Gap Analysis Report

D5-2

MALORCA

Grant:	**698824**
Call:	**H2020-SESAR-2015-1**
Topic 2:	**Data Science in ATM**
Consortium coordinator:	**DLR**
Edition date:	**28. February 2018**
Edition:	**1.00**

Founding Members
EUROPEAN UNION
EUROCONTROL

SESAR
JOINT UNDERTAKING

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Hartmut Helmke	Project Leader	2017-12-12
Gerald Siol	D3-1 leader	2017-12-13
Matthias Kleinert	Contributor	2018-02-05
Heiko Ehr	WP5 Leader	2018-02-26
Aneta Cerna	WP1 Leader	2018-02-08
Christian Kern	Dissemination Manager	2018-02-08
Dietrich Klakow	USAAR	2018-02-22
Petr Motlicek	WP4 Leader	2018-02-23

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
Heiko Ehr	WP5 Leader	2018-02-26

Approved for submission to the SJU By — Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
Hartmut Helmke	Project Lead	2018-02-18

Rejected By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date

Document History

Edition	Date	Status	Author	Justification
0.01	2017-04-24	Draft	Hartmut Helmke	Initial Draft
0.01	2017-12-12	Draft	Hartmut Helmke	
0.01	2017-12-13	Draft	Gerald Siol	App. A.1, C.2 - C.5

	2017-12-31	Draft	Hartmut Helmke	Review and improvement of Appendix
	2018-01-09	Draft	Gerald Siol	App. B.5 code example changed
0.02	2018-01-14	Draft	Hartmut Helmke	Results of T2 started, empty chapter for T1, O1, O2 and conclusions added
	2018-01-15	Draft	Gerald Siol	App. B.2-B.5 corrections because code changes
0.03	2018-02-03	Draft	Hartmut Helmke	Results of O2 added
	2018-02-04	Draft	Hartmut Helmke	Results of O1 added
	2018-02-05	Draft	Heiko Ehr	Description of O1/O2 improved
	2018-02-05	Draft	Matthias Kleinert	Description of O1/O2 improved
	2018-02-05	Draft	Hartmut Helmke	Review of input from Heiko and Matthias
0.04	2018-02-06	Draft	Heiko Ehr / Matthias Kleinert	Description O1/O2/T1 improved
0.05	2018-02-08		Aneta Cerna	T1 chapter text, A1 and A2 appendix
	2018-02-08	Draft	Hartmut Helmke	Abstract added
0.06	2018-02-21	Draft	Hartmut Helmke	T2 2nd iteration added, acceptance of all previous changes, review of rest
	2018-02-22	Draft	Dietrich Klakow	Draft of Major Gaps chapter
0.07	2018-02-23	Draft	Hartmut Helmke	Acceptance of all changes of DK in Major Gaps chapter and then review
0.08	2018-02-23	Draft	Petr Motlicek	Review and update for sections 1 - 4
0.09	2018-02-24	Draft	Hartmut Helmke	Acceptance of all prev. of HHe and DK and review of PM updates from day before
			Hartmut Helmke	Gap analysis of DK with respect to precision and recall) in ch.3 moved to chapter 4 and improved
0.10	2018-02-23	Draft	Petr Motlicek	Review and update for section 5 – 9, abbreviations
0.11	2018-02-26	Draft	Heiko Ehr	Review
	2018-02-26	Draft	Aneta Cerna	T1 chapter changes
0.96	2018-02-26	Draft	Hartmut Helmke	Acceptance of all changes accept those from Helmke, deleting all comments, even if not processed, conclusion chapter added
	2018-02-27	Draft	Dietrich Klakow	Review ch. 8 and 9
	2018-02-27	Draft	Hartmut Helmke	Fixing comments of DK
1.00	2018-02-28	Final	Hartmut Helmke	Creating final version

MALORCA

MACHINE LEARNING OF SPEECH RECOGNITION MODELS FOR CONTROLLER ASSISTANCE

This project has received funding from the SESAR Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 698824.

Abstract / Executive Summary

MALORCA's Proof-of-Concept is split into two technical (T1, T2) and two operational (O1, O2) activities. T1 is a workshop with technical experts to evaluate the ABSR prototype implementation. T2 is an offline evaluation to quantify the improvements of the Assistant Based Speech Recognizer (ABSR system) with respect to the amount of available training data. O1 involves controllers who concentrate only on the different outputs of a baseline ABSR system and a trained system. O2 puts the trained ABSR system in a simulation environment with a replay of historic radar data and controller voice real recordings from Prague and Vienna. ABSR is used here to support the controllers in maintaining radar labels. D5-2 first presents the outcomes of the four activities, and the controller's feedback in debriefing sessions and finally analyses the results with respect to gaps and further challenges.

Table of Contents

1 Introduction

1.1 Purpose of the document

This document is the deliverable D5-2 of the MALORCA project to the SJU[1] and describes the results of the proof-of-concept trials performed in January in Prague and Vienna with controllers and experts from DLR, Austro Control and ANS CR. Furthermore the technical part T2 is described performed from December 2017 to February 2018 at Idiap, USAAR and DLR premises.[2] It contains the work organization, scenario development, POC planning and execution. Each aspect is covered in a separate chapter or section. In addition to the experiment the POC also described the metrics for the data analysis.

1.2 Intended readership

The following readers internal to the MALORCA project could be interested at this POC:

- MALORCA partners

Even though this document is confidential, the following external readers might be interested in this document:

- SESAR Joint Undertaking
- Airport owners/providers

[1] The opinions expressed herein reflect the authors' view only. Under no circumstances shall the SESAR Joint Undertaking be responsible for any use that may be made of the information contained herein.

[2] In grant this deliverable D5-2 was promised to be prepared by Idiap. The deliverable lead of D5-2, however, changed to DLR in order to be able to meet the deadline, see also list of authors of the document.

1.3 Structure of the document

Chapter 1: (this section) describes the purpose and scope of the document and the intended readership.

Chapter 2 described the technical Proof-of-Concept Trial T1 itself and the results.

Chapter 3 and 0 describes the two iterations and its results of the technical proof-of-concept phase T2. Chapter 5 and 6 describe the two operation Proof-of-Concepts Trials O1 and O2 and their results. The feedback from the controllers is summarized in chapter 7. The detailed gap analysis especially with respect to achieving higher TRL levels is presented in chapter 8 before the last chapter, which contains the conclusions.

The appendix contains more details with respect to the four Proof-of-Concept trials..

2 Technical Proof-of-Concept T1

2.1 Experimental Setup

The setup for the life demo is almost the same as described in O2 in section 6. The configurations for Prague and Vienna are shown in Table 22 and Table 24. The pre-recorded audio files are replaced by a microphone which is used as input for the ABSR system. Commands are given by a controller into the microphone and the ABSR system tries to recognize them. As shown in Figure 18 the output of the ABSR system is given in the radar label. The speech log (in upper left corner in Figure 18) is shown to the controller to check the recognition output. During the exercises the software was modified so that every recognized command is highlighted in green (not only those that are within the situational context (e.g. plausible) in the current situation), because for testing purposes the controllers also attempted some commands that are unusual for the presented radar situation. Those commands are of course not predicted by the ABSR system and would therefore normally be rejected by the plausibility checker component.

2.2 Results

2.2.1 Test Scenario 1

The goal was to demonstrate how the actual prototype configuration fulfils predefined requirements and to verify that the prototype is generally compliant with generic operational requirements in D1-2.

- Testing location: Prague
- Testing platform: provided by DLR
- Participants: Hartmut Helmke, Matej Nesvadba, Aneta Cerna, Christian Windisch, Petr Motlicek
- Date: 23-24.1.2018

Basically all the requirements with the SHALL priority were selected for T1 testing if not stated otherwise. The complete checklists of selected requirements for T1 testing are presented in Appendix A.1. If the requirement was not a subject of test, the reason was given.

The results are presented in the chapter 6 of D1-2 (“Proof of concept results”) and main findings are summarized here:

- Vertical movements:
  - *Stop descend* command was not recognized.

    Reason: This never appeared in transcribed data, so that it was not modelled.
  - *Maintain altitude* command was not recognized.

    Reason: The command was modelled, but in this case not expected by command predictor and so it did not appear in the HMI.
- Speed reduction (example: 220kts):

  According the ICAO phraseology, speed instruction should follow the format of: INCREASE (or REDUCE) SPEED TO (number) KILOMETRES PER HOUR (or KNOTS) [OR GREATER (or OR LESS)].

  In case of a speed reduction to 220 knots, the prototype gives incorrectly the output of 2220kts.

  Reason: The common way of giving speed reduction was different from that, usually omitting the "TO" before (number). The range of possible speed number should be limited.
- Intercept localizer – It was correctly recognized but sent to the user as Cleared for ILS approach.

  Reason: It was present only once in transcribed training data -> incorrectly assign to Cleared for ILS approach command.
- Expect RWY – It was recognized as expect ILS.

  Reason: It was recognized only in 1 of 4 cases, due to lack of training data provided within MALORCA, performance is low, and additionally it was excluded from training itself.
- Corrections (e.g. command value correction)

  Reason: callsign correction was recognized correctly, correction of command value was recognized as the correction as such but no impact to the user output.
- Reaction time:

  The focus of MALORCA was on machine learning and related improvements of ASR. The reaction time during the validation shows higher values (than required) which is not acceptable for further validations on higher TRL or for the operational use. Regarding the actual TRL (low), it is acceptable and does not affect MALORCA validations and reach MALORCA objectives.

  Briefing with controllers shows that even two seconds do not need to be sufficient. However, the pilot feedback was not included so that controller got the impression that ASR needs to be much faster.

2.2.2 Test Scenario 2

The goal was to demonstrate how the actual prototype configuration fulfils predefined requirements and verify that the prototype is generally compliant with the technical requirements in D1-2 Annex.

- Testing location: Prague
- Testing platform: DLR
- Participants: Christian Windisch, Aneta Cerna, Hartmut Helmke, Mittul Singh, Petr Motlicek

- Date: 23-24.1.2018

The complete checklists of selected requirements are presented in Appendix A.2. If the requirement was not a subject of test, the reason was given.

The results are presented in chapter 5.2 in D1-2 Annex and main findings are summarized here:

- Noise of Audio data:

  Real data shows the difference in Signal-to-Noise Ratio (SNR) level (approximately 17dB VIE, 22dB PRG, estimated manually from few audio recordings), which essentially correlates to ASR performance. If we want to increase SNR, the quality of data needs to be improved as well. Even though the requirement was not fully fulfilled, it does not block to reach objectives; however, the difference is clearly visible in T2 results.

3 First Iteration of Technical Proof-of-Concept T2

3.1 Experimental Setup

The following Figure 1 shows the general setup. First we use the untranscribed data from August 2016 and learn from them. This results in an ABSR System called ABSR August. We use this system and evaluate with all the testing data and get our metrics of August. Then we add the untranscribed data from September and retrain our models on the data of August and September. We get a system ABSR September and determine the rates. Further we continue with the data from October and November. In detail we do not use the data from one month, but rather we select the data according to the total length of the dataset: practically, we first select 25%, then the first 50%, then the first 75% and then all (100%) data. In fact, this is nearly equivalent to taking the data on monthly basis.

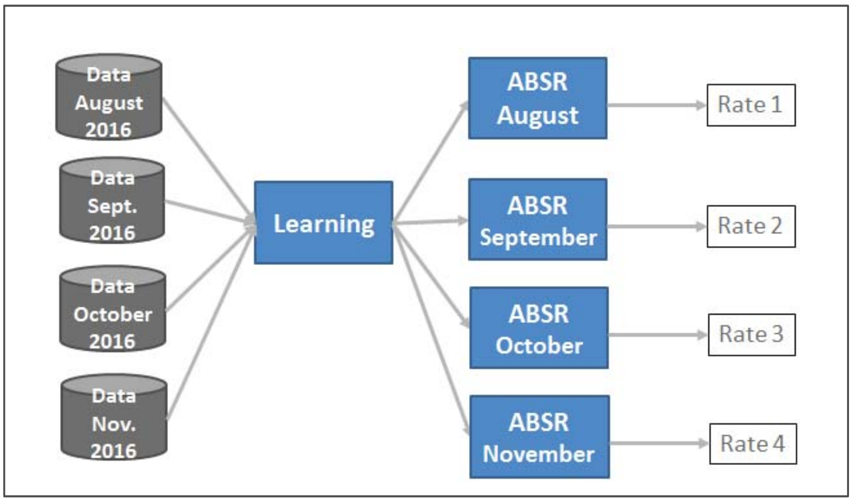

Figure 1: General setup for iterative improvement of recognition models

Figure 2 shows our approach in more detail. We start with no (0%) untranscribed data and add 70% of the transcribed data and we train both the acoustic and language models (AM, LM) of ASR engine. For the command prediction model (CPM) we use 10% of the untranscribed data, but we exclude transcribed data. With this baseline model (model-0%) we evaluate the command recognition, command recognition error rates, etc. The baseline recognizer is used to transcribe all untranscribed wave files and the command prediction model classifies all the command outputs into “good” and “bad” learning examples. Generated command hypotheses which are not predicted are classified as bad examples. Further we use 25% of the untranscribed data plus the 70% of the transcribed data to improve/train the AM and LM. We obtain ASR_1 system and apply it for automatic transcription of the untranscribed data. 25% of the untranscribed data (only this) is used to create an improved CPM, which is again used to classify all the transcriptions resulting from ASR_1 into good and bad examples. We repeat the process with 50% of the data, etc.

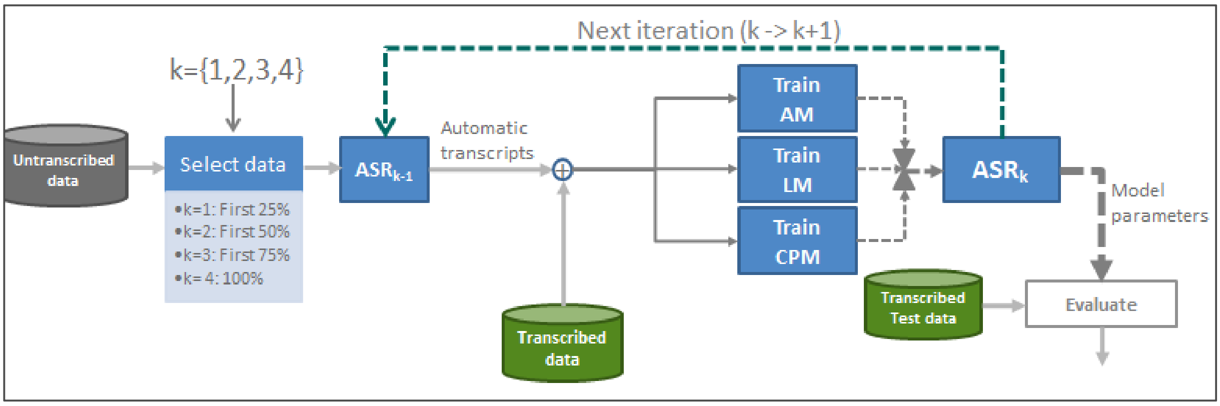

Figure 2: Detailed setup for iterative improvement of recognition models

3.2 Results

Table 1 shows the results when we first train the three models (AM, LM, CPM) with no untranscribed data (row 2 "0%/10%") resp. 10% for the CPM. Row 3 ("25%") shows the results when we train the ABSR system with 25% of the untranscribed data, etc.

	Command RecognRate	Command Error Rate	Pure Command Recogn Rate	Pure Command Error Rate	ctxErrRate	ctx avg
0%/10%	77.7%	0.44%	84.6%	7.64%	8.7%	225
25%	83.1%	0.48%	86.9%	6.43%	5.3%	303
50%	85.2%	0.52%	88.0%	6.30%	4.1%	357
75%	86.1%	0.53%	88.3%	5.94%	3.5%	396
100%	86.2%	0.78%	88.2%	6.34%	2.9%	412

Table 1: Iterative updates of Prague models by adding 25% of new untranscribed data.

Columns 2 to 3 report the command recognition rate (Command RecognRate) and command recognition error rate (Command ErrorRate), if the plausibility checker is applied (i.e. situational context is generated by command hypotheses generator and checked against the command generated by ASR module. Columns 4 to 5 show recognition and error rates if the plausibility checker is not applied (i.e. we apply only a single filter to reject NO CALLSIGN NO CONCEPT commands). In this case, we get an improvement in error rate by 5.56% absolute (since less commands are rejected), however, this significantly increases in recognition error rates (as situational context is not applied). Figure 3 also shows the differences between column 2 and 3 on the one hand and column 4 and 5 on the other hand.

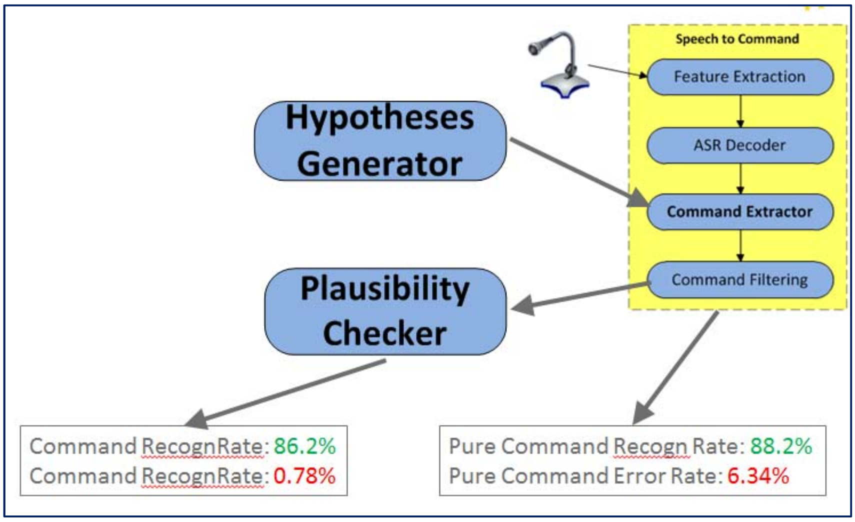

Figure 3: Difference of Pure and other rates

Column "ctxErrRate" shows the percentage of commands given by the controllers which were not predicted by the command hypotheses generator (i.e. CPM) and "ctx avg" shows the average (absolute) number of commands which were predicted by the CPM

Figure 4 and Figure 5 graphically show the results of the Table 1.

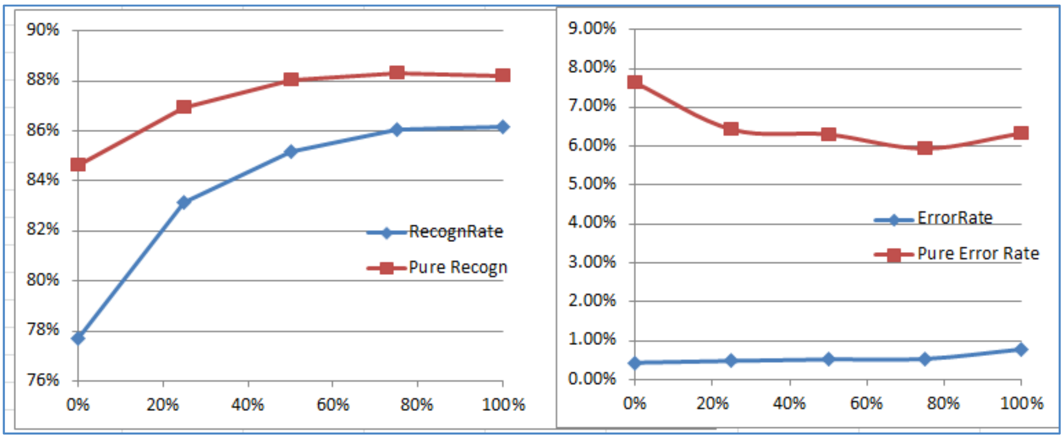

Figure 4: Learning Curve of Command Recognition and Command Error Rate for Prague.

