Reducing Controller Workload with Automatic Speech Recognition

Hartmut Helmke, Oliver Ohneiser, Thorsten Mühlhausen, Matthias Wies
German Aerospace Center (DLR)
Institute of Flight Guidance
Lilienthalplatz 7, 38108 Braunschweig, Germany
{Hartmut.Helmke|Oliver.Ohneiser|Thorsten.Muelhausen|Matthias.Wies}@DLR.de

Abstract—**Air traffic controllers normally manage all aircraft information with flight strips. These strips contain static information about each flight such as call sign or weight category. Additionally, all clearances regarding altitude, speed, and direction are noted by the controller. Historically paper flight strips were in operation, but modern controller working positions use electronic flight strips or electronic aircraft labels. However, independent from the type, considerable controller effort is needed to manually maintain strip information consistent with commands given to the aircraft. Automatic Speech Recognition (ASR) is a solution which requires no additional work from the controller to maintain radar label information. The Assistant Based Speech Recognizer developed by DLR and Saarland University enables command error rates below 2%. Validation trials with controllers from Germany and Austria showed that workload reduction by a factor of three for label maintenance is possible.**

Keywords—Controller Assistance; Speech Recognition; Workload; Aircraft Radar Label Maintenance

I. Introduction

A. Problem

Air traffic controllers interact with the Air Traffic Control (ATC) infrastructure most of their time on duty. This human-machine interaction causes much workload. The majority of the executive controller's workload consists of speaking to pilots and other controllers as well as documenting given clearances and other ATC relevant information. Acquisition of information and keeping records often is redundant work. Some of the controllers' tasks could already be supported by automatically acting assistance systems.

Air traffic controllers normally manage all aircraft information with flight strips. These strips contain static information about each flight such as call sign, weight category, destination, and route. Additionally, all clearances are noted by the controller. These clearances can be related to altitude, speed, and direction (heading or waypoints), but also to procedures like ILS clearances (ILS=Instrument Landing System). Furthermore, special situations like the declaration of an emergency are noted here.

Historically one paper flight strip for each aircraft was in operation. The information written via pencil is only accessible as a reminder for staff in the direct vicinity of the paper flight strip. Hence, the controller who wrote the information and maybe controllers like the planning controller sitting next to him get access. After a frequency shift of an aircraft to the next airspace sector, another controller at a different working position is responsible. He has no access to the pencil written information and, therefore, has to ask the pilot again for all necessary and relevant information. This causes additional radio frequency load and increases controllers' workload. Hence, paper flight strips have the disadvantage that information is not available and transferable in digital form.

Modern controller working positions use electronic flight strips or electronic aircraft labels. However, in many control centers paper flight strips are still in use, especially in high density terminal maneuvering areas (TMA).

Although there are good reasons for replacing paper flight strips by modern controller working positions with integrated electronic flight strips or electronic aircraft labels, justified concerns exist: Considerable controller effort is needed to manually maintain strip information consistent with commands given to the aircraft, because the information is either written down with a ball pen respectively an electronic pen or via mouse input. Both the next responsible controllers and other stakeholders will have benefits, but the controller, digitizing the clearances primary has additional effort. Air Navigation Service Providers (e.g. Austro Control), having replaced paper flight strips by electronic versions, reported an increase of controllers' workload resulting in a decrease of ATC efficiency (e.g. reduced flow and punctuality, increased delay) [1].

Although data link might replace voice communication in 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. But even if most of the controller pilot communication is based on data link, the responsible controller has to enter his commands into the system which then sends them to the pilot. Improved display software menus, in which controllers have to enter given commands, are another obvious solution for reducing controllers' workload. This may ease the command input process of controllers. However, there is still double work for them, i.e. speaking commands into a microphone for the pilot and entering the same commands in digital form via mouse or keyboard into the system. This input is necessary for a System Wide Information Management (SWIM).

The work was conducted in the AcListant® and AcListant®-Strips project [3], which were supported by DLR Technology Marketing and Helmholtz Validation Fund (AcListant = Active Listening Assistant).

B. *Solution*

Automatic Speech Recognition (ASR), however, can be a means to avoid the double input. The controller just speaks to the pilot via radio telephony communication. ASR transforms commands so that they appear in the aircraft radar labels in digital form. This of course requires a reliable speech recognition system. Our approach, Assistant Based Speech Recognition (ABSR), uses speech recognition embedded in a controller assistant system, which provides a dynamic minimized world model to the speech recognizer. The speech recognizer and the assistant system improve each other. The latter significantly reduces the search space of the first one, resulting in low command recognition error rates [2]. ASR can also ease the work for managing UAS (Unmanned Aircraft System) for both the ATC controller and the remote pilot also sitting on the ground.

C. *Results*

We compared in the AcListant®-Strips project [3] two possible methods to insert given controller commands into the radar labels. The first input method was the baseline. Controllers used the computer mouse for manual input. The second input method automatically worked with ABSR analyzing radio telephony channel between controller and pilot. The controller may confirm, correct, or reject the output of the speech recognizer. In November and December 2015 we performed validation trials for quantifying the benefits of using speech recognition.

D. *Paper structure*

In this paper we concentrate on quantifying benefits with respect to workload reduction. In [4] we will present efficiency improvements with respect to reduced flight time, kerosene savings and increased runway throughput. We present “Related Work” with respect to speech recognition and controller workload measurement in the next section. The “Validation” section presents the performed validation exercise. A summary and future work is explained in section “Conclusions and Outlook”. The references and the abbreviations are the last parts of this document.

II. Related Work

General ASR applications can be divided into three different categories:

1. Dictation software is used in the professional market. In consumer products they are not widely accepted due to their lack of adaptivity [5].
2. Hand-free command and control is characterized by short utterances to control technical devices [6].
3. Spoken dialog systems can be found at Siri® [7], Google’s search by voice [8], or train table dialog systems [9].

Despite that broad use of ASR systems, mainly large vocabulary and reliable recognition rates still are a challenge for speech-to-text systems. One promising approach to improve ASR performance is using context knowledge regarding expected utterances. These attempts go back to the 80s [10] [11]. This information may heavily reduce the search space and lead to less miss recognitions. Context was also used in dialogue systems with continually updating grammar in a Recursive Transition Network (RTN) [12].

