PAPER • OPEN ACCESS

On the problem of the manned aircraft modification to UAVs

To cite this article: I V Kovalev *et al* 2019 *J. Phys.: Conf. Ser.* **1399** 055100

On the problem of the manned aircraft modification to UAVs

I V Kovalev[1, 2, 3, 4], A A Voroshilova[2] and M V Karaseva[1, 3]

[1]Reshetnev University, 31, Krasnoyarsky Rabochy Av., Krasnoyarsk, Russia, 660037
[2]Krasnoyarsk City Hall of Science and Technology, 61, Uritzkogo, Krasnoyarsk, Russia, 660049
[3]Siberian Federal University, 79, Svobodny Pr., Krasnoyarsk, Russia, 660041
[4]China Aviation Industry General Aircraft Zhejiang Institute Co., Ltd

E-mail: kovalev.fsu@mail.ru

Abstract. The paper considers the experience of the manned aircraft modification to UAVs and technological solutions for the manned transport aircraft modification to UAVs. A sequence of stages in the implementation of the task of the manned aircraft modification to UAVs based on the presented review is formulated. The paper presents that the developed or converted UAVs have different software and hardware platforms; therefore, the importance of their cross-platform organization is noted. The basic principles, the typical architecture of the UAVs control platform is based on, including on-board software, are formulated. This creates the background for the development of the on-board software technology concerning the automation of the architectural design of the subsystem software and the automation of the system testing design of the software based on the formalized specifications.

1. Introduction

Nowadays, there is no any single concept for the manned aircraft modification to UAVs. However, some companies-developers of both civil and military aircrafts have the experience in this field, as well as technological solutions for the manned transport aircraft modification to UAVs. Both supporters and opponents of this direction note the necessity for novelty and manufacturability of the proposed approach. Also, they noted the necessity to work out the sequence for implementing a specific project that may include both research on a manned aircraft and the developing mathematical models and algorithms for UAVs control systems.

The classification problems of autonomous unmanned objects were considered by the authors in [1]. It is noted that a class of UAVs include both classic winged drones and rocketry, as well as multi-rotor systems (quad-, hexa-, octo-copters) that are gaining popularity. In the case of air flights, weight and size are very important, since raising a usual human into the sky is already a substantial problem requiring a completely different level of hardware compared to, for example, raising equipment for video and photography. The absence of a pilot can fundamentally cut weight and reduce a size of the aircraft. Accordingly, production and flight costs are reduced. In most cases there is no need for fuel engines; it is reasonably to have only electric motors running on batteries.

Small sizes and lack of fuel engines, which are sources of heat, significantly complicate the detection of such aircrafts. The main disadvantage of unmanned aerial vehicles is the vulnerability of the radio communication channel for remote control. The autonomous mode solves this problem, since even a complete loss of the channel does not disable a device; it continues to perform the task. The lack of a radio channel is also a significant advantage in terms of the difficulty of detecting an autonomous unmanned object (ABO). At the moment, autonomous unmanned objects are used both in the military industry (surveillance aircrafts, military equipment), and in everyday use. It is already practicing the delivery of consumer goods using autonomous unmanned multirotor systems.

2. Manned aircraft modification to UAVs

In the paper [2], considering the modification of manned military aircrafts to UAVs, the author notes the successful implementation of the unmanned version of the Boeing military fighter F-16 Fighting Falcon. The first flight of the QF-16 UAV as an unmanned version of the F-16 Fighting Falcon was in September 2013 and lasted one hour. According to AirForceTimes [3], the fighter took off from the Tyndal air base in Florida; flight tests were conducted over the Gulf of Mexico. During the test flight, the QF-16 UAV was tested for the possibility of supersonic flight, as well as the ability to perform evasive actions.

