Automated data processing of aviation occurrences reporting

Maksym Liubarets[1,*,†] and Ivan Ostroumov[1,†]

[1] National Aviation University, Liubomyra Huzara Ave., 1, Kyiv, 03058, Ukraine

Abstract

Detailed study of aviation occurrence is a crucial measure to provide the required level of flight safety. Results of aviation occurrence investigation should identify factors that affect flight safety and provide a detailed study of these factors' impact on degradation of flight situation. Outcomes of aviation occurrence investigation are published in specific reports which include safety recommendations for preventing such occurrence in the future. Aviation occurrence investigation is provided by a certified authority, which accumulates all possible data about a particular aviation event at the beginning stage. A gap between the time an occurrence happened and the time of an investigation's start might significantly affect an investigation's quality and overall success. In the study, we consider designing a specific software for automation of reporting of aviation occurrences to minimize a delay before an investigation starts. It is supposed to provide an authority with initial information, including occurrence class, which is classified by proposed algorithm. Proposed software structure collects all the reports from different sources, analyzes them, filters out irrelevant data, and notifies personnel about an occurrence. An application runs as a service and is highly configurable, flexible, and robust.

Keywords

software, automation, aviation occurrences reporting, incidents, accidents, aviation safety, air navigation

1. Introduction

The safety of air transportation depends on multiple factors action. Ensuring the safety of aviation is based on identification of dangerous factors and minimization of its action on the aviation system. The safety of aviation is grounded on a probability-based approach which minimizes the risk of dangerous state occurrence. In case, aviation occurrence has taken place it means that some factors affect flight safety and it should be identified and action should be minimized to prevent the appearance of these factors action in the future. The most significant factor is human [1, 2]. Also, simple failure in the system has a high rate of occurrence [3, 4]. Any aviation event which affects flight safety may be related to Aviation occurrence (AO) and could be classified based on the level of impact on humans into incidents and accidents [5, 6]. Accident means significant safety reduction which causes serious injuries to people. All other safety events could be classified as incidents. An example of incident could be any onboard equipment malfunction that leads to a violation of minimums of separation between airplanes [7, 8].

Aviation occurrence reporting is essential for ensuring continuous improvements in aviation safety. It provides valuable data for investigating incidents and accidents, helping to shape safety protocols and regulations.

Mandatory reporting systems (MORS) are required by aviation regulations, ensuring that specific incidents are systematically reported. The International Civil Aviation Organization (ICAO) outlines global standards for reporting aircraft accidents and serious incidents [6]. It mandates the reporting of occurrences that involve aircraft accidents and serious incidents that could have led to accidents.

ADP'24: International Workshop on Algorithms of Data Processing, November 5, 2024, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

m.lyubarets@gmail.com (M. Liubarets); ostroumov@ukr.net (I. Ostroumov)

0009-0002-8986-3840 (M. Liubarets); 0000-0003-2510-9312 (I. Ostroumov)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

CEUR-WS.org/Vol-3895/paper10.pdf

The Federal Aviation Administration (FAA) requires the reporting of AOs such as engine failures, equipment malfunctions, and runway incursions [9]. The European Union Aviation Safety Agency (EASA) governs reporting through [10], which mandates the submission of occurrences related to safety issues, air traffic management, and technical defects. Pilots, air traffic controllers, airline operators, maintenance personnel, and manufacturers are responsible for reporting serious incidents and accidents, typically within 48–72 hours of occurrence. Reportable events might include: engine or system failures, runway incursions - unauthorized presence on the runway, near mid-air collisions and loss of separation, hazardous operational conditions - flight into restricted airspace. Mandatory reporting data is stored in national databases: FAA Aviation Safety Reporting System (FAA ASRS) and ECCAIRS. This data supports investigations into the root causes of incidents, provides trends in aviation safety, and shapes future regulations.

Voluntary reporting systems (VORS) gather additional safety information that may not meet mandatory criteria but can still pose potential risks. Key programs are:

- FAA's ASRS [11] encourages confidential reporting of safety-related issues, offering immunity from FAA enforcement in most cases;
- EASA's Voluntary Reporting System [12] encourages reporting under a Just Culture framework, ensuring anonymity unless gross negligence is involved;
- CHIRP (UK) The Confidential Human Factors Incident Reporting Program [13] for reporting of human factor issues.

Voluntary reporting covers safety risks, such as minor human errors, technical malfunctions, or lapses in communication, providing insights into potential hazards that may not be captured by mandatory reporting. A key principle of VORS is anonymity, allowing aviation personnel to report without fear of punitive action, provided no willful misconduct is involved. Voluntary reporting helps improve aviation safety by providing human factors analysis, identifying latent hazards before they escalate into serious incidents and supporting proactive safety measures. All these automatic systems include robust algorithms [14, 15] which could provide basic static analysis and report generation.

Both mandatory and voluntary reporting systems are integrated in aviation occurrence investigations and play a critical role in investigation process. Reports from these systems are essential for initial data collection, helping investigators understand the chain of events, system failures, and human errors contributing to an occurrence [16, 17]. This data is cross-referenced with other evidence to support the investigation. Data from mandatory and voluntary reports helps investigators detect wider safety trends, resulting in safety recommendations aimed at preventing future occurrences. The analysis of reports leads to the issuance of safety recommendations, such as airworthiness directives and operational bulletins, and helps to improve training and risk management systems.

