



Verification and test case development method based on civil aircraft operation scenario

Hongyu Li¹ · Miao Wang¹ · Gang Xiao¹ · Guoqing Wang¹

Received: 21 November 2020 / Revised: 24 April 2021 / Accepted: 28 April 2021 / Published online: 10 May 2021
© Shanghai Jiao Tong University 2021

Abstract

Due to the characteristics of civil aircraft system integration, requirement-based testing cannot fully verify the complex cross-linking between multiple systems. With the gradual deepening of research on flight process in recent years, method of verification based on scenarios has been applied to test verification process of civil aircraft. This paper studies the verification process based on typical operation scenarios of civil aircraft. A method for establishing behavior model of operation scenario based on SysML is proposed. The model covers key parameters and behavior information of scenario which can support closed-loop simulation of aircraft. In addition, the method of generating test cases based on SysML model is optimized, and a set of use cases suitable for scenario-based testing is formed through the combination of test data and path. This paper takes the ground proximity warning function verification scenario as an example to show the method of scenario modeling and test case development, which can be applied to test verification process of operation scenario, and supports the closed-loop simulation and testing.

Keywords Operation scenario · SysML · Scenario-based verification · Test case

1 Introduction

To improve competitiveness, manufacturer apply new technologies in new aircraft type, which results in more complex systems, especially avionics systems. With the application of IMA, the degree of coupling between avionics systems has increased, presenting a highly comprehensive feature, which brings challenges to test verification. During the development process of civil aircraft, the design problems could be found through experimental verification as early as possible, which can reduce the workload of subsequent system design iterations, thereby shortening the design cycle. In addition, digital or semi-physical tests are used to replace

aircraft tests, which can reduce the workload of verification process and save development costs [1, 2].

2 Scenario-based verification

The development of civil aircraft follows the double V model process of SAE ARP4754A [3], that is, the civil aircraft must complete requirements capture, function analysis definition, validation and verification at aircraft level, system level, and equipment level. Decompose layer by layer according to aircraft level requirements to obtain system level and equipment level requirements, which conforms to the top-down development process. For a single device or system, it can be judged whether it meets the design requirements through test verification based on input and output. However, for system integration level or aircraft level verification, the traditional method of judging based on system response is difficult to fully discover problems. For example, the test verification result of a single system meets the design requirements, but when the system is tested in a comprehensive test or on-board test, it will be exposed to hidden problems due to factors such as verification scenario conditions and human-in-the-loop. Usually, the test verification of a

✉ Miao Wang
miaowang@sjtu.edu.cn

Hongyu Li
lihongyu39@sjtu.edu.cn

Gang Xiao
xiaogang@sjtu.edu.cn

Guoqing Wang
wang_guoqing@careri.com

¹ Shanghai Jiao Tong University, Shanghai, China

single system does not consider or possess the corresponding test conditions (Fig. 1).

To improve the efficiency of experimental verification, the concept of scenario-based verification in software engineering has been gradually promoted. The traditional requirement-based verification method needs to design test cases according to system requirements one by one. The scenario-based verification method considers the aircraft in a virtual operating scenario, and verifies the flight target, function or performance that the aircraft needs to achieve in the scenario. Scenario-based verification is more suitable for multi-system integration and human-in-the-loop complex cross-link verification (Fig. 1).

In recent years, there have been more studies on civil aircraft scenarios. Xu proposed the definition and development methods of flight scenarios from the perspective of airworthiness certification. The article established the mapping relationship between scenarios and minimum flight crew criteria [4]. Yin formulated flight scenarios for pilot workload analysis, and studied pilot–aircraft interaction and human factors in scenarios [5]. In the process of designing and verifying specific systems, scenario-based methods are gradually being promoted. Yu combined the MBSE R&D ideas and processes with the analysis of airborne radar antenna servo system. The article establishes task scenarios and models to support requirement capture and function analysis [6]. Zhang constructed MBSE process from requirement model to physical performance model according to the characteristics of commercial aviation engine design and development [7]. Based on practical engineering experience, Yuan proposed an aircraft system test verification method based on working conditions/scenarios. In the system comprehensive cascade test, the test process was determined by defining the working conditions of various aircraft in normal and fault conditions [8].

In the research of test verification methods based on scenario models, Zhu uses SysML as modeling language to establish unified conversion rules to complete MBSE process of requirement capture, function modeling and

requirement verification [9]. Feng proposed an optimization method based on UML activity diagram to automatically generate test paths [10], and Niu focused on the behavior model algorithm of coverage-based testing [11]. In addition, the research on generating test cases based on UML is relatively extensive, and many optimization algorithms have been proposed [12–15].

This paper proposes a method of verification and test case generation based on typical operating scenarios of civil aircraft. Different from the previous research, this article focuses on the operation scenarios of civil aircraft. In Chapter 3, the composition elements of scenario model are sorted out and refined, and the behavior model is established based on SysML. In Chapter 4, according to the key parameters of scenario model and behavior information, the test data and path are generated, respectively, and finally, the test case set is obtained through the combination of path and data.

3 Operation scenario modeling

The first task of scenario-based verification is to establish operation scenario model. Scenario is a combination of events with certain relationship, which refers to specific environment and aircraft status, including all stages of aircraft entire life cycle in its design, manufacturing, test flight, delivery, and operation. The construction of operation scenarios should be considered from the perspective of users, not only involving environmental factors, but also considering the status of aircraft itself, as well as the impact of stakeholders including crews, air traffic control, and airlines.

