Computational Analysis Of Air plane Cockpit-Voice-Recording-Transcripts

Andre Döring Jan Sanders
Mark McGovern
AG Rechnernetze und verteilte Systeme
www.rvs.uni-bielefeld.de
RCS-OCC-01-07

November 11, 2001

Contents

1 Introduction **3**
1.1 The Idea 3
1.2 Overview 3

2 CVR transcripts used **4**
2.1 Nagoya Accident — China Airlines 26.APR 1994 4
2.2 Vnukovo Accident — Vnukovo Flight 2801 - 29.AUG 1996 4
2.3 Airborne Express Flight 827 4
2.4 Lufthansa Flight 2904 - 14.SEP 1993 5
2.5 Birgen Air 5

3 The Interpreter **7**
3.1 System Requirements 7
3.2 General Design 7
3.3 Input used 8
3.3.1 Transcript 8
3.3.2 Attribute Grammar 8
3.4 Hierarchical State Machine 10
3.5 The Modules 11
3.5.1 *Preprocessing* 11
3.5.2 *Identify Speaker* 11
3.5.3 *Identify Addressee* 11
3.5.4 *Identify Type Of Expression* 12
3.6 *Analysis of the Topic* 13
3.7 *Context-Logging* 13
3.8 Output And Output Format 13
3.9 Diagraph 14

4 Testing And Results **15**
4.1 The Procedure 15
4.2 The Results 16
4.2.1 Birgen Air Accident 16
4.2.2 Nagoya Accident 16
4.2.3 Airborne Express Flight 827 16
4.2.4 UN Flight KSV 2375 16
4.2.5 Lufthansa Flight 2904 16

5 Problems And Methods Of Resolution **17**
5.1 Problems and possible solutions 17
5.1.1 Problem-Handlers 17

6 Future Perspectives **18**

Abstract

Computational analysis of actual natural language is hard. We decided to focus our interest on a specialised language part. This part is a technical language spoken in air plane cockpits and by air traffic control (ATC).We use transcripts from cockpit voice recordings in air plane crashes.

Our goal was to develop a rigorous algorithmic procedure for ATC-Language-Transcripts which uses semantic analysis to recognise the topic of single sentences according to the context and dialog-acts in the transcript.

The results are a rough design of a possible interpreter structure, an attribute-grammar for the interpreter according to keywords included in the transcripts and an algorithm we used to check our interpreter-design by hand.

In our working process we faced several problems. For example long lists of substantives or instructions and incomplete sentences (e.g. sentences without a verb) were hard to interpret. To solve these problems we decided to build specialised problem-handlers.

We gained some interesting insights in approximation of intuitive semantics in natural spoken language with a computer driven interface.

1 Introduction

1.1 The Idea

Language can be represented as a complex system of rules and dictionaries divided into several linguistic levels, such as phonetics, phonology, morphology, sematics and pragmatics. We want to develop an algorithmic system to analyse the semantics of natural spoken sentences. We focused on building an algorithm that can be used by a computer driven interpreter.

This reports our research in the interpretation of natural spoken sentences in a restricted technical domain.

The language we chose is called ATC-Language. This language is (naturally) spoken by airtraffic control and airplane pilots. It has a general specification by a fixed set of rules and words used. Because of these restrictions it is much easier to build a formal grammar for this language than for general language.

ATC language analysis is performed in cases of airplane crashes. Many examples are documented on the webpages of NTSB in the U.S., BASI in Australia and AAIB in G.B. The dialogues during these incidents have been transcribed to written versions (Cocckpit-Voice-Recording: CVR).

The result of the interpretation is information about the topic of sentences in the dialogue context between all actors. The interpreter counts several values such as word-count, sentence-count, count of (not)-interpreted sentences etc. According to these values we hope to get more information about the human factors in those incidents.

1.2 Overview

This report describes the structure of a rigorous algorithmic procedure for semantic analysis of dialogue semantics and restricted context. This could be used in a possible implementation of a computer driven interpreter system for the analysis of semantics in natural spoken language. When we use the word interpreter it is synonymous to this algorithm.