In Europe, any volunteer can submit an AO report via ECCAIRS2 – SRIS2 [18]. A submission is being performed in four basic steps:

1. Select a state to report in, a personal or organization behalf.
2. Fill in a reporting form including aircraft identification fields: State of registry, Aircraft registration, Aircraft category, Manufacturer/model, Serial number, Operator, Year built, Call sign; flight details fields; Last departure point, Planned destination, Operation type, Flight phase, Current flight rules, Occ. on ground; Narrative details fields; Narrative language, Narrative text; Airspace fields details Airspace type, Airspace class.
3. Attach data files.
4. Fill personal details optionally.

ECCAIRS states that there are over 260 safety recommendations per year, the database holds 2.3 million occurrence reports and is being used by 65 authorities.

MORS and VORS form the backbone of aviation safety management by providing essential data for reactive investigations and proactive safety measures [19]. The combination of mandatory regulations and voluntary participation fosters a culture of safety that encourages learning from errors and improving industry-wide safety standards.

In Ukraine, the National Transport Investigation Bureau (NTIB) is responsible authority for conducting technical investigations of AOs, including serious incidents, incidents, emergencies, damage to civilian aircraft and aircraft on the ground, and violations of the airspace utilization procedure [20, 21].

In a period from 2018 until 2023, NTIB has received 391 AO reports. All the received reports led either to NTIB's investigations conducted or participated. A statistic of AO received reports is shown in Figure 1.

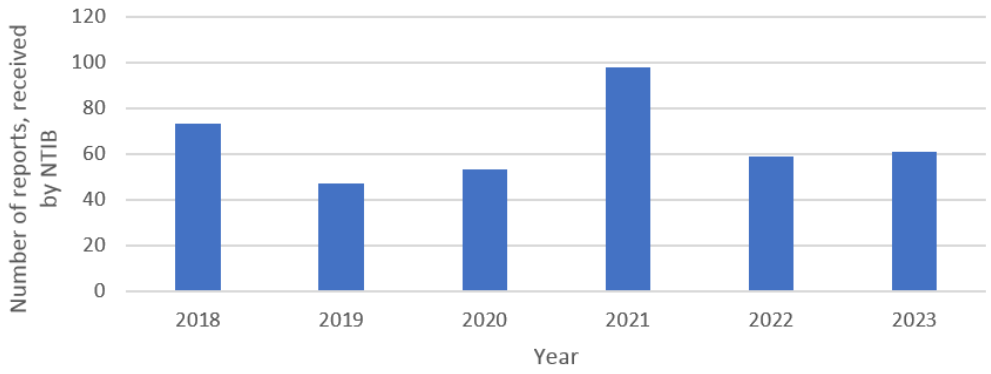

Figure 1: AO reports received by NTIB.

AO reporting is being regulated by [22]. There are few options of reporting [23]:

- via official NTIB email, providing information within email's body or in an attached document;
- via official NTIB website occurrence report electronic form [24];
- by fax;
- via aeronautical fixed telecommunication network (AFTN);
- via phone, providing information to an operator on duty.

In this paper, we study the software architecture of a complete solution for automation of AO reports receiving and storing. Additionally, the algorithm of classification of AO type, based on keyword matching, is proposed. A key feature of suggested approach is to process all the types of incoming AO reports by a single automated system that prepares and stores an AO's information providing a possibility to conduct an investigation immediately, based on processed data.

2. Automation of receiving and storing of aviation occurrence reports

Proposed automated system of AO reports receiving and storing is a standalone software module of Automated Management System for Accidents Investigation (AMSAI). Notification Service (NS) is a standalone module of AMSAI. We use Delphi programming language to develop AMSAI. Software compiled with Delphi is fast as it is an exe-file in native machine code. It is almost the same speed as C++, but with the advantage of not dealing with error-prone pointers. As Delphi does not have any dependencies like C#, it can be easily run on different workstations. There are many high-quality components for Delphi that make it easier to develop. NS is designed to run as a Windows service. Structure scheme od developed software is shown in Figure 2.

Raw AO reports income from following sources:

- Phone. In case an NTIB operator is on duty, he receives AO report, finds out all the details and manually creates a record in the AMSAI's database (DB) via web-interface of administrator's module. Otherwise, an answering machine records a report in .wav audio format [25] and stores it in a specific directory of the file server;
- AFTN. All messages are being converted in plain text format and being stored in a specific directory of the file server;
- Fax. Incoming messages are being converted in pdf format and being stored in a specific directory of the file server;
- E-mail. NTIB has predefined AO reporting e-mail address which is utilized for receiving all messages related to AO reports;
- Website. NTIB has a web-page where AO reports can be submitted via electronic form. After submission, AO report data are being forwarded via email to AO reporting email address.

