The conception of an intelligent system for troubleshooting an aircraft

Aleksandr Yu. Yurin [1,2], Yuri V. Kotlov [2] and Vladimir M. Popov [2]

[1] *Matrosov Institute for System Dynamics and Control Theory, Siberian Branch of Russian Academy of Sciences (ISDCT SB RAS), Lermontov St. 134, Irkutsk, Russia*
[2] *Moscow State Technical University of Civil Aviation, Irkutsk Branch (MSTUCA), 3, Kommunarov str., Irkutsk, 664003, Russia*

Abstract

The paper presents the conception of an intelligent system for troubleshooting an aircraft. The intelligent system requirements, its structure (architecture), and the domain conceptual model are described. In particular, the case-based and rule-based expert systems are defined as the main subsystems. The first one is designed to store information about malfunctions that are not accounted for in the current version of the documentation and find a solution by demonstrating similar problem situations. The second one provides the formation of plans to eliminate failures and malfunctions based on information from troubleshooting manuals. The features of their implementation are considered.

Keywords

Intelligence system, conception, troubleshooting, aircraft, RRJ-95

1. Introduction

One of the factors affecting the time and quality of maintaining and troubleshooting an aircraft is the rapid and operational use of relevant documentation that, in turn, has a large volume and complex structure, and contains texts, diagrams, drawings, and diagrams. For example, about 8 tons of paper documentation is delivered with the A-320 aircraft. The maintenance manual can be a document of up to 50,000 pages. It is quite difficult to use and maintain up-to-dating such a mass of documents.

To solve the task of the rapid and operational use of the documentation the electronic document management systems have been created, for instance, the AirNav Maintenance [1] designed for AirBus family aircraft. At the same time, the existing software is an interactive electronic technical manual without the possibility of expanding (or training) this system by entering new information and accumulating (storing) information about the identified new malfunctions and methods of their troubleshooting.

In this paper, we propose to create an intelligent decision support system for troubleshooting an aircraft, called the AirTech Assistant, designed for use by technical personnel engaged in the maintenance and repair of the Sukhoi Superjet (RRJ-95) aircraft.

The developed software will contain the necessary amount of documentation in electronic form and provide support for decision-making when forming work plans. Case-based [2] and rule-based [3] reasoning were selected as the main artificial intelligence methods for the implementation of the main system's functions. The first one is intended to store information about malfunctions that are not accounted for in the current version of the documentation and find a solution by demonstrating similar malfunctions with decisions. The second one is intended to form a troubleshooting plan based on formalized manuals.

The AirTech Assistant will reduce the load on technical personnel during troubleshooting, as well

ITAMS 2021 – Information Technologies: Algorithms, Models, Systems, September 14, 2021, Irkutsk, Russia
EMAIL: iskander@icc.ru (A. 1); yukotlov@rambler.ru (A. 2); povlamix@mail.ru (A. 3)
ORCID: 0000-0001-9089-5730 (A. 1); 0000-0002-7911-4857 (A. 2); 0000-0002-5899-3012 (A. 3)

as provide the collection (accumulation) and processing of information about equipment failures and malfunctions. In this paper, the conception of this software is considered in terms of the main functional requirements, the structure (architecture) and domain conceptual model, as well as the features of the implementation of the main subsystems.

The paper is organized as follows. Section 2 presents a state of art. Section 3 contains the description of the conception, including the requirements for the system functions, the structure of software, and the domain model, while Section 4 presents some concluding remarks.

2. State of art

The task of finding and troubleshooting failures and malfunctions of an aircraft is solved based on troubleshooting manuals, often presented either in printed form or in the form of electronic documents. Automation of its using and processing does not lose relevance, while various solutions are offered.

