Eleventh USA/Europe Air Traffic Management Research and Development Seminar (ATM2015)

Assistant-Based Speech Recognition for ATM Applications

Hartmut Helmke, Jürgen Rataj, Thorsten Mühlhausen,
Oliver Ohneiser, Heiko Ehr, Matthias Kleinert
German Aerospace Center (DLR),
Institute of Flight Guidance,
Braunschweig, Germany
Firstname.Lastname@dlr.de "ü" = "ue"

Youssef Oualil, Marc Schulder
Spoken Language Systems Group
Saarland University (UdS),
Saarbrücken, Germany
YOualil@lsv.uni-saarland.de;
MSchulder@lsv.uni-saarland.de

***Abstract*——Situation awareness of today's automation relies so far on sensor information, data bases and the information delivered by the operator using an appropriate user interface. Listening to the conversation of people is not addressed until today, but an asset in many working situations of teams. This paper shows that automatic speech recognition (ASR) integrating into air traffic management applications is an upcoming technology and is ready for use now.**

Apple's Siri® or Google's Voice Search® are based on hundreds of thousands of hours of training data. This paper presents an assistant based speech recognition system (ABSR), based on only 40 hours of training data. ABSR uses speech recognition embedded in a controller assistant system, which provides a dynamic minimized world model to the speech recognizer. ASR and assistant system improve each other. On the one hand, the latter significantly reduces the search space of the first one, resulting in low command recognition error rates. On the other hand, the assistant system gains benefits from ASR, if the controllers' mental model and the model of the system deviate from each other. Then the controller cannot rely on the output of the system anymore, i.e. the assistant system is useless during these time intervals. By using ABSR the duration of these time intervals is reduced by a factor of two.

Keywords—Arrival Management (AMAN), Automatic Speech Recognition (ASR), Workload

I. INTRODUCTION

Conversation is a core element of society concerning its further development since centuries. Hence, a significant part of human collaboration is coordinated via voice, especially when complex contexts or meta-concepts are considered. By tracing communication new actors can get an idea of the actual and planned situations and interpret the actions, so that they can easily integrate themselves into this environment. Listening actors can follow the train of thoughts and are able to contribute to problem solving activities in an appropriate manner with own ideas.

Nowadays, people get more and more supported by technical systems like assistant or decision support systems which can be found in nearly every working and leisure environment. Latest applications, like those by Apple (Siri®) [1], [2] and Google (Voice Search®) [3], use ASR as input interface for a direct communication between human and machine to trigger a defined action, as for instance a query. At least in nearly every medium-sized car [4], speech recognition is used to give voice instructions to corresponding assistance systems. However, all these systems still require improvements regarding their recognition rate, necessitating large future investments into ASR technology to receive the impression talking to a real human.

In an air traffic control (ATC) working environment, communication between the involved parties is the most important mean to control the air traffic. Controlling aircraft in the vicinity of an airport is an example of such a working environment in which two working groups communicate, i.e. pilots and controllers. All pilots in the same sector are supported by a dedicated controller (team). They use a unique frequency for communication within this sector. This enables a party line effect, i.e. all actors – excluding today's assistant systems – can create a common mental model of the current situation and of future actions.

Furthermore, the complexity of today's assistant systems steadily increases. On the one hand, computer power and system complexity continuously increases. On the other hand, automation lacks in intuitive interfaces for humans, which would enable a smooth and fluent interaction. An important ability is to follow human communication to react and adapt to a situation and reduce the amount of dedicated interaction between user and machine.

Today, however, communication is still split into two different worlds: one in which humans communicate via radio links, and another in which machines communicate via computer networks. These two worlds are connected by a human machine interface used by humans to inform the machines and vice versa. Intents and plans of both humans and machines are the basis for these two worlds. As controllers are responsible for air traffic control, they sometimes implement plans deviating from those of the automation. If these deviations occur in situations with high workload, the controllers do not have time to inform the assistant system about their strategies and intentions. In these cases, the automation may suggest advisories contrary to the intent of the controller because the support system is not aware of operator induced deviations. Even worse, the operators have additional effort to inform the support systems about their communication. This situation may persist until the assistant system realizes the deviation, e.g. through the analysis of radar data. Hence, the system requires attention from the controllers, exactly when the controllers would urgently need the support of the system due to high workload.

To overcome this situation and to enable air traffic management (ATM) systems to follow the conversation between controller and pilots, Automatic Speech Recognition is an important element of future ATM assistant systems. The ability to listen has to be implemented into the assistant system. This allows following the conversation and to synchronize the intents and strategies of human and machine world.

Crucial for user acceptance is the quality of ASR, especially recognition time and rate. Actual studies in the automotive environment achieve word recognition rates between 78% and 87% [5]. Such rates are far too low for an application in the ATC environment and here need to understand the whole command. Due to the high number of utterances per hour in ATC, too many corrections would be necessary for the controller, which would strongly decrease the acceptance of such a system.

The integration of ASR into ATM systems has been attempted since (at least) the early 90s. We briefly review this prior work in section II. In section III it is shown how ASR has to be integrated with an assistant system, so that acceptable error rates are achieved. This enables common situation awareness without lack of information on the part of the automation and without discrepancies between voice communications and data link information. In other words, the ASR system is not blindly listening, but it is rather aware of the situation and, therefore, can actively anticipate the spoken commands. The recognized commands are then processed and relevant information is extracted to dynamically derive new planning and advisory. This win-win situation between the ASR component and the assistance system results in a more accurate and reliable assistance.

Section V describes the different AMAN support levels offered to the controller during the trials, which are described in section IV. They have been performed with a combination of DLR's arrival manager 4D-CARMA (4D Cooperative ARrival MAnager) [6], the speech recognizer from Saarland University (UdS) and approach controllers from Austrian, Czech and German ANSPs in the Air Traffic Validation Center at the DLR premises in Braunschweig. Section VI describes the first validation results. The last section describes further steps and summarizes the results.

II. BACKGROUND

Human-machine interaction systems have received a significant improvement in their performance in the last decade, leading to more sophisticated human-machine applications. Voice-enabled systems in particular are increasingly deployed and used in many different areas. The most popular use case among these is ASR deployed on most mobile phones. In fact, ASR systems are becoming a significant component in hand-free systems and a cornerstone for tomorrow's applications, such as smart homes [1]. ASR applications can be generally classified to three different categories based on the application purpose:

1. Hand-free command & control: The purpose of these applications is to control a given system by relatively short spoken sentences. This type of application is widely spread, such as in mobile phones, TV sets or car navigation systems [4].
2. Dictation software: These systems mainly target the professional market as their adaptivity is not yet good enough for widely accepted consumer products [8].
3. Spoken dialog systems: These systems benefit from the advances of dialog management research, as well as ASR. Common examples are dialog systems used for train time table consultation [9]. Similarly, Siri® [1] and Google's search by voice [3] are assistant systems that integrate question answering and spoken dialog functionalities.

The ATM world, following and deploying the advances of today's research and technology, is increasingly developing ASR-based applications to provide more sophisticated assistant systems. ASR is a potential extension of many existing systems where speech is the primary mode of communication, such as Arrival Managers (AMAN), Surface Managers (SMAN), and Departure Managers (DMAN). First commercial implementations of an AMAN have been operational at hubs (Frankfurt, Paris) since the early 90s. Today, their application is still limited to the coordination of traffic streams between different working positions (e.g. sector and approach controllers) [10].