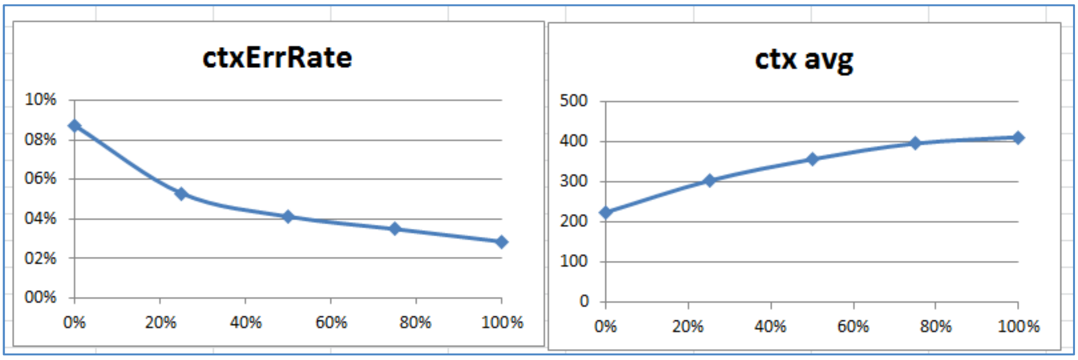

Figure 5: Learning Curve of Command Prediction Error Rate (ctxErrRate) and Number of Predicted Commands (ctx avg) for Prague.

We present the same results for Vienna in Table 2, Figure 6 and Figure 7.

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
0%/10%	58.0%	1.9%	65.5%	14.9%	17.8%	419
25%	62.4%	2.2%	66.9%	14.4%	12.2%	549
50%	65.8%	2.3%	68.6%	13.4%	9.6%	647
75%	67.6%	2.7%	70.0%	12.1%	8.7%	708
100%; iter1	68.8%	2.6%	70.8%	11.7%	8.0%	773

Table 2: Iterative Improvement of **Vienna** models by adding 25% of new untranscribed data

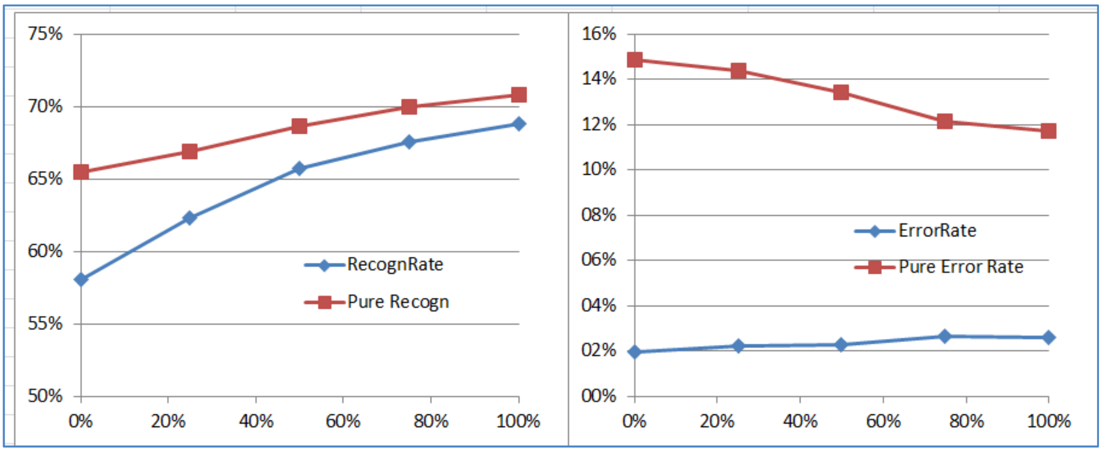

Figure 6: Learning Curve of Command Recognition and Command Error Rate for Vienna.

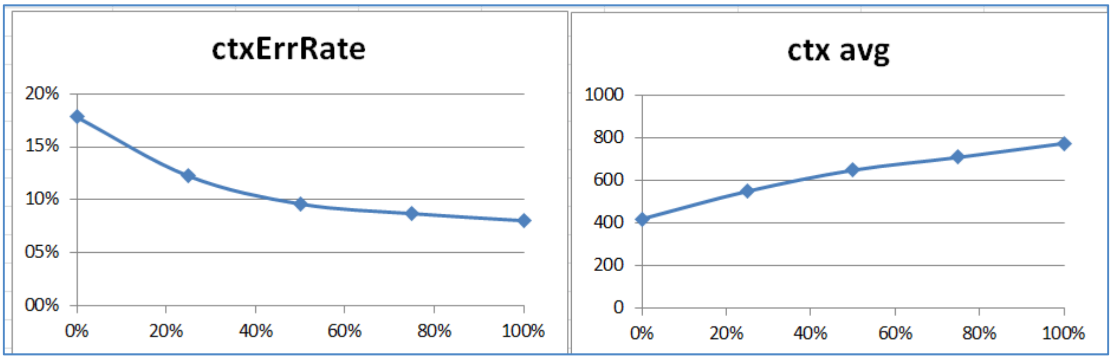

Figure 7: Learning Curve of Command Prediction Error Rate (ctxErrRate) and Number of Predicted Commands (ctx avg) for Vienna.

3.3 Interpretation of Results

In the following section we analyse the data presented in Table 1 and Table 2. When needed deeper analysis is performed.

1. The learning curve of Vienna does not reach its saturation limits. Increasing data size by a factor of 2 (from 25% to 50% and from 50% to 100%) still improves the performance. The Command Recognition Rate increases by 3.4% (absolute) from 25% data size to 50% data size and again by 3.1% (absolute) from 50% to 100%. Table 3 shows the differences (deltas) for all measurements of Table 2. In red we mark situations when the measurement results in a performance decrease (i.e. an increase of command recognition error rate is a decrease of performance).
   In Table 4: Extrapolation of measurements for more available data for Vienna, we extrapolate our data of Table 2 by the values used in Table 3 assuming that we get 200%, 400% or 800% of the currently available data.

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
25%-->50%	3.4%	-0.1%	1.8%	1.0%	2.6%	-99
50%->100%	3.1%	-0.3%	2.2%	1.3%	1.6%	-126

Table 3: Differences (deltas) in performance for measurements when data size is increased by a factor of 2 for Vienna.

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
100%	68.8%	2.6%	70.8%	11.7%	8.0%	773
200%	71.6%	3.1%	72.8%	10.6%	7.4%	923
400%	74.1%	3.3%	74.8%	9.4%	7.4%	1093
800%	76.3%	3.6%	76.8%	8.3%	7.4%	1283

Table 4: Extrapolation of measurements for more available data for Vienna.

2. For Prague data it seems that we have also not reached saturation, i.e. if we perform the same analysis as for Vienna data we achieve the results presented in Table 5 and Table 6 .

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
25%-->50%	2.0%	-0.04%	1.1%	0.1%	1.2%	-54
50%->100%	1.0%	-0.26%	0.2%	0.4%	1.3%	-55

Table 5: Difference (delta) in performance for measurements when data size is increased by a factor of 2 for Prague.

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
100%	86.2%	0.78%	88.2%	6.3%	2.9%	412
200%	86.7%	1.28%	88.3%	5.9%	2.9%	467
400%	86.9%	1.98%	88.4%	5.5%	2.9%	522
800%	87.0%	2.88%	88.5%	5.1%	2.9%	577

Table 6: Extrapolation of Measurements for more available data for Prague.

3. If we however analyse only the Prague performance when augmenting the data from 75% to 100%, it seems that we have reached already saturation, i.e. the currently available 100% of learning data is already sufficient to reach the best performance. Table 7 shows a very slight improvement for recognition rate, but indicates much higher increase in terms of the command recognition error rate.

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
75%	86.1%	0.53%	88.3%	5.94%	3.5%	396
100%	86.2%	0.78%	88.2%	6.34%	2.9%	412

Table 7: Prague measurements of Table 1 only for 75% and 100%.

4. In November 2016, Prague frequency values have changed. Most of the additional learning data added from 75% to 100% use these new frequencies. Test set using the new frequencies is now covered much better whereas performance on “old” data decreases. Table 8 shows the results for the case when we consider only test data after the new frequencies are active. We recognize a significant improvement in all measurements.
   → If the domain knowledge (i.e. configuration of approach) changes, system retraining is necessary to avoid a performance decrease.

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
0%/10%	73.4%	0.36%	82.8%	9.0%	12.8%	284
25%	76.1%	0.36%	83.4%	8.6%	10.7%	361
50%	75.7%	0.36%	82.2%	9.4%	9.6%	452
75%	76.4%	0.36%	82.3%	8.0%	9.0%	452
100%	79.7%	1.7%	83.0%	7.9%	4.8%	474

Table 8: Prague measurement if we consider only testing data with new frequency values.

5. The command prediction error rate ("ctxErrRate") of 4.8% (see Table 8, last row) is still higher than the average rate of 2.9% (see Table 7, last row) for the total Prague data. We observe that we reach a command prediction error rate of 8.5% for the AEC controller and of 1.1% for the PEC controller. More detailed analysis indicates that (as already reported in D3-6) two different recording conditions for the AEC controller exist: (1) the AEC controller is supported by a PEC controller. In that case a handover to the PEC controller is done by the AEC controller (phraseology HANDOVER RADAR/DIRECTOR and HANDOVER_FREQUENCY 119.01). (2) We have low traffic and the AEC controller also performs the job of the PEC controller. Aircrafts are directly handovered from AEC to tower position phraseology (HANDOVER TOWER and HANDOVER_FREQUENCY 118.11). No learning data for the first case was provided, so the HANDOVER frequency 119.01 was never predicted. The handover to "RADAR/DIRECTOR" was hypothesised, because the position name did not change. Only the frequency value was changed. .
   → We manually add the frequency value 119.01 to the set of possible frequencies. Command prediction error rate could be reduced from 7.6% to 4.8% (LKPR12) and from 5.8% to 0.0% (LKPR05).

6. Prague command prediction error rate is much better than for Vienna (2.9% versus 8.0%). One reason is of course the higher command recognition rate for Prague. However, also the standard deviation for Vienna is higher (6.3% versus 2.8%):

   a. Prague radar data covers only 60 to 90 minutes of runtime. Vienna radar data sometimes covers more than 14 hours, i.e. a callsign which is landing at Vienna airport may start in the same data set. The ARR/DEP classification of a callsign, however, is kept for the whole day.

We manually corrected ARR which got a CLIMB command and correct also DEP which got a DESCEND or CLEARED_ILS command.[3]

b. Only runway 34 for arrivals and runway 29 for DEP were modelled. In some scenarios however both runway 34 and runway 29 were used for inbounds. Commands for these aircraft were often wrongly predicted. We exclude all directories from evaluation which contain a significant number of inbounds at runway 29.

7. Table 1 Table 2 and Table 8 show an increase in command error rate whereas ctxErrRate decreases. More data provides a better recall for context (more alternatives are covered) whereas the precision (that is the quality of the best predicted command) is degrading. This indicates that the prediction of the commands needs improvements.

[3] We applied this heuristics also to Prague data resulting in 10 less command prediction errors (168-158).

4 Second Iteration of Technical Proof-of-Concept T2

4.1 Experimental Setup

The problem analysis performed in section 3.3 results in many error corrections and the whole learning cycle with 0%, 25% … 100% was repeated (see Figure 2) resulting in the following improved results.

4.2 Results

Table 9 shows the results when we first train the three models (AM, LM, CPM) with no untranscribed data (row 2 “0%/10 %”) resp. 10% for the CPM. Row 3 (“25 %”) shows the results when we train with 25% of the untranscribed data etc.

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
0%/10%	79.8%	0.29%	85.9%	7.90%	8.1%	158
25%	90.2%	0.32%	93.7%	2.2%	4.4%	255
50%	91.3%	0.37%	93.5%	2.3%	3.0%	329
75%	91.7%	0.45%	93.6%	2.4%	2.5%	383
100%	91.9%	0.60%	93.7%	2.4%	2.3%	396

Table 9: Iterative Improvement of **Prague** models by adding 25% of new untranscribed data.

The semantics of all columns was already described in the previous chapter. We get an improvement from Pure Command Error Rate to Command Error Rate by 1.80% (resp. 5.56% in previous iteration) absolute and only a decrease in recognition rate by 1.8 if we use command hypotheses generator (i.e. CPM) always for the 100% case.

Column “ctxErrRate2 shows the percentage of commands given by the controllers which were not predicted by the CPM and “ctx avg” shows the average number of commands which were hypothesised by the command hypotheses generator.

Figure 8 and **Figure 9** graphically show the results of Table 9.

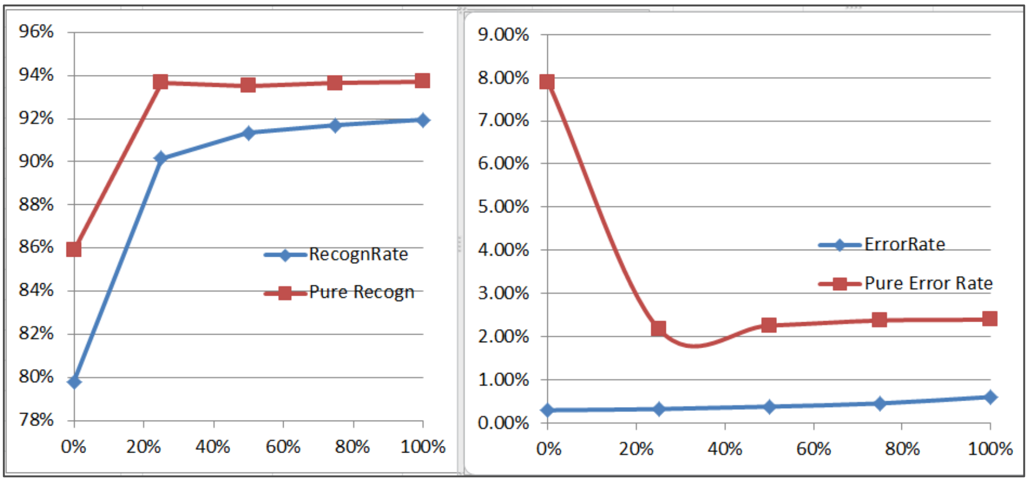

Figure 8: Learning Curve of Command Recognition and Command Error Rate for Prague

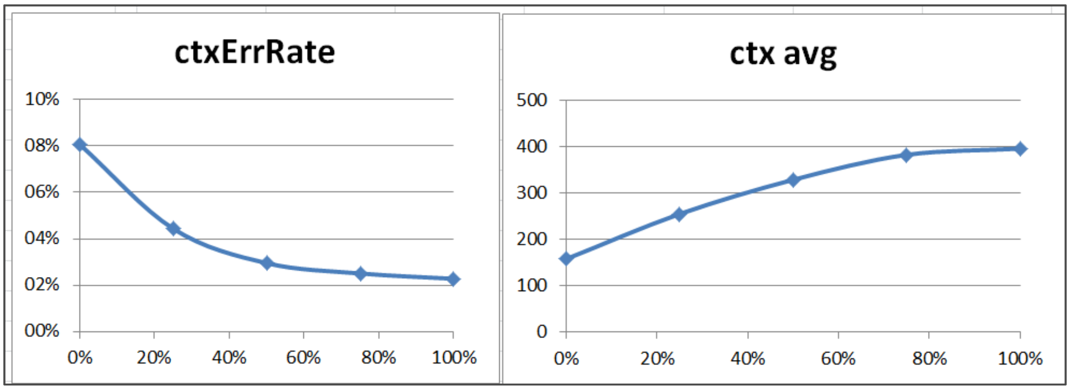

Figure 9: Learning Curve of Command Prediction Error Rate (ctxErrRate) and Number of Predicted Commands (ctx avg) for Prague approach.

We present the same results for Vienna in **Figure 10** and **Figure 11** and in Table 10.

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
0%/10%	60.0%	1.6%	67.2%	18.9%	15.2%	234
25%	80.2%	3.5%	84.0%	7.4%	6.7%	477
50%	82.4%	2.8%	84.7%	6.7%	4.6%	644
75%	84.2%	3.0%	85.6%	7.0%	3.5%	796
100%; iter1	85.2%	3.2%	86.4%	6.6%	3.2%	891

Table 10: Iterative Improvement of **Vienna** models by adding 25% of new untranscribed data.

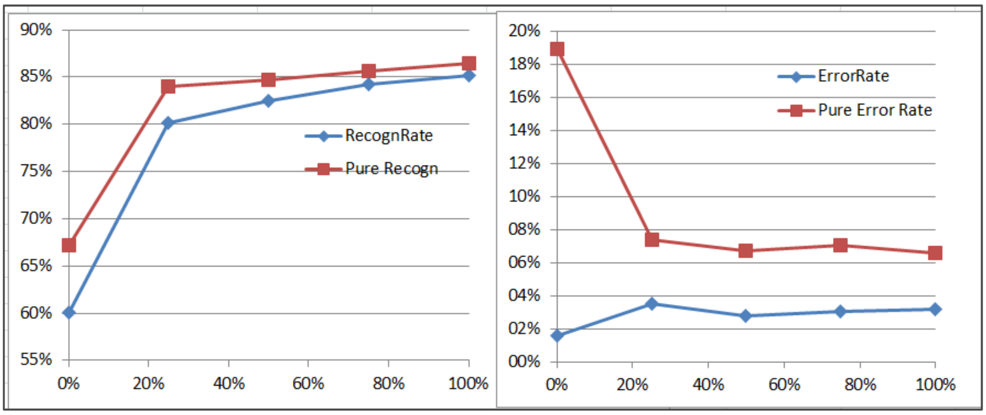

Figure 10: Learning Curve of Command Recognition and Command Error Rate for Vienna.

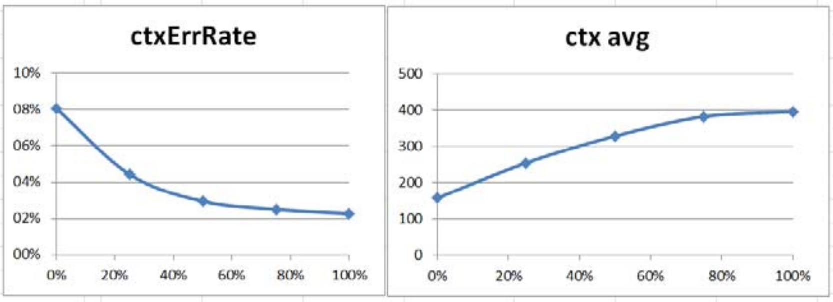

Figure 11: Learning Curve of Command Prediction Error Rate (ctxErrRate) and Number of Predicted Commands (ctx avg) for Vienna.