First integrations of speech recognition in ATC systems especially for training started in the late 80s [13]. ASR applications in ATC domain benefit from ICAO (International Civil Aviation Organization) standard phraseologies that have to be used by both controllers and pilots. Nowadays enhanced ASR systems are used in ATC training simulators [14]. Pseudo pilots, who control the aircraft in ATC simulation environments, can be removed by using a dynamic cognitive controller model [15]. The system Voice Recognition and Response (VRR) of UFA (Burlington, MA) used by DFS (DFS Deutsche Flugsicherung GmbH, German Air Navigation Service Provider) is also used to reduce the number of pseudo pilots for controller training [16].

ASR applications go also beyond simulation and training. ATC events could automatically be detected in order to assess controller workload. ASR is used to get more objective feedback concerning controllers’ workload [17]. The often used ISA score (Instantaneous Self-Assessment) [18] and NASA-TLX score (National Aeronautics and Space Administration Task Load Index [19]) only provide subjective feedback from the controllers themselves. In order to have an objective workload measure, we used the secondary task performance measures method. This method identifies the amount of additional work the controller (or generally operator) can perform in addition to the normal primary work of air traffic control [20]. Thereby, the secondary task performance serves as an index for the workload of the controller [21]. The advantages of the secondary task method are that it is easy to use and that it is sensitive to variations in workload [22].

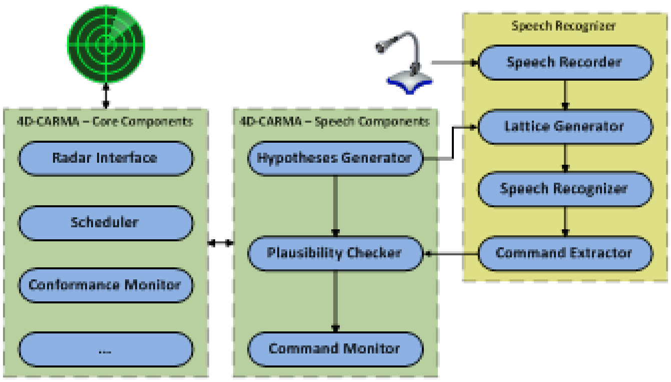

Fig 1. Components of an ABSR system (taken from [2]).

Chen and Kopald used speech recognition to build a safety net for airport surface traffic to avoid aircraft using a closed runway [23]. In [24] Chen et al. improved their approach by integrating context information to favor more situationally-probable hypotheses, and enabling the rejection of erroneous results via deductive reasoning during post-recognition processing. Their approach is derived from our approach presented in Lisbon 2015. Our Hypotheses Generator (Fig. 1 in [2] respectively Fig 1 in this paper) dynamically updates the recognition lattice of the speech recognizer and our Plausibility Checker performs the post-recognition processing.

Oualil et al. [25] analyzed the benefits of using context information for pre-processing versus using context for post-recognition. They favored post-processing, but their results based only on analysis of three controllers. Chen et al. [24] report, however, that there might be many false alarms to tower controllers, because safety critical commands are quite rare compared to the recognition error rate. The Word Error Rate (WER) is generally used as a metric to analyze ASR performance. The real spoken word sequence is called gold standard [26]. The WER is derived from Levenshtein distance [27] and defined as the distance between recognized and gold word sequence:

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

The numerator is given by the sum of the number of never spoken word insertions (ins(s)), the number of ASR missed and thus deleted words (del(s)), and the number of substituted words (sub(s)). The denominator contains the number of actually spoken words (W(s)). Alternatively, the number of sentences with at least one error may be counted as the sentence error rate (SER). Both, WER and SER, are not a good measure for speech analysis in ATC. The command error rate (CmdER) should be preferred. The correct recognition of each word in "Hello Speedbird six seven five descend flight level eight zero" is not crucial. However, extraction of the concept "BAW675 DESCEND FL 80" is important. We used the definitions of command recognition (CmdRR), command error (CmdER) and command deletion rate (CmdDR) according to [2].

$$CmdRR(s) = \frac{cor(u)}{C(u)} \tag{2}$$

C(u) is the number of commands spoken by a controller in an utterance. cor(u) is the number of commands correctly recognized by the ABSR system, which are not rejected by the Plausibility Checker.

$$CmdDR(s) = \frac{del(u)}{C(u)} \tag{3}$$

del(u) is the number of commands recognized by ABSR, but (correctly or accidently) rejected by the Plausibility Checker plus the number of commands given by the controller, but not recognized at all.

$$CmdER(s) = \frac{ins(u) + sub(u)}{C(u)} \tag{4}$$

ins(u) is the number of commands never spoken by the controller, but *recognized* and not rejected. subs(u) denotes the number of commands substituted by ASR and not rejected. Table 1 shows the development of recognition and error rates. The first two rows were already reported in [2].

TABLE 1: ABSR COMMAND RECOGNITION, DELETION AND ERROR RATES

Validation Trial	CmdRR	CmdER	CmdDR
Oct. 14 AcListant Pre-Trials	91.2%	2.4%	8.8%
Feb./Mar. 15 AcListant Trials	91.6%	3.0%	8.4%
Nov./Dec. 15 AcListant-Strips Trials	95.2%	1.7%	4.4%

III. VALIDATION

We evaluated two possible methods to insert given controller commands into aircraft radar labels. The "manual" way is to use the mouse. By left clicking on one of the five interactive grey label cells, a drop-down menu opens (see Fig 2). The controller has to select the intended value and add possible further values in other cells, which are displayed in yellow afterwards. This color marks them as unconfirmed. After completing the input of all necessary values for the respective aircraft, the controller has to confirm all these values by clicking on the green check mark.

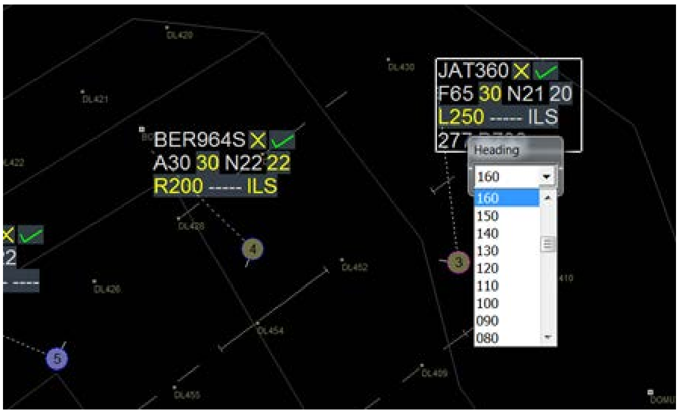

Fig 2. Drop-Down menu for heading input.

