Apron Controller Support by Integration of Automatic Speech Recognition with an Advanced Surface Movement Guidance and Control System

Matthias Kleinert[1], Shruthi Shetty[1], Hartmut Helmke[1], Oliver Ohneiser[1], Hanno Wiese[2], Mathias Maier[3], Susanne Schacht[3], Iuliia Nigmatulina[4], Seyyed Saeed Sarfjoo[4], Petr Motlicek[4]

[1]Institute of Flight Guidance, German Aerospace Center (DLR), Braunschweig, Germany,

[2]Fraport AG Frankfurt Airport Services Worldwide, Frankfurt, Germany,

[3]ATRiCS Advanced Traffic Solutions GmbH, Freiburg, Germany,

[4]Idiap Research Institute, Martigny, Switzerland

[1,3,4]firstname.lastname@{dlr.de, atrics.com, idiap.ch}

[2]h.wiese@fraport.de

***Abstract*—Digital assistants in air traffic control today have access to a large number of sensors that allow monitoring of traffic in the air and on the ground. Voice communication between air traffic controller and pilot, however, is not used by these assistants. Whenever the information from voice communication has to be digitized, controllers are burdened to enter the information manually. Research shows that up to one third of controllers working time is spent on these manual inputs. Assistant Based Speech Recognition (ABSR) has already shown that it can reduce the amount of manual inputs from controllers. This paper presents how a modern digital assistant, a so-called A-SMGCS, can utilize the outputs of ABSR. The combined application is installed in the complex apron simulation training environment of the Frankfurt airport. This allows on the one hand the integration of recognized controller commands into the A-SMGCS planning process. On the other hand, ABSR performance is improved through the usage of A-SMGCS information. The implemented ABSR system alone reaches Word Error Rates of 3.1% for the text recognition process, which results in a callsign recognition rate of 97.4% and a command recognition rate of 91.8%. The integration of ABSR in the A-SMGCS brings a reduction of workload for controllers, which increases the overall performance and safety.**

Keywords—Apron Controller; Assistant Based Speech Recognition; Speech Understanding; A-SMGCS; STARFiSH

I. INTRODUCTION

A. *Problem*

In air traffic control (ATC), as in many other areas, there is a permanent need to increase the performance of the overall system. This need exists in particular at highly congested airports. However, an increase of efficiency must never come at the expense of safety. A decisive factor in this equation of efficiency and safety is the air traffic controller (ATCO), who has a major influence on the overall system performance. A key approach to increase efficiency is through digitization and automation. The means used to achieve this are digital assistants that support ATCOs in carrying out their work. This leads to a reduction of workload and allows ATCOs to guide the air traffic more efficiently while maintaining the same level of safety.

Today, the most advanced digital assistants in ATC already have access to a large number of sensors that allow monitoring of traffic in the air and on the ground. Together with manual input from the ATCO, the assistants are able to detect potentially hazardous situations and alert the ATCO. Voice communication between ATCO and pilot, however, as one of the most central sources of information in ATC, is not used by these assistants. Whenever the information from voice communication has to be digitized, ATCOs are burdened to enter the information manually. Research results show that up to one third of the working time of controllers is spent on these manual inputs [1]. This results in a decrease of the overall efficiency, because ATCOs spend less time on the optimization of traffic flow. The time needed for manual inputs will even increase in the next years as future regulations require more manual inputs e.g., Commission Implementing Regulation (EU) 2021/116 [2].

Assistant Based Speech Recognition (ABSR) has already shown that it can significantly reduce manual inputs from ATCOs by automatically recognizing and understanding ATCO-pilot communication as well as providing the required outputs for digital assistants automatically. The Active Listening Assistant (AcListant®) project [3] originally introduced ABSR [4] as a new form of automatic speech recognition (ASR). AcListant® coupled ABSR with a research prototype for arrival management to enable an early adaptation of the arrival route planning via voice information and support ATCOs with more precise arrival sequences [5], [6]. The successor project AcListant®-Strips proved the benefit of ABSR through the reduction of workload via automatic radar label maintenance [1], which leads to a more efficient ATC [7]. The Horizon 2020 funded project MALORCA [8] introduced a semi-automatic adaptation process for ABSR to reduce costs and provided research prototypes for radar label maintenance for Prague and Vienna approach. Also, other SESAR projects like, PJ.10-96-W2 [9], PJ.05-97-W2 [10] and HAAWAII [11] are currently working on different research applications for ABSR, e.g., incorporating pilot speech, providing automatic readback error detection, enabling automatic flight plan management. All projects clearly show that including voice information via ABSR into digital assistants is a valuable feature which not only provides benefits for existing applications, it also enables new ones. All projects so far incorporated ABSR into specific

research prototypes which were adjusted to prove the capabilities of a digital assistant equipped with ABSR. A first integration into a commercial product from ATC is needed to bring ABSR closer to industrialization.

B. Solution

The project STARFiSH (Safety and Artificial Intelligence Speech Recognition), funded by the German ministry of education and research, couples ABSR with the TowerPad® from the company ATRiCS, a commercial Advanced Surface Movement Guidance and Control System (A-SMGCS). This combined application of A-SMGCS and ABSR is integrated into the complex apron training simulator of the Frankfurt airport. The developed ABSR-system is one of the first implementations of the ABSR-architecture proposed by the SESAR 2020 project HAAWAII [12] [13].

The paper aims to validate the quality of the implemented ABSR system in a complex apron environment of a major airport and to present an implementation of ABSR into a commercial A-SMGCS. On the one hand, this involves integrating ABSR output into the A-SMGCS to incorporate recognized commands into the planning process to relieve apron controllers from the burden to enter this information manually. On the other hand, ABSR recognition performance should be improved through the usage of A-SMGCS information.