4.3 Interpretation of Results

In the following section we analyse the data presented in Table 9 and Table 10. When required, a more detailed analysis is performed.

1. The learning curve of Vienna does not reach its saturation limits. Increasing data size by a factor of 2 (from 25% to 50% and from 50% to 100%) still improves the values. The Command Recognition Rate increases by 2.3% (absolute) from 25% data size to 50% data size and again by 2.7% (absolute) from 50% to 100%. Table 11 shows the difference in performance (deltas) for all measurements of Table 10. In red we highlight situations when the measurement results in a performance decrease (an increase of command recognition error rate is a decrease of performance).
   In Table 12 we extrapolate our data of Table 10 by the values used in Table 11assuming that we get 200%, 400% or 800% of the currently available data.

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
25%-->50%	2.3%	0.7%	0.7%	0.7%	2.1%	-167
50%->100%	2.7%	-0.4%	1.7%	0.1%	1.3%	-247

Table 11: Delta for measurements when data size is increased by a factor of 2 for Vienna.

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
100%	85.2%	3.2%	86.4%	6.6%	3.2%	891
200%	86.7%	3.6%	87.9%	6.5%	2.7%	1141
400%	88.2%	4.0%	89.4%	6.4%	2.2%	1391
800%	90.2%	4.4%	90.9%	6.3%	1.7%	1641

Table 12: Extrapolation of measurements for more available data for Vienna.

2. For Prague data it seems that we have also not reached saturation, i.e. if we perform the same analysis as for Vienna data we get the results presented in Table 13 and Table 14.

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
25%-->50%	1.2%	-0.06%	-0.1%	-0.1%	1.5%	-74
50%->100%	0.6%	-0.22%	0.2%	-0.1%	0.7%	-67

Table 13: Delta for measurements when data size is increased by a factor of 2 for Prague.

	RecognRate	ErrorRate	Pure Recogn	Pure Error Rate	ctxErrRate	ctx avg
100%	91.9%	0.60%	93.7%	2.4%	2.3%	396
200%	92.3%	0.80%	93.8%	2.3%	1.9%	466
400%	92.5%	1.10%	93.9%	2.2%	1.7%	536
800%	92.6%	1.30%	94.0%	2.1%	1.6%	606

Table 14: Extrapolation of measurements for more available data for Prague.

3. If we however analyses only the Prague improvements from 75% to 100%, it seems that we have already reached a saturation, i.e. the currently available 100% of learning data is already sufficient.

Table 15 shows a very slight improvement for recognition rate, but a higher increase of the command recognition error rate.

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
75%	91.7%	0.45%	93.6%	2.4%	2.5%	383
100%	91.9%	0.60%	93.7%	2.4%	2.3%	396

Table 15: Prague measurement of Table 1 only for 75% and 100%.

4. Again as already reported earlier, change of frequencies which occur in November 2016 has a strong impact on the precision. Deeper analysis with more data recorded after Nov. 2016 would be necessary to accept or reject this hypothesis. .

	Command RecognRate	Command Error Rate	Pure Command RecognRate	Pure Command Error Rate	ctxErrRate	ctx avg
0%/10%	78.5%	0.18%	81.7%	9.7%	5.7%	170
25%	86.0%	0.38%	88.4%	4.4%	4.2%	263
50%	87.7%	0.27%	88.6%	4.0%	2.3%	338
75%	87.0%	0.8%	88.2%	4.4%	2.1%	390
100%	88.6%	1.9%	89.7%	4.4%	1.8%	405

Table 16: Prague measurements if we consider only testing data with new frequency values recorded in 2017.

5. The command prediction error rate (“ctxErrRate”) of 1.8% (see Table 16, last row) is now better than the average rate of all sessions with 2.3% (see Table 15, last row) for the total Prague data.

6. Prague command prediction error rate is still higher than for Vienna (1.8% versus 3.2%). The difference is much smaller. The reason for this has already been reported in chapter 3 (callsigns are inbounds and outbounds and some data is for runway 29).

7. Command Recognition Error rates (only column “Command Error Rate”) go up when the amount of data is increased even though recognition rate goes up: as the amount of data is increased the available context is increased significantly as also can be seen from the table. However, this increases the space of command alternatives. The model does not differentiate between good extensions of the context on not so good (or not so likely) extensions of the context. In technical terms, the recall (incorrectly rejected commands) in context goes up while the precision (number of falsely accepted commands) decreases. This means that the context model would need to be improved, to make better predictions.

4.4 No context versus static versus dynamic context integration

We also evaluated the effect of (1) using no information from the CPM (aka command prediction model as output from command hypotheses generator) when performing automatic transcription, (row "no context"), (2) using only the callsign information (row "static context") and (3) using also the predicted commands and values for a callsign (row "dynamic context")[4]. Table 17: Results when using no context, static or dynamic context for Prague evaluation data based 100% of the untranscribed learning data. Table 17 shows the results.

	Command RecognRate	Command Error Rate	Pure Command Recogn	Pure Command Error Rate
no context	85.74%	0.50%	87.54%	6.68%
static context	91.84%	0.57%	93.82%	2.01%
dynamic context	91.60%	1.33%	93.01%	2.85%

Table 17: Results when using no context, static or dynamic context for Prague evaluation data based 100% of the untranscribed learning data.

It should be highlighted again: In row "no context" all reported rates used **no** context information (also pure column 4 and 5). In the rows "static context" and "dynamic context" all reported rates depend on the usage of context information. The difference between of "pure" is just before and after using the "plausibility checker" component (see Figure 12 and Figure 13).

[4] The information which automatic transcriptions are good and which are bad examples are however used when adapting acoustic and language model.

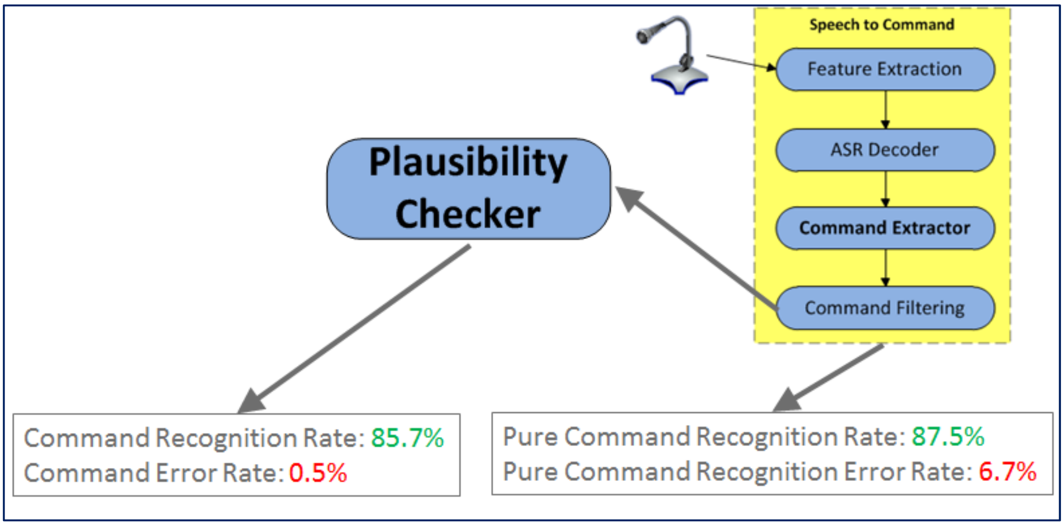

Figure 12: Rates without using situational context information for command recognition. Situational context information is applied only for plausibility checking of ASR output.

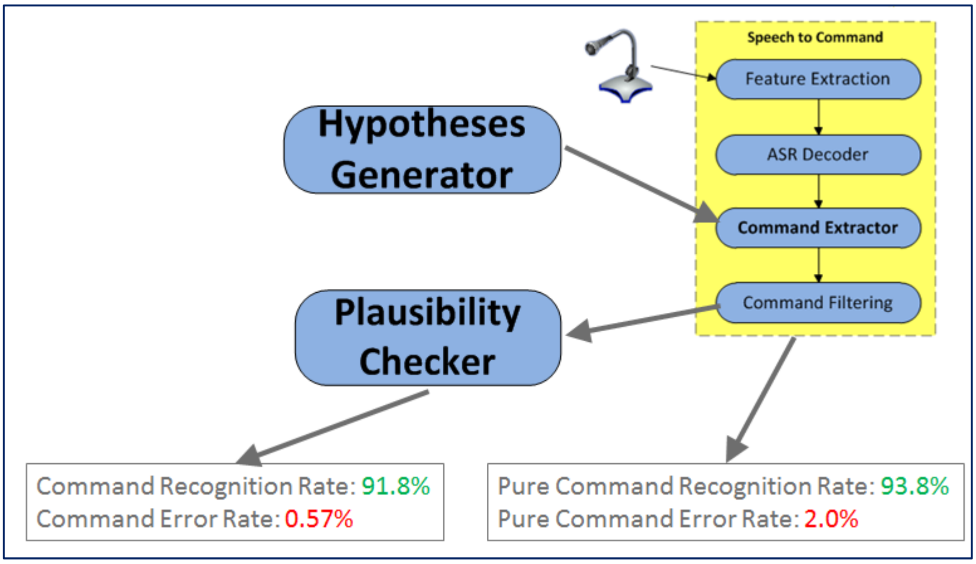

Figure 13: Rates with using situational context information for command recognition and also for plausibility checking of ASR output.

Table 18 shows the same results for Vienna.

	Command RecognRate	Command Error Rate	Pure Command Recogn	Pure Command Error Rate
no context	70.01%	1.80%	70.27%	1.47%
static context	83.29%	3.72%	83.40%	3.42%
dynamic context	83.77%	4.50%	84.02%	4.18%

Table 18: Results when using no context, static or dynamic context for Vienna evaluation data based 100% of the untranscribed learning data.

The results in Table 17 and Table 18 clearly show that the usage of situational context information (the set of possible commands generated by command hypotheses generator, depending on the current situation given in radar data) improves the command recognition rates. This is true for the pure outputs of ASR (columns 4) and also for the output after using the checker (columns 2).

Not so clear are the results when dynamic context is exploited (also type and values of a command) or only static context is exploited (only callsign information). For Prague, all metrics got worse and for Vienna all metrics slightly improved. Further analysis is needed to understand the reasons in more detail.

We can conclude that the amount of output used from the CPM has no effect on the output of the automatic transcription, but the context itself improves recognition rates.

5 Operational Proof-of-Concept O1

5.1 Experimental Setup

The operational part O1 for the proof of concept trials is based on pairwise comparison of the output of two different ABSR systems:

1. ABSR system developed in WP3 trained only with manually transcribed data (denoted to as "basic");
2. ABSR system of WP3 improved with all available both manually and automatically transcribed data (i.e. generated using unsupervised training) (denoted to as "advanced").

Both systems operate on the available recorded voice and radar data sets. Recordings which result in the same command sequences (recognitions) obtained from both systems are ignored in O1. Only recordings which result in different recognitions for both systems are further analysed. N (35 for Vienna and 36 for Prague) recordings with different recognitions are randomly selected[5] (assuming the correct transcription is not necessary). Each voice utterance is presented to the controller i.e. the controller can listen to the recording as often as he/she likes. The output of both ABSR systems is also shown to the controller.

[5] The selection process was random, otherwise we might get a bias in the experiment outcome, i.e. the selections may favour the trained ABSR system.

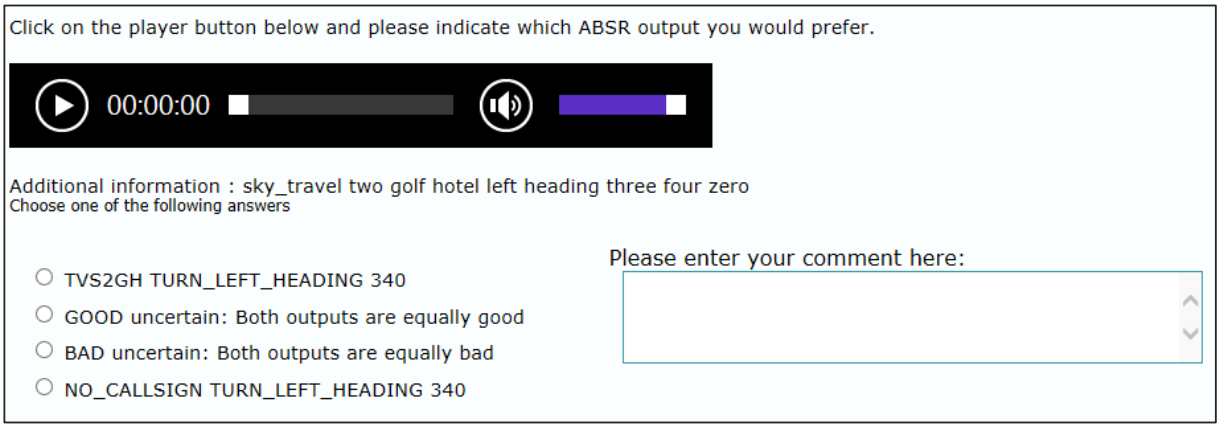

Figure 14 Output of basic and trained ABSR systems.

Figure 14 shows an example of both basic and advanced ABSR systems. The controller can replay a voice recording as often as necessary (pressing the play button). Under the graphically visualised voice recording, the controller can find a word-by-word transcription of what was actually said, as shown in in Figure 14. The controller then always has four choices:

- The output of the trained ABSR system is the better one
- The output of the basic ABSR system is the better one
- Both outputs are equally good
- Both outputs are equally bad

In all selected examples the controller does not know which recognition output is generated by which ABSR system (advanced or basic). As for example in Figure 14, that means that the controller will only see two command recognition outputs TSV2GH TURN_LEFT_HEADING 340 and NO_CALLSIGN TURN_LEFT_HEADING 340.

The basic ABSR system for Prague (baseline - also marked with A in the graphics and tables of next section) is created without using any untranscribed training data for AM and LM. For developing CPM, 10% of untranscribed training data are used.. The results are stored in SVN in iter10.allHyp.* and were created on 12th December 2017. The trained ABSR system for Prague (ML-Improved - also marked with B in the graphics and tables of next section) is created with 100% of untranscribed training data for all models, but learning is not performed incrementally as described in T2. It is the best model available for end of November 2017. The results are stored in SVN in iter09.allHyp.* and created 08-12-2017.[6]

Table 19 shows the command recognition rates etc. obtained for the two different versions of the ABSR system for Prague.

		Rates with Checker used			Pure ASR Rates without Checker		
		Recogn	Error	Reject	Recogn	Error	Reject
Baseline	iter10	77.7%	0.4%	22.6%	84.6%	7.6%	8.5%
ML-Improved	iter09	87.4%	0.7%	12.8%	89.1%	3.5%	8.2%

Table 19: Command Recognition, Command Recognition Error and Command Rejection Rates for Prague Approach for Baseline and Trained ABSR Systems.

Overall, from the basic ABSR to advanced ABSR system (exploiting 100% of available untranscribed training data), we can see an improvement of 9.7% absolute in command recognition rate (87.4% - 77.7%) and of 5.8% absolute in command prediction error rate (8.7% - 2.9% as given in Table 1) and a decrease of 0.3% (Table 19) absolute in command recognition error rate. The improvement on ASR side alone without using checker is 4.5% absolute in command recognition rate (89.1 – 84.6 in Table 19)

401 differences among basic basic and advanced ABSR outputs between iter09 and iter10 were recognized (of course not always iter09 was better). We selected from each controller one session and from this session each time 3 differences in recognition output. The selection process was random. This results in 36 differences in recognition output for the evaluation in the step by step experiment O1. The questions were presented to four male controllers. The order in which the questions were presented was random. The order of the possible answers was also randomly selected, i.e. sometimes recognizer A was given as the first answer, sometimes recognizer B and sometimes it was “both are equally bad” or “both are equally bad”.

The basic ABSR system for Vienna (baseline resp. system A) is developed without using untranscribed training data for training AM and LM, 10% of untranscribed data was used for learning CPM. The results are stored in SVN in iter10.allHyp.* and were created on 13th December 2017. The advanced

[6] If a callsign was not in the set of predicted commands, the result shown in A or B to Prague controllers may nevertheless be this result and not "NO_CALLSIGN NO_CONCEPT". This was a minor bug in the checker component which was not corrected before creating the data of the O1 experiment. The presented results (with respect to error rate) would even be better for both recognizers. This error was already created when Vienna data for SxS was created.

BSR system for Vienna (ML-Improved resp. system B) is developed with 100% of untranscribed training data, but not incrementally. It is the best model available end of November 2017. The results are stored in SVN in iter05.allHyp.* and created 11-12-2017.

Table 20 shows the command recognition rates etc. obtained for the two different versions of the ABSR system for Vienna.

ABSR system		Rates with Checker used			Pure ASR Rates without Checker		
		Recogn	Error	Reject	Recogn	Error	Reject
Basic	iter10	64.9%	1.8%	34.4%	66.0%	14.1%	21.0%
Advanced	iter05	72.1%	2.7%	26.3%	73.2%	7.8%	20.1%

Table 20: Command Recognition, Command Recognition Error and Command Rejection Rates for Vienna Approach for basic and advanced ABSR Systems.

Overall, we have from basic system to an advanced one (using 100% of available untranscribed training data) an increase of 7.2% absolute in recognition rate, but also an increase of 0.9% absolute in command recognition error rate. The improvement on ASR side alone without using checker is 7.2% absolute in command recognition rate and of 6.3% in command recognition error rate.

For each of the 17 controllers for which command transcriptions were available, we selected one session resulting in 522 differences between two ABSR recognition outputs. We selected for each controller two differences (respectively for one controller we selected three differences) so that we have 35 differences in total for the SxS experiment (side by side experiment O1). The questions were presented to four male and one female controller. The order in which the questions were presented was random. The order of the presented answers was also randomly selected, i.e. sometimes recognizer A was the first answer, sometimes recognizer B and sometimes it was “both are equally bad” or “both are equally bad”.

The results for Prague and Vienna are presented in the next section.

5.2 Results

Figure 15 shows a bar chart of the O1 results for Prague and Vienna controllers. Both controllers preferred the results of the trained ABSR system (ML-Improved/advanced ABSR system) against the basic ABSR (baseline/basic) system which has not seen any untranscribed data.