In the process of verification based on requirements, test cases need to be generated based on captured requirements. In scenario-based verification, requirements are captured and decomposed to generate corresponding scenarios as input, and then test cases are generated based on operation or task scenario. There is no one-to-one correspondence between requirements and scenarios. One requirement may need to be verified in several operation scenarios, and a single operation scenario may be able to verify more than one requirement.

For a complete scenario-based verification process, it is necessary to first establish scenarios that can be used to identify specific requirements. Due to the complex correspondence between aircraft requirements and scenarios as described above, it is possible to reduce the duplication of design and modeling efforts in subsequent validation by sorting out the specific requirements associated with individual scenarios as they are validated. This traceable requirements and scenario correspondence can guide the design of scenario-based verification upfront.

Therefore, the workflow of scenario-based verification could be described as searching for the relationship between requirements and scenarios, defining the operation scenarios

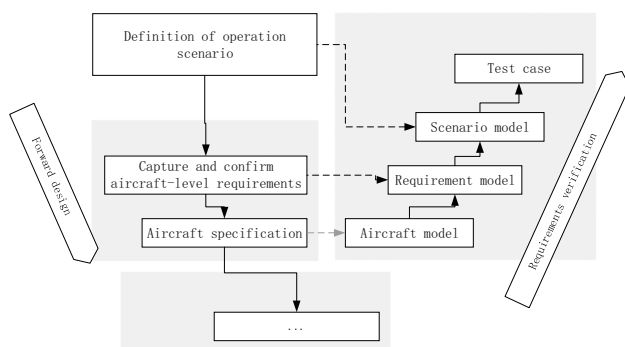


Fig. 1 Scenario-based verification process

corresponding to requirements that need to be verified, and developing test cases for experimental verification.

3.1 Definition of operation scenario

General operation scenarios are divided into normal operation scenarios, abnormal operation scenarios and failure scenarios. Among them, the normal scenario is the normal operation process of aircraft, covering all stages of flight, while the abnormal scenario refers to the unfavorable weather conditions and urgent air traffic control situation. The above two parts are defined according to specific flight objectives and flight phases. The failure scenario is for the typical aircraft system or equipment, and operation scenario is designed for its function failure. In other words, the failure scenario is different from above two types of scenarios in design idea.

According to the definition of operation scenario of civil aircraft, the scenario mainly describes the environment and aircraft status. Therefore, the composition of scenario mainly includes two parts: objective factors and subjective factors. The objective factors refer to weather environment and aircraft system status, while the subjective factors mainly involve tasks and operations of human-in-the-loop.

According to the key information types, the dimensions of operation scenario are divided into three parts, including environmental, aircraft, and task factors as shown in Fig. 2.

Environmental factor refers to the external environment where the aircraft is located, including weather, geography, and abnormal weather. Aircraft factor refer to the configuration of aircraft and malfunction. Task factors mainly refer to flight plan, crew task and permit approval. The detailed division is shown in Table 1. Environment, aircraft, and task factors are divided into different dimensions, and the three dimensions of space together constitute a complete space of operation scenario [16–18].

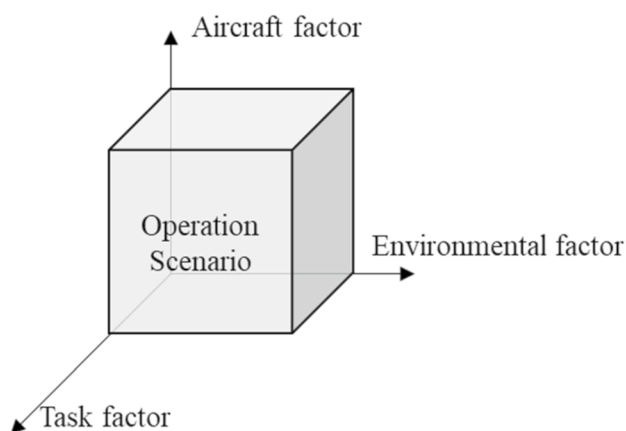


Fig. 2 Dimensions of operation scenario

Table 1 Dimensions of operation scenario

| Dimension | Sub-dimensions | |
|----------------------|---|---|
| Environmental factor | Weather | Wind speed Temperature Humidity |
| | Geography | Airport location Altitude Plateau climate |
| | Abnormal weather | Crosswind Thunderstorm Turbulence |
| Aircraft factor | Structure | Weight Balance Minimum equipment list |
| | Malfunction | Failure information Abnormal event |
| Task factor | Flight plan Crew task Air traffic control | |

3.2 Modeling

Operation scenario modeling is implemented by SysML, focusing on information transmission, operation interaction, status change and collaborative decision-making in each scenario.