After that the testing and results are described, problems are discussed and future perspectives are mentioned.

2 CVR transcripts used

The transcripts are selected on availability. These were the transcripts we use to estimate and improve our design. All keywords and phrases are taken from here.

The following sections will shortly describe what happened in these incidents or accidents. The descriptions are mostly quotations from the original accident-report.

2.1 Nagoya Accident — China Airlines 26.APR 1994

[...] "China Airlines Airbus Industries A300B4-622R B 1816 took off from Taipei International Airport at 0853 UTC (1753 JST) on April 26, 1994 and continued flying according to its flight plan. About 1116 UTC (2016 JST), while approaching Nagoya Airport for landing, the aircraft crashed into the landing zone close to E1 taxiway of the airport.

On board the aircraft were 271 persons: 256 passengers (including 2 infants) and 15 crew members, of whom 264 persons (249 passengers including 2 infants and 15 crew members) were killed and 7 passengers were seriously injured. The aircraft ignited, and was destroyed."[...] (see [2])

2.2 Vnukovo Accident — Vnukovo Flight 2801 - 29.AUG 1996

The aircraft was on the final approach to the runway. 3.7km before landing they encountered strong turbulence and crashed into a mountain Operafjellet.

Eighteen different factors have been detected by the analysts. [...] "For example, inadequate planning, unsatisfactory crew resource management and monitoring, a lack of a suitable procedure for offset localiser approaches in connection with an inappropriate rule requiring the landing course to be set instead of the localiser course, not solving navigational problems at safe altitude, not discontinuing the approach when procedural uncertainties exist, limited knowledge of the operating language and the actual airspace with respect to service given." [...] (see [3]).

2.3 Airborne Express Flight 827

[...]"The empty freighter aircraft was just reconfigured at Greensboro when it was flown to the ABX hub in Wilmington (IFR). The aircraft nosed down into White River Mt. at 3400ft and exploded shortly after passing overhead Giles County, 26 minutes after take-off. The DC-8 was manoeuvring between FL130 and FL150 when the aircraft continued descending below FL130 at a low airspeed, trying to recover from a stall. The stall was part of the post-modification functional evaluation flight programme.

PROBABLE CAUSE: " The National Transportation Safety Board determines that the probable causes of this accident were the inappropriate control inputs applied by the flying pilot during a stall recovery attempt, the failure of the non-flying pilot-in-command to recognise, address and correct these inappropriate control inputs, and the failure of Airborne Express to establish a formal, functional evaluation flight program that included adequate program guidelines, requirements and pilot training for performance of these flights.

Contributing to the cause of the accident were the inoperative stick shaker stall warning system and Airborne Express DC-8 flight training simulator's inadequate fidelity in reproducing the air plane's stall characteristics." " [...] (see [4])

2.4 Lufthansa Flight 2904 - 14.SEP 1993

[...] "The aircraft was cleared for a Warsaw Runway 11 approach and were told of the existence of winds hear on the approach. The Airbus' right gear touched down 770m from the Runway 11 threshold. The left gear touched down 9 seconds later, 1525m from the threshold. Only when the left gear touched the runway, automatic systems depending on oleo strut (shockabsorber) compression unlocked the use of ground spoilers and engine thrust reversers. The wheel brakes, depending on wheel rotation being equivalent of circumfential speed of 72 kts began to operate after about 4 seconds. Seeing the approaching end of the runway and the obstacle behind it, the pilot steered the aircraft off the runway to the right. The aircraft left the runway at a speed of 72 kts and rolled 90m before it hit the embankment and an LLZ aerial with the left wing. A fire started in the left wing area and penetrated into the passenger cabin.

PROBABLE CAUSE: "Cause of the accident were incorrect decisions and actions of the flight crew taken in situation when the information about winds hear at the approach to the runway was received. Winds hear was produced by the front just passing the aerodrome; the front was accompanied by intensive variation of wind parameters as well as by heavy rain on the aerodrome itself.