C. Paper Structure

The next section presents related work on speech recognition in ATC and applications surrounding it. Section III gives an overview of the implemented ABSR architecture and the integration with the A-SMGCS. Section IV explains the general application of an A-SMGCS and how ABSR outputs are incorporated. Section V describes the Frankfurt simulation environment used for validations and explains the evaluated metrics, whereas Section VI presents the results. Section VII concludes the paper and gives an outlook on next steps.

II. Related Work

The use of ASR in aviation training began in the 1980s [14]. Improved ASR systems are nowadays used in simulators to replace expensive simulation pilots in ATCO training, e.g., FAA [15], DLR [16], MITRE [17], DFS [18]. ASR applications beyond the scope of training [19], allow objective recording of controller workload [20], displaying warnings when a clearance is given for a closed or occupied runway [21], [22], verifying whether a pilot correctly repeats the ATCO instruction [23].

Although ASR systems such as Siri are widely used and aviation phraseology is standardized, recognizing and understanding controller-pilot communication is still a major challenge. The use of common and widely used commercial ASR tools has not produced acceptable results yet. Specific reasons for the poor performance include the variety of accents, the deviation of controllers from standard phraseology [24], and also the fact that aviation phraseology cannot be described by a common language model for every day speech. Cordero et al. reported word recognition rates of no more than 20% for various commercial-off-the-shelf recognizers [19].

A promising approach to improve ASR performance is to further exploit contextual knowledge about expected utterance. Attempts to do this date back to the 1980s [25], [26]. This knowledge can greatly reduce the search space and thus significantly improve recognition performance [16].

DLR and Saarland University coupled an Arrival Manager with ASR and developed a system for the Düsseldorf approach area that also recognizes complex commands and does not require strict adherence to the standard phraseology. As a result, both the Arrival Manager and ASR were improved [5], [6].

Although speech recognition has been the subject of aviation research, the voice communication between ATCO and pilot remains hidden from the current generation of A-SMGCS in use. However, the next generation of A-SMGCS will become a mandatory tool for many European airports and as a result of the European Commission Implementing Regulation (EU) 2021/116 [2], many ATCO clearances have to be available to the system in a timely manner, e.g., for monitoring purposes.

Systems currently available on the market from the five established suppliers (Saab Group, Indra, Thales Group, ATRiCS and ADB Safegate) require that these inputs are made manually by the ATCO via traditional user interfaces like mouse, touch or keyboard. Initial experience from the introduction of these next-generation A-SMGCS indicates additional workload for ATCOs, which can lead to reduced situation awareness and as a result in delays and a possible cap on airport capacity.

III. ABSR Architecture

The STARFiSH project implements as one of the first projects a modern ABSR system as defined by the architecture of the SESAR 2020 project HAAWAII. The core of the ABSR mainly relies on three modules (Fig. 1), which perform speech to text transformation (S2T), prediction of relevant context (Callsign Prediction) and extraction of semantic meaning (Concept Recognition). The only mandatory input in the ABSR is an ATC audio signal. Nevertheless, providing at least surveillance data as additional input to create context information is recommended and, therefore, used within STARFiSH. Fig. 1 gives an overview about the integration of the ABSR components (light blue) in the context of the project.

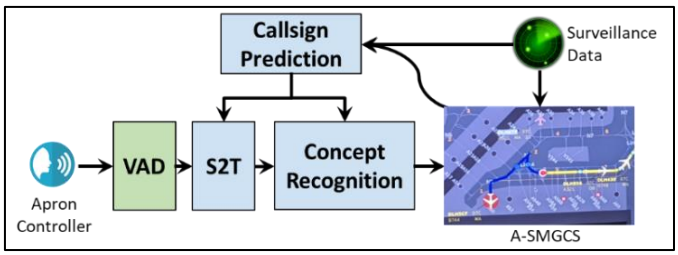

Figure 1. Application of HAAWAII architecture to the STARFiSH project

Voice Activity Detection (VAD). The audio signal comes as a continuous stream from the voice communication system. Therefore, some trigger is needed for ABSR to determine when a controller is speaking and the recognition and understanding

should be executed. The most precise trigger would be an access to the push-to-talk (PTT), which controllers are triggering via hardware to start and end their transmissions. Due to technical reasons PTT could not be used as a trigger in this project. To compensate that problem, STARFiSH utilizes an online VAD which determines, based on acoustics, when a transmission has started and ended. This decision is made by extracting the endpoint silence phones from the S2T module. The start and end of speech segments are detected based on the duration of pre-detected silence states and a probability of reaching to the final decoding state. We considered five pre-defined rules for detecting the end of a segment.[1]

Speech-to-text (S2T). Whenever the VAD detects a transmission, the signal is forwarded to S2T and the recognition process starts instantaneously to transform the audio signal into word sequences. This means the S2T delivers intermediate recognitions as soon as a controller starts speaking and updates the recognized words continuously until the end of the transmission. The S2T is implemented as a hybrid Deep Neural Network Hidden Markov Model (DNN-HMM). It is based on an HMM architecture combined with a Convolutional Neural Network Factorized Time Delayed Neural Network (CNN-TDNNF) architecture. The whole architecture is trained by the lattice-free Maximum Mutual Information objective function. This system follows the standard Kaldi recipe which uses Mel Frequency Cepstral Coefficients (MFCC) and i-vector as input features, and 3-fold speed perturbation and one third frame sub-sampling. Typical 3-gram language model (LM) was trained and adapted using in-domain data.

