ATC has been studied and a corpus exists [10] [19].

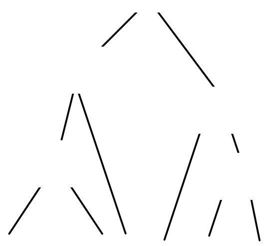

Figure 2. Syntactic analysis - Analyzing a sentence into grammar units (GU)

Moving to upper levels of language processing we come to syntax. In many languages the rules of grammar and their exceptions are well documented. Grammatical knowledge can easily be formalized and therefore it is possible to write a computer program that knows and uses it. To decipher the input, the systems build a parse tree similar to the one in Figure 2. The parse tree shows how the words in the input constitute a sentence from the grammar that the parser knows. More than one legal parse tree is often available for an input. If the parser is required to select between them it commonly uses probability based on analysis of how the language is used and which of the trees it found to be most common. Information like that are collected in language corpuses [12] like the one made by Ward [19].

Automatic Speech Recognition (ASR)

Automatic Speech Recognition (ASR) is the technology used to translate speech into text that can then be analyzed by NLP. The human voice has a very broad range of loudness, timbre and pitch. The ASR needs to adapt to this and so focuses on the shape of the speech wave and its frequency. A related complication is that pronunciation varies widely, especially for languages that are spoken as commonly as English and even within individual dependent on how he feels and his health (think of someone with a bad head cold) [12]. This is especially relevant in ATC where the common language is English but frequently not native to the speaker. An ASR can be trained to listen to only one user. This will increase the recognition level substantially. However, this also limits the usage of the system and is not applicable in many LTSs. This is the case in ATC; the ASR has to be able to understand everyone that it listens to [12].

To analyze the speech, the ASR records it (see Figure 3). Then it matches the recorded speech sounds to phonemes. Phonemes are the smallest units of language that bear meaning and words can be described by the phonemes that comprise it. From this matching it may be possible to build many words. If the ASR is required to return only one result it can use probability, context or grammar [11] [12]. In VoiceXML grammar is used when LTSs notify the ASR about keywords that it should be listening for (see chapter 'VoiceXML' below) [1].

Most commercial ASRs will give results with recognition rate, which assesses the likelihood that the given result matches the correct words. This is important information that can help analyze the performance of the ASR. If the LTS that is receiving the result is only looking for one result, the ASR chooses the most likely candidate and sends it to the underlying LTS with the recognition rate. Many ASRs can be made to reject any result that has recognition rate below a given value. This will, on one hand, lower the number of false recognitions, when the assumption about what it heard is simply wrong, by the ASR. On the other hand, it will also increase the number of false rejections, when it heard correctly but was not very certain about it [1].