Although ASR performance improved significantly in the last decade, it is far from being a solved problem, in particular for large vocabulary applications. Limited vocabulary (in-domain) applications, however, are being more successfully deployed. Moreover, the existence of prior information about the task at hand and the expected spoken sentences summarily referred to as context can significantly improve performance. Early usage of context goes back to Young et al.'s works [11], [12], where they made use of sets of contextual constraints of varying specificity to generate several grammars. These grammars are then consecutively used during recognition until a satisfactory recognition hypothesis is found. Along this idea, Fügen et al. [13] also used dialogue-based context to improve ASR quality in their dialogue system where a Recursive Transition Network (RTN) representing the grammar is continually updated.

The first attempts to integrate ASR in ATM systems goes back to Hamel et al. [14] who described the application of speech technology in ATC training simulators in the early 90s, however with limited success. Schäfer [15] used an ASR system to replace pseudo pilots in a simulation environment. He used a dynamic cognitive controller model. He, however, does not use an assistant system to dynamically generate the context so that assistant system and ASR improve each other. Dunkelberger et al. [16] described an intent monitoring system which combines ASR and reasoning techniques to increase recognition performance: In a first step, a speech recognizer analyses the speech signal and transforms it into an N-best list of hypotheses. The second step uses context information to reduce the N-best list. The approach presented in this paper, however, uses context to directly reduce the recognition search space, rather than only rejecting the resulting hypotheses. The latter approach would only reduce error rate without increasing recognition rate.

These early attempts opened the gate to more concrete and successful applications of ASR in ATC training simulators. The FAA (Federal Aviation Administration) reports the successful usage of advanced training technologies in the terminal environment [17]. The Terminal Trainers prototype, developed by CAASD (Center for Advanced Aviation System Development, USA), is a complex system which includes speech synthesis, speech recognition, multimedia lessons, game-based training techniques, simulation, and interactive training tools. The German air navigation service provider DFS uses the system Voice Recognition and Response (VRR) of UFA (Burlington, MA) for controller trainings since August 2011 [18] to reduce the number of required pseudo pilots in its flight service academy. These systems, however, are being used only for training purposes, due to the ASR standard phraseology limitations. In a more general application of ASR to ATM domain, Cordero et al. proposed to take advantage of the speech-to-text output of ASR systems to perform controller workload assessment, which was consequently used for automatic ATC events detection [19].

Although data link might replace voice communication in the ATC environment, voice communication and data link with their different advantages will coexist for a long time at least in General Aviation. Here voice communication will remain the central means of coordination in the foreseeable future. The agreements coordinated by voice, automatically have to be integrated into SWIM (System Wide Information Management) based on reliable speech recognition. The same applies for different types of on-board equipment of airliners. This supports the coexistence of varying levels of automation in different tightly-coupled subsystems. Furthermore, careful transitions between different levels of automation are more easily possible.

ASR systems generally use the Word Error Rate (WER) metric for evaluation. This metric is defined as the distance between the recognized word sequence and the sequence of words which were actually spoken, referred to as the gold standard (see pp. 362-364 in [20]). WER is defined as a derivation of Levenshtein distance [21]:

$$WER(s) = \frac{ins(s) + del(s) + sub(s)}{W(s)} \qquad (1)$$

Here, ins(s) is the number of word insertions (words never spoken), del(s) is the number of deletions (words missed by ASR), sub(s) is the number of substituted words, and W(s) is the number of words actually said. In ATM, however, the WER is not descriptive enough as metric. One would rather prefer a metric assessing the rate of correctly recognized concepts. It is not important that ASR correctly recognizes “Good morning Lufthansa one two tree descend level one two zero”, but that the concept “DLH123 DESCEND 120 FL” is correctly extracted. The command error rate (CER) quantifies this metric.

The proposed ASR system directly builds on the pilot study conducted by Shore et al. [22], [23]. The goal of this study was to provide a proof of concept for integrating situational context information into ASR for ATC task. Reported results strongly indicate that incorporating context information significantly reduces recognition error rates [24]. He, however, did not consider the problem of dynamically deriving the situational context. The work presented in this paper solves this problem by the “Hypotheses Generator” component, see next section.

III. Assistant Based Speech Recognition

In a pilot study with a limited set of callsigns and commands, Shore [23] reported command error rates below 5%. He used an acoustic model derived from the Wall Street Journal recognition corpus. This was the basis of the AcListant® project started in Feb. 2013 [25]. Due to the considerably larger number of callsigns and commands in this project, initial experiments using the same corpus returned command error rates greater than 70%. In the first project phase many hours of speech samples from real controller pilot communication were recorded and annotated, i.e. every controller utterance was written down word by word. The resulting model already improved recognition rates drastically from 30% to 80%, but error rates were still above 20%. Therefore, we concentrated on providing dynamically generated situational context to improve performance.

Figure 1 describes the concept of assistant based speech recognition (ABSR). An “Assistant System” (in our case the Arrival Manager 4D-CARMA) analyses the current situation of the airspace and predicts possible future states.

The output of the Assistant System, the context, (e.g. aircraft sequences, distance-to-go, minimal separation, aircraft state vectors) is used by the “Hypotheses Generator” component. The “Hypotheses Generator” does not know exactly which commands the controller will give in the future, but it knows which commands have a higher probability than others in the current situation.

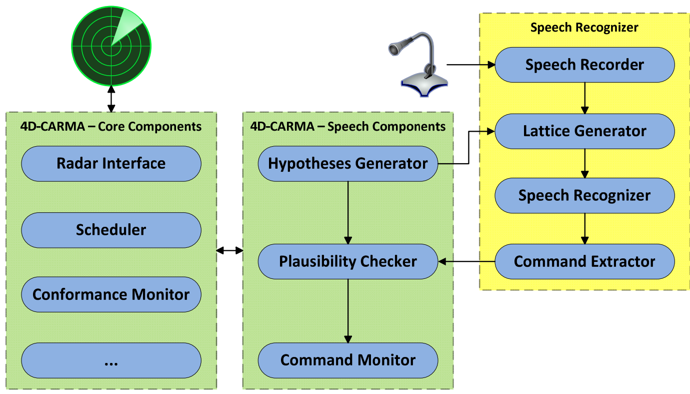

Figure 1. Components of Assistant Based Speech Recognition

These hypotheses are entered into the “Automatic Speech Recognition” block, which itself consists of the following components: the “Speech recorder”, the “Lattice generator”, “Speech Recognizer”, and the “Command Extractor”. A microphone is connected to the “Speech Recorder”, being responsible for recording the speech signal. This is normally a wave file (16 kHz, mono). The lattice generator creates a reduced search space for the “Speech Recognizer” using the output of the “Hypotheses Generator”. For the input “DLH496 REDUCE 230-240” it is of course not sufficient to generate only the two sentences “Lufthansa four nine six reduce two **three** zero knots” and “Lufthansa four nine six reduce two **four** zero knots”. The controller may for example articulate the callsign as “four nine six”, “hansa”, “hansa four nine six” etc. Also, while “speed two three zero” is possible in the hypotheses input, “increase two three zero” is not. The “Speech