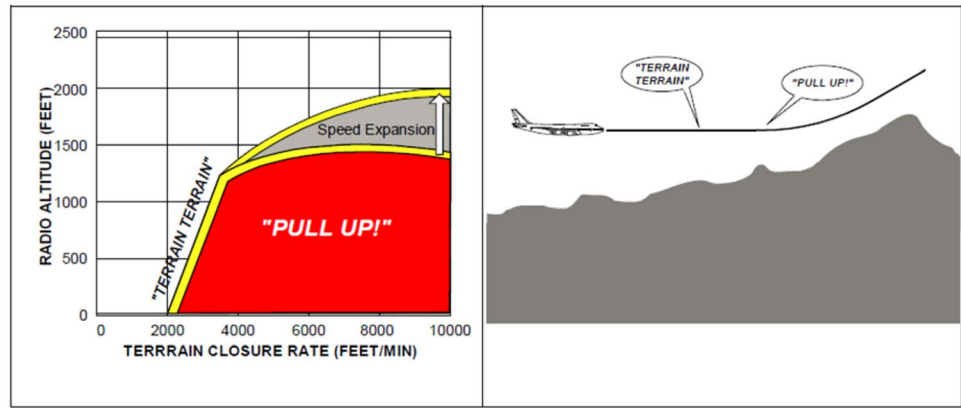
The modeling requirements for flight scenario should be divided into Aircraft (A/C), other aircraft, Air Traffic Control (ATC), Airlines Operations Center (AOC), etc. According to the flight requirements and purposes, the scenario should cover air-ground collaborative interaction, A/C-ATC-AOC cooperative interaction and other processes.

Taking the ground proximity warning function verification scenario during flight as an example, the behavior model diagram is established based on SysML. First, clarify the flight target of the scenario. During the airworthiness certification process of civil aircraft, it is necessary to complete the TAWS (Terrain Awareness Warning System) test flight certification test. The design of this scenario revolves around TAWS mode 2 (Excessive Terrain Closure Rate) test flight. The ultimate goal of scenario is to verify the TAWS warning function and aircraft's ability to maneuver safely.

TAWS mode 2 warning is divided into two trigger states. Among them, mode 2A covers most of the flight phases. The scenario established in this paper focus on mode 2A. The warning threshold curve of Mode 2A is shown in the left figure, and the right figure is the description of how airplane flight trigger mode 2A. The mode 2A warning threshold curve data used in this article comes from Honeywell's EGPWS (MKV-A) product description (Fig. 3).

After the design outline of operation scenario is determined, the main participants and the overall mission process are clarified around the flight requirements and objectives.

Fig. 3 EGPWS Mode 2A description



The ground proximity warning function verification scenario mostly revolves around aircraft and pilot, and it needs to report to area control center (ACC) during climbing. Therefore, the participants in this scenario include aircraft and ACC. The specific process of ground proximity warning function verification is to select the appropriate terrain and flight path, the aircraft flies along the expected trajectory to the landmark point, confirm the triggering of the multi-segment warning of TAWS mode 2A, then pull up and climb, report to ACC about altitude changing.

As the task flow is clarified, key dimensions and parameter values need to be selected in a targeted manner. It is not possible to perform permutation and combination tests on values of all dimensions. The key parameters involved in the ground proximity warning verification scenario mainly include terrain and altitude information of landmark points, the expected flight trajectory and speed, the logical judgment of TAWS, the recovery action and information exchange between A/C and ACC. Other parameters not listed above may also affect the flight. But from the point of view of ground proximity alarm function verification, it is not very relevant to the purpose of scenario design.

When using SysML behavior diagrams for modeling, participants, task processes, and key parameters need to be included. The scene model activity diagram is shown below, which mainly includes two participants, A/C and ACC. In the behavior model diagram, the green patterns represent behavior actions, the blue patterns represent conditional judgment and merge nodes, and the black words are related parameters (Fig. 4).

The activity diagram contains multiple loops and branch structures, describing the aircraft flying along expected trajectory, covering multiple results of triggered and untriggered warnings. As it involves looping to determine whether to trigger a warning, it is necessary to calculate the reserved time T according to initial parameters during scenario design, indicating that the aircraft is about to cross the landmark point. In this case, warning is not triggered yet, and the aircraft needs to be maneuvered directly. While