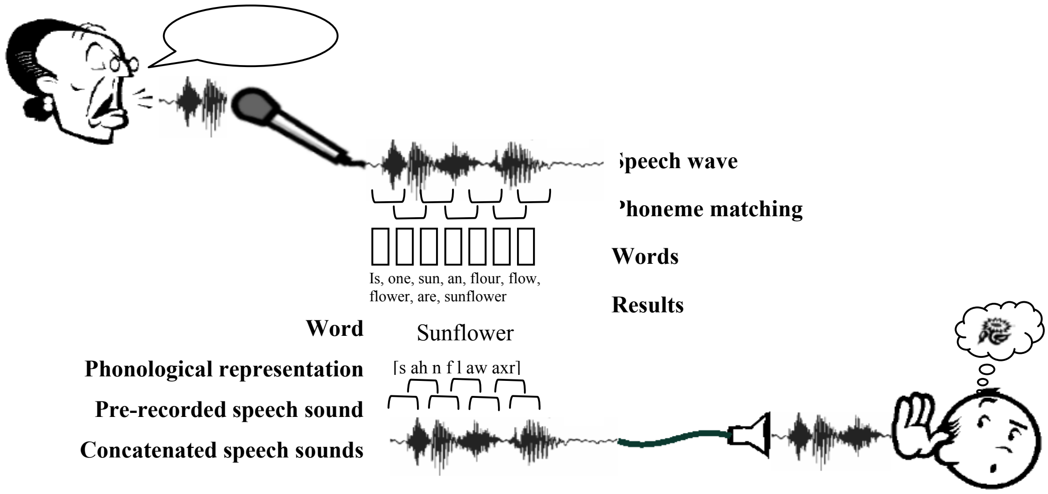

Figure 3. Speech to text and back to speech

Current commercial ASRs boast of 95-98% recognition rate in single/double word trials, down to 55-80% in continuous speech [4] [5]. A lot of research has been done into human speech perception, which place it, around 96 and 99% accurate, an example of such research that is very relevant here is a study done by Cardosi which finds that between 0.73% and 3.36% of all communication is incorrectly or incompletely recognized in ATC communication, dependent on the complexity [6]. Because of the formality and limited vocabulary of ATC and with the correct setup and careful testing this level of recognition can be achieved.

Text-to-speech (TTS) Generation

In Text-To-Speech generation text is translated into speech. There are two ways of generating speech which typically is made up of pre-recorded speech [12] [14][17].

The coarsest form of pre-recorded TTS generators has a pre-recorded limited vocabulary and then pastes together the words to form sentences. This is very limiting for the application of the TTS [12] [14][17].

More sophisticated generation is based on prerecorded input, which is then cut into very short segments that represent phonemes. The function of the TTS generator is to map the letters of the words to the appropriate phoneme for pronunciation. If we concatenate phonemes into a string of sounds the result will not sound very smooth and the flow of speech will be greatly compromised. This contributes to people feeling uncomfortable using it. It has been shown that to make the voice sound more real and to make it flow convincingly the phonemes must be cut down into smaller pieces called speech sounds. They are then concatenated from the mid speech sound before it and until the mid speech sound after it behind it i.e. two speech sounds (diphones) at a time [2] [12] [17].

The first step in turning text into speech is to write the word out phonologically like we have done in Figure 3. Here characters are used to represent speech sounds. The TTS generator then maps the representation to the appropriate sounds and plays for the user [12].

VoiceXML

VoiceXML is based on eXtendable Markup Language (XML). XML is designed to represent data but has many advantages that have been widely embraced by the IT community. The biggest advantage is that it is a platform independent language and thus it can serve as a means of communication between applications written in different programming languages for different platforms.

VoiceXML has made development of voice-based applications easier by placing all processing of the application on the receiving end of the call. The caller only needs to have the ability to send and receive sound [1].

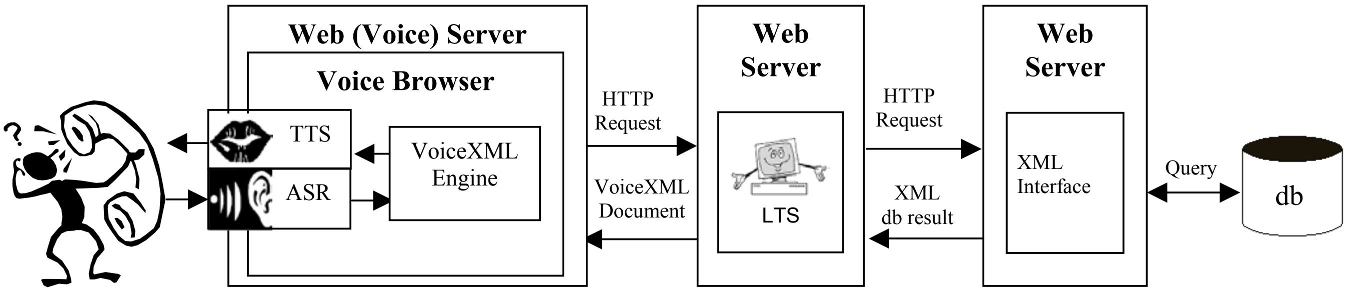

Figure 4. Voice XML system architecture (adapted from [1]).

The VoiceXML scripts are run in a voice browser, the same way HTML scripts are run in web browsers. A user accesses the LTS by using any voice-carrying device such as a phone. When the user calls the voice browser sends a HTTP request to the host of the LTS to request what it is supposed to show the user, in the same way as a web browser. The web server returns a VoiceXML document, which the browser shows the user by playing either a sound file or generated speech from the TTS generator. The voice browser hosts both the ASR and TTS (see Figure 4) [1] [16].