In particular, [4] considers a system for troubleshooting an aircraft, including a mathematical model of equipment and ensuring the interaction of this model with entries in the flight log (electronic mobile application) and electronic documents. Using this model, a correspondence is established between the equipment failure and its causes. At the same time, this paper does not specify the type of aircraft and the model used but offers a fundamental solution at the conceptual level. [5] describes a compromise solution aimed at upgrading/expanding the onboard maintenance system (OBMS) [6] by adding technical documentation to its memory with the ability to quickly access it. An algorithm is proposed for switching from a specific failure to a link to open the desired page of the troubleshooting manual. In [7], the formalization of the troubleshooting task and the method for synthesizing the optimal troubleshooting strategy for general aircraft equipment are proposed.

The considered and other solutions have some disadvantages: the lack of implementation details that ensure the reproduction of the proposals, including the models used; in some cases, general solutions are proposed without specifying the type of aircraft, while their adaptation is laborious.

The most famous foreign example of such systems is the AirNav Maintenance [1] developed by the Airbus Company. This software is designed for the maintenance of aircraft equipment, search, and troubleshooting. The AirNav Maintenance includes the following functional blocks:

- AMM (Aircraft Maintenance Manual;
- TSM (Troubleshooting Manual);
- IPC (Illustrated Parts Catalog);
- ASM (Aircraft Schematic Manual);
- AWM (Aircraft Wiring Manual);
- AWL (Aircraft Wiring List);
- ESPM (Electrical Standard Practices Manual).

From our point of view, the following subsystems are the most interesting:

- AMM: it contains information in the form of task cards necessary for maintenance, repair, replacement, adjustment, adjustment, inspection, and control of equipment and systems on the aircraft. These works (operations) are usually performed on the ramp or in the maintenance hangar. The information necessary for the maintenance of aircraft equipment is given by the supplier or manufacturer of the component. The AMM also contains information about the inspection and maintenance of aircraft structures.
- TSM: is a specific module integrated into the A319/A320/A321, A330, A340, and A380 navigation subsystem. It is designed for finding and eliminating failures.

Further, these works will be considered as a basis for the development of the AirTech Assistant conception, designing, and software implementation.

3. The conception of the AirTech Assistant

The conception of our software includes a description of its main elements at the conceptual level, including purpose, requirements, structure, and basic concepts and relationships of the domain that will be used when creating data and knowledge bases.

3.1. General information

The AirTech Assistant (Figure 1) is designed to support decision-making by technical personnel during maintenance and repair of aviation equipment, in particular when searching and troubleshooting.

The purpose of this stage of the project is to create the first experimental version of the intelligent system.

The object of automation is the activity of personnel for the maintenance and repair of aircraft. The subject of automation is an algorithm for decision support in searching and troubleshooting the aircraft power supply system.

Initial data:

1. OBMS information about malfunctions;
2. Information about new failures and malfunctions that are not accounted for in the current version of the documentation, and their statistical indicators;
3. Information about malfunctions from the troubleshooting manuals.

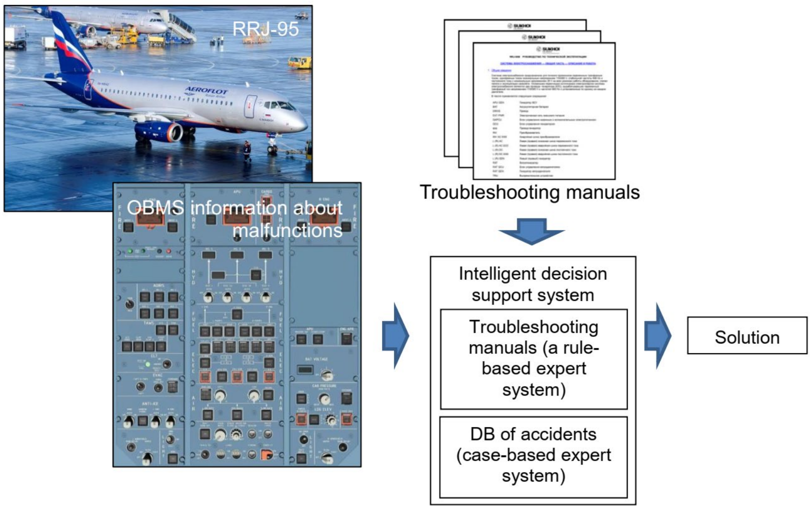

Figure 1: The schema of the system with an indication of input data.