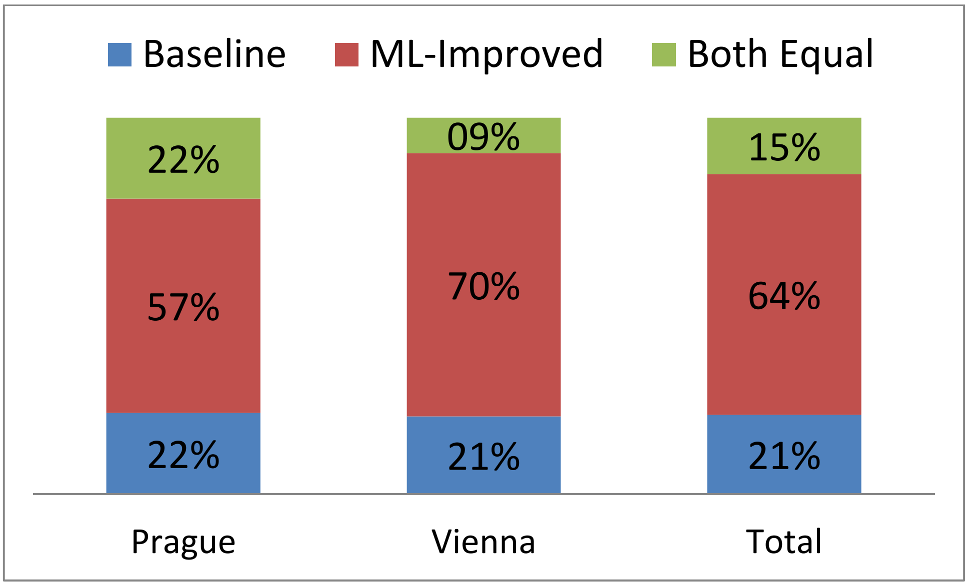

Figure 15 Bar Chart of SxS-Experiment for Prague, Vienna and both together.

Figure 16 and Figure 17 show how often which recognizer was preferred, “A” resp. “B” mean, that recognizer A (i.e. baseline/basic) resp. B (ML-Improved/advanced) were preferred. “U” means “no decision, both are equally bad”, “G” means “no decision, both are equally good”. In Figure 15 “U” and “G” are subsumed under “Both Equal”.

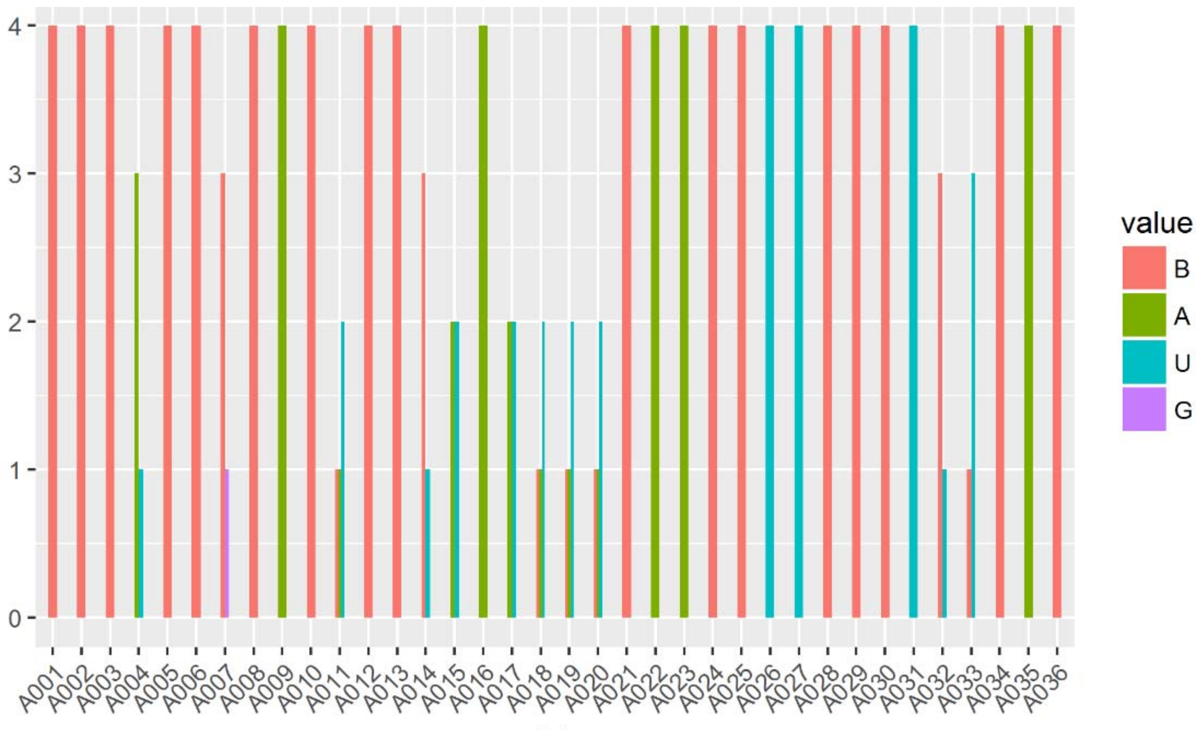

Figure 16 Box Plot, how often, which recognizer was preferred to the other for Prague.

Of special interests are the questions when both recognizer A and recognizer B were selected by different controllers. Questions A018, A019 and A020 are for Prague candidates for a more detailed analysis.

In question A018, the transcribed utterance was “oscar kilo sierra uniform romeo contact ostrava radar one one nine decimal three seven five naslysenou.”

The basic recognizer has recognized:

- NO_CALLSIGN NO_CONCEPT

The advanced one has recognized:

- NO_CALLSIGN HANDOVER OSTRAVA_RADAR
- NO_CALLSIGN HANDOVER_FREQUENCY 119.375

In question A019 the transcribed utterance was “oscar kilo echo yankee echo and contact vodochody tower one three three decimal zero seven five ahoj”

The basic recognizer has recognized:

- NO_CALLSIGN NO_CONCEPT

The advanced one has recognized:

- NO_CALLSIGN HANDOVER VODOCHODY_TOWER
- NO_CALLSIGN HANDOVER_FREQUENCY 133.075

In question A020 the transcribed utterance was “oscar kilo india bravo alpha descend five thousand feet qnh one zero zero nine”

The basic recognizer has recognized:

- NO_CALLSIGN NO_CONCEPT

The advanced one has recognized:

- NO_CALLSIGN DESCEND 5000 ALT
- NO_CALLSIGN QNH 1009

These answers clearly show that the controllers do not like an answer with NO_CALLSIGN. If the callsign is not recognized, even though the rest of the command is recognised correctly, this has no value for the controllers. They have to input in both cases all given commands manually.

Figure 17 shows the differences also for Vienna. Questions A002, A003, A006, A012, A017, A023, A025, A032 and A034 get very different answers.

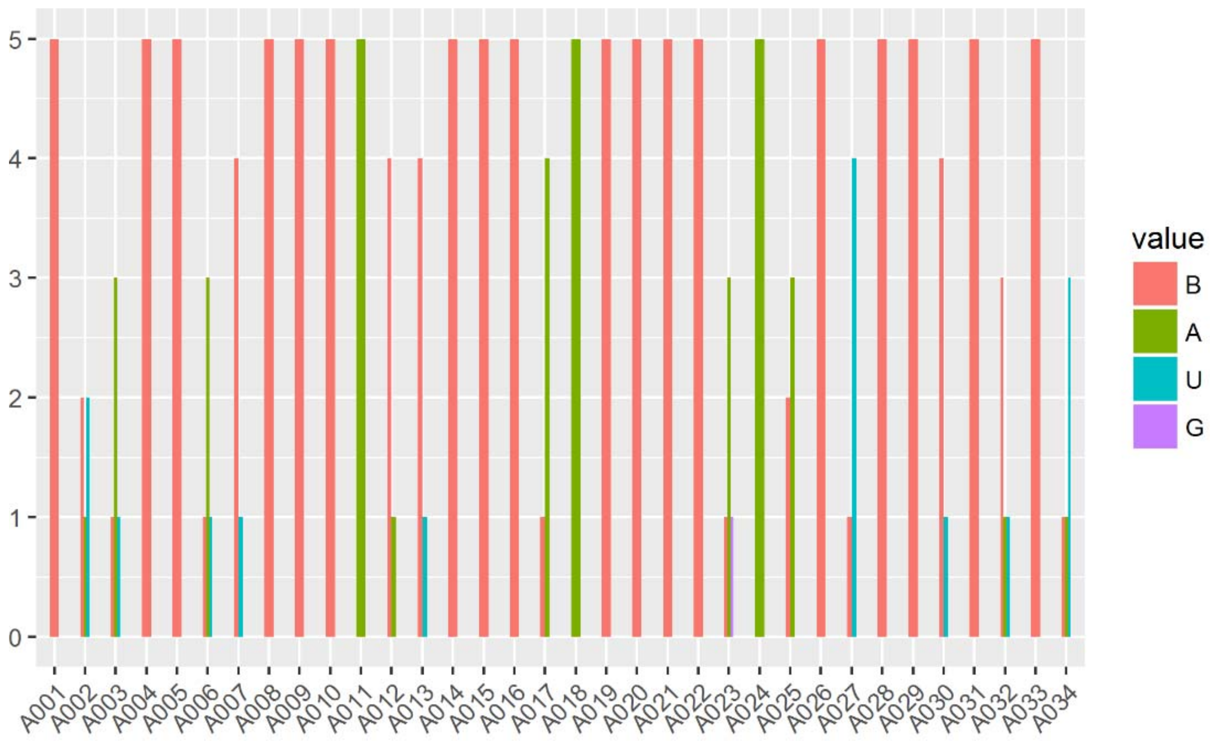

Figure 17 Box Plot, how often which recognizer was preferred to the other for Vienna.

In question A025, the transcribed utterance was “fly_niki five two delta roger contact director one one nine decimal eight callsign only ciao” Additionally we showed to the controllers the text “Only NLY228D is in the air, and a AUA522D. The NLY228D was handed over, i.e. the controller was speaking to NLY228D”.

The basic recognizer has recognized:

- NO_CALLSIGN NO_CONCEPT

The advanced one has recognized:

- AUA522D HANDOVER DIRECTOR
- AUA522D HANDOVER_FREQUENCY 119.8

In this case we see the problem of using context. The recognizer uses additional information to derive what the controller means. The basic recognizer A also has used context, but the training data was not sufficient to recognize any callsign which was in the air.

In question A003 we had the opposite problem. The transcribed utterance was “montenegro five hundred contact budapest one three three two goodbye”. There was only a MGX501 in the air. This information was not shown to the controller.

The basic recognizer has recognized:

- NO_CALLSIGN HANDOVER BUDAPEST_RADAR
- NO_CALLSIGN HANDOVER_FREQUENCY 133.2

The advanced one has recognized:

- MGX501 HANDOVER BUDAPEST_RADAR

- MGX501 HANDOVER_FREQUENCY 133.2

Three controllers preferred the basic ABSR. One prefers B and one was not able to decide. One controller made the comment “callsign wrong”.

In question A006 the transcribed utterance was “air_berlin six one eight echo right three one zero cleared ils three four report established”

- NO_CALLSIGN TURN_RIGHT_HEADING 310
- NO_CALLSIGN CLEARED_ILS 34
- NO_CALLSIGN REPORT_ESTABLISHED

The advanced one has recognized:

- NO_CALLSIGN NO_CONCEPT

Vienna controllers prefer at least some information although when the callsign information is missing.

6 Operational Proof-of-Concept O2

6.1 Experimental Setup

The operational part O2 for the proof of concept trials is based on monitoring the output of the ABSR system with real radar and audio recordings (from the ops-room).

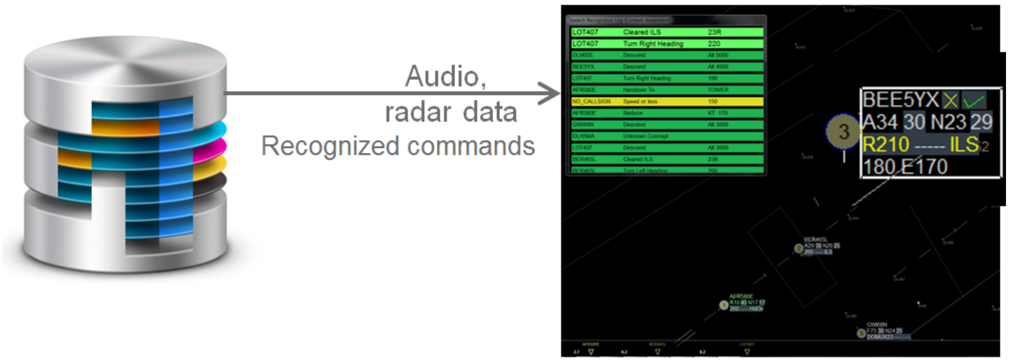

Figure 18: Basic Setup of Proof-of-Concept-Trial with Replay of Radar and Voice Data.

Figure 18 shows the basic idea of the O2 trials. Audio and radar data are recorded in the ops-room and transcribed. After the recording and transcription phase, the audio recordings are presented to the controller together with a replay of the recorded radar data. The audio recordings are additionally used as input for the trained ABSR system and the online recognitions of that system are also presented to the controller. That means, the controller listens to the voice recordings (his/her own or those of a colleague), sees the corresponding radar data and the trained ABSR system attempts to recognize the voice recordings in real time, i.e. no pre-recorded recognition outputs are presented. As shown in Figure 18, the output of the ABSR system is displayed in the radar label in yellow colour (BEE5YX REDUCE 210, and BEE5YX CLEARED_ILS 23L in the example in Figure 18). The speech log (in the upper left corner in Figure 18) is not shown to the controller by default, but can be shown on request.

If the speech recognizer fails (false recognition or recognizes nothing) the controller has to correct or add the command sequence manually. This may include a rejection of the false recognition. The controller has to use the mouse and keyboard for corrections. The scenario lasts approx. 30 minutes including a 10 minute training phase. Details are provided in the following tables.

The SVN software revision used during the Proof of Concept trials in Prague was 2529, date 2018-01-18. For Vienna it was 2576, date 2018-01-25. The following tables show the configuration for each trial.

Prague Replay	
Position	PEC
Speech Scenario	MALORCA/SpeechData/Prague/SEGMENTED/LKPR01/2016-08-13__PEC_1550-1700
Radarfile	20160813_1553_20160813_1700.rdv

Table 21: Configuration Prague Replay

Prague Live	
Position	PEC
Speech Scenario	MALORCA/SpeechData/Prague/SEGMENTED/LKPR01/2016-08-13__PEC_1550-1700
Radarfile	20160813_1553_20160813_1700.rdv

Table 22: Configuration Prague Live

Vienna Replay	
Position	Feeder
Speech Scenario	/MALORCA/SpeechData/Vienna/SEGMENTED/LOWW12/Feeder34_20160709_C2
Radarfile	20160709_0838_20160709_1000.rdv

Table 23: Configuration Vienna Replay

Vienna Live	
Position	Balad
Speech Scenario	MALORCA/SpeechData/Vienna/SEGMENTED/LOWW18/BALAD_20160701_B5
Radarfile	20160701_0903_20160701_1000.rdv

Table 24: Configuration Vienna Live

6.2 Results

Table 25 and Table 26 show the results of O2 for Prague and Vienna controller.

Proof of Concept O2 (Replay)				
Prague:				
Controller	Number of Commands	Number of ABSR Errors/Rejections	corrected by controller[7]	detected by controlle
C1	99	9	9	9
C2	99	9	6	9
C3	99	9	7	9
C4	99	9	9	9
All	396	36	31	36

Table 25: Controller commands Prague

Proof of Concept O2 (Replay)				
Vienna:				
Controller	Number of Commands	Number of ABSR Errors/Rejections	corrected by controller[8]	detected by controlle
C1	122	16	16	16
C2	122	16	16	16
C3	122	16	16	16
C4	122	16	15	16
C5	122	16	16	16
All	610	80	79	80

Table 26: Controller commands Vienna

[7] "Corrected" means that the controller has corrected the wrong recognition command within the replay. Not corrected only happened because of the problems of the implemented HMI. Every error was recognized by the controllers and reported to DLR stuff with remark that correction is not done due to problems with HMI. It was not distinguished if this was possible for the controller resp. if the implementation of HMI does not support the correction.

[8] See footnote 7at page 27

7 Feedback from controllers

7.1 Feedback via Questionnaires

After the replay, the life demo and the O1 side by side (SxS) experiment a questionnaire was presented to the controllers. The controllers were asked to answer the following questions by assigning a digital value between 1 and 6 to each of the questions. 1 means “totally disagree” whereas 6 means “totally agree”. Table 27 summarizes the mean values and standard deviations to the answers for Prague (4 controllers), Vienna (5 controllers) and for all controllers.

	Prague		Vienna		Prague/Vienna	
	Avg	Sigma	Avg	Sigma	Avg	Sigma
The explanation within the briefing was helpful.	5.5	1.0	5.8	0.4	5.7	0.7
I could imagine to work with Speech Recognition support for radar label maintenance	5.75	0.5	5	1.0	5.3	0.9
I understood the application of Speech Recognition Support for Radar Label Maintenance	6	0.0	5.4	0.5	5.7	0.5
Today’s support of Speech Recognition was adequate for the presented scenario.	4.8	1.3	5.2	0.4	5.0	0.9
Speech Recognition Support of MALORCA system would be (already) helpful for my workplace.	4.0	1.4	4.4	1.1	4.2	1.2
The application of the MALORCA system would provide an improvement for my work.	4.8	1.0	5.2	0.8	5.0	0.9
The number of command corrections was proper with respect to the scenario (traffic density, complexity,..).	4.3	1.0	5	0.7	4.7	0.9

	Avg Prague	Sigma Prague	Avg Vienna	Sigma Vienna	Avg Prague/Vienna	Sigma Prague/Vienna
It was easy to do a corrective action and I was able to maintain situational awareness.	**5.5**	**1.0**	**5.8**	**0.4**	**5.7**	**0.7**
The performance of the MALORCA system was stable and individual disruptions e.g. wrong recognition did not disturb me from controlling the situations.	**4.75**	**1.0**	**5.2**	**0.4**	**5.0**	**0.7**
ASR will cause safety problems	**2**	**0.8**	**2.5**	**0.6**	**2.3**	**0.7**
Was it possible to follow the ATCo-pilot communication although you had no chance to influence traffic?	**5.75**	**0.5**	**5.6**	**0.5**	**5.7**	**0.5**

Table 27: Results of questionnaire (green with high value for ASR system, yellow values with higher standard deviation).

7.2 Feedback in Debriefing Session

Immediately after the questionnaire in written form, a formal debriefing was performed. Two DLR employees and the controller were involved in the debriefing lasting between 20 and 30 minutes.[9] Not all of the eleven questions were explicitly asked to the controller. The de-briefing always started with the first question "Which working configuration did suit you best and why?" and an introduction what a working configuration is (mouse or mouse with ABSR support). Depending on the answers, the other questions were also presented. However, all controllers provided answers to all of the following eleven questions although they were not always explicitly asked.

1.	Which working configuration did suit you best and why?
	Did you prefer mouse or did you prefer Speech Recognition with necessary correction by mouse. **Answers of Prague Controllers** • ASR + Mouse correction is preferred, but only when recognition rate is high (better than current way of working, that means only mouse)

[9] In the four debriefings performed with Prague controllers one "observer" from ANS CR also participated passively, i.e. just observing and not asking questions.