ASR is the key to allowing speech as an input. In ATC environment the ASR must understand every speaker. This is facilitated by grammar. The LTS includes grammar in the VoiceXML document returned to the voice browser. The grammar describes the legal vocabulary for the context and thus what the ASR should be listening for. That way, the ASR does not have to take the speech input and try to match it to every word in its dictionary, but only to the grammar given by the LTS. If it does not fit any of the words there it is irrelevant anyway. If the input fits the grammar the ASR returns the input to the LTS, any other input is irrelevant to the application anyhow. Thus the ASR rejects it and the LTS can request that the user repeat his utterance [1].

Grammar is used in LTSs that listen for keywords rather than the meaning of a whole utterance. The LTS knows what is to be expected next in the communication and instructs the ASR to listen for it. As an example it knows the which parts make up a status report and so the grammar would include the words to be expected in a status report for the ASR to listen for [1].

The interaction between user and system is two-way and so the voice browser receives from the LTS what it is supposed to say to the user as a response either a sound file or text with instructions on how it is supposed to be read by the TTS [1].

VoiceXML communication is session based. Each time a conversation is initiated by calling in, a session is started and it is ended only when the user or the voice browser hangs up. This means that it is easy to have the LTS remember where the controller and pilot are in the communication. If these variables are saved to a database, the LTS can remember what has been said and use it later, for example to check a status report against the last report. Figure 4 describes how VoiceXML applications can interact with databases. The LTS uses the same technique used to display information from a database on a web page. The LTS sends an HTTP Request to a web server that queries the database and return the result to the LTS [1] [16].

Below we will continue by discussing how LTSs can be incorporated into the current ATC system in Iceland.

System Architecture

The Flight Data Processing System (FDPS) is a specially designed ATC system for the Reykjavik ATC Center made by Tern systems (www.tern.is). It manages all the data on aircraft in the airspace and gives an overview of the status. The controller can view the flight plan for each aircraft on digital flight strips and view where it currently is located either on a radar or situation display. Conflict probe and text communication modules are included. The interaction with the radio operator takes place through the text communication module. The controllers can manipulate the data on digital flight strips just as they did with their paper predecessors. For further information on the FDPS, see www.caa.is.

The system architecture for the system we are proposing for the Icelandic ATC environment is based on the VoiceXML technology (described in Figure 4) and is shown in Figure 5. Note that for this environment we are using radio rather than telephony communication. Therefore we must allow for radio communication to enter and leave the system. More importantly, for any LTS to work in this environment it will have to interact with the FDPS. Since there are many very different systems involved XML should be used as a bridge between components. The LTS makes a request to the web server that then queries the FDPS databases and returns the result in XML format (see Figure 5). With this architecture the LTS will for example be able to request a flight plan for a certain aircraft directly from the FDPS database.

Below we will discuss the design of the LTS seated prominently in the center of our system architecture figure below and the implementation of a limited prototype that was demonstrated to professional Air Traffic Controllers.

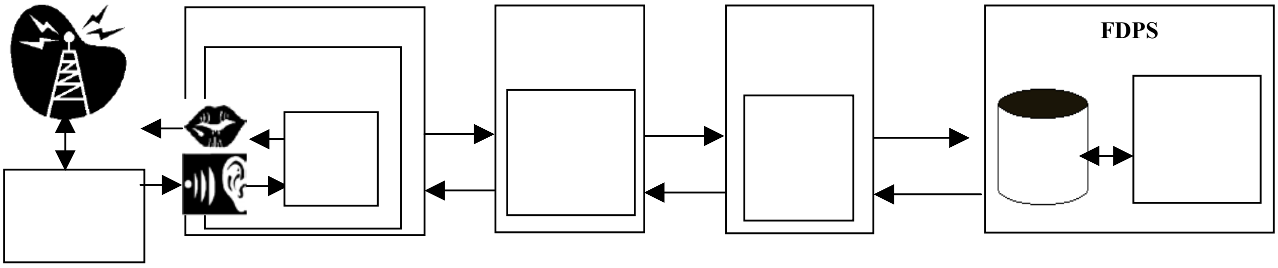

Figure 5. System architecture including the language technology system (LTS).