The tests were conducted jointly by the American aircraft manufacturer Boeing and pilots of the 82nd squadron of air targets. Before the takeoff, the U.S. Air Force pilot from the QF-16 UAV cockpit pre-took off and then left the aircraft. The further control was carried out by a ground station. According to experts, the flight QF-16UAV was in full accordance with expectations. However, in this case, the QF-16UAV should be considered as an optionally manned aircraft, that can be controlled both from the cockpit and the ground station. The first manned QF-16 UAV test took place in May 2012. The piloted flight lasted 66 minutes.

It becomes clear that the manned aircraft modification to UAVs is not only the experiment taking into account the plans concerning the repair and convert to over 100 decommissioned F-16 fighter jets of Block 15, 25, and 30 versions that were in long-term storage at Davis-Monten Air Base in Arizona. The top engineer of the project Paul Cejas noted that the QF-16 UAV is ideal for operations on the enemy territory and dangerous missions. At the same time pilots are not in danger. The wide aerobatic capabilities that will not be limited by the pilot's physical endurance threshold, as well as the absence of a complex and heavy human life support system on the board of UAVs, will make alteration of existing and creating new unmanned fighter drones. It is the main direction in the development of military aircraft engineering. In particular, during the first test flight, the QF-16 UAV was tested at 7G overloads, although it is able to perform actions with 9G overloads in the unmanned version.

Also, one of the first cases of the aircraft modification to UAVs is the Aeronautics Defense Dominator Medium-Altitude Long-Endurance (MALE) UAVs (Israel-Austria) project based on the well-known Diamond DA42 TwinStar (Austria) [4]. UAV Dominator is used by the Israeli military. In addition, three UAVs were exported to Mexico and Turkey.

The Dominator [4] is powered by two Centurion 2.0 diesel engines, each of which produces 135hp (100kW) of output power. The Centurion 2.0 is a four-stroke diesel engine certified by the Federal Aviation Administration (FAA) in October 2006. It is fitted with wet sump oil systems, a three bladed MT-propeller with hydraulic pitch controller and reduction gear box. Thielert Aircraft Engines is the prime contractor involved in the production of the engines. The Dominator can fly at a maximum speed of 354km/h. The range and service ceiling are 300km and 9,100m respectively. The UAVs can loiter in air for a maximum of 28 hours. A full authority digital engine control (FADEC) incorporated in the engine facilitates automated missions. Both kerosene and diesel in any mixture ratio can be used as the fuel. The width and depth of the engine are 0.79m and 0.8m respectively. The height is 0.63 m. The dry weight is 134 kg.

The real-time data captured by the Dominator is retrieved, processed, stored and monitored at the ground control station (GCS). Communication between the GCS and Dominator takes place through a line of sight data link up to 300km range. The GCS is designed and manufactured by Aeronautics Defence Systems.

The conceptual project [5, 6] of the Ukrainian light aircraft V-24 Softex Aero modification to the unmanned V-24 UAS (both with the possibility of installing a satellite control channel and without it) confirms the prospects in this field of engineering. The concept of the V-24 Softex Aero manned aircraft modification to UAVs was developed as a part of an international tender. The fundamental aspects were the novelty and adaptability of approaches, cost savings and the achievement in the wide range of targets for the UAVs version (a strike drone, police UAVs, monitoring and multipurpose aircrafts).

Despite the fact that some UAVs developers believe that initially designed UAVs for specific tasks will have a greater prospect. It should be noted that the re-equipment of manned aircrafts and helicopters into large drones will allow using the already developed service base and a large number of spare parts. The important point is that in this case, significant cost savings are achieved for the engineering and production process.

This is confirmed by the new American startup Dorsal Aircraft. This way of the manned transport aircraft converting to UAVs is presented in [7]. The American startup Dorsal Aircraft presented the project of converting the C-130H Hercules manned transport aircraft to an unmanned carrier of standard cargo containers.

In the case of implementation, a project involving the complete removal of the original fuselage of the aircraft will help to upgrade obsolete and decommissioned aircrafts. The conversion technology can be adapted to other types of transport aircraft.