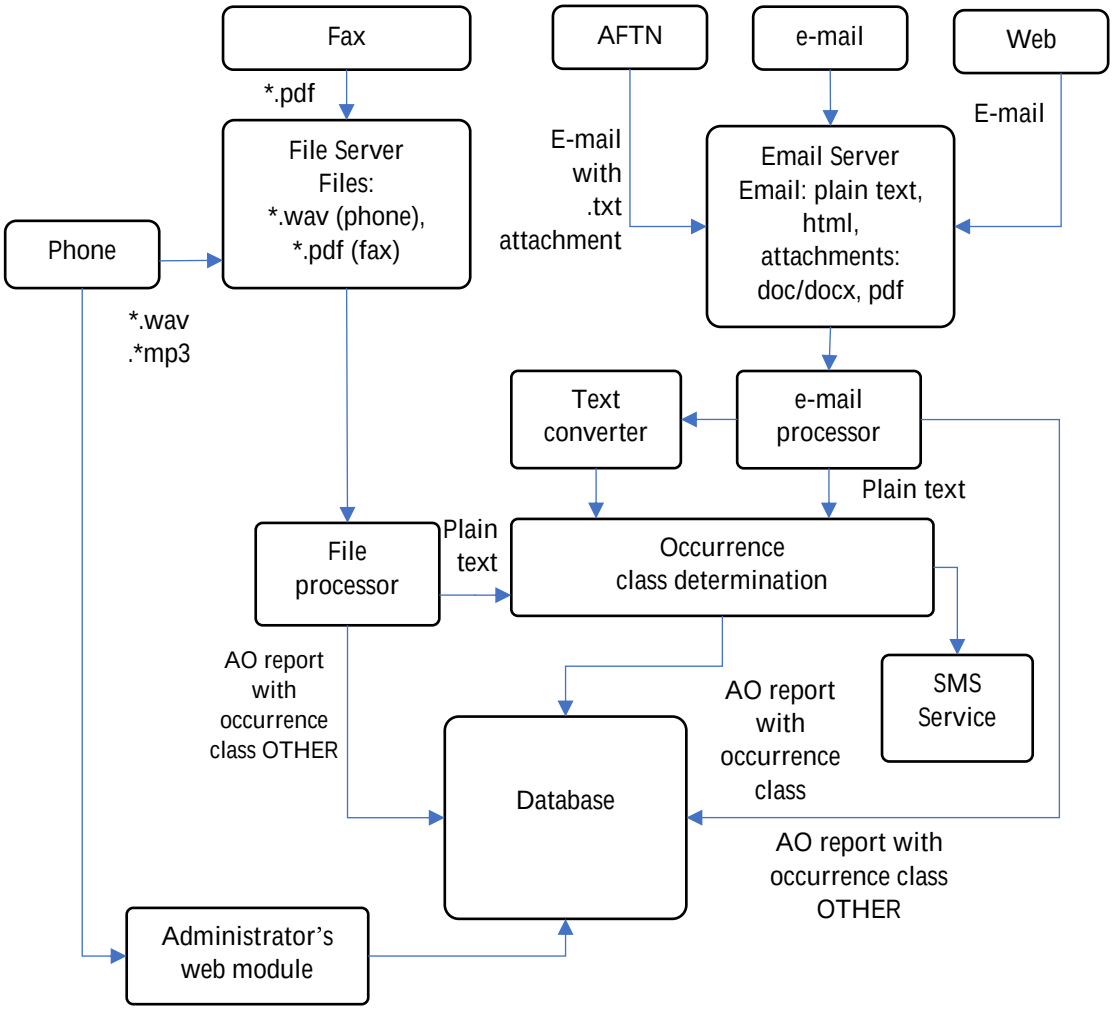

Figure 2: Structure scheme of AMSAI's Notification Service.

2.1. Database infrastructure

In AMSAI, a Firebird database (DB) engine has been chosen to operate with, as this engine is fast, open source and reliable [26]. It utilizes the Multi-Generational Architecture, which allows to keep the mark on the original record versions, leaving them untouched, in case if a transaction is rolled back. As a result, Firebird disk writes are very reduced compared to databases that use the traditional transaction log architecture. Due to performance reasons, AMSAI's DB infrastructure includes two separate databases: main DB and notifications DB, where all the big raw notifications texts are stored.

As they require much space, it was decided to store the amounts of plain text data apart from main DB.

Notifications DB includes a single table AO_TEXTS with fields: ID is a unique identifier (UID) of type Integer; NTF_ID is a foreign key referencing NTF table in main DB, AO_TEXT is a raw AO report text of type Blob. NS database structure is represented in Figure 3. Tables descriptions are listed in Table 1.

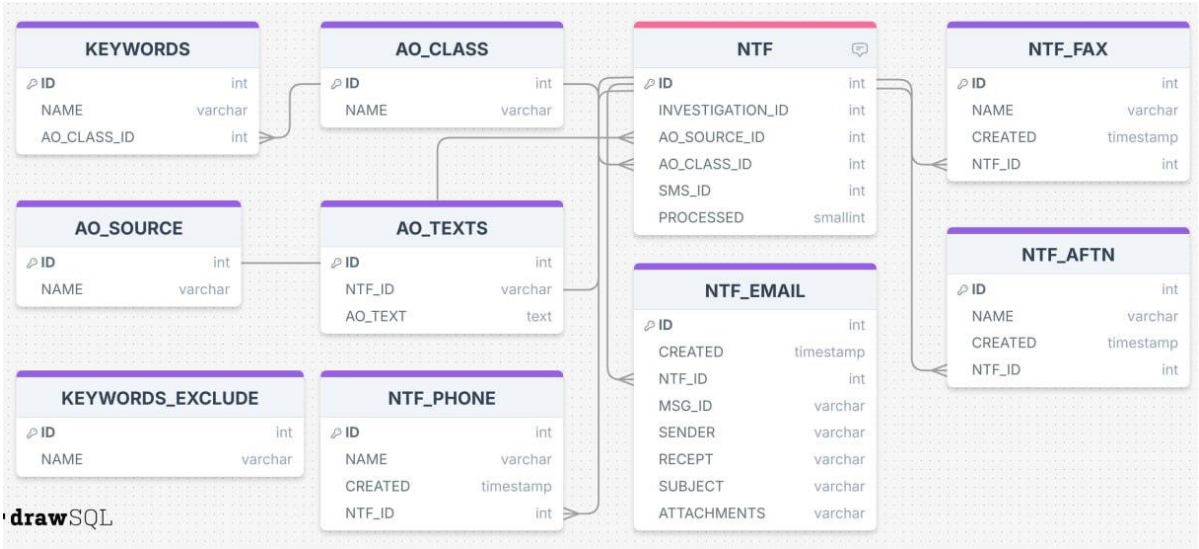

Figure 3: NS Database structure.