Concept Recognition. Every time a word sequence is forwarded it is analyzed by the concept recognition and transformed into relevant ATC commands as defined by SESAR project PJ.16-04 CWP HMI [27] and extended by HAAWAII [28]. An input word sequence could be *"lufthansa four nine nine taxi to alfa five eight via lima and november eight"*. The transformation then results in the following commands:

DLH499	TAXI TO	A58
DLH499	TAXI VIA	L N8

For intermediate recognitions from S2T, the concept recognition can provide an early recognition of the callsign or if requested by the application even an early recognition of the commands. The implementation relies on a rule-based algorithm, which determines the relevant parts in a step-by-step manner [28]. The concept recognition does not only transform the sequence of words into ATC commands, it also takes the decision whether a transformation could have been the result of a misrecognition and neglects a command if it seems unlikely. This decision is based on heuristic rules that define which commands are allowed or likely to appear together within the same transmission. A simple example would be a TURN LEFT and a TURN RIGHT within one transmission which usually does not make sense and is therefore neglected.

Callsign Prediction. This module takes surveillance data and in the case of STARFiSH also flight plan information from the A-SMGCS into account to determine if a callsign could be part of a controller voice transmission. The surveillance data is used to provide an overview of available callsigns in the airport area. The flight plan information then helps to determine, which of the callsigns are likely to be addressed in the near future. For that purpose, the callsign prediction takes a look at the responsible controller position, the target startup approval time (TSAT), the actual take off time (ATOT), the actual landing time (ALDT) and the actual in block time (AIBT). Relevant callsigns are forwarded to S2T and concept recognition to incorporate the callsigns in the process and enhance the quality.

A-SMGCS. The targeted application of the STARFiSH project takes the output of the concept recognition and incorporates it in the planning process so that apron controllers do not have to update the system manually. Furthermore, the recognitions are forwarded to the simulation pilot stations and carried out by the respective aircraft automatically. More information and details about the integration of the recognized commands into the A-SMGCS can be found in the next section.

IV. A-SMGCS Coupled with Speech Recognition

The original purpose of an A-SMGCS is to enable controllers to cope with the increasing number of operations at today's airports with more complex layouts and enhanced capacity, even in low visibility conditions [29]. For this, an A-SMGCS collects, fuses and enhances data from many different sources, most importantly surveillance data including onboard sensors (radar, MLAT, GPS via ADS-B, cameras, etc.) and flight plan data bases. A modern A-SMGCS goes one step further. As a digital assistant, it can detect and warn about potentially hazardous situations and inefficient plans or even mitigate these by controlling infrastructure such as stop bars. However, these new services require even more data. The instructions and clearances, including taxi routes, that a controller issues to pilots, need to be available to the A-SMGCS.

European legislation has acknowledged the value of these new A-SMGCS services for safe operations and requires providers of ATC at complex airports to deploy systems that can monitor clearances [2]. It seems clear that the mandated deployment of these systems can only succeed if the gains in safety and efficiency from automation are not cannibalized by a workload increase due to manual inputs for controllers [30]. Research results already show that up to one third of controllers working time is required on such manual inputs [1]. At the same time the radio telephony (R/T) transmissions from controller to pilot, which are a regulatory requirement, already contain the required information for modern A-SMGCS services.

The approach in STARFiSH is to replace the need for the majority of manual inputs by automatically recognizing the speech within R/T transmissions. Building on the architecture from Section III, the recognized commands are checked for

[1] Rules for endpoint detection defined in https://github.com/kaldi-asr/kaldi/blob/master/src/online2/online-endpoint.h

plausibility before being considered by the A-SMGCS. The commands from ABSR (output of the concept recognition) are fed into two components:

a) The controller A-SMGCS manages the world state (clearances, routes, instructions) for the apron controllers as a utility for situation awareness and planning. It also displays the traffic situation and other properties of the aircraft created by the traffic simulator.[2]

b) The simulation pilot A-SMGCS manages the world state for simulation pilots which control the trajectory and other properties of the simulated aircraft. The simulation pilot commands are translated automatically into realistic aircraft movements by the connected traffic simulator.

Fig. 2 shows how both A-SMGCS systems are connected to ABSR and the rest of the simulation. Independent of the user's role at the workstation (pilot or controller), the A-SMGCS processes commands received from the ABSR as follows:

- Check if a command is plausible and discard it when it is not consistent with the traffic situation, e.g., a "TAXI TO" command to a runway line-up for an inbound flight.
- Forward the command to the correct working position (e.g., EAST), but discard commands for flights that are not under control of a position, e.g., when ABSR recognizes a wrong callsign not controlled by a certain working position.
- Highlight the flight on the HMI of the working position that manages it, so the user knows that the ABSR has recognized this callsign and associated commands.
- Display an information whenever ABSR indicates that no command ("NO_CONCEPT") was recognized. The user has to enter the information manually instead.
- Delay execution of the command, if the next command may change the meaning, e.g.

  DLH499 TAXI TO A58
  DLH499 TAXI VIA L N8

  The first command here sets a new destination, which will trigger the A-SMGCS to calculate a new route. However, the next command specifies part of the route, which must be considered as well. So, the new route will only be displayed after the second command or when the ABSR marks the transmission as complete.
- Execute functions that need to be triggered due to the context. E.g., in some situations, a TAXI command implies a "continue", in others it does not.
- Augment commands that contain insufficient or ambiguous information. "Give way to A320 from the right" is, e.g., unambiguous for the pilot, but the A-SMGCS needs to evaluate all possible situations of A320 flights that will come close to the pilot's aircraft at possible crossings and intersections and then select the most plausible one.
- Display the results of the command on the work station, offer "UNDO" for commands that can be easily undone and have proven the necessity to take back quickly in case of an error, e.g., handover of aircraft to a wrong position.