3.2. Main functions

The AirTech Assistant should provide the following main functions:

1. Managing (entering, editing, storing) information about aircraft systems, technical operation, malfunctions (failures), and troubleshooting based on the aircraft documentation package;
2. Search for information about failures and malfunctions based on OBMS information to form a list of possible failed systems;
3. Accumulation (entering, editing, storage) of information about new failures and malfunctions that are not accounted for by the current version of the documentation (supporting of the case base);
4. Formation of a work plan for the search, confirmation, and troubleshooting failures and malfunctions based on the hybrid information contained in the current version of the documentation and the case base;
5. Maintenance of the repair and maintenance process based on a domain-specific interface.

3.3. Main subsystems

To implement the AirTech Assistant functions, the architecture, which contains the following main modules, is proposed:

1. Storage of documentation (manuals) in electronic form (in the form of PDF files or HTML pages containing information from operating manuals, maintenance manuals, troubleshooting) and information about an aircraft;
2. A case-based expert system for accumulating, searching, and storing information about failures and malfunctions, including:
    a. a database (or a case-based knowledge base) with information about failures and malfunctions;
    b. a search subsystem (solver) for cases retrieval;
    c. a subsystem for the acquisition (accumulation) of knowledge, which provides an extension of the case base.
3. A rule-based expert system for the defining systems who are the candidate for failure and the formation of plans of work on search, verify and troubleshoot failures, including:
    a. a knowledge base with information about the failures and task cards are required to maintain, repair, replacement, setup, adjustment, inspection, and control of equipment and systems on the aircraft;
    b. a search engine (solver) for logical inference;
    c. a query subsystem;
    d. a knowledge acquisition subsystem that provides the expansion of the rule base through visual programming and formalisms of event trees, decision tables, and state transition diagrams describing work plans for finding, confirming, and troubleshooting failures and malfunctions.
4. The subsystem for generating reports.
5. The domain-specific user interface.

3.3.1. A case-based expert system

When implementing the case-based expert system, our experience of creating such systems for solving diagnostic problems in the petrochemical industry [8, 9] and predicting emergencies [10] was used. The main issues when creating systems of this class are the following: a definition of a case model; choosing metrics used for the search and case retrieval; defining a method for adapting the solutions obtained.

Let's look at these issues in more detail.

A case model. A case is a structured representation of the accumulated experience in the form of data and knowledge, providing its subsequent automated processing with the aid of specialized software [8]. The case allows one to structure units of experience, while the choice of a case model (structure) depends on the tasks solved. The general structure (model) of cases includes two main parts:

- identifying (characterizing) part that describes the experience in a way that allows one to assess the possibility of its reuse in a certain task;
- learning part that describes the solution (decision) of the task or part of it.

From the formal point of view most decision-making tasks can be described by a set of characteristics (properties), and can be formalized as follows [11]:

$$M^{Task} = \{p_1, \ldots, p_M\},\ p_i \in Prop,\ Prop = \cup_i p_i,\ i=[1,N] \qquad (1)$$

where M^{Task} is a task model; p_i are task properties (significant characteristics), $Prop$ is a set of properties.

According to (1) the task model is defined as follows:

$$M^{Task\_CBR}: Problem^{CBR} \rightarrow Decision^{CBR},$$

where M^{Task_CBR} is a task model in terms of case-based reasoning; $Problem^{CBR}$ is a task (problem) description, $Decision^{CBR}$ is a decision of a problem, while:

$$Problem^{CBR} = \langle c^*, C \rangle,\ C = \{c_1 \ldots c_K\},\ c^* \notin C,$$

where c^* is a new case, C is a case base.

$$Decision^{CBR}=\{d_1 ,..., d_R\},\ d_i=(c_i,s_i), c_i \in C,\ s \in [0;1]$$

where $Decision^{CBR}$ is a task decision in the form of a set of retrieved cases with similarities s_i.