Actions of the flight crew were also affected by design features of the aircraft which limited the feasibility of applying available braking systems as well as by insufficient information in the aircraft operations manual (AOM) relating to the increase of the landing distance." [...]

2.5 Birgen Air

[...] "On February 7, the FAA issued a Press Release (Office of public affair Press Releases) clarifying the role played by the U.S. FAA (Federal Aviation Administration) and the NTSB (National Transportation Safety Board) in the investigation. On March 1, a short statement of Factual Information from a preliminary review of CVR and FDR data was made available by the NTSB on behalf of the Dominican Republic civil aviation authorities. On March 18, a longer Press Release, accompanied by the CVR transcript, explained further what the FDR and CVR data indicated. David Learmount in Flight International, 27 March - 2 April 1996, deduced from the CVR transcript four salient observations on the crew behaviour and I provide a fifth from the B757 Operations Manual B757 Air Data System description and schematic diagram(JPEG, GIF). To paraphrase Learmount's points, although confusion about operation of computer-assisted systems (autopilot, warning annunciation) played a role, this confusion would not have arisen but for inappropriate pilot decisions and actions. However, a blocked pilot tube and inappropriate pilot behaviour are not the only potential factors under study. The NTSB has identified a potential improvement in B757/767 operating manual as a result of further analysis (short note).

Note on the Puerto Plata and Cali accidents, highlighting the human-computer interface (HCI) issues, appeared in RISKS-18.10, was rebroadcast on Phil Agre's RRE mailing list (May 7th), and became the subject of the what's happening column of the British Computer Society HCI interest group magazine Interactions, July/August 1996, p13." [...] (see [6])

3 The Interpreter - A Rigorous Algorithmic Procedure

3.1 System Requirements

We intend the interpreter to fulfil the following requirements.

- It has to be a rigorous algorithmic procedure. The procedure will be sufficient for an eventual computational implementation.It will define computational details and characteristics.
- Input to the system will be transcriptions of natural spoken language. The input format will be a XML document developed by Hettenhausen/Hölz.
- Additional input is needed about the basic situation the CVR was taken from. (e.g. flight-mode, condition of aircraft, role-model for all involved Speakers, environmental variables)
- The system distinguishes between relevant (ATC language conform) communication and irrelevant communication. The distinction is made expression-wise.
- Relevant expressions will be semantically analysed. Speaker, addressee, topic and eventual action shall be determined.
- All relevant expressions will be considered in connection with each other to determine pragmatics of the dialogue.
- Contextual situated actions, verbal and non verbal, are recognised.
- Statistical data shall be gained from the transcript to provide a means of classification for CVR transcripts. We have only investigated what data and statistics are useful

3.2 General Design

The algorithm consists of a chain of interacting modules. The modules take data, transform it or simply classify it, then hand data to the next module. Some modules depend on others, some can work in isolation.

All data directly related to the transcribed dialogue will have to be in the transcript. The algorithm uses other information that is independent of single transcripts but is needed for CVR transcript analysis as a whole. Thomas Hettenhausen and Oliver Hölz are developing a corpus of XML annotated CVR transcript. We will use their work and their XML format as input for our interpreter. We need sufficient knowledge of ATC phrases and vocabulary. ATC procedures will have to be taken into account.

The output consists of an annotated version of the transcript with additional statistical data.

3.3 Input used

All data was gathered by simply looking at the transcripts. We first analysed the dialogues using our own intuition and the help of Prof. Peter Ladkin who is an expert in aviation safety and a pilot on his own. We derived a list of keywords and phrases used from the scripts which we use as a basis for the attribute grammar described below.

We devised a hierarchical state machine, described below, which represents the states and flight modes the aircraft can be in. The data is incorporated into the state machine structure. Features of ATC procedures are incorporated into the state machine structure

3.3.1 Transcript

The transcript is the substantial part of the algorithm's input data. The transcript must contain all information directly related to the transcribed dialogue. (See System Requirements)

We distinguish between the dialogue, which is the actual conversation that took place, and the transcript which carries information about the dialogue and a written version of the dialogue itself.

In order to be able to follow the dialogue, extensive information is needed about the dialogue's environment.

