DLR-IB-FL-BS-2025-107

Ontology for Air Traffic Control Command Annotation

Description

Hartmut Helmke,
Matthias Kleinert,
Oliver Ohneiser

Deutsches Zentrum
DLR für Luft- und Raumfahrt

Document properties

Title	Ontology for Air Traffic Control Command Annotation
Subject	Description
Institute	DLR, Institute of Flight Guidance
Compiled by	Hartmut Helmke
Participants	Matthias Kleinert, Oliver Ohneiser
Classification	A/I (open access)
Date	2025-07-04
Version	3.00
Release Note	Document is released in accordance to Document Control Sheet

Abstract

The voice communication between air traffic control (ATC) / apron control and aircraft pilots is essential for the safe and efficient flow of air traffic. Despite this fact, technical ATC systems have no access to the information within this verbal communication. Speech Recognition and Understanding technologies can close this gap so that technical systems can access information from the voice communication channel and build downstream applications upon this knowledge. This process requires a clear definition of all relevant information items that can be part of ATC voice communication, e.g., callsign, different kind of commands and so on. For this purpose, this document describes an ontology that conceptually defines relevant elements within ATC communications. This includes the definition of the ontology and relevant elements, best practices on how to get from spoken words to ontology commands and a machine-readable format to transfer the information among different systems.

Index of contents

1 Introduction 6
1.1 Background 6
1.2 Structure of the Document 6
1.3 General Guidelines 6

2 Ontology for Command Annotation 7

3 Commands of the Annotation Ontology 9

4 General Rules for ATC Command Annotation 10
4.1 Multiple commands in one utterance 10
4.2 Guessing about utterance meanings 11
4.3 Usage of runway name in commands 11
4.4 Speaking to more than one callsign (break, break) 13
4.5 More than one speaker per utterance 14
4.6 Conditional Clearances 15
4.6.1 Conjunction keyword "UNTIL" 15
4.6.2 Conjunction keyword "WHEN" 16
4.6.3 Conjunction keyword "IF" 17
4.6.4 Conjunction keyword "AFTER" 17
4.6.5 Complex conditional Clearances 18
4.7 Best Practice Examples of Various Command Types 19

5 Machine-Readable Format for Exchange of Ontology Information 20
5.1 Definition of Key-Value-Pairs 20
5.2 Simple Examples 22
5.3 Complex Examples with Plausibility 24
5.4 Complex Examples with Intermediate Output for Transmissions 25
5.5 Examples with word classification 26

References 27

Appendix 28

6 Vertical Commands 29
6.1 DESCEND 29
6.2 CLIMB 30
6.3 ALTITUDE 31
6.4 MAINTAIN ALTITUDE 32
6.5 Vertical commands with additional qualifiers 32

6.6 STOP_CLIMB 33
6.7 STOP_DESCEND 33
6.8 STOP_ALTITUDE 33
6.9 MAINTAIN PRESENT_ALTITUDE 33

7 Changing Direction Commands 35
7.1 Heading, Turn, Turn_By 35
7.2 MAINTAIN HEADING 37
7.3 CONTINUE PRESENT_HEADING 37
7.4 DIRECT_TO, TRANSITION, FOLLOW_ROUTE 38
7.4.1 DIRECT_TO 38
7.4.2 DIRECT_TO with TIME constraint 42
7.4.3 TRANSITION 42
7.4.4 FOLLOW_ROUTE 42
7.4.5 NAVIGATION_OWN 43

8 Vertical Speed Commands 45
8.1 RATE_OF_CLIMB 45
8.2 RATE_OF_DESCENT 45
8.3 RATE_OF_DESCENT OWN, EXPEDITE, MAX 45
8.4 VERTICAL_RATE 46
8.5 VERTICAL_RATE OWN, EXPEDITE, MAX 46
8.6 EXPEDITE_PASSING 46
8.7 RATE_OF_CLIMB OWN, EXPEDITE, MAX 47

9 Speed-Commands 48
9.1 REDUCE 48
9.2 INCREASE 48
9.3 SPEED 48
9.4 MAINTAIN PRESENT_SPEED 49
9.5 MAINTAIN SPEED 49
9.6 REDUCE_BY and INCREASE_BY 50
9.7 TIME_CMD commands 50
9.8 Special Speed Values: NO_SPEED_RESTRICTIONS etc. 51

1 Introduction

The voice communication between air traffic control (ATC) / apron control and aircraft pilots is from a technical point of view just a transmission of audio data among different systems. These underlying systems have no deeper understanding about the information contained within the audio data. Speech Recognition and Understanding technologies can provide a mean to digitize the audio signal, i.e., to transform the audio data into a format that allows other technical systems to interpret the contained information to build downstream applications upon this knowledge.

1.1 Background

The transformation of ATC audio data is often done in two steps. The first step is a Speech-to-Text (STT) transformation often referred to as Automatic Speech Recognition (ASR). In this step the audio data is transformed into a sequence of words (transcription), which represents the content of the audio data in written form. The transcription of a given audio input could look like the following:

```
"lufthansa three echo romeo descend flight level one two zero rate
of descent is two thousand five hundred feet or greater"
```

This transcription describes more or less in a human readable form, what is contained within the audio data, but it does not provide a clear interpretation of the meaning behind the transcription. Therefore, in a second step a Text-to-Concept (TTC) transformation (annotation) is required also referred to as Speech Understanding. This transformation takes the transcript, identifies relevant information, e.g. addressed callsigns and commands, and transforms it into a format, which allows a clear interpretation and transmission among different systems and applications.

1.2 Structure of the Document

The following Chapter 2 provides an overview of an ontology that can be used for the TTC transformation. This includes a definition on what conceptual elements are contained within ATC communication, e.g., callsign, command type, command value. Building up on that, Chapter 3 describes the allowed content of the conceptual elements in more detail. Chapter 4 contains "best practices" about the transformation from transcriptions to ontology annotations. Chapter 5 describes how the ontology can be implemented in a machine-readable format based on JSON. Furthermore, there is a list of references. The Appendix contains best-practice examples of converting transcriptions into ATC-ontology conform annotations per command type from chapters 6 onwards.

1.3 General Guidelines

The defined ontology covers the majority of existing commands in air traffic control and apron control. However, it is (and might never be) 100% complete. If you figure out repeated necessity of new/adopted command types/values/qualifiers where existing suggestions do not make sense, i.e., there must be some reason for electronic systems to be aware of the relevant additional or modified information, please feel free to suggest them to the authors of this document (Hartmut.Helmke@dlr.de, Oliver.Ohneiser@dlr.de, Matthias.Kleinert@dlr.de). Your suggestions are very welcome and will be discussed and maybe agreed on a more global level between speech recognition and understanding partners in air traffic control.

2 Ontology for Command Annotation

This chapter describes an ontology, which provides a clearly defined format to be used as output of the TTC transformation. It provides a definition of the relevant elements within ATC radio transmissions and furthermore describes what content these elements can hold. This chapter does not detail, how to transcribe an ATC transmission. "speed bird alfa nine one" could be e.g. transcribed as "speed bird alpha 9 1" or "speed_bird alpha nine one" etc. The interested reader can find the agreed transcriptions rules in [1]. These specify that, e.g., "[hes]" means "hesitation", "[unk]" means "unknown", "*" is used for swallowed parts, or "[NE German] [/NE]" is used in case of non-English language in an utterance.

The here presented ontology was originally defined by fourteen European partners from the air traffic management (ATM) industry and air navigation service providers (ANSPs) funded by *SESAR 2020* [2].

The ontology was improved and refined through the course of various projects, such as *STARFiSH* [3], "*HMI Interaction Modes for Airport Tower*" [4] [5] in the tower environment, "*HMI Interaction modes for approach control*" [6], *HAAWAII* [7] [8], and various further collaborations such as ontology comparisons for ATC with MITRE [9] and NASA [10]. This section describes the general structure of the ontology and its elements.

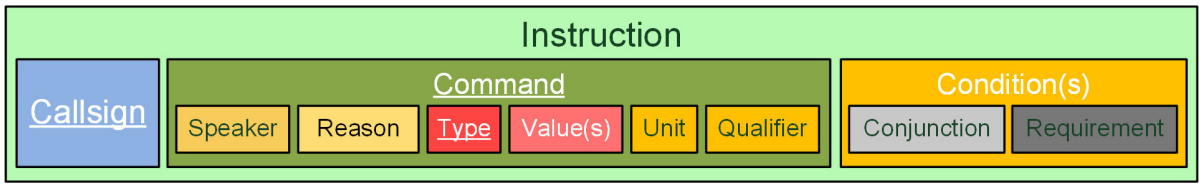

Figure 2-1 Block diagram of European ontology elements

Figure 2-1 illustrates all defined conceptual elements of the ontology, which are deemed relevant for voice communications between air traffic controllers and pilots. The specification at its core classifies that an ATC communication always contains one or multiple *Instruction(s)*. An *Instruction* subsequently is described by the following elements:

- **Callsign** – The element contains the aircraft callsign with airline designator and identification code all in capital letters, e.g., "DLH3ER", "AUA454", or "OENKF". In cases, where a callsign cannot be identified, i.e. was not said, this element has to be set to the special value "NO_CALLSIGN".
- **Command** – An abstract high-level element described by the elements it contains (see below).
- **Condition** – An abstract high-level element described by the elements it contains (see below).

As mentioned above the high-level element "*Command*" is described through the elements it contains. The contained elements are the following:

- **Type** – By definition each *Command* always has to be defined by a (Main) *Type* in capital letters, e.g., DESCEND, CLIMB, REDUCE. Some types are furthermore split into *Type (Main)* and *Type (Sub)*, e.g. TAXI VIA (main: TAXI, sub: VIA), TAXI TO (main: TAXI, sub: TO), MAINTAIN SPEED (main: MAINTAIN, sub: SPEED), MAINTAIN ALTITUDE (main: MAINTAIN, sub: ALTITUDE). In cases where no type at all can be identified for a *Callsign* the special

value "NO_CONCEPT" has to be set as value. The complete list of allowed Type (Main) and Type (Sub) follows in chapter 3.

- **Value(s)** - Depending on the element *Type*, no, one or multiple *Value(s)* can follow, e.g., 100, RWY23, MOBSA (for details on allowed *Value(s)* see the respective *Type* definition in chapter 3).
- **Unit –** Depending on the element *Type* and the presence of a discrete *Value* a unit might be possible, e.g., FL (Flight level) or kt (knots) (for details on allowed *Units* see the respective *Type* definition in chapter 3).
- **Qualifier** – Depending on the element *Type* one or more *Qualifiers* can follow, e.g., for the *Type* HEADING the *Qualifier* LEFT or RIGHT (for details on allowed *Qualifiers* see the respective *Type* definition in chapter 3).
- **Speaker** – The *Speaker* element can have the value "ATCO" or "PILOT". If the element is not set, the ontology defines "ATCO" as the implicit default.
- **Reason** – The element *Reason* only applies if the *Speaker* element is set to "PILOT". In this case the value can be empty, "REQUEST" or "REPORTING".

