Check for updates

The Automation Tool Development for Aircraft Cockpit Display Systems Verification in Part of Text Data

Sergey Aleksandrovich Dyachenko(✉), Dmitry Mikhailovich Ilyashenko, and Evgeny Sergeevich Neretin

Moscow Aviation Institute (National Research University), Moscow, 4, Volokolamskoe Avenue, Moscow 125993, Russian Federation
kaf703@mai.ru

Abstract. Nowadays there are lots of means to automate the aircraft avionics software verification. The model-based software development made a formal verification approach popular (e.g. model checking). Such ways allow to verify the matching of the system model contained the end number of states and requirements expressed in the temporal logic language. However, the software operation in the target platform is out of scope here. Software verification within a hardware-in-the-loop simulation takes into account the target features. In this case, the verification automation tools are capable to simulate the input signals, to check the equipment's feedback by monitoring output signals, to process the captured data, and to generate the verification protocols. At the same time, there are some onboard high-critical systems (e.g. cockpit displays, flight warning) are being required the human-operator during the testing. They generate visual and aural information which has to be perceived by for the crew members. Thus, their verification is complicated to be automated accounting the mentioned aspect that leads to increased time and cost expenses. The paper describes the software tool that has been developed to partially automate the verification of avionics equipment which required human visual checking. The tool performs recognizing of the textual data displayed on the cockpit screens. It allows reducing the time and cost expenses within aircraft testing and recertification.

Keywords: Civil aircraft · Avionics · Software · Verification · Cockpit display · Image recognition · Data processing

1 Introduction

The modern aircraft avionics is a sophisticated intellectual complex. Currently, almost every vehicle system performs its functions using software contained a huge amount of code lines. At the same time, on-board software is one of the most sensitive parts in terms of the errors made during development.

The recent disasters caused by software errors include the crash of Boeing 737 MAX on October 29, 2018 near Jakarta and March 10, 2019 near Addis Ababa. The tragedies were caused by the incorrect operation of the maneuvering characteristics augmentation system. The results of the investigations showed that in both cases, the

B. Li et al. (Eds.): IoTaaS 2020, LNICST 346, pp. 329–335, 2021.
https://doi.org/10.1007/978-3-030-67514-1_26

failure of the angle of attack sensor led to the discordant data issuing which served as the reason for the nosediving (i.e. the described situation was not provided in the software) [1, 2]. Thus, the embedded software development process of aircraft onboard equipment requires the increased responsibility to ensure flight safety. This aspect is especially relevant for high-critical systems which failure can lead to an accident.

The process of onboard systems development is carried out according to V-model regulated by the ARP4754A standard (see Fig. 1) [3] at the current stage of aviation evolution. This approach describes a multi-iterative definition & design and integration & verification cycle with a focus on thorough product testing.

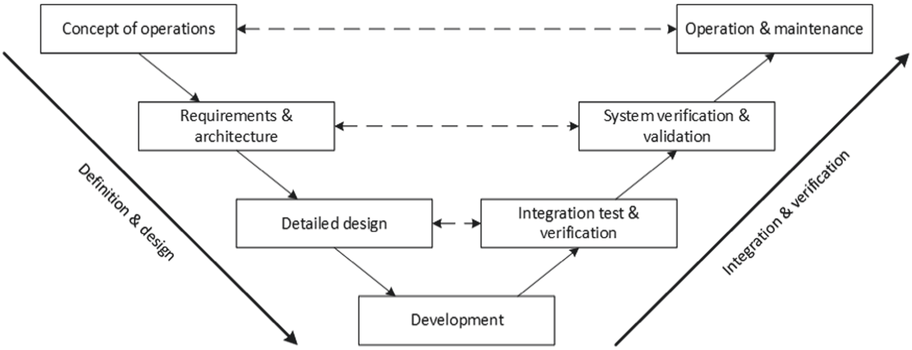

Fig. 1. V-model of aviation onboard systems development.

The listed V-model for the onboard equipment development includes the following nominal levels: aircraft, system, component, hardware, and software. Verification at every level is necessary for the product safe use in subsequent stages since it is aimed to assess the software and established requirements compliance as well as to detect and record errors potentially introduced.

Significant time is being spent on code verification. Taking into account the limited term of modern liner development, it is necessary to reduce the time expenses by using test automation tools for example. Among the such products providers are ScienceSoft (The USA), A1QA (The USA), Kualitatem (The USA), TestingXperts (Great Britain), Oxagile (The USA), AVIAOK (Russia), GosNIIAS (Russia), BugRaptors (India) [4].