Modern transport aircrafts have potentially large carrying capacity, "selected" by additional on-board equipment, control systems, providing comfort necessary for a crew. Moreover, a part of the load capacity is selected by various systems that facilitate loading and unloading, including cranes, rollers on the floor and mounting systems for cargo pallets. Removing all these systems will increase the usable space inside the fuselage and the mass of cargo that the aircraft can take on board.

The project, presented by Dorsal Aircraft, involves the preservation of the fuel system, wing consoles, center wing, tail and landing gear of the C-130H transport aircraft (this version of the transporter was produced from 1974 to 1996). It is supposed to install a new control system instead of the old one. The original fuselage of the aircraft is sent for recycling. Instead, the company intends to assemble a “box” fuselage with aerodynamic panels.

A new fuselage is planned to be assembled from the upper strong beam section and side supporting structures on which the tail swing doors and lower cargo hatches should be mounted. In the upper beam section, several electric motors with cables should be installed. It is assumed that in loading hatches and doors will be opened and three standard sea cargo containers will be brought into the interior of the fuselage at once. Then electric motors will lift up the containers with the help of cables, and all hatches and doors will be closed.

It will be possible to attach all stored original elements of the aircraft to the new fuselage. The resulting transport unmanned aerial vehicle can be used to transport cargo on domestic flights. It will be necessary to modify one of the jet cargo aircraft to an unmanned aerial vehicle for transatlantic and trans-pacific flights. Presumably, the use of such modified vehicles will increase the volume of air cargo.

The analysis of projects [2-7] helps to formulate a basic sequence of stages for the task implementation of the manned aircraft modification to UAVs. These steps are as follows:

- development of the preliminary mathematical model using special software packages (for example, Matlab [8], Simulink [9], etc.);
- adaptation of the control algorithm and the necessary research conducting on a manned aircraft;
- testing on the basis of a UAVs customer in a manned version with testing the FLY-BY-WIRE or FLY-BY-OPTICS control system (conducting a flight program with a pilot, including, for example, reading parameters in autopilot mode, clarifying antenna positioning on aircraft, etc.);
- specification of the mathematical model;
- development of a set of design documentation and the prototype transfer;
- technology transfer for the independent production and work on the UAVs control system.

3. Platform architecture by UAVs control

The given brief overview shows that the developed or modified UAVs have different software and hardware platforms. Often (in the case of the manned aircraft modeling to UAVs), these platforms have been in operation for decades and require modernization or installation of new ones. It is obvious that the analysis of the control algorithm (the logic of the onboard control system operation) of the UAVs at all the stages of the task implementation determines the typical architecture and software interfaces for the UAVs onboard software (OBS). In this case, it is important to ensure the possibility of the effective technology creation for this BPO of the UAVs development and maintenance.

Formulate the basic principles, the typical architecture of the UAVs control platform is based on [10, 11].

- The OBS of UAVs is developed and considered as a set of the UAVs subsystem software. They are, in turn, created and considered as embedded software for these subsystems. Thus, such a super-subsystem as on-board software of the unmanned aerial vehicles UAVs does not appear on it (that is exactly what happens in today's UAVs). In this case, the process of creating (modification) and operating the aircraft is greatly simplified.
- All non-specific functions for organizing the computational process and controlling the equipment of subsystems are implemented in the software of the onboard control system (BCS) as the software operation environment and the program control environment, respectively. The software for each UAVs subsystem solves problems specific to this subsystem, using applied program interfaces and tools provided only by these environments.
- The implementation of the integrated modes of UAVs operation is carried out using integrated control tools included in the DCS software.

These principles help to make the OBS of UAVs more mobile, more intelligent and simplify the process of its development and maintenance significantly. The advantages of this approach in various aspects of its impact on the development and maintenance technology of the OBS of UAVs were considered earlier in [12–15].

Note that this approach has no analogues both in Russian practice and in the practice of the European Space Agency [16-18] and, often, confuses the UAVs customer at the initial stages of the contract, but it always highly appreciates at the next stages and in the UAVs operation process, since it simplifies significantly the interaction between customers and contractors regarding the development of the OBS of UAVs.