The second input method is supported by an assistance based speech recognizer (ABSR) developed by Saarland University (UdS) and DLR. The radio telephony voice channel between controller and pilot is analyzed by ABSR. The recognized commands are then visualized in yellow, i.e. they are still unconfirmed, in the corresponding five interactive cells. The controller may confirm or reject the output of the speech recognizer. In the latter case or if ABSR creates no output manual interaction of the controller is still necessary.

First, we describe now the validation hypotheses and second, the performed experiments. Third, we present the measurements and finally fourth, the results with respect to the hypotheses.

A. *Hypotheses*

In the AcListant®-Strips validation plan [28] that was the basis of our validation the following workload related hypotheses were formulated:

ABSR support for radar label maintenance (in contrast to mouse only input) …

1. … reduces the command input time,
2. … decreases controllers' workload, i.e.
   a. reduces the ISA score values and
   b. reduces the NASA-TLX score values,
3. … reduces the number of discrepancy in the radar label with respect to the given clearances,

4. … reduces the discrepancy time in the radar label with respect to the given clearances,
5. … increases free cognitive resources of the controller, i.e. the time a controller needs for performing a parallel secondary task is reduced,
6. … increases the number of given controller clearances (commands),
7. … enables more consecutive controller clearances (commands), i.e. time between two consecutive clearances is reduced.

Furthermore we had hypotheses with respect to ATC efficiency (e.g. flow, flight time, flight distance, fuel burn), see [4] for more details.

B. Experiments

The validation process of the AcListant®-Strips project is implemented according to the European Operational Concept Validation Methodology (E-OCVM) [29]. Therefore, two pre-validation trials have been conducted prior to the final validation trials in November and December 2015 in an iterative way. Pre- and final validation took place at the Air Traffic Validation Center at the DLR premises in Braunschweig.

The basic setup is shown in Fig 3. It consists of one controller working position and two pseudo pilot stations to handle air traffic. The simulated airspace comprises Düsseldorf (EDDL) approach TMA with only arrival traffic on runway 23R being modelled. The controller working position is equipped with RadarVision, an advanced radar screen [30] and a speech log screen (ASR Log in Fig 3). The radar overview shows inbounds approaching the airport within the next 10 minutes.

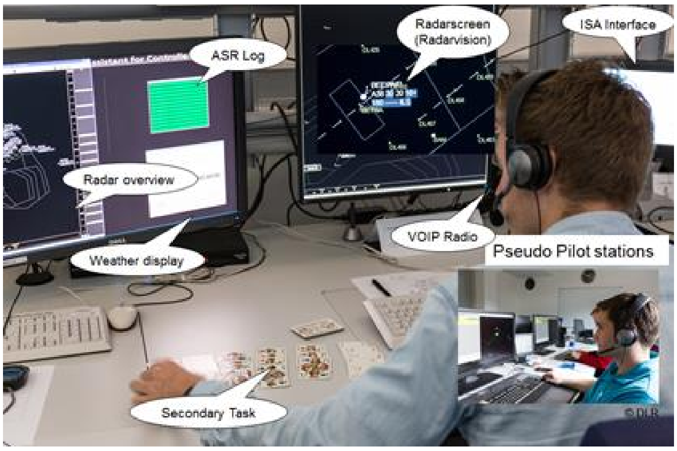

Fig 3. Basic validation setup.

The speech recognition engine directly uses the microphone output signal from the controller's headset. Both subjective ISA [18] and NASA-TLX score [19] as well as objective workload parameters were analyzed during our trials. The time needed for a secondary task, i.e. sorting a deck of cards and naming missing ones, results in objective workload measurements. We used two different approach scenarios, the combined pickup/feeder (PF) and the feeder (F) scenario. In the first one the controller acts as pickup and feeder controller with medium traffic (approx. 35 arrivals per hour, see Fig 4). It lasts 60 minutes with a 5 minutes runway closure at the beginning and an emergency flight in the middle.

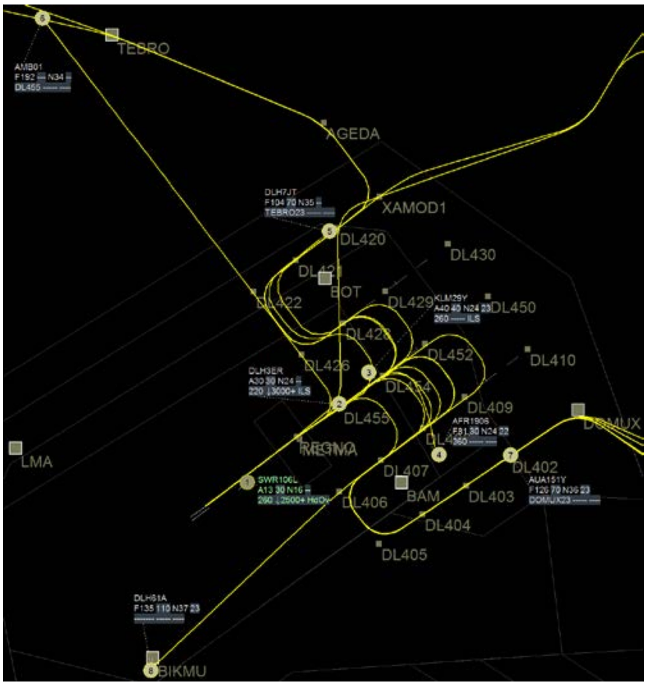

Fig 4. Responsibility area of pickup/feeder scenario.

In the feeder scenario the controller only acts as feeder controller with very high traffic (60 arrivals per hour, see Fig 5). It lasts 45 minutes. 60 arrivals per hour on one runway are of course impossible due to minimum separation regulations. Holdings or long path stretchings are necessary to safely handle this amount of traffic. Controllers were asked to give feedback during pre-trials when the flow of aircraft would be too high. A simulated pickup controller would then automatically reduce the flow by deleting the next two inbounds.