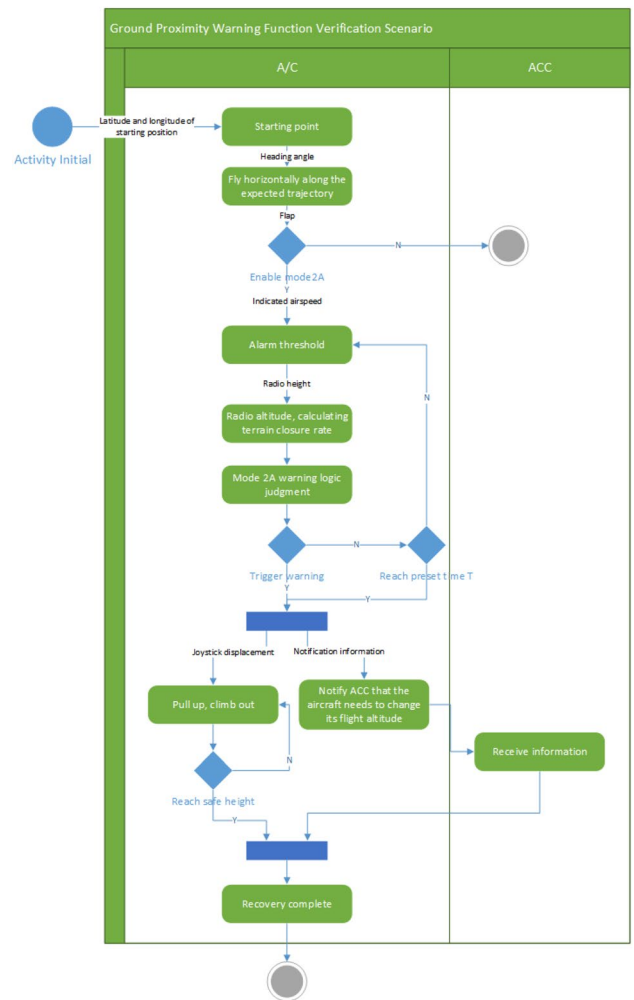


Fig. 4 Modeling of ground proximity warning function verification scenario

maneuvering, ACC needs to be notified of changes in flight level, involving branch structures of parallel actions.

The modeling of scenario is completed based on SysML behavior diagram. At the same time, parameters and timing

concepts are added to lay the foundation for subsequent development of test cases development.

4 Test case development

The development of test cases based on the scenario model starts from two aspects. The first is the generation of test data, and then the logical paths of behavior operations in the test scenario need to be combined to form test cases. The main process is shown in Fig. 5 below.

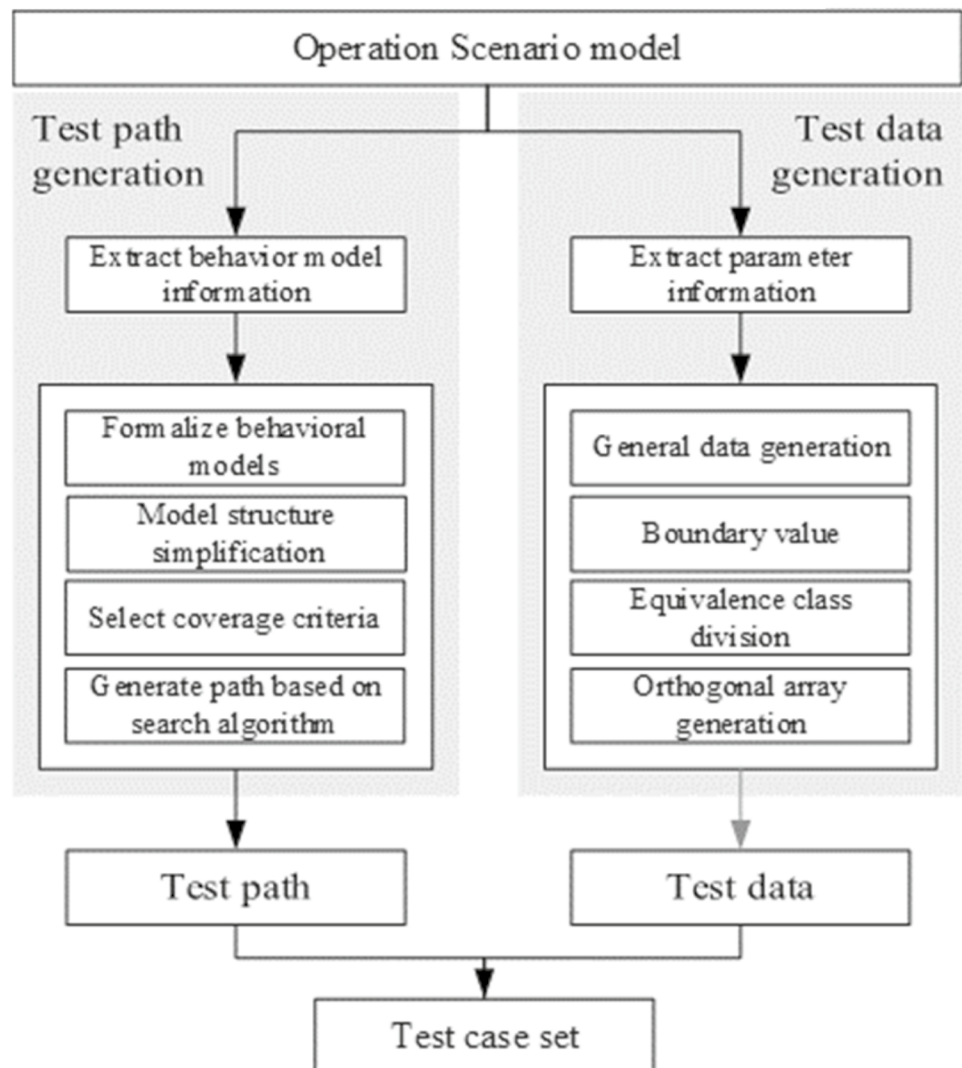
According to the definition of scenario-based testing in this article, test cases are composed of data and models. The test case is a time series data table covering key parameters of system that covers all participants. Through the attached time stamp, events such as aircraft system status changes, crew operations, tripartite coordination, and fault injection in the closed-loop simulation system can be realized.

4.1 Test data generation

The purpose of test data generation is to cover input conditions of all functional requirements in the test scenario. Therefore, based on the test requirements, the approximate value range of parameter is delineated. Refer to the detailed operation data to define the boundary values of key parameters in scenario and divide the effective and invalid equivalence classes.