Recognizer" transforms the speech signal into a feature vector FT and searches the most probable sequence of words $\widehat{W}$, which is licensed by the generated search lattice.

$$\widehat{W} = \underset{W\,e\,SL}{\arg max}\, P(W \mid FT) \tag{2}$$

As P(W/FT) is often unknown, we use Bayes' Theorem to reformulate the equation:

$$\widehat{W} = \underset{W\,e\,SL}{\arg max}\, P(FT \mid W)\, \frac{P(W)}{P(FT)} \tag{3}$$

We are not interested in the absolute value of the probability. It is sufficient to find the best fitting word sequence $\widehat{W}$. Therefore we can eliminate the a priori probability for the feature vector P(FT).

$$\widehat{W} = \underset{W\,e\,SL}{\arg max}\, P(FT \mid W)\, P(W) \tag{4}$$

The a priori probability of the words P(W) we derive from the language model (grammar). The conditional probabilities P(FT/W) are given by the acoustic model. More details can be found in the master thesis of Anna Schmidt [26].

The most probable word sequence $\widehat{W}$ might be e.g. "***Lufthansa four nine six*** *thank you normal speed however* ***maintain one seven zero*** *knots* ***or greater*** *to six miles final* ***descend altitude tree thousand****…"* We are not interested in every single word of this utterance. We need the relevant concepts, which are marked in bold face. This is the task of the "Command Extractor", which creates from the above example the command sequence "DLH496 SPEED_OR_ABOVE 170, DLH496 DESCEND 3000 ALT" The command extractor also assigns a plausibility value to each extracted command.

On average, the "Lattice Generator" needs 5 seconds to generate a new search lattice. The "Speech Recognizer" needs one second per sentence on average. A rejection time of more than two seconds is, however, not acceptable. As "Lattice Generator" and "Speech Recognizer" run in parallel, the "Speech Recognizer" can immediately start as soon as the controller has finished his utterance. It always uses the latest search lattice.

The extracted commands are sent back to the assistant system, namely to the "Plausibility Checker" component, which also uses e.g. the context knowledge, the plausibility values, and the command hypotheses to reject recognized commands. Commands which are not in the current context are further checked by future radar data if they have very high plausibility values. Otherwise, they are rejected at once. A turn left and a turn right command for the same aircraft in the same utterance is also immediately rejected etc. The "Plausibility Checker" divides the recognized commands into three sets

- Commands (immediately) accepted by AMAN,
- commands (further) monitored, and
- commands (immediately) rejected.

The first two sets are input into the "Command Monitor" component. It tries to verify "commands further monitored" through upcoming radar data. If an aircraft with a turn left command starts a left turn after the command, it is probable that the speech recognizer output was correct. In this case the commands are transferred into the set of "commands, accepted by AMAN". The commands in this set are also further observed and checked against the radar data. If a descend command with an end value of flight level 100 was recognized and the aircraft is not descending after some time or the aircraft passes flight level 90, this command is transferred to the set of "commands rejected". The "Command Monitor" ensures that in the case of ASR failure the system behave in the same way as without a speech recognizer. In this case, the controller's actual utterance is unknown and the assistant system performs e.g. sequence updates solely based on radar data.

Our approach requires the following three main functions:

1. Creation of hypotheses about possible future airspace situations and the corresponding commands (component "Hypotheses Generator" in combination with 4D-CARMA core components)
2. Highly reliable speech recognition based on dynamic update of hypotheses (components of Speech Recognizer Block plus "Plausibility Checker" and "Command Monitor")
3. Updating of assistant systems based on the obtained voice communication information (Using output of "Plausibility Checker" and "Command Monitor")

IV. Validation Trials

The validation process of the AcListant project is implemented according to the European Operational Concept Validation Methodology (E-OCVM) [27]. Therefore, several validation trials have been conducted since the start of the project in spring 2013 in an iterative way. The described results are based on the last but one iteration loop taking place in October 2014.

The basic setup consists of one controller working position and two pseudo pilot stations to handle the air traffic. The simulated airspace is the TMA of the airport Düsseldorf (EDDL) with only arrival traffic being modelled. The controller working position is equipped with an advanced radar screen, which is described in detail in the following section and a speech log screen. Flight information is handled by paper flight strips to reflect the situation at Düsseldorf. The exception is the manual input scenario (see below), where the controllers use computer devices to document their clearances. The speech recognition directly uses the microphone output signal from the controller. To ensure diversity in these trials, the three participating controllers are selected in a way that there are both male and female participants as well as both native German speakers and a speaker from Eastern Europe in the team. Two of the controllers already have experience with advanced speech recognition, one is a novice. To allow a one day effort per test person, the number of runs is limited to six, each of which has a duration of approximately 50 minutes.

As seen in Figure 2, two validation questions are addressed by the trials. The first question addresses the functionality benefits of an AMAN with additional input compared to a simple AMAN. The second one addresses the reduction of controller workload concerning the way the input is delivered to the assistant system. The additional inputs are the communicated advices between controller and pilots. In both cases the first step is the analysis of the baseline, i.e. the situation without any controller assistant system (represented by square 1 in Figure 2). To answer the first question, a run with a standard AMAN without additional input (square 2) and a run with advanced AMAN with additional input created by ASR support (square 4) are conducted in addition to square 1. To answer the second question, additional runs with an advanced AMAN, either with manual input device (mouse and keyboard) (square 3) or with ASR support (square 4) are performed. Manual resp. ASR inputs are used as additional information for the AMAN's planning cycle.

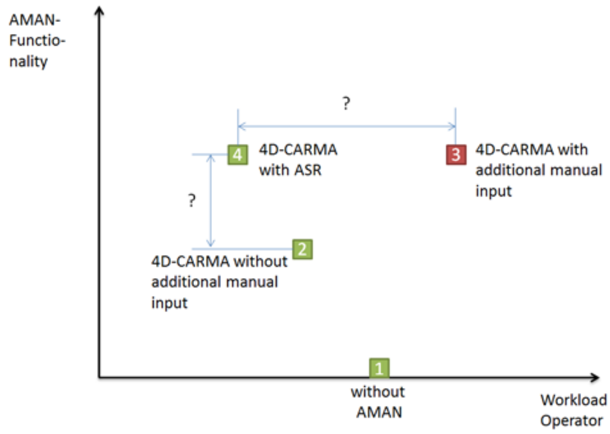

Figure 2. AMAN functionality vs. workload diagram

As the AMAN support is especially useful in situations, which are unexpected and provoke a new sequence, special scenarios are set up. For the first one, an aircraft emergency due to a sick passenger is simulated and for the second one a temporary runway closure is implemented. To reduce the learning effects, the callsigns are changed for each of the runs with the same scenario. Therefore, each controller has to manage the six runs shown in TABLE I. . The numbers reflect the order, in which the trials were set up in October 2014. .

