

DOI:10.16356/j.1005-2615.2025.05.020

FADEC 软件需求形式化建模与测试用例 生成的实例研究

董泽华¹, 胡 军¹, 沈翔宇², 熊 波², 董亚炯¹, 戴嘉磊¹

(1. 南京航空航天大学计算机科学与技术学院, 南京 211106; 2. 中国航发控制系统研究所, 无锡 214000)

摘要: 研发具有安全关键特征的全权限数字化发动机控制(Full authority digital engine control, FADEC)软件是当前大飞机航空发动机控制系统研制的重要任务。本文工作面向大飞机FADEC软件研发中的需求分析与测试挑战,基于变量关系模型(Variable relation model, VRM)提出了对条目化自然语言描述的FADEC软件需求形式化建模、分析和基于模型测试用例生成的技术方法,并对FADEC中启动燃油控制软件(Start fuel control, SFC)功能实例进行了研究。研究包括对FADEC自然语言需求文档开展结构化预处理,生成领域概念库;通过需求规范化生成形式化建模框架;基于变量关系模型开展多范式的分析;基于需求模型自动生成测试用例;对FADEC需求建模分析中的领域特征问题进行总结分析等。本文对FADEC软件需求提供了建模与测试的工程经验。

关键词: 计算机软件与理论; 机载软件形式化建模; 变量关系模型; 自然语言需求建模; 测试用例自动生成

中图分类号: TP311.5

文献标志码: A

文章编号: 1005-2615(2025)05-0999-14

Case Study on Formal Modeling and Test Case Generation for FADEC Software Requirements

DONG Zehua¹, HU Jun¹, SHEN Xiangyu², XIONG Bo², DONG Yajiong¹, DAI Jialei¹

(1. College of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 211106, China;

2. AECC Aero Engine Control System Institute, Wuxi 214000, China)

Abstract: The development of full authority digital engine control (FADEC) software, characterized by its safety-critical nature, is a pivotal task in advancing control systems for large aircraft aero-engines. This study tackles key challenges in requirement analysis and testing encountered during FADEC software development. It introduces a formal modeling method based on the variable relation model (VRM) for specifying FADEC software requirements that are initially described in itemized natural language. The work further investigates techniques for automatically generating test cases from the resulting requirement model and demonstrates the entire approach through a functional case study based on the start fuel control (SFC) software within a FADEC system. The methodology encompasses: The structured preprocessing of natural language requirements to build a domain concept repository; the establishment of a formal modeling framework through requirement standardization, the multi-paradigm analysis using the VRM, the automated test case generation, and a critical analysis of domain-specific challenges in FADEC requirement modeling and validation. This paper provides engineering experience for modeling and test of FADEC software requirements.

基金项目: 国家自然科学基金和“叶企孙”联合基金重点项目(U2241216; Y2022-V-0001-0027)。

收稿日期: 2025-05-19; **修订日期:** 2025-09-07

通信作者: 胡军, 男, 副教授, E-mail: hujun@nuaa.edu.cn。

引用格式: 董泽华, 胡军, 沈翔宇, 等. FADEC 软件需求形式化建模与测试用例生成的实例研究[J]. 南京航空航天大学学报(自然科学版), 2025, 57(5): 999-1012. DONG Zehua, HU Jun, SHEN Xiangyu, et al. Case study on formal modeling and test case generation for FADEC software requirements[J]. Journal of Nanjing University of Aeronautics & Astronautics(Natural Science Edition), 2025, 57(5): 999-1012.

Key words: computer software and theory; formal modeling for airborne system; variable relation model (VRM); natural language requirement modeling; automatic test case generation

航空发动机是现代飞行器的核心系统之一。图 1 展示了航空发动机的控制方式已经从液压机械控制、电子系统控制发展到现在主流的全权限数字化发动机控制(Full authority digital engine control, FADEC)^[1]。FADEC 的重要特征是通过计算机软件对发动机的功能与性能进行数字化精确控制。目前国际上大型飞机的发动机控制系统均采用 FADEC 方式来实现诸如余度管理、智能管理和架构冗余等先进功能,其软件系统复杂度呈快速增长趋势,这对研制现代 FADEC 软件系统并保证其满足航空领域的高安全性要求带来了重要挑战^[2-3]。

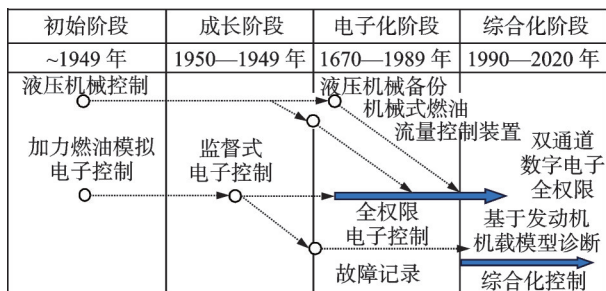


图 1 发动机控制阶段发展

Fig.1 Development stages of engine control

民用大飞机中的 FADEC 软件需要通过适航安全符合性认证,即 FADEC 软件系统的研制必须满足国际机载软件适航标准 RTCA DO-178C《机载系统和设备合格审定中的软件考虑》^[4-5]。DO-178C 标准的核心要点在于以机载软件的需求为核心,展开机载软件研发的各类任务,如:软件测试验证活动中必须包括基于需求的测试用例设计和执行验证等。DO-178C 中还对基于形式化方法的机载软件分析与验证技术给出了明确的要求,即 DO-333 标准^[6-9]。计算机科学领域中的形式化方法具备在工程领域提升机载软件安全性的能力^[10],但是如何应用形式化方法理论,并结合实际 FADEC 应用领域来形成航空工程中适用且有效的方法和工具,以对 FADEC 软件应用需求进行构造、分析与验证,在国内外仍然都是一个非常大的挑战。

基于模型的系统工程(Model-based system engineering, MBSE)^[11-12]是面向复杂系统研制工程的一种系统建模、分析与验证的方法学。在 DO-178C 的附加适航标准 DO-331^[13-14]中也给出了 MBSE 方法在航空领域中应用的要求,其基本思想

是通过对复杂机载软件研制过程中的各类阶段制品(如:软件需求、软件设计等)进行多层级建模,使得系统能够基于各类软件模型展开功能以及安全性方面的分析与验证,以尽可能在软件研制早期发现潜在的缺陷错误,并且还能应用严格的形式化方法和测试用例的自动生成提供良好的软件系统模型。

本文工作面向大飞机 FADEC 软件系统的测试验证问题,采用关系变量模型(Variable relation model, VRM)^[15-17]对一些典型的 FADEC 软件需求进行形式化表格的建模,提出了一个对 FADEC 软件需求形式化建模、分析到基于模型测试用例生成的技术框架;开展了实例分析研究,且提炼了对 FADEC 软件需求进行建模与测试的工程经验。

1 背景知识

1.1 发动机控制系统的数字化发展

基于技术演进的维度,航空发动机控制系统在过去的几十年中经历了非常大的发展和变化,其系统控制变量从 20 世纪 40 年代的单一参数发展到当代变循环发动机的数十个参数体系^[18];系统架构从液压机械控制升级为以发动机控制软件为核心的 FADEC 双通道数字化控制方式,并在软件功能层面实现了解析余度、伺服回路判故等先进技术;目前正在朝着深度数字化的“智能多电”架构发展^[19-20]。高安全高可靠发动机控制软件的研发已经成为现代航空发动机技术快速发展的核心任务之一。