The original data required for the test covers key parameters contained in the environment and aircraft factors, and all possible values of key parameters should be considered. For weather dimension in environmental factors, it is necessary to refer to ICAO or FAA's definitions of weather conditions for civil aviation operations to find the value ranges. For example, for the definition of wind speed, the relevant documents describe the wind speed less than ten knots as a case, so the wind speed of ten knots is a key value. For geographic dimensions, data from several typical domestic airports can

Fig. 5 Test case generation process



be selected to cover various airports such as international airports and plateau airports, as well as key information such as altitude, runway length, scene weather, and equipment. For abnormal weather conditions, also refer to the definition of abnormal weather in civil aviation regulations, including crosswind intensity, visibility, etc., to find the value range of key parameters.

Taking the ground proximity warning function verification scenario as an example, the key parameters are determined according to the judgment basis of TAWS, including indicated airspeed, radio altitude, and closure rate. Since the triggering of mode 2A warning depends on terrain, it is only necessary to refer to design outline of TAWS subject flight test and select typical landmark points and flight tracks. Therefore, the latitude and longitude of landmark point and initial position as well as the flight heading are no longer used as the basis for generating test data. This article also assumes that there is no direct coupling between radio altitude and closure rate, so as to fully verify the mode 2A warning threshold. The following are the possible value ranges of three key parameters in the scenario (Table 2).

After determining the value range of each parameter, the possible input data are divided into several equivalence classes, including valid equivalence classes and invalid equivalence classes. The equivalence class represents a set of valid or invalid states of input condition. For the test scenario, the effective equivalence class refers to reasonable and effective input data. If the aircraft weight does not exceed the maximum take-off weight in the take-off scenario, the weight value at this time is reasonable and effective. Exceeding the maximum take-off weight will cause the aircraft fail to take-off normally, and the data should be classified as invalid and equivalent. Selecting representative data from each part as the value classification of test case can ensure the coverage of test case and reduce unnecessary test cases in design.

Boundary value analysis is a supplement to equivalence class division. It selects test data on the "edge" of equivalence class to check its boundary value. If the input condition is specified as a range with a and b as boundary, the test data should include a , b , slightly greater than and slightly less than a and b . At the same time, corresponding test data should be designed for the predefined boundary values of internal data structure of the scenario (Table 3).

Table 2 Parameter value range

| Parameter | Ranges | |
|----------------------------|--------|------|
| Indicated airspeed (knots) | 200 | 320 |
| Radio altitude (ft) | 0 | 3000 |
| Terrain closure rate (FPM) | None | None |

Table 3 Parameter boundary value

| Parameter | Boundary value | | | |
|----------------------------|----------------|------|------|------|
| Indicated airspeed (knots) | 200 | | 320 | |
| Radio altitude (ft) | 30 | 1220 | 1650 | 2450 |
| Terrain closure rate (FPM) | 2038 | 3545 | 5733 | 9800 |

According to the key parameter list defined by scenario model, trace the specific meaning and value range, complete the equivalence class division and boundary value processing, and finally generate the corresponding test data as input of test scenario. At the same time, the data need to be combined with logical path to form a test case, that is, the basic test path needs to be generated from SysML behavior model.

4.2 Test path generation

Considering the characteristics of SysML behavior model, the scenario model is first defined formally, and then simplified into a structured directed graph. The search algorithm is used to traverse and combined with test coverage criteria to generate test path. Since the direct traversal of behavior model in a complex flight scenario may lead to path explosion, the loop and concurrency modules in the model need to be simplified first.

Formal definition refers to the transformation of unstructured SysML behavior models into tuple based on the principles of mathematical sets. Each graph corresponds to a tuple. The tuple contains action state node, object node, relationship between nodes, flow relationship between state and object, as well as the collection of initial nodes, end node. A tuple can formally describe the activity diagram completely.

Refer to the formal method of SysML, define the activity diagram as a tuple form, $AD = (A, O, T, C, F, a_i, a_f)$, where A is the set of active states, and O is the object node set, T is the set of transition edges, C is the control condition on the transition relationship, and F is the flow relationship between state and object. a_i represents the initial activity state; a_f represents the termination activity state; DN (Decision Node) is the branch node; MN (Merge Node) is the branch merge node; FN (Fork Node) is the concurrent bifurcation node; JN (Join Node) is Concurrent convergence node [19–21] (Fig. 6).

After the formal definition, the loop structure and concurrent structure of behavior diagram need to be identified and simplified. The loop module refers to the loop part, from loop entrance to exit node. The entrance, exit nodes, and all nodes on the path form the entire cycle module. The loop module is simplified into a branch structure, and the compound node L is used to represent the loop path. The test path passes through the compound node L once, which means it passes through the loop path once (Fig. 7).