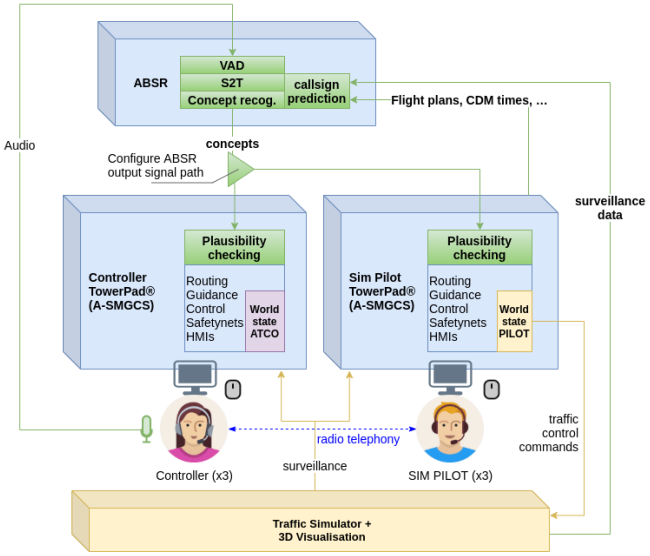

Figure 2. STARFiSH Simulation and Validation Setup

V. Experimental Setup and Metrics

The A-SMGCS coupled with ABSR was integrated into the apron training simulator of the Frankfurt airport to validate the quality of the ABSR architecture in a complex simulation environment and to evaluate the benefits of ABSR in a commercial application. This section describes the simulation environment, traffic scenarios and metrics used for evaluations.

A. Simulation environment and scenarios

The validation was executed in the Fraport apron simulator (manufactured by ATRiCS) which includes the traffic simulation, the A-SMGCS for controllers and pilots and 3D visualization for the outside view of the controllers (see Fig. 2). Movements outside the areas of apron controller responsibility have been automated, i.e., landing, runway exit, lineup and take off. The validations were done with the three apron controller working positions East, Center and West. Each controller used an A-SMGCS Integrated Controller Working Position (ICWP) and a radio for communication to the pilots. For each apron controller there was also one simulation pilot for monitoring and controlling the simulation and to give R/T readback.

In the near feature apron controllers at Frankfurt airport need to use an A-SMGCS were they have to input all commands given via radio to the pilot additionally into the system (i.e., pushback, taxi, change of taxi routes etc.). This way of working was simulated with and without (baseline) the support of ABSR.

[2] It does not alert of hazardous situations, as many A-SMGCS would, in order to mimic the system that Fraport's apron controllers are currently familiar with.

In the baseline scenario apron controllers worked with an A-SMGCS, where all inputs were manually entered with a mouse, since no ABSR was available. This was compared to the solution scenario, where an ABSR System delivered the inputs automatically and controllers had to correct the system in case of a misrecognition. Additionally, the influence of automatically executing the controller commands within the simulation was evaluated. For this purpose, the recognized commands from apron controllers were, in some scenarios, forwarded to the pilot A-SMGCS and executed automatically. In these scenarios the pilots were only responsible for the R/T readback and correcting actions in case of misrecognitions. In all other scenarios the pilots had to manually trigger all the actions in the simulation in addition to giving R/T readback.

The baseline and the solution scenario have been simulated in both operational directions for Frankfurt airport, i.e., either runways "25" or "07" have been used. The operational direction influences the entry and exit points of the traffic and the work is differently distributed among the controller positions. All simulations had a very high traffic volume.

The simulations took place for five days and the group of apron controllers changed each day, i.e., every day three new apron controllers participated for the positions East, Center and West. Each day started with an introduction and a short training, followed by six different validation runs with a duration of 30 minutes. Table I shows the setup of the different runs and showcases whether the output of ABSR was used at the respective station, or not. The order of the runs varied each day.

TABLE I. SETUP OF SIMULATION RUNS FOR ONE SIMULATION DAY (ORDER OF RUNS DIFFERENT ON EACH DAY)

Run	RWY direction	Pilot stations	ATCO stations
1	25	ABSR ON	ABSR OFF
2	07	ABSR ON	ABSR ON
3	25	ABSR ON	ABSR ON
4	07	ABSR ON	ABSR OFF
5	25	ABSR OFF	ABSR OFF
6	25	ABSR OFF	ABSR ON

B. *ABSR metrics*

To evaluate the performance of the ABSR system, different evaluation metrics are used. The performance of the S2T module is evaluated using Word Error Rate (WER). It is computed by the Levenshtein distance between the recognized word sequence and the spoken word sequence. The Levenshtein distance considers the substitutions, insertions and deletions [31].

The performance of the Concept Recognition module is evaluated by comparing the automatically extracted commands, so called automatic annotations, with the correct commands manually created and verified by human experts, so called gold annotations. The evaluation is carried out using the following three metrics: command recognition rate (RecR), command recognition error rate (ErrR) and command rejection rate (RejR). The RecR is defined as the number of correctly recognized commands divided by the total number of actually given commands. A command is said to be recognized if and only if all the elements of the command such as the command type, value, unit, qualifier, condition etc., as defined by the ontology [28], are all correctly recognized. ErrR is the percentage of wrongly extracted commands, divided by the total number of gold annotations. RejR is the percentage of gold annotations, which are not extracted at all. Table II below illustrates the above defined metrics using an example.