TABLE I. SCENARIO SEQUENCE

Simulation configuration	*Emergency*	*Runway Closure*
Baseline	1	4
Standard AMAN	3	-
Advanced AMAN with manual input	-	5
Advance AMAN + ASR	6	2

The emergency scenario is used to answer the validation question of the functionality benefits of an AMAN, The runway closure scenario is used to address both validation questions, i.e. the functionality benefits and the workload reduction by using ASR instead of mouse and keyboard. For conducting the trails, each controller starts with an additional short training scenario to become familiar with the environment.

V. INCREASED LEVELS OF AMAN-SUPPORT

The four simulation configurations described in TABLE I. offer different levels of support to controllers using DLR's radar display "RadarVision" [28]. The baseline configuration includes a state-of-the-art radar screen showing airspace layout of Düsseldorf. Each aircraft position is shown by a circle with an attached label, which shows callsign, aircraft type, altitude and speed. A speed vector (predicted aircraft position in e.g. 90 seconds) can be shown to the controller on request. The support of the AMAN configurations stepwise increases, starting with configuration 4D-CARMA (square 2 in Figure 2). Supporting data calculated by the AMAN contains touchdown sequences, as well as four-dimensional trajectories and derived distances. On the timeline every aircraft is displayed as a label with its sequence number, calculated distance-to-go [29] (nautical miles until threshold also known as "miles-to-fly"), weight class and callsign.

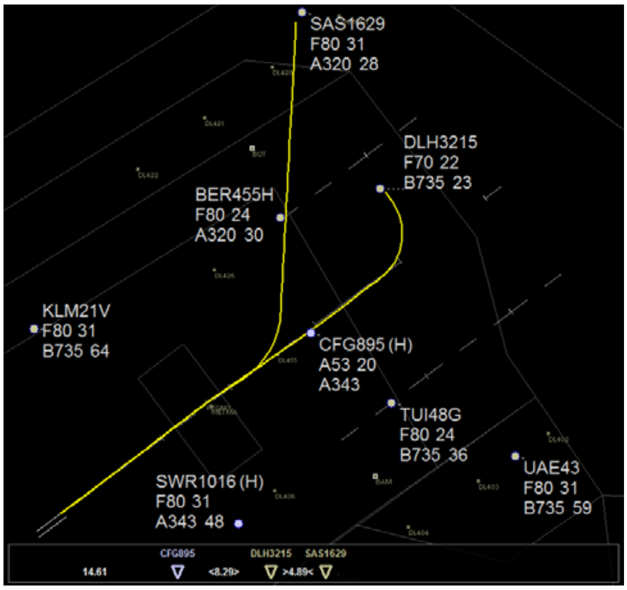

Figure 3. Two blue "Heavy" and six brown "Medium" aircraft approach an airport with two parallel runways and a specific airspace structure. The 'Centerline Separation Range' on the bottom shows separation of all aircraft and trends of separation on final (CFG895), base (DLH3215) and for "Directs" (SAS1629) with assigned waypoints on the centerline (DL455). (">4.89<" means distance in NM decreases since last radar update).

Furthermore, there is a "Centerline Separation Range" [30], [31] at the bottom of the radar screen (see Figure 3). This window shows the projected centerline separation in nautical miles between aircraft flying already on the centerline, base leg or which got a direct in approach from the controller.

The benefit of the "centerline separation range" for the controller is a good comparability of remaining flight distances for different kinds and origins of aircraft on one virtual centerline. Minimum separation between e.g. two direct flights from two different cardinal directions onto the same runway can be established already very early and far away from the centerline.

Highlighting an aircraft using mouse-over function in an AMAN configuration shows planned trajectory and an extended label (see Figure 4). As described in the previous chapter the controller has to input additional information to reach the highest AMAN functionality, see square 3 and 4 in Figure 2. For manual input a mouse interface is used to quantify the benefits between ASR and mouse or keyboard. By clicking on a label or head symbol a drop-down menu offers various values of which one has to be chosen and confirmed. Values encompass heading, speed or altitude, resp. waypoints for "direct-to" commands, controller positions for "handover" commands, runways for "cleared ILS" commands or other holding and path stretching possibilities.

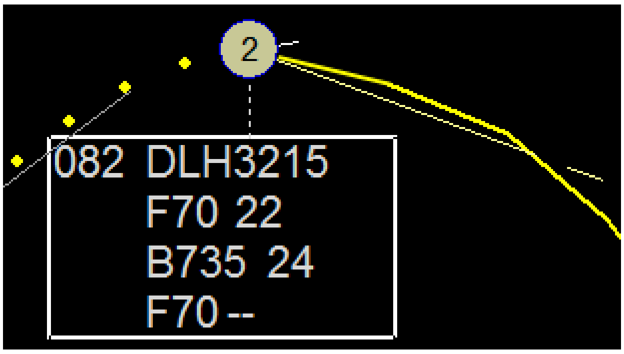

Figure 4. Integrated display information for a single highlighted aircraft. Extended label includes current heading "082", callsign "DLH3215", current flight level "70", current speed ("22"=220 knots), aircraft type "B735", distance-to-go "24", last altitude and speed controller command "F70 –". The head symbol is brown for weight category "Medium", contains current sequence number "2", and has a small white current heading line and a thin yellow short term flight prediction line for the position in 70 seconds marked with a break for the next 60 seconds position. Yellow dots show the history of radar positions; the thick yellow line is the AMAN-planned 4D-trajectory.

These additional inputs enable the high sophisticate version of AMAN to know the controllers intent as early as possible. This knowledge is used for an update of the planning of the AMAN if the actual intent of the controller does not fit to the actual AMAN planning. The impact of the additional input can be easily understood e.g. by the first simulation scenario. In this scenario, the information that the controller will guide the emergency aircraft directly to the runway enables the high sophisticated AMAN to immediately reschedule the whole arrival situation. Without this information the AMAN need more than 30 seconds to recognize the new situation. During this time span the controller will not be supported by the system. Displaying the wrong situation is more distracting for the controller than offering no support to the controller.

On a more abstract level of the description above we can talk about differences between the mental model of the controller and the model of the system concerning the actual and the predicted situation. At the moment the controller plans to prioritise the emergency aircraft the situational models of controller and of the system deviate. This deviation exists as long as the assistance system gets information about the situational change. After gaining the information the system is able to update its situational model towards the mental model of the controller. The needed information can be provided by mouse input or by speech recognition of the communication between controller and pilot. The choice of the additional input source has no effect on the support of the AMAN, but on the workload level, see square 3 and 4 in Figure 2.

VI. VALIDATION RESULTS

In the first subsection we define the used measurements. In the second part we present the results we obtained during the October trials described in section IV. In the next subsection we concentrate on our current results (Jan. 2015) obtained after tailoring the used parameters and correction of some errors.

A. Derived Measurements

The controller using an ABSR system is interested in a high recognition rate and a low error rate. The "Speech Recognizer" (see yellow part in Figure 1) together with the "Hypotheses Generator" is responsible for the high recognition rate, whereas the "Plausibility Checker" enable a low error rate, by rejecting some outputs of the "Command Extractor". TABLE II. shows that we have to distinguish between two error sources:

TABLE II. ERRORS OF FIRST AND SECOND ORDER

	Plausibility Checker accepts (rejection is false)	**Plausibility Checker rejects (rejection is true)**
ASR is correct	The desired behavior. ASR correct and not rejected **Probability**: $1 - \alpha =$ P(correct / accepted)	**2nd order error** / false alarm rate**:** Command rejected although ASR output of recognized command was correct. **Probability:** $\beta =$ P(correct / rejected)
ASR is wrong	**1st order error:** Command accepted although ASR is wrong. **Probability** $\alpha =$ P(err / accepted)	ASR is wrong and it is rejected ASR has no negative effect. **Probability:** $1\text{-}\beta =$ P(err / rejected)

Therefore, we have to distinguish three command sets:

- commands given by the controller (the real commands),
- commands recognized by ASR, and
- commands shown to the controller.