- ASR works properly
- Mouse is essential at least to correct ASR failures
- On DEP frequency ASR might be preferred (if recognition speed increases) on Director position mouse input is preferred

Answers of Vienna Controllers

- ASR + Mouse correction is preferred, if it works properly.
- Recognition of the simulation was sufficient
- ASR + Mouse imaginable, but recognition has to be better (at least 95%)
- ASR + Mouse is preferred, if ASR can be switched off
- The presented system should be activated not tomorrow, but already today. Currently (in winter period) workload of controllers is lower in certain timeframes, therefore a test in the ops room or at least in shadow mode in the ops room should/could be done
- ASR recognition rate is already acceptable, although an improvement would be very welcome

2. What could be improved to make the Speech Recognition prototype handier in your daily routine?

Forget that you have Dusseldorf waypoints in the menu; forget the bad interface for mouse corrections. How should recognitions be shown in the label? Is there too much information from the speech recognizers in the rardar labels? Not enough recognition information in the radar labels (SQUAWK, QNH, … also necessary)?[10]

Answers of Prague Controllers

- Unexpected / Unusual heading must be recognized better. The same is true for DESCEND for DEP etc.

[10] The additional information (for this question and also for the other questions) was provided to the controller if the interviewer got the impression that the conversation otherwise would stop.

- Break, break, and Disregard should be recognized; CLIMB for ARR and INCREASE for ARR should also be recognized.
- Faster and more accurate, accuracy might be improved by learning, VFR flights starting with OSCAR, KILO in callsign are a problem (callsign spelling in general has to be improved), also recognition for abbreviations of callsigns from VFR has to be improved.
- If NO_CALLSIGN is recognized, the same callsign as in previous utterance should be highlighted, non-standards (e.g. CLIMB for ARR) must be recognized, MAINTAIN_SPEED_OR_GREATER was a problem

Answers of Vienna Controllers

- Recognition output should already be delivered while controller is still speaking
- Process of recognition should be faster
- RATE_OF_DESCEND together with Flight level didn't work properly (however RATE_OF_DESCEND as single command often recognized)
- Some airlines where not recognized (e.g. fraction)
- Recognition of callsigns should be very precise. It is better to not recognize any callsign than recognizing a wrong callsign
- REDUCE 160 until 4 miles final was not correctly recognized (resp. not correctly displayed in HMI, recognition from ASR interface O.K., but then marked by DLR interface as wrong)
- DIRECT LANUX was often misrecognized with DIRECT MOTIX (although it was in trainings data)
- Recognition of callsign fraction (code NJE) never works (also not in offline mode of iter2*.*.cpt), however the word fraction is correctly recognized

3. What would you consider a problem with the implementation of the prototype?

You get tomorrow the ABSR integrated into your working environment and you have the 90% recognition rate with 2% of error rate and approx. 8% of rejection.

Answers of Prague Controllers

- 99% of recognition rate is enough, 10% of error rate are too high
- 95% of recognition rate might be enough, but it must be faster
- A system with 90%/1% is preferred to 95%/5%
- Recognition time, recognition is needed immediately after push to talk button is released

Answers of Prague Controllers

- ASR with 90% Recognition Rate and 2% error rate is preferred over ASR with 95% and 4% (4 of 5 controllers)
- 90% recognition rate is sufficient
- It is better to have no recognition than having a false recognition
- 4 controllers prefer 90%/2%, one controller prefers 95%/4%, an argument for 95/4 was that each not recognized command creates new workload resulting in unsafe situations. Argument for 90/2, an error always creates more workload than a rejection, if ASR fails (with error or rejection) does not make a difference, controller will recognize it anyway or with support of mode-S input

4. What general problems of the ASR system you observe? e.g. general problems with OK* callsigns (five letter callsigns), DESCEND always a problems …

See question 2

Answers of Vienna Controllers

- Recognition of abbreviated callsigns was sometimes a problem, e.g. OKL instead of OEAKL

5. What would be the main reasons for you to NOT use the speech recognition system in your work?

Answers of Prague Controllers

- Recognition rate could be a show stopper (no clear answer if recognition error rate or recognition rate itself is the problem)
- Recognition Rate not sufficient
- Low response times
- Recognition speed

Answers of Vienna Controllers

- Recognition to slow
- Too much false recognitions or no recognitions (requires corrections)
- If Speeds, Headings, flight levels and waypoints are not reliable recognized
- Levels have to be recognized very reliably
- Important are level, heading, speed and waypoint commands
- If workload of controller is high (high traffic), an even faster ASR is needed, than controller has not the time to wait for ASR output. This requirement

mentioned by several controllers.
- Misrecognition Rate of callsigns too high

6. The ASR system could lead to more safety critical situations. How do you comment that statement and how to avoid?
The system has a negative influence on my mental picture of the situation?

Answers of Prague Controllers

- If ASR performance is bad, the controller will ignore if (so a switch off button must be available)
- Safety is not an issue with respect to ASR and "ASR or mouse", no difference with respect to mental picture predicted
- No situation awareness loss assumed
- For safety a checking with respect to transponder / mode-S aircraft output is an option → so also no safety problems with ASR are expected
- ASR will have no effect on mental picture, if error rate is below 1% safety is not an issue, not danger that ATCo trust system too much

Answers of Vienna Controllers

- Not safety critical (additional check is provide through Mode-S), but for handovers to other positions the controller must have the possibility to undo a wrong handover. Otherwise a telephone call to handover position would be necessary to undo handover. Maybe HANDOVER command is the only command type for which not an automatic acceptance (after predefined time of no rejection) is allowed.
- No Influence on situational awareness
- Could become safety critical if controller starts trusting the system to much
- No influence on mental picture, because controllers are trained to detect errors
- Situational awareness with ACG operation HMI TOPSKY better than with prototype RadarVision adapted for O2 proof-of-concept trials
- Error Rate of ASR is not the problem with respect to safety, today we also have 10% of controller readback errors resp. say again request (HHe: number seems quite high)
- Each deviation from usual working method may result in a safety critical situation. Therefore, a higher recognition is preferred and the resulting higher error rate is accepted.

7. Did you enjoy the trials and if not why?

Answers of Prague Controllers

- Yes, I like to test new “things”.
- Fine
- Simulation was enjoyed

Answers of Vienna Controllers

- Simulation was enjoyed
- Simulation was interesting
- Very much
- Taking part in trials was funny

8. What would you propose to make the simulation more realistic? We of course have the constraints not to go into the ops room?

Is it better to have the speech recognizer in the simulator, but how to evaluate recognition performance on life data?

Answers of Prague Controllers

- Testing in peak hours with heavy traffic
- Adding speed vectors to HMI, readbacks are missing
- Simulation with pseudo pilots
- Higher traffic density, maybe not in the beginning of the scenario

Answers of Vienna Controllers

- Using TOPSKY HMI in a simulation environment
- Simulation with pseudo pilots
- Faster recognition
- Switch off and switch on of ASR functionality

9. Did you miss something in the simulation?

Answers of Prague Controllers

- Pilot readbacks
- Pan-pan and medical aircraft

Answers of Vienna Controllers

- Pilot readbacks

10. Under which circumstances would you like to test the ASR prototype? (simulator, real life traffic, tower environment, transcription of controller-pilot communication …)

Answers of Prague Controllers

- Heavy traffic
- In a tower environment ASR maybe not needed so much, since not so many commands have to be in the system to work properly
- Testing in simulator, Approach area preferred, runway incursion prevention could be an application in tower environment,
- On DEP position (DEP normally get not heading information, maybe 20% get a heading, DEP more often get DIRECT_TO command)

Answers of Vienna Controllers

- Using BALAD sector instead of Feeder
- System could be tested live in real working environment, if it is possible to switch it off
- In the ops room

11. Do you have any additional comments?

Answers of Vienna Controllers

- Callsign recognition could maybe also supported by moving the mouse to the actual position, according label or just using an eye tracker which highlights the recognized callsigns after a recognition is rejected (just another option to avoid inputting the whole command again, if only callsign recognition has failed.
- Simulator of ACG works with and without pseudo-pilots; the aircraft are guided by mouse inputs, if pseudo-pilots are not available. Therefore a simulation with ASR, but without pseudo-pilots, would be possible in Vienna.

8 Major Gaps

This chapter focuses on major gaps, i.e. significant issues that have not been addressed in MALORCA. At the time of writing, these are the four major gaps:

- Increase the complexity of the airspace
- Increase portability to other mid-sized airports
- Increase the amount of untranscribed data used
- Work on pilot speech in particular pilot read back

These gaps are now detailed in the following sections.

8.1 Increase the complexity of the airspace

In MALORCA we considered Vienna and Prague approach and departure. Both are mid-sized airports with moderate though increasing traffic. Therefore the complexity for all MALORCA modules is relatively manageable. This picture changes completely when we e.g. go to London. London has five major airports and a couple of smaller ones (see **Figure 19:**). These airports, however, cannot be operated separately, because London airspace is taken as one entity. All London airports together have more than 1 million movements per year (for comparison: Vienna has about 250 thousand and Prague around 140 thousand movement per year). The size as such already increases the processing effort and the risk of miss recognitions. Not only the total number of call signs in the context way is bigger, but also the number of controllers is much larger, making it necessary to cope with a much larger variety in speaking style accent and ways to handle the phraseology.

Figure 19: Illustration of the London airspace (Source: Wikipedia).

8.2 Portability to other mid sizes airports.

While London might have the airspace with the highest complexity, there is a different route to maximise impact.of the MALORCA findings In Figure 20, you see the number of movements per year for big and mid-sized airports, based on ICAO runway safety list (https://www.icao.int). There are about 200 airports with between 50,000 and 250,000 movements per year. They cover all parts of the planet. In total these 200 airports have 23 million movements per year. If all these airports could be equipped with a MALORCA system the impact would be even larger than for London area only.

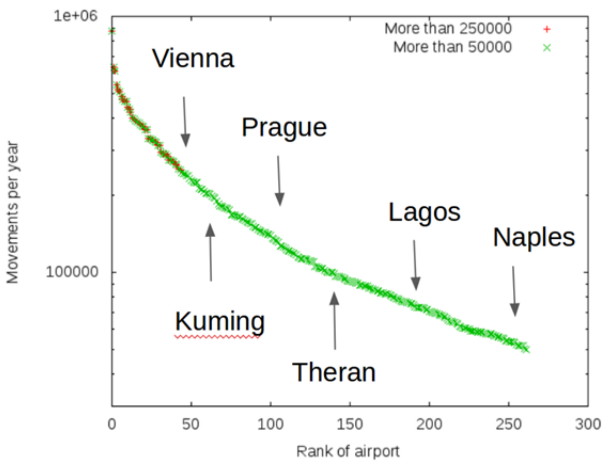

Figure 20 Airports sorted by number of movements per year. Selected airports are highlighted showing the diversity.

In order to achieve this, different gaps need to be addressed. Most importantly attention to more diversity is important. Not only different accents have to be handled, but we expect also different ways of handling the phraseology deviations and more importantly handling elements of the utterances that are outside the phraseology (e.g. like “dobry den” and other types of greetings).

8.3 Increase the amount of untranscribed data

In MALORCA we have seen the clear benefit of using additional untranscribed data. However, only up to 20 hours for Prague and 25 hours for Vienna have been used. Prague alone produces 4,000 hours of speech (net!) per year under exactly the same conditions that the MALORCA project considered, the total amount of speech (not only approach) is way more than 10,000 hours per year. Even with the existing architecture, further improvements in accuracy can be expected. For instance, the basic system has been pre-developed with relatively limited amount of data. This means the size of applied models such as AM, LM or CPM needs to be also limited in terms of number of parameters. Larger models could not be trained as we would start dealing with typical machine learning problems such as model overfitting. We can presume that in case of having much larger datasets available for training the initial version of ABSR, models with much larger amount of parameters could be trained. Large datasets would also allow for an end-to-end architecture (i.e. the speech utterance (as wave file) is the input and the abstract representation of the semantics is the direct output without intermediate output (e.g. phonemes and words)). Based on experience with end-to-end systems for dictation we would expect an additional improvement.

8.4 Recognizing pilot read back

A final major gap is the capability of being able to recognize pilot read back. This is important in general for example to be able to analyse a complete controller-pilot conversation, but also to be able whether a command was transmitted from the controller to the pilot successfully (e.g. read back and/or hear back error detection).

In order to address this, a couple of technical challenges have to be solved. The first one is again variability in English accents of pilots. Because pilots can come from any country, a huge variability in accents is to be expected. A second one is background noise from the cockpit and final one is transmission distortions from the radio channel. A non-technical issue would be ethical legal issues, as pilots cannot provide their agreement or disagreement that their speech is processed by an automatic system. This might be an issue in some or many countries.

9 Conclusions

MALORCA's Proof-of-Concept is split into two technical and two operational activities:

- During the first technical activity (T1) a workshop with technical experts was conducted in Prague. The goal was to demonstrate how the actual prototype configuration fulfils predefined requirements and to verify that the prototype is generally compliant with generic operational requirements in D1-2. The objective of MALORCA, however, was not to fulfil all or even some of the requirements. The requirements were defined by the two participating ASNSPs with respect to future operational implementation of Assistant Based Speech Recognition in the ops room. Nevertheless most of the requirements especially with respect to command recognition rate and command recognition error rate were acceptable. Some command types were not recognized, but the reason was mostly related to missing training data. A challenge not addressed in MALORCA is the recognition time.
- The first operational part O1 consists of a pairwise comparison of the output of two different ABSR systems (1) ABSR system developed in WP3 trained only with manually transcribed data (denoted to as "basic") and (2) ABSR system of WP3 improved with all available transcribed data (denoted to as "advanced"). In two third of the cases the controllers preferred the advanced system.
- The second operational part O2 puts the trained ABSR system in a simulation environment with a replay of historic radar data and controller real voice recordings from Prague and Vienna. ABSR is used here to support the controllers in maintaining radar labels. More than 1,000 commands were presented to nine different controllers from Prague and Vienna. In one case the system had a command recognition rate of 91% and in the other case of 87%. In all the cases the ABSR system helps the controllers to maintain the radar labels. Each time the ABSR system fails in recognizing the correct command the controller was able to detect the difference. The ABSR system, therefore, arises no new safety issues.
- In the second technical part (T2) we simulated the process of continuously learning on monthly bases from untranscribed data. First we use the untranscribed data from August 2016 and learn from them. This results in an ABSR System called ABSR August. We use this system and evaluate with all the testing data and get our metrics of August. Then we add the untranscribed data from September and retrain our models on the data of August and September. We get a system ABSR September and determine the rates. Further we continue with the data from October and November. The hypothesis that unsupervised learning will improve command recognition rates was successfully tested for Vienna and Prague. Command recognition rates improved from 80% to 92% for Prague respectively from 60% to 85% for Vienna.

MALORCA started the idea that radar data as an additional sensor will improve unsupervised learning of speech recognition models. This was successfully evaluated in laboratory tests and with technical experts as well as with controllers from Austro Control and ANS CR. Even 8 kHz voice recordings are not a show stopper, even though development effort significantly increased. However, noisy signal quality or bad initial command transcription as well as different interpretations of the semantics of a given controller command significantly decreases the ABSR system performance. This is clearly demonstrated by the different achieved command recognitions error rates which were below 0.6% for Prague and restricted to 3.2% for Vienna.

In MALORCA, we have shown that even with small amounts of transcription data (in combination with large amounts of out-of-domain data) we can achieve close-to 92% of command recognition rate. In terms of human effort, developed machine learning algorithms have significantly brought down the transcription effort. Nevertheless manual effort for pre-processing the radar data is still needed which should be reduced if learning directly in the ops-room or from thousands of hours is intended. This result together with the easy adaptable basic ABSR system for approach control will be the key to developing and deploying ABSR to different approach areas.

Appendix A Material for the POC

A.1 Checklist of generic requirements for T1

Identifier	Title	T1-Live (YES/NO)	Reason
GEN-FUN-001	Area of interest	YES	
GEN-FUN-002	Sector dependent setting of SYSTEM operational status	NO	This requirement is generic and relevant for later stage of ASR maturity. It has SHOULD priority.
GEN-FUN-003	Start recognition immediately	NO	This requirement is generic and is related to the operational use. It was not planned for testing, SHOULD priority. Current implementation in MALORCA is that recognition starts at release of PTT button.
GEN-FUN-004	Provide callsign information immediately	NO	This requirement is generic and is related to the operational use. It was not planned for testing, SHOULD priority. Current implementation in MALORCA is that recognition starts at release of PTT button.
GEN-FUN-005	Provide complete command information when utterance is completed	YES	
GEN-FUN-006	Recognition of callsign	NO	First and third part of requirement was possible to test. The callsign was recognized if present but not sent immediately; see GEN-FUN-004. The requirement is rather related to advanced logic that is to be applied to voice recognition system as black box. However MALORCA focus on voice

			recognition engine only.
GEN-FUN-007	Linking of commands to callsign	YES	
GEN-FUN-008	Output of recognition from THE SYSTEM	NO	This requirement is needed when integrating with the end system (ATC system). There is no integration btw. ASR and ATC system within the scope of MALORCA project.
GEN-FUN-009	Mode of operation	NO	SHOULD priority + the same reason as for GEN-FUN-008.
GEN-LOG-001	External Data Flows Logging	NO	This requirement is generic and relevant for later stage of maturity. It has WILL priority.
GEN-LOG-002	Internal Activity Logging	NO	This requirement is generic and relevant for later stage of maturity. It has WILL priority.
GEN-LOG-003	Archive period	NO	This requirement is generic and is related to the operational use. Not relevant for the MALORCA project.
REC-FUN-001	Recognition of commands for lateral movement	YES	
REC-FUN-002	Recognition of commands for vertical movement	YES	
REC-FUN-003	Recognition of commands for rate of climb/descent	NO	Not modelled for Prague site, SHOULD priority. Not implemented in MALORCA project due to lack of transcribed training data.
REC-FUN-004	Recognition of commands for speed adjustment	YES	
REC-FUN-005	Recognition of commands for STAR	NO	Should priority. Not implemented in MALORCA project

			due to lack of transcribed training data.
REC-FUN-007	Recognition of commands for approach clearance	YES	
REC-FUN-008	Recognition of commands for handover process	YES	
REC-FUN-009	Recognition of commands for published holding	NO	Should priority. Not implemented in MALORCA project due to lack of transcribed training data.
REC-FUN-012	Recognition of information for (future) landing RWY assignment	YES	
REC-FUN-013	Recognition of commands for go around	YES	Should priority.
REC-FUN-014	Recognition of command to indicate the separation between messages transmitted to different aircraft in a very busy environment	NO	Should priority, not implemented in MALORCA project due to lack of transcribed training data. Remarks: During analysing the recorded data, it was recognized that the ATCo usually release PTT button in the middle of utterance. If so, the system will recognize the commands correctly.
REC-FUN-015	Recognition of command to indicate that an error has been made in transmission and to correct this error	YES	SHOULD priority.
REC-FUN-016	Recognition of command to indicate the transmission shall be ignored	NO	This requirement is generic and was not planned for testing within MALORCA project, SHOULD priority.
REC-FUN-017	Offline configuration of command types properties	NO	This requirement is generic and was not planned for testing within MALORCA project, WILL priority.
SYS-INP-001	Aircraft State: Processing of Asterix CAT62	YES	
SYS-INP-002	Dynamic Airport Data	NO	This is not implemented during MALORCA project. Only QNH

			information is extracted from the recognized commands.
SYS-INP-003	Flight plan data, flight data and their updates	YES	
SYS-OFF-001	Offline configuration support tool	NO	This requirement is generic and was not planned for testing within MALORCA project, WILL priority.
SYS-OFF-002	Waypoint List	YES (remotely)	It was done by DLR, location: Braunschweig, January 2018, see D5-2 in appendix.
SYS-OFF-003	Control Region Boundary (Unit-Test)	YES (remotely)	It was done by DLR, location: Braunschweig, January 2018, see D5-2 in appendix.
SYS-OFF-004	Runway-Configuration (Unit-Test)	YES (remotely)	It was done by DLR, location: Braunschweig, January 2018, see D5-2 in appendix.
SYS-OFF-005	Controller-Working-Position-Configuration (Unit-Test)	YES (remotely)	It was done by DLR, location: Braunschweig, January 2018, see D5-2 in appendix.
SYS-OFF-006	Recording configuration is changeable (Unit-Test)	YES (remotely)	It was done by DLR, location: Braunschweig, January 2018, see D5-2 in appendix.
SYS-OFF-007	System offline configuration options (Unit-Test)	YES (remotely)	It was done by DLR, location: Braunschweig, January 2018, see D5-2 in appendix.
SYS-OFF-008	System offline configuration options	NO	This requirement is generic and was not planned for testing within MALORCA project, WILL priority
SYS-ON-001	System online configuration options	NO	This requirement is generic and was not planned for testing within MALORCA project, WILL priority
SYS-LNR-001	Unsupervised learning improves static context	NO	Tested in T2.