Let's formalize a case description:

$$c_i = \{Prop_i^{Problem},\ Prop_i^{Decision}\},$$

where $Prop_i^{Problem}$ is an identifying (characterizing) part of a case, $Prop_i^{Decision}$ is a learning part of a case. In addition, each of these parts contains task properties and the composition of the parts has a problem-specific character:

$Prop_i^{Problem} = \{p_1,\ldots, p_m\}$, $Prop_i^{Decision} = \{p_{m+1},\ldots, p_N\}$, $Prop_i^{Problem} \cup Prop_i^{Decision} = Prop$, $Prop_i^{Problem} \cap Prop_i^{Decisio} = \varnothing$.

Thus, when defining the task properties filling a case model, it is necessary to clarify the concepts of the domain that make up these components. For this purpose, we designed the conceptual model (Figure 2).

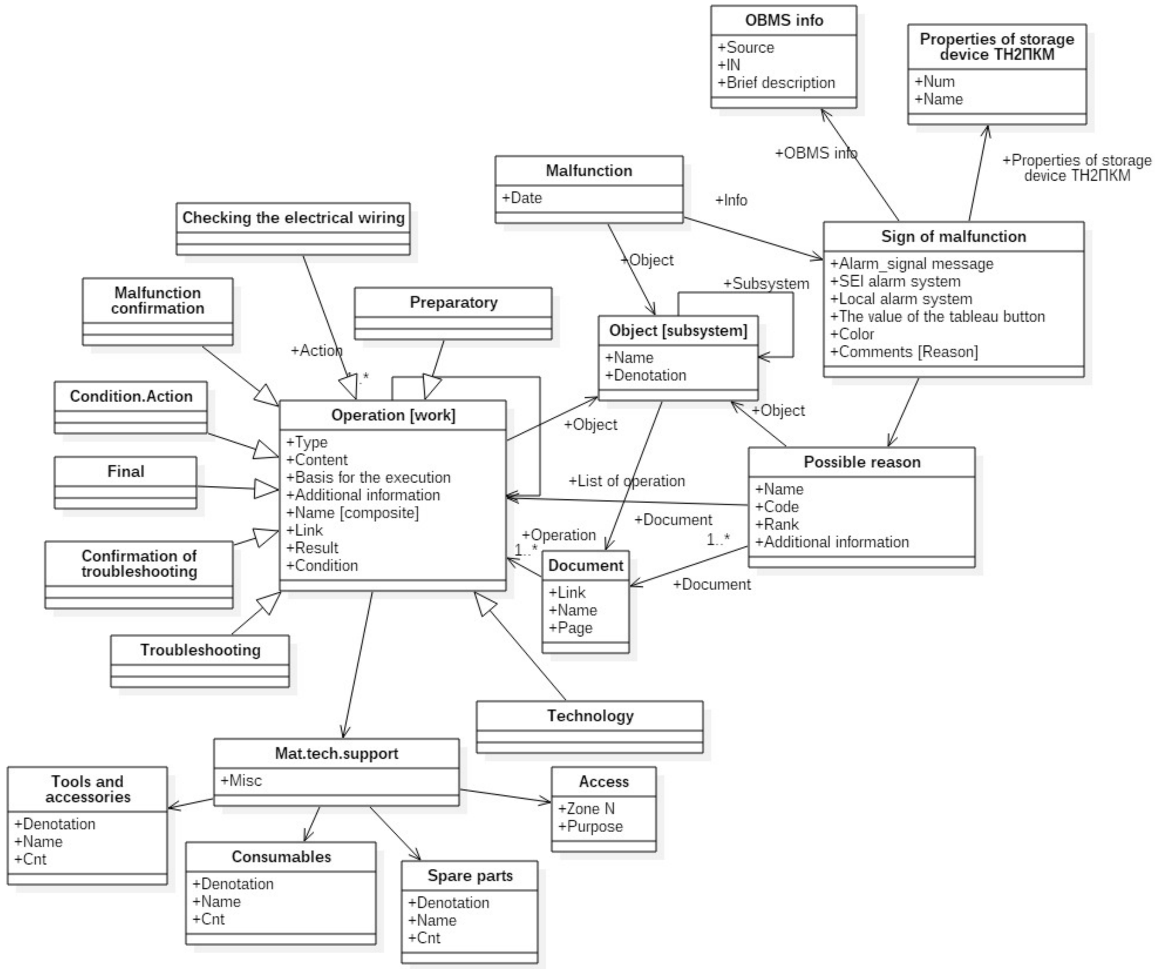

Figure 2: A fragment of the domain conceptual model.