TABLE III. SEVEN CASES HOW TO REACT TO AN UTTERANCE

	Plausibility Checker accepts ***Given / Recognized /Shown***	**Plausibility Checker rejects** ***Given / Recognized /Shown***
ASR is correct	Hdg. 250 / 250 / 250 **C7** RcR_{ASR} =100%; RcR_{Ttl} =100%	Hdg. 250 / 250 / rejected **C3** RcR_{ASR} =100%; RjR_{Ttl} =100%
ASR is wrong	Hdg. 250 / 240 / 240 **C5** ErR_{ASR} =100%; ErR_{Ttl} =100% No Cmd / 240 / 240 **C6** ErR_{ASR} =100%; ErR_{Ttl} =∞	Hdg. 250 / 240 / rejected **C1** ErR_{ASR} =100%; RjR_{Ttl} =100% No Cmd / 240 / rejected **C2** ErR_{ASR} =100%; RcR_{Ttl} =100% Hdg. 250 / nothing / rej. **C4** DlR_{ASR} =100%; RjR_{Ttl} =100%

RcR, RjR, ErR, specify the recognition, rejection and error rates; rates values not specified are zero.

TABLE III. explains the seven different cases which may occur. We assume that a heading command with a value of 250 degrees is given. ASR may recognize a correct command (Hdg. 250), a wrong command (e.g. Hdg. 240) or may recognize nothing (del.). Cases C2 and C6 are special cases. No command is given, but ASR has recognized a command. This problem occurs e.g. if a long controller utterance containing only a single command is split into two or more commands by ASR.

We define the ASR recognition (RcR), ASR deletion (DlR) and ASR error rate (ErR) (# denotes "number of").

$$RcR_{ASR} = \frac{\#\ Cases\ C3, C7}{\#\ given\ commands} * 100\% \quad (5)$$

$$DlR_{ASR} = \frac{\#\ Case\ C4}{\#\ given\ commands} * 100\% \quad (6)$$

$$ErR_{ASR} = \frac{\# \, Cases \, C1, C2, C5, C6}{\# \, given \, commands} * 100\% \qquad (7)$$

Accordingly we define the Total rates resulting from the commands shown to the controller (case C2 is not counted):

$$RcR_{Ttl} = \frac{\# \, Cases \, C7}{\# \, given \, commands} * 100\% \qquad (8)$$

$$RjR_{Ttl} = \frac{\# \, Case \, C1, C3, C4}{\# \, given \, commands} * 100\% \qquad (9)$$

$$ErR_{Ttl} = \frac{\# \, Cases \, C5, C6}{\# \, given \, commands} * 100\% \qquad (10)$$

The recognition rate, when considering only the ASR component; is higher compared to the combined system of ASR and *Plausibility Checker*. The prize would be an increase in the error rate because some errors are not rejected. The *Plausibility Checker* decreases the error rate. The prize we have to pay is an increase of the rejection rate resulting in a decrease of the recognition rate.

We measure the performance of the *Plausibility Checker* by the parameters α and β. α measures the remaining errors (1st order errors). It is defined by the conditional probabilities:

$$\alpha = \mathrm{P(error \mid accepted)} = \frac{\mathrm{P(err + acc)}}{\mathrm{P(accepted)}} = \frac{\mathrm{P(Case\ C5, C6)}}{\mathrm{P(Case\ C5, C6, C7)}} \qquad (11)$$

β measures the case that the Plausibility Checker rejects a command although ASR has recognized it correctly (2nd order errors):

$$\beta = \mathrm{P(correct \mid rejected)} = \frac{\mathrm{P(cor + rej)}}{\mathrm{P(rejected)}} = \frac{\mathrm{P(Case\ C1, C4)}}{\mathrm{P(Case\ C1, C3, C4)}} \qquad (12)$$

For the validation of the hypothesis that the usage of ASR improves the conformance of the planning of the controller with the planning of the system we compared the conformance of the system planned trajectory to the flown trajectory, i.e. the observed radar data. The update rate of the planned trajectory was 0.2 Hz. We measured how often we observed a time, lateral or longitudinal deviation between both trajectories for each aircraft.

B. Results of the first Validations Campagn

In section IV in TABLE I. we defined six runs for each controller. One run had to be performed twice, due to a network failure, and one guest controller performed an extra run, resulting with the three training runs in 23 runs altogether. The training runs are not considered for the two validation questions. Approximately 43,750 words were spoken and 4,230 commands were given, i.e. each command had an average length of 10.3 words. The average recognition time per utterance (consisting sometimes of more than one command) was approximately 1.3 seconds on average, i.e. the time "Speech Recognizer" and "Command Extractor" needed. The "Lattice generator" needed on average below 6.8 seconds. The following TABLE IV. summarizes the observed rates of all 23 performed runs respectively for only the 20 runs considered for answering the validation questions.

TABLE IV. ASR AND USER RATES OBSERVED DURING TRIALS

	Recognition Rate	*Deletion/Rejection Rate*	*Error Rate*
ASR 23	91,5%	1,5%	12,6%
Ttl 3	87,4%	12,6%	4,2
ASR 20	90.7%	1.4%	12.2%
Ttl 20	86.7%	12.4%	3.9%

α and β are for the total rates: 4.5% and 31.9% for the 23 runs; 4.2% and. 31.6%;for the 20 runs.

Although the recognition rate of 87.4% with an error rate of 4.2% is not good we already could validate the hypothesis that ASR improves the conformance of controller's model and the internal model of the assistant system.

The following TABLE V. shows how the usage of ASR increases the conformance between system model and mental model of the controller, i.e. the functionality benefits of an AMAN supported by ASR. We compared for this purpose the 9 runs of the emergency scenario (three controllers each performing baseline, AMAN and AMAN+ASR).

TABLE V. NON-CONFORMANCE OF PLANNED AND FLOWN TRAJECTORIES WHEN COMPARING DIFFERENT AMAN SUPPORT LEVELS

Support Condition	*Baseline*	*AMAN*	*AMAN+ASR*
Controller C1	12.95%	19.65%	7.38%
Controller C2	20.88%	19.24%	12.41%
Controller C3	22.35%	20.88%	5.77%
Average	**18.70%**	**19.92%**	**8.52%**

In all three support conditions the number of samples (aircraft) was 23.

Each controller guides 23 aircraft in each run. In the baseline the results of the AMAN were not shown to the controller. Nevertheless the AMAN runs in background generating trajectories. Aircraft in an arrival flow are not independent from each other. Therefore, we combined the ith aircraft of controller C1, C2 and C3 into one single average value. In this way we get 69 measurements for the non-conformance by comparing the planned with the flown trajectories.