The transcript must contain:

1. the dialogue
2. annotations concerning speaker and time; additionally the duration of the expression may be helpful but is not required.
3. the roles the speakers were in: Captain, 1st Officer, Pilot flying, Pilot not flying, AT Controller, other Aircrafts' crews, non-pilot speakers in the cockpit

Some information cannot be explicitly found in the transcripts. It has to be interpreted form the transcript.

1. the flight state the aircraft is in at the beginning of the dialogue
2. the condition the aircraft is in at the beginning of the dialogue
3. position, altitude, speed and direction of the aircraft at the beginning of the dialogue

3.3.2 Attribute Grammar

We used an attribute grammar for pattern matching. The attribute grammar was derived from a key phrases list containing most standard phrases in actual ATC communication.

The key phrases from the key-phrase list are taken as the grammar's attributes. The attribute grammar serves as a computable representation of the phrases. Each attribute has three sets of values. The last two sets must at least contain one element. There is no other limit to the number of elements. The first set must contain exactly one element. The sets are as follows:

CONTEXT The element of this set declares in what context (topic) the key-phrase is used. In case that one key-phrase can be used in several contexts there are two entries in the attribute grammar for each usage. This way the other sets will only contain elements related to the context in question.

CO_CONTEXT This set contains context elements, too. But the elements in this set are contexts a keyword can be connected with. e.g.: *altitude 200ft*. *altitude* is used when talking about height. *2000ft* is a distance measure. So one CO_CONTEXT element for *altitude* would be DISTANCE_MEASURE.

FLIGHT_PHASE This set contains all flight phases the aircraft can be in when a keyword is used.

List of CONTEXTs (and CO_CONTEXTs) All context names start a C_ followed by a non-capital. When a context name consists of two words the first letter of each following word is a capital. Context names are written as one word. Context names following in brackets are sub categories of the preceding context name.

- C_all
- C_altitude (C_absoluteAltitude, C_relativeAltitude, C_altitudeChange)
- C_communication (C_radioFrequency, C_radioAlphabet, C_radioNumeral, C_confirmation, C_instruction)
- C_direction (C_north(C_nw,C_nw), C_west, C_south(C_sw,C_se), C_east, C_left, C_right, C_up, C_down)
- C_distance (C_absoluteDistance, C_relativeDistance, C_distanceChange)
- C_flightState (C_taxiing, C_takeoff, C_climb, C_cruise, C_approach, C_descend, C_landing, C_emergency, C_urgency,C_goaround)
- C_heading (C_absoluteHeading, C_relativeHeading, C_headingChange)
- C_measurement (C_angleMeasurement, C_altitudeMeasure, C_speedMeasure, C_heightMeasure, C_distanceMeasure, C_volumeMeasurement, C_timeMeasurement)
- C_none
- C_object (C_flyingObject, C_groundObject(C_airport, C_airportObject(C_taxiway, C_runway(C_runwayObject), C_tower)), C_plane, C_terrain, C_airTrafficControl)
- C_permission(C_cleared, C_notCleared)
- C_planePart (C_controlableSystem, C_staticSystem)
- C_position (C_absolutePosition, C_relativePosition, C_positionChange, C_mark, C_threshold)
- C_quality
- C_speed (C_absoluteSpeed, C_relativeSpeed, C_speedChange)
- C_technical
- C_weather (C_fog,C_ ice, C_rain, C_snow, C_visibility, C_wind(C_windshear))

The Flight-phases These flight phases are not the same as the flight-phases described in the AIM[7]. When talking about the flight phases used in the AIM we will explicitly say "AIM-Flight-phases". Otherwise the attribute grammar flight-phases are meant. All elements start with F_ followed by a non-capital.