Hex Agent Technology

Conversational agents are examples of LTSs. They listen to, understand and can take part in natural language conversations. Hex Agents are conversational agents. They are run on Hex Agent Server and using the Hex Agent Creator it is possible for an agent designer to implement an agent with only 20% of the effort that is needed to code a conversational agent in VoiceXML. For further information on Hex Software see www.hex.is.

Hex Agents are created in the Hex Agent Creator, a graphical user interface that makes it possible for agent creator to focus on the design rather than the programming of the agent. The Creator builds the agent, which is run on the Hex Agent Server. The Hex Agent Server then generates the VoiceXML scripts and grammar for the voice browser.

The foundation of Hex Agents is the template. The templates describe what triggers a particular response with the agent and is the basis of the agent's grammar. The templates can be dynamic; that is the agent creator defines a template that is a call sign. The relevant call sign is then fetched from the FDPS runtime. That way the grammar would always include call signs of every aircraft in the airspace because they are likely to be heard.

When a template is triggered the agent responds appropriately. The agents have various ways of responding. In ATC a vocal response is disrupts the flow of communication and thus should not be used except in emergency. However, the agent can also input what he hears into the FDPS or trigger a warning on the air traffic controllers' monitor if it finds anything suspect.

As an example a template could include just a call sign. When a call sign is heard the agent fetches the relevant data from the FDPS. Then it listens to the communication, comparing for example a status report to last report and if everything is in order, sends a little OK signal to the controller.

A demonstration Hex Agent was designed for the purposed of administering a trial; the ATCAgent. The functionality was designed by analyzing typical air traffic control tasks using scenarios and UML modeling as described below. The ATCAgent listens for the users inputs and responds accordingly. That way it simulates the functionality of the system we are proposing. A simulation is sufficient to administer a Wizard of Oz usability test. The trial is described in the next chapter.

Trial

In Baum's Wizard of Oz the wizard fooled his citizens to thinking he had superpowers by using smoke and loudspeakers. In the same way, a simulation of an LTS can make the user believe that they are using a fully functional application if the simulation is convincing enough.

Following the Wizard of Oz method the demonstration was only administered to a limited number of controllers. The participants acted out scripted conversations with the agent and the agent responded to simulate the functionality [2]. That way we demonstrated the functionality of the system to the participants in order to for them to understand the system and give feedback on the idea of using LTSs in ATC communication. The trial was primarily done to look into three things:

1. *Correctness* of the ATCAgent was measured with the recognition rate.
2. *Reliability* of the ATCAgent was measured with the hit/miss/false alarm rate.
3. *Satisfaction* of the Air Traffic Controllers was measured with a questionnaire.

The above measurements were collected during the ATCAgent Trial from the Voice Server, Hex Agent System and the Air Traffic Controllers themselves respectively. The results were analyzed to show how the controllers reacted to the system and how the system reacted to them. This made it possible for us to identify things that should be emphasized in further development.

To define the functionality for the ATCAgent we used scenarios and Unified Modeling Language (UML) sequence diagrams [3] [8].

Scenarios

The trial used a few scenarios from the OE and is designed for case 1 in Figure 1. Here we will describe how we analyzed one scenario to define the functionality of the ATCAgent.

The characters in our scenarios are:

- *Paula*, pilot of Fairline (FAL) 904.
- *Arnold*, air traffic controller
- *Felicity*, controller from an adjacent facility.
- *Roger*, radio operator.

Scenario: Handover – Current situation

This scenario describes how Arnold receives information on incoming flights to his sector and plans how to handle them when they arrive to make sure they arrive safely into the sector.

Please read the text with the UML sequence diagram in Figure 6 and use the numbers in the scenario for reference.

The scenario describes how Arnold and Paula communicate through Roger from the point when Arnold first hears about Paula's flight entering his airspace until Paula has received a new clearance from him, relayed by Roger.

Felicity calls up Arnold. She has an estimate on a flight**,** which is half an hour from entering the airspace.

