文章编号:1674-8190(2023)04-168-09

基于系统架构与运行框图的机载软件故障建模与分析应用

温晓玲,姜梦岑,艾笑天,柳庆宇
(航空工业沈阳飞机设计研究所 所办,沈阳 110035)

摘 要: 已有技术难以规范且有效地识别机载软件故障及其原因,无法解决机载软件研制过程中应用等问题,因此提出基于系统架构与运行框图的机载软件故障建模与分析技术。首先,基于功能失效分析的机载软件故障识别方法,从数据取值、时序约束等角度识别机载软件故障;然后,基于系统静态体系架构与动态运行框图,构建故障层次组成模型以及故障逻辑关系模型,支撑机载软件故障树的规范高效建立;其次,基于标准要求和软件需求,提出软件故障原因分析与安全性需求制定策略;最后,针对机载刹车控制软件开展工程应用。结果表明:本文提出的机载软件故障建模与分析技术符合标准要求、规范可行,能够形成机载软件故障分析验证的V&V闭环。

关键词: 机载软件;故障建模;故障原因分析;软件安全性;系统架构
中图分类号: V247; TP311.5 **文献标识码:** A
DOI: 10.16615/j.cnki.1674-8190.2023.04.18

Airborne software fault modeling and analysis application based on system architecture and operation diagram

WEN Xiaoling, JIANG Mengcen, AI Xiaotian, LIU Qingyu
(Institute Office, AVIC Shenyang Aircraft Design and Research Institute, Shenyang 110035, China)

Abstract: The existing technologies are difficult to identify the airborne software faults and the corresponding causes, and thus cannot be applied in the development process of the airborne software, so the airborne software fault modeling and analysis technique based on the system architecture and operation diagram are proposed. Firstly, the airborne software fault identification approach based on the function failure analysis is proposed, which can be used to identify the airborne software faults caused by the interface data, time constraint and so on. And, the fault hierarchy modeling approach based on the static system architecture and the fault logic relationship modeling approach based on the dynamic operation diagram are constructed respectively, which can be used for constructing the software fault tree effectively. Then, the software fault reason analysis and safety requirement development strategy based on the software requirement and criteria is proposed. Finally, the engineering application of the airborne brake control software is conducted. The results show that the proposed airborne fault modeling and analysis technique is feasible, and consistent with the standard requirement, which is suitable for constructing the V&V loop of the airborne software fault analysis and validation.

Key words: airborne software; fault modeling; fault reason analysis; software safety; system architecture

收稿日期: 2022-08-12; **修回日期:** 2022-11-25
基金项目: 航空工业联合基金“十三五”项目(6141B050301)
通信作者: 姜梦岑, jmc302@126.com
引用格式: 温晓玲, 姜梦岑, 艾笑天, 等. 基于系统架构与运行框图的机载软件故障建模与分析应用[J]. 航空工程进展, 2023, 14(4): 168-176.
WEN Xiaoling, JIANG Mengcen, AI Xiaotian, et al. Airborne software fault modeling and analysis application based on system architecture and operation diagram[J]. Advances in Aeronautical Science and Engineering, 2023, 14(4): 168-176. (in Chinese)

0　引　言

机载软件对系统任务完成与运行安全有着决定性的影响,一旦发生失效,轻则导致任务失败,重则导致飞行事故[1]。因此,机载软件通常具有较高安全性要求[2-3],国内外已经制定了一系列标准[4-5],用于保障机载软件安全性满足指标要求。

软件安全性的核心即是对软件故障的识别与控制。因此,故障建模与分析技术成为航空机载软件质量领域的研究热点,其中,基于故障树分析(FTA)的软件故障建模分析技术由于具有表达直观、逻辑清晰等特点,在国内外已经得到大量的研究和应用。张杰等[6]将FTA技术应用于航空发动机控制软件,支撑软件故障诊断与定位过程;檀德宾[7]将FTA技术应用于服务器集群公共安全系统,通过故障树节点实现故障的跟踪与修复;汪相国等[8]、张辉等[9]基于对嵌入式控制软件以及舰载指控系统软件的特点分析,支撑故障树建模过程;林红等[10]、王思琪等[11]借助Petri网、状态机等形式化方法来实现故障树分析以及最小割集的求解;石柱等[12]将FTA技术引入某型星载嵌入式软件研制过程;樊茜等[13]、张红林等[14]将优先与门、顺序门等动态逻辑门融入故障树分析;T. Balaje等[15]针对大型系统故障树,使用二元决策图进行最小割集计算;Y. Nataliya等[16]提出依据SysML模型进行故障树构建,并进行定量评估分析;S. Jung等[17]提出一种基于形式化需求的软件故障树分析技术;V. Philippov[18]则将故障树技术应用于空中交通管制系统软件的可靠性建模。