FADEC 系统的核心架构采用双通道冗余容错设计,如图 2 所示。主控通道与热备通道在独立运行状态下通过共享式故障切换模块实现协同工作。当主控通道触发实时故障检测机制时,系统执行无扰动通道切换逻辑,将控制权移交至热备通道,同时启动内置测试设备对故障通道实施诊断与恢复操作。该架构通过通道级冗余设计与无缝切换机制,保障航空动力控制系统的高可靠性运行。

双通道冗余架构的形式化建模需解决多源信号协同处理与容错逻辑验证等关键问题。在信号采集阶段,控制模块需对双通道输入的异构数据实施实时仲裁与有效性表决;当单通道发生信号失效时,系统需在时间窗内完成健康通道无缝切换并激活故障通道的自修复流程。

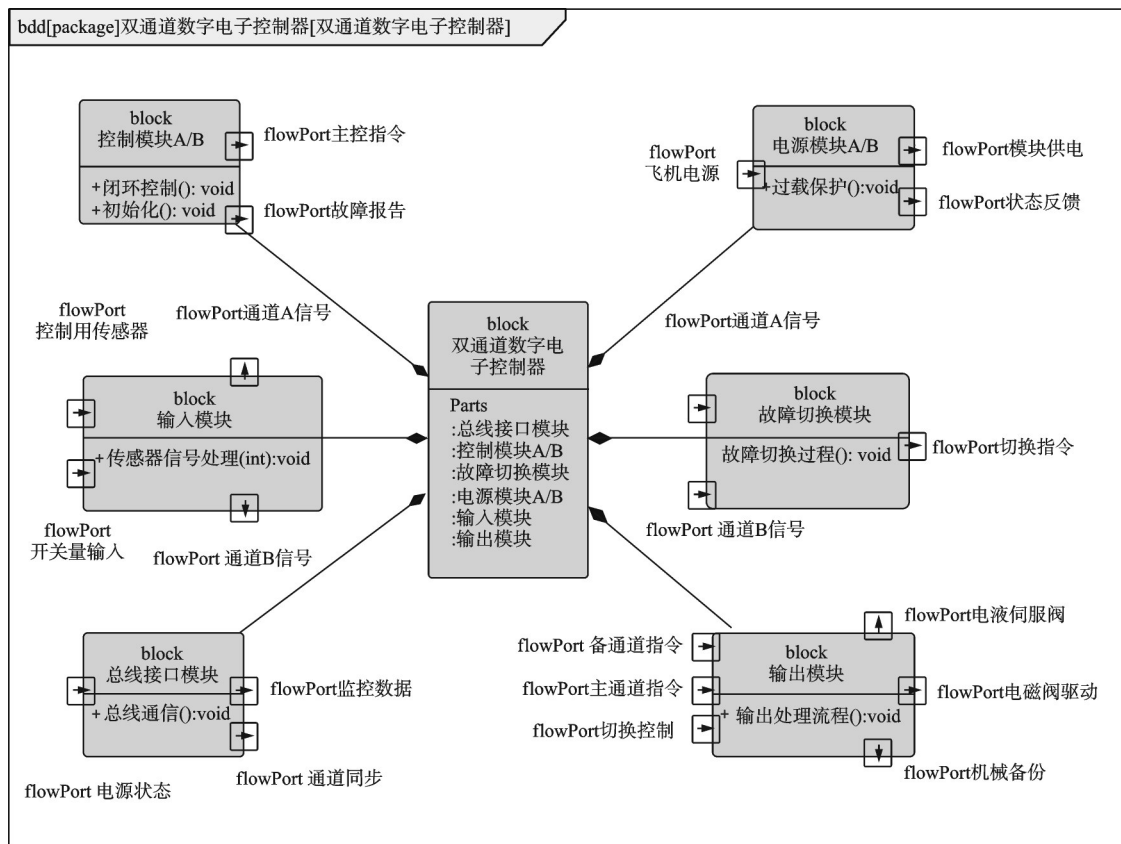


图 2 双通道数字电子控制器组成结构

Fig.2 Composition structure of dual-channel digital electronic controller

1.2 VRM 形式化需求建模与分析

基于模型的形式化验证领域中有很多形式化方法,但是大部分方法采用符号和逻辑公式来描述形式化模型,不熟悉形式化符号的工程人员难以有效地应用这些形式化方法。因此,本文采用了一种表格化的形式化变量关系需求模型 VRM^[15-17]来对 FADEC 控制软件的需求进行建模分析与测试用例生成。VRM 模型基于航空领域工程人员经常使用的二维表格关系,同时具备严格的形式化语义,在工程实践中已逐步被领域工程人员理解和应用。VRM 模型来源于四变量模型^[21]和需求状态机语言 (Requirement state machine language, RSML)^[22],其形式化语法和语义定义参见文献^[23-26]。在前期相关的研究工作中,本文研究团队已经构建了一个以 VRM 形式化模型为核心的软件需求建模分析与验证的软件工具 ART^[16-17,23-24]。ART 面向航空机载的安全关键软件需求,简化了四变量模型中的监视变量和控制变量集合,引入中间变量对模型进行分层处理^[23],融合了 RSML 语言中的 and-or 表格表达范式,可以将复杂的谓词逻辑关系以二维表格形式描述,并实现了需求模型分析验证的核心算法群。本文工作是在 ART 平台的基础上,研究 FADEC 软件需求里

与发动机控制相关的双通道表决、信号重构等领域特征,扩展 VRM 领域建模的方法和能力,使得 ART 工具能够有效地处理 FADEC 需求建模和测试用例生成。

2 相关研究工作

目前在航空应用领域,从机载软件需求的角度来看国外的相关理论、技术和工具的发展情况,大致可以分为如下 4 类。(1)从实际安全关键系统的开发工程经验中形成的理论与技术,如:四变量理论模型^[21]、SCR (Software cost reduction) 方法^[27-28]、T-VEC 工具^[29]等。(2)从通用的软件工程领域产生的软件需求规约方法,如:统一建模语言 UML^[30-31]中的用例 (UseCase) 模型的需求捕获与描述方法以及 StateCharts 状态机模型^[32]、从 UML 扩展而来的系统建模语言 SysML^[33]中用于描述系统需求的参数模型,其典型的工具包括 Raphso-dy^[34]、Statmate^[35]等。(3)从电子硬件系统设计的同步数据流语言发展而来的需求建模与代码生成技术,如:MATLAB 公司的 Simulink 工具^[36]、基于 Esterel 技术的 SCADE 工具^[37]等。其他一部分面向自然语言需求的工作包括文献^[38-39],此类工作的主要思想是通过模板化的手段将自然语言

总体上,燃油控制系统起动流程分为4个阶段:首先由油泵建立燃油压力,抽取油箱燃油加压至系统管路,形成稳定喷射所需憋压;随后燃油持续填充至总管储备,确保后续供油响应速度;当发动机转速达到阈值且点火信号触发时,系统向气缸喷射定量燃油,形成空燃混合气;最终在发动机稳定运行后切换闭环模式,通过氧传感器实时反馈燃烧数据,动态修正喷油脉宽以维持最佳空燃比,实现排放与效能的平衡控制。这些阶段过程都由SFC功能模块进行控制。

3.1 条目化需求的预处理

在目前的FADEC需求验证系统工程实践中,是以条目化的自然语言形式来描述软件的需求内容,但是自然语言描述的需求存在结构化语义不足和二义性等潜在问题,这难以满足FADEC软件对安全关键需求的要求。因此,首先需要对自然语言描述的需求进行处理,规范化VRM建模元素,构建可复用的领域概念库,其处理流程如图4所示^[23]。