3.1 Environment of software operating

The environment of software operating provides tools for organizing the computing process and ensures the constancy of the first level software interface. It allows to develop and maintain on a priority basis in multiprogramming mode and in real time the flow of synchronous and asynchronous tasks needed to control a specific UAVs. It also helps to control and document the computing process and restore it in abnormal situations. For example, MODULA-2 provides an interface and RUN-TIME support for target control programs implemented in the MODULA-2 language [19]. In its modern form, the programmatic operating environment contains the main set of features that are typical for the embedded real-time operating system.

3.2 Software control environment

The program control environment supports real-time instruction and control tools and provides a consistent second-level programming interface. The environment is usually implemented in the form of a set of libraries and is intended for its application in the implementation of UAVs subsystem control programs, i.e., object subsystem software. Such an environment should contain an interpreter of control instructions, tools of program-time and event control, means of issuing control actions and access to telemetric information.

The software control environment provides an application programming interface independent of the computing platform. The object subsystems can be implemented in the MODULA-2 programming language (or other MODULA-like languages) using the tools provided by these environments. The

Journal of Physics: Conference Series **1399** (2019) 055100 doi:10.1088/1742-6596/1399/5/055100

software of the object's subsystems with the constancy of the interfaces of these tools becomes independent from the applied computing platform. Thus, in most cases, in changing the computing platform, only the environment of software functioning is affected.

Programming the control instructions interpreter for the specific UAV can be carried out by subsystem developers if a problem-oriented programming language is provided for this, i.e. modules development for object control instructions.

3.3. Integrated control tools

The integrated control tools provide opportunities for organizing duty monitoring and diagnostics of UAVs subsystems and autonomous integrated control of these subsystems at the autonomous UAVs control level as a whole.

The integrated control tools allow implementing the logic of UAVs operating as a whole directly to its designers, using a problem-oriented programming language as it was mentioned above. These tools ensure the consistency of the third level software interface.

The service layer is present in the OBS in the case of modification a manned aircraft to a UAV, instead of the initial environment for programmed operating, a new (independently developed or commercial) real-time operating system (OS) is applied. The task of this layer, taking advantage of the capabilities of a commercial OS, is to maintain the constancy of the first level software interface to ensure the portability of the remaining components of the OBS of UAVs.

It is necessary to develop a set of canonical program interfaces of the above three performing the analysis of the operating a large number of UAVs of one or several classes. In this case, the onboard software is developed as the embedded software for the UAVs subsystems, combined by the onboard control system software and integrated control macro programs, and confirmed within the UAVs as a whole and the UAVs subsystems.

The binding of parts of the onboard software to the UAVs subsystems helps to define clearly the requirements for the software of the subsystems and the OBS in general, as well as the responsibility for the architectural development, system testing and confirmation of the subsystems software and the OBS in general.

Thus, in creating a BPO for a specific UAVs, the principles considered for the architectural development of the OBS enable to implement the effective models for the project's control, configuration and verification of the OBS, thereby increasing the efficiency of solving current problems and reducing time and cost of the project as a whole. The development and maintenance quality of the OBS is assured.

These principles not only provide the opportunity to reuse a significant volume of onboard software code, but also contribute to the accumulation of knowledge and solutions. Also, they have the opportunity in the near future to go over to the automated development of the OBS based on libraries of the unified architectural projects and components.

4. Distribution of responsibilities and obligations among developers

Formulate different versions for the distribution of responsibility and obligations in the development and maintenance of onboard software for the UAVs on the basis of the proposed principles and architectural solutions.

4.1 Basic version

So, in accordance with the principles of architectural development of the considered onboard software, the development and maintenance of software subsystems is assigned to the structural units (departments) of these systems design. In each of these departments, a sector or programming group is organized for the software subsystem programming. Thus, the designers of the subsystems and the programmers of the subsystems are under the responsibility of one general leader and in one target problem area. This situation eliminates the need to shift responsibility from designers to programmers and vice versa in solving problems arising during the development and maintenance of both software subsystems and the subsystems themselves. Their interaction becomes less formal, making it possible to apply not only a cascading approach to the development of the onboard software, but also iterative, evolutionary and even elements of "rapid development".