SYS-LNR-002	Unsupervised learning improves dynamic context	NO	Tested in T2.
SYS-LNR-003	Unsupervised learning improves acoustic model	NO	Tested in T2.
SYS-LNR-004	Unsupervised learning improves language model	NO	Tested in T2.
SYS-LNR-005	Unsupervised learning improves concept generator	NO	Tested in T2.
SYS-LNR-005	Unsupervised learning improves context integrator	NO	Tested in T2.
PER-REC-001	Recognition Rate Requirement is still in conflict with PE-REC-001 and PER-REC-003.	NO	Tested in T2 (different metrics). It is generic requirement, important for operational use. It is out of scope of MALORCA project to measure with this metrics.
PER-REC-002	Error Rate Requirement is still in conflict with PE-REC-002 and PER-REC-003.	NO	Tested in T2 (different metrics). It is generic requirement, important for operational use. It is out of scope of MALORCA project to measure with this metrics.
PER-REC-003	Rejection Rate Requirement is still in conflict with PE-REC-001 and PER-REC-002.	NO	Tested in T2 (different metrics). It is generic requirement, important for operational use. It is out of scope of MALORCA project to measure with this metrics.
PER-REC-004	Multiple Commands	YES	
PER-REC-005	Reaction Time	YES	
SYS-SAF-001	Safety and Security Requirements	NO	This requirement is generic and was not planned for testing within MALORCA project, WILL priority
EXT-IN-001	Voice Communication Interfaces	NO	No access to live system, for recorded data it is fulfilled already.

EXT-IN-002	Surveillance Data Interfaces	NO	No access to live system
EXT-IN-003	Flight Status information	NO	No access to live system
EXT-IN-004	Flight Status Information	NO	No access to live system
EXT-IN-005	Flight Status Information	NO	No access to live system
EXT-IN-006	Time Synchronization	NO	No access to live system
EXT-OUT-001	Output Interface – network layer.	NO	No access to HMI of live system
EXT-OUT-002	Output interface – application layer	NO	No access to HMI of live system
EXT-OUT-003	Output interface – relation layer	NO	No access to HMI of live system
EXT-OUT-004	Output interface– timestamp	NO	No access to HMI of live system

A.2 Checklist of Technical Requirements for T1

Identifier	Title	T1-Subject (yes/no)	Reason
MLS-DAT–001	Transcribed data requirement (fulfilled if definition of data with silence is used)	NO	Was already checked in D2-1, D2-3
MLS-DAT-002	Un-transcribed data requirement (fulfilled if definition of data with silence is used)	NO	Was already checked in D2-2, D2-4
MLS-DAT-003	Recording configuration length	YES	
MLS- RAD-004	Matching of voice and corresponding radar data	NO	Validated in D1-3, both data have time-stamp and computers are synchronized.
MLS-AUD–001	Audio Data duration per controller	YES	
MLS-AUD- 002	Audio Data run number per controller	YES	
MLS-AUD-003	Audio Data controller number	YES	
MLS-AUD-004	Audio Data controller gender	YES	
MLS-AUD-005	Audio Data controller age	YES	
MLS-AUD-006	Audio Data controller accents	YES	
MLS-AUD-007	Noise of Audio Data controller accents	YES	
MLS-AUD-008	Sampling rate 8 kHz	YES	
MLS-AUD-009	Audio data in wave format	YES	
MLS-AUD-010	Empty or unusable transmissions are excluded	YES	
MLS-AUD-011	Only controller utterances with contents	YES	
MLS-AUD-012	Representative deviation from ICAO standard phraseology	YES	
HYP-GEN-001	Aircraft allocated context (O2, T1-Live)	NO	Validated already in D3-6
HYP-GEN-002	Outbound Responsibility (O2)	NO	Validated already in D3-6
HYP-GEN-003	Overflight Responsibility (O2)	NO	Validated already in D3-6

HYP-GEN-004	Limited Responsibility Landing RWY 34 Vienna (O2)	NO	Validated already in D3-6 and D2-1, D2-2
HYP-GEN-005	Validated already in D3-6 (O2)	NO	Validated already in D3-6 and D2-3, D2-4
HYP-GEN-006	Limited Responsibility	NO	Validated already in D3-6
Validation Audio Data– 001	Unknown validation audio data	NO	Tested in O2 and T2. PRG: O2 we have 4 controllers (requirement is at least 3) which were not in training data set. For T2 controllers are splitted into training (Prague 9) and test data (3 controllers for Prague) VIE: T2 in Vienna have 5 controllers in test data.
Validation Configuration–001	Recognition configuration validation	NO	See json and osm files (DLR side) for creating CPM and dict files and related rules on Idiap/USAAR side for AM/LM

Appendix B List of recognized and transcribed commands for O2 – Prague

The following table shows the list of commands for the operational test O2 in Prague. The left colum contains a manual transcription of the commands (the gold commands) that should be recognized by the ABSR system. The right column contains the real recognitions of the ABSR system. If the real recognition deviates from the manual transcription the line is marked in red. Sometimes not all commands are recognized. Then the "Recognition" column is empty. Sometimes the output of the ASR system contains more command than really said (e.g. two command in transcription, but three commands recognized). Then one "Transcription" column is empty.

Transcription	Recognition
TVS2EF HEADING 335	TVS2EF HEADING 335
CSA73B NO_CONCEPT	CSA73B NO_CONCEPT
TVS4BD TURN_RIGHT_HEADING 100	TVS4BD TURN_RIGHT_HEADING 100
TVS4BD SPEED 210	TVS4BD SPEED 210
TVS4BD TURN_RIGHT_HEADING 150	TVS4BD TURN_RIGHT_HEADING 150
TVS4BD TURN_RIGHT_HEADING 210	TVS4BD TURN_RIGHT_HEADING 210
TVS4BD CLEARED_ILS 24	TVS4BD CLEARED_ILS 24
TVS2EF SPEED_OR_BELOW 220	TVS2EF SPEED_OR_BELOW 220
TVS2EF DESCEND 4000 ALT	TVS2EF DESCEND 4000 ALT
TVS4BD SPEED 180	TVS4BD SPEED 180
TVS2EF TURN_LEFT_HEADING 270	TVS2EF TURN_LEFT_HEADING 270
TVS2EF CLEARED_ILS 24	TVS2EF CLEARED_ILS 24
TVS2EF SPEED 180	TVS2EF SPEED 180
TVS58J INIT_RESPONSE	TVS58J INIT_RESPONSE
TVS58J MAINTAIN_HEADING	TVS58J MAINTAIN_HEADING
TVS58J DESCEND 4000 ALT	TVS58J DESCEND 4000 ALT
TVS58J QNH 1022	TVS58J QNH 1022
TVS4BD SPEED 160	TVS4BD NO_CONCEPT
NO_CALLSIGN HANDOVER TOWER	NO_CALLSIGN HANDOVER TOWER
NO_CALLSIGN HANDOVER_FREQUENCY 118.1	NO_CALLSIGN HANDOVER_FREQUENCY 118.1
TVS2EF SPEED 160	
TVS2EF HANDOVER TOWER	TVS2EF HANDOVER TOWER
TVS2EF HANDOVER_FREQUENCY 118.1	TVS2EF HANDOVER_FREQUENCY 118.1
TVS58J SPEED_OR_BELOW 200	TVS58J SPEED_OR_BELOW 200
TVS58J TURN_LEFT_HEADING 330	TVS58J TURN_LEFT_HEADING 330
TVS58J TURN_LEFT_HEADING 280	TVS58J TURN_LEFT_HEADING 280
TVS58J CLEARED_ILS 24	TVS58J CLEARED_ILS 24
TVS58J REPORT_ESTABLISHED	TVS58J REPORT_ESTABLISHED
TVS4DF INIT_RESPONSE	TVS4DF INIT_RESPONSE

Transcription	Recognition
TVS4DF DESCEND 5000 ALT	TVS4DF DESCEND 5000 ALT
TVS4DF QNH 1022	TVS4DF QNH 1022
TVS4DF SPEED_OWN	TVS4DF SPEED_OWN
NO_CALLSIGN NO_CONCEPT	NO_CALLSIGN NO_CONCEPT
	NO_CALLSIGN MAINTAIN_SPEED
NO_CALLSIGN HANDOVER TOWER	NO_CALLSIGN HANDOVER TOWER
NO_CALLSIGN HANDOVER_FREQUENCY 118.1	NO_CALLSIGN HANDOVER_FREQUENCY 118.1
TVS4DF TURN_LEFT_HEADING 330	TVS4DF TURN_LEFT_HEADING 330
	TVS4DF EXPECT_ILS 24
TVS4DF DESCEND 4000 ALT	TVS4DF DESCEND 4000 ALT
TVS6HK INIT_RESPONSE	TVS6HK INIT_RESPONSE
TVS6HK DESCEND 5000 ALT	TVS6HK DESCEND 5000 ALT
TVS6HK QNH 1022	TVS6HK QNH 1022
TVS6HK SPEED_OWN	TVS6HK MAINTAIN_SPEED
TVS4DF TURN_LEFT_HEADING 320	TVS4DF TURN_LEFT_HEADING 320
TVS4DF TURN_LEFT_HEADING 280	TVS4DF TURN_LEFT_HEADING 280
TVS4DF CLEARED_ILS 24	TVS4DF CLEARED_ILS 24
TVS6HK DESCEND 4000 ALT	TVS6HK DESCEND 4000 ALT
TVS6HK CLEARED_ILS 24	TVS6HK CLEARED_ILS 24
TVS4DF HANDOVER TOWER	TVS4DF HANDOVER TOWER
TVS4DF HANDOVER_FREQUENCY 118.1	TVS4DF HANDOVER_FREQUENCY 118.1
TVS6HK HANDOVER TOWER	TVS6HK HANDOVER TOWER
TVS6HK HANDOVER_FREQUENCY 118.1	TVS6HK HANDOVER_FREQUENCY 118.1
DLH1LR INIT_RESPONSE	DLH1LR INIT_RESPONSE
DLH1LR DESCEND 4000 ALT	DLH1LR DESCEND 4000 ALT
DLH1LR QNH 1021	DLH1LR QNH 1021
CSA5CT INIT_RESPONSE	CSA5CT INIT_RESPONSE
CSA5CT MAINTAIN_HEADING	CSA5CT MAINTAIN_HEADING
CSA5CT DESCEND 4000 ALT	CSA5CT DESCEND 4000 ALT
CSA5CT QNH 1021	CSA5CT QNH 1021
DLH1LR TURN_RIGHT_HEADING 150	DLH1LR TURN_RIGHT_HEADING 150
DLH1LR DESCEND 3000 ALT	DLH1LR DESCEND 3000 ALT
CSA5CT TURN_LEFT_HEADING 20	CSA5CT TURN_LEFT_HEADING 20
DLH1LR TURN_RIGHT_HEADING 180	DLH1LR TURN_RIGHT_HEADING 180
CSA5CT TURN_LEFT_HEADING 360	CSA5CT TURN_LEFT_HEADING 360

Transcription	Recognition
DLH1LR TURN_RIGHT_HEADING 210	DLH1LR TURN_RIGHT_HEADING 210
DLH1LR CLEARED_ILS 24	DLH1LR CLEARED_ILS 24
DLH1LR REPORT_ESTABLISHED	DLH1LR REPORT_ESTABLISHED
CSA5CT TURN_LEFT_HEADING 330	CSA5CT TURN_LEFT_HEADING 330
CSA5CT TURN_LEFT_HEADING 280	CSA5CT TURN_LEFT_HEADING 280
CSA5CT CLEARED_ILS 24	CSA5CT CLEARED_ILS 24
NO_CALLSIGN HANDOVER TOWER	NO_CALLSIGN HANDOVER TOWER
NO_CALLSIGN HANDOVER_FREQUENCY 118.1	NO_CALLSIGN HANDOVER_FREQUENCY 118.1
CSA7KG INIT_RESPONSE	CSA7KG INIT_RESPONSE
CSA5CT SPEED 160	CSA5CT SPEED 160
NO_CALLSIGN HANDOVER TOWER	NO_CALLSIGN HANDOVER TOWER
NO_CALLSIGN HANDOVER_FREQUENCY 118.1	NO_CALLSIGN HANDOVER_FREQUENCY 118.1
CSA7KG DIRECT_TO FAF24	
CSA7KG DESCEND 4000 ALT	CSA7KG DESCEND 4000 ALT
CSA7KG QNH 1021	CSA7KG QNH 1021
CSA7KG CLEARED_ILS 24	
BER492D INIT_RESPONSE	BER492D INIT_RESPONSE
BER492D DESCEND 70 FL	BER492D DESCEND 70 FL
EXU202 INIT_RESPONSE	EXU202 INIT_RESPONSE
EXU202 DESCEND 4000 ALT	EXU202 DESCEND 4000 ALT
EXU202 QNH 1021	EXU202 QNH 1021
EXU202 SPEED 250	EXU202 SPEED 250
BER492D DESCEND 4000 ALT	BER492D DESCEND 4000 ALT
BER492D QNH 1021	BER492D QNH 1021
BER492D MAINTAIN_HEADING	BER492D MAINTAIN_HEADING
	BER492D EXPECT_ILS 24
NO_CALLSIGN NO_CONCEPT	NO_CALLSIGN NO_CONCEPT
BER492D REDUCE_OR_BELOW 240	BER492D REDUCE_OR_BELOW 240
CSA7KG HANDOVER TOWER	CSA7KG HANDOVER TOWER
CSA7KG HANDOVER_FREQUENCY 118.1	CSA7KG HANDOVER_FREQUENCY 118.1
BER492D NO_CONCEPT	BER492D NO_CONCEPT
NO_CALLSIGN REDUCE_OR_BELOW 220	NO_CALLSIGN NO_CONCEPT
THY5WP INIT_RESPONSE	THY5WP INIT_RESPONSE
THY5WP DESCEND 5000 ALT	THY5WP DESCEND 5000 ALT
THY5WP QNH 1021	THY5WP QNH 1021

10 List of recognized and transcribed commands for O2 - Vienna

The following table shows the list of commands for the operational test O2 in Vienna. The left column contains the manual transcription of the commands (gold command) that should be recognized by the ABSR system. The right column contains the real recognitions of the ABSR system. If the real recognition deviates from the manual transcription, the line is marked in red. Sometimes not all commands are recognized. Then the "Recognition" column is empty. Sometimes the output of the ASR system contains more command than really said (e.g. two command in transcription, but three commands recognized). Then one "Transcription" column is empty.

Transcription	Recognition
AUA9088 NO_CONCEPT	AUA9088 NO_CONCEPT
NLY869Z SPEED 250	NLY869Z SPEED 250
NO_CALLSIGN NO_CONCEPT	NO_CALLSIGN NO_CONCEPT
AUA9088 REDUCE_NOT_BELOW 250	
AUA9088 DESCEND 3000 ALT	AUA9088 DESCEND 3000 ALT
AUA9088 DESCEND 3000 ALT	AUA9088 ALTITUDE 3000 ALT
AUA9268 NO_CONCEPT	AUA9268 NO_CONCEPT
AUA9088 SPEED_OR_ABOVE 220	AUA9088 SPEED_OR_ABOVE 220
NLY869Z REDUCE 220	NLY869Z REDUCE 220
AUA9268 MAINTAIN_HEADING	AUA9268 MAINTAIN_HEADING
AUA9088 SPEED_OR_ABOVE 180	AUA9088 SPEED_OR_ABOVE 180
AUA9088 CLEARED_ILS 34	AUA9088 CLEARED_ILS 34
NLY869Z DESCEND 3000 ALT	NLY869Z DESCEND 3000 ALT
NLY869Z CLEARED_ILS 34	NLY869Z CLEARED_ILS 34
AUA9268 TURN_RIGHT_HEADING 280	AUA9268 TURN_RIGHT_HEADING 280
AUA9268 DESCEND 3000 ALT	AUA9268 DESCEND 3000 ALT
AUA9268 REDUCE 220	AUA9268 REDUCE 220
AUA9268 TURN_RIGHT_HEADING 310	AUA9268 TURN_RIGHT_HEADING 310
AUA9268 CLEARED_ILS 34	AUA9268 CLEARED_ILS 34