Felicity (1): "*Good morning, Reykjavik. I have an estimate on Fairline 904*"

Arnold (2): "*Fairline 904 go ahead*"

Felicity (3): "*Fairline 904 estimating 60 north 40 west at 0804, flight level 340, MACH 080, route 62 north 30 west 63 north 20 west 63 north 10 west ISVIG.*"

Arnold enters the information into the FDPS (4) and reads it back to Felicity (5): "*Fairline 904 estimating 60 north 40 west at 0804, flight level 340, MACH 080, route 62 north 30 west 63 north 20 west 63 north 10 west ISVIG.*"

Felicity (6): "*This is correct. Thank you and goodbye.*"

Arnold: "*Thank you.*"

Arnold makes a decision that FAL904 needs to change flight level to 380 and enters it into the FDPS, which it sends to Roger, to be relayed to Paula when FAL904 enters the airspace.

Arnold (7-8):

```
TGC837 031515
FF BICCZZZX
031515 BIRDZOZF
(CLE-FAL904-REYKJAVIK OAC CLEARS FAL904 AT
62N30W CLIMB TO FL380)
```

Roger has all information on flights in front of him on his workstation (GUFCOM) (9). He receives the clearance and prepares for FAL904 to enter the airspace.

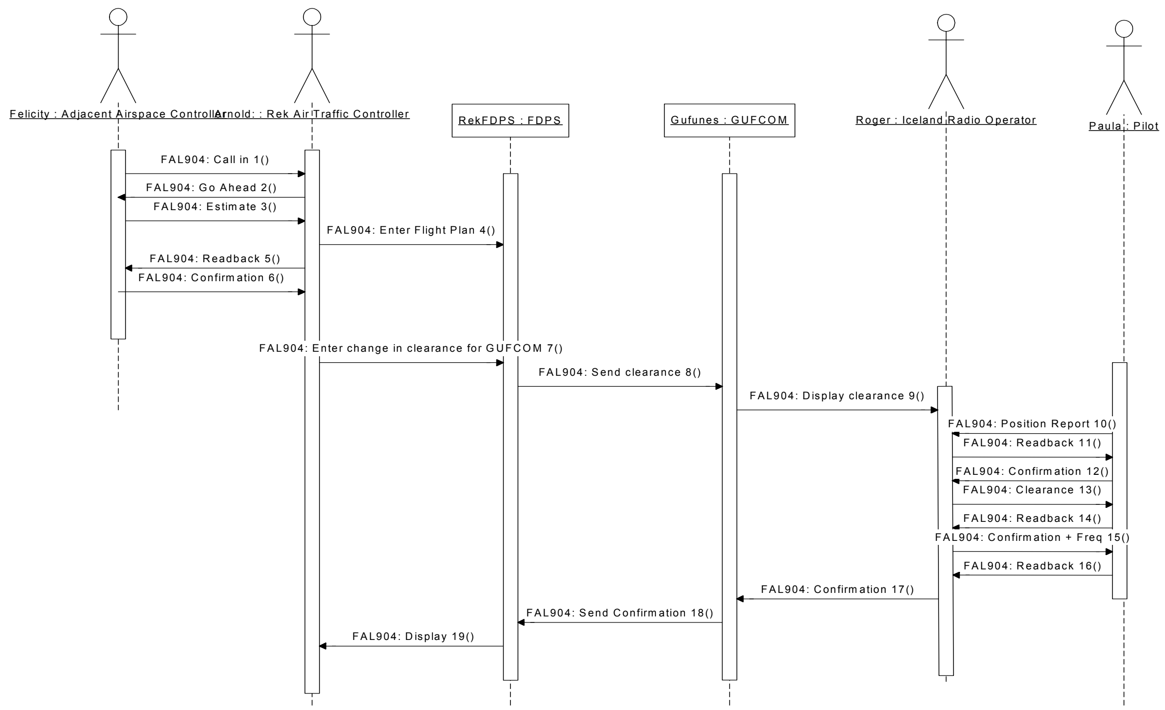

Figure 6. Handover - Current situation

Entering the Icelandic airspace, Paula calls up Iceland Radio (10): "*Iceland Radio, Fairline 904, with you overhead 60 north 40 west at 0803, flight level 340. Estimating 63 north 30 west at 0839. Next 63 north 20 west.*

Roger responds with a read-back (11): "*Roger, Fairline 904, good morning. 60 north 40 west at 0803, flight level 340. Estimating 63 north 30 west at 0839. Next 63 north 20 west.*"

Paula confirms the readback (12): "Affirmative, Iceland Radio."