表1中给出了经初步规范化处理的SFC软件需求的部分条目示例。根据VRM形式化模型中基于模式转换触发机制与状态保持特征,提出基于需求语义的条目分类方法:模式转换型需求、条件型需求和事件型需求。例如:(1)需求771因包含“第二阶段供油→第三阶段供油”的显式模态迁移,归为模式转换型;(2)需求768规定 $Speed_Rate <$

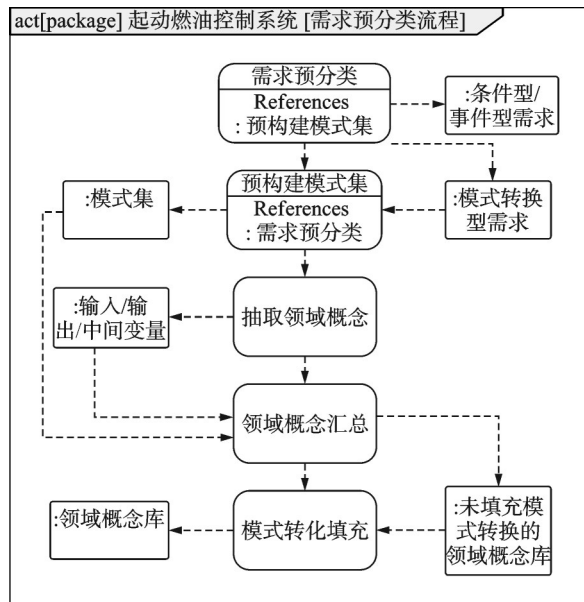


图4 需求的预处理流程框架

Fig.4 Preprocessing framework for requirements

7%时需保持油针最小位置与电磁阀接通状态,划为条件型;(3)需求770要求 $\text{Speed_Rate} \geq 7\%$ 时断开电磁阀,因其触发条件依赖前序状态($\text{Speed_Rate} < 7\%$)的迁移过程,应定义为事件型。对表1进行分类后,结果如表2所示。

根据模式转换型需求,进行模式集的预构建。VRM领域概念库中允许建立多个模式集,每个模式集的定义为

$$M = \langle \text{name, Modes, Conversions} \rangle \quad (1)$$

表1 SFC需求条目示例

Table 1 Examples of SFC requirement items

编号	需求内容
768	Speed_Threshold 未达到 7% 前,必须将油针关闭在最小位置且接通停车电磁阀。
770	若 Speed_Threshold $\geq$ 20%,必须断开停车电磁阀。
779	若 Speed_Threshold $\geq$ 20%,必须按 25 kg/h 的流量进行燃料主路充填。
771	若 Speed_Threshold $\geq$ 20% 持续时间达到 1 s 或 Speed_Threshold $\geq$ 25%,则第二阶段供油结束,必须进行第三阶段供油。
781	必须根据 Speed_Threshold 转速进行开环基准流量的计算。
783	必须根据 Tempt 和 Pressure 对基准流量进行修正。
789	必须根据 Tempt 修正系数对基准流量进行修正。
791	必须根据 Pressure 修正系数对基准流量进行修正。
793	必须根据起动时刻 tExhauststart 和 Pressure_Inlet 计算热机修正系数,对基准流量进行修正。
800	必须根据 Speed_Threshold 计算 SpeedGradient 基准。
802	必须根据 Temp 和 Pressure_Inlet 对 SpeedGradient 基准进行修正。
804	必须根据 Pressure_Inlet 对 SpeedGradient 基准进行修正。
806	必须根据 Temp_Inlet 修正系数对 SpeedGradient 基准进行修正。
807	通讯模式 1 或通讯模式 3 时,必须[SHALL]对 Pressure_Inlet 信号进行极值故障诊断。极值故障诊断方法见信号极值故障诊断方法。
808	必须[SHALL]采用通讯的总线连续信号的表决要求进行 Pressure_Inlet 信号的表决。表决方法见信号表决方法章节。
809	必须[SHALL]按“信号处理概述”的要求判断双通道 Pressure_Inlet 故障。

表 2 SFC 示例需求分类

Table 2 SFC sample requirement classification

编号	需求类型
768	条件型需求
770	事件性需求
779	事件性需求
771	模式转换型需求
781	条件型需求
783	条件型需求
789	条件型需求
791	条件型需求
793	条件型需求
797	模式转换型需求
800	条件型需求
802	条件型需求
804	条件型需求
806	条件型需求
807	事件性需求
808	事件性需求
809	事件性需求

本阶段通过需求解析提取离散模式实例,并建立模式集合。需指出的是,此时因未完成变量抽取与约束推导,Conversions 中模式迁移规则暂未实例化。

以 SFC 系统的供油控制需求(需求 779/771/797)为例,构建模式集 mFuelControl,其包含 3 个基础工作模式:预压调节(m_1)、燃料主路充填(m_2)和燃烧初始化供油(m_3)。通过需求文本解析,明确各模式的定义边界,但模式间迁移规则(如 $m_2 \rightarrow m_3$ 的触发条件)尚未实例化,需通过后续变量关联分析完成动态行为映射。

根据原始需求文档的特征,将需求条目里的建模元素继续划分为输入变量、中间变量与输出变量 3 类。其中输入变量通过提取需求前提中独立存在且未被其他需求引用的参数确定,例如需求 768~781 中的 Speed_Rate 参数,用于表征系统外部输入接口;输出变量依据跨系统交互特征定义,如需求 800 涉及的 Speed_Gradient 参数,反映系统对外部组件的控制指令;中间变量则服务于复杂逻辑解耦,例如需求 797 通过定义 tDelta_Exhaust = TExhaust-TExhaustStart 简化温升阈值条件。最终所构建的 SFC 示例的变量字典与数据类型映射关系如表 3 所示。

在完成变量抽取的基础上,对预定义模式集的迁移规则进行动态回填。以 mFuelControl 模式集为例:

表 3 SFC 系统领域概念库建模元素

Table 3 Modeling elements of the SFC system domain concept library

Model 标识	物理名称	类别	数据类型
Speed_Threshold	Speed_Threshold 信号	输入	Float
Temp_Inlet	Temp_Inlet 信号有效值	输入	Float
Pressure_Inlet	Pressure_Inlet 信号有效值	输入	Float
tExhaust	tExhaust 信号有效值	输入	Float
tExhaustStart	起动时刻 tExhaust	输入	Float
Temp_InletStart	起动时刻 Temp_Inlet	输入	Float
Pressure_InletStart	起动时刻 Pressure_Inlet	输入	Float
FuelFlow_Setpoint	起动燃油流量给定值	输出	Float
SpeedGradient	起动 SpeedGradient 值	输出	Float
tDiff_Exhaust	tExhaust 当前与初始的差	中间	Float
tFuelFlow_P1	FuelFlow_SetpointKt 参数	中间	Float
tFuelFlow_P2	FuelFlow_SetpointKp 参数	中间	Float

m_1 (预压调节) $\rightarrow m_2$ (燃料主路充填)的迁移由需求 779 触发,转换条件回填至输入变量 ipSpeed_Rate $\geq 20\%$;

$m_2 \rightarrow m_3$ (燃烧初始化供油)的迁移规则源于需求 771,通过逻辑析取表达式 $(ipSpeed_Rate \geq 20 \wedge ipTimerSpeed \geq 1) \vee (ipSpeed_Rate \geq 25)$ 实现,其中 ipTimerSpeed_Rate 代表了 $Speed_Rate \geq 20\%$ 的持续时间。