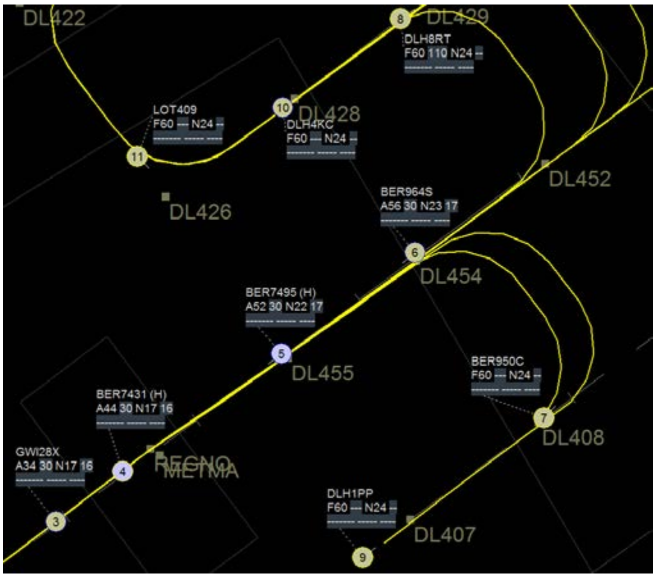

Fig 5. Area of responsibility in the feeder scenario.

This procedure, however, was not feasible. The controllers' request for help was too late, resulting in not reproducible study results. Often, the final with all aligned approaching aircraft was already 40 NM long, when deletion finally was requested. Therefore, in the main trials, we automatically deleted the next two inbounds by the simulated pickup controller, if the final gets longer than 22 NM.

We had the two different scenarios (PF, F) plus a training (T) scenario and three different input modalities: (1) *Mouse only*, (2) speech recognition plus mouse correction (*ABSR+Mouse*) (3) speech recognition plus correction via multi-touch display (*ABSR+MT*). The sequence of the following main scenarios (only reported in this paper) was shifted to avoid biased results due to training effects:

- Pickup/Feeder with *Mouse only* (PF-1)
- Pickup/Feeder with *ABSR + Mouse* (PF-2)
- Feeder with *Mouse only* (F-1)
- Feeder with *ABSR + mouse* (F-2)

To allow a one day effort per test person, the number of runs was limited to seven (PF-1, PF-2, F-1, F-2, F-3, T-1/2, T-3). The multi-touch modality during the Feeder scenario (F-3) was always one of the first two scenarios after the initial training runs (T-1/2, T-3). It was just used as another training scenario, but the controller was not told in advance, that the scenario is not evaluated. We ended four times with the modality 1 (*Mouse only*) and also four times with modalities 2 respectively 3 (*ABSR+Mouse*).

Eight controllers (four from DFS Deutsche Flugsicherung and four from Austro Control) participated in the trials. Two of them were female, six were male. They were between the age of 22 and 53 (*Mean* = 36, *Standard Deviation SD=Sigma* = 11) and their total work experience as a controller ranged between 1 and 32 years (*Mean* = 14, *SD* = 11). In pre-trials we marked all commands automatically as rejected, if the controller neither clicks on ACCEPT nor on REJECT within 20 seconds. The feedback of the controllers in the pre-trials, however, was – due to very high ABSR recognition rates – to automatically accept a command after 20 seconds, if no explicit action is performed by the controller!

C. *Measurements*

From the hypotheses formulated in the "Hypotheses" subsection we derived the following measurements. More details to the measurement values presented in this paper can be found in the final AcListant®-Strips validation report [31].

1) *ISA scores*

During all runs (scenarios) the controllers were asked for a (subjective) retrospective self-assessment of their workload during the last five minutes on a scale of five values from 1 (Under-utilized: The controller has little or nothing to do) to 5 (Excessive: The controller is overloaded. Some tasks are not completed. The controller feels he/she is not in control). Table 2 shows the results. Subjective workload decreases by around 10% when label maintenance is supported by ABSR.

TABLE 2: RESULTS FOR ISA SCORE

Scenario	Input	Mean	Sigma = SD	Median
Pickup/Feeder	Mouse only	2.9	0.5	2.9
Pickup/Feeder	ABSR+Mouse	2.6	0.4	2.6
Feeder	Mouse only	2.9	0.5	2.9
Feeder	ABSR+Mouse	2.6	0.5	2.7

2) *NASA-TLX scores*

After each run the controllers performed the NASA-TLX questionnaire. The column "Overall Workload" of Table 3 shows the average and standard deviation of overall rated and weighted workload of the controllers and the mental demand. Especially the weight "Mental Demand" was rated lower in the condition with ABSR support.

TABLE 3: RESULTS FOR NASA-TLX SCORE (MEAN AND SD)

Scenario	Input	Overall Workload	Mental Demand
Pickup/Feeder	Mouse only	9.5 / 2.2	31.0 / 19.1
Pickup/Feeder	ABSR+Mouse	7.6 / 2.9	20.8 / 12.5
Feeder	Mouse only	8.7 / 2.3	29.1 / 15.9
Feeder	ABSR+Mouse	7.0 / 2.9	15.8 / 9.9

3) *Command input times*

The radar display RadarVision logs various times. This includes clicking left with the mouse on one of the five interactive label cells (altitude, speed, direction, rate of descent/climb, miscellaneous in Fig 2) respectively on the green check mark (ACCEPT) or the yellow cross (REJECT). By calculating the time between the end of an input action (normally ACCEPT) and the beginning (click on one of the five command types), we get a time roughly representing the duration needed to perform the label maintenance with the mouse. The controllers were told to sequentially handle aircraft by aircraft.

Due to high ABSR recognition rate in the scenario with ABSR support in most cases the yellow information was accepted by only clicking the check mark without prior correction activities with a mouse. Therefore, no duration could be computed for the majority of the cases, because there is no indication when the input action actually started. In order to be able to compare those runs with "Mouse only" simulation runs we have to estimate the time a controller needs prior to the end of the input action.