Roger now gives the new clearance (13): "*Fairline 904 from Reykjavik Oceanic. At 63 north 30 west climb to flight level 380*"

Paula reads back the new clearance (14): "*Fairline 904, at 63 north 30 west climb to flight level 350*".

Roger confirms the new clearance (15): "*Fairline 904 readback correct*".

Roger sends a message to Arnold's FDPS that the clearance has been relayed (17-18):

```
GTC512 031518
FF ENOBZOZX BIRDZQZX KIADXAAY BICCYSYW
031518 BICCZZZE
(RBK-CLE-FAL904-REYKJAVIK    OAC    CLEARS
FAL904 AT 62N030W CLIMB TO FL380
FAL904 RB
VD3JY/1518 5337
```

The FDPS displays to Arnold that the clearance has been delivered (19). Arnold believes FAL904 is climbing to FL380 whereas FAL904 is really climbing to FL350.

Scenario: Handover – The LTS relays the communication

In the next interaction we have added the LTS into the communication as described in Figure 1, case 1. This makes the communication between Arnold and Paula more direct and Roger in effect redundant. After the handover from Felicity, Arnold enters the clearance into the FDPS. The LTS receives the clearance and waits for FAL904 to enter the airspace. The numbers in the text refer to Figure 7.

Entering the Icelandic airspace, Paula calls up Iceland Radio (9): "*Fairline 904, Iceland Radio with you overhead 60 north 40 west at 0803, flight level 340. Estimating 63 north 30 west at 0839. Next 63 north 20 west*".

The LTS gets the clearance for FAL904 from the FDPS (10).

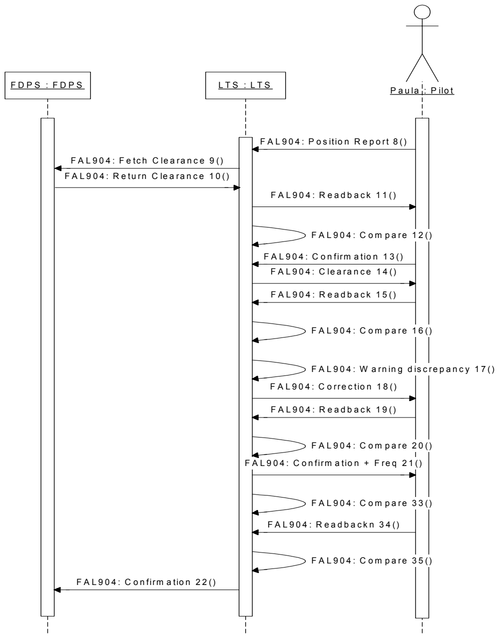

Figure 7. Handover – Includes the LTS

The LTS responds to Paula's report with a read-back (11): "*Roger, Fairline 904, good morning. 60 north 40 west at 0803, flight level 340. Estimating 63 north 30 west at 0839. Next 63 north 20 west*."

Paula confirms the readback (12): "*Affirmative, Iceland Radio*."

The LTS now gives the new clearance from Arnold (13): "*Fairline 904 from Reykjavik Oceanic. At 63 north 30 west climb to flight level 380*"

Paula reads back the new clearance (14): "*Fairline 904, at 63 north 30 west climb to flight level 350*".

The LTS compares the readback to the clearance again to find the error (15).

The LTS corrects the discrepancy.Language technology system (16): "*Fairline 904. Correction, cleared to climb to flight level 380*".

Paula responds (17): "*Fairline 904, climb to flight level 380*".

The LTS compares the readback to the clearance (18) and confirms the new clearance (19): "*Fairline 904 readback correct*".