3.2 需求的规范化处理

将上述所得到的自然语言条目需求继续进行 VRM 模型的形式化和规范化处理。在这个过程中,主要考虑条件型映射(F_C)与事件型映射(F_E)两类情况。

(1)条件型需求的规范化模板定义为

$$F_C:\langle condition \rangle \Rightarrow \langle subject \rangle := \langle value \rangle \quad (2)$$

式中:subject 为受控变量主体,value 为基于变量值域的目标赋值,condition 为触发赋值的先决条件。

以需求 768 为例,其自然语言描述“Speed_Rate 未达到 20% 前,必须将油针关闭在最小位置且接通停车电磁阀”被解构为两条独立的条件映射规则,规范化结果如表 4 所示。

(2)事件型需求的规范化模板定义为

$$F_E:\langle event \rangle \Rightarrow \langle subject \rangle := \langle value \rangle \quad (3)$$

表 4 需求 768 规范化结果

Table 4 Standardization results of requirement 768

Condition	Subject	Value
ipSpeed_Threshold < 20	opFuelFlow_Setpoint	0
	opCLM_bStopValueCmd	True

式中 event 采用 Event ∧ Guard 形式描述状态迁移过程:

@T: 状态由假变真 (prev (cond) =False ∧ cond=True);
@F: 状态由真变假 (prev (cond) =Ture ∧ cond=False);
@C: 状态发生改变 (prev (cond) ≠ cond)。
以需求 770 为例,其描述“Speed_Threshold ≥ 20% 时断开停车电磁阀”的事件逻辑解析为
@T (ipSpeed_Rate ≥ 20) ≡ prev (ipSpeed_Rate <

20) ∧ (ipSpeed_Rate ≥ 20)

需求 770 规范化处理结果如表 5 所示。基于条件映射与事件映射规则,对 SFC 系统的核心需求进行规范化表达,生成 VRM 规范化模型。表 6 给出了部分结果。

表 5 需求 770 规范化处理结果

Table 5 Standardization results of requirement 770		
Event	Subject	Value
@T (ipSpeed_Threshold ≥ 20)	opCLM_bStopValue Cmd	False

表 6 SFC 系统需求规范化处理部分结果

Table 6 Partial results of standardization processing for SFC system requirements	
编号	形式化语言描述
791	当满足以下条件,tMid_FuelFlow_P2 应能够设置为 Expression(A)
793	当满足以下条件,tCompensation_Factor 应能够设置为 Expression(B)
797	当满足以下条件,tCompensation_Factor 应能够设置为表达式 Expression(C)
	当满足以下条件,tCompensation_Factor 应能够设置为表达式 Expression(D)
	当满足以下条件,tFactor 应能够设置为 Expression(E)
800	当满足以下条件,tMid_StartSpeedGradientDem 应能够设置表达式 Expression(F)
802	当满足以下条件,tMid_StartSpeedGradientDem 应能够设置为 Expression(G)
804	当满足以下条件,tMid_FuelFlow_P2 应能够设置为 Expression(H)
806	当满足以下条件,tMid_SpeedGradientDemKt 应能够设置为 Expression(I)

3.3 SFC 需求模型的形式化分析

在前面工作的基础上,基于 ART 平台对所建立的需求形式化 VRM 模型进行分析,对 SFC 示例模型进行多范式的模型分析^[15,23]。第 1 轮次分析的结果显示共存在 31 处需求缺陷,其中一部分结果如表 7 所示。

对上述分析的初步结果进一步归类总结,其统计结果如表 8 所示。在获取到这些形式化分析的

表 7 SFC 系统建模分析部分错误类型

Table 7 Parts of error types in SFC system modeling and analysis	
序号	错误类型
错误 1	错误类型:第一范式检查,违反输入完整性
错误 2	错误类型:第四范式检查,缺失输出值
错误 3	错误类型:第二范式检查,违反条件完整性
错误 4	错误类型:第四范式检查,缺失输出值
错误 5	错误类型:第四范式检查,输出值有误
错误 6	错误类型:第二范式检查,违反条件完整性
错误 7	错误类型:第四范式检查,缺失输出值
错误 8	错误类型:第四范式检查,输出值有误
错误 9	错误类型:第二范式检查,违反条件完整性
错误 10	错误类型:第四范式检查,缺失输出值
错误 11	错误类型:第四范式检查,输出值有误
错误 12	错误类型:第二范式检查,违反条件完整性

表 8 SFC 需求模型分析结果统计

Table 8 Statistical analysis results of the SFC requirement model	
统计项目	数量
基本范式错误	0
输入完整性错误(第一范式)	1
条件一致性错误(第二范式)	10
事件一致性错误(第三范式)	0
输出完整性错误(第四范式)	20

错误信息之后,继续根据领域工程背景和条目化需求描述的语义,与工程人员进一步确认这些错误的真正含义,并对错误原因进行分析,修改和调整原始需求条目的描述内容。表 9 和 10 分别给出了错误 12 和错误 28 的反馈分析结果。

表 9 错误 12 详细信息

Table 9 Detailed information of error 12			
错误类型:第二范式检查,违反条件完整性			
错误定位:表格 tMid_ke			
错误内容:当变量取值为下表任意一行的组合时输出变量无值			
tExhaustStart	Pressure_Inlet	Temp_Exhaust Mode	
(-2 000,13)	(-2 000,0.08)	(-2 000,13)	M1
(-2 000,13)	(-2 000,0.08)	(13,2 000)	M2
(-2 000,13)	0.08	(13,2 000)	M4

表 10 错误 28 详细信息

Table 10 Detailed information of error 28

错误类型:第四范式检查,输出完整性错误

错误定位:表格 opSpeedGradientDem

错误内容:该表格变量值域中的值 500、2 000 未在表中任何一行取得

表 9 指出 tMid_ke 未覆盖所有的输出情况,需求 793 仅提出了在 Exhaust_Start ≥ 13 且 Pressure_Inlet ≥ 0.08 时 tMid_ke 的取值情况,而忽略了其他输入情况时, tMid_ke 应该如何取值,因此不满足条件完整性。根据条件完整性的要求,该需求应当做如下修改:

Temp_ExhaustStart < 13 °C, tMid_ke = A

Temp_ExhaustStart ≥ 13 °C 且 Pressure_Inlet < 0.08 , tMid_ke = B

Temp_ExhaustStart ≥ 13 °C 且 Pressure_Inlet ≥ 0.08 , tMid_ke = C

此时可覆盖 tMid_ke 的所有输入组合情况,可满足条件完整性。

表 10 指出输出变量 opSpeedGradientDem 的取值 [500, 2 000] 无法在现有的输入组合中取到,追溯到需求中,该变量的取值方式如下

[opSpeedGradientDem] = Mid_ke $\times$

Int_FuelFlow_P1 $\times$ Int_FuelFlow_P2 $\times$
[Pressure_Inlet] / 0.101 3 (4)

该变量根据表达式取值,在所有的输入组合中均无法取到该变量的所有输出值,因此无法满足输出完整性要求。修改方式如下:根据需求修改该变量的值域,使其所有的数值都可以取到。经过上述错误反馈分析以及需求条目修改调整的过程,得到了一个修改版本的条目化需求文档。基于这个新版本的需求内容,展开后续轮次的需求模型形式化分析,直至在形式化分析结果中不再包括错误信息。

总体来看,通过对 SFC 软件需求进行形式化建模与分析,可将发现的需求缺陷归为以下两类:

(1) 需求完整性缺陷。主要表现为变量状态覆盖不全与边界条件缺失。例如:错误 12 中,原始需求未描述 tMid_ke 在其他输入组合下应当如何取值;错误 28 中,原始需求未覆盖输出变量所有的输出值。

(2) 需求一致性缺陷。由于原始需求文档对需求主体的描述过于分散,且数量庞大,可能会出现前后矛盾的问题。如错误 5 中,有两条冲突的需求,在同样的条件下对输出变量 opSpeedGradient-

Dem 定义了不同的输出,因此要对原始需求进行完善,从而满足需求一致性要求。

4 FADEC-SFC 需求的测试用例自动生成

在本文所建立的 FADEC-SFC 需求形式化模型基础上,展开基于模型的测试用例自动化生成。下面以 SFC 中的输出变量 opFuelFlow_Setpoint 的处理逻辑需求为例,给出基于机载安全关键软件 MC/DC(修正判定/条件覆盖)准则的测试用例过程的说明。表 11 所示为 opFuelFlow_Setpoint 以条目化需求描述的处理逻辑。

表 11 opFuelFlow_Setpoint 计算流程

Table 11 Calculation process of opFuelFlow_Setpoint

FuelFlow_Setpoint 计算流程

(1) [Speed_Threshold] $\geq 35\%$ $\parallel$ ([tExhaust] - [tExhaustStart]) ≥ 50 °C, 则[状态 1_起动开环供油][无效], [状态 1_起动闭环供油][有效]

(2) [状态 1_起动开环供油][有效]时,且[tExhaustStart] < 13 °C, 则[FuelFlow_Setpoint] = Expression(A)

(3) [状态 1_起动开环供油][有效]时,且[tExhaustStart] ≥ 13 °C && [Pressure_InletStart] < 0.08 MPa, 则[FuelFlow_Setpoint] = Expression(B)

(4) [状态 1_起动开环供油][有效]时,且[tExhaustStart] ≥ 13 °C && [Pressure_InletStart] ≥ 0.08 MPa, 则[Mid_ke] = Expression(C)

对需求中所涉及的各类变量类型进行提取和归类,按第 3 节流程对需求进行预处理后,该需求的输入变量如表 12 所示。

表 12 opFuelFlow_Setpoint 需求输入变量

Table 12 Input variables for opFuelFlow_Setpoint requirements

变量名	类型	值域	描述
Speed_Threshold	Float	[0, 120]	发动机转速
tExhaustStart	Float	[0, 1 000]	tExhaust 起始温度
Pressure_InletStart	Float	[0, 1 000]	Pressure_Inlet 起始压强
Temp_Inlet	Float	[0, 1 000]	Temp_Inlet 温度
Pressure_Inlet	Float	[0, 1 000]	Pressure_Inlet 压强
tExhaust	Float	[0, 1 000]	tExhaust 温度

表 12 中 6 个外部输入变量通过双重作用机制影响输出变量 opFuelFlow_Setpoint 的计算逻辑,例如 Speed_Threshold、tExhaustStart、Pressure_InletStart 影响 opFuelFlow_Setpoint 的计算应该采用哪个表达式,而 Temp_Inlet、Pressure_Inlet 影响 opFuelFlow_Setpoint 表达式中的具体数值。

如表 13 所示,根据需求特征设计 4 个中间变量,其取值也来自系统的输入变量。为了防止混乱,本文不再展开这些中间变量的计算逻辑。

表 13 opFuelFlow_Setpoint 需求中间变量
Table 13 Intermediate variables for opFuelFlow_Setpoint requirements

变量名	类型	值域	描述
StartWfm0	Float	[0,120]	Wfm0Dem 基准值
FuelFlow_P1	Float	[0,1 000]	根据 Temp_Inlet 得到
OpenWfmKp1	Float	[0,1 000]	根据 Pressure_Inlet 得到
Midke	Float	[0,1 000]	根据 tExhaustStart 得到
OpenFlag	Bool	{0,1}	起动开环供油

基于需求文档解析生成的 opFuelFlow_Setpoint 条件映射规则如表 14 所示,其通过逻辑分层结构将自然语言需求转化为形式化约束表达式。为简化例子复杂度,采用符号 A、B、C 代替原有表达式:A 代表 FuelFlowInit(A),B 代表 k1*Pressure_InletStart+k2(B),C 代表 FuelFlow_Nominal(C)。

表 14 opFuelFlow_Setpoint 取值流程表
Table 14 Value process table for opFuelFlow_Setpoint

条件	输出取值
OpenFlag=1 and tStart<13	A
OpenFlag=1 and tStart≥13 and Pressure_InletStart<0.08	B
OpenFlag=1 and tStart≥13 and Pressure_InletStart≥0.08	C
OpenFlag=0	0

为了生成测试路径集,本文通过解析条件表,构建了 opFuelFlow_Setpoint 信号的需求语义树,如图 5 所示,其判定逻辑包含两个核心维度:

(1) 状态标识判定模块。通过 Speed_Rate、Temp_Exhaust 与 Temp_ExhaustStart 参数确定 OpenFlag 状态标识的布尔取值,建立设备启闭状态判定规则。

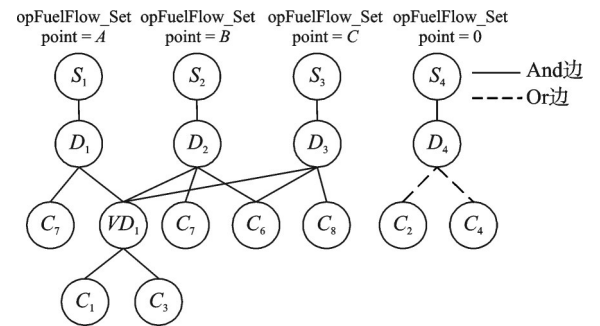


图 5 opFuelFlow_Setpoint 需求语义树
Fig.5 opFuelFlow_Setpoint requirement semantic tree

(2) 输出信号生成模块。在此基础上,结合 Temp_ExhaustStart 与 Pressure_InletStart 参数推导出 opFuelFlow_Setpoint 输出变量的逻辑表达式,形成信号生成机制。

根据需求语义树节点定义规范,各节点类型的功能说明如表 15 所示。

表 15 opFuelFlow_Setpoint 需求语义树变量表
Table 15 Semantic tree variable table for opFuelFlow_Setpoint requirements

变量名	含义
C ₁	Speed_Threshold<35
C ₂	Speed_Threshold≥35
C ₃	tExhaust-tExhaustStart<50
C ₄	tExhaust-tExhaustStart≥50
C ₅	tExhaustStart<13
C ₆	tExhaustStart≥13
C ₇	Pressure_InletStart<0.08
C ₈	Pressure_InletStart≥0.08
VD ₁	OpenFlag=1
D ₁	OpenFlag = 1 and tExhaustStart<13
D ₂	OpenFlag= 1 and tExhaustStart≥13 and Pressure_InletStart<0.08
D ₃	OpenFlag= 1 and tExhaustStart≥13 and Pressure_InletStart≥0.08
D ₄	OpenFlag = 0
S ₁	opFuelFlow_Setpoint = A
S ₂	opFuelFlow_Setpoint = B
S ₃	opFuelFlow_Setpoint = C
S ₄	opFuelFlow_Setpoint = 0

(1) 语句节点(S):对应需求中的结果声明部分,如 S₁ 节点描述输出表达式 opFuelFlow_Setpoint=A;