Table 1
NS database tables description

Table name	Description
KEYWORDS	Stores keywords to include for AO class identifying
KEYWORDS_EXCLUDE	Stores keywords to exclude for AO class identifying
AO_CLASS	Dictionary of AO class. Predefined content is follows: 1 is Accident (catastrophe); 2 is Accident; 3 is Serious incident; 4 is Incident; 5 is Other (Airspace violation); 6 is Other (undefined).
AO_SOURCE	Dictionary of AO report source. Predefined content is follows: 1 is AFTN; 2 is FAX; 3 is EMAIL; 4 is WEBSITE; 5 is PHONE
NTF	Main table for AO reports. Being reviewed by an administrator and can be a basis for a new investigation.
NTF_AFTN	Stores AO reports, received from AFTN
NTF_FAX	Stores AO reports, received by fax
NTF_PHONE	Stores AO reports data, received by phone
NTF_EMAIL	Stores AO reports data, received by email

2.2. File processor algorithm

File processor is represented as a procedure. The algorithm of the procedure is following: a given directory is being scanned for any files with extensions that correspond to directory type. For Phone directory the extensions are *.wav, *.mp3., for Fax directory is *.pdf. It is made by executing *FindFirst* method [27] with parameters of [directory name]*.[file extension], *faAnyFile* as an attribute and *searchResult* as a holder of attributes of first found file. *FindFirst* method returns 0 if at least file has been found. In this case *searchResult* variable will be set with file properties “name”, “size”, “time”

and “attrs”. A notification email is being created and sent to eligible staff. Then a number of SQL queries are being executed, which are represented in Table 2. Then file processor continues to search for files using *FindNext(searchResult)* method in a loop until the method returns a value other than zero. Finally, *FindClose(searchResult)* is being executed to clean up the resources, utilized earlier.

Table 2
SQL queries, executed by file processor

SQL Query text	Description
Insert into NTF (AO_SOURCE_ID, AO_CLASS_ID, PROCESSED) values (:ao_source_id, :ao_class_id, 0)	A new notification is being created with parameters: *ao_source_id* = 2 (Fax) or 5 (Phone) or *ao_class_id* = 6 (Other)
Insert into [:table_name] (NAME, NTF_ID) values (:NAME, :NTF_ID)	Storing data about processed file in corresponding table. Parameter table_name is set to ntf_fax or ntf_phone depending on the value of *ao_source_id* field. Parameter: *NTF_ID* is set by DB engine and is returned as a result of the previous SQL query.
Insert into AO_TEXTS (NTF_ID, AO_TEXT) values (:NTF_ID, :TEXT)	Storing a raw text of AO report file

2.3. Email processor algorithm

Email processor is also represented as a separate procedure. A connection to email server is being established by means of Indy POP3 component [28] *IdPOP31* with defined parameters host, port, username, userpassword. Method *IdPOP31.connect* is being executed and returns “True” if the component has connected successfully. *IdPOP31.CheckMessages* property stores a number of new received messages.

Each message is being retrieved be executing *IdPOP31.retrieve* method with parameters *messageNumber*, starting with 0 and ending with the total number of messages and *idMessage* as a holder of a retrieved message. After successful retrieval a property *idMessage.msgId* representing unique message identifier in a scope of email server, is taken and following SQL query is being executed to check if a message with given msgId has not been processed yet:

*select * from ntf_email where msg_id = idMessage.msgId*

If the query returns empty record set, procedure of email parsing is being executed on the next step of processing. Source of email can be identified analyzing *idMessage.Subject* property. If it equals to AFTN PRIMARY MESSAGE, a message is being processed as AFTN text message and parameter *AO_SOURCE_ID* is set to 1. In case it equals to NTIB AO REPORT, it is identified as NTIB website AO report message and parameter *AO_SOURCE_ID* is set to 4. Values of electronic form fields are being extracted from the email and *AO_CLASS_ID* parameter is set accordingly. In any other case a message is identified as mandatory or volunteer AO report.

Email message can have different structure. In most cases, it is a plain text. In this case a plain text data can be taken from *IdMessage.Body.Text* property. Alternatively, an email can include attachments. Total number of attachments is defined in *IdMessage.MessageParts* property. If the value is greater than zero, each part is being analyzed in a loop. Depending on a value of *ContentType* attribute of each message part, following actions are performed. *ContentType* attribute is defined in method *TIdAttachmentFile (IdMessage. MessageParts[N])*. Content data from each message part can be extracted from the property *IdMessage.MessageParts.Items[N]*. Options for this value are *TIdText* and *TIdAttachmentFile*. The former means that the content of *N*-th message part is a sequence of characters. The latter means that the content of *N*-th message part is a file. If *ContentType* equals *text/plain* and a message part is a plain text, no actions needed. In case of file, it is being saved to temporary directory, and then opened as text. If *ContentType* equals *text/html*, corresponding conversion from HTML to text, described in section 2.4, is being performed or file is being opened as text. If *ContentType* equals *application/pdf*, a pdf file is being opened as text file. If it is not encrypted, its content still can be analyzed. If *ContentType* equals *application/msword* or

application/vnd.openxmlformats-officedocument. wordprocessingml.document, a Microsoft office or OpenXML file is being saved in temporary directory and converted to plain text by text converter.