As mentioned above the high-level element "*Condition"* is described through the elements it contains. The contained elements are the following:

- **Conjunction** – Depending on the semantics of the *Condition*, one of the four *Conjunctions* UNTIL, WHEN, IF, AFTER must be used if the *Requirement* is not a Conditional-Keyword.
- **Requirement** – The Requirement can be a Command that needs to be fulfilled, or a Conditional Keyword.

The following tables showcases, how a given sequence of words should be represented in the above described ontology.

Word Sequence: speed bird six nine six victor keep speed one eight zero knots until five miles final									
Instruction									
Callsign	**Command**							**Condition(s)**	
	Speaker	**Reason**	**Type**		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
			Main	Sub					
BAW696V			MAINTAIN	SPEED	180	kt		UNTIL	5 NM FINAL

The following utterance contains two different instructions for the same callsign.

Word Sequence: lufthansa four nine eight taxi to alfa four eight via lima and november seven									
Instruction									
Callsign	**Command**							**Condition(s)**	
	Speaker	**Reason**	**Type**		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
			Main	Sub					
DLH498			TAXI	TO	A4				
DLH498			TAXI	VIA	L N7				

3 Commands of the Annotation Ontology

Chapter 2 outlined the general structure and elements of the ontology with type, value(s), unit, qualifier, and so on. Building up on that, this chapter describes the allowed content of the individual elements within *Command* and *Condition*. In general, the ontology distinguishes between the following different *Command* type classes:

1. Vertical types,
2. Direction changing types,
3. Vertical speed types,
4. Speed types,
5. Cleared and Cancel types,
6. Frequency changing and Squawk types,
7. Information types,
8. Airborne holding types,
9. VFR clearance types,
10. Reporting types,
11. Movement progress rules types,
12. Taxi types,
13. Pushback and start-up types,
14. Ground hold types, and
15. Miscellaneous types.

These *Command* type classes contain roughly 90 different (Main) types. They are listed and maintained in the tab "*Commands*" (first page) of the public domain table "OntologyATCCommands" with a date indicating the version saved as PDF next to this report. The definition always starts with the "*Type (Main)*". For each "*Type (Main)*" the table shows the allowed content of the other ontology elements, namely *Type (Sub), Value(s), Unit, Qualifier(s)* and *Condition(s)*.

The following important schema applies to this table:

- Words printed in **bold** are direct values (terminals) of the respective ontology element, e.g., **ALTITUDE** or **MAINTAIN** for *Type (Main)*.
- Non-bold words are parameters (non-terminals), i.e., these are not direct values of the respective ontology element (e.g., altitude or flight level in column *Value(s)*). The actual content of non-terminals is described in the tab "*Parameters*" (second page) of the public domain table "OntologyATCCommands" as described above.
- Cells marked in orange contain optional elements, e.g., **BETWEEN** in *Type (Sub)*.

4 General Rules for ATC Command Annotation

This chapter contains some "best practices" about the transformation from transcriptions to annotations.

4.1 Multiple commands in one utterance

The usual case is that more than one command is given in one transmission/utterance. Our ontology allows an arbitrary number of commands in one utterance. We add the callsign to the command annotation each time. This makes processing easier, if different callsigns are provided in an utterance ("break break" case).

Word Sequence: condor one five three turn left heading zero nine zero descend below flight level eight zero

Instruction

Callsign	Command							Condition(s)	
	Speaker	Reason	Type		Value(s)	Unit	Qualifier	Conjunction	Requirement
			Main	Sub					
CFG153			HEADING		090		LEFT		
CFG153			DESCEND		80	FL	BELOW		

The callsign is repeated in each instruction even if uttered only once.

Word Sequence: austrian five seven eight x-ray descend altitude nine thousand QNH one zero one eight

Instruction

Callsign	Command							Condition(s)	
	Speaker	Reason	Type		Value(s)	Unit	Qualifier	Conjunction	Requirement
			Main	Sub					
AUA578X			DESCEND		9000	none			
AUA578X			INFORMATION	QNH	1018				

The unit of the altitude is not said. Therefore, we use *none*. The INFORMATION QNH has by definition no unit in the ontology. Therefore, no "none" is needed (and also not allowed).

Word Sequence: hansa nine one six delta to avoid holding for sequencing reduce further two two zero knots turn right heading three six zero

Instruction

Callsign	Command							Condition(s)	
	Speaker	Reason	Type		Value(s)	Unit	Qualifier	Conjunction	Requirement
			Main	Sub					
DLH916D			REDUCE		220	kt			
DLH916D			HEADING		360		RIGHT		

Even if only "hansa" was uttered, we map to "DLH".

Word Sequence: affirm three thousand feet

Instruction

Callsign	Command							Condition(s)	
	Speaker	Reason	Type		Value(s)	Unit	Qualifier	Conjunction	Requirement
			Main	Sub					
NO_CALLSIGN			AFFIRM						
NO_CALLSIGN			ALTITUDE		3000	ft			

If the callsign has not been uttered, we use "NO_CALLSIGN" for each command in the utterance.

Word Sequence: croatia three two zero thanks									
Instruction									
Callsign	**Command**							**Condition(s)**	
	Speaker	**Reason**	**Type**		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
			Main	Sub					
CTN320			NO_CONCEPT						

If there is no ATC concept in the utterance that can be mapped to a command type of the ontology, we use "*NO_CONCEPT*" for the type field.

4.2 Guessing about utterance meanings

If the controller/pilot utters inconsistent information (in transcription), the annotation should be as "honest" as possible as shown in the example below. Let's assume for the next example, that the correct frequency of KAUNAS_APPROACH is 124.200 and that PALANGA_APPROACH has the frequency 124.300. We cannot decide, which of the two information is wrong (either the controller should have uttered "palanga approach" or "decimal two"). Thus, we annotate as it was said. This gives the chance to count inconsistencies, which can be corrected by downstream processes. The frequencies can be checked at https://skyvector.com.

Word Sequence: wizz air seven zero one contact kaunas approach one two four decimal three bye							
Instruction							
Callsign	**Command**					**Condition(s)**	
	Type		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
	Main	Sub					
WZZ701	CONTACT		KAUNAS_APPROACH				
WZZ701	CONTACT_FREQUENCY		124.300				
WZZ701	FAREWELL						

4.3 Usage of runway name in commands

If the controller/pilot utters a runway (such as "runway zero five right") anywhere in a logically connected part of an utterance, it shall be used for ALL command types that require a runway. Hence, runway "three four" if defined as "RW34" is used for LINEUP and CLEARED TAKEOFF even if it was only uttered once.

Word Sequence: german air force six one two wind is three two zero degrees three knots cleared to lineup and takeoff runway three four							
Instruction							
Callsign	**Command**					**Condition(s)**	
	Type		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
	Main	Sub					
GAF612	INFORMATION	WINDDIRECTION	230				
GAF612	INFORMATION	WINDSPEED	3	kt			
GAF612	LINEUP		RW34				
GAF612	CLEARED	TAKEOFF	RW34				

For transitions, STARs, and SIDs it is recommended to use the format "SISAL_2N" with five upper-case letters, an underscore, a number, and an upper-case letter if reasonable. Of course, this must match with the AIP/database value style. The command type "INFORMATION ACTIVE_RWY" will only be added to an annotation, if the runway in a logical part of an utterance is not used for any other command of the same utterance (CLEARED/CANCEL/ABORT xxx; CONTINUE APPROACH, INTERCEPT_xxx, GO_AROUND, BACKTRACK, LINEUP/LINEUP_BEHIND, VACATE, CROSS,

HOLD_SHORT, EXPECT RUNWAY, INFORMATION Approach_Type, JOIN_TRAFFIC_CIRCUIT, CONFIRM_ACCEPT) except the controller says "runway in use".

Word Sequence: bel avia eight nine two five debrecen tower good day startup and pushback is approved you are cleared to destination uniform mike mike sierra from runway zero five right follow perit five delta initially climb to seven thousand feet squawk three one zero four

Instruction

Callsign	Command					Condition(s)	
	Type		Value(s)	Unit	Qualifier	Conjunction	Requirement
	Main	Sub					
BRU8925	STATION		DEBRECEN_TOWER				
BRU8925	STARTUP						
BRU8925	PUSHBACK						
BRU8925	CLEARED	TO	UMMS				
BRU8925	INFORMATION	ACTIVE_RWY	RW05R				
BRU8925	CLEARED	VIA	PERIT_5D				
BRU8925	CLIMB		7000	ft			
BRU8925	SQUAWK		3104				

Word Sequence: nine alfa juliett india mike budapest tower roger enter via sorok point proceed to highway runway in use one three right report over highway point

Instruction

Callsign	Command					Condition(s)	
	Type		Value(s)	Unit	Qualifier	Conjunction	Requirement
	Main	Sub					
9AJIM	STATION		BUDAPEST_TOWER				
9AJIM	ENTER_CTR	VIA	SOROK				
9AJIM	DIRECT_TO		HIGHWAY				
9AJIM	INFORMATION	ACTIVE_RWY	RW13R				
9AJIM	REPORT	PASSING	HIGHWAY				

The controller might say "lufthansa one two three cleared to land runway one five left break break speed bird four five six cleared for takeoff runway one five right". In this example "break break" divides two logical parts of an utterance. Hence, everything until "break break" uses "15L", everything after "15R". However, logical parts may also exist for clearances to just one callsign: "lufthansa one two three hold short of runway three four cleared for takeoff runway one five". If a runway is used as a value for a certain command type as it is uttered directly before or after, it must be used: "DLH123 HOLD_SHORT RW34, DLH123 CLEARED TAKEOFF RW15".

4.4 Speaking to more than one callsign (break, break)

Sometimes the controller speaks to more than one aircraft in one utterance as sketched above. The controller should use "break break" in these cases, but independent from correct or wrong phraseology used, we repeat each of the callsigns.

Word Sequence: CSA two five one descend to altitude four thousand feet QNH one zero one nine cleared ILS approach runway two three break break air silesia seven three one turn right heading two zero zero cleared ILS approach runway one eight left

Instruction

Callsign	Command							Condition(s)	
	Speaker	Reason	Type		Value(s)	Unit	Qualifier	Conjunction	Requirement
			Main	Sub					
CSA251			DESCEND		4000	ft			
CSA251			INFORMATION	QNH	1019				
CSA251			CLEARED	ILS	RW23				
SUA731			HEADING		200		RIGHT		
SUA731			CLEARED	ILS	RW18L				

Word Sequence: lufthansa six yankee uniform bye bye croatia three two zero standby break break sky travel one two eight seven stop descend altitude seven thousand feet immediately stop descend seven thousand feet reduce speed two three zero knots or below

Instruction

Callsign	Command					Condition(s)	
	Type		Value(s)	Unit	Qualifier	Conjunction	Requirement
	Main	Sub					
DLH6YU	FAREWELL						
CTN320	CALL_YOU_BACK						
TVS1287	STOP_DESCEND		7000	ft			
TVS1287	STOP_DESCEND		7000	ft			
TVS1287	REDUCE		230	kt	OR_BELOW		

4.5 More than one speaker per utterance

If audio files with ATC communication are not perfectly split, a single file can contain spoken words from multiple persons, e.g., from an ATCO and a pilot or even from ATCO(s) and pilot(s). If the rules for transcription are followed, the rules for annotation are easy to determine with the following example.