(2) 判定节点(D):表征复合逻辑条件,如 D₁ 节点定义联合条件 OpenFlag=1 and Temp_ExhaustStart<13;

(3) 判定子节点(VD):解析中间变量状态,如 VD₁ 节点判定 OpenFlag = 1;

(4) 条件节点(C):描述不可拆分原子条件,如 C₁ 节点对应阈值约束 Speed_Rate ≥ 35。

基于需求语义树结构,通过遍历算法可生成满足 MC/DC 准则的测试用例覆盖路径集合。遍历过程中由根节点至叶节点生成全条件组合路径(如 D₁→VD₁→C₁→S₁),其路径集完整覆盖各原子条件节点(C类节点)的真/假独立作用场景,如表 16 所示。其中:X₁ 为 Speed_Threshold,X₂ 为 tExhaustStart,X₃ 为 tExhaust,X₄ 为 |tExhaustStart—

tExhaust|, X_5 为 Pressure_InletStart, X_6 为 Open-Flag, X_7 为 opFuelFlow_Setpoint。

表 16 opFuelFlow_Setpoint 变量测试用例集

Table 16 Test case set for opFuelFlow_Setpoint variables

序号	输入变量					中间变量	输出变量
	X_1	X_2	X_3	X_4	X_5	X_6	X_7
1	35	25	12	13	0.01	0	0
2	32	12	15	3	0.01	1	A
3	30	13	10	3	0.01	1	B
4	28	20	28	8	0.08	1	C
5	26	25	80	55	0.01	0	0

本文工作基于模型驱动的自动化测试框架实现了上述测试用例生成流程,如图 6 所示。

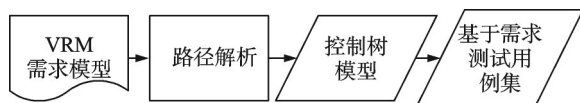


图 6 VRM 模型测试用例自动生成流程

Fig.6 Automated generation process for VRM model test case

自动生成流程可分为 3 个阶段:(1)通过 ART 平台将 VRM 需求模型转换为标准化 XML 中间文件,构建机器可解析输入;测试用例生成模块解析 XML 模型并执行自动生成算法,例如针对 op-SpeedGradient 信号,依据输入参数 ipSpeed_Rate、ipTemp_Inlet 与 ipT4550 的等价类划分生成 7 组独

立测试用例,每组用例对应特定输入组合的预期输出验证规则;最后系统输出 SFC 控制模块全量测试用例共 166 条,覆盖了中间变量和输出变量的所有测试路径,如表 17 所示。

表 17 测试用例生成条目

Table 17 Generation entries for test case

类型	变量名称	测试用例数
中间变量	tFuelFlow_P1	23
	tFuelFlow_P2	11
	tStartup_FuelFlow_Setpoint	44
	tMid_ke	23
	tMid_Startup_Gradient	11
	tMid_SpeedGradientP1	11
输出变量	opWfmDem	24
	opFuelFlow_Setpoint	4
	StartFuelControl	15

图 7 给出了单条用例的结构描述,用例格式针对自动化执行规程而设计,主要包含 3 部分,分别是测试用例信息描述、设置初值和验证输出。信息描述介绍该用例的 id 与测试变量;设置初值代表是为了初始化变量,防止引用时变量为空;最后验证方法使用关键词 Verify,由测试脚本判断最终答案。这个例子展示了中间变量 tFuelFlow_P1 根据输入变量 ipTemp_Inlet 的取值,验证该变量的输出是否为 $0.33 \times \text{ipTemp} - 4.32$ 。未来的工作中,可以使用测试脚本自动读取测试用例进行测试,完成测试用例的执行工作。

Start of test case	42 985
Start of test prompt testing:	
Requirements tested:	969.672 1.tFuelFlow_P1: $0.33 \times \text{ipTemp_Inlet} - 4.32$
Action:	
End of test prompt	
Start of test script	
1 Comment_python	
2 Set	tFuelFlow_P1_OUT 0
3 Set	ipTemp_Inlet 399.99
4 Verify	tFuelFlowP1_OUT $=0.33 \times \text{ipTemp} - 4.32$
End of test script	
End of test case	

图 7 SFC 系统测试用例

Fig.7 SFC system test cases

5 对 FADEC-SFC 实例研究的其他分析

上文给出了本研究面向 SFC 控制系统需求所展开的全流程形式化建模、分析与测试用例生成的完整实例流程。在此过程中,本文还对过程中

所遇到的一些其他问题进行考虑和分析,限于篇幅,简要描述如下。

(1)针对双通道架构的冗余信号容错机制,本文基于需求规范提出一种异构信号重构算法,如算法 1 所示。当通道 1 的 N1 信号失效时,系统通过通道 2 的 N2 与 T2 参数跨通道信号融合,实现失效

信号的动态重构与状态恢复。

算法 1 N1 信号重构方法

输入: N2, T2

输出: N1

(1) if T2_Dual_Fault = false:

(2) N1 = Reconstruction (T2)

(3) else if T2_Dual_Fault = true:

(4) if (T1_other_Communication_Fault = false)

N1=Reconstruction (Reconstruction (T1_other))

(5) else if H_SL_Dual_Fault = false&Ma_Dual_Fault = false:

N1 =Reconstruction(Reconstruction (H_SL, Ma))

(6) else if P0_Dual_Fault = false && Ma_Dual_Fault = false

N1 =Reconstruction((Reconstruction(P0), Ma)

(7)else:

N1=N1_Pre

根据第 3 节的需求规范化转换流程,结果如表 18 所示。

表 18 N1 信号重构需求规范化表示

Table 18 Standardized representation of N1 signal reconstruction requirements

N1 信号重构方法规范化表示
(1)N1 = reconstruction (T2) 当满足以下条件:tv_T2 failure = False;
(2)N1 = reconstruction (reconstruction (T1_other)) 当满足以下条件: tv_T2 failure = False and tv_T1_other_communication_failure = False;
(3)N1 = reconstruction (reconstruction (H_SL, Ma)) 当满足以下条件: tv_T2 failure = False and tv_T1_other_communication_failure = True;
(4)N1 = reconstruction (reconstruction (P0, Ma)) 当满足以下条件: tv_T2 failure = False and tv_P0_communication_failure = False;
(5)N1 = pre_N1 当满足以下条件: tv_T2 failure = False and tv_P0_communication_failure = True。

(2)双通道冗余信号输入至控制模块时需执行多源异构信号实时仲裁机制。如表 19 所示,系统支持 3 类表决方法:基于通道自诊断的状态表决、跨通道输出结果对比表决以及混合式动态表决架构。

本文采用混合式动态表决架构,基于通道健康

表 19 信号表决方式	
Table 19 Signal voting method	
表决方式	表决特点
基于诊断的通道表决	当任一通道检测到自身故障时,表决逻辑会立即切换到另一正常通道继续运行。
双通道结果对比表决	两个通道实时输出信号通过比较器进行同步比对,若结果一致则输出有效信号,若出现差异则触发诊断流程。
混合式表决架构	根据通道健康状态动态调整表决权重: ①正常状态:双通道等权表决; ②单通道诊断异常:切换为单通道主导模式。

状态而进行动态权重分配策略,如表 19 所示。该架构通过多模态切换逻辑实现信号仲裁优化:双通道有效时执行等权重仲裁;单通道诊断异常时切换至健康通道主导模式;双通道同步失效时激活末态保持机制。如算法 2 所示,表决系统依据实时诊断状态自适应调整仲裁模式,并通过阈值约束机制确保权重切换过程满足航空控制系统的时序确定性要求。