In this version, the subsystem designers carry out architectural development and system testing of the subsystem software. Programmers perform the detailed design, programming and autonomous testing of the software components of the subsystem.

An exception is software development of the onboard control system. This software is developed by a system programming department for a specific UAVs class, and another department responsible for creation the logic of UAVs operating as a whole develops BCS.

This is due to the fact that these departments have quite different target problem areas, and their intersection is realized not by developing programs in a programming language, but at the level of UAVs control instructions and macro programs developed in problem-oriented languages by BCS designers.

The integrated OBS problems are solved in the BCS software, and the system programming department additionally acts as a head department and determines the technology for developing the OBS, it provides unified development tools and controls the configuration and manufacture of the OBS.

Thus, from the functional point of view on UAVs there is no any OBS subsystem, but a set of subsystems of the particular product with the embedded software.

This approach, in contrast to the traditionally used in foreign and domestic projects, significantly simplifies the verification and confirmation of the onboard software and its maintenance. It leads to a significant reduction in labor costs and its duration.

4.2 Promising version

The analysis of the volumetric characteristics of the OBS of UAVs is able to identify a number of subsystems where the software volume does not exceed 1-10% of the entire UAVs volume. This means that the development of programs is carried out by one or two programmers. It is difficult to maintain a high level of technology in such a small team, and problems arise in maintaining a workable team to provide long-term support.

In this case, a more effective version for the responsibility and work distribution will be the tasks transfer for the detailed design of components, their programming and stand-alone testing to one of the departments with large teams of programmers that are closer in the target problem area. At the same time, the department designing the subsystem remains responsible for the software of the subsystem as a whole. The designers of this department should perform work on the architectural design of the subsystem, system testing, confirmation and maintenance while testing and operation.

The need for a more formalized interface between designers and programmers is leveled out by a small volume of the problem being solved. At the same time, the level of technology is significantly increased, the development time is reduced and the problem of long-term maintenance is solved.

The considered approach to the development of the onboard software and the work organization about its creation and maintenance leads to the fact that when they are used in the development and operation of UAVs, the concept of “OB subsystem” disappears, which, that is always present in domestic and European practice [20].

Appling this approach, the OBS is structurally a part of the BCS, and functionally considered as a part of the UAVs subsystems, while the integral issues are considered and implemented as a part of the BCS software, which, in accordance with the described approach, is a part of the OBS of UAVs.

Thus, both in developing contractual requirements for the UAV and its subsystems being created and in confirming their fulfillment, only subsystems and UAVs appear in general, which greatly simplifies the relationship between the Customer and the Contractor and reduces the amount of work.

A similar simplification occurs in operating the UAVs. All problems arising in the UAVs are considered in the responsibility areas of the UAVs subsystems developers as a part of the operational documentation for the subsystems and the UAVs as a whole.

5. Conclusion

The developed principles of the architectural design of the UAVs and typical program interfaces provide the possibility of creating an effective technology for the development and maintenance of UAVs. These principles help to create the OBS that can be transferred to various computing platforms, adapted for the use on new and modified UAVs (manned aircraft modification) and changed in operation.

This approach to the development of the OBS of UAVs has no analogues in the domestic and European practice of creating UAVs and significantly reduces the complexity of the work on its development.

The developed principles of the onboard software architectural construction and typical software interfaces create the background for the OBS development technology in the direction of automating the architectural design of software subsystems and automating the design and system testing of software subsystems and onboard software as a whole based on formalized specifications (requirements) for OBS and software subsystems.

Acknowledgments.
The authors are grateful to the China Aviation Industry General Aircraft Zhejiang Institute Co., Ltd for supporting this study and creating favorable conditions for research.