Next step of the algorithm is executed in order to identify *AO_CLASS_ID* parameter. After *AO_CLASS_ID* value is calculated, SQL queries represented in Table 3 are being executed.

Table 3
SQL queries, executed by email processor

SQL Query text	Description
Insert into NTF (AO_SOURCE_ID, AO_CLASS_ID, PROCESSED) values (:ao_source_id, :ao_class_id, 0)	A new notification is being created with parameters *ao_source_id* = 1 (AFTN) or 3 (Email) or 4 (NTIB Website), *ao_class_id = AO_CLASS_ID*
Insert into NTF_AFTN (NTF_ID, NAME, CREATED*) values (:NTF_ID, :FILE_NAME, now())*	In case if *ao_source_id* = 1, storing data about processed email with AFTN text file attachment. Parameter *NTF_ID* is set by DB engine and is returned as a result of the previous SQL query. Parameter *NAME* is taken from the file name of email attachment.
Insert into NTF_EMAIL (NTF_ID, MSG_ID, SENDER, RECEPT, SUBJECT, ATTACHMENTS) values (:NTF_ID, :MSG_ID, :SENDER, :RECEPIENTS, :SUBJECT, :ATT)	In case if *ao_source_id* = 3, storing data about processed email with email parts/attachments. Parameters *MSG_ID* is a message uuid set by email server, *SENDER* is a message sender, *RECEPIENTS* is message recipients, *SUBJECT* is a message subject, *ATT* is a list of attachments file names.
Insert into AO_TEXTS (NTF_ID, AO_TEXT) values (:NTF_ID, :TEXT)	Storing a raw text of AO report email

A plain text data from each message part or message body, in case *IdMessage.MessageParts* equals zero, is used as input parameter for aviation occurrence class identification algorithm (AOCIA). Default value of *AO_CLASS_ID* is set to 100, which is bigger than maximum value of 6. Execution of AOCIA for each email message part returns a number with minimum value of 1 for “Accident/catastrophe”, and maximum value of 6 for “Other/Undefined”. Final value is calculated according to (1).

$$AO\_CLASS\_ID = \min_{i=1\ldots N} AOCIA(i) \quad (1)$$

where *N* is a number of email message parts.

Finally, after all SQL queries are successfully executed, email processor deletes processed message on email server, by calling *IdPOP31.delete* method with the parameter of current message index.

2.4. Text converter algorithm

Text converter algorithm is represented by two functions, one is designed for processing HTML data, other processes files of doc/docx format. Both return a string, containing plain text: words, separated by space character. In case of converting HTML data to plain text the THtmlParser library is utilized [29]. Firstly, an HTML text is being converted to TDocument object, by executing *THTMLParser.Parse* method with a string parameter. Secondly, the result string is being compiled by executing *THtmlFormatter.GetHTML* method with a TDocument parameter.

- In case of converting Microsoft word document, built-in tool for processing such files is being used. Steps are following:
- Create MsWord application ole-object (*MSWordApp := CreateOleObject('Word.Application')*)
- Make it hidden (*MSWordApp.Visible := False*)
- Open a file (*MSWordApp.Documents.open(PathToFile)*)
- Get the number of characters to select (*charCount := MSWordapp . Documents . item(1) . Characters.Count*)
- Select entire text and retrieve the selection (*ResuIt:=MSWordApp.Documents.item(1).Range(0, CharsCount).Text*)
- Close the file (*MSWordApp.documents.item(1).Close*)

- Quit MsWord application (*MSWordApp.Quit*)

2.5. Aviation occurrence class identification algorithm

The approach of AO class identification is based on searching for keywords in a given plain text data, that are matched to a specific *AO_CLASS_ID* value. Keywords are preliminary defined in the table *KEYWORDS* and can be modified by an administrator later. For catastrophe, the list of words is died, catastrophe, fatal, dead. For accident - crash, accident, injured, accid, emergency. For serious incident - fire, serious incident. For incident - incident, emergency landing, bird, For other (airspace violation) - airspace violation, airspace infringement. For other (undefined) - accredited representative, investigator in charge, investigation, occurrence, failure, notification. Additionally, excluded keywords notion is introduced. The goal of excluded keywords is if the algorithm identifies AO class as 1 – 6 and at least one of excluded keywords is found, AO class will be reset and no further processing is needed. This will ensure that service messages that are not related to any AO will be filtered out. Table's *KEYWORDS_EXCLUDED* content should be compiled according to an internal requirement of an investigation authority, and as well can be edited by an administrator at any time.

Due to the majority of AO reports are specific, there is no need to implement a complex semantic algorithm, which would calculate semantic similarity, capturing inflections, synonyms or word's related forms. For a simpler method that handles different word forms (such as pluralization, verb conjugations, or inflections), without dealing with synonyms or deeper semantic meaning, stemming or lemmatization would be the best approaches.

Since Delphi doesn't have built-in support for advanced natural language processing (NLP), it was decided to focus on stemming, which is easier to implement than lemmatization. Stemming reduces a word to its base or root form by removing suffixes (e.g., running and run, cars and car). The goal is to implement a basic stemmer in Delphi, allowing for word search in texts that can handle various forms of a given word by reducing it to its root. The Porter Stemmer [30] is one of the most widely used algorithms for stemming in NLP and is relatively simple to implement. It handles common word forms like pluralization and verb conjugations.