- F_taxiing
- F_takeoff
- F_climb
- F_cruise
- F_approach
- F_descend
- F_landing
- F_goaround
- F_emergency
- F_urgency
- F_all
- F_none

3.4 Hierarchical State Machine

The HSM is not yet fully designed. But the requirements for this HSM are:

- One top level state machine is needed to represent the whole flight.
- More than one non-top-level state machine must be able to be active at a time
- For each consecutive flight-phase one state machine is needed.
- For each urgency and emergency the aircraft can be in, there must be one state machine.
- All state machines must comply with ATC regulations, in the sense that they must be able to represent non-ATC-conformant behaviour, so ...
- ..two kinds of states in each machine are needed:

  DO_STATES These states are well defined states the aircraft can be in following ATC regulations and procedures.

  DONT_STATES These states can only be reached when ATC procedures were not followed

  One state machine resembling the technical status of an aircraft

  One state machine resembling position, altitude and direction of an aircraft

  One state machine resembling the status of persons in the system

  One state machine resembling environmental conditions

  One state machine resembling airport conditions

3.5 The Modules

The modules form the computing part of our algorithmic procedure. Each module is designed to perform one specific task. The modular design is extensible. Single modules can be changed without looking at other modules.

When in some descriptions the module works on *context* this means that preceding expressions are taken into account. Since we focused on single modules and their interaction, we have not yet defined how context is to be evaluated in detail. We will leave that open in this document. Following is a list of modules. Each module is explained in a paragraph. The corresponding headline gives is the module's name.

3.5.1 *Preprocessing*

All other modules depend on preprocessing. The purpose of this module is to check that all required data is included in the script (see Transcript). It works on the transcript as a whole.

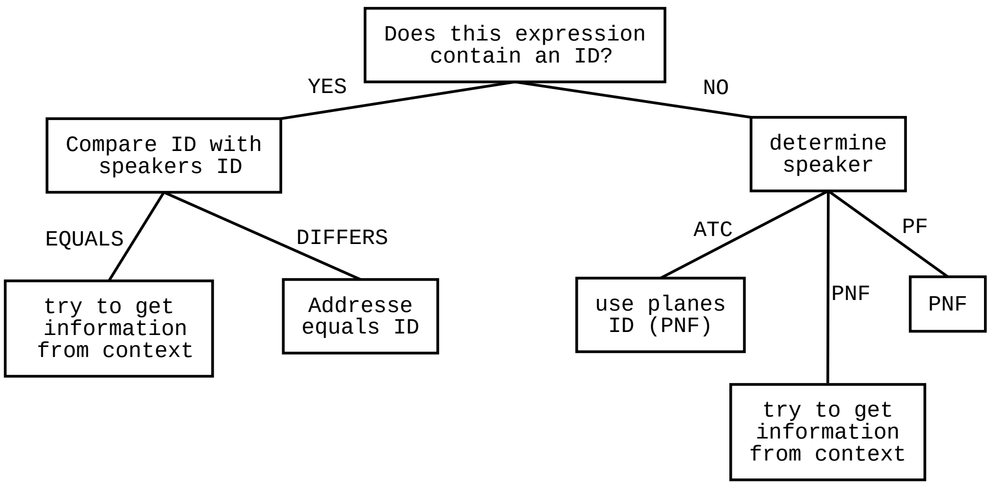
3.5.2 *Identify Speaker*

This module works on a single expression. The purpose of this module is to identify the speaker of an expression. It can work independent from other modules. In case of a natural speaker (e.g. Pilot, AT-Controller) identification is very easy because the information is already given in the transcript. Machine generated sounds (e.g. Warnings) have to be classified from the examined statement.

3.5.3 *Identify Addressee*

This module works on a single expression. The purpose of this module is to identify the addressee of an expression. This module needs the information gained from *Identify Speaker*.

To determine the addressee a decision tree is followed:



PF = Pilot Flying, PNF Pilot not Flying, ATC = Air Traffic Controller, ID = aircraft identification

Try to get information from context means that previous phrases have to be examined. This feature is not yet specified.

3.5.4 *Identify Type Of Expression*

This module works on a single expression. The purpose of this module is to determine the sentence type (e.g question, order, reply). This module uses a set of submodules and it depends on the speaker and addressee identification modules. All submodules handle one sentence type. The output of the submodules is the probability that a sentence is of a certain type. This module will then determine from the set of results the sentence type by means of statistical evaluation. The features were derived by analysing the scripts.