算法 2 双通道表决方法

输入: S1, S2, pre_Vote

输出: Vote

If (S1_flag = true and S2_flag = true):

If(abs(S1-S2)> Speed_Threshold value):Vote = S1

Else Vote = average(S1, S2)

Else if(S1_flag = false and S2_flag = true):Vote = S2;

Else if(S2_flag = false and S1_flag = true):Vote = S1;

Else: Vote= pre_Vote

(3)FADEC 系统的需求与传统布尔逻辑控制系统的需求在形式上存在一些差异,其中一个特征多参数耦合的连续计算机制。以输出变量 FuelFlow_Setpoint 为例,该参数的计算过程包含多阶段动态修正,如式(5)所示。首先基于 Spd_Rate、Temp_Inlet、Pressure_Inlet、SpeedGradient 参数计算基准流量 Mid_StartFuelFlow_Setpoint; 其次根据 Temp_Inlet 与 Pressure_Inle 推导温度修正系数 K_t 与压力修正系数 K_p ;最终通过非线性函数合成得到 FuelFlow_Setpoint 控制量。此类复合计算模式体现了航发控制系统参数强耦合、计算强实时的特性。

$$[\text{SetPoint}] = K_e \times \text{StartSetpoint} \times K_t \times K_p \times \frac{[\text{Pressure}]}{0.1013} \times \text{Sqrt}\left(\frac{288.15}{[\text{Temp}] + 273.15}\right) \quad (5)$$

式中: SetPoint 为输出变量, K_e 为温度修正参数,

StartSetpoint 为输出变量初始值, K_t 为温度修正系数, K_p 为压强修正系数; Pressure 为部件压强, Temp 为部件温度。

(4) 针对浮点型变量的形式化建模与分析, 本文采用基于离散化状态转换的需求分析范式^[26]。相较于布尔型变量的二值空间, 浮点型变量的连续值域会导致状态空间爆炸问题, 需解决值域完整性覆盖与多变量输入组合合法性两类问题。本文采用了离散枚举映射方法, 将浮点输出变量离散化为

Enum 型变量, 其枚举值数量与需求文档中赋值表达式种类严格对应; 当输入参数满足特定约束时, 输出变量被唯一映射至对应枚举值。

(5) FADEC 系统的燃油控制逻辑中存在部分基于插值计算的核心需求, 如表 20 所示。此类需求针对多维参数空间(如转速 Speed_Rate、进气温度 Temp_Inlet 和压力 Pressure_Inlet)定义离散化的基准流量映射关系, 并通过线性插值策略实现连续工况点的动态修正计算。

表 20 插值表计算相关需求

Table 20 Requirements related to interpolation table calculation

序号	需求内容
781	必须根据 Speed_Threshold 转速进行开环基准流量的计算。
789	必须根据 Temp 修正系数对基准流量进行修正。修正系数根据 Temp 线性插值。
791	必须根据 Pressure_Inlet 修正系数对基准流量进行修正。修正系数根据 Pressure_Inlet 线性插值。
781	必须根据 Speed_Threshold 转速进行开环基准流量的计算。

以燃油温度修正系数 K_t 计算需求(需求 789)为例, 表 21 定义了进气温度 Temp_Inlet 与修正系数的离散映射关系, 本文对比分析了两种插值方法。

表 21 Temp_Inlet- K_t 插值表

Table 21 Temp_Inlet- K_t interpolation table

Temp_Inlet	200	220	240	260	300	320	340	370	400
K_t	1	2	3	4	5	6	7	8	9

(1) 拉格朗日多项式插值法。通过构建经过所有离散点(如表 21 中 Temp_Inlet- K_t 标定点)的高阶连续多项式实现插值计算, 其数学表达式如式(6)所示。对某个多项式函数, 已知给定的 $k+1$ 个取值点, 有

$$L(x) = \sum_{j=0}^k y_j l_j(x) \quad (6)$$

式中: $L(x)$ 为插值多项式, k 代表已知数据点的最大索引; j 代表循环变量, y_j 代表第 j 个已知数据点的函数值; $l_j(x)$ 代表拉格朗日基函数。

该方法虽能严格遍历所有标定点, 但随标定点数量增加将生成高次多项式, 导致计算复杂度呈指数级增长, 难以满足航发控制系统的实时性要求。

(2) 分段线性插值法。采用相邻标定点间的线性关系进行插值计算, 其表达式如下

$$y = y_0 + \frac{(x - x_0) \cdot (y_1 - y_0)}{x_1 - x_0} \quad (7)$$

式中: x 和 y 为待求的变量, 其中 x 为给定的自变量(位于 x_0 和 x_1 之间), y 为需要计算的插值结果; x_0

和 y_0 为第 1 个已知点的横、纵坐标, 代表计算的起始基准; x_1 和 y_1 为第 2 个已知点的横、纵坐标。

该方法计算效率较拉格朗日插值法更高, 且插值结果与工程实测数据的平均误差符合工程实际需求。经与控制系统工程师协同验证, 最终选定线性插值作为以上插值表的计算方法。

6 结 论

本文面向大飞机 FADEC 软件系统的测试验证问题, 采用变量关系模型对一个典型的 FADEC-SFC 系统软件需求实例进行形式化建模, 提出了一个对 FADEC 软件需求从形式化建模、分析到基于模型测试用例生成的技术框架; 展开了实例分析研究, 并且总结分析了 FADEC 软件需求进行建模与测试的工程经验。

下一步研究工作将围绕以下维度展开: 增强 VRM 建模语言对时序属性与复杂表达式的原生支持, 拓展时间自动机理论在模型分析中的应用; 开发分层建模机制以支持更大规模需求规模系统的形式化建模与分析; 完善测试脚本生成框架, 实现测试驱动文件的全自动构建流程, 以形成持续迭代工程实用方法, 推动形式化方法在 FAEDC 系统领域中需求建模分析等的深入应用。

参考文献:

- [1] GARG S. Aircraft turbine engine control research at NASA glenn research center[J]. Journal of Aerospace Engineering, 2013, 26(2): 422-438.
- [2] LIU J J, ZHANG Q, YU J N. Comparative statistics