The talking sequence here is pilot 1, then pilot 2, then ATCO, then pilot 2 again and finally ATCO again. Next to the used annotation field SPEAKER (it is just the speaker type in annotation), the field REASON is used (see *REQUEST*; not used here is the REASON *REPORTING*).

Word Sequence:

Pilot1: good morning condor three eight zero

Pilot2: lufthansa one two request descend to flight level three one zero

ATCO: three eight zero hello break break hansa one two descend flight level three one zero

Pilot2: thank you descending three one zero lufthansa one two

ATCO: condor three eight zero now reduce speed three hundred knots

Instruction

Callsign	Command							Condition(s)	
	Speaker	Reason	Type		Value(s)	Unit	Qualifier	Conjunction	Requirement
			Main	Sub					
CFG380	PILOT		GREETING						
DLH12	PILOT	REQUEST	DESCEND		310	FL			
CFG380	ATCO		GREETING						
DLH12	ATCO		DESCEND		310	FL			
DLH12	PILOT		DESCEND		310	FL			
CFG380	ATCO		REDUCE		300	kt			

4.6 Conditional Clearances

A conditional clearance issued by a controller, only becomes effective depending on the satisfaction of a requirement. The conditional clearance contains one of the following keywords: UNTIL, WHEN, AFTER or IF.

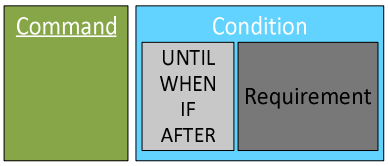

Figure 4-1 Clearance with Command, Conjunction keyword and Requirement

4.6.1 Conjunction keyword "UNTIL"

UNTIL is used to specify that the command is only effective UNTIL the following requirement is NOT fulfilled. It can, e.g., be used with UNTIL PASSING, UNTIL waypoint, UNTIL time, etc. The vertical speed clearance starts now and ends when flight level 70 is reached. Therefore, UNTIL is used in the example below.

Word Sequence: finn air five kilo alfa descend with fifteen hundred feet per minute or greater until passing flight level seven zero							
Instruction							
Callsign	**Command**					**Condition(s)**	
	Type		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
	Main	Sub					
FIN5KA	RATE_OF_DESCENT		1500	ft_min	OR_GREATER	UNTIL	PASSING ALTITUDE 70 FL

The next command contains a conditional clearance with UNTIL semantics, which is valid for the vertical rate command.

Word Sequence: lufthansa eight november descend flight level one four zero at two thousand five hundred feet per minute or greater until passing flight level one six zero							
Instruction							
Callsign	**Command**					**Condition(s)**	
	Type		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
	Main	Sub					
DLH8N	DESCEND		140	FL			
DLH8N	RATE_OF_DESCENT		2500	ft_min	OR_GREATER	UNTIL	PASSING 160 FL

This is a conditional clearance. The aircraft should fly with 160 or greater until five miles before DME. Then a reduction may start, but not earlier. No unit (e.g. knots) is specified. Therefore, we use "none".

Word Sequence: ryan air nine whiskey bravo one sixty or greater until five DME							
Instruction							
Callsign	**Command**					**Condition(s)**	
	Type		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
	Main	Sub					
RYR9WB	SPEED		160	none	OR_GREATER	UNTIL	5 none DME

4.6.2 Conjunction keyword “WHEN”

WHEN is used to specify that the command gets effective WHEN the following requirement is fulfilled. The condition is in the middle of the clearance, but in command annotation the condition is always added at the end (see example below).

Word Sequence: lufthansa four delta when you pass flight level six five cleared direct mike india charlie							
Instruction							
Callsign	**Command**					**Condition(s)**	
	Type		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
	Main	Sub					
DLH4D	DIRECT_TO		MIC		none	WHEN	PASSING ALTITUDE 65 FL

The condition is said before the command, but the condition is always the last element of the ontology (here in the case of the condition with passing an altitude). The condition is valid for both CONTACT commands and must be annotated twice, i.e., once for each CONTACT(_FREQUENCY) command.

Word Sequence: wizz air three nine one runway one three right line up cleared for takeoff wind is one four zero degrees and five knots when passing one thousand five hundred feet contact approach on one two nine decimal seven							
Instruction							
Callsign	**Command**					**Condition(s)**	
	Type		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
	Main	Sub					
WZZ391	LINEUP		13R				
WZZ391	CLEARED	TAKEOFF	none				
WZZ391	INFORMATION	WIND DIRECTION	140				
WZZ391	INFORMATION	WINDSPEED	5	kt			
WZZ391	CONTACT		BUDAPEST_ APPROACH			WHEN	PASSING 1500 ft
WZZ391	CONTACT_ FREQUENCY		129.700			WHEN	PASSING 1500 ft

This is a conditional clearance. The aircraft should fly with 160 or greater until five miles before DME. Then a reduction may start, but not earlier. “on conversion” means that the speed should be 280 or less, when the speed is changed (due to altitude and speed) from Mach numbers to calibrated air speed (CAS) or vice versa. The pilot will not change the speed to 280 knots now, but maybe in 30 minutes or even later. Therefore, the condition SPEED_CONVERSION is necessary.

Word Sequence: ice air two zero five on conversion two eight zero knots indicated or less							
Instruction							
Callsign	**Command**					**Condition(s)**	
	Type		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
	Main	Sub					
ICE205	SPEED		280	none	OR_LESS	WHEN	SPEED_CONVERSION

4.6.3 Conjunction keyword “IF”

IF is used to specify that the command gets only effective IF the following requirement is fulfilled (now) or it will never get effective.

The command did not contain a clearance to intercept the localizer. This will depend on ability of ABP712 to turn left on heading 270. It is just information that the turn is for future interception of localizer of runway 24.

Word Sequence: B air seven one two if able turn left heading two seven zero to intercept localizer runway two four									
Instruction									
Callsign	**Command**							**Condition(s)**	
	Speaker	**Reason**	**Type**		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
			Main	Sub					
ABP712			HEADING		270		LEFT	IF	ABLE

Figure 4-2 shows that most keywords of the condition (e.g., PASSING) can be used both with UNTIL and with WHEN. Different to PASSING, we use REACHING if the state remains for some time. [1]

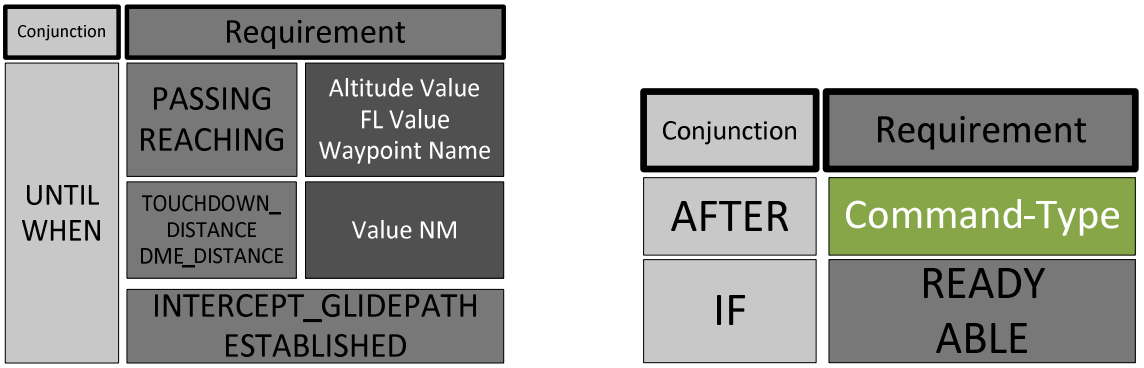

Figure 4-2 Elemens of a condition

4.6.4 Conjunction keyword “AFTER”

AFTER is nearly the same as WHEN. The difference between AFTER and WHEN is the following:

- AFTER means a procedure or manoeuvre (e.g., TOUCH_GO) must be completed;
- WHEN means the position, level or another aircraft parameter must get a defined value.

AFTER is used if the requirement part in Figure 4-2 is a command type.

We can assume that a REDUCE is given before to the AUA133, but that knowledge is irrelevant for the annotation in the below example.

Word Sequence: austrian one three three when completed reduce descend flight level nine zero									
Instruction									
Callsign	**Command**							**Condition(s)**	
	Speaker	**Reason**	**Type**		**Value(s)**	**Unit**	**Qualifier**	**Conjunction**	**Requirement**
			Main	Sub					
AUA133			DESCEND		90	FL		AFTER	REDUCE

[1] This is only a subset of the possible conditions.

The current frequency should be contacted again AFTER finishing the de-icing process (even if the word 'when' is used, it is not WHEN READY DEICE, because this would mark the start of the de-icing).

Word Sequence: delta six seven five runway three four left de ice pad taxi via bravo and when you are done de icing back with me

Instruction

Callsign	Command							Condition(s)	
	Speaker	Reason	Type		Value(s)	Unit	Qualifier	Conjunction	Requirement
			Main	Sub					
DAL675			TAXI	TO	ICE_34L				
DAL675			TAXI	VIA	B				
DAL675			CONTACT	ME				AFTER	DEICE

4.6.5 Complex conditional Clearances

4.6.5.1 Utterance with multiple conditions

The condition "2000 ft" is related to the MISS_APP_PROC command and not to the HEADING command. The controller says "until". Nevertheless, we annotate with "WHEN". Then, we use AFTER because CLEARED TOUCH_GO is a command type and not just a value as used for WHEN.

Word Sequence:

wizz air seven zero one cleared touch and go for runway three four wind is two nine zero degrees six knots after touch and go please continue on runway heading until passing two thousand feet and then join to the missed approach procedure

Instruction

Callsign	Command					Condition(s)	
	Type		Value(s)	Unit	Qualifier	Conjunction	Requirement
	Main	Sub					
WZZ701	CLEARED	TOUCH_GO	RW34				
WZZ701	INFORMATION	WINDDIRECTION	290				
WZZ701	INFORMATION	WINDSPEED	6	kt			
WZZ701	HEADING		RUNWAY_DIR			AFTER	CLEARED TOUCH_GO
WZZ701	CLEARED	MISS_APP_PROC				WHEN	PASSING 2000 ft

4.6.5.2 One Command with multiple different conditions

We first have a WHEN-part. Reduction to 160 knots is not started now, but later, when aircraft is established on localizer. New speed is not kept forever (until next command), but only until distance to touchdown is four miles (outer marker passed).

Word Sequence:

CSA seven zero seven hotel when established localizer reduce speed one six zero knots to maintain until four miles final

Instruction

Callsign	Command							Condition(s)	
	Speaker	Reason	Type		Value(s)	Unit	Qualifier	Conjunction	Requirement
			Main	Sub					
CSA707H			REDUCE		160	kt		WHEN	ESTABLISHED
								UNTIL	4 NM FINAL

4.7 Best Practice Examples of Various Command Types

The Appendix provides many different practical examples – including edge cases – for transforming a transcription into an ATC-ontology conform annotation organized by command types. Refer to the beginning of the Appendix for more detailed information.

5 Machine-Readable Format for Exchange of Ontology Information

This chapter describes how the ontology from chapter 2 with the corresponding allowed values for the different elements from chapter 3 can be implemented in a machine-readable format. The used format is based on JSON and can be transmitted via different protocols.