There are a sufficient number of ready-made solutions for the software level but there are almost no automation tools for testing at the system level (although the amount of tests at this stage is also significant).

To date, the main necessary information for vehicle handling is displayed by the cockpit display system (CDS) contained the large-format liquid crystal displays placed in the cockpit.

The typical CDS of a modern civil aircraft includes:

- multifunctional displays (MFD);
- control panels;
- head-up display (optional);
- technical vision systems (optional).

The MFD is supposed to display a variety of formats such as:

- primary flight display;
- navigation display;
- engine and warning display (EWD);
- system pages (e.g. fuel, flight control, brake, air, doors), etc.

The EWD is intended to display information about the engines, basic data about onboard systems operation as well as the alerting messages about equipment failures obtained from flight warning system (FWS) (see Boeing 777 EWD example in Fig. 2).

Fig. 2. The Boeing 777 EWD example [5].

One of the most critical EWD zones is the FWS messages zone since it's aimed to warn the crew about the incorrect systems functioning and their failures.

Typically, the following message priority levels are available (from highest to lowest) [6]:

- "WARNING" (highest priority alarms which require the immediate corrective actions of the pilots in a short time);
- "CAUTION" (messages require the immediate crew notification with further corrective actions in a long time);
- "ADVISORY" (notifications that will probably require the pilot's actions without time limitations);
- "MEMO" (lowest priority messages which do not require the pilot's actions and intended to inform the crew about systems operation).

When “WARNING” and “CAUTION” messages appear, crew members have to confirm their reading by pressing the light buttons (master warning & master caution) installed on the dashboard. Unacknowledged messages are indicated on the EWD with a special marker – a symbol (for example, a rectangle) displayed in front of them.

2 Purpose of the Research

The purpose of the research is the tool development to automate the modern and perspective aircraft CDS software verification.

The goal achieving is ensured by a software application design that performing text data recognition (e.g. FWS messages displayed on the EWD). The input data for the tool is photos of CDS screens are being obtained under system verification on the bench. The detailed description of the proposed solution will be given in the next chapters.

3 Software and Algorithms Used for the Task Implementation

Task implementation is carried out through text recognition algorithms. They are applied to images obtained by framing a video stream from a web-camera installed opposite the CDS displays.

The following means were selected for the task implementation:

- Python 3 programming language for software development;
- Tesseract neural network for text recognition in the EWD photos;
- TechSAT ADS2 environment for setting signals values that provide the FWS messages generation.

Python 3 was chosen due to a large number of libraries for optical character recognition with enough scripts execution speed. The Tesseract provides high recognition accuracy by analyzing a lot of image features and hidden relationships between them. The TechSAT ADS2 represents an integrated software environment and hardware platform for prototyping aerospace engineering systems including:

- simulation within systems design;
- high-speed development cycle;
- integration of the leading industry solutions into a common platform;
- support for systems certification according to SAE ARP4754A [3] processes;
- support for interfaces widely used in the aerospace industry (e.g. ARINC 429 [7], ARINC 664 [8], ARINC 825 [9], etc.).

4 Description of Developed Software and Its Use During the Testing

The developed software performing the previously described functionality has a three-component architecture:

- input image processing module;
- logic module;
- output data generating module.

At first, the verifier sets the expected results matrices including the message's color, body, and markers presence. Next, the parameter values are set for generating messages on the EWD. After they are displayed on the CDS screen, the camera catches them in a video stream (see Fig. 3). The received video frames are fed alternately to the developed tool input.

Fig. 3. Avionics integration bench and webcam mounted on it

After, the software provides text recognition of the input image and generates a resulted matrix containing an array of recognized data. After, the comparison of the expected and resulted matrices is performed which allows to conclude the test has been passed or failed.

The webcam should be installed opposite the MFD parallel to the CDS display surface to minimize the optical distortion effects. One of the tasks of the verifier is to configure the camera to ensure minimal distortion in the generated video stream.

The webcam video is converted by the input image processing module to the form necessary for the subsequent recognition. Factually, the following operations are being performed: stream cropping, image rotation (if required), and required area cropping, contrast adjustment.

The processed frame is fed to the logic module which divides the image into text lines sequentially (see Fig. 4) which handled by Tesseract neural network. Markers are recognized by searching for the similar symbols (in particular, rectangles) among the reference images specified a priori. The lines color is determined by matching it to the set color ranges in HSV coordinates.