Transcription	Recognition
AUA9088 SPEED 160	AUA9088 SPEED 160
AUA9088 HANDOVER TOWER	AUA9088 HANDOVER TOWER
AUA9088 HANDOVER_FREQUENCY 123.8	AUA9088 HANDOVER_FREQUENCY 123.8
NLY3187 DESCEND 4000 ALT	NLY3187 DESCEND 4000 ALT
NLY869Z SPEED 200	NLY869Z SPEED 200
AFL2184 DESCEND 4000 ALT	AFL2184 DESCEND 4000 ALT
AUA528C INIT_RESPONSE	AUA528C NO_CONCEPT
NLY3187 TURN_LEFT_HEADING 100	NLY3187 TURN_LEFT_HEADING 100
NLY869Z SPEED 160	NLY869Z SPEED 160
NO_CALLSIGN NO_CONCEPT	NO_CALLSIGN NO_CONCEPT
NLY3187 MAINTAIN_HEADING 50	NLY3187 MAINTAIN_HEADING 50
NLY3187 DESCEND 3000 ALT	NLY3187 DESCEND 3000 ALT
AFL2184 DESCEND 3000 ALT	AFL2184 DESCEND 3000 ALT
AFL2184 NO_CONCEPT	AFL2184 NO_CONCEPT
NLY3187 TURN_LEFT_HEADING 360	NLY3187 TURN_LEFT_HEADING 360
NLY3187 CLEARED_ILS 34	NLY3187 CLEARED_ILS 34
NLY869Z HANDOVER TOWER	NLY869Z HANDOVER TOWER
NLY869Z HANDOVER_FREQUENCY 123.8	NLY869Z HANDOVER_FREQUENCY 123.8
AUA528C REDUCE 200	AUA528C REDUCE 200
AUA9268 SPEED 200	AUA9268 SPEED 200
AFL2184 HEADING 180	AFL2184 HEADING 180
AUA528C INTERCEPT_LOCALIZER 34	AUA528C CLEARED_ILS 34
AFL2184 TURN_RIGHT_HEADING 230	AFL2184 TURN_RIGHT_HEADING 230
AUA9268 SPEED 160	AUA9268 SPEED 160
AUA9268 HANDOVER TOWER	AUA9268 HANDOVER TOWER
AUA9268 HANDOVER_FREQUENCY 123.8	AUA9268 HANDOVER_FREQUENCY 123.8
AFL2184 REDUCE 200	AFL2184 REDUCE 200
AFL2184 TURN_RIGHT_HEADING 310	AFL2184 TURN_RIGHT_HEADING 310
AFL2184 CLEARED_ILS 34	AFL2184 CLEARED_ILS 34
NLY3187 REDUCE 200	NLY3187 REDUCE 200
AFL2184 REDUCE 180	AFL2184 REDUCE 180
AUA528C CLEARED_ILS 34	AUA528C CLEARED_ILS 34
AFL2184 REDUCE 160	AFL2184 REDUCE 160
NLY3187 REDUCE 160	NLY3187 REDUCE 160
NLY3187 HANDOVER TOWER	NLY3187 HANDOVER TOWER
NLY3187 HANDOVER_FREQUENCY 123.8	NLY3187 HANDOVER_FREQUENCY 123.8
NO_CALLSIGN NO_CONCEPT	NO_CALLSIGN NO_CONCEPT
AUA528C REDUCE 180	AUA528C REDUCE 180
AFL2184 HANDOVER TOWER	AFL2184 HANDOVER TOWER

Transcription	Recognition
AFL2184 HANDOVER_FREQUENCY 123.8	AFL2184 HANDOVER_FREQUENCY 123.8
AUA528C REDUCE 160	AUA528C REDUCE 160
AUA528C HANDOVER TOWER	AUA528C HANDOVER TOWER
AUA528C HANDOVER_FREQUENCY 123.8	AUA528C HANDOVER_FREQUENCY 123.8
BAW696V DESCEND 3000 ALT	BAW696V DESCEND 3000 ALT
BAW696V TURN_RIGHT_HEADING 250	BAW696V TURN_RIGHT_HEADING 250
BAW696V SPEED_OR_ABOVE 180	BAW696V SPEED_OR_ABOVE 180
	BAW696V EXPECT_ILS 34
NO_CALLSIGN NO_CONCEPT	NO_CALLSIGN NO_CONCEPT
AUA122A DESCEND 4000 ALT	AUA122A DESCEND 4000 ALT
NLY486N MAINTAIN_HEADING	NLY486N MAINTAIN_HEADING
BAW696V TURN_RIGHT_HEADING 310	BAW696V MAINTAIN_HEADING 310
BAW696V CLEARED_ILS 34	BAW696V CLEARED_ILS 34
	NLY486N HEADING 070
AUA122A TURN_RIGHT_HEADING 250	AUA122A TURN_RIGHT_HEADING 250
AUA122A REDUCE 200	AUA122A REDUCE 200
NLY486N TURN_LEFT_HEADING 70	NLY486N TURN_LEFT_HEADING 70
AUA122A TURN_RIGHT_HEADING 310	AUA122A MAINTAIN_HEADING 310
AUA122A DESCEND 3000 ALT	AUA122A DESCEND 3000 ALT
AUA122A CLEARED_ILS 34	AUA122A CLEARED_ILS 34
NLY486N DESCEND 4000 ALT	NLY486N DESCEND 4000 ALT
BAW696V SPEED 160	BAW696V SPEED 160
NLY486N TURN_LEFT_HEADING 30	NLY486N TURN_LEFT_HEADING 30
BAW696V HANDOVER TOWER	BAW696V HANDOVER TOWER
BAW696V HANDOVER_FREQUENCY 123.8	BAW696V HANDOVER_FREQUENCY 123.8
EZY74BV INIT_RESPONSE	EZY74BV NO_CONCEPT
NLY486N DESCEND 3000 ALT	
AUA122A SPEED 160	AUA122A SPEED 160
NLY486N TURN_LEFT_HEADING 10	NLY486N TURN_LEFT_HEADING 10
NLY486N CLEARED_ILS 34	NLY486N CLEARED_ILS 34
EZY74BV TURN_RIGHT_HEADING 250	EZY74BV TURN_RIGHT_HEADING 250
NLY486N SPEED_OR_ABOVE 180	NLY486N NO_CONCEPT
AUA122A HANDOVER TOWER	AUA122A HANDOVER TOWER
AUA122A HANDOVER_FREQUENCY 123.8	AUA122A HANDOVER_FREQUENCY 123.8
EZY74BV TURN_RIGHT_HEADING 310	EZY74BV TURN_RIGHT_HEADING 310

Transcription	Recognition
EZY74BV CLEARED_ILS 34	EZY74BV CLEARED_ILS 34
EZY74BV DESCEND 3000 ALT	EZY74BV DESCEND 3000 ALT
NLY486N SPEED_OWN	NLY486N NO_CONCEPT
NLY486N HANDOVER TOWER	NLY486N HANDOVER TOWER
NLY486N HANDOVER_FREQUENCY 123.8	NLY486N HANDOVER_FREQUENCY 123.8
EZY74BV HANDOVER TOWER	EZY74BV HANDOVER TOWER
EZY74BV HANDOVER_FREQUENCY 123.8	EZY74BV HANDOVER_FREQUENCY 123.8
AUA176 DESCEND 3000 ALT	AUA176 DESCEND 3000 ALT
AUA176 TURN_RIGHT_HEADING 250	AUA176 TURN_RIGHT_HEADING 250
	AUA176 EXPECT_ILS 34
AUA178V DESCEND 3000 ALT	AUA178V DESCEND 3000 ALT
NO_CALLSIGN NO_CONCEPT	NO_CALLSIGN NO_CONCEPT
AUA176 TURN_RIGHT_HEADING 310	
AUA176 CLEARED_ILS 34	
AUA178V TURN_RIGHT_HEADING 250	AUA178V TURN_RIGHT_HEADING 250
AUA178V TURN_RIGHT_HEADING 310	AUA178V TURN_RIGHT_HEADING 310
AUA178V INTERCEPT_LOCALIZER 34	AUA178V CLEARED_ILS 34
AUA176 HANDOVER TOWER	AUA176 HANDOVER TOWER
AUA176 HANDOVER_FREQUENCY 123.8	AUA176 HANDOVER_FREQUENCY 123.8
NO_CALLSIGN NO_CONCEPT	AUA176 NO_CONCEPT
AUA178V HANDOVER TOWER	AUA178V HANDOVER TOWER
AUA178V HANDOVER_FREQUENCY 123.8	AUA178V HANDOVER_FREQUENCY 123.8
AUA152L INIT_RESPONSE	AUA152L NO_CONCEPT
AUA152L DESCEND 3000 ALT	AUA152L DESCEND 3000 ALT
AUA152L TURN_RIGHT_HEADING 250	AUA152L TURN_RIGHT_HEADING 250
AUA302 INIT_RESPONSE	AUA302 NO_CONCEPT
AUA302 DESCEND 3000 ALT	AUA302 DESCEND 3000 ALT
AUA152L TURN_RIGHT_HEADING 310	
AUA152L CLEARED_ILS 34	

Appendix C Results of Unit-Tests

C.1 SYS-OFF-003 Control Region Boundary (Unit-Test)

In this test, the predicted commands are calculated with the controller responsibility area specified in Figure 22 first and afterwards with a smaller responsibility area specified in Figure 23. Figure 21 shows part of the code of this test.

```
/*****************************************************************************/
/*****************************************************************************/
/*****************************************************************************/
void GenerateRasterContextPrague::sys_off_003_RqTest()
/*****************************************************************************/
/*****************************************************************************/
/* SYS-OFF-003 changing the reponsibility area of the controller.
   At start check with normal sized area. Then recordConfig is replaced by
   another one, with smaller responsibility area.
   Test is successful if some commands are no longer in the set of
   predicted commamds.
*/
/*****************************************************************************/
{
   INIT_TEST_FUNC(sys_off_003_RqTest);
   // first check before changes
   sys_off_checkBefore();
   // make the changes
   string jsonfile = "$(GENTRAG_HOME)/tools/src/checkContext/data/testdata/prague/recordingConfiguration.json";
   string jsontestfile = "$(GENTRAG_HOME)/tools/src/checkContext/data/testdata/prague/mini_recordingConfiguration.json";
   C_File jf(jsonfile);
   jf.renameFile("test.bak");
   C_File jft(jsontestfile);
   jft.renameFile(jsonfile);
```

Figure 21 Start of Unit-Test to show fulfilment of requirement SYS-OFF-003


```
{
        "ICAO": "LKPR",
        "InboundDirection": [
                "24", "30"
        ],

        "OverflightsOnFrequency": true,
        "ResponsibilityAreaPolyon": {
                "outer": [
                        { "Latitude": 49.927171321685655, "Longitude": 12.585148107295682 },
                        { "Latitude": 51.02243000019004, "Longitude": 14.347028620701678 },
                        { "Latitude": 50.746897144447274, "Longitude": 15.543233214589375 },
```

{ "Latitude": 49.439334843884936, "Longitude": 15.851133503396342 },
{ "Latitude": 49.33958180193022, "Longitude": 15.454221319165104 },
{ "Latitude": 49.13293925866966, "Longitude": 13.480030716574944 }
] ,
"inner": "none",
"AltitudeMin": "none",
"Comment": "22.02.2017 from 24500 to 0",
"AltitudeMax": "FL 280"
},

"ResponsibilityAreaPolyonARR": {
"outer": [
{ "Latitude": 49.927171321685655, "Longitude": 12.585148107295682 },
{ "Latitude": 51.02243000019004, "Longitude": 14.347028620701678 },
{ "Latitude": 50.746897144447274, "Longitude": 15.543233214589375 },
{ "Latitude": 49.439334843884936, "Longitude": 15.851133503396342 },
{ "Latitude": 49.13293925866966, "Longitude": 13.480030716574944 }
],
"inner": "none",
"AltitudeMin": "FL 20",
"Comment": "22.02.2017 from 24500 to 1700 (error), otherwise only 2900",
"AltitudeMax": "FL 280"
},
"ResponsibilityAreaPolyonDEP": {
"outer": [
{ "Latitude": 50.17600668516628, "Longitude": 13.643586524525736 },
{ "Latitude": 50.4442989211868, "Longitude": 14.135253795826195 },
{ "Latitude": 50.320914105882764, "Longitude": 14.423341480150883 },
{ "Latitude": 49.62550654559214, "Longitude": 14.74057827562313 },
{ "Latitude": 49.964246159964496, "Longitude": 13.921878218552758 }
],
"inner": "none",
"AltitudeMin": "none",
"Comment": "22.02.2017 from 13700 to 0",
"AltitudeMax": "FL 150"
},

"ResponsibilityAreaPolyonOVL": {
"outer": [
{ "Latitude": 50.511583740293176, "Longitude": 14.096496344481492 },
{ "Latitude": 50.84995238629133, "Longitude": 14.786231514119738 },
{ "Latitude": 49.86191097653569, "Longitude": 15.281715935164025 },
{ "Latitude": 49.33958180193022, "Longitude": 15.454221319165104 },
{ "Latitude": 49.40582571132225, "Longitude": 15.002218755614166 },
{ "Latitude": 49.496279985980586, "Longitude": 14.702727078814398 }
] ,
"inner": "none",


```
                "AltitudeMin": "none",
          "Comment": "22.02.2017 from 25400 to 0",
                "AltitudeMax": "FL 280"
        }
}
```


Figure 22: Recording configuration describing full responsibility area of controller (requirement SYS-OFF-003).


```
{
        "ICAO": "LKPR",
        "InboundDirection": [
                "24", "30"
        ],
        "OutboundDirection": [
                "24"
        ],
        "OverflightsOnFrequency": true,

        "ResponsibilityAreaPolyon": {
                "outer": [
                        {  "Latitude": 49.9, "Longitude": 12.5 },
                        {  "Latitude": 50.0, "Longitude": 12.5 },
                        {  "Latitude": 50.0, "Longitude": 12.8 },
                        {  "Latitude": 49.9, "Longitude": 12.8 }
                         ] ,
                "inner": "none",
                "AltitudeMin": "none",
          "Comment": "22.02.2017 from 24500 to 0",
                "AltitudeMax": "FL 280"
        },

                "ResponsibilityAreaPolyonARR": {
                "outer": [
                        {  "Latitude": 49.9, "Longitude": 12.5 },
                        {  "Latitude": 50.0, "Longitude": 12.5 },
                        {  "Latitude": 50.0, "Longitude": 12.8 },
                        {  "Latitude": 49.9, "Longitude": 12.8 }
                         ],
                "inner": "none",
                "AltitudeMin": "FL 20",
          "Comment": "22.02.2017 from 24500 to 1700 (error), otherwise only 2900",
                "AltitudeMax": "FL 280"
```

```
        },
        "ResponsibilityAreaPolyonDEP": {
                "outer": [
                        {  "Latitude": 49.9, "Longitude": 12.5 },
                        {  "Latitude": 50.0, "Longitude": 12.5 },
                        {  "Latitude": 50.0, "Longitude": 12.8 },
                        {  "Latitude": 49.9, "Longitude": 12.8 }
                         ],
                "inner": "none",
                "AltitudeMin": "none",
          "Comment": "22.02.2017 from 13700 to 0",
                "AltitudeMax": "FL 150"
        },

        "ResponsibilityAreaPolyonOVL": {
                "outer":  [
                        {  "Latitude": 49.9, "Longitude": 12.5 },
                        {  "Latitude": 50.0, "Longitude": 12.5 },
                        {  "Latitude": 50.0, "Longitude": 12.8 },
                        {  "Latitude": 49.9, "Longitude": 12.8 }
                 ] ,
                "inner": "none",
                "AltitudeMin": "none",
          "Comment": "22.02.2017 from 25400 to 0",
                "AltitudeMax": "FL 280"
        }
}
```


Figure 23: Recording configuration describing smaller responsibility area of controller (requirement SYS-OFF-003).

Callsign AUA578X (an arrival) is at position "Latitude": 50.1825, and "Longitude": 14.4903.

The bigger responsibility area results in the following subset of predicted commands:

```
Context which is send by interface before changing configurations:
1513100186 DYNAMIC RESET
ADD AUA578X CLEARED_ILS 24 1
ADD AUA578X CLEARED_ILS 30 1
ADD AUA578X CLEARED_NDB 24 1
ADD AUA578X CLEARED_NDB 30 1
ADD AUA578X CLEARED_RNAV 24 1
ADD AUA578X CLEARED_RNAV 30 1
ADD AUA578X DESCEND 3000 ALT 1
ADD AUA578X DESCEND 3500 ALT 1
ADD AUA578X DESCEND 4000 ALT 1
ADD AUA578X DESCEND 5000 ALT 1
```


Afterwards no commands are predicted for the AUA578X.

We test this by the following code.

```
   // sys-off-003 pre, check that these cmd are in context
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
CLEARED_ILS 24"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
CLEARED_NDB 30"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
DESCEND 3500 ALT"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
TURN_RIGHT_HEADING 340"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
TURN_RIGHT_HEADING 290"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
TURN_RIGHT_HEADING 25"));
```

And afterwards:

```
   // check that these cmd are no longer in context
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
CLEARED_ILS 24"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
CLEARED_NDB 30"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
DESCEND 3500 ALT"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
TURN_RIGHT_HEADING 340"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
TURN_RIGHT_HEADING 290"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
TURN_RIGHT_HEADING 25"));
```

We have shown that by changing the coordinates of the responsibility area we can change the set of predicted commands. If a commands is not predicted it is not expected and therefore it will not be accepted by the interface of the speech recognizer.

The requirement is fulfilled!

C.2 SYS-OFF-004 Runway change (Unit-Test)

Similar to appendix C.1 we show that we can change the allowed runways, see Figure 25 and Figure 26:

```
/******************************************************************************/
/******************************************************************************/
/******************************************************************************/
void GenerateRasterContextPrague::sys_off_004_RqTest()
/******************************************************************************/
/* SYS-OFF-004 changing the set of possible runways
   At start check with  "InboundDirection": ["24", "30"], and afterwards with
   "InboundDirection": ["27", "33"],
   We expected different CLEARED_ILS commands.
/******************************************************************************/
{
   INIT_TEST_FUNC(sys_off_004_RqTest);

   // first check before changes
   sys_off_checkBefore();

   string jsonfile = "$(GENTRAG_HOME)/tools/src/checkContext/data/testdata/prague/recordingConfiguration.json";
   string jsontestfile = "$(GENTRAG_HOME)/tools/src/checkContext/data/testdata/prague/rwy_recordingConfiguration.json";
   C_File jf(jsonfile);
   jf.renameFile("test.bak");
   C_File jft(jsontestfile);
   jft.renameFile(jsonfile);
```

Figure 24 First part of unit test for requirement SYS-OFF-004

```
"ICAO": "LKPR",
"InboundDirection": [
    "24", "30"
],
"OutboundDirection": [  ],
"OverflightsOnFrequency": true,
```

Figure 25 Original recordingConfiguration.json file with inbound runways 24 and 30

```
"ICAO": "LKPR",
"InboundDirection": [
    "27", "33"
],
"OutboundDirection": [  ],
"OverflightsOnFrequency": true,
```

Figure 26 Changed rwy_recordingConfiguration.json file with inbound runways 27 and 33

We test with the following software code:

```
   // runways which were generated before changing runway configuration.
   // Now they are not expected anymore in set of predicted commands
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
CLEARED_ILS 24"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
CLEARED_NDB 24"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
CLEARED_ILS 30"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
CLEARED_NDB 30"));

   // runways which were generated when changing runway configuration
```



```
    assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X CLEARED_ILS 27"));
    assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X CLEARED_NDB 27"));
    assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X CLEARED_ILS 33"));
    assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X CLEARED_NDB 33"));
```

We have shown that by changing the inbound and outbound runways we can change the set of predicted commands. If a commands is not predicted it is not expected and therefore it will not be accepted by the interface of the speech recognizer.

The successful test shows that the requirement is fulfilled.