5.1 Definition of Key-Value-Pairs

The following table contains the definition of the key-value-pairs which are allowed within the JSON format for ontology instructions:

Main elements		
Key	**Value**	**Description**
filename	String	String in the format "YYYY-MM-DD__hh-mm-ss-ms.wav" to identify where the audio information is stored that the following output is based on.
sender	String	String uniquely identifying who is transmitting the information.
endpoint (optional)	String	"true" or "false" indicating if end of a transmission is reached. Allows intermediate output for transmissions. If not set, "true" is assumed.
mainlanguage (optional)	String	Identification of the main language within the utterance with a two-letter abbreviation (ISO-639-1-code). This can be "de" for "German". If not set, "en" for "English" is assumed. Object - {"v": "**LANG_STRING**", "p": "**PLAUSIB_V***"}
abstraction_layer_word_sequence	String or Object	String (**WORD_STRING**) - Containing the recognized sequence of words Object - {"v": "**WORD_STRING**", "p": "**PLAUSIB_V***"}
abstraction_layer_ontology_command	Array of Object(s)	**Array contains objects with own Key-value pairs:** speaker, commands
abstraction_layer_classific_seq	String	String with word classifications for "abstraction_layer_word_sequence" with respect to "abstraction_layer_ontology_command", e.g. csgn, valu, unkn, etc.
abstraction_layer_cmdNumber_seq	String	String containing integer positive numbers plus 0 and -1. They specify for each word, for which command a word is used. -1 means not used at all. Counting starts at 0, see the examples in section 5.5.

Abstraction layer ontology command – Nested elements		
Key	**Value**	**Description**
speaker (optional)	String	Identification of speaker as "ATCO" or "PILOT". If not set, "ATCO" should be assumed. If different speakers appear with the same role, add number for distinction, e.g., "ATCO1", "ATCO2", … Object - {"v": "**SPEK_STRING**", "p": "**PLAUSIB_V***"}
commands	Array of Object(s)	**Array contains objects of instructions with own Key-value pairs:** csgn (callsign), reas (reason), type (type main), sndT (type second), valu (value), vare (value rest), qual (qualifier), unit (unit), cond (condition)

Commands – Nested elements		
Key	**Value**	**Description**
csgn	String or Object	String (**CSGN_STRING**) – airline designator + identification ("DLH3ER") or "NO_CALLSIGN" Object - {"v": "**CSGN_STRING**", "p": "**PLAUSIB_V***"}

reas (optional)	String	String for reason only applies if the speaker is set to “PILOT”. In this case the value can be empty, “REQ” (Request) or “REP” (Reporting).
type	String or Object	String (**TYPE_STRING**) – value from column “Type (Main)” (see chapter 3) or “NO_CONCEPT” Object - {“v”: “**TYPE_STRING**”, “p”: “**PLAUSIB_V***”}
sndT (optional)	String or Object	String (**SNDT_STRING**) – value from column “Type (Sub)”. Depends on “Type (Main)” (see chapter 3) Object - {“v”: “**SNDT_STRING**”, “p”: “**PLAUSIB_V***”}
valu (optional)	String or Object	String (**VALU_STRING**) – value from column “Value(s)”. Depends on “Type (Main)” (see chapter 3) Object - {“v”: “**VALU_STRING**”, “p”: “**PLAUSIB_V***”}
vare (optional)	String or Object	String (**VARE_STRING**) – same as **VALU_STRING** for all values beyond the first (space separated) Object - {“v”: “**VARE_STRING**”, “p”: “**PLAUSIB_V***”}
qual (optional)	String or Object	String (**QUAL_STRING**) – value from column “Qualifier(s)”. Depends on “Type (Main)” (see chapter 3) Object - {“v”: “**QUAL _STRING**”, “p”: “**PLAUSIB_V***”}
unit (optional)	String or Object	String (**UNIT_STRING**) – value from column “Unit”. Depends on “Values(s)” (see chapter 3) Object - {“v”: “**UNIT _STRING**”, “p”: “**PLAUSIB_V***”}
cond (optional)	String or Object	String (**COND_STRING**) – value from column “Condition(s)”. Depends on “Type (Main)” (see chapter 3) Object - {“v”: “**COND_STRING**”, “p”: “**PLAUSIB_V***”}
*__PLAUSIB_V__ - A number between 0 and 1. Describes how plausible the defined ontology elements seem to be with respect to the text in “abstraction_layer_word_sequence”.		

5.2 Simple Examples

The following examples demonstrate how to use the above defined key-value pairs of the JSON format for "simple" use cases which only use the minimum of required key-value pairs for the transmission of information.

The first example shows the transmission of information in cases where no instruction can be identified in a transmission. For more information see the bullet point list below the example.

```
{
  "filename": "2016-06-15__10-28-22-01.wav",
  "sender": "ASRU.pc1.instance1",
  "abstraction_layer_word_sequence": "hello world",
  "abstraction_layer_ontology_command": [
    {
      "commands": [
        {"csgn": "NO_CALLSIGN", "type": "NO_CONCEPT"}
    ]}
  ]
}
```

- `"filename"` – Describes that a wave-file with the name `"2016-06-15__10-28-22-01.wav"` contains the audio data for this transmission.
- `"sender"` – A process or machine identifying as `"ASRU.pc1.instance1"` send this information.
- `"abstraction_layer_word_sequence"` – The recognized sequence of words for the audio data of this transmission is `"hello world"`.
- `"abstraction_layer_ontology_command"` – Holds an array of objects with all identified instructions from the transmission. In the shown example no "real" instruction is identified, therefor just a single instruction is contained in the `"commands"` array with the value for `"csgn"` set to `"NO_CALLSIGN"` and the value for `"type"` set to `"NO_CONCEPT"`.

The next example shows the transmission of one identified "CLIMB" instruction. The bullet point list below the example provides further information.

```
{
  "filename": "2016-06-15__10-29-19-03.wav",
  "sender": "ASRU.pc1.instance1",
  "abstraction_layer_word_sequence": "lufthansa three echo romeo climb to seven
thousand",
  "abstraction_layer_ontology_command": [
    {
      "commands": [
        {"csgn": "DLH3ER", "type": "CLIMB", "valu": "7000", "unit": "none"}
    ]}
  ]
}
```

- `"filename"` – Describes that a wave-file with the name `"2016-06-15__10-29-19-03.wav"` contains the audio data for this transmission.
- `"sender"` – A process or machine identifying as `"ASRU.pc1.instance1"` send this information.
- `"abstraction_layer_word_sequence"` – The recognized sequence of words for the audio data of this transmission is `"lufthansa three echo romeo climb to seven thousand"`.
- `"abstraction_layer_ontology_command"` – Holds an array of objects with all identified instructions from the transmission. In the shown example one `"CLIMB"` instruction is identified with a value of `"7000"`. Even so it is clear from the value that the presented number must refer to an altitude the field `"unit"` contains `"none"` since no unit was explicitly mentioned within the transmission.

The next example shows the transmission of three instructions for two different callsigns. The bullet point list below the example provides further information.

```
{
  "filename": "2016-06-15__10-28-54-02.wav",
  "sender": "ASRU.pc1.instance1",
  "abstraction_layer_word_sequence": "austrian four five four descend altitude seven
thousand feet QNH is one zero zero three break break KLM two alfa delta descend one
one zero",
  "abstraction_layer_ontology_command": [
    {
      "commands": [
        {"csgn": "AUA454", "type": "DESCEND", "valu": "7000", "unit": "ft"},
        {"csgn": "AUA454", "type": "INFORMATION", "sndT": "QNH", "valu": "1003"},
        {"csgn": "KLM2AD", "type": "DESCEND, "valu": "110", "unit": "none"}
    ]}
  ]
}
```

- `"filename"` – Describes that a wave-file with the name `"2016-06-15__10-28-54-02.wav"` contains the audio data for this transmission.
- `"sender"` – A process or machine identifying as `"ASRU.pc1.instance1"` send this information.
- `"abstraction_layer_word_sequence"` – The recognized sequence of words for the audio data of this transmission is `"austrian four five four descend altitude seven thousand feet QNH is one zero zero three break break KLM two alfa delta descend one one zero"`.
- `"abstraction_layer_ontology_command"` – Holds an array of objects with all identified instructions from the transmission. In the shown example three instructions are extracted for two different callsigns. For the first `"DESCEND"` instruction the `"unit"` `"ft"` is presented. For the second `"DESCEND"` the `"unit"` is set to `"none"` because it is not mentioned in the transmission. The second instruction `"INFORMATION"` has the subtype `"QNH"`.

The last rudimentary example just shows the annotation of a pilot utterance without its transcription. The bullet point list below the example provides further information.

```
{
  "filename": "2025-06-16__21-07-33-69_P.wav",
  "abstraction_layer_ontology_command": [
    {
      "speaker": "PILOT",
      "commands": [
        {"csgn": "SWR2YB", "type": "REDUCE", "valu": "160", "unit": "kt"}
    ]}
  ]
}
```

- `"filename"` – Describes that a wave-file with the name `"2025-06-16__21-07-33-69_P.wav"` contains the audio data for this transmission. The `"_P"` indicates that it comes from a pilot.
- `"abstraction_layer_ontology_command"` – Holds the `"speaker"` value, which is `"PILOT"` here, and an array of objects with the identified `"REDUCE"` instruction from the transmission.

5.3 Complex Examples with Plausibility

This example shows how plausibility values between 0 and 1 can be used for different elements of the JSON format. For more information see the bullet point list below the example.

```
{
  "filename": "2016-06-15__10-28-54-02.wav",
  "sender": "ASRU.pc1.instance1",
  "mainlanguage": [{"v": "en", "p": "0.9"}],
  "abstraction_layer_word_sequence": {"v": "austrian four five four descend altitude
seven thousand feet QNH is one zero zero three break break KLM two alfa delta descend
one one zero", "p":  "0.8"},
  "abstraction_layer_ontology_command": [
    {
      "speaker": [{"v": "PILOT", "p": "0.8"},{"v": "ATCO", "p": "0.15"}],
      "commands": [
        {"csgn": "AUA454", "type": {"v": "DESCEND", "p": "0.6"}, "valu": "7000",
         "unit": "ft"},
        {"csgn": "AUA454", "type": "INFORMATION", "sndT": "QNH",
         "valu": {"v": "1003", "p": "0.7"}},
        {"csgn": {"v": "KLM2AD", "p": "0.5"}, "type": "DESCEND, "valu": "110",
         "unit": "none"}
    ]}
  ]
}
```

- `"mainlanguage"` – The majority of words seems to be in English with 90% plausibility indicated by `"en"`.
- `"abstraction_layer_word_sequence"` – Instead of just the recognized sequence of words for the audio data this key now holds an object with two nested keys. The key `"v"` shows the concrete value so the recognized sequence of words and the key `"p"` assigns a plausibility value to the concrete value in this case `"0.8"`.
- `"abstraction_layer_ontology_command"` – Holds an array of objects with all identified instructions from the transmission. In the example above there is the optional `"speaker"` key with value `"Pilot"` with a plausibility of 80% and `"ATCO"` with a plausibility of 15%. In the shown example different keys within the `"commands"` array contain objects instead of direct strings. The objects always contain under `"v"` the concrete value of the respective ontology element and have under `"p"` an assigned plausibility value for the respective element. Every element within the `"commands"` array can have an individual plausibility value assigned to it.