For each submodule there is a set of features the sentence types have. The probability computation is not yet fully specified, but the feature set is. Following is a list of submodules and their corresponding feature sets.

Instruction

- form of verb is imperative
- subject is not mentioned OR a 2nd person sg. pronoun
- no continuous form
- no 1st person sg.

Confirmation of Instruction

- object reference is the same as in preceding instruction
- previous instruction is necessary
- if a pronoun is used it has to be checked if the pronoun fits to aforementioned objects
- most confirmations have a subject in 1st person sg.
- usage of expressions of confirmation (e.g. done, yes, yes sir, roger)
- repetition of instruction

Question

- verb (or aux. verb) is in 1st position of the expression OR
- expression starts with w-word (e.g. what, where)
- answer follows
- possibly transcribes question mark at the end of an expression

Answer

- preceding question
- same object reference as in question
- may contain answer phrases (e.g. yes, no, ok, is "adj", "adj")

Ratification of an Action

- can only follow an instruction, a question or a machine generated message
- verb form is mostly simple past
- subject is 1st person sg. or pl.
- may contain ratification phrase (e.g. done, ready, checked, set)
- a state-value couple is given (e.g. altitude (is) 2000ft, heading (is) 239, flaps down)

General Proposition

- all pieces of information that are given unrequested

3.6 *Analysis of the Topic*

This module works on single expressions. It does not depend on other modules' results. Every sentence is assumed to have a topic. Topics that cannot be identified as being ATC related are considered "off". All possible ATC topics are predefined and were gained by analysis of the ATC transcripts (see Attribute Grammar, CONTEXT).

The identification of an expressions topic makes use of the attribute grammar. By means of attribute value unification known phrases are to be identified. Additionally we take into account ATC regulations that define a set of rules for well-formed expressions.

3.7 *Context-Logging*

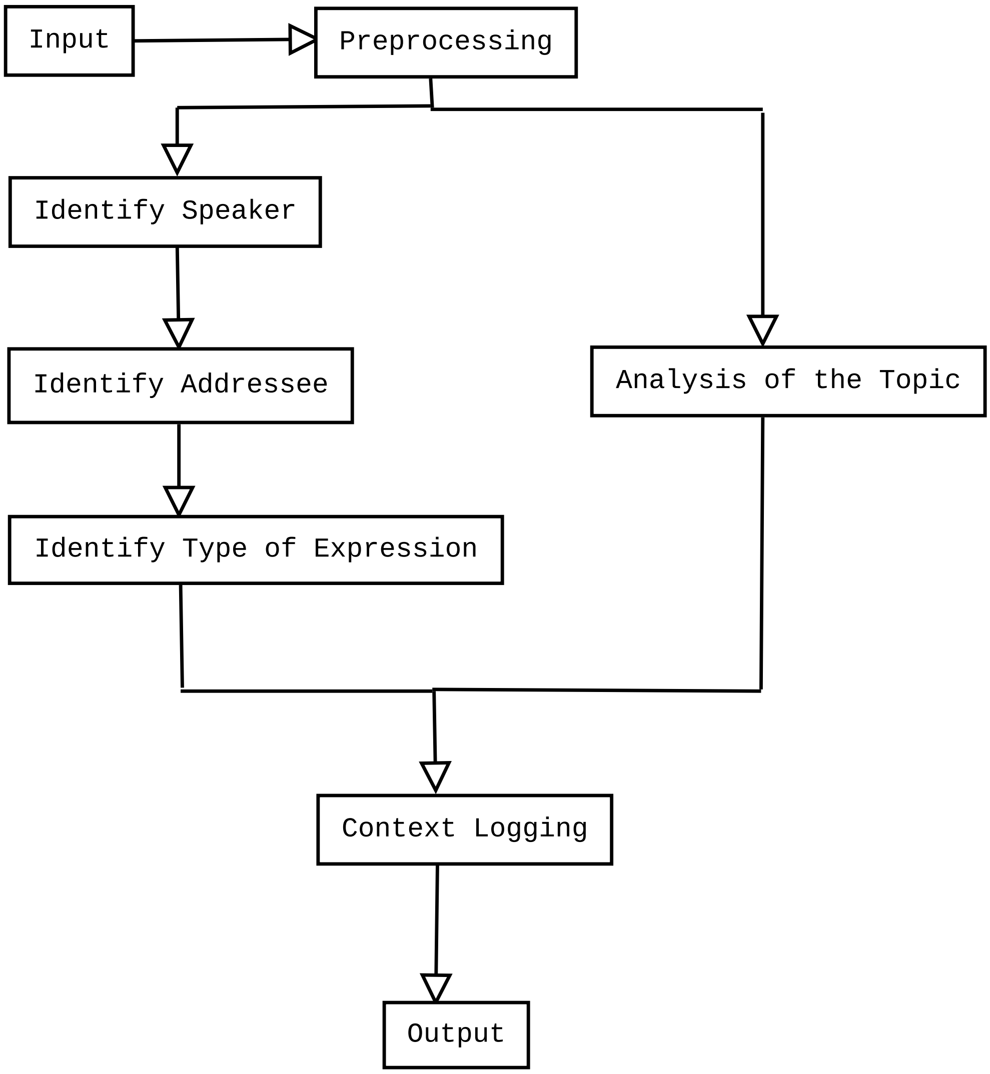
This module works on singe expressions. It depends on all the aforementioned modules. It's purpose is to store all gained data and provide access to it. Data collected after the computation of an expression is passed to the state machine. The context logging module will also annotate related expressions (e.g. Questions - Answered, Instructions - Confirmations). All will be formed to an annotated document containing all information of the transcripts and all data that was developed during analysis. All this information must be unambiguous. Therefore the development of an logging grammar may be necessary.