Case retrieval. Many methods can be used for case retrieval [12]: nearest neighbor, decision trees, etc. The most popular method is the nearest neighbor, based on an assessment of similarity with the aid of different metrics, for example, Euclidean, City-Block-Metric, etc. In our case we will use the Zhuravlev metric [13] with the normalization:

$$d_G(\bar{x},\bar{y}) = \sum_{i=1}^{N} w_i h_G(x_i, y_i) \Big/ N,$$

$$h_G(x_i, y_i) = \begin{cases} \text{for quantitative} \begin{cases} 1, if |x_i - y_i| < \xi \\ 0, otherwise \end{cases}, \\ \text{for qualitative} \begin{cases} 1, x_i = y_i \\ 0, x_i \neq y_i \end{cases} \end{cases}$$

where w_i is the information weight, and ξ is the constraint on the difference between the values of properties. At the same time, the normalization (or standardization) is as follows:

$$x_{ik} \to \left(x_{ik} - \min_k x_{ik} \right) \Big/ \left(\max_k x_{ik} - \min_k x_{ik} \right).$$

Case reuse. This issue is the most difficult from the point of view of computer-aided processing, since it requires the involvement of experts for a meaningful interpretation of the solutions obtained [14, 15]. In most cases, the result of a meaningful interpretation is the adaptation, i.e. some transformation with taking into account the current task features.

In our case, we propose to use the so-called "zero adaptation", based on copying part of the solutions.

3.3.2. A rule-based expert system

Our experience of creating rule-based expert systems [16] allows us to select techniques for the formalization, conceptualization, and codification of logical rules. The main issues when creating systems of this class are the following: the structure of rules, technologies for implementing logical inference, and techniques for input/acquisition of rules.

The structure of rules will be defined based on the domain model (Figure 2), and the prototypes presented in [17].

To support the input/acquisition of rules we propose to use the following techniques:

- Describing the rules in the form of decision tables of a specialized type [18], which provides: description name of the rules; a clear definition of the concluding part of a rule by using the "#" symbol; the ability to specify not only the properties but also its belonging to a particular class by using string separator "::" in the column name.
- Describing the rules in the form of event trees [17], which allows one to use visual programming methods;
- Describing the rules in the form of state transitions diagrams, which provide, unlike event trees, to create cyclic graphs.

4. Conclusion

The paper considers the conception of an intelligent system for troubleshooting an aircraft, namely the AirTech Assistant. The requirements for its functionality, the structure, and the domain conceptual model are described. Case-based and rule-based expert systems are defined as the AirTech Assistant main subsystems. The features of the implementation are considered.

The AirTech Assistant will be a client-server web application and the following technologies and means will be used for its realization: the Yii Framework, PHP, MySQL. The prototypes of knowledge bases will be designed with the aid of the Personal Knowledge Base Designer [19].

5. Acknowledgements

The present study was supported by the Ministry of Education and Science of the Russian Federation (Project no. 121030500071-2 "Methods and technologies of a cloud-based service-oriented platform for collecting, storing and processing large volumes of multi-format interdisciplinary data and knowledge based upon the use of artificial intelligence, model-driven approach and machine learning"). Results are achieved using the Centre of collective usage «Integrated information network of Irkutsk scientific educational complex».

6. References

[1] AirNav-Maintenance, 2021. URL:https://www.airnav.com/maintenance/index.html.

[2] A. Aamodt, E. Plaza, Case-based reasoning: foundational issues, methodological variations, and system approaches, AI Communications 7(1) (1994) 39-59. doi:10.3233/AIC-1994-7104.

[3] P. Jackson, Introduction to Expert Systems, 3rd Edition, Addison-Wesley, 1998.