5.4 Complex Examples with Intermediate Output for Transmissions

This example shows how one ATC utterance can be transmitted with intermediate outputs. This allows the transmission of information even before an utterance is complete. The example below contains the additional key `"endpoint"` with the value `"false"` or `"true"`. The purpose of this key is to indicate if the end of an utterance has been reached. The three outputs below show how a transmission for one single utterance changes over time. In the first transmission `"endpoint"` is set to `"false"` and a short word sequence is recognized which leads to only an identified callsign within the `"commands"` array.

```
{
  "filename": "2016-06-15__10-28-54-02.wav",
  "endpoint": "false",
  "sender": "ASRU.pc1.instance1",
  "abstraction_layer_word_sequence": "austrian four five four descend altitude",
  "abstraction_layer_ontology_command": [
    {
      "commands": [
        {"csgn": "AUA454", "type": "NO_CONCEPT"}
    ]}
  ]
}
```

The second transmission still has the `"endpoint"` set to `"false"`, because the utterance is not finished yet, but a longer sequence of words is recognized which results in two extracted instructions within the `"commands"` array.

```
{
  "filename": "2016-06-15__10-28-54-02.wav",
  "endpoint": "false",
  "sender": "ASRU.pc1.instance1",
  "abstraction_layer_word_sequence": "austrian four five four descend altitude seven
thousand feet QNH is one zero zero three",
  "abstraction_layer_ontology_command": [
    {
      "commands": [
        {"csgn": "AUA454", "type": "DESCEND", "valu": "7000", "unit": "ft"},
        {"csgn": "AUA454", "type": "INFORMATION", "sndT": "QNH", "valu": "1003"}
    ]}
  ]
}
```

Finally, the third and last transmission sets the `"endpoint"` to `"true"`, which indicates that no more transmissions will be send for this ATC utterance. This also means that the recognized sequence of words is final and will not change anymore, which in this case results in a total of three instructions in the `"commands"` array.

```
{
  "filename": "2016-06-15__10-28-54-02.wav",
  "endpoint": "true",
  "sender": "ASRU.pc1.instance1",
  "abstraction_layer_word_sequence": "austrian four five four descend altitude seven
thousand feet QNH one zero zero three break break KLM two alfa delta descend one one
zero",
  "abstraction_layer_ontology_command": [
    {
      "commands": [
        {"csgn": "AUA454", "type": "DESCEND", "valu": "7000", "unit": "ft"},
        {"csgn": "AUA454", "type": "INFORMATION", "sndT": "QNH", "valu": "1003"},
        {"csgn": "KLM2AD", "type": "DESCEND, "valu": "110", "unit": "none"}
    ]}
```

```
  ]
}
```

5.5 Examples with word classification

This example shows the usage of the keyword `"abstraction_layer_classific_seq"`. The array `"abstraction_layer_classific_seq"` must always contain as many elements as the array `"abstraction_layer_word_sequence"`. If words are not used "unkn" must be provided.

```
{
  "filename": "2016-06-15__10-28-54-02.wav",
  "abstraction_layer_word_sequence":
          "austrian four five four descend altitude five thousand feet",
  "abstraction_layer_classific_seq": "csgn csgn csgn csgn type type valu valu unit",
  "abstraction_layer_cmdNumber_seq": "0 0 0 0 0 0 0 0 0",
  "abstraction_layer_ontology_command": [
    {
      "commands": [
        {"csgn": "AUA454", "type": "NO_CONCEPT", "valu": "5000", "unit": "ft"}
    ]}
  ]
}
```

The same applies for the keyword `"abstraction_layer_cmdNumber_seq"`. Here we have only one command. So, all words are used for the first command (counting starts at 0.)

In the next example "affirm" plus the callsign are used for extracting the first command. "good morning" is used for the second command. "thanks" is used for no classification and "identified" is used for the last command INIT_RESPONSE.

```
{
  "filename": "2016-06-15__10-28-54-02.wav",
  "abstraction_layer_word_sequence":
          "austrian triple five affirm good morning thanks identified",
  "abstraction_layer_classific_seq": "csgn csgn csgn type type type unkn type",
  "abstraction_layer_cmdNumber_seq": "0 0 0 0 1 1 -1 2",
  "abstraction_layer_ontology_command": [
    {
      "commands": [
        {"csgn": "AUA555", "type": "AFFIRM"},
        {"csgn": "AUA555", "type": "GREETING"},
        {"csgn": "AUA555", "type": "INIT_RESPONSE"}
    ]}
  ]
}
```

References

[1] H. Helmke et al., D3.1 Transcription and Annotation Handbook, version 1, HAAWAII Project, November 5, 2020.

[2] H. Helmke et al., „Ontology for transcription of ATC speech commands of SESAR 2020 solution PJ.16-04," in *IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, London, United Kingdom, 2018.

[3] Kleinert, Matthias and Shetty, Shruthi and Helmke, Hartmut and Ohneiser, Oliver and Wiese, Hanno and Maier, Mathias and Schacht, Susanne and Nigmatulina, Iuliia and Saeed, Seyyed and Sarfjoo, Petr Motlicek, „Apron Controller Support by Integration of Automatic Speech Recognition with an Advanced Surface Movement Guidance and Control System," in *12th SESAR Innovation Days*, Budapest, Hungary, 2022.

[4] Deutsches Zentrum fur Luft und Raumfahrt (DLR), „Virtual/augmented reality applications for tower (SESAR Solution PJ.05-W2-97.1)," [Online]. Available: https://www.remote-tower.eu/wp/project-pj05-w2/solution-97-1/.

[5] Sesar Joint Undertaking, „Industrial research project: digital technologies for tower," SESAR 3 Joint Undertaking, [Online]. Available: https://www.sesarju.eu/projects/DTT.

[6] Europaische Kommission, „PJ.10 W2 Separation Management and Controller Tools," [Online]. Available: https://cordis.europa.eu/programme/id/H2020_SESAR-IR-VLD-WAVE2-10-2019/de.

[7] Deutsches Zentrum für Luft- und Raumfahrt (DLR), „HAAWAII: highly automated air traffic controller workstations with artificial intelligence integration," [Online]. Available: https://www.haawaii.de/wp/.

[8] H. Helmke et al., „Readback error detection by automatic speech recognition to increase ATM safety," in *14th USA/Europe Air Traffic Management Research and Development Seminar (ATM2021)*, Virtual Conference, 2021.

[9] S. Chen, H. Helmke, R. M. Tarakan, O. Ohneiser, H. Kopald und M. Kleinert, „Effects of Language Ontology on Transatlantic Automatic Speech Understanding Research Collaboration in the Air Traffic Management Domain," *Aerospace,* 2023.

[10] H. Helmke, S. S. Clarke, O. Ohneiser, M. Kleinert, S. Shetty und K. Kalyanam, *Natural Language Understanding for Air Traffic Control - How much Information is Lost Over the Atlantic?,* Montréal, Canada: 44th Digital Avionics Systems Conference (DASC), 2025.

Appendix

The following sections contain transcription and annotation examples for controller and pilot utterances organized by command types. For better oversight, they are numbered as succeeding sections beginning with number 6. New sections for further command types will be added and updated with ontology enhancements if applicable. Most of the examples originate from actual controllers or pilots performing radio telephony in an operational environment or in a human-in-the-loop simulation; just a minority of examples contains artificial content to demonstrate some of the command types. It is recommended to use the search function of the document viewing tool if looking for a certain keyword or command type.

The tables in this Appendix each provide the *Word Transcription* following the agreed transcription rules including the ICAO alphabet[2] (first row), *Additional information*, e.g., from which ATC environment/context the example comes from (second row), the *Command Annotation* with callsigns, types, and further ontology elements in a human-readable format (third row), and an *Explanation* why the decision for the command annotation serving the main purpose of the current section has been made (fourth row) with the second and fourth row being optional. Usually, each section starts with a simple utterance of the presented command type. However, then we present a row of difficult cases as they require the guidance on how to annotate – not the simple cases. The utterance examples often contain more than one command as they stem from practical examples. Hence, there might as well be additional information for the further commands of an utterance or for the transcription of an utterance even if they do not serve the main purpose of the current section. For all REPORT commands, there are two additional rows for the *Pilot answer* that can follow the controller's request to report something and an *Explanation* for the latter.

[2] See https://www.icao.int/pages/alphabetradiotelephony.aspx

6 Vertical Commands

Command types DESCEND, CLIMB, and ALTITUDE are three of the main types for flight movement changes in the vertical dimension. ALTITUDE is used if it is not clear from the individual transmission (without considering more context like radar information or previous utterances) whether the aircraft should climb or descend. If for example the addressed aircraft is known to be a departure, it might be clear that the controller will issue a CLIMB. Nevertheless, command annotation should be "ALTITUDE", if the controller does not explicitly say the word "climb" or other words that clearly state that it is a CLIMB command. "FL" or "ft" should follow the value, if the controller said the corresponding words. Otherwise we use "none" to indicate that no unit could be detected in the spoken content

In the following sections, we show examples for most of the vertical command types.

6.1 DESCEND

Word Transcription	**air france three four one six descend to flight level two hundred**
Additional information	By default, we use the verb to indicate flight maneuvers in transcriptions, i.e., we use 'descend' and not the noun 'descent'. Of course, there are exceptions where the noun is definitely meant such as 'rate of descent'.
Command Annotation	AFR3416 DESCEND 200 FL
Explanation	We add the "FL" information, if the controller has spoken it. Altitudes are provided in feet or in flight level. Therefore, it is important for plausibility checks that the unit is also transcribed, if it is spoken.

Word Transcription	**niki four two six charlie descend three thousand feet left to the waypoint nine seven one**
Additional information	If no waypoint would have been said for the turn, it would have been TURN LEFT.
Command Annotation	NLY426C DESCEND 3000 ft NLY426C DIRECT_TO WW971 LEFT
Explanation	

Word Transcription	**finn air seven mike bravo madrid radar contact descend six thousand QNH one zero one eight**
Additional information	
Command Annotation	FIN7MB STATION MADRID FIN7MB INIT_RESPONSE FIN7MB DESCEND 6000 none FIN7MB INFORMATION QNH 1010
Explanation	

Word Transcription	**tom jet five eight quebec roger descend flight level two two zero cross corda two four zero or below**
Additional information	This is an oceanic clearance.
Command Annotation	TOM58KQ DESCEND 220 FL TOM58KQ DRECT_TO CORDA TOM58KQ DESCEND 240 FL OR_BELOW UNTIL CORDA
Explanation	However, we do not use a command type CROSS_WP, but a DESCEND with condition. Hence, there can be multiple DESCEND commands in one utterance. This is to guarantee a certain flight level at a certain waypoint.

Word Transcription	**iberia two six uniform when ready descend flight level three one zero to be leveled by golf india romeo**
Additional information	This is an oceanic clearance.
Command Annotation	IBE26U DESCEND 310 FL WHEN READY UNTIL GIR
Explanation	There are two conditions: 'WHEN READY' and 'UNTIL GIR'.