3.8 Output And Output Format

This is only relevant for a software implementation.

The output format of the log will be a well specified XML document.

3.9 Diagraph


Input
Preprocessing
Identify Speaker
Identify Addressee
Analysis of the Topic
Identify Type of Expression
Context Logging
Output

4 Testing And Results

4.1 The Procedure

We evaluated the design by parsing the transcripts by hand according to the algorithm.

We did all the processing according to our specification. We read sentence by sentence, counted every word and marked every keyword that is in our grammar. Then we tried to determine out speaker, addressee, objects, type of information and topic. After that we marked a sentence as interpreted or not. We used object references of sentences before and after the current one to get the topic and objects. We updated the flight mode where necessary. The output was a tabular with all results in one column for each sentence.

We expected to have a success rate between 30% and 40% using the aforementioned test. Our success rate, depending on the transcript, is between 48% and 77%. Although we expect to get lower success rate for an implemented software system. The reason for that is that humans will subconsciously use context and knowledge about the world in the analysis.

Before we performed another manual test of our interpreter design, we checked the transcripts intuitively. This way we want to build a basis for comparing the algorithm with other means of interpretation (as out own interpretation). We chose to check the Warsaw, the Nagoya and the Puerto-Plata accident script. These seem to be different enough from each other.

Most part of our intuitive analysis consists of dialogue structures (action - response) and semantic/pragmatic interpretation.

We found that an intuitive analysis heavily relies on past *and* future context. Whenever there is no obvious interpretation of an expression conclusions can be drawn from expressions long before or after the examined one. We will try to implement this sort of behaviour at least approximately. The interpreter should be able to analyse the *close* environment of the examined expression.

Our intuition easily identifies pragmatics of expressions regardless of syntactics or semantics. E.g. orders formed as questions, uttered keywords as orders and/or confirmations. Syntactics and semantics can be more easily recognised than pragmatics. A possible approach to this problem may be a semantic recogniser (that uses a syntactic one) to classify expressions. A pragmatic recogniser may then use available semantic, syntactic, role model, flight state and (close) environmental information.