Using defined functions, the final AOCIA is represented as following:

- Temporary variable *tempClassEvent* is set to 100;
- Keywords and exclusions are extracted from database. The order of keywords is set to descending by field *AO_CLASS_ID*, starting with those that are matched with AO class highest value of 6 (Other/Undefined) and ending with lowest value of 1 (Accident/catastrophe);
- Search for each keyword in a loop is being performed. If keyword is found and field's *AO_CLASS_ID* value is less than *tempClassEvent* - the value of *tempClassEvent* is set to *AO_CLASS_ID value*;
- After all keywords are processed, the variable *tempClassEvent* holds the lowest value of AO class matching values that were found;
- If *tempClassEvent* < 6, exclusions need to be processed. Similar to previous step, all the exclusions are being searched in given plain text data. If at least one is found, *tempClassEvent's* value is being reset to 6, which means that a given AO report should be filtered out and not be processed;
- If *tempClassEvent* = 100, it means that no keywords were found in the text. *tempClassEvent's* value is being set to 6 (Other/Undefined);
- *tempClassEvent* variable is assigned as the result value of the algorithm.

2.6. Email and SMS notifications of staff, logging

NS has a feature of getting NTIB staff to be notified about incoming AO reports. After each AO report has been processed, a notification email is being sent to a specific email-address. Detailed description

of SMS sending process is out of scope of this study. It's worth mentioning, that in case identified *AO_CLASS_ID* value <=2, SMS is being sent immediately to all the related persons, otherwise SMS will be sent by schedule.

Also, NS provides detailed logging of all conducted actions. Log files are created circularly, as one file at each date.

As it implies from Table 3, in all cases one record has been added into NTF table and in each case a notification email has been sent. Depending on *AO_SOURCE_ID*, AO report data has been stored in related tables. In third case, as AO class was identified as accident, additionally, an SMS message has been sent to the related NTIB staff. Fourth case is represented in Figure 4. It can be seen that one SMS notification has been sent to a person who was on duty at the time, not immediately, but later, according to a schedule; the contents of AO report; the contents of notification email, which was sent to a responsible person, and a list of keywords that resulted to AO class identification.

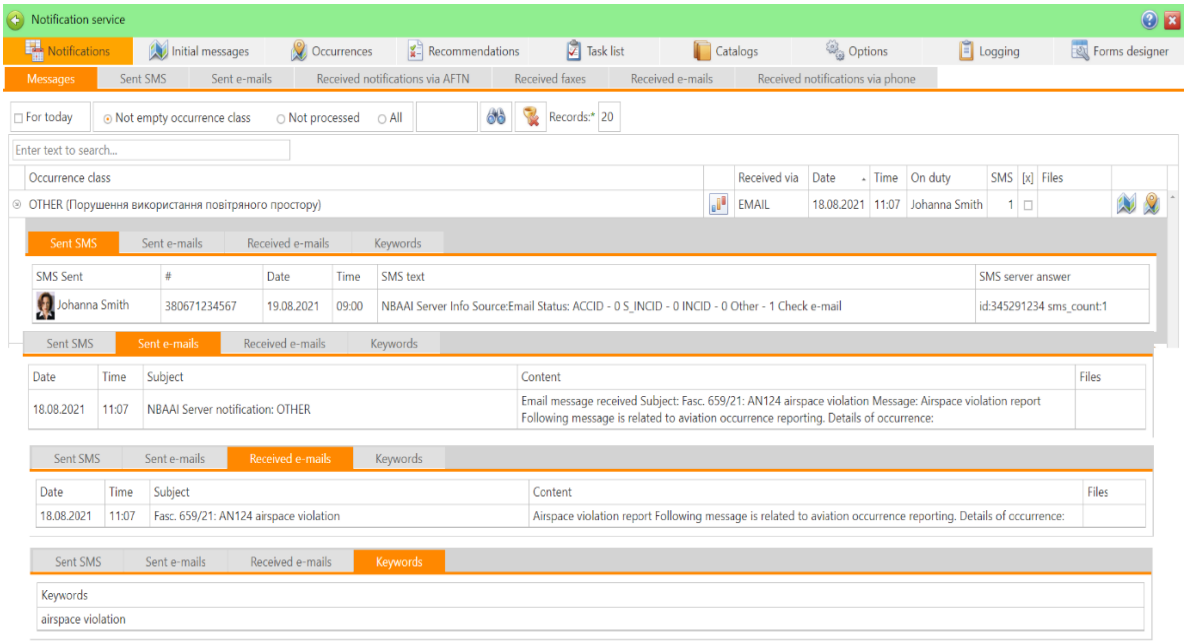

Figure 4: AMSAI's user interface for reviewing AO reports.

AMSAI's user interface provides a possibility for an administrator to review all registered AO reports and create investigation cards, based on a decision whether a specific notification requires conducting of an investigation. An administrator is able to review a list of AO reports along with information about identified AO class, source, date and time, who was on duty at the time an AO report has been received, a number of SMS sent, all the related files. It is an option to change AO class permanently if it was identified by AOCIA wrongly. Once an AO report has been processed, it can be marked as done. Also, it is possible to create an investigation card based on the information provided by AO report.

3. Results

All the described algorithms have been executed with different input parameters. A list of input parameters and output data are presented in Table 4.