6.2 CLIMB

Word Transcription	**faxi three two six reykjavik control [NE Icelandic] godan dag [/NE] identified climb flight level two three zero**
Additional information	
Command Annotation	FXI326 STATION REYK_RADAR FXI326 GREETING FXI326 INIT_RESPONSE FXI326 CLIMB 230 FL
Explanation	

Word Transcription	**air portugal three three four confirm your final level break iberia three four five climb flight level three two zero**
Additional information	'break' instead of 'break break' used. We have not special words for all situations. Therefore, the default REPORT_MISCELLANEOUS is used for the first part of the utterance.
Command Annotation	TAP334 REPORT_MISCELLANEOUS IBE345 CLIMB 320 FL
Explanation	

Word Transcription	air portugal four zero four climb flight level three one zero to be leveled by dalin
Additional information	This is an oceanic clearance.
Command Annotation	TAP404 CLIMB 310 FL UNTIL DALIN
Explanation	

Word Transcription	air portugal four zero four climb flight level three one zero to be leveled by ten two four
Additional information	This is an oceanic clearance.
Command Annotation	TAP404 CLIMB 310 FL UNTIL 1024
Explanation	

Word Transcription	euro wings three zero november roger you are cleared for three hundred climb flight level three two zero and direct holstein hotel lima zulu
Additional information	This is an en-route clearance.
Command Annotation	EWG30N AFFIRM EWG30N ALTITUDE 300 none EWG30N CLIMB 320 FL EWG30N DIRECT_TO HLZ none
Explanation	This utterance contains an ALTITUDE command AND a CLIMB command. What the controller probably meant is that a “CLIMB 300 FL” was already cleared sometime earlier, but is now adjusted to “CLIMB 320 FL”. Hence, it would make sense, to discard the command “ALTITUDE 300 none” and just take the final, valid “CLIMB” command. However, this would not reflect what has actually been said. Therefore, two commands regarding altitude should be annotated here. It is up to the following application what to do with those “contradictory” information.

6.3 ALTITUDE

Word Transcription	niki nine two eight echo cleared level two four zero
Additional information	The controller said “niki” although “flyniki” is the official designator,but the command annotation should result in both cases in NLY
Command Annotation	NLY928E ALTITUDE 240 FL
Explanation	It is not clear whether it is a descend or a climb command. Therefore, we choose ALTITUDE. ALTITUDE is the superset of DESCEND and CLIMB command.

Word Transcription	niki nine two eight echo cleared two four zero
Additional information	
Command Annotation	NLY928E ALTITUDE 240 none
Explanation	It is not said whether 240 zero is a flight level or an altitude in feet. Therefore, we add none. Of course, it is clear from the context, but it was not spoken. We want to annotate what is spoken.

Word Transcription	ryan air two eight whiskey mike clear flight level three four zero
Additional information	
Command Annotation	RYR28WM ALTITUDE 340 FL
Explanation	It is not clear whether it is a descend or a climb command. Therefore, we choose ALTITUDE. ALTITUDE is the superset of DESCEND and CLIMB command.

6.4 MAINTAIN ALTITUDE

Word Transcription	three two one maintain flight level one nine zero
Additional information	There is only an Austrian three two one in the air.
Command Annotation	AUA321 MAINTAIN ALTITUDE 190 FL
Explanation	MAINTAIN ALTITUDE might be of course misleading because in many cases it means to maintain a flight level and not an altitude. MAINTAIN PRESENT_ALTITUDE is only used if no vertical value is provided.

Word Transcription	austrian one two three reaching and maintain six thousand feet entering sasal holding
Additional information	
Command Annotation	AUA123 MAINTAIN ALTITUDE 6000 ft AUA123 HOLDING SASAL
Explanation	The word “reaching” is not especially annotated.

6.5 Vertical commands with additional qualifiers

The words ABOVE, BELOW, OR_ABOVE, OR_BELOW could be added.

Word Transcription	hansa one two three so descend four thousand feet or below
Additional information	
Command Annotation	DLH123 DESCEND 4000 ft OR_BELOW
Explanation	

Word Transcription	hansa one two three so descend below four thousand feet
Additional information	
Command Annotation	DLH123 DESCEND 4000 ft BELOW
Explanation	The aircraft should not descend to four thousand feet, but below.

6.6 STOP_CLIMB

Word Transcription	air malta one three zero nine stop climb flight level one one zero
Additional information	
Command Annotation	AMC1309 STOP_CLIMB 110 FL
Explanation	The aircraft might be in a climb to a higher flight level, but receives the command to stop the current climb at a specific flight level.

6.7 STOP_DESCEND

Word Transcription	air france one one three eight stop descend one four zero
Additional information	Flight level is not said
Command Annotation	AFR1138 STOP_DESCEND 140 none
Explanation	

Word Transcription	air france one one three eight stop descend below level one four zero
Additional information	
Command Annotation	AFR1138 STOP_DESCEND 140 FL BELOW
Explanation	Although this command makes no sense in real life, because a pilot does not know where to stop, we just show the resulting command annotation.

6.8 STOP_ALTITUDE

Word Transcription	emirates five five stop at flight level one four zero
Additional information	Not clear if aircraft is descending or climbing
Command Annotation	UAE55 STOP_ALTITUDE 140 FL
Explanation	

6.9 MAINTAIN PRESENT_ALTITUDE

Word Transcription	austrian one two three maintain present altitude or below

Additional information	
Command Annotation	AUA123 MAINTAIN PRESENT_ALTITUDE OR_BELOW
Explanation	Qualifier OR_BELOW (or OR_ABOVE) are allowed for this command type, but not ABOVE or BELOW.

7 Changing Direction Commands

7.1 Heading, Turn, Turn_By

Word Transcription	**lufthansa nine mike november turn left heading three hundred**
Additional information	
Command Annotation	DLH9MN HEADING 300 LEFT
Explanation	

Word Transcription	**norshuttle four delta india continue left heading zero six zero**
Additional information	
Command Annotation	NAX4DI HEADING 060 LEFT
Explanation	“Continue” could be an indicator to use “MAINTAIN HEADING 060” here (no qualifier allowed). However, the aircraft does currently not fly heading 060, but should make a turn to reach this heading (the aircraft should maintain its left turn, but the important information is where to stop the turn). Hence, “HEADING” with direction is used here that allows to include a value (060) and a qualifier (LEFT).

Word Transcription	**austrian three nine two papa heading zero six zero**
Additional information	
Command Annotation	AUA392P HEADING 060 none
Explanation	The controller did not say left or right, therefore the qualifier is “none”.

Word Transcription	**oscar kilo oscar kilo victor immediate left turn to the west and squawk three one zero three**
Additional information	
Command Annotation	OKOKV HEADING WEST LEFT OKOKV SQUAWK 3103
Explanation	“WEST” is allowed (next to e.g., SOUTH/EAST/NORTH) instead of a three-digit value. An alternative would have been to use “TURN LEFT” and “HEADING WEST none”, but we agreed on the shown solution.

Word Transcription	**german air force six one two papa tower good day startup is approved you are cleared to destination echo delta delta kilo after departure continue on**

	runway heading and climb to two thousand five hundred feet and squawk three five one one
Additional information	
Command Annotation	GAF612 GREETING GAF612 STATION PAPA_TOWER GAF612 STARTUP GAF612 CLEARED TO EDDK GAF612 HEADING RUNWAY_DIR GAF612 CLIMB 2500 ft GAF612 SQUAWK 3511
Explanation	RUNWAY_DIR is another option how to use the alternative HEADING command.

Word Transcription	**luxair nine two eight three turn right by two zero degrees**
Additional information	
Command Annotation	LGL9283 TURN_BY 20 RIGHT
Explanation	Although a unit (degrees) is provided we decided not to annotate with units.

Word Transcription	**lufthansa one two three leave point ageda heading zero nine zero**
Additional information	Word transcription provided as an example only, not based on recorded data. The controller may issue a clearance for an aircraft to fly a specific heading after crossing a waypoint.
Command Annotation	DLH123 HEADING 090 none WHEN PASSING AGEDA
Explanation	

Word Transcription	**lufthansa one two three make a three sixty turn right for spacing**
Additional information	Word transcription provided as an example only, not based on recorded data. The controller may issue a clearance for an aircraft to make one orbit left/right.
Command Annotation	DLH123 TURN_BY 360 RIGHT
Explanation	This is not a holding and not ORBIT, because it is already clear, that it leaves after one round.

Word Transcription	**sky west thirty two seventy one DFW ground right one eighty onto kilo and then kilo bravo to the ramp**
Additional information	

Command Annotation	SKW3271 STATION DFW_GND SKW3271 TURN_BY 180 RIGHT SKW3271 TAXI VIA K KB SKW3271 TAXI TO RAMP
Explanation	The command TURN_BY can also be used for vehicles/aircraft on the ground. Here, an aircraft shall make a U-Turn to the right on another taxiway that goes parallel to the current taxiway.

Word Transcription	**ryan air five four six seven turn right heading two eight five degrees continue down altitude five thousand feet QNH one zero one nine**
Additional information	
Command Annotation	RYR5467 HEADING 285 RIGHT RYR5467 DESCEND 5000 ft RYR5467 INFORMATION QNH 1019
Explanation	The spoken words might not be according to phraseology standards, but we annotate the intended meaning, therefore DESCEND is annotated.

Word Transcription	**delta echo charlie echo charlie papa tower crossing the airfield is approved at one thousand feet report cross completed**
Additional information	
Command Annotation	DECEC STATION TOWER_PAPA DECEC ALTITUDE 1000 ft DECEC REPORT_MISCELLANEOUS
Explanation	The part “crossing the airfield is approved” is not annotated as a command. “HEADING RUNWAY_DIR CROSS” would be something different.

7.2 MAINTAIN HEADING

Word Transcription	**austrian one two zero one continue on present heading two nine zero**
Additional information	Here we provide a value. Therefore, we use MAINTAIN HEADING. If we do not provide a value as in “… continue on present heading” we should use CONTINUE PRESENT_HEADING
Command Annotation	AUA1201 MAINTAIN HEADING 290
Explanation	290 as heading value is provided.

7.3 CONTINUE PRESENT_HEADING

Word Transcription	**sky travel two india juliett continue present heading**

Additional information	
Command Annotation	TVS2IJ CONTINUE PRESENT_HEADING
Explanation	We have a maintain heading without a heading value. Therefore, we transcribe with CONTINUE PRESENT_HEADING

Word Transcription	**lufthansa two zero one roger continue on present heading**
Additional information	
Command Annotation	DLH201 CONTINUE PRESENT_HEADING
Explanation	No heading value is provided. Therefore, we use "CONTINUE PRESENT_HEADING" and not "MAINTAIN HEADING none".

Word Transcription	**lufthansa four echo charlie maintain heading**
Additional information	
Command Annotation	DLH4EC CONTINUE PRESENT_HEADING
Explanation	No heading value is provided. Therefore, we use "CONTINUE PRESENT_HEADING" and not "MAINTAIN HEADING none".