[4] O.V. Perfiliev, S. Ryzhakov, and V.A. Dolzhikov, Intelligent system for finding malfunctions an aircraft, Proceedings of the Samara Scientific Center of the Russian Academy of Sciences 4(3) (2018) 326-331. (in Russian)

[5] A.M. Savvina, Proposal for the modernization of the onboard maintenance system of the SSJ 100 aircraft, Crede Experto: transport, society, education, language 3(22) (2019) 27-35. (in Russian)

[6] AUC EATK: On-board aircraft maintenance system for RRJ-95B, Moscow, 2013. (in Russian)

[7] N.N. Makarov, Synthesis of an algorithm for the functioning of an information and control system for monitoring and diagnosing the state of general aircraft equipment, News of universities: Aviation Technology 1 (2008) 46-50. (in Russian)

[8] O.A. Nikolaychuk, A.Yu. Yurin, Automating the identification of mechanical systems' technical state using case-based reasoning, IEEE Intelligent Systems. Processing of the 3rd International IEEE Conference Intelligent Systems (2006) 30-35. doi:10.1109/IS.2006.348389.

[9] A.F. Berman, O.A. Nikolaychuk, A.Yu. Yurin, A.I. Pavlov, A methodology for the investigation of the reliability and safety of unique technical systems, Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 228 (2014) 29-38. doi:10.1177/1748006X13494820.

[10] A.F. Berman, O.A. Nikolaychuk, A.Yu. Yurin, A.I. Pavlov, Decision support system for the prevention and elimination of technogenic emergencies based on a case-based approach, Technologies of technosphere safety 5 (51) (2013) 13.

[11] G.S. Maltugueva, A.Yu. Yurin, Improving case-based reasoning with the aid of multi-criteria and group decision-making methods, Proceedings of the 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (2019) 1031-1036. doi:10.23919/MIPRO.2019.8756874.

[12] L.R. De Mantaras, D. Mcsherry, D.Bridge, D.Leake, B.Smyth, S.Craw, B.Faltings, M.L.Maher, M.T. Cox, K. Forbus, M.Keane, A. Aamodt, I.Watson, Retrieval, reuse, revision and retention in case-based reasoning, Knowledge Engineering Review 20(3) (2005) 215–240. doi:10.1017/S0269888906000646.

[13] I.Yu. Zhuravlev, I.B.Gurevitch, Pattern recognition and image recognition, Pattern recognition, classification, forecasting: Mathematical techniques and their application 2 (1989) 5-72. (in Russia)

[14] D. McSherry, An adaptation heuristic for case-based estimation, Lecture Notes in Computer Science 1488 (1998) 184–195. doi:10.1007/BFb0056332.

[15] A.Y. Yurin, Group decision-making methods for adapting solutions derived from case-based reasoning, Scientific and Technical Information Processing 42(5) (2015) 375-381. doi:10.3103/S014768821505010X.

[16] A.F. Berman, O.A. Nikolaichuk, A.Yu.Yurin, K.A. Kuznetsov, Support of Decision-Making Based on a Production Approach in the Performance of an Industrial Safety Review, Chemical and Petroleum Engineering 50(1-2) (2015) 730-738. doi:10.1007/s10556-015-9970-x.

[17] N.O. Dorodnykh, Y.V. Kotov, O.A. Nikolaychuk, V.M. Popov, A.Yu.Yurin, End-user development of knowledge bases for semi-automated formation of task cards, CEUR Workshop Proceedings 2913 (2021) 60-73. doi:10.47350/ICCS-DE.2021.05.

[18] A.Yu. Yurin, N.O. Dorodnykh, Creating Web Decision-Making Modules on the Basis of Decision Tables Transformations, Communications in Computer and Information Science 1341 (2021) 167–184. doi:10.1007/978-3-030-68527-0_11.

[19] A.Yu. Yurin, N.O. Dorodnykh, Personal knowledge base designer: Software for expert systems prototyping, SoftwareX 11 (2020) 100411. doi:10.1016/j.softx.2020.100411.