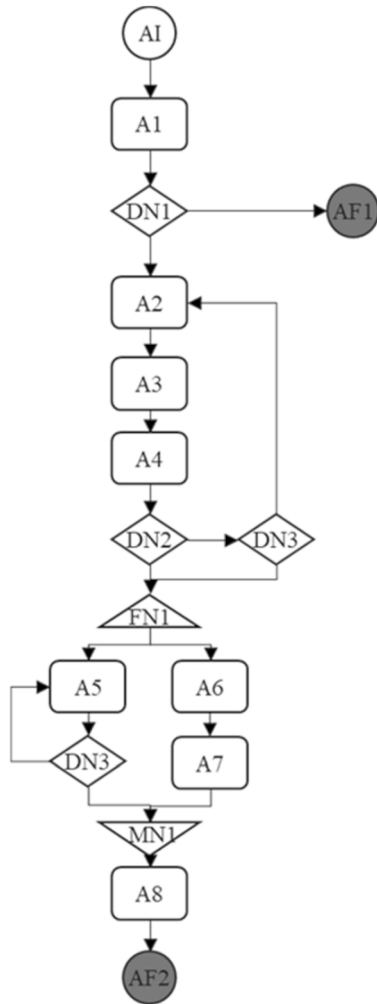


Fig. 6 Simplify the model to a control flow graph

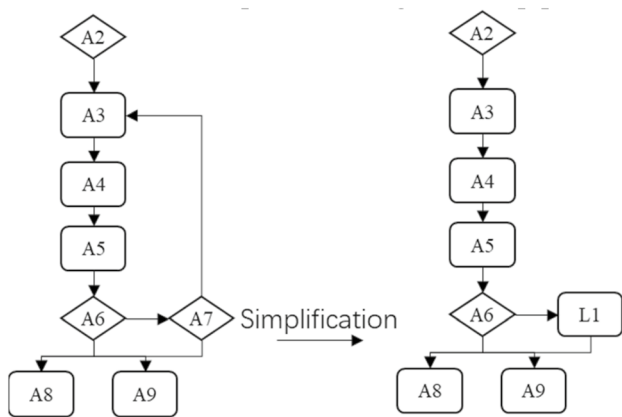


Fig. 7 Loop module simplification

The concurrency module refers to the existence of multiple paths between the entry and exit nodes. Replace the concurrent module with composite node X. Similarly, the

test path passes through the composite node X once, which means it passes through the concurrent module once (Fig. 8).

The basic idea of test path generation method based on search algorithm is to transform the test path generation problem into optimization problem of function solution, and then solve it through a deep search algorithm. In the process of problem transformation, it is necessary to select test coverage standard according to actual test requirements. Then, the coverage criterion is abstracted as a fitness function to guide the candidate solution set to evolve toward the optimal solution. The test path set that meets the conditions is obtained after multiple searches.

This article uses basic path coverage criteria to generate test paths based on search algorithms. Traverse the nodes in set A and add them to the test path one by one. For branch nodes of the simplified model, the corresponding test path is obtained according to the number of branches. And continue to traverse in subsequent nodes, until traversing to the end of active node. The traversal ends and the final test path collection is obtained (Fig. 9).

After searching the behavior diagram to obtain test path set, it is also necessary to sort the interior of simplified structure described above to generate internal test path. Combine the two parts to get the final test path set.

4.3 Test case generation

According to the relationship between behavior and parameter information in behavior model, the test path and data are combined to obtain final test case. Take the ground proximity warning function verification scenario in this article as an example. As test path is determined, the judgment conditions on the path can be converted into constraints of key parameters. Filter the generated test data based on constraints. All data combinations that conform to the current test path should be combined with the path to form test cases, so that the coverage rate of scenario can be achieved.

The test cases applicable to this scenario are given below for reference.

Test case 1: Start → Aircraft flies along the expected trajectory from initial position → The flap position is not in the

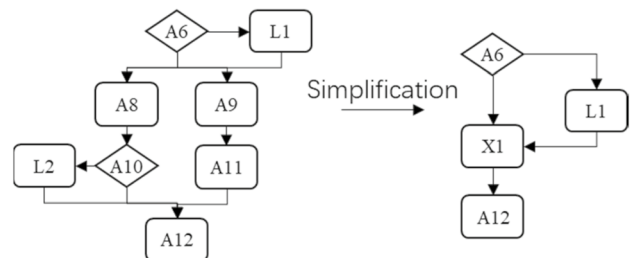


Fig. 8 Concurrency module simplification

Fig. 9 Search algorithm

Algorithm: Test path generation

```

: Node
: TestPath
1: Function CreateTestPath(TestPath, Node)
2:   CurrentNode ← Node
3:   if CurrentNode = aI then TestPath add CurrentNode
4:   DeepSearch(TestPath, CurrentNode)
5:   if CurrentNode = aF then return TestPath
6:   if CurrentNode = A or CN then TestPath add CurrentNode
7:   if CurrentNode = DN then
8:     for e in DN
9:       Copy (TestPath[e], TestPath)
10:      CurrentNode ← Node
11:      DeepSearch(TestPath, CurrentNode)
12:  return TestPath

```

landing configuration → Mode 2A can be triggered → The current warning threshold is determined according to the airspeed of 252 knots → Min terrain clearance is 1500 feet, closure rate is 4000 fpm, and closure rate drops by 20 per second → Failed to enable the mode 2A warning → Circulation judgment → Exceeding the reserved time T → Pull up and maneuver → Notify ACC → Climb to safe level.

Test case 2: Start → Aircraft flies along the expected trajectory from initial position → The flap position is not in the landing configuration → Mode 2A can be triggered → The current warning threshold is determined according to the airspeed of 220 knots → Min terrain clearance is 1400 feet, closure rate is 3545 fpm, altitude change rate is -20 fps → Enable mode 2A warning in 10 s → Pull up and maneuver → Notify ACC → Climb to safe level.