7.4 DIRECT_TO, TRANSITION, FOLLOW_ROUTE

7.4.1 DIRECT_TO

See subsection 7.4.5 for DIRECT_TO in combination with NAVIGATION_OWN.

Word Transcription	**lufthansa one two three expect runway three four wind one one zero degrees eight knots direct to sorok**
Additional information	SOROK is a waypoint
Command Annotation	DLH123 EXPECT RUNWAY RW34 DLH123 INFORMATION WINDDIRECTION 110 DLH123 INFORMATION WINDSPEED 8 kt DLH123 DIRECT_TO SOROK none
Explanation	We annotate as "EXPECT RUNWAY runway-value" only if the controller says explicit "expect runway …". Otherwise it should be annotated as an ACTIVE_RWY command.

Word Transcription	**lufthansa one two three runway three four wind one one zero degrees eight knots direct to sorok**
Additional information	

Command Annotation	DLH123 INFORMATION ACTIVE_RWY RW34 DLH123 INFORMATION WINDDIRECTION 110 DLH123 INFORMATION WINDSPEED 8 kt DLH123 DIRECT_TO SOROK none
Explanation	If the controller doesn't specify "expect runway…", we annotate it in any other situation like "INFORMATION ACTIVE_RWY runway-value"

Word Transcription	**november three two one alfa victor proceed delta mike four five three [hes][hes] delta mike four five nine**
Additional information	Here we have a problem: if we assume that "[hes][hes]" is just for "correction", then annotation should be "N321AV DIRECT_TO DM459 none, N321AV CORRECTION" If we assume that "[hes][hes]" means that the controller is thinking, i.e. "[hes][hes]" does not exist; we need to annotate two waypoints, "N321AV DIRECT_TO DM453 DM459 none". Here we assume the first interpretation.
Command Annotation	N321AV DIRECT_TO DM459 none N321AV CORRECTION
Explanation	

Word Transcription	**ryan air six five hotel charlie fly direct barcelona**
Additional information	Also, Barcelona is a name of a town in this case it is the airport (LEBL)
Command Annotation	RYR65HC DIRECT_TO LEBL none
Explanation	For airports, we use the ICAO-code (LEBL) for annotation instead of the IATA-code (BCN) or the written name (barcelona).

Word Transcription	**ryan air six five hotel charlie fly direct bubovice**
Additional information	Bubovice is an airport and not a waypoint with letter code LKBU.
Command Annotation	RYR65HC DIRECT_TO LKBU none
Explanation	For airports, we use the ICAO-code (LKBU) for annotation.

Word Transcription	**air algerie one one four nine proceed sadaf**
Additional information	In the waypoints data base file we have specified that the word "sadaf" is mapped to waypoint SADEF.
Command Annotation	DAH1149 DIRECT_TO SADEF none
Explanation	Proceed is mapped to command type DIRECT_TO.

Word Transcription	**november three two one alfa victor proceed delta mike four five three delta mike four five nine**
Additional information	Better phraseology would be via delta mike four five three to delta mike four five nine.
Command Annotation	N321AV DIRECT_TO DM453 DM459 none
Explanation	We have two waypoints. Here we have a special order. Aircraft should fly to DM453 and then to DM459. N321AV DIRECT_TO DM453 none N321AV DIRECT_TO DM459 none WHEN PASSING DM453 Might be also possible, but that should be the transcription for the utterance “november three two one alfa victor proceed delta mike four five three after passing delta mike four five three direct to delta mike four five nine”

Word Transcription	**turkish five delta turn now left make a left turn to waypoint delta lima four eight nine**
Additional information	
Command Annotation	THY5D DIRECT_TO DL489 LEFT
Explanation	

Word Transcription	**sky travel two india juliett proceed direct papa romeo five three zero**
Additional information	
Command Annotation	TVS2IJ DIRECT_TO PR530 none
Explanation	

Word Transcription	**russia six six two one descend four thousand feet QNH one zero one four via papa romeo five three two cleared ILS runway two four**
Additional information	
Command Annotation	SDM6621 DESCEND 4000 ft SDM6621 INFORMATION QNH 1014 SDM6621 DIRECT_TO PR532 none SDM6621 CLEARED ILS RW24
Explanation	The keyword “via” is used to indicate a DIRECT_TO (if it is not said by a tower or apron controller, because it might be CLEARED VIA, TAXI VIA or VACATE VIA then).

Word Transcription	**sky dubai seven seven nine confirm proceeding to papa romeo five three two and now descend flight level eight zero**
Additional information	
Command Annotation	FDB779 DIRECT_TO PR532 none FDB779 DESCEND 80 FL
Explanation	The clearance DIRECT_TO is repeated with a confirm keyword.

Word Transcription	**KLM five six tango baltu is approved**
Additional information	
Command Annotation	KLM56T DIRECT_TO BALTU none
Explanation	Approved for DIRECT_TO unusual, but it is used.

Word Transcription	**aero flot two zero one two descend to four thousand feet QNH one zero two zero and via papa romeo five three two cleared for ILS approach runway two four**
Additional information	“Descend to four thousand feet” could be understood as “DESCEND 24000 ft” if there is a misinterpretation of “to” and “two”
Command Annotation	AFL2012 DESCEND 4000 ft AFL2012 INFORMATION QNH 1020 AFL2012 DIRECT_TO PR532 none AFL2012 CLEARED ILS RW24
Explanation	Via with a waypoint name is annotated as DIRECT_TO.

Word Transcription	**oscar echo kilo romeo juliett cleared to laibach via graz flight planned route climb one zero thousand feet QNH one zero two two IFR starts passing seven thousand feet**
Additional information	This is not a clearance of the tower controller, but of the sector controller, immediately after takeoff.
Command Annotation	OEKRJ CLEARED TO LJLJ OEKRJ CLEARED VIA GRZ OEKRJ CLIMB 10000 ft OEKRJ INFORMATION QNH 1022 OEKRJ CLEARED IFR WHEN PASSING 7000 ft
Explanation	It is a clearance for a VFR with a condition to change to IFR. We choose CLEARED TO and CLEARED VIA, because we would use the same annotation, if given by tower, although “DIRECT_TO GRZ LJLH none” might have also been a solution. However, a DIRECT_TO must immediately be followed, an IFR-Clearance can be followed after some time.

7.4.2 DIRECT_TO with TIME constraint

Word Transcription	**airbaltic two hotel uniform level three eight zero approaching isvig at one two one zero**
Additional information	This is a report from the pilot
Command Annotation	BTI2HU REPORTING ALTITUDE 380 FL BTI2HU REPORTING DIRECT_TO ISVIG WHEN TIME 1210
Explanation	Reporting at what time ISVIG will be reached.

7.4.3 TRANSITION

Word Transcription	**austrian five two two delta servus ILS approach three four initially balad three november transition standby for shortcut**
Additional information	We use TRANSITION for transitions, but not for waypoints or SIDs (Standard instrument departures) (then we use CLEARED VIA) and not for arrival routes (then we use FOLLOW_ROUTE).
Command Annotation	AUA522D EXPECT_ILS RW34 AUA522D TRANSITION BALAD_3N AUA522D CALL_YOU_BACK
Explanation	The standby request/information results in NO_CONCEPT and is therefore not added. NO_CONCEPT only added to command annotation if we have no other command resulting from the utterance.

7.4.4 FOLLOW_ROUTE

Word Transcription	**lot three eight seven cleared nukro three victor arrival**
Additional information	Compare with TRANSITION and CLEARED VIA
Command Annotation	LOT387 FOLLOW_ROUTE NUKRO_3V
Explanation	Although the controller used the phraseology “cleared …”, we annotate with type “FOLLOW_ROUTE”. The ontology only contains the command types TRANSITION, DIRECT_TO and FOLLOW_ROUTE. “nukro tree victor” is specified as arrival route NUKRO_3V. Also different word sequences which could be used for this arrival route are specified, e.g. “november uniform kilo romeo oscar tree victor arrival” or “nukro three victor arrival”.

Word Transcription	**monarch eight seven four expect lores one papa**
Additional information	

Command Annotation	MON874 FOLLOW_ROUTE LORES_1P
Explanation	

Word Transcription	**tarom four two one bisba three yankee two five right**
Additional information	Word transcription always in small letters.
Command Annotation	ROT421 FOLLOW_ROUTE BISBA_3Y ROT421 EXPECT_RUNWAY RW25R
Explanation	Depending on contents of database for the arrival route names BISBA_3Y_25R could also be an option. Then, however, without EXPECT_RUNWAY command.

Word Transcription	**scanwing one two four mike cleared approach runway three zero**
Additional information	
Command Annotation	SCW124M FOLLOW_ROUTE ROUTE_30
Explanation	

Word Transcription	**fraction seven three seven foxtrot squawk five zero seven two cleared nukro four victor arrival**
Additional information	
Command Annotation	NJE737F SQUAWK 5072 NJE737F FOLLOW_ROUTE NUKRO_4VARR
Explanation	The representation of “nurko four victor arrival” depends on how it is specified in the configuration. In this case as “NUKRO_4VARR”.

Word Transcription	**aeromexico zero zero one continue climb one nine zero established on bravo delta zero zero two**
Additional information	The semantics of the command is to climb to FL 190 and to follow the route BD002.
Command Annotation	AMX001 CLIMB 190 none AMX001 FOLLOW_ROUTE BD002
Explanation	

7.4.5 NAVIGATION_OWN

Word Transcription	**CSA four mike foxtrot own navigation to odnem**
Additional information	
Command Annotation	CSA4MF NAVIGATION_OWN CSA4MF DIRECT_TO ODNEM none

Explanation	Both commands are given, i.e. NAVIGATION_OWN and DIRECT_TO. We annotated in the order they are said.

Word Transcription	**speed bird three zero charlie victor resume own navigation direct rapet**
Additional information	
Command Annotation	BWA30CV NAVIGATION_OWN BWA30CV DIRECT_TO RAPET none
Explanation	Both commands are given, i.e. NAVIGATION_OWN and DIRECT_TO.

Word Transcription	**break break sky travel six echo hotel vectoring terminated resume own navigation to papa romeo five three zero**
Additional information	The break, break in the beginning is not annotated, because we have not two callsigns in the same utterance.
Command Annotation	TVS6EH NAVIGATION_OWN TVS6EH DIRECT_TO PR530 none
Explanation	The termination of vectoring is not covered by any command type.

Word Transcription	**lot five two two turn right and resume own navigation to papa romeo four zero nine**
Additional information	Combined turn with waypoint, but navigation command in between
Command Annotation	LOT522 TURN RIGHT LOT522 NAVIGATION_OWN LOT522 DIRECT_TO PR409 none
Explanation	LOT522 NAVIGATION_OWN, LOT522 DIRECT_TO PR409 RIGHT would also be thinkable. However, it could be that the aircraft shall turn right, then resume its own navigation to the waypoint PR409 which might me left-hand. Hence, without “and resume own navigation”, “DIRECT_TO PR409” would have been correct.