Table 4
Input parameters and output results of AO reports processing algorithms

AO report source	AO report text	Output values		Additional information
		AO class (AO_CLASS_ID)	AO Source (AO_SOURCE_ID)	
AFTN	XAA0129 230358 FF OPLAZIZX OPLRZQZX OPKCZRZA OPKCZIZX OAKXZQZX 230356 VTBDZPZX (FPL-EVA67-IS -B744/H-SHIJDRYWZG/S -VTBD-N0514F300 FRANK2 LIMLA L507 CEA/M084F300 R460 TEPAL/N0507F300 R460	6	1	Records added in tables: NTF, AO_TEXTS, NTF_AFTN. Email "AFTN message received" was sent.
FAX	Pdf file	6	2	Records added in tables: NTF, AO_TEXTS, NTF_FAX. Email "Fax message received" was sent.
EMAIL	NOTIFICATION Notification from Investigator-In-Charge of Aircraft Accident and Incident appointed by the Minister of Justice of the Republic of _____ regarding an accident or serious incident in accordance with Chapter 4 of ICAO Annex 13 and Article 9 para 2 of Regulation (EU) No 996/2010.	2	3	Records added in tables: NTF, AO_TEXTS, NTF_EMAIL. Email "Email message received" was sent. SMS with notifications report to the current staff on duty was sent immediately.
EMAIL	Airspace violation report Following message is related to aviation occurrence reporting. Details of occurrence:	5	3	Records added in tables: NTF, AO_TEXTS, NTF_EMAIL. Email "Email message received" was sent. SMS message has been scheduled.
PHONE	Record.wmv file	6	5	Records added in tables: NTF, AO_TEXTS, NTF_PHONE. Email "Phone message received" was sent.

4. Discussion

Proposed algorithm of aviation occurrence class identification is a simplest version of nowadays AI-related NLP algorithms based on word embeddings, where words with similar meanings or forms are represented by vectors in a continuous space. The most common methods for generating embeddings include using Word2Vec, GloVe, or BERT models. Semantic similarity can be easily calculated using Python programming language. Whereas, suggested implementation covers most of the requirements of AO reporting in Ukraine. Definitely, the preliminary contents of keywords are quite basic. However, a possibility to adjust the list of keywords and additionally regulate AOCIA by exclusions, makes it possible for the administrator to filter out unneeded notifications and to make the algorithm identify AO class more precisely.

Overall functionality of NS provides stable processing of all incoming AO reports from different sources. In the case if NS does not work, or one of its components is down, the administrator can be aware of that since NS reports of self-availability directly to a specially designed monitoring module of AMSAI. In this case the administrator will be notified by email, that one of the services is not working, and conduct appropriate measures in order to restore NS functioning.

Limitations of proposed algorithms can be defined as following:

- types of email attachments are limited to pdf and two types of documents, .doc/.docx, which reduces NS' flexibility to handle files of various formats;

- NTIB email address for AO reports is supposed to be used only for AO reporting. In case this email is used for wide purposes, NS can create a lot of records with *AO_CLASS_ID* = 6. This might prove difficulties for an administrator to process all incoming AO reports on everyday basis;

Further research will mostly be focused on enhancing of semantic algorithms of AO class identification and extending of email processor capabilities on processing new attachment file types.

5. Conclusions

Developed software provides automatic processing of all submitted mandatory and volunteer aviation occurrences reports within a single automated system. It handles data received from different sources: phone, fax, AFTN, email and website. Notification service runs in a background as Microsoft Windows service, is highly configurable, flexible and robust. The performance of the software is high as it's been compiled into native machine code.

As additional feature, it notifies NTIB staff about any incoming AO report by means of SMS and email. Preliminary determination of aviation occurrence class allows NTIB experts to react quickly on such important reports as accidents and catastrophes. This leads to a possibility to conduct an investigation as earlier as possible, which, in its turn, increases chances to gather needed evidence, determine the factors that caused such occurrence, issue a number of recommendations, implementing of those will prevent happening of such occurrence in the future and contribute a lot to the global flight safety.

References

[1] O. Solomentsev, M. Zaliskyi, O. Sushchenko, Y. Bezkorovainyi, Y. Averyanova, Data Processing through the Lifecycle of Aviation Radio Equipment," in: proceedings of the IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 2022, pp. 146–151, doi: 10.1109/CSIT56902.2022.10000844.

[2] M. Zaliskyi, O. Solomentsev, V. Larin, Y. Averyanova, N. Kuzmenko, Model Building for Diagnostic Variables during Aviation Equipment Maintenance, in: proceedings of the IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 2022, pp. 160–164. doi: 10.1109/CSIT56902.2022.10000556.

[3] I. V. Ostroumov, N. S. Kuzmenko, Risk Analysis of Positioning by Navigational Aids, , in: proceedings of the Signal Processing Symposium (SPSympo), Krakow, Poland, 2019, pp. 92–95, doi: 10.1109/SPS.2019.8882003.

[4] M. Zaliskyi, O. Solomentsev, O. Holubnychyi, I. Ostroumov, O. Sushchenko, Yu. Averyanova, Y. Bezkorovainyi, K. Cherednichenko, O. Sokolova, V. Ivannikova, R. Voliansky, B. Kuznetsov, I. Bovdui, T. Nikitina, Methodology for substantiating the infrastructure of aviation radio equipment repair centers, CEUR Workshop Proceedings, 3732 (2024) 136–148. URL: https://ceur-ws.org/Vol-3732/paper11.pdf.