For that we used the Keystroke-Level-Model (KLM) [32], which defines execution times for different types of human-computer interaction, e.g. press or release a button, move the mouse to a specific position on the screen, the mental process of thinking what to do next. Since the calculation of input commands via mouse starts with the first click in the respective label, we ignore the time the controller needs to move the mouse to the label. We estimate the additional time compared to the "Mouse only" scenario with 1,200 milliseconds for every command that was accepted without any correction. This time correlates with the duration needed for a single mental process thinking of what to do next. We assume e.g. the time a controller needs to move the mouse to the relevant label is the same in both input conditions. As the aircraft call sign was almost always correctly recognized, the relevant radar labels were highlighted in most of the cases and, therefore, were easier to find. On the other hand the controllers know to which aircraft they just talked to also in a “Mouse only” run.

TABLE 4: SIMULATION TIME NEEDED FOR A COMMAND INPUT [%]

Scenario	Input	Mean	Sigma	Median
Pickup/Feeder	Mouse only	30.6%	12.3%	28.3%
Pickup/Feeder	ABSR+Mouse	11.0%	3.1%	11.6%
Feeder	Mouse only	27.4%	11.2%	25.0%
Feeder	ABSR+Mouse	9.5%	2.0%	9.3%

Table 4 shows the percentage of (simulation) time needed for label maintenance. If controllers are supported by ABSR, the maintenance time could be reduced by a factor of approximately three.

4) *Discrepancies in radar label and their duration*

These subjective workload measurements were complemented by objective measurements with respect to workload. For this purpose we manually transcribed all the 11,280 given commands during the different scenarios. The *gold command* is the command really given by the controller (due to transcription). *Label command* is the command which appears in the radar label of the aircraft on the radar screen first. We distinguished four different main states for each radar label field influenced by ABSR respectively mouse input:

- *Consistent*: *gold command* and *label command* have the same value for that label field (e.g. speed value),
- *WaitingForSensor*: *gold command* and *label command* have the same value, but the controller has given a command by voice to the pilot, which is not entered yet into the radar label field (waiting for ABSR output, waiting for mouse input, ABSR output is wrong or missing and waiting for manual correction),
- *WaitingForRealWorld*: *gold command* and *label command* have the same value, but a command was entered by mouse or ABSR which was not yet given by voice (sometimes controller enter commands already by mouse while speaking to the pilot),
- *Inconsistent*: previous state was WaitingForSensor and an input from ABSR or mouse causes that *gold* and *label command* are different. Inconsistent states result also from previous state WaitingForRealWorld and a controller command results in a difference of *gold* and *label command*.

Table 5 shows for each run, how often the state *Inconsistent* was observed for a radar label field. We scaled the numbers by the total number of commands of the runs, so that longer runs are comparable with shorter ones.

TABLE 5: NUMBER OF INCONSISTENT STATES

Scenario	Input	Mean	Sigma	Median
Pickup/Feeder	Mouse only	4.8	3.3	3.5
Pickup/Feeder	ABSR+Mouse	4.1	1.9	4.7
Feeder	Mouse only	4.9	3.2	4.0
Feeder	ABSR+Mouse	2.3	1.4	2.1

The number of inconsistent states reduces when ABSR support is available but is still high. First we assumed that an explanation could be the auto-acceptance of commands after twenty seconds. The results, shown in Table 6, however, do not support this assumption. The controllers use auto-acceptance mostly in ABSR scenarios with more than 250 commands on average. The resulting error number is, however, very small. Less than 10% could be explained by this. Deeper analysis is still required.

TABLE 6: INCONSISTENCY ANALYSIS WITH RESPECT TO AUTO-ACCEPTANCE

Scenario	Input	Auto-Accepance Count	Inconsistencies due to Auto-Acceptance
All	Mouse	0.26	0.24
All	ABSR+Mouse	10.9	0.25

We also measured how long a field in a label remains in an inconsistent state, see Table 7. T-tests provide no significant difference between support with ABSR and no support scenarios. If a label field value is inconsistent it mostly remains inconsistent until a new command of the same type is given to that aircraft, which is not satisfactory. In Fig 1 and [2] a Command Monitor component is described which checks the recognized commands with respect to the observed radar data. Its result just needs to be shown to the controller. This functionality, however, was not used during AcListant®-Strips trials.

TABLE 7: DURATION OF INCONSISTENT STATE [S]

Scenario	Input	Mean	Sigma	Median
Pickup/Feeder	Mouse only	342	267	217
Pickup/Feeder	ABSR+Mouse	636	394	670
Feeder	Mouse only	279	351	93
Feeder	ABSR+Mouse	23	40	5

5) *Free Cognitive Resources (Secondary Task)*

We wanted to know, if ABSR also increases safety. Counting the number of incidents in which safety limits were violated was no option due to ethical issues and also due to statistical significance. An indirect approach was chosen. The time for performing a secondary task served as a measure: A deck of 48 playing cards (German Doppelkopf cards [33]) was used. The controllers were asked to sort the cards according to their kind (aces, kings, queens, jacks, tens, and nines) in a first step and thereafter to identify up to four missing cards in a second step, which were randomly taken out of the deck before. The time needed to sort the cards and to correctly identify the missing cards was used as a measure for the workload of the controllers. To minimize the intrusiveness to the primary task (the task to control the aircraft), the controllers were only allowed to sort the cards when no instructions to aircraft and no inputs to the radar labels had to be performed during the scenarios. The supervisor of the experiment monitored that controllers’ main attention was focused on controlling aircraft.

The advantages of sorting cards as the secondary task are that this task can easily be paused and continued again by the controller at any time and as often as required. Additionally, pausing of the secondary task does not require the controller to memorize the last status or setting of this task during the pause. Again, this minimizes the intrusiveness to the primary task. Most controllers were able to sort the card deck two, three or even four times during one run. In Table 8 we use the average sorting time of a controller. However, in the feeder scenario two controllers were so busy with command maintenance that they were not able to sort any card deck at all. In the feeder/pickup scenario only one controller was unable. We exclude these measurements from Table 8. The feeder scenario is, therefore, based on six controllers and the pickup/feeder scenario on seven controller measurements..

TABLE 8: TIME FOR SECONDARY TASK IN SECONDS

Scenario	Input	Mean	Sigma	Median
Pickup/Feeder	Mouse only	638	451	481
Pickup/Feeder	ABSR+Mouse	331	218	272
Feeder	Mouse only	377	142	333
Feeder	ABSR+Mouse	292	98	253

Controllers performed the secondary task much faster when being supported by ABSR than typing in all commands just with the mouse.