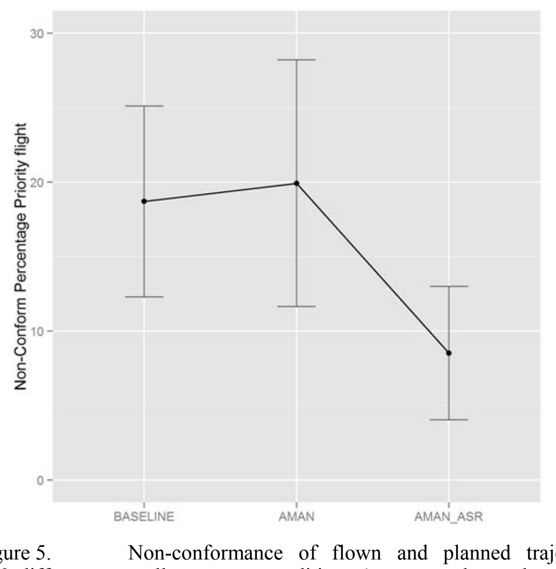

Figure 5. Non-conformance of flown and planned trajectory during 3 different controller support conditions (average value and standard deviation interval)

Figure 5 shows the average value and the interval of the standard deviation for the non-conformance frequency. An F-Test with two respectively 66 degrees of freedom was performed: F(2,66)=20.37 with a p value of less than 1%. It was confirmed that AMAN with ASR is an improvement compared to both AMAN without ASR and to the baseline without AMAN. Bonferroni adaptation shows no significant difference between pure AMAN and baseline.

With the runway closure scenario it was evaluated if we get the same improvements with respect to non-conformance of AMAN and controller models when using mouse and keyboard input instead of ASR input (see TABLE VI.). The performed F-Test with one and 38 degrees of freedom could not falsify, that AMAN+ASR is worse than using mouse and keyboard input (F(1, 38) =0.18; p = 0.89). Controller feedback indicates, however, a heavy workload increase, when using mouse input.

TABLE VI. NON-CONFORMANCE OF PLANNED AND FLOWN TRAJECTORIES FOR COMPARING MOUSE WITH ASR SENSOR

Support Condition	*AMAN + mouse + keyboard*	*AMAN+ASR*
Controller C1	11.08%	9.94%
Controller C2	8.99%	11.60%
Controller C3	13.29%	12.37%
Average	**11.12%**	**11.30%**

In both support conditions the number of sample (aircraft) was 20.

C. Problem Analyzis and improvements for the next campaign

In the last subsection we showed that ASR improves the conformance of controller's model and the internal model of the assistant system, although our recognition rate was only 87.4%. Remaining errors and false rejections were analyzed. The resulting improvements which will also be used in the final validation trials in February and March 2015 are presented now.

Some commands were not modeled, e.g. "hansa six yankee whisky speed own discretion" resulting in "OWN_SPEED" or "speedbird four one reduce minimum clean speed" resulting in "REDUCE_ MINIMUM_ CLEAN". Sometimes ASR wrongly recognizes commands of the same command class type in one utterance, e.g. REDUCE and INCREASE or HEADING and DIRECT_ TO. In this case we reject both commands, resulting in a decreased error rate, but an increased rejection rate. Our third improvement was in the "Hypotheses Generator" component. During the October validations trials we had a hypotheses error rate of 4.3%. Our current hypotheses error rate is below 1.6%, i.e. in less than 1.6% of the cases the controller gives a command not expected by the hypotheses generator. These remaining context errors may result in an ASR error and then perhaps in a total error. In most cases this, however, results in a wrong command rejection. TABLE VII. shows the results we get after these improvements.

Most of the remaining errors or rejections occur when the controller deviates from the modeled grammar, so called out-of-grammar utterances. Here are some examples:

- "air_berlin two four charly expect aeh some delay due to the runway closure of runway two three right about five minutes delay i call you back for further for the time continue on the domux two three transition"

TABLE VII. RATES AFTER GRAMMAR AND HYPOTHESES GENERATOR IMPROVEMENT, AND REJECTING COMMANDS OF SAME COMMAND CLASS

	Recognition Rate	*Deletion/Rejection Rate*	*Error Rate*
ASR	95,8%	0.3%	7.4%
Total	94.3%	5.7%	3.5%

α=3,6% and β=26.0% for the total rates for the 20 scenarios.

- "jersey seven kilo whisky did you copy the information as well runway two three right is closed for about five minutes and i call you back when it is open again"
- "air_berlin six six zero delta turn left heading two one zero cleared ils approach runway two three right intercepting from the north"

These deviations will happen in real life, but they are the exceptions. We cannot cover them all by repeated grammar update because this would result in reduced recognition rate for the normal utterances. Therefore, our approach during the final validation trials is to use plausibility values for the single command. This means that the ASR system assigns to each command a plausibility value between 0.0 and 1.0. A one means that the ASR system is very sure that the output command is correctly recognized. The "Plausibility Checker" component will reject all commands which are below a given threshold. TABLE VIII. shows the obtained results if ASR also calculates plausibility values and threshold value 0.85 is used.

TABLE VIII. RATES OF TABLE VII. WHEN USING PLAUSIBILITY VALUES

	Recognition Rate	*Rejection Rate*	*Error Rate*
ASR	95,8%	0,3%	7,4%
Total	91,2%	8,8%	2.4%

α=2.6% and β=52.8% for the total rates for the 20 scenarios.

Figure 6 shows the relationship between different plausibility values on the one hand and (total) error and (total) rejection rate on the other hand. If we increase the plausibility value, we reduce the error rate, but as a negative side effect we increase the rejection rate. A value in the interval of [0.7..0.9] seems to be a good choice, i.e. recognition rates between 91 and 92%.

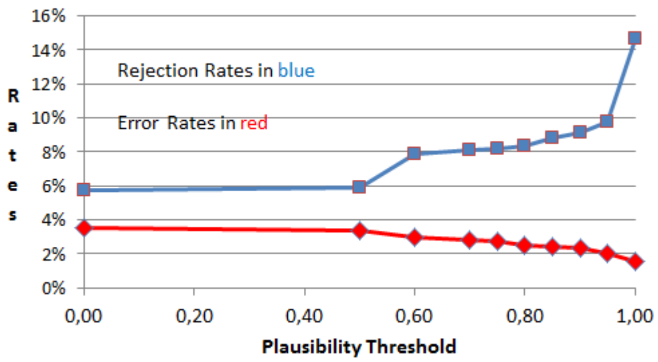

Figure 6. Total Error and Rejection Rates for different plausibility values

We performed the same evaluations not using the context of the assistant system, i.e. we just used ASR and not assistant based speech recognition. ASR recognition rates of 84% and error rates of 19.7% were observed. Using at least the "Plausibility Checker, but still not the "Lattice Generator", we got the total rates shown in TABLE IX. for the different threshold values.

TABLE IX. PERFORMANCE OF ASR WITOUT LATTICE GENERATOR

Plaus. Threshold	0.0	0.6	0.7	0.8	0.9	1.0
Ttl Recogn Rate	82.6%	80.1%	78.7%	77.5%	74.5%	57,8%
Ttl Error Rate	1.4%	1.3%	1.2%	1.0%	0.9%	0.6%
Ttl Reject. Rate	17.4%	20.0%	21.3%	22.5%	25.5%	42.2%
Rec Rate wih LG	94.3%	92.1%	91.9%	91.7%	90.4%	85.3

The last row shows again the recognition rate using the full ABSR approach with "Lattice Generator".