TABLE II. EXAMPLE TO ILLUSTRATE CONCEPT RECOGNITION METRICS

Example		
Gold Annotation	**Automatic Annotation**	
DLH4AR TURN LEFT	DLH4AR TURN RIGHT	⊖
DLH4AR TAXI VIA A B	DLH4AR TAXI VIA A B	⊕
	DLH4AR TAXI TO V143	⊖
AUA1AB PUSHBACK	AUA1AB NO_CONCEPT	○
BAW123_NO_CONCEPT	BAW123 NO_CONCEPT	⊕
Result		
RecR = 2/4 = 50% (green ⊕)	*ErrR = 2 / 4 = 50% (purple ⊖)*	*RejR =1/4 = 25% (yellow ○)*
The example above also illustrates that the sum of RecR, ErrR, and RejR can exceed 100%.		

Similarly, callsign extraction rates are calculated using metrics such as callsign recognition rate (CaRecR), callsign recognition error rate (CaErrR) and callsign rejection rate (CaRejR). For every utterance, each callsign is considered only once, except for when multiple callsigns are annotated or extracted. Detailed information on defined metrics are in [32].

C. *A-SMGCS metrics*

To evaluate how much interaction with the A-SMGCS is required with and without ABSR we recorded the number of HMI interactions at each position and for each simulation run, categorized into 48 different tasks such as "edit route", "clear pushback", "select label". The expectation was that the number of interactions are significantly lower when ABSR support is available. However, it was clear from previous experiments that controllers would not correct every error made by the ABSR since a missing input at the controller A-SMGCS in many cases has no direct consequence: If, e.g., a pushback command was not input on the controller A-SMGCS, then the aircraft would not be flagged as pushing back, but would still push after a few seconds, because the pilot would take care of that even when ABSR fails to recognize the command. That means for the HMI interactions pilots are also evaluated, since they have to input everything which makes their results more reliable.

D. *Workload and performance metrics*

NASA TLX [33] questionnaire was used to measure the individual workload of the participating controllers by self-estimation on a scale of 1-20 for the degree of high mental, physical, and temporal demand, lack of performance, and high effort and frustration. The controllers filled in a questionnaire after each run.

The SHAPE SASHA [34] questionnaire was used after each run to measure the situational awareness of controllers by estimation on a scale of 0-6 whether they have been ahead of traffic, did focus on a single problem, risked forgetting something, were able to plan before, were surprised, or had to search for an information item.

Additionally, we wanted to get an estimation for the automation trust of the apron controllers, and used the SHAPE SATI [34] questionnaire. It measures the automation trust by a self-estimation on a scale of 0-6 whether the system seems useful, reliable, accurate, understandable, and robust and makes them feel confident. The controllers filled in a questionnaire after each day, and were asked to focus on the automation feature of ABSR, i.e., immediate highlighting of recognized callsigns and integration of recognized commands at the TowerPad® ICWP.

The questionnaires reflect the subjective experience of the controllers. It can be argued that, for many real-world applications, this is the most important measure. However, in order to have more objective data on the impact on workload, we designed a secondary task that requires similar skills to the main task of controlling traffic, e.g., color perception and quick orientation in the user interface, yet can be executed in parallel.

The task is based on the Stroop test [35] and implemented in an open source app[3] that records the number of correctly executed tests over time. A higher number of correct tests indicates more mental capacity available for the secondary task, so less workload capacity is consumed by the primary task [36]. After 10 minutes of a simulation run, the controllers had to perform the secondary task for ten minutes in addition to controlling the traffic.

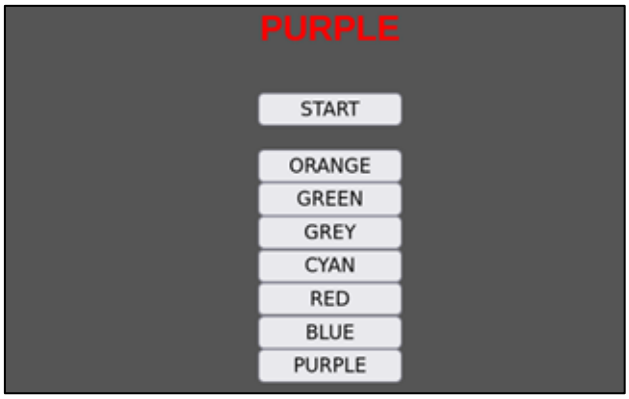

Figure 3. Stroop Test Screenshot

When the user presses the “START” button (Fig. 3), the app shows a word for a color printed in a different color. The user has to select the color of the print from a list of buttons labelled with the names of colors. So, in the example in Fig. 3 the user has to select RED. As the brain is faster in perceiving the text than recognizing the color and translating it into a word, this task creates mental load and requires focus.

[3] Our implemtation of the stroop test can be accessed here: https://github.com/MathiasMaier/workload-gauge

VI. Results and Discussion

This section shows the results of the quality measures for the ABSR system. Afterwards the metrics related to the A-SMGCS, workload and performance of the controllers are presented.

The WER and concept recognition metrics for the ABSR system are presented for the online and offline mode. Online performance measures the live performance of the ABSR achieved during the final simulation trials carried out end of June 2022. This means these results include a certain proportion of errors made by the VAD as the PPT signal was not available (see Section III) and, therefore, some transmissions were not detected correctly. Offline performance measures the quality of the ABSR system with a perfect splitting of the audio stream performed offline, i.e., a simulation of how the system could perform if PPT would have been available.