6) Number of given controller clearances

Table 9 shows the number of given commands. Please note: This is not the number of utterances. A controller utterance may contain between zero ("Good morning") and four commands in our trials. There is no significant difference between the ABSR and the mouse scenario. In the feeder scenario, lasting only 45 minutes instead of 60 minutes as the pickup/feeder scenario, of course fewer commands were given.

TABLE 9: NUMBER OF CONTROLLER CLEARANCES

Scenario	Input	Mean	Sigma	Median
Pickup/Feeder	Mouse only	322	33	323
Pickup/Feeder	ABSR+Mouse	324	32	329
Feeder	Mouse only	233	24	240
Feeder	ABSR+Mouse	245	39	245

Contrary to the initial hypothesis the controller did not reduce the number of given commands when his workload increases, i.e. when mouse input is necessary. He/she, however, changes priorities. In Table 10 it is shown, how often a command was given, but did not appear in the label. With higher workload the controller concentrates on his primary task, i.e. controlling aircraft. Maintenance of radar labels is also important for other stakeholders. However, there are situations in which priority for label maintenance is quite low.

TABLE 10: NUMBER OF FORGOTTEN AND WRONG COMMANDS

Scenario	Input	Mean	Sigma	Median
Pickup/Feeder	Mouse only	12.1%	7.4%	8.7%
Pickup/Feeder	ABSR+Mouse	4.9%	2.4%	4.7%
Feeder	Mouse only	6.7%	5.0%	4.5%
Feeder	ABSR+Mouse	4.4%	2.8%	4.0%

7) Consecutive controller clearances

The time interval between two consecutive utterances, i.e. the time difference between first utterance finished and next utterance started by the controller is measured. We only consider differences which are less than 15.1 seconds. Table 11 shows the results. Fig 6 contains the resulting histogram.

TABLE 11: PERCENTAGE OF CONSECUTIVE COMMANDS WITH TIME DIFFERENCE LESS THAN 15.1 SECONDS [%]

Scenario	Input	≤ 2	≤ 5	≤ 8	≤ 10	≤ 12	≤ 15
Pickup/Feed.	Mouse only	0	5	33	25	17	19
Pickup/Feed.	ABSR+Mo	0	5	31	26	19	19
Feeder	Mouse only	1	4	27	27	19	22
Feeder	ABSR+Mo	1	5	26	30	19	20

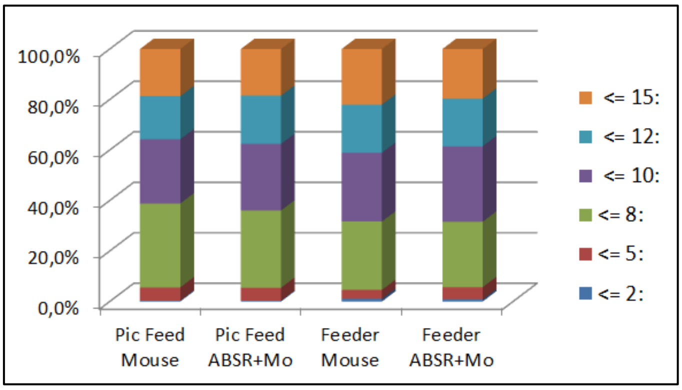

Fig 6. Historgram of time gap in seconds between two utterances.

D. Results

According to the E-OCVM methodology [29] we first derived the validation hypotheses, defined the measurements, designed the experiments to derive the needed measurements, and calculated the needed measurements. In this subsection we derive conclusions whether the measurements falsify or support the hypotheses.

For all our measurements we performed paired-t-tests. Each hypotheses was validated three times, first for the pickup/feeder scenario, then for the feeder scenario, and then for both scenarios together. We now present our approach in detail for the hypotheses "ABSR reduces the command input time in the feeder scenario".

A statistical test consists of a hypothesis H_0, a test value T, and a critical area to falsify the hypothesis. It should be clear, that a t-test can only falsify a hypothesis. Therefore, we formulate the counter hypothesis H_0, which we want to falsify, "ABSR increases the command input time in the feeder scenario compared to the mouse input modality". Our test value is defined by

$$T = (D - \mu_0)\,\frac{\sqrt{n}}{SD} \qquad (5)$$

We define a new measurement: the difference of the percentage in the ABSR supported run and the mouse supported run. Parameter n is the number of the defined new measurements (8 in our case). D is the difference between the two mean values of Table 4, i.e. the mean value of the new measurements (9.5% minus 27.4%=-17.9% in our case for the feeder scenario). SD is the standard deviation of the new measurements (10.8% was calculated from the 16 measurements [31]). We choose μ_0 as 0%, because we are just interested in checking whether ABSR input is less time consuming than mouse input. We calculate a value T of minus

4.71. If our hypothesis is that ABSR is 5% better than mouse input, we have to set μ_0 to 5%.

T obeys a t-distribution with 2*n-2 degrees of freedom. We can reject our hypothesis H_0 with probability of α (p-value), that the input time with ABSR is bigger than the input with mouse only, if the calculated value for T is less than the value of the inverse t-distribution with 2*n-2 degrees of freedom at position $t_{2*n-2,\ \alpha}$ (in our case minus 1.76). Therefore, the hypothesis H_0 is rejected (-4.71 < -1.76). We could even calculate the minimal α so that $T < t_{2*n-2,\ \alpha}$ still holds. This is in our case α=0.017%. The results very strongly support the hypothesis. We could also calculate the maximal value for μ_0 from Eq. (5) so that we still could reject H_0. The value for α=10% is 12.8%, i.e. with an error probability α of 10% the measurements even support that the percentage of needed time for label maintenance is at least 12.8% better (i.e. less) with ABSR support than with mouse only. On the other hand the improvement will not be better than 23.0% with a probability of 90%.

Table 12 shows that the measurements support the hypothesis that "ABSR decreases the command input time in the feeder scenario compared to the mouse input modality" with a p-value of 0.017%. For the pickup/feeder scenario the p-value is 0.08%. For the combination of both scenarios with already 32 measurements we have a value of 0.000075%, i.e. there is no doubt. As the counter hypotheses H_0 is rejected, we mark the cells in green. Yellow cells on the other hand will mark cells when we could not reject the counter hypothesis by our measurements.