Fig. 4. Example of FWS messages set displayed on EWD

In the output data generating module all recognized messages output to the resulted array line-by-line. Then the expected and resulted matrices comparison is performed and the verification protocol is being generated in HTML or Microsoft Word formats (see Fig. 5).

1. General

Protocol ID: 223-368_20200527-tp
Protocol name: CDS_EWD in part of FWS messages
Date and time of verification performed: 27.05.2020, 09:10
Bench configuration: 96:28:9.2i

2. Tests performed

2.1 Test #1

#	Action	Expectation			Result			Conclusion	Comment
1.	<<RequirementID>>								
1.1	Set parameters values by uploading the following files placed in the attachment to TechSAT ADS 2: TEST_1.M.xls	Marker	Body	Color	Marker	Body	Color	PASSED	
		None	FCS SLATS JAM	Amber	None	BRK PLT PEDALS FAULT	Red		
		None	ENG L OIL PRESS LOW	Amber					
		None	BRK PLT PEDALS FAULT	Red	None	FCS SLATS JAM	Amber		
					None	ENG L OIL PRESS LOW	Amber		
		None	ADS 3 CAPT	White	None	HYD 2 ELEC PUMP ON	Green		
		None	HYD 2 ELEC PUMP ON	Green	None	BEACON LIGHT DIM	White		
		None	BEACON LIGHT DIM	White	None	ADS 3 CAPT	White		

Fig. 5. Example of generated verification protocol in Microsoft Word format

5 Testing of the Developed Software

To confirm the developed software operability, a series of about 1120 experiments were carried out. The results have been received show that the text and markers recognition accuracy is 97.19%, the color detection accuracy is 98.26%. Unsuccessful cases were related to the impossibility of detection (wrong recognitions have not been gotten).

Despite obtaining not 100% recognition accuracy, the obtained indicators allow to use the tool during testing. If the test fails, the verificator has to repeat it manually in part of undetected messages.

6 Conclusion

The developed software and the described methodology for its application can significantly reduce the verification time of formats containing text (in particular, EWD) and economic expenses for CDS verification at the system level.

The tool usage within Irkut MC-21 aircraft CDS testing reduced:

- number of manual checks in about 40–45 times respect to fully manual testing;
- verification time in 4 times;
- verification financial expenses in 8 times by reducing the number of verificators.

The mentioned indicators do not take into account neither the time savings are gotten within CDS certification nor an effect from the putting aircraft into exploitation earlier and the cost of errors potentially made by the verifiers during manual testing.

In the future, this project is planned to be finalized to verify not only textual but also graphic (symbolic) information, as well as to automate the FWS verification by recognizing audio signalization.

References

1. KNKT.18.10.35.04 Aircraft Accident Investigation Report. Lion Mentari Airlines. Boeing 737-8 (MAX) PK-LQP. Tanjung Karawang, West Java, Republic of Indonesia. 29 October 2018. Komite Nasional Keselamatan Transportasi Republic of Indonesia, Jakarta, Republic of Indonesia (2019)
2. AI-01/19 Interim Investigation Report on Accident to the B737-8 (MAX) Registered ET-AVJ operated by Ethiopian Airlines on 10 March 2019. Ministry of Transport of The Federal Democratic Republic of Ethiopia, Addis Ababa, Ethiopia (2020)
3. ARP4754A Guidelines for Development of Civil Aircraft and Systems. SAE International, Warrendale, The USA (2010)
4. Top 20 Software Testing Companies in 2019. https://medium.com/@andy_dassan/top-software-testing-companies-in-2019-c418b24f69d0. Accessed 10 June 2020
5. CARE INO III. 3D in 2D Planar Display Project. D4-1: Review of Cockpit Displays Design. EUROCONTROL, Brussels, Belgium (2008)
6. AC 25.1322-1 Flightcrew Alerting. FAA, Washington, The USA (2010)
7. ARINC Specification 429P1-17. MARK 33 Digital Information Transfer System (DITS). Part 1: Functional Description, Electrical Interface, Label Assignments and Word Formats. ARINC, Annapolis, The USA (2004)
8. ARINC Specification 664P1-1. Aircraft Data Network. Part 1: Systems Concepts and Overview. ARINC, Annapolis, The USA (2006)
9. ARINC Specification 825-2. General Standardization of CAN (Controller Area Network) Bus Protocol for Airborne Use. ARINC, Annapolis, The USA (2011)