Table III shows the WER of the S2T component from Section III. In total the utterances of 14 apron controllers have been evaluated. We see that the offline recognitions of the S2T are better than the online recognitions with overall WERs of 3.1% and 5.0%, respectively. It is also interesting to observe that the average WERs of female apron controllers (2.6% resp. 3.7%) were better than those of male apron controllers (3.3% resp. 5.5%). On the other hand, out of the total 14 apron controllers, only 4 were females.

The question often arises as to which WER is good enough. This question cannot be answered easily, because ultimately it does not matter how many words are correctly recognized. What does matter is the ability of the system/application to grasp the meaning behind the recognized words. Some errors on word-level may change the meaning of an utterance, while others have no effect at all. Therefore, it is not possible to define a general threshold for WER, but a low WER obviously allows conclusions to be drawn about the quality of the implemented S2T component. Compared to humans, the WER presented in Table III can be considered very good. In the literature, the WER of humans is estimated somewhere between 4% and 11%, depending on the circumstances, e.g., only correction of pre-transcriptions, fast transcriptions with one chance to recognize everything, or listening multiple times to one utterance and making corrections along the way [37].

TABLE III. WER - Evaluation of Speech-To-Text

Type of Recognition	WER	
Offline Recognition	3.1%	Male - 3.3%
		Female - 2.6%
Online Recognition	5.0%	Male - 5.5%
		Female - 3.7%

The following Tables IV and V illustrate the performance of the concept recognition using the metrics defined in Section V. These tables show the total number of utterances (#Utter), commands (#Cmds), command recognition (RecR), error (ErrR)

and rejection rates (RejR), and also the callsign recognition (CaRecR), error (CaErr) and rejection rates (CaRejR). The performance is evaluated on only a subset (~56%) of the entire data for which manually verified annotations are available.

TABLE IV. COMMAND EXTRACTION RATES

Type of Recognition	#Utter	#Cmds	RecR	ErrR	RejR
Offline Recognition	5,495	13,251	91.8%	3.2%	5.4%
Online Recognition	5,432	13,168	88.7%	4.3%	7.5%

TABLE V. CALLSIGN EXTRACTION RATES

Type of Recognition	#Utter	CaRecR	CaErrR	CaRejR
Offline Recognition	5,495	97.4%	1.3%	1.3%
Online Recognition	5,432	95.2%	2.3%	2.4%

From Table IV, we see that we obtain recognition rates of 91.8% and 88.7% when concept recognition is run on offline and online word recognitions, respectively. Similarly, the error rates are also better in the offline recognitions. The improvement in the output of the concept recognition in the offline mode directly corresponds to the better word-level recognitions by S2T. Similarly, from Table V we see that the callsign recognition rates are also better with offline recognition as compared to the online recognition with recognition rates of 97.4% and 95.2%, respectively.

Table VI shows the offline rates for each command type for the most frequently occurring and important command types, which are relevant for the application. From the table, we observe that the concept recognition rates are above 86% with error rates less than 4% for all command types, except for GIVE_WAY, where we obtain recognition and error rates of 69.6% and 10.2%, respectively. GIVE_WAY is a complex command which can be given using very different phraseology, not all of which are modelled, thereby leading to lower command recognition rates.

TABLE VI. RECOGNITION RATES PER COMMAND TYPE

Command Type	#Cmds	RecR	ErrR	RejR
TAXI VIA	2922	86.9%	3.9%	9.1%
HOLD_SHORT	1837	89.3%	0.8%	9.9%
TAXI TO	1406	89.0%	1.1%	9.9%
CONTACT_FREQUENCY	1387	95.7%	0.7%	3.6%
CONTINUE TAXI	1102	95.4%	0.0%	4.6%
GIVE_WAY	728	69.6%	10.2%	20.3%
CONTACT	672	98.4%	0.3%	1.3%
PUSHBACK	663	92.3%	1.2%	6.5%
TURN	359	89.2%	3.9%	6.9%
HOLD_POSITION	223	93.4%	0.0%	6.6%

As mentioned before, using VAD incurs a certain number of errors in identifying the start and end of a transmission. Since we are ultimately interested in extracting the correct concepts, we try to fix this problem at the concept-level by automatically merging recognitions which seem to belong to the same utterance. On day 1 of the final simulation trials this automatic merging process was not applied, the CaRecR for that day deteriorated from 98.6% to 95.4% using the VAD. This means there was a decrease of 3.2% absolute when using VAD. On the other hand, on the remaining days with the automatic merging, the CaRecR only reduced from 96.7% to 95.1%, meaning the decrease was only around 1.6% absolute, i.e., merging on command level partially compensates the errors induced by VAD and makes the system more robust.

Table VII shows how many HMI interactions remain for controllers and pilots in runs with ABSR compared to baseline runs without ABSR. The total shows a clear decrease which is about 85% for controllers and 60% for pilots. The average number of inputs performed for some classes of commands, e.g., taxi, pushback and hold short, dropped quite dramatically. This proves that many required inputs are automated through ABSR. The lower decrease for pilots is probably due to the additional tasks in controlling the simulation and the fact that pilots must correct errors while controllers were observed to skip some corrections.