References

[1] Saramud M V, Kovalev I V, Losev V V, Poskonin M V and Kalinin A O 2017 On the classification of autonomous unmanned objects *Conference on Topical Issues in Aeronautics and Astronautics* **2** 64–6
[2] Naumenko P Unmanned conversion: modification of a manned aircraft in an UAV https://pavelaviator.livejournal.com/2047.html
[3] New target fighter made its first unmanned flight 2013 https://lenta.ru/news/2013/09/25/qf16/
[4] Dominator MALE UAV. Air Force Technology. https://www.airforce-technology.com/projects/dominator-uav/
[5] Serial production of the aircraft family for the Ministry of Defense of Ukraine: business aircraft (administrative) / multi-purpose aircraft / UAV, based on the V-24 Softex aircraft platform / P.O. Naumenko // *15-th scientific conference "Creation and modernization of armament and military equipment in modern conditions", 17-18 September 2015, Chernihiv: Research and testing complex "State Research and Testing Center of the Armed Forces of Ukraine"* 2015
[6] Krivtsov V S, Naumenko P O, Grebenikov G O *et al* 2007 Concept of UAV complex *Open information and computer integrated technologies: compilation of academic works of N.E. Zhukovsky National Aerospace University «KhAI»* **35** 5–15
[7] Grigory Kopiev N + 1 Americans will remake transporters into unmanned container carriers" https://yandex.ru/turbo?text=https%3A%2F%2Fnplus1.ru%2Fnews%2F2018%2F03%2F22%2Fcontainer&d=1
[8] Liu J 2006 MATLAB *Simulation for Advanced PID Control* (Bejing: Publishing House of Electronics Industry)
[9] Aerospace Blockset *MathWorks. Zhentrkompetentzii* URL: http://matlab.ru/products/ aerospace-blockset
[10] Koltashev A 2003 A Practical Approach To Software Portability Based On Strong Typing And Architectural Stratification *Lecture Notes in Computer Science* pp 98–101
[11] Koltashev A A 2004 Technology of transfer of on-board software *Open Systems* **4** 13-20
[12] Saramud M V, Kovalev I V, Losev V V and Petrosyan M O 2018 Application of FreeRTOS for Implementation of the Execution Environment of Real-time Multi-version Software *International Journal on Information Technologies & Security* **3 10** 75–82
[13] Saramud M V, Kovalev I V, Losev V V and Kuznetsov P A 2018 Software interfaces and decision block for the execution environment of multi-version software in real-time operating systems *International Journal on Information Technologies & Security* **1 (10)** 25–34

[14] Kovalev I V, Losev V V, Saramud M V, Kovalev D I, Petrosyan M O and Brezitskaya V V 2017 To the question on implementation of multi-version execution environment software of onboard autonomous unmanned objects by means of real-time operating system *Siberian Journal of Science and Technology* **18 (4)** 744–7
[15] Kovalev I V, Losev V V, Saramud M V, Kuznetsov P A and Petrosyan M O 2017 To the Question of the Organization of a Learning Environment for Developers of Cross-Platform OnBoard Software for Unmanned Aerial Vehicles. *TOJET: The Turkish Online Journal of Educational Technology – November 2017, Special Issue for INTE 2017* pp 700-5
[16] ECSS-E-ST-10C2009 System engineering general requirements. ECSS Secretariat ESA-ESTEC Requirements & Standards Division Noordwijk, The Netherlands p 100
[17] ECSS-E-10 Part 1B2004 Space product assurance. Software product assurance. ECSS Secretariat ESA-ESTEC Requirements & Standards Division Noordwijk, The Netherlands p 70
[18] ESA PSS-05-0 1991 ESA software engineering standards. Issue 2. *Prepared by: ESA Board for Software Standardisation and Control (BSSC)* p 154
[19] Hernandez J and Sanchez J A 1993 RT-MODULA2: an embedded in MODULA language for writing concurrent and real time programs *ACM SIGPLAN Notices* **27 2**
[20] Mishurov A V and Panko S P 2017 The use of international standards in the design of new tracking, telemetry and command subsystems for spacecrafts The Research of the Science City *Association «Technology Platform «National Information Satellite System»* **1(19)** pp 11–5