C.3 SYS-OFF-005 Change Controller Working Position (Unit-Test)

Similar to appendix C.1 we show that we can change controller working position.

```
/******************************************************************************/
void GenerateRasterContextPrague::sys_off_005_RqTest()
/******************************************************************************/
/* Test SYS-OFF-005: change controller working position PEC / AEC
   checkBefore works with PEC, then swich to AEC
   AC AUA578Y is outside PEC area, but inside AEC */

/******************************************************************************/
{
   INIT_TEST_FUNC(sys_off_005_RqTest);

   // first check before changes
   sys_off_checkBefore();

   string jsonfile = "$(GENTRAG_HOME)/tools/src/checkContext/data/testdata/prague/recordingConfiguration.json";
   string jsontestfile = "$(GENTRAG_HOME)/tools/src/checkContext/data/testdata/prague/recordingConfiguration_AEC.json";
   C_File jf(jsonfile);
   jf.renameFile("test.bak");
   C_File jft(jsontestfile);
   jft.renameFile(jsonfile);
```

Figure 27 First part of unit test for requirement SYS-OFF-005

We test with the following software code:

```
// sys-off-005 precheck
    assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578Y CLEARED_ILS 24"));
    assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578Y CLEARED_ILS 30"));
```

```
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578Y
CLEARED_NDB 30"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578Y
DESCEND 4000 Alt"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578Y
DESCEND 3500 Alt"));
```

Before the change there are no commands for AUA578Y. After change we test if there are some commands for AUA578Y:

```
   // additional ctx if in AEC
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578Y
CLEARED_ILS 24"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578Y
CLEARED_ILS 30"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578Y
CLEARED_NDB 30"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578Y
DESCEND 4000 ALT"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578Y
DESCEND 3500 ALT"));
```

The successful test shows that the requirement is fulfilled.

C.4 SYS-OFF-006 recordingConfig changeable (Unit-Test)

This needs no extra test, because it is covered by SYS-OFF-003, SYS-OFF-004 and SYS-OFF-005.

As these tests are successful, the requirement is fulfilled.

C.5 SYS-OFF-002 Change Waypoints (Unit-Test)

Similar to appendix C.1 we show that we can change waypoints. One Waypoint gets removed from the waypoint.osm and the raster.xml file, two other waypoints are added.

In the original osm the waypoint ERASU is defined as:

```
<node id="-2101" lat="50.26875278" lon="14.47814167">
  <tag k="name" v="ERASU"/>
  <tag k="airspace" v="waypoint"/>
  <tag k="DIRECT_TO" v="yes"/>
  <tag k="PronouncedAs-0" v="D_B  to be done"/>
  <tag k="PronouncedAs-1" v="erasu"/>
  <tag k="PronouncedAs-2" v="echo romeo alpha sierra uniform"/>
</node>
```

For testing we change this to:

```
<node id="-2101" lat="50.26875278" lon="14.47814167">
  <tag k="name" v="ERATU"/>
  <tag k="airspace" v="waypoint"/>
  <tag k="DIRECT_TO" v="yes"/>
```



```
    <tag k="PronouncedAs-0" v="D_B  to be done"/>
    <tag k="PronouncedAs-1" v="eratu"/>
    <tag k="PronouncedAs-2" v="echo romeo alpha tango alpha"/>
  </node>
```

In the raster.xml file we replace:

```
<cmdgrp rule='?IOS==ARR:DIRECT_TO' values='AKEVA,ERASU,NIRGO,OKG,PR511,PR512,
PR516,PR517,PR521,PR522,PR530,PR531,PR532,RAPET,RATEV,VARIK,VOZ' />
```

by:

```
<cmdgrp rule='?IOS==ARR:DIRECT_TO' values='AKEVA,ERATU,NIRGO,OKG,PR511,PR512,
PR516,PR517,PR521,PR522,PR530,PR531,PR532,RAPET,RATEV,VARIK,VOZ' />
```

In the test we first make a checkBefore and then replace the osm and xml files, see code below. Then we check if changes have worked.

```
void GenerateRasterContextPrague::sys_off_002_RqTest()
/*****************************************************************************/
/* change waypoints                                                         */
/*****************************************************************************/
{
   /* Wegpunkte rausnehmen bzw. dazugeben */
   INIT_TEST_FUNC(sys_off_002_RqTest);

   // first check before changes
   sys_off_checkBefore_S2();

   string orgfile = "$(GENTRAG_HOME)/tools/src/checkContext/data/rasterData/AEC-lkpr-raster.xml";
   string testfile = "$(GENTRAG_HOME)/tools/src/checkContext/data/rasterData/wp_AEC-lkpr-raster.xml";
   C_File jf(orgfile);
   jf.renameFile("./test.bak", true);
   C_File jft(testfile);
   jft.renameFile(orgfile, true);

   string org_wp_osm_file = "$(GENTRAG_HOME)/tools/src/checkContext/data/testdata/prague/waypoints.osm";
   string test_wp_osm_file = "$(GENTRAG_HOME)/tools/src/checkContext/data/testdata/prague/sys002_waypoints.osm";
   C_File osmf(org_wp_osm_file);
   osmf.renameFile("./wp_osm.bak", true);
   C_File osmft(test_wp_osm_file);
   osmft.renameFile(org_wp_osm_file, true);

   string wavDirName = "$(GENTRAG_HOME)/tools/src/checkContext/data/testdata/prague/wavOneRegion";
   string config_file = "$(GENTRAG_HOME)/tools/src/checkContext/data/rasterData/AEC-lkpr.cfg";

   Context context(config_file, wavDirName, nullptr, true);
```

Figure 28 First part of unit test for requirement SYS-OFF-002

We test with the following software code:

```
   // sys-off-002 pre, check that these cmd are in context
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
DIRECT_TO AKEVA"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
DIRECT_TO ERASU"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
DIRECT_TO NIRGO"));
   // and these not
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
DIRECT_TO ERATU"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
DIRECT_TO EXTRA"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
TRANSITION EXTRA"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
HOLDING EXTRA"));
```

Before the change there are no commands using waypoint ‘EXTRA’. After renaming ERASU to ERATU and adding EXTRA in the waypoint.osm and the raster.xml, we test:

```
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
DIRECT_TO AKEVA"));
   assertTestCont(!context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
DIRECT_TO ERASU"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
DIRECT_TO ERATU"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
DIRECT_TO NIRGO"));
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
DIRECT_TO EXTRA"));
   // Transition and Holding should not change (its an other entry in OSM file)
   assertTestCont(context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
HOLDING ERASU"));
   assertTestCont( !context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
TRANSITION EXTRA"));
   assertTestCont( !context.mpc_contextOutput->GetDynContext().isMemberOf("AUA578X
HOLDING EXTRA"));
```

Because of changing the name ERASU to ERATU the first one should never appear in context, instead the new name should appear there. The additional Waypoint ‘EXTRA’should appear in a DIRECT_TO, but not in TRANSITION or HOLDING because they use different entries in the osm file.

The successful test shows that the requirement is fulfilled.

C.6 SYS-OFF-007 System offline configuration options

The original requirement is:

THE SYSTEM SHALL/WILL[11] allow following OFFLINE configuration options:

- Detailed configuration of Sectors and controller positions:
  - Name
  - Type
  - Boundary
  - Area of responsibility
  - Area of Interest
  - Frequency
  - Including adjacent units
- Detailed configuration of airspace, its structure and layout
  - Airspace limits
  - Airspace restrictions
  - Route structure
  - Configuration of STARs
  - Configuration of SIDs
  - Configuration of waypoints (SYS-OFF-002)
  - Configuration of holdings
- Detailed configuration of Aerodrome
  - Configuration of RWYs (SYS-OFF-004)
  - Configuration of types of Approaches

Detailed configuration of Command types and their properties (REC-FUN-017)

  - Processing Y|N
  - Sending to external system Y|N
  - Item Mandatory|Optional
- Detailed configuration options for recognition
- Detailed configuration of interfaces
- Detailed configuration of Recording (SYS-OFF-006)
- Detailed configuration of user roles

Detailed configuration options of HMI

[11] WILL (for HMI part) / SHALL for rest.

We now detail (with smaller fonts), how the many sub-requirements of this requirement are implemented resp. are challenges for MALORCA successor projects aiming at higher TRL.

- Detailed configuration of Sectors and controller positions:
  - Name (not implemented, it is related to HMI, see also left part of Figure 31,)
  - Type (not implemented, it is related to HMI)
  - Boundary
  - Area of responsibility
  - Area of Interest

See for all 3 recording Configuration.json file:

```
"InboundDirection": [
    "24", "30"
],
"OutboundDirection": [  ],
"OverflightsOnFrequency": true,
```

Figure 29 Area of responsibility are here inbounds and overflights, but no outbounds

```
"ResponsibilityAreaPolyon": {
    "outer": [
        {  "Latitude": 49.927171321685655,   "Longitude": 12.585148107295682 },
        {  "Latitude": 51.02243000019004,   "Longitude": 14.347028620701678 },
        {  "Latitude": 50.746897144447274,   "Longitude": 15.543233214589375 },
        {  "Latitude": 49.439334843884936,   "Longitude": 15.851133503396342 },
        {  "Latitude": 49.33958180193022,   "Longitude": 15.454221319165104 },
        {  "Latitude": 49.13293925866966,   "Longitude": 13.480030716574944 }
         ] ,
    "inner": "none",
    "AltitudeMin": "none",
    "Comment": "22.02.2017 from 24500 to 0",
    "AltitudeMax": "FL 280"
},
```

Figure 30 General boundary specified by an including outer polygon and an excluding inner polygon


```
    "ResponsibilityAreaPolyonARR": {
    "outer": [
        {  "Latitude": 49.927171321685655,   "Longitude": 12.585148107295682 },
        {  "Latitude": 51.02243000019004,   "Longitude": 14.347028620701678 },
        {  "Latitude": 50.746897144447274,   "Longitude": 15.543233214589375 },
        {  "Latitude": 49.439334843884936,   "Longitude": 15.851133503396342 },
        {  "Latitude": 49.13293925866966,   "Longitude": 13.480030716574944 }
        ],
    "inner": "none",
    "AltitudeMin": "FL 20",
    "Comment": "22.02.2017 from 24500 to 1700 (error), otherwise only 2900",
    "AltitudeMax": "FL 280"
},
"ResponsibilityAreaPolyonDEP": {
    "outer": [
        {  "Latitude": 50.17600668516628,   "Longitude": 13.643586524525736 },
        {  "Latitude": 50.4442989211868,   "Longitude": 14.135253795826195 },
        {  "Latitude": 50.320914105882764,   "Longitude": 14.423341480150883 },
        {  "Latitude": 49.62550654559214,   "Longitude": 14.74057827562313 },
        {  "Latitude": 49.964246159964496,   "Longitude": 13.921878218552758 }
        ],
    "inner": "none",
    "AltitudeMin": "none",
    "Comment": "22.02.2017 from 13700 to 0",
    "AltitudeMax": "FL 150"
},

"ResponsibilityAreaPolyonOVL": {
    "outer":  [
    {  "Latitude": 50.511583740293176,   "Longitude": 14.096496344481492 },
    {  "Latitude": 50.84995238629133,   "Longitude": 14.786231514119738 },
    {  "Latitude": 49.86191097653569,   "Longitude": 15.281715935164025 },
    {  "Latitude": 49.33958180193022,   "Longitude": 15.454221319165104 },
    {  "Latitude": 49.40582571132225,   "Longitude": 15.002218755614166 },
    {  "Latitude": 49.496279985980586,   "Longitude": 14.702727078814398 },
    {  "Latitude": 49.85986,   "Longitude": 14.258769 }
    ] ,
    "inner": "none",
    "AltitudeMin": "none",
    "Comment": "22.02.2017 from 25400 to 0",
    "AltitudeMax": "FL 280"
}
```


Figure 31 Detailed area of interest for each responsibility area specified by outer and inner polygons

- Frequency

```
"ThisFrequencyName": [{
    "Name": "DIRECTOR",
    "Frequency": "119.01"
}]
```

```
"HandoverTo": [{
    "oldName": "RUZYNE_TOWER",
    "Name": "TOWER",
    "oldFrequency": "118.1",
    "Frequency": "118.11"
}, {
    "Name": "RADAR",
    "oldFrequency": "127.575",
    "Frequency": "127.58"
}, {
    "Name": "RADAR",
    "oldFrequency": "120.525",
    "Frequency": "120.53"
}, {
    "Name": "KBELY_TOWER",
    "oldFrequency": "120.875",
    "Frequency": "120.88"
}, {
    "Name": "VODOCHODY_TOWER",
    "oldFrequency": "133.075",
    "Frequency": "133.08"
}, {
    "Name": "RUZYNE_INFORMATION",
    "oldFrequency": "126.1",
    "Frequency": "126.1"
}],
```

Figure 32 specification of own frequency and handover to positions in frequency.json or in recordingConfiguration.json file

- Including adjacent units

```
"HandoverFrom": [{
    "Comment": "Not updated still after 2016-11-11",

    "Name": "RADAR",
    "Frequency": "127.575"
},{
    "Name": "RADAR",
    "Frequency": "120.525"
},{
    "Name": "TOWER",
    "Frequency": "118.1"
},{
    "Name": "RUZYNE_INFORMATION",
    "Frequency": "118.1"
},{
    "Name": "VODOCHODY_TOWER",
    "Frequency": "133.075"
},{
    "Name": "KBELY_TOWER",
    "Frequency": "120.875"
}],
```

Figure 33 specification handover from frequency in frequency.json or in recordingConfiguration.json file

- Detailed configuration of airspace, its structure and layout
  - Airspace limits (see Figure 29 to Figure 31)
  - Airspace restrictions (see Figure 29 to Figure 31)
  - Route structure (currently not supported for commandprediction)

- Configuration of STARs (currently not supported for commandprediction)
- Configuration of SIDs (currently not supported for commandprediction)
- Configuration of waypoints (SYS-OFF-002) (see requirement SYS-OFF-002 and waypoints.osm file)
- Configuration of holdings (see holdings.json file)

```
"Holdings": [{
    "Name": "ERASU",
    "PronouncedAs":
    ["D_B AO1_I M_I UH0_I K_I S_E",
    "erasu"
    ],
    "MinFL": 40,
    "MaxFL": 160
},
{
    "Name": "RATEV",
    "PronouncedAs":
    ["D_B AO1_I M_I UH0_I K_I S_E",
    "ratev"
    ],
    "MinFL": 40,
    "MaxFL": 160
},
```

Figure 34 part of holdings.json file

- Detailed configuration of Aerodrome
  - Configuration of RWYs (SYS-OFF-004) (see recordingsConfiguration file and SYS-OFF-004)
  - Configuration of types of Approaches (currently not supported for commandprediction)

Detailed configuration of Command types and their properties (REC-FUN-017) (see supportedCommands .json file)

- Processing Y|N
- Sending to external system Y|N
- Item Mandatory|Optional

```
"Commands": [
{
    "Type": "DESCEND",
    "SupportedInThisAirspace": true,
    "Values": [{
            "Unit": "Alt",
            "Min": 3000,
            "Max": 6000,
            "StepSize": 1000
            },
            {
            "Unit": "FL",
            "Min": 60,
            "Max": 250,
            "improve": "use 130, but currently same value for CLIMB  and DESCEND are used",
            "StepSize": 10
            }
    ],
    "ConditionalClearances": false
},
{
    "Type": "CLIMB",
    "SupportedInThisAirspace": true,
    "Values": [{
            "Unit": "Alt",
            "Min": 3000,
            "Max": 6000,
            "StepSize": 1000
            },
            {
            "Unit": "FL",
            "Min": 60,
            "commentMax": "at least once already 240 observed",
            "Max": 250,
            "improve": "use 160, but currently same value for CLIMB  and DESCEND are used",
            "StepSize": 10
            }
    ],
    "ConditionalClearances": false
```

```
{
    "Type": "DIRECT_TO",
    "SupportedInThisAirspace": true,
    "ConditionalClearances": false
},
{
    "Type": "CLEARED_ILS",
    "SupportedInThisAirspace": true,
    "ConditionalClearances": false
},
{
    "Type": "INTERCEPT_LOCALIZER",
    "SupportedInThisAirspace": true,
    "ConditionalClearances": false
},
{
    "Type": "CLEARED_NDB",
    "SupportedInThisAirspace": true,
    "ConditionalClearances": false
},
{
    "Type": "CLEARED_RNAV",
    "SupportedInThisAirspace": true,
    "ConditionalClearances": false
},
{
    "Type": "CLEARED_VISUAL",
    "SupportedInThisAirspace": false,
    "ConditionalClearances": false
},
{
    "Type": "EXPECT_RUNWAY",
    "SupportedInThisAirspace": false,
    "ConditionalClearances": false
},
```


Figure 35 Excerpt from supportedCommands.json file

- Detailed configuration options for recognition (see previous figures, sub-requirement needs more details in the future if higher TRL are needed)
- Detailed configuration of interfaces (sub-requirement needs more details in the future if higher TRL are needed)
- Detailed configuration of Recording (SYS-OFF-006) (see requirement SYS-OFF-006)
- Detailed configuration of user roles (not implemented, it is related to HMI)
- Detailed configuration options of HMI (not implemented, it is related to HMI)

Abbreviations

ABSR	Assistant Based Speech Recognition
ACC	Area Control Center
ACG	Austro Control Österreichische Gesellschaft für Zivilluftfahrt mit beschränkter Haftung
AcListant	Active Listing Assistant
ANS CR	Air Navigation Services of the Czech Republic
ANSP	Air Navigation Service Provider
APP	Approach, Approach Control Unit, working position approach (often also called feeder or pickup position)
ASR	Automatic Speech Recognition
ATCO	ATCO Air Traffic Controller
COOPANS	COOPeration between ANSProvider
DLR	Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Centre)
DM	Dissemination Manager
DoD	Definition of Done
Idiap	Idiap Research Institute
MALORCA	Machine Learning of Speech Recognition Models or Controller Assistance
MLS	MALORCA Learning System
NTP	Network Time Protocol
LOWW	Vienna Airport
PIC	Pilot in command
PL	Project Leader
PMP	Project Management Plan
POC	Point of Contact
PRG	Prague
PTT	Push-to-Talk
SES	Single European Sky
SID	SESAR Innovation Days
SJU	SESAR Joint Undertaking
tbd	To be defined
TMA	Terminal Manoeuvring Area
TWR	Aerodrome Control Tower
UdS	See USAAR
USAAR	Saarland University
WP	Work Package

Appendix D List of Tables

Appendix E References

[1] Cerna, Aneta; Monhart, David: MALORCA project: D1-1: Operational Concept Document, version 3.00; 20. February 2017

[2] Cerna, Aneta; Helmke, Hartmut; Finke, Michael; Nesvadba, Matej; Motlicek, Petr; Szaszak, György; Oualil, Youssef:: MALORCA project: D1-2: System Requirement Specification Document, version 1.00; 04. August 2016

[3] Cerna, Aneta; Helmke, Hartmut; Finke, Michael; Nesvadba, Matej; Motlicek, Petr; Szaszak, György; Oualil, Youssef; Windisch, Christian; Kern: MALORCA project: D1-2: System Requirement Specification Document, version 2.00; 23. September 2016

[4] Helmke, H.; Rataj, J.; Mühlhausen, Th.; Ohneiser, O.; Ehr, H.; Kleinert M.; Oualil, Y.; Schulder, M.; Klakow, D.;
Assistant-Based Speech Recognition for ATM Applications,
23-26. June 2015, Lisbon, Portugal, 11th FAA/Eurocontrol ATM-Seminar

[5] 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
Proceedings of the 5th International Conference on Applied Human Factors and Ergonomics AHFE 2014, Advances in Human Aspects of Transportation: Part II, N. Stanton, S. Landry, G. Di Bucchianico, and A. Vallicelli, Eds. Krakow, Poland: CRC Press, 2014, ISBN: 978-1-4951-2098-5, pp. 492-503.