[5] T. Dong, Q. Yang, N. Ebadi, X. R. Luo, P. Rad, Identifying incident causal factors to improve aviation transportation safety: Proposing a deep learning approach. Journal of advanced transportation (2021) 1–15. doi: 10.1155/2021/5540046.

[6] Aircraft accident and Incident Investigation, Annex 13 to the convention on International Civil Aviation, ICAO, 2016.

[7] I.V. Ostroumov, N.S. Kuzmenko, Accuracy improvement of VOR/VOR navigation with angle extrapolation by linear regression. Telecommunications and Radio Engineering, 78(15) (2019) 1399–1412. doi: 10.1615/TelecomRadEng.v78.i15.90.

[8] I.V. Ostroumov, N.S. Kuzmenko, Accuracy assessment of aircraft positioning by multiple Radio Navigational aids. Telecommunications and Radio Engineering, 77(8) (2018) 705–715. doi: 10.1615/TelecomRadEng.v77.i8.40.

[9] General Operating and Flight Rules. Title 14 CFR Part 91, 2022. https://www.ecfr.gov/current/title-14/part-9.
[10] Regulation (EU) No 376/2014 on the reporting, analysis, and follow-up of occurrences in civil aviation, 2014. https://www.easa.europa.eu/document-library/regulations/regulation-eu-no-3762014.
[11] Aviation Safety Reporting System, NASA.,2021, https://asrs.arc.nasa.gov.
[12] Voluntary reporting systems under Just Culture, European Aviation Safety Agency, 2021, https://www.easa.europa.eu/document-library/general-publications/just-culture.
[13] CHIRP. Confidential Human Factors Incident Reporting Program, 2023, https://chirp.co.uk.
[14] T. Nikitina, B. Kuznetsov, Y. Averyanova, O. Sushchenko, Method for Design of Magnetic Field Active Silencing System Based on Robust Meta Model, in S. Shukla, H. Sayama, J.V. Kureethara, D.K. Mishra, (Eds.) Data Science and Security. Lecture Notes in Networks and Systems, Springer, Singapore, 2024, vol. 922, pp. 103–111, doi: 10.1007/978-981-97-0975-5_9.
[15] B. Kuznetsov, I. Bovdui, T. Nikitina, R. Voliansky, O. Sushchenko, Y. Bezkorovainyi, Y. Averyanova, Efficiency of overhead power lines magnetic field combined electromagnetic silence. Proceedings of SPIE, 13279 (2024) 1327907. doi: 10.1117/12.3043769.
[16] T. Nikitina, B. Kuznetsov, N. Ruzhentsev, O. Havrylenko, K. Dergachov, V. Volosyuk, O. Shmatko, A. Popov, Algorithm of Robust Control for Multi-stand Rolling Mill Strip Based on Stochastic Multi-swarm Multi-agent Optimization, in S. Shukla, H. Sayama, J.V. Kureethara, D.K. Mishra, (Eds.) Data Science and Security. Lecture Notes in Networks and Systems, Springer, Singapore, 2024, vol. 922, pp. 247–255, doi: 10.1007/978-981-97-0975-5_22.
[17] I.V. Ostroumov, N.S. Kuzmenko, Compatibility analysis of multi signal processing in APNT with current navigation infrastructure. Telecommunications and Radio Engineering, 77(3) (2018) 211–223, doi: 10.1615/TelecomRadEng.v77.i3.30.
[18] Report an occurrence, ECCAIRS2 – SRIS2. 2024, https://e2.aviationreporting.eu/reporting/unregistered.
[19] O. Holubnychyi, M. Zaliskyi, O. Sushchenko, O. Solomentsev, Y. Averyanova, Self-Organization Technique with a Norm Transformation Based Filtering for Sustainable Infocommunications Within CNS/ATM Systems, in: I. Ostroumov, M. Zaliskyi (Eds.), Proceedings of the International Workshop on Advances in Civil Aviation Systems Development. Lecture Notes in Networks and Systems, Springer, Cham, 2024, vol. 992, pp. 262–278. doi: 10.1007/978-3-031-60196-5_20.
[20] About State Program of civil aviation safety program, 21.03.2017 No. 1965, Verkhovna Rada of Ukraine, https://zakon.rada.gov.ua/laws/show/1965-19.
[21] Manual of Aircraft Accident and Incident Investigation Part II Procedures and Checklists, Doc 9756 AN/965, ICAO, 2012.
[22] On the approval of the Exemplary List of events in the field of civil aviation that are subject to notification, 29.03.2019, No. 396, Verkhovna Rada of Ukraine, https://zakon.rada.gov.ua/rada/show/v0396763-19.
[23] Provisional instructions on providing NTIB with reports on aviation occurrences and serious incidents involving civil aircraft, 06.03.2020, No. 19, https://nbaai.gov.ua/wp-content/uploads/2020/09/notification_instruction.pdf.
[24] NTIB aviation occurrence reporting, https://nbaai.gov.ua/report.
[25] S. Whibley, M. Day, P. May, M. Pennock, Wav format preservation assessment, 2016, British Library: London, UK.
[26] Firebird 3 Quick Start Guide, https://www.firebirdsql.org/file/documentation/html/en/firebirddocs/qsg3/firebird-3-quickstartguide.html.
[27] Delphi basics, https://www.delphibasics.co.uk/RTL.php.
[28] Indy Project, https://www.indyproject.org.
[29] THtmplParser, https://github.com/JackTrapper/delphi-html-parser.
[30] S. Katam, The porter stemmer, 2014, Indiana State University, http://cs.indstate.edu/~skatam/paper.pdf.