Test cases can be converted into time series data tables, which can be used as input and output for simulation verification, such as MBSE modeling tool simulation, MATLAB real-time simulation, etc. Taking test case 2 as an example, the relevant parameters can be converted into time series data table shown in Fig. 10 below.

The threshold curve of MKV-A can be used to illustrate the process of test case 2. As the min terrain clearance decreases, the aircraft state can trigger a warning in Mode 2A, corresponding to the red mark in Fig. 11. After the recovery maneuver, the aircraft's altitude gradually increased and the warning stopped.

| radio_altitude | altitude_rate | correcte | flap | warning | Elevator | inform2acc |
|----------------|---------------|----------|------|---------|----------|------------|
| 1400 | 3545 | 220 | 0 | 0 | 0 | 0 |
| 1380 | 3545 | 220 | 0 | 0 | 0 | 0 |
| 1360 | 3545 | 220 | 0 | 0 | 0 | 0 |
| 1340 | 3545 | 220 | 0 | 0 | 0 | 0 |
| 1320 | 3545 | 220 | 0 | 0 | 0 | 0 |
| 1300 | 3545 | 220 | 0 | 0 | 0 | 0 |
| 1280 | 3545 | 220 | 0 | 0 | 0 | 0 |
| 1260 | 3545 | 220 | 0 | 0 | 0 | 0 |
| 1240 | 3545 | 220 | 0 | 0 | 0 | 0 |
| 1220 | 3545 | 220 | 0 | 1 | 0 | 0 |
| 1200 | 3545 | 220 | 0 | 1 | 1 | 1 |
| 1220 | 3545 | 220 | 0 | 1 | 1 | 0 |
| 1240 | 3545 | 220 | 0 | 0 | 1 | 0 |
| 1260 | 3545 | 220 | 0 | 0 | 1 | 0 |
| 1280 | 3545 | 220 | 0 | 0 | 1 | 0 |

Fig. 10 Part of the data converted by test case 2

The Aerospace Blockset toolbox of MATLAB was used to build the simulation model, and the parameter timing table of the test case was imported through the reserved interface for simulation testing, and the simulated "Warning" alarm results were compared with the expected results, as shown in Fig. 12.

The ground proximity warning function only needs to verify that the warning logic meets the design requirements and has no relation to the aerodynamic performance of the aircraft itself, only the min terrain clearance and closure rate as parameters for logical judgement. The results are, therefore, influenced by the accuracy of actual

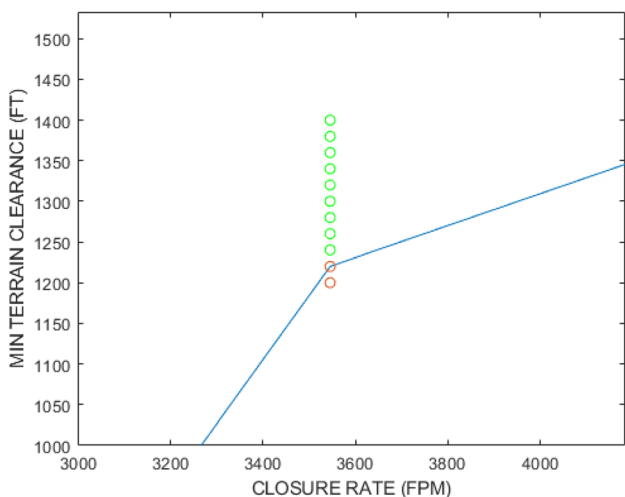


Fig. 11 Aircraft state and threshold curve

terrain data and the simulation results may differ from the expected results.

5 Conclusion

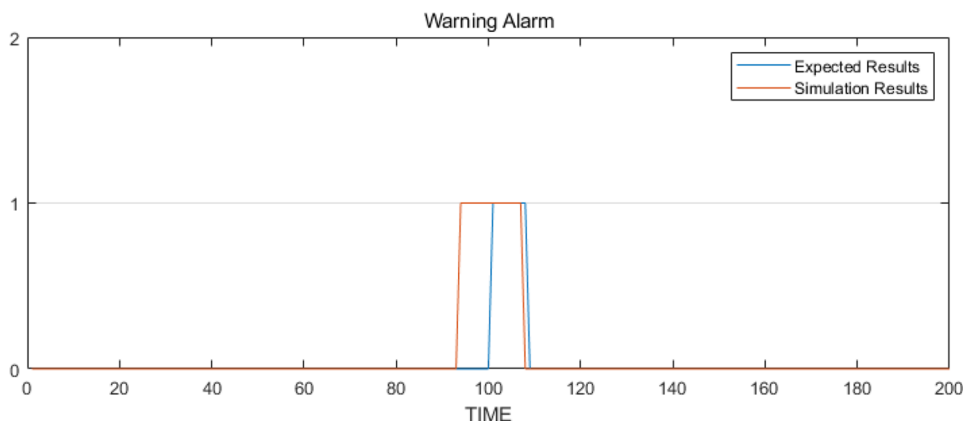
This paper proposes the verification method based on typical civil aircraft operation scenario. The verification process includes scenario definition, modeling, and test case generation. Based on the characteristics of civil aircraft operation scenarios, the article uses SysML behavior diagrams containing key parameters and timing information for modeling, which can support subsequent scenario-based verification and closed-loop simulation. In view of the complex structure of scenario model, during the formalization process of SysML model, the article identifies and simplifies the loop and branch structure that

easily lead to path explosions. Search algorithm is used to improve the efficiency of test case generation. This method can be extended to other verification process for typical operation scenarios.