Some expressions indicate difficulties/problems or its absence. If problems have been noticed the crew does not always know or mention it. In some expressions the use of keywords mark difficulties/problems. In others one can *read between the lines* that there must be something wrong. These last are very difficult to recognise. Maybe we will find some way to improve the recognition rate of problem indicating phrases. Even harder than these is the recognition of expressions the crew uses to express the absence of problems. Some expressions mention unusual behaviour with explicitly stating that there is no problem seen yet.

4.2 The Results

4.2.1 Birgen Air Accident

This is the first completely checked transcript. It has 92 sentences with 272 words. We have 144 words in our keyword list and interpreted 57 of the 92 sentences. That is 62% of the sentences.

4.2.2 Nagoya Accident

This transcript is 273 sentences long. 145 sentences are interpretable by our system. That is 53% of the sentences. The word count is 1489 words of which 718 are known. That is 48% of the words.

When we first looked at the script it seemed that the ratio would be a lot smaller. The cause for this seemingly high hit ratio is as follows. The not interpretable sentences are very long and stretch over many lines. When both Captain and 2nd Pilot are concentrating on the job they talk a lot less off topic. The interpretable sentences are large in numbers but do not cover much of the script length. Because we decided to estimate our interpreter by counting expressions (one expression = everything after the speakers mark) the hit ratio is higher than expected.

4.2.3 Airborne Express Flight 827

The Transcript is 82 sentences long. 49 sentences are interpretable by out system. That is 59% of the sentences. The transcript could have a very much higher hit ration if things on board would be talked about less casually. 453 of the 488 words were recognised.That is 92%

4.2.4 UN Flight KSV 2375

The transcript is 50 sentences long. 35 sentences are interpretable. That is 70% of the sentences. 288 of 322 words were recognises.

4.2.5 Lufthansa Flight 2904

The transcript is 109 sentences long. 85 sentences are interpretable. That is 77% of the sentences. 323 of 352 words were recognised.

5 Problems And Methods Of Resolution

5.1 Problems and possible solutions

Before analysing the transcripts we expected that problems would arise. Some expressions are not interpretable with our standard concept so it needed an extension. We concentrated on finding problems where expressions were on topic. None of the problem handlers is yet fully specified. It will be one of our future task to design a specified problem-handler for all problem-categories.

5.1.1 Problem-Handlers

Here is a list of problem-handlers for yet known problems. Not any of them is yet specified.

- for checklists
- "high-keyword-density" handler. Some sentences contain a lot of keywords and several topics. Here a problem handler will be needed to simplify the expressions.
- "long-sentence" handler. Long sentences with many topics (one following another) need to be split into sub sentences all containing just one topic.
- "additional-aircraft" handler. A solution has to be found when there are more than just one aircraft an ATC. Radio transmission of other aircraft will have to be interpreted, too.

6 Future Perspectives

For future tests of our interpreter design, analysis software tools will be needed. To broaden our perspective more scripts need to be tested. Otherwise we risk concentrating too much on too little or spending an extraordinary amount of time on analysing scripts. The usage of analysis software will provide more objective data than human testing. The tools can indicate the chances and restrictions of machine interpretation and provide a basis for the actual design. In a second stage the software generated data can be made subject to machine interpretation too (e.g. statistics, probabilities, datamining).

References

[1] Aviation-Safety-Network: http://aviation-safety.net

[2] Nagoya-Accident-Report:

http://www.rvs.uni-bielefeld.de/publications/Incidents/DOCS /ComAndRep/Nagoya/nagoyarep/nagoya-top.html#index

[3] Vnukovo-CVR-Transcript and report:

http://aviation-safety.net/cvr/cvr_vko2801.htm

[4] Airborne Express CVR-Transcript and Report:

http://aviation-safety.net/cvr/cvr_abx827.shtml

[5] Lufthansa CVR-Transcript and Report:

http://aviation-safety.net/cvr/cvr_lh2904.shtml

[6] Birgen Air

http://www.rvs.uni-bielefeld.de/publications/Incidents/DOCS/ FBW.html#B757-BirgenAir

[7] Aeronautical Information Manual / Federal Aviation Manual, McGraw-Hill Companies, Inc, Charles F. Spence