Table IX shows that the explicit integration of the command hypotheses into the speech recognition itself and not only into the rejection process increases recognition quality by more than 10 percent (absolute, not only relative). The very small error rates of approx. 1% together with the high rejection rate of 20% shows that in case of ASR failure without context the recognized commands substantially differ from the command hypotheses. Hence, it is much easier to reject the false output (e.g. "DLH123 REDUCE 16" knots).

VII. CONCLUSIONS AND NEXT STEPS

A. Next steps

Our results are based on a first validation performed in October 2014 with two male and one female controller which resulted in approx. 4,000 controller commands given in 23 simulation runs. In February and March 2015 a second validation campaign will be conducted with 12 controllers. We expect 72 simulation runs and 12,000 controller commands.

Most of the remaining recognition errors result from the fact that each controller uses her/his own phraseology subset. Our current approach was just to update the used grammar manually if we add a new controller to our data set. This is, however, not practical for real application in an ATC environment. Therefore, we are focusing on automatic learning of a language model, which will replace the usage of the grammar.

In an ATC environment (independent of ASR) low error rates, e.g. 0.1% or even 10^{-6}, are expected. Although great progress has been made in speech recognition we expect that these error rates will not be possible. We are developing assistant systems for the controller. Their task is to support the controller. If ASR helps the controller to increase efficiency or reduce workload it already has a great benefit. An example for integration of ABSR into electronic strips may illustrate this. If ABSR is correct, the controller just has to confirm the input into the strips. Let's assume this takes two seconds. If ABSR is wrong or rejected, let us assume that a manually correction lasts ten seconds and the manual input of a complete clearance without ABSR by mouse and keyboard needs five seconds.

TABLE X. WORKLOAD FOR DIFFERENT RECOGNITION RATES

Recognition Rate	0%	20%	70%	90%	95%	100%
Needed input time C1	4.4h	7,5h	3,9h	2,5h	2,1h	1,8h
Needed input time C2	8.9h	8.9h	4.4h	2.7h	2.2h	1.8h

C1 is the experienced controller, described above (needing 5s for mouse input, 2s for ASR confirmation, and 10s for correction). C2 described the results for the controller preferring paper flight strips (needing 10s for mouse input, 2s for ASR confirmation, and 12s for correction)

If we further assume that a controller position is operated for 16 hours per day and that the controllers at this position give 200 commands per hour, we can calculate the benefits of ABSR integration into electronic flight strips for different command recognition rates. TABLE X. shows that we can save between 1.9h and 6.2 h of controller workload with 90% recognition rate. 95% recognition only slightly increases these workload savings. The presented assistant based ASR system is ready for use.

B. Summary

The goal of this work was to show that assistant based speech recognition (ABSR) is ready for use for ATM applications. We selected DLR's Arrival Manager 4D-CARMA as a demonstration example. We created a validation setup consisting of six experiments for different controllers. On the one hand the experiments will be used to quantify the workload reduction of ASR for the controller, on the other hand to measure the increased support for the controller. The AMAN supports the controller in our validation setup especially in unusual traffic situation (e.g. runway closure, emergency) with sequence and distance-to-go information.

Important for the controller is on the one hand the command error rate of the speech recognizer especially when the output is directly used without acceptance or rejection through the controller. This is the case if ASR is used as additional sensor to update an assistant system. We focus on a low total error rate ErR_{Ttl}. TABLE V. shows that an error rate of 4% (TABLE IV. significantly improves the conformance of flown and planned trajectory. In other applications, e.g. maintenance of electronic flight strips, the controller on the other hand is just interested in a high recognition rate RcR_{ASR}, i.e. it makes no difference for the user if the ASR output is wrong or if it is "only" rejected. TABLE X. shows that recognition rates of 90% are sufficient. If we, however, think of electronic flight strip maintenance without explicit acceptance of the controller, but with rejection possibilities error rates below 3% are requested. Without controller monitoring, however, we need error rates below 0.1% which are currently not achievable.

We presented an architecture which uses the context information of an assistant system (e.g. an AMAN) for both the reduction of the search space of the speech recognizer and for rejecting misrecognitions of the ASR system. Only using the assistant system for search space reduction already enables recognition rates of more than 95%, but still with error rates of more than 7%. Using also the knowledge of the assistant system for rejections reduces error rate below 2.5%. The prize is a decrease of recognition rate from 95% to 91%, i.e. an increase of the rejection rate.

ACKNOWLEDGMENT

The work was conducted in the AcListant® project, which is supported by DLR Technology Marketing and Helmholtz Validation Fund. We also like to thank Jörg Buxbaum, Rocco Bandello from DFS (Deutsche Flugsicherung GmbH) and Jiri Janda (Air Navigation Services of the Czech Republic) for their valuable inputs when preparing the trials.

REFERENCES

[1] SRI International "Siri-based virtual personal assisstant technology" http://www.sri.com/engage/ventures/siri

[2] Homepage of Apple https://www.apple.com/ios/siri

[3] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, "Google search by voice: A case study," in Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics, pp. 61–90. Springer, 2010.

[4] S.W. Hamerich, "Towards advanced speech driven navigation systems for cars," in "Intelligent Environments," 2007. IE 07. 3rd IET International Conference, Sept. 2007, pp. 247-250.

[5] W. Li, Y. Zhou, N. Poh, F. Zhou, and Q. Liao, 'Feature Denoising Using Joint Sparse Representation for In-Car Speech Recognition', IEEE Signal Processing Letters 20 (7) , 681-684, 2013.

[6] H. Helmke, R. Hann, M. Uebbing-Rumke, D. Müller, and D. Wittkowski, "Time-based arrival management for dual threshold operation and continuous descent approaches," 8th USA/Europe ATM R&D Seminar, 29. Jun. - 2. Jul. 2009, Napa, California (USA), 2009.

[7] R. V. Chand, M. V. Chary, "Wireless Home Automation System With Acoustic Controlling," Sept. 2013, IJETT Journal, Vol 4-Issue 9.

[8] Speech Technology, "Nuance healthcare expands dragon medical portfolio,"http://www.speechtechmag.com/Articles/News/News-Feature/Nuance-Healthcare-Expands-Dragon-Medical-Portfolio-77155.aspx, 2011.

[9] DialRC, "The Dialog Research Center," http://dialrc.org.

[10] N. Hasevoets, and P. Conroy, "AMAN Status Review 2010," Eurocontrol, Edition number 0.1, 17 December, 2010.

[11] Sheryl R. Young, Wayne H. Ward, and Alexander G. Hauptmann, "Layering predictions: Flexible use of dialog expectation in speech recognition," in "Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI89)," 1989, pp. 1543–1549, Morgan Kaufmann.

[12] Sheryl R. Young, Alexander G. Hauptmann, Wayne H. Ward, Edward T. Smith, and Philip Werner, "High level knowledge sources in usable speech recognition systems," Commun. ACM, vol. 32, no. 2, pp. 183–194, Feb. 1989.