Compared with the method of Requirement-based Verification, the Scenario-based Verification focus on cross-linking relationship between different systems. In the complex operation scenarios of civil aircraft, this is reflected in the interaction between different aircraft systems and different participants. The method of generating test cases based on scenarios follows the concept of model-based testing. The difference lies in the need to build test scenarios from flight process. The verification of operating scenario can meet the coverage requirements of test target more easily.

In future, the work will focus on improvement, optimization and promotion of method. First of all, scenario modeling needs to make full use of activity diagram, sequence diagram and state diagram. The activity diagram model that has been completed so far is inferior to other model diagrams in the performance of system state transition and timing relationship; second, in the process of generating test data, consideration should be given to scenarios where there are many key parameters. Using the method of generating orthogonal test arrays can reduce the number of test trials; finally, the combination method of test data and path needs to be standardized to improve the efficiency of test case generation, thereby supporting the verification of more complex scenario models.

Fig. 12 Comparison of expected results and simulation results



Acknowledgements This paper is sponsored by National Program on Key Basic Research Project (2014CB744903), National Natural Science Foundation of China (61673270), Natural Science Foundation of Shanghai (20ZR1427800), New Young Teachers Launch Program of Shanghai Jiao Tong University (20X100040036), Shanghai Pujiang Program (16PJD028), Shanghai Industrial Strengthening Project (GYQJ-2017-5-08), Shanghai Science and Technology Committee Research Project (17DZ1204304) and Shanghai Engineering Research Center of Civil Aircraft Flight Testing.

References

1. Qing L, Shuang W (2017) Research on virtual integrated test technology of civil aircraft[J]. *Civil Aircraft Design Res* 2017(02):1–7
2. Yang W (2020) Research and simulation of model-based system engineering in avionics system design[J]. *Digital Technol Appl* 38(02):128–130
3. SAE International (2010) Guidelines for development of civil aircraft and systems: ARP 4754A [S]. SAE Aerospace Recommended Practice
4. Minmin X, Yuwen J (2014) Study on flight scenarios for airworthiness certification[J]. *Civil Aircraft Design Res* 02:66–69+83
5. Yin T, Shan Fu (2014) Workload-oriented flight scenario for evaluation of pilot-aircraft interaction. *Procedia Eng* 80:656–667
6. Chi Y, Chao Y, Gangfeng Z (2019) Model based airborne radar antenna servo system analysis for mission scenario-oriented[C]. *Proceedings of the 2019 (4th) Chinese Aeronautics Science and Technology Conference*, pp 1098–1104
7. Yujin Z, Bo H, Wenhe L (2020) Research on MBSE unified modeling and design method of commercial aeroengine for operation scenario[J]. *Comput Integr Manufact Syst* 1–17
8. Wenqi Y (2019) Aircraft system test verification method based on working conditions/scenarios[J]. *Civil Aircraft Design Res* 01:98–101
9. Zhu S, Tang J, Gauthier J-M, Faudou R (2019) A formal approach using SysML for capturing functional requirements in avionics domain[J]. *Chin J Aeronaut* 32(12):2717–2726
10. Tingting F, Zhenyu L (2019) Optimization of automatic generation of test scenarios based on UML activity diagram model[J]. *Electron Design Eng* 27(24):152–156
11. Yingbei N (2020) Construction and application of aerospace software test model[J]. *Software* 41(03):268–271
12. Qing Li, Shucheng H (2017) A test case generation method based on UML activity diagram[J]. *J Jiangsu Univ Sci Technol (Natural Science Edition)* 31(03):333–338
13. Cuijuan C (2017) Research on the strategy of scenario method test case generation based on UML activity diagram[J]. *J Chifeng Univ (Natural Science Edition)* 33(10):24–25
14. Kundu D, Sarma M, Samanta D (2015) A UML model-based approach to detect infeasible paths. *J Syst Softw* 107:71–92
15. Yin Y, Yiqun Xu, Miao W et al (2017) An automated test case generation approach based on activity diagrams of SysML. *Int J Perform Eng* 13(6):922
16. United States. Federal Aviation Administration (2018) Advisory circular 25–7D: flight test guide for certification of transport category airplanes. Federal Aviation Administration, Washington, DC
17. United States. Federal Aviation Administration (2017) Advisory circular 120–71B: standard operating procedures and pilot monitoring duties for flight deck crewmembers. Federal Aviation Administration, Washington, DC
18. International Civil Aviation Organization (2010) Meteorological service for International Air Navigation: annex 3 to the convention on International Civil Aviation[M]. International Civil Aviation Organization
19. Huifeng C, Xiaofei Y, Jianmin J (2018) Checking the correctness of UML activity diagrams[J]. *Softw Eng* 021(003):5–9
20. Hebiao Y, Yunping Li (2011) Functional test scenario generation method based on UML activity diagram[J]. *Comput Eng* 37(21):55–57
21. Wentao H, Yanzhou Z, Nan S, Yawei Y (2015) Formal description of SysML activity diagram based on new activity calculus[J]. *Comput Appl Softw* 32(10):49–53