8 Vertical Speed Commands

8.1 RATE_OF_CLIMB

Word Transcription	**austrian nine zero three echo rate of climb initially one thousand feet per minute or less there will be jet traffic behind to bring above**
Additional information	
Command Annotation	AUA903E RATE_OF_CLIMB 1000 ft_min OR_LESS AUA903E INFORMATION TRAFFIC none
Explanation	

8.2 RATE_OF_DESCENT

Word Transcription	**vueling one eight eight nine descend at two thousand feet per minute or greater**
Additional information	
Command Annotation	VLG1889 RATE_OF_DESCENT 2000 ft_min OR_GREATER
Explanation	We use qualifier “OR_GREATER”.

Word Transcription	**lufthansa six juliett papa descend seven thousand feet and rate one thousand five hundred or greater**
Additional information	
Command Annotation	DLH6JP DESCEND 7000 ft DLH6JP RATE_OF_DESCENT 1500 none OR_GREATER
Explanation	No unit is provided therefor “none” as unit (happens quite often)

8.3 RATE_OF_DESCENT OWN, EXPEDITE, MAX

Word Transcription	**austrian five two two delta own rate of descent thanks**
Additional information	
Command Annotation	AUA522D RATE_OF_DESCENT OWN
Explanation	This phraseology exists even without the two words “of descent” resulting in the same annotation. OWN is the sub type name.

Word Transcription	**austrian five two two delta increase rate of descent two thousand feet per minute or greater**
Additional information	

Command Annotation	AUA522D RATE_OF_DESCENT 2000 ft_min OR_GREATER
Explanation	“Increase” has not further meaning in this case.

8.4 VERTICAL_RATE

Word Transcription	**air berlin eight nine nine one flight level one four zero at two thousand five hundred feet per minute**
Additional information	
Command Annotation	BER8991 ALTITUDE 140 FL BER8991 VERTICAL_RATE 2500 ft_min
Explanation	It is not said, whether 2500 feet per minute is a climb rate or a descent rate. Therefore, we annotate with the more general type VERTICAL_RATE. In this case RATE_OF_DESCENT or RATE_OF_CLIMB could not be derived from this utterance.

Word Transcription	**air berlin eight nine nine one descend flight level one four zero at two thousand five hundred feet per minute or greater**
Additional information	
Command Annotation	BER8991 DESCEND 140 FL BER8991 RATE_OF_DESCENT 2500 ft_min OR_GREATER
Explanation	It is not said, whether 2500 feet per minute is a climb rate or a descent rate, we can, however, derive it from the descend command in the same utterance Therefore, we do NOT annotate with the general type VERTICAL_RATE. In this case RATE_OF_DESCENT or RATE_OF_CLIMB could be derived from this utterance, because also DESCEND is said.

8.5 VERTICAL_RATE OWN, EXPEDITE, MAX

Word Transcription	**air france seven zero alfa maximum feet per minute due to traffic please**
Additional information	
Command Annotation	AFR70A VERTICAL_RATE MAX
Explanation	Even if there is no value, the vertical speed rate should be maximized. MAX is a sub type and not a qualifier.

8.6 EXPEDITE_PASSING

Word Transcription	**speed bird six four eight expedite passing flight level two nine zero**

Additional information	The aircraft could have received a descend command to flight level 250 before. Hence, the aircraft should expedite the descent to pass 290 FL, but afterwards, the aircraft can again use its own descent profile from 290 FL to 250 FL. This command is a bit ambiguous (one out of five pilots may level off at 290 FL even if they have received the explained other/lower clearance before; hence controllers might issue such clearances differently).
Command Annotation	BAW648 EXPEDITE_PASSING 290 FL
Explanation	

8.7 RATE_OF_CLIMB OWN, EXPEDITE, MAX

Word Transcription	**emirates five two four please increase your rate of climb due to traffic**
Additional information	
Command Annotation	UAE55 RATE_OF_CLIMB EXPEDITE
Explanation	Even if there is no value, the rate-of-climb should be increased.

9 Speed-Commands

Command types REDUCE, INCREASE, and SPEED are three of the main types for speed related commands. SPEED indicates that it is not clear from the utterance whether the aircraft should increase or reduce. If we know it is a departure, it might be clear that the controller will issue an INCREASE. Nevertheless, command annotation should be "SPEED", if the controller does not explicitly say the word "increase" or "reduce". We also add the unit "kt" or "MA". Furthermore, we have special speed commands (e.g. REDUCE_MIN_CLEAN_SPEED) and MAINTAIN SPEED as well as relative speed commands such as REDUCE BY, INCREASE_BY.
The words OR_LESS resp. OR_GREATER could be added as qualifier. LESS or GREATER are also possible, although seldomly used.

9.1 REDUCE

Word Transcription	**austrian three nine two papa reduce speed one eighty**
Additional information	
Command Annotation	AUA392P REDUCE 180 none
Explanation	No unit is spoken; therefore, we annotate with none.

Word Transcription	**wizz air seven zero one we have departing traffic please reduce speed**
Additional information	
Command Annotation	WZZ701 INFORMATION TRAFFIC none WZZ701 REDUCE none none
Explanation	REDUCE is used without a speed value. Therefore, "none none" is necessary. The second none is used because no unit (e.g. knots) is said.

9.2 INCREASE

Word Transcription	**tui jet two papa hotel increase speed two seventy or greater for sequencing**
Additional information	
Command Annotation	TUI2PH INCREASE 270 OR_GREATER
Explanation	

9.3 SPEED

Word Transcription	**monarch two six zero eight fly speed three five zero knots**
Additional information	
Command Annotation	MON2608 SPEED 350 kt

Explanation	We have no information whether to reduce or to increase.

Word Transcription	**ryan air one two alfa charlie … mach number point seven six maximum**
Additional information	In enroute phase often Mach numbers instead of CAS values are provided.
Command Annotation	RYR12AC SPEED 0.76 MA OR_BELOW
Explanation	The word “maximum” indicates the maximum allowed speed. Therefore, we add OR_BELOW, because that is the semantics of maximum in this context.

Word Transcription	**tom jet five eight kilo quebec cross corda flight level one nine zero and maintain flight level two hundred at three five zero knots**
Additional information	In an oceanic clearance, the command can be quite complex.
Command Annotation	TOM58KQ ALTITUDE 190 FL UNTIL CORDA TOM58KQ MAINTAIN ALTITUDE 200 FL AFTER CORDA TOM58KQ SPEED 350 kt AFTER CORDA
Explanation	The condition in the second and third command is not said. So, it will be a challenge for any language processing algorithm. Nevertheless, the semantics is quite clear.

9.4 MAINTAIN PRESENT_SPEED

Word Transcription	**austrian seven six eight zulu maintain present speed or above**
Additional information	
Command Annotation	AUA768Z MAINTAIN PRESENT_SPEED OR_GREATER
Explanation	We use MAINTAIN PRESENT_SPEED if no value is specified. Otherwise we use MAINTAIN SPEED plus value.

9.5 MAINTAIN SPEED

Word Transcription	**alitalia five one two maintain speed two hundred knots or greater**
Additional information	
Command Annotation	AZA512 MAINTAIN SPEED 200 kt OR_GREATER
Explanation	We use MAINTAIN SPEED if the word “maintain” was said before speed with value.

9.6 REDUCE_BY and INCREASE_BY

Word Transcription	**tui jet two papa hotel reduce speed by twenty knots**
Additional information	Reduce speed relatively to the current speed
Command Annotation	TUI2PH REDUCE_BY 20 kt
Explanation	It should be REDUCE_BY and not REDUCE_SPEED_BY to be consistent with absolute REDUCE command. We also have INCREASE_BY. Here SPEED_BY makes no sense.

Word Transcription	**tui jet two papa hotel increase mach by zero point one**
Additional information	Just for consistency we also have a relative increase of mach speed.
Command Annotation	TUI2PH INCREASE_BY 0.10 MA
Explanation	We do not use INCREASE 0.10 MA to point out the difference to the absolute INCREASE command. The ending zero is necessary.

9.7 TIME_CMD commands

Word Transcription	**delta two one five buenos dias just to check your oceanic clearance pasas time zero nine three five and to be no later zero nine three five yes I confirm no later zero nine three five flight level three eight zero mach point eight two call you back for level change**
Additional information	The controller repeats the time clearance of this oceanic clearance three times. The altitude and speed clearance are valid already now, although they might be also valid when passing PASAS.
Command Annotation	DEL215 TIME_CMD 0935 WHEN PASSING PASAS DEL215 TIME_CMD 0935 OR_EARLIER DEL215 TIME_CMD 0935 OR_EARLIER DEL215 ALTITUDE 380 FL DEL215 SPEED 0.82 MA DEL215 CALL_YOU_BACK
Explanation	The TIME_CMD command is repeated three times. It is the task of subsequent modules to filter out the three commands so that we get only one command. In the first-time command only the time is said without “no later”. In the second and third one “no later” is always said. Here the waypoint is not mentioned again. Therefore, we only use the conditional part in the first command. CALL_YOU_BACK has no other parameters.

Word Transcription	**delta two one five cross corda between eight hundred and zero eight zero nine**
Additional information	Oceanic clearance

Command Annotation	DEL215 TIME_CMD BETWEEN 0800 0809 WHEN PASSING CORDA
Explanation	

Word Transcription	**delta two one five cross corda earlier than eight hundred**
Additional information	Oceanic clearance
Command Annotation	DEL215 TIME_CMD 0800 OR_EARLIER WHEN PASSING CORDA
Explanation	

Word Transcription	**two zero four three or later there iceair seven five zero**
Additional information	
Command Annotation	ICE750 TIME_CMD 2043 OR_LATER
Explanation	No condition at all specified. This was a readback to “iceair seven five zero cross rapax at time two zero four three or later” and this a wrong read back because the waypoint is missing

Word Transcription	**blue twenty eight twenty six you got a forty seven time to kennedy**
Additional information	
Command Annotation	JBU2826 TIME_CMD 47 JBU2826 CLEARED TO KJFK
Explanation	The TIME_CMD can also be used on the ground.

9.8 Special Speed Values: NO_SPEED_RESTRICTIONS etc.

Word Transcription	**speed bird six nine seven no ATC speed limit**
Additional information	
Command Annotation	BAW697 NO_SPEED_RESTRICTIONS
Explanation	

Word Transcription	**can we go back to free speed**
Additional information	Request from a pilot
Command Annotation	PILOT REQUEST NO_CALLSIGN NO_SPEED_RESTRICTIONS
Explanation	As this is a pilot request, we need to fill the field SPEAKER and REASON

Word Transcription	**speed bird nine five one lima high speed approved**
Additional information	

Command Annotation	BAW951L HIGH_SPEED_APPROVED
Explanation	

Word Transcription	**austrian seven six eight zulu reduce to final approach speed**
Additional information	
Command Annotation	AUA768Z REDUCE_FINAL_APPROACH_SPEED
Explanation	

Word Transcription	**austrian seven six eight zulu reduce to minimum approach speed**
Additional information	
Command Annotation	AUA768Z REDUCE_MIN_APPROACH_SPEED
Explanation	

Word Transcription	**aero flot two zero two four reduce minimum clean speed and …**
Additional information	
Command Annotation	AFL2024 REDUCE_MIN_CLEAN_SPEED
Explanation	

Word Transcription	**austrian seven six eight zulu resume normal speed**
Additional information	
Command Annotation	AUA768Z RESUME_NORMAL SPEED
Explanation	