TABLE 12: PAIRED-T-TEST FOR INPUT TIME REDUCTION

Scenario	Pickup/Feeder	Feeder	Both
min α	0.08%	0.02%	0.000075%

For the other hypotheses we just show the results in the following tables. Table 13 and Table 14 just confirm the objective measurements from Table 12 with the subjective questionnaires of the ISA and NASA TLX.

TABLE 13: PAIRED-T-TEST FOR WORKLOAD REDUCTION (ISA SCORE)

Scenario	Pickup/Feeder	Feeder	Both
min α	0.58%	3.4%	0.14%

TABLE 14: PAIRED-T-TEST FOR FOR WORKLOAD REDUCTION (NASA TLX)

Scenario	Pickup/Feeder	Feeder	Both
min α	6.1%	5.6%	1.3%

The number of discrepancies in the radar label (Table 15) and the discrepancy duration (Table 16) did only significantly decrease with ABSR support in the feeder scenario. Additional tool support is needed for the controller to point to possible inconsistencies in the radar label. This is easily possible by checking label contents against radar data, ADS-B data or mode-S data. This would not reduce the number of discrepancies, but their duration.

TABLE 15: PAIRED-T-TEST FOR NUMBER OF DISCREPANCIES IN RADAR LABEL

Scenario	Pickup/Feeder	Feeder	Both
min α	34.2%	2.9%	6.6%

TABLE 16: PAIRED-T-TEST FOR DISCREPANCY DURATION IN RADAR LABEL

Scenario	Pickup/Feeder	Feeder	Both
min α	no result	3.3%	no result

We checked also the hypothesis that ABSR support for radar label maintenance (in contrast to mouse only input) reduces the number of forgotten commands. Table 17 shows that this counter hypothesis was rejected.

TABLE 17: PAIRED-T-TEST FOR NUMBER OF FORGOTTEN COMMANDS

Scenario	Pickup/Feeder	Feeder	Both
min α	1.1%	8.2%	0.5%

Table 18 shows that ABSR also increases the free cognitive resources, i.e. the time which would be available for handling unexpected events. The controller seldom works and should not work at the performance limit [34].

TABLE 18: PAIRED-T-TEST FOR FREE COGNITIVE RESOURCES

Scenario	Pickup/Feeder	Feeder	Both
min α	0.25%	3.4%	0.1%

In Table 19 and Table 20 show we show that our experiments do not give any evidence that the selected input means have an effect on the number of given commands respectively the time between two consecutive commands.

TABLE 19: PAIRED-T-TEST FOR NUMBER OF GIVEN COMMANDS

Scenario	Pickup/Feeder	Feeder	Both
min α	38.6%	4.7%	10.3%

TABLE 20: PAIRED-T-TEST FOR TIME BETWEEN CONSECUTIVE COMMANDS

Scenario	Pickup/Feeder	Feeder	Both
min α	30.2%	no result	no result

IV. CONCLUSIONS AND OUTLOOK

This paper concludes our work in the context of Assistance Based Speech Recognition (ABSR), which started with the work of Shore et al. in 2011 [35] [36]. The potential of using context information from an Arrival Manager was shown. Word Error Rate Reductions by a factor of 5 could be possible. Our ATM 2013 paper presented already a possible ATC application of ABSR [37]: faster adaptation of an Arrival Manager, if the controller intentionally deviates from the proposal of the assistant system. 2015 we demonstrated that with the use of ABSR acceptable speech recognition (>90%) and error rates (<3%) are possible [2] and, furthermore, that ABSR significantly reduces the deviation between the controllers' plan and the plan of the Arrival Manager and, at the same time, significantly reduces the controllers' workload [38].

In this paper we were able to quantify the benefits of using ABSR with respect to controller workload reduction and in [4] with respect to ATM efficiency. The results are statistically significant. We used the Düsseldorf approach area as a demonstration area. In order to reduce adaptation costs, DLR, Saarland University, Idiap Research Institute together with the air navigation service providers from Austria and Czech

Republic have started the SESAR funded project MALORCA (Machine Learning of Recognition Models for Controller Assistance) in April 2016. This project aims at automatically learning models for recognition and for the Arrival Manager from recorded radar data and untranscribed controller pilot voice communication [39]. The recognized given controller commands can also be uplinked to UAS in electronic form, regardless of the remote pilot's availability. Therefore, ABSR can be an enabler for Unmanned Traffic Management (UTM) without a change of controller communication tasks.

In the conducted study of the project AcListant®-Strips, we used electronic aircraft labels, where the information is directly entered into the aircraft radar labels at the situation data display. Five interactive cells in the label represent given commands for altitude, speed, direction, rate of climb/descent, and miscellaneous information. We compared two possible methods to insert given controller commands into the interactive cells. The "manual" way is to use only the mouse. The second input method is based on Assistance Based Speech Recognition (ABSR). Manual input is only needed when speech recognition fails. We evaluated both input modes with respect to workload reduction and with respect to ATC efficiency improvements.

The NASA-TLX workload index improves by 20% when ABSR support is available. Sorting of cards, i.e. a secondary task, was roughly 50% faster in the ABSR supported condition. These values result in more mental capacity for other controller tasks when using ABSR-support for label maintenance. This conclusion is verified by the total time needed for mouse clicking in aircraft labels. Controllers need 30% of their time just for entering and confirming the clearances without and only 10% with ABSR support. The proportion is slightly higher in the feeder scenario due to the dense traffic.

Besides to higher workload and more time needed for label maintenance, the percentage of information lost when forced to manually input clearances into the system increases. Controllers' attention stays at the radar situation. However, this causes a neglect of entering information of – in the best case – lower importance into the system.

To sum it up, ABSR of AcListant®-Strips significantly reduces air traffic controllers' workload and improves radar aircraft label quality. Controllers' feedback was extremely positive. They want the AcListant®, i.e. ABSR, in their ops rooms.

Acknowledgment

We would like to thank – besides all the other controllers and experts – especially Jörg Buxbaum and Christoph Avenarius from DFS (Deutsche Flugsicherung GmbH) and Andreas Legler from Austro Control for their valuable inputs when preparing the trials for AcListant®-Strips.