The LTS inputs the information to the FDPS that the clearance has been confirmed (21-22):

```
GTC512 031518
FF ENOBZOZX BIRDZQZX KIADXAAY BICCYSYW
031518 BICCZZZE
(RBK-CLE-FAL904-REYKJAVIK OAC CLEARS
FAL904 AT 63N030W CLIMB TO FL380
FAL904 RB
VD3JY/1518 5337)
```

The FDPS displays the confirmation to Arnold as before. There is now no discrepancy.

Trial Environment

The above scenarios were used to define the functionality of the ATCAgent. In the trial the the ATCAgent simulated the part of the LTS as described in the scenarios. The ATCAgent did so by taking its part in the script the scenarios were adapted to. In the trial roles were assigned to the participants and administrator, to act out with the LTS. To make sure that the ATCAgent was always following the conversation it always gave a response to the input from participant and controller. This would of course not be acceptable in normal circumstances since it slows down the communication substantially. However, it was more important for this trial to keep everyone synchronized than to keep up the pace of the communication. The ATCAgent then also gave warnings when it did not understand the user and responded appropriately when discrepancies were made in the communication according to script.

The trial was administered to seven Air Traffic Controllers. The seven participants were on average 34 years old with 11 years of experience. The gender distribution was roughly equal (3 male/4 female). Their primary work is in the oceanic or en-route environment. On average each trial run lasted around 30 minutes.

As described in Figure 5, during the trial the ATCAgent was called up over the phone. The utterance of the participants was received by an ASR that matched it, if possible, with known words in the grammar from the ATCAgent. The Voice Browser then sent an HTTP request to the Hex Agent Server with the input. The ATCAgent returned a VoiceXML document with the appropriate response and the grammar for the next input. The answer was delivered using a TTS generator.

Results

In short, the air traffic controllers response gives just cause to believe that further research should be carried out on using language technology in ATC. The system's response to the controllers was also above expectation; it responded appropriately in 97% of the cases. Understandably this is not acceptable in a safety critical environment but from a demonstration agent this result can be greatly improved. The air traffic controllers' feedback indicates what needs to be researched:

First, the LTS needs to be adapted into the ATC environment, most importantly to read and write, data to the FDPS database.

Second, all types of communication between air and ground need to be thoroughly documented. When we have done that, the different types of communication can be analyzed for keywords that the LTS to listen for and recognize. If this is done properly it will make the design of the LTS easier because the flow of the communication defines the LTS. A corpus has been made [10] [19].

Third, we need to make sure that the voice server is set for the environment and that it gives the LTS what it needs. Configuration should be focused most importantly on:

- *Filtering* of radio static, which is not a trivial matter, especially in HF.
- *Types of input* – We need to analyze when to expect long/short correspondence and how to react differently so that it allows the user to take a breath without cutting into the communication when it knows there is more coming.
- *Recognition rate* – Careful testing can show us what is the perfect balance between too many rejects and too many falsely recognized inputs.

Fourth, the design should be able to allow the LTS to stand by without losing the state it is in between communication.

Fifth, there needs to be a way for the LTS to display warnings to the user either through the FDPS or a separate program that can pop up windows or send other types of indications to the users. These indications should be carefully designed in cooperation with the users to be most useful.

Conclusion

To introduce language technology into ATC it is an important first step to develop, as described above, a fully functioning LTS for the oceanic environment. This can serve two equally important functions. First to continue the research of the technology needed to make such an LTS function smoothly. Second to build a reputation for an LTS in this field. Every air traffic controller in training in Iceland begins by controlling the oceanic airspace and then improves his skills to control more complex airspaces such as TRACON or tower. The LTS has to go the same route to build trust with the people it is aimed to support, the controllers. With further development it would be able to assist where it is most needed, that is in time sensitive situations where the information is critical and no time for read backs and double checks i.e. during approach and landing.

When an LTS like this has proven its worth, further research should be done on how artificial intelligence could be incorporated into the system, so that it can make recommendations on controlling the airspace in addition to monitoring the communication.

Acknowledgements

This research was supported by the Icelandic Civil Aviation Administration, the University of Iceland and Hex Software.