[13] Christian Fügen, Hartwig Holzapfel, and Alex Waibel, "Tight coupling of speech recognition and dialog management – dialog-context dependent grammar weighting for speech recognition," in "International Conference on Speech and Language Processing, ICSLP," 2004, Jeju Island, Korea, Oct. 2004, ISCA.

[14] C. Hamel, D. Kotick, and M. Layton, "Microcomputer System Integration for Air Control Training," Special Report SR89-01, Naval Training Systems Center, Orlando, FL., USA, 1989.

[15] D. Schäfer, "Context-sensitive speech recognition in the air traffic control simulation," Eurocontrol EEC Note No. 02/2001 and PhD Thesis of the University of Armed Forces, Munich, 2001.

[16] K. Dunkelberger, and R. Eckert, "Magnavox Intent Monitoring System for ATC Applications," Magnavox, 1995.

[17] FAA, "2012 National Aviation Research Plan (NARP),", March 2012.

[18] S. Ciupka, „Siris big sister captures DFS," original German title: "Siris große Schwester erobert die DFS," transmission, Vol. 1, 2012.

[19] J. M. Cordero, M. Dorado, and J. M. de Pablo. "Automated speech recognition in ATC environment" In Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems (ATACCS '12). IRIT Press, Toulouse, France, France, 46-53.

[20] D. Jurafsky, and J. H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition," 2nd edition. Englewood Cliffs, NJ, USA: Prentice-Hall, 9th Feb. 2008.

[21] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in: Soviet Physics -- Doklady 10.8, Feb. 1966.

[22] T. Shore, Fr. Faubel, H. Helmke, and D. Klakow, "Knowledge-Based Word Lattice Rescoring in a Dynamic Context," Interspeech 2012, Sep. 2012, Portland, Oregon.

[23] T. Shore, "Knowledge-based word lattice re-scoring in a dynamic context," master thesis, Saarland University (UdS), 2011.

[24] AcListant homepage: www.AcListant.de

[25] H. Helmke, H. Ehr, M. Kleinert, F. Faubel, and D. Klakow, "Increased Acceptance of Controller Assistance by Automatic Speech Recognition," in: "Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013)", Chicago, IL, USA, 2013.

[26] A. Schmidt, "Integrating Situational Context Information into an Online ASR System for Air Traffic Control," Master Thesis, Saarland University, 2014.

[27] Euroncontrol, "E-OCVM Version 3.0 Volume I – European Operational Concept Validation Methodology", Februar 2010

[28] O. Ohneiser, "RadarVision - User Manual for Air Traffic Controllers," original German title: "RadarVision - Benutzerhandbuch für Lotsen," German Aerospace Center, Institute of Flight Guidance, Research report DLR IB 112-2010/54, Braunschweig, 2010.

[29] O. Ohneiser, H. Helmke, H. Ehr, H. Gürlük, M. Hössl, T. Mühlhausen, Y. Oualil, M. Schulder, A. Schmidt, A. Khan, and D. Klakow, "Air Traffic Controller Support by Speech Recognition," in "Proceedings of the 5th International Conference on Applied Human Factors and Ergonomics AHFE 2014, Advances in Human Aspects of Transportation: Part II," N. Stanton, S. Landry, G. Di Bucchianico, and A. Vallicelli, Eds. Krakow, Poland: CRC Press, 2014, ISBN: 978-1-4951-2098-5, pp. 492-503.

[30] O. Ohneiser, "Flight guidance support to integrate conventional equipped aircraft into a time based arrival flow," original German title: "Führungsunterstützung zur Integration konventionell ausgerüsteter Luftfahrzeuge in einen zeitbasierten Anflugstrom," in "Future visualization systems for vehicle- and process guidance," original German title: "Fortschrittliche Anzeigesysteme für die Fahrzeug- und Prozessführung," vol. 2012-01, M. Grandt, and S. Schmerwitz, Eds. Bonn, Germany: German Society of Aerospace, 2012, pp. 175–192.

[31] M. Schaper, M. Temme, O. Gluchshenko, L. Christoffels, and A. Pick, "Coupling of ATM Planning Systems with Different Time Horizons," Air Transport and Operations Symposium 2012, Delft, 2012.

AUTHOR BIOGRAPHY

Heiko Ehr is Mathematic-technical Assistant. He joined DLR's Institute of Flight Guidance in 1997. Mr. Ehr worked on several projects like 4D-Planer and its successor 4D-CARMA, Flow Monitor and Traffic Monitor. He is an expert in system integration and interface development. Additionally he is the IT-Manager of the department of Controller Assistance.

Hartmut Helmke received his Diploma degree in Computer Science from the University Karlsruhe (Germany) in 1989 and his doctor degree (PhD) from the chemical engineering faculty of the Technical University of Stuttgart in 1999. In 1989 he joined DLR's Institute of Flight Guidance in Braunschweig. Since 1999 he concentrates on Controller Assistant Systems, especially Arrival Management being responsible of the Arrival Manager 4D-CARMA and. leading now the AcListant® project. Prof. Helmke is an assistant professor for Computer Science since 2001.

Matthias Kleinert received his Bachelor degree in Computer Science from Ostfalia University with a thesis on controller intent recognition. After finishing his semester at the University of Wisconsin Parkside, in Kenosha, USA he joined DLR's Institute of Flight Guidance in 2012. Currently he develops software for DLR's AMAN 4D-CARMA and is a master student of Computer Science at the Technical University Braunschweig, Germany.

Thorsten Mühlhausen received his diploma in Electrical Engineering from the University of Braunschweig (Germany) in 1994 and his doctor degree (PhD) from the faculty of Transportation and traffic sciences "Friedrich List" of the Technical University of Dresden in 1999. He joined the DLR Institute of Flight Guidance in 1995. Since 2012 he is member of the "ATM Simulation" department and responsible for real time HIL simulations.

Oliver Ohneiser received his master degree in Computer Science from the Technical University of Braunschweig (Germany) in 2011. He joined DLR in 2006 and is a scientific assistant in the department "Controller Assistance" of the Institute of Flight Guidance in Braunschweig since 2010. Mr. Ohneiser does his doctorate in the field of migration tolerant human machine interfaces.

Jürgen Rataj is head of the "Controller Assistance" department at the Institute of Flight Guidance. He has more than 20 years of experience with assistance systems in different application areas, e.g. aircraft engines, fixed wing aircraft, helicopter, road traffic as well as controller assistance. In addition, Mr. Rataj is assistant professor for ATM at the Ostfalia University of Applied Sciences.

Youssef Oualil joined the Spoken Language Systems group (LSV) in UdS as a doctoral researcher in October 2010. His research of interest mainly includes hand-free and distant speech recognition systems, signal processing and language modeling. Currently, Youssef is leading the speech recognition team in the AcListant project, Youssef received his computer science and applied mathematics engineering degree from ENSEEIHT, France in 2010.

Marc Schulder is a doctoral researcher at the Spoken Language Systems group in Saarland University. He previously received his master degree in computational linguistics from Saarland University in November 2013. Marc takes care of grammar design and text-to-concept processing in the AcListant® team. His scientific interests are machine learning and semantic processing, such as the extraction of information from spoken languages.