TABLE VII. PERCENTAGE OF INTERACTIONS REMAINING WITH ABSR

Interaction Type	Remaining HMI interactions with ABSR	
	Controller	Pilot
cancel hold short	3,77%	33,94%
clear pushback	3,51%	16,02%
clear taxi	3,52%	19,31%
give way	18,35%	53,83%
hold short	4,88%	26,27%
select/label/deselect	14,08%	37,63%
routing	20,77%	39,03%
handover	5,58%	22,09%
Total	**15,38%**	**40,39%**

Table VIII shows the average and standard deviation for correctly performed secondary tasks for baseline and ABSR runs. For Center and West, a paired T-test shows significant evidence (alpha 0.9 and 2.0) that the mental load is lower with ABSR, since the average number of correctly performed tasks is higher. For the East position no significant evidence is available.

TABLE VIII. STROOP EVALUATION – AVERAGE AND SIGMA PER POSITION

Position	Average		Sigma	
	Baseline	ABSR	Baseline	ABSR
East	56.5	53.0	49.5	42.5
Center	32.8	51.1	36.3	40.7
West	89.5	113.6	41.0	52.7

The overall NASA TLX workload index was 8.6 on average, 7.5 with ABSR (better) and 9.7 in the baseline. Mental and temporal demand and effort got higher indices in all circumstances than physical demand, frustration and the performance aspect. In all single aspects, the index values are higher, i.e., better with ABSR compared to the baseline (notably, except for 'performance' at East and 'frustration' at West). Considering that controllers estimated their own performance already quite good, the main improvement by ABSR lies in lowering the mental and temporal workload and the effort. The overall SASHA situation awareness index was 4.6 with ABSR

and 4.2 (worse), in the baseline, especially the risk of forgetting and the ability to plan and organize was better with ABSR. The overall automation trust index of 4.6 showed a high trust in the system. Highest rated aspect was usefulness (5.1). Accuracy (4.3) and robustness (4.2) were rated a bit lower.

VII. Conclusion And Outlook

A modern commercial A-SMGCS was combined with ABSR. This increased on the one hand the ABSR performance. Word Error Rates of 3.1% on unseen voice recordings were achieved. The callsign recognition error rate was only 1.3%, when information from the A-SMGCS was used. Command recognition rates of 91.8% were measured, which includes that every single part of a command is correctly extracted.

On the other hand, ABSR could dramatically reduce the manual workload for apron controllers. In the coming years these controllers will have to digitize almost every spoken command, either manually by mouse and keyboard or with ABSR support. The experiments with 14 apron controllers showed a reduction of the needed interactions by a factor of 7. The performance on the secondary task increased by 55% resp. 27% for the Center and West position, whereas it decreased a bit for the East position. This shows that ABSR often enables additional safety buffers, because the time, the apron controller spent on the secondary task, will be available in the operational environment for monitoring and short-term reactions. Feedback from post-run questionnaires also clearly show that the controllers were more flexible and efficient with ABSR. Even during high-workload situations they were more likely to give individual situation-dependent taxi clearances, whereas without ABSR more standard routing was used.

Additionally, also simulation pilots were supported by ABSR. They only had to input commands misrecognized by ABSR, which reduced manual interactions by more than 50%. An additional source of error had to be addressed, since PTT was not available and ABSR had to decide in real-time the start and end of transmissions via acoustics. Offline analysis show that the word error rate can be reduced by 60% relative, i.e., from 5.0% to 3.1%, whereas semantic extraction is quite robust against these errors as the command extraction rate decreases only from 91.8% to 88.7%.

The next steps will be to move ABSR from the simulation to the operational environment of Fraport and to improve the user interface of the simulation pilots so that their manual effort can also be reduced by a factor of 7.

Acknowledgment

The project STARFiSH funded by the German Federal Ministry of Education and Research. Two projects from SESAR2020 industrial research and exploratory research, which have received funding from the SESAR Joint Undertaking under the European Union's grant agreement No. 734141 and 884287. The projects are named PJ.16-04-W1 (CWP HMI) and HAAWAII (exploratory research). The authors also want to thank all the apron controllers from Fraport who actively supported the development of the demonstrated application through contribution of their audio recordings for training purposes and participation in multiple simulation trials throughout the STARFiSH project.

References

[1] H. Helmke, O. Ohneiser, T. Mühlhausen, and M. Wies, "Reducing controller workload with automatic speech recognition," in IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, CA, USA, 2016.

[2] Commission Implementing Regulation (EU) 2021/116 of 1 February 2021 on the establishment of the Common Project One supporting the implementation of the European Air Traffic Management Master Plan provided for in Regulation (EC) No 550/2004 of the European Parliament and of the Council, amending Commission Implementing Regulation (EU) No 409/2013 and repealing Commission Implementing Regulation (EU) No 716/2014.

[3] AcListant® (Active Listening Assistant) http://www.aclistant.de, n.d.

[4] 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.

[5] 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.

[6] 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 N. Stanton, S. Landry, G. Di Bucchianico, and A. Vallicelli (Eds.), Proceedings of the 5th International Conference on Applied Human Factors and Ergonomics AHFE 2014, Advances in Human Aspects of Transportation: Part II, pp. 492-503, Krakow, CRC Press, 2014.

[7] H. Helmke, O. Ohneiser, J. Buxbaum, and C. Kern, "Increasing ATM efficiency with assistant-based speech recognition," in 12th USA/Europe Air Traffic Management Research and Development Seminar (ATM2017), Seattle, WA, USA, 2017.

[8] The project MALORCA (Machine Learning of Speech Recognition Models for Controller Assistance) http://www.malorca-project.de, n.d.

[9] PJ.10-96-W2: SESAR2020 funded industrial research projects under the European Union's grant agreement 874470, https://cordis.europa.eu/programme/id/H2020_SESAR-IR-VLD-WAVE2-10-2019/de, n.d.