References

[1] Andersson, Eve Astrid, Stephen Breitenbach, Tyler Burd, Nirmal Chidambaram, Paul Houle, Daniel Newsome, Xiaofei Tang, Xiaolan Zhu. (2001). *Early Adopter VoiceXML*. Birmingham, UK: Wrox Press Ltd.

[2] Balentine, Bruce, and David P. Morgan. (2001). *How to Build a Speech Recognition Application* (2nd ed.). San Ramon, CA: Enterprise Integration Group Press.

[3] Booch, Grady, James Rumbaugh, and Ivar Jacobsen. (1999). *The Unified Modeling Language User Guide.* Reading, MA: Addison Wesley.

[4] Broughton, Michael. (2002). *Measuring the Accuracy of Commercial Automated Speech Recognition Systems During Conversational Speech.* From the Workshop on "Virtual Conversational Characters: Applications, Methods, and Research Challenges" 29th November, 2002, Melbourne, Australia http://www.vhml.org/workshops/HF2002/papers/broughton , accessed January 2003.

[5] Cane, Jack. (1998). *Comparing Dragon NaturallySpeaking and IBM ViaVoice Gold, ENW International*, http://www.enw-ltd.com/vv-dragonComparison.htm, accessed January 2003.

[6] Cardosi, Kim (1993). An Analysis of En Route Controller-Pilot Voice Communications. Final Report No. DOT-VNTSC-FAA-93-2.

[7] Cardosi, Kim, P. Falzarano, and S. Han, (1998). *Pilot-Controller Communication Errors: An Analysis of Aviation Safety Reporting System (Automatic Speech RecognizerS) Reports*. DOT/FAA/AR-98/17 http://www.volpe.dot.gov/opsad/docs/asrs1.doc, accessed January 2003.

[8] Carroll, John M. (2000). *Making Use: scenario-based design of human-computer interaction*. Cambridge, MA: MIT Press.

[9] Finlay, Janet, and Alan Dix. (1996). *An Introduction to Artificial Intelligence*. London: UCL Press.

[10] Godfrey, John J. (1997). *Air Traffic Control Complete.* ISBN: 1-58563-024-1 http://www.ldc.upenn.edu/Catalog/LDC94S14A.html*, accessed January 2003.*

[11] Hausser, Roland R. (2001). *Foundations of computational linguistics: Human-computer communication in natural language* (2nd ed.). Berlin: Springer.

[12] Jurafsky, Dan. and James H. Martin. (2000). *Speech and Language Processing.* Upper Saddle River, NJ: Prentice Hall

[13] Ragnarsdottir, Margret Dora. (2003). "*Do You Copy?" Using Language Technology to Support Communication in Air Traffic Control.* Unpublished M.Sc. Thesis. University of Iceland.

[14] Rosen, Stuart, and Peter Howell. (1991). *Signals and Systems for Speech and Hearing*. London, UK: Academic Press.

[15] Schaefer, Dirk. (2001). *Context-Sensitive Voice Recognition in the Air Traffic Control Simulation. EEC* Note No. 02/2001. http://137.193.200.177/ediss/schaefer-dirk/meta.html*, accessed January 2003.*

[16] Tober, Eric D., Robert Marchand, and Jim Ferrans. (2001). *The VoiceXML Forum's VoiceXML Tutorials*. Voice XML Forum. http://www.voicexml.org/tutorials/intro1.html*, accessed January 2003.*

[17] Van Santen, Jan P.H., Richard W. Sproat, , Joseph P. Olive, and Julia Hirschberg (eds.) (1995). *Progress in speech synthesis.* New York: Springer.

[18] Voice recognition Trains Next-Generation of Air Traffic Controllers. *SpeechTechnology Magazine*. (2002, August 6). http://www.speechtechmag.com/pub/industry/1029-1.html*, accessed January 2003*

[19] Ward, Karen. (1992). *A Speech Model of Air Traffic Control Dialogue.* Unpublished M.Sc. Thesis. Oregon Graduate Institute of Science and Technology. http://citeseer.nj.nec.com/ward92speech.html*, accessed January 2003.*

[20] Wickens, Christopher D., Sallie E. Gordon, and Yili Liu. (1997). *An Introduction to Human Factors Engineering.* New York, NY: Addison Wesley Longman Inc.