- and analysis of accidents of two main types of commercial transport aviation in recent 30 years[J]. *Safety & Security*, 2022, 43(10): 21-28.
- [3] International Civil Aviation Organization (ICAO). Global aviation safety plan 2020—2022: ICAO[R]. Montreal, Canada: [s.n.], 2020.
- [4] RTCA, EUROCAE. Software considerations in airborne systems and equipment certification: DO-178B/ED-12B[R]. Washington, DC: RTCA Inc., 1992.
- [5] RTCA. Software considerations in airborne systems and equipment certification: DO 178C[R]. Washington D C: Radio Technical Commission for Aeronautics, Inc., 2011.
- [6] LEMPIA D L, MILLER S P. Requirements engineering management findings report: DOT/FAA/AR-08/34[R]. Washington D C, USA: Department of Transportation, Federal Aviation Administration, 2008.
- [7] LEMPIA D L, MILLER S P. Requirements engineering management handbook: DOT/FAA/AR-08/32[R]. Washington DC, USA: Department of Transportation, Federal Aviation Administration, 2009.
- [8] Federal Aviation Administration. Advanced avionics handbook[M]. New York: Skyhorse Publishing, 2011.
- [9] COFER D D, MILLER S P. DO-333 certification case studies[C]//Proceedings of NASA Formal Methods Symposium. Berlin, Heidelberg: Springer, 2014.
- [10] COOK S C, PRATT J M. Advances in systems of systems engineering foundations and methodologies [J]. *Australian Journal of Multi-Disciplinary Engineering*, 2020, 17(1): 9-22.
- [11] 胡晓义,王如平,王鑫,等.基于模型的复杂系统安全性和可靠性分析技术发展综述[J]. *航空学报*, 2020, 41(6): 523436.
- HU Xiaoyi, WANG Ruping, WANG Xin, et al. Recent development of safety and reliability analysis technology for model-based complex system [J]. *Acta Aeronautica et Astronautica Sinica*, 2020, 41(6): 523436.
- [12] KOHEN H, WENGROWICZ W, DORI D. Simulating real-life practice as a software product owner for students in a model-based systems engineering MOOC [C]//Proceedings of the 8th Kinneret Conference on Software Engineering Education.[S.l.]:[s.n.], 2020.
- [13] Radio Technical Commission for Aeronautics. DO-331 model-based development and verification supplement to DO-178C and DO-278A[M]. [S. l.]: RTCA Press, 2011.
- [14] HUI J. Research on RTCA/DO-331 standard[J]. *Civil Aircraft Design & Research*, 2018, 130(3): 124-129.
- [15] 石梦焯. 基于形式化模型的系统需求与设计安全性分析方法研究[D]. 南京: 南京航空航天大学, 2020.
- SHI Mengye. Research on system requirements and design safety analysis method based on formal model [D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2020.
- [16] 胡建成,胡军,汪文轩,等.一种面向领域自然语言需求的形式化需求模型生成方法研究[J]. *小型微型计算机系统*, 2021, 42(8): 1639-1648.
- HU Jiancheng, HU Jun, WANG Wenxuan, et al. Constructing formal specification models from domain specific natural language requirements[J]. *Journal of Chinese Computer Systems*, 2021, 42(8): 1639-1648.
- [17] WANG W X, HU J, HU J C, et al. Automatic test case generation from formal requirement model for avionics software[C]//Proceedings of the 6th Symposium on System and Software Reliability (ISSSR). Chengdu, China: IEEE, 2020: 12-20.
- [18] 高亚辉,倪焯斌,姜成平,等.航空发动机控制系统及关键技术现状与展望[J]. *南京航空航天大学学报*, 2024, 56(4): 577-596.
- GAO Yahui, NI Yebin, JIANG Chengping, et al. Research status and prospect of aeroengine control systems and key technologies[J]. *Journal of Nanjing University of Aeronautics & Astronautics*, 2024, 56(4): 577-596.
- [19] 廖忠权. 罗罗的飞行电气化之路[J]. *航空动力*, 2020(1): 16-19.
- LIAO Zhongquan. The flight electrification path of Rolls-Royce[J]. *Journal of Aerospace Power*, 2020(1): 16-19.
- [20] 廖忠权. 2022世界电动系统进展[J]. *航空动力*, 2023(1): 23-26.
- LIAO Zhongquan. Electric power system progress in 2022 [J]. *Journal of Aerospace Power*, 2023(1): 23-26.
- [21] PARNAS D L. Software aspects of strategic defense systems[J]. *Communications of the ACM*, 1985, 28(12): 1326-1335.
- [22] LEVESON N G, HEIMDAHL M P E, HILDRETH H, et al. Requirements specification for process-control systems[J]. *IEEE Transactions on Software Engineering*, 1994, 20(9): 684-707.
- [23] 王康星,胡军,王立松,等.机载软件层次化需求的形式化建模与分析[J]. *南京航空航天大学学报(自然科学版)*, 2025, 57(1): 195-204.
- WANG Kangxing, HU Jun, WANG Lisong, et al. Formal modeling and analysis method for hierarchical requirements of airborne software[J]. *Journal of Nan-*

- jing University of Aeronautics & Astronautics (Natural Science Edition), 2025, 57(1): 195-204.
- [24] 张漾. 基于 VRM 模型的形式化验证与测试用例生成技术研究[D]. 南京: 南京航空航天大学, 2022.
- ZHANG Yang. Research on formal verification and test case generation technology based on variable relational model[D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2022.
- [25] 丁鼎, 胡军, 王康星, 等. 面向航空软件需求模型的 MC/DC 测试用例生成方法[J]. 小型微型计算机系统, 2025, 46(7): 1783-1792.
- DING Ding, HU Jun, WANG Kangxing, et al. Modified condition/decision coverage test case generation method for aviation software requirements model[J]. Journal of Chinese Computer Systems, 2025, 46(7): 1783-1792.
- [26] 吕佳润. VRM 需求模型的语义分析方法研究[D]. 南京: 南京航空航天大学, 2023.
- LV Jiarun. Research on semantic analysis method of VRM requirement model[D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2023.
- [27] HEITMEYER C L. Software cost reduction[EB/OL]. (2002-01-30). <https://onlinelibrary.wiley.com/doi/10.1002/0471028959.sof307/tables>.
- [28] HAGER J A. Software cost reduction methods in practice: A post-mortem analysis[J]. Journal of Systems and Software, 1991, 14(2): 67-77.
- [29] BLACKBURN M R, BUSSER R D. T-VEC: A tool for developing critical systems[C]//Proceedings of the 11th Annual Conference on Computer Assurance. Gaithersburg: IEEE, 1996: 237-249.
- [30] BOOCH G, RUMBAUGH J, JACOBSON I. The unified modeling language user guide[M]. Reading UK: Addison-Wesley, 1999.
- [31] FOWLER M. UML distilled: A brief guide to the standard object modeling language[M]. 3rd ed. Reading, UK: Addison-Wesley, 2003.
- [32] HAREL D. Statecharts: A visual formalism for complex systems[J]. Science of Computer Programming, 1987, 8(3): 231-274.
- [33] WOLNY S, MAZAK A, CARPELLA C, et al. Thirteen years of SysML: A systematic map study[J]. Software and Systems Modeling, 2020, 19: 111-169.
- [34] GERY E, HAREL D, PALACHI E. Rhapsody: A complete life-cycle model-based development system [C]//Proceedings of the 3rd Conference on Integrated Formal Methods. Turku: Springer, 2002: 1-10.
- [35] HAREL D, LACHOVER H, NAAMAD A, et al. Statemate: A working environment for the development of complex reactive systems[J]. IEEE Transactions on Software Engineering, 1990, 16(4): 403-414.
- [36] 王正林, 刘明. 精通 MATLAB[M]. 北京: 电子工业出版社, 2011.
- WANG Zhenglin, LIU Ming. Proficient in MATLAB [M]. Beijing: Publishing House of Electronics Industry, 2011.
- [37] KURIAN E, BRAIONE P, BRIOLA D, et al. Automated test case generation for safety-critical software in scade[C]//Proceedings of 2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). [S.l.]: IEEE, 2023: 483-494.
- [38] YANG Z B, HU K, ZHAO Y W, et al. Verification of AADL models with timed abstract state machines [J]. Journal of Software, 2015, 26(2): 202-222.
- [39] WANG F, YANG Z B, HUANG Z Q, et al. Approach for generating AADL model based on restricted natural language requirement template[J]. Journal of Software, 2018, 29(8): 2350-2370.
- [40] FEILER P H, GLUCH D P, HUDAK J J. The architecture analysis design language (AADL): An introduction[M]. Pittsburgh: Carnegie Mellon University, 2006.

(编辑: 刘彦东)