[10] O. Ohneiser, H. Helmke, S. Shetty, M. Kleinert, H. Ehr, Š. Murauskas, T. Pagirys, G. Balogh, A. Tønnesen, G. Kis-Pál, R. Tichy, V. Horváth, F. Kling, W. Rinaldi, S. Mansi, G. Piazzolla, H. Usanovic, "Understanding Tower Controller Communication for Support in Air Traffic Control Displays," SESAR Innovation Days 2022, Budapest, 2022.

[11] HAAWAII (Highly Automated Air Traffic Controller Workstations With Artifical Itelligence Integration) https://www.haawaii.de, n.d.

[12] H. Helmke, K. Ondřej, S. Shetty, H. Arilíusson, T. S. Simiganoschi, M. Kleinert, O. Ohneiser, H. Ehr, J.-P. Zuluaga and P. Smrz, "Readback Error Detection by Automatic Speech Recognition and Understanding,, Results of HAAWAII project for Isavia's Enroute Airspace," SESAR Innovation Days 2022, Budapest, 2022.

[13] M. Kleinert, H. Helmke, P. Motlicek, HAAWAII project Deliverable D1.3: "Architecture Design Document," version 01.00.00, 05. Nov. 2020.

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

[15] FAA, 2012 National Aviation Research Plan (NARP), 2012.

[16] D. Schäfer, Context-sensitive speech recognition in the air traffic control simu-lation, München: Eurocontrol EEC Note No. 02/2001 and PhD Thesis of the University of Armed Forces, 2001.

[17] K. B. a. R. R. R. Tarakan, "An automated simulation pilot capabil-ity to support advanced air traffic controller training," 26th Congress of the Intern. Council of the Aeronautical Sciences, Anchorage, AK, 2008.

[18] S. Ciupka, „Siris große Schwester erobert die DFS," transmission, No. Vol. 1, 2012.

[19] J. Cordero, M. Dorado und J. 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), Toulouse, IRIT Press, 2012, pp. 46-53.

[20] J. Cordero, N. Rodríguez, J. de Pablo und M. Dorado, ”Automated Speech Recognition in Controller Communications applied to Workload Measurement,“ in 3rd SESAR Innovation Days, Stockholm, 2013.

[21] S. C. a. 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, 2015.

[22] S. Chen, H. Kopald, A. Elessawy, Z. Levonian und R. Tarakan, ”Speech Inputs to Surface Safety Logic Systems,“ in IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), Prague, 2015.

[23] S. Chen, H. Kopald, R. Chong, Y. Wei and Z. Levonian, ”Read Back Error Detection using Automatic Speech Recognition,“ in 12th USA/ Europe Air Traffic Management Research and Development Seminar (ATM2017), Seattle, 2017.

[24] H. Said, M. Guillemette, J. Gillespie, C. Couchman and R. Stilwell, ”Pilots & Air Traffic Control Phraseology Study,“ in International Air Transport Association, 2011.

[25] S. Young, W. Ward and A. Hauptmann, ”Layering predictions: Flexible use of dialog expectation in speech recognitionMorgan Kaufmann, 1989, pp.,“ in Proceedings of the 11th Intern. Joint Conference on Artificial Intelligence (IJCAI89), Morgan Kaufmann, 1989, pp. 1543-1549.

[26] S. Young, A. Hauptmann, W. Ward, E. Smith and P. Werner, ”High level knowledge sources in usable speech recognition systems,” in Commun,“ ACM, Bd. 2, Nr. 32, pp. 183-194, 1989.

[27] H. Helmke, M. Slotty, M. Poiger, D.F. Herrer, O. Ohneiser et al., “Ontology for transcription of ATC speech commands of SESAR 2020 solution PJ.16-04,” IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), London, United Kingdom, 2018.

[28] M. Kleinert, H. Helmke, S. Shetty, O. Ohneiser, H. Ehr, A. Prasad, P. Motlicek, J. Harfmann; “Automated Interpretation of Air Traffic Control Communication: The Journey from Spoken Words to a Deeper Understanding of the Meaning,” IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, 2021.

[29] ICAO Advanced Surface Movement Control and Guidance Systems (A-SMGCS) Manual, Doc 9830 AN/452, First Edition 2004.

[30] Lisanne, Bainbridge, "Ironies of automation," Automatica, Vol.19, No. 6, pp. 775-779, 1983.

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

[32] H. Helmke, S. Shetty, M. Kleinert, O. Ohneiser, H. Ehr, A. Prasad, P. Motlicek, A. Cerna, and C. Windisch, “How to Measure Speech Recognition Performance in the Air Traffic Control domain? The Word Error Rate is only half of the truth!,” Satellite Workshop of Interspeech, Brno, Czech Republic, 2021.

[33] S. G. Hart, “NASA-task load index (NASA-TLX); 20 years later,” Proceedings of the human factors and ergonomics society annual meeting, pp. 904-908, Sage CA: Los Angeles, CA: Sage publications, 2006.

[34] D. M. Dehn, “Assessing the impact of automation on the air traffic controller: the SHAPE questionnaires,” Air traffic control quarterly 16.2, pp. 127-146, 2008.

[35] J. R. Stroop, “Studies of interference in serial verbal reactions,” Journal of experimental psychology 18.6, pp. 643-662, 1935.

[36] S. M. Casner, B. F. Gore, “Measuring and evaluating workload: A primer,” NASA Technical Memorandum, 216395, 2010.

[37] A. Stolcke, J. Droppo, “Comparing human and machine errors in conversational speech transcription”, in Proc. Interspeech, pp. 137– 141, Stockholm, 2017