References

[1] H. Helmke, "Grant Preparation: MALORCA – Part B, version 0.16, 18 March 2016 and AcListant project: "Feedback of Austro Control controllers during debriefing sessions", Braunschweig, December 2014.

[2] H. Helmke, J. Rataj, T. Mühlhausen, O. Ohneiser, H. Ehr, M. Kleinert, Y. Oualil, and M. Schulder, "Assistant-Based Speech Recognition for ATM Applications", in "11th USA/ Europe Air Traffic Management Research and Development Seminar (ATM2015)", Lisbon, Portugal, 2015.

[3] AcListant homepage: www.AcListant.de, AcListant = Active Listening Assistant, n.d.

[4] H. Helmke et al., "Increasing ATM Efficiency with Assistant Based Speech Recognition", unpublished.

[5] 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.

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

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

[8] 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," Springer, 2010, pp. 61–90.

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

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

[11] S.R. Young, A.G. Hauptmann, W.H. Ward, E.T. Smith, and P. Werner, "High level knowledge sources in usable speech recognition systems," Commun. ACM, vol. 32, no. 2, Feb. 1989, pp. 183–194.

[12] C. Fügen, H. Holzapfel, and A. 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.

[13] 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.

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

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

[17] 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, pp. 46-53.

[18] C.S. Jordan and S.D. Brennen, "Instantaneous self-assessment of workload technique (ISA)," Defence Research Agency, Portsmouth, 1992.

[19] S.G. Hart and L.E. Staveland, "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research," in "Human mental workload," P.A. Hancock and N. Meshkati, Eds., Amsterdam, North-Holland, pp. 139–183.

[20] W. B. Knowles, "Operator Loading Tasks," Human Factors, 1963.

[21] G. D. Ogden, J. M. Levine, and E. J. Eisner, "Measurement of Workload by Secondary Tasks," Human Factors, 1979.

[22] N. A. Stanton, P. M. Salmon, G. H. Walker, C. Baber, and D. P. Jenkins, "Human Factors Methods - A Practical Guide for Engineering and Design," Ashgate, 2005.

[23] S. Chen and H. Kopald, "The Closed Runway Operation Prevention Device: Applying Automatic Speech Recognition Technology for Aviation Safety," in "11th USA/ Europe Air Traffic Management Research and Development Seminar (ATM2015)," Lisbon, Portugal, 2015.

[24] S. Chen, H.D. Kopald, A. Elessawy, Z. Levonian, and R.M. Tarakan, "Speech Inputs to Surface Safety Logic Systems," in "IEEE/AIAA 34th

Digital Avionics Systems Conference (DASC)", Prague, Czech Republic, 2015.

[25] Y. Oualil, M. Schulder, H. Helmke, A. Schmidt, and D. Klakow, "Real-Time Integration of Dynamic Context Information for Improving Automatic Speech Recognition," Interspeech, Dresden, Germany, 2015.

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

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

[28] M. Wies, "AcListant-Strips: Validation Plan," in German, DLR; IB-112-2015/50, Braunschweig, 2015.

[29] Eurocontrol, "E-OCVM Version 3.0 Volume I – European Operational Concept Validation Methodology", Februar 2010.

[30] O. Ohneiser, H. Helmke, H. Ehr, H. Gürlük, M. Hössl, Th. 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, pp. 492-503.

[31] H. Helmke, O. Ohneiser, M. Wies and M. Kleinert, "AcListant-Strips: Validation Results of Main Trials," internal report, DLR-IB-FL-BS-2016-19, Braunschweig, 2016.

[32] D. Kieras, "Using the Keystroke-Level Model to Estimate Execution Times," http://www-personal.umich.edu/~itm/688/KierasKLMTutorial2001.pdf, 2001.

[33] www.doko-verband.de, website of German Doppelkopf Union in German, n.d.

[34] T. Edwards and B. Kirwan, "Working on the edge of performance," Hindsight, Vol. 20, Winter 2014, pp.72-76.

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

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

[37] H. Helmke, H. Ehr, M. Kleinert, F. Faubel, and D. Klakow, "Increased Acceptance of Controller Assistance by Automatic Speech Recognition," in "10th USA/ Europe Air Traffic Management Research and Development Seminar (ATM2013)", Chicago, IL, USA, 2013.

[38] H. Gürlük, H. Helmke, M. Wies, H. Ehr, M. Kleinert, T. Mühlhausen, K. Muth, and O. Ohneiser, "Assistant based speech recognition - another pair of eyes for the Arrival Manager," in "IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)", Prague, Czech Republic, 2015.

[39] MALORCA homepage: www.malorca-project.de, MALORCA = Machine Learning of Recognition Models for Controller Assistance, n.d.

LIST OF ABBREVIATIONS

ABSR	Assistance Based Speech Recognition
AcListant	Active Listening Assistant
ADS-B	Automatic Dependent Surveillance – Broadcast
ANSP	Air Navigation Service Provider
ASR	Automatic Speech Recognition
ATC	Air Traffic Control
CmdDR	Command Deletion/Rejection Rate
CmdER	Command Error Rate
CmdRR	Command Recognition Rate
DFS	DFS Deutsche Flugsicherung GmbH
DLR	German Aerospace Center
EDDL	Düsseldorf ICAO locator
E-OCVM	European Operational Concept Validation Methodology
F	Feeder scenario of AcListant trials
ILS	Instrument Landing System
ICAO	International Civil Aviation Organization
ISA	Instantaneous Self-Assessment
KLM	Keystroke-Level-Model
MT	Multi-touch
NASA-TLX	National Aeronautics and Space Administration Task Load Index
NM	Nautical Mile (1 NM = 1.852 kilometers)
PF	Pickup/Feeder scenario of AcListant trials
RTN	Recursive Transition Network
SD	Standard Deviation (Sigma)
SER	Sentence Error Rate
SWIM	System Wide Information Management
T	Training scenario of AcListant trials
TMA	Terminal Maneuvering Area (TRACON)
UAS	Unmanned Aircraft System
UdS	Saarland University
UTM	Unmanned Traffic Management
VRR	Voice Recognition and Response
WER	Word Error Rate