现有研究多集中于将故障树应用于特定软件的故障分析、诊断与定位等工作[6-9,12,18]以及最小割集计算[10,15]、形式化验证[10,11,16-17]、动态逻辑门[13-14]等理论研究,而对于如何依据机载软件需求,实现符合标准要求、规范客观的故障建模分析过程的研究相对较少。即已有工作多依赖于人的主观经验或者复杂的形式化方法来构建故障树,而并未阐述事件节点(顶事件、中间事件、底事件)和关系节点(与门、或门等)等故障树元素的识别过程与建模依据,存在着规范性较差、随意性强等不足,难以形成规范有效的软件故障建模分析能力,对于具有复杂运行特征的机载软件来说,很难准确且有效地构建机载软件的故障树模型,因而无法充分识别机载软件的故障模式及原因。

针对此问题,本文提出基于系统架构与运行框图的机载软件故障建模与分析技术。首先,依据静态系统架构和动态运行框图,构建机载软件故障分析模型;然后,在故障模型基础上,参考DO-178C、GJB/Z 102A等标准[2,4-5]以及历史数据经验,制定故障原因分析准则,分析导致故障的原因;其次,针对所识别的故障原因,参考GJB/Z 102A等标准,从“事前预防”或“事后控制”等角度,制定软件安全性需求获取策略;最后,在机载刹车控制软件上进行应用,验证本文技术的有效性和可行性。

1　机载软件系统故障建模分析框架

构建的机载软件故障建模分析框架如图1所示。

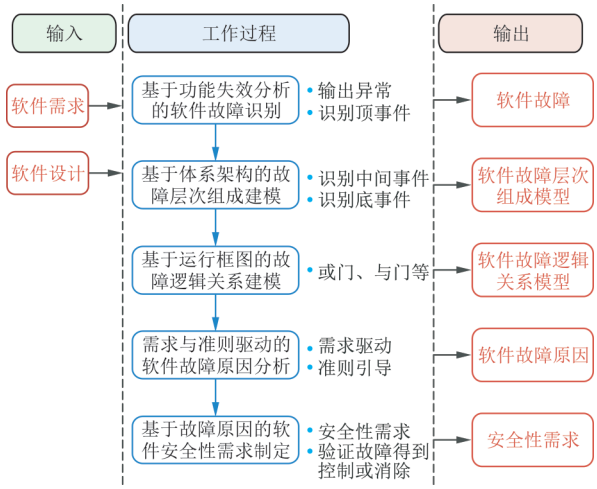

图1　机载软件系统故障建模分析框架

Fig. 1　The fault modeling analysis framework of the airborne software

基于系统架构与运行框图的机载软件故障建模与分析技术主要有以下5个步骤。

步骤1:基于功能失效的软件故障识别。通过功能输出接口,从数据、时序、通信等角度,识别功能失效状态及其影响,即识别的软件故障。

步骤2:基于静态体系架构的故障层次组成建模。依据软件体系架构中的组成项(包括子功能项、接口项等),自上而下确定可能导致软件故障的中间事件以及底事件。

步骤3:基于动态运行框图的故障逻辑关系建模。依据运行框图中的串并联关系,确定中间事件或底事件之间的逻辑关系,“AND”“OR”等。

步骤4:需求与准则驱动的软件故障原因分析。参考标准要求或准则规范,从数据取值、通信、时序、设备、功能逻辑等角度,识别导致“故障树模型底事件”的各种原因。

步骤5:基于故障原因的软件安全性需求获取。针对识别的故障原因,从“事前预防”和“事后控制”等角度,制定安全性需求,验证软件故障已经得到有效的控制或者缓解。

由上述5个步骤可知,基于系统架构与运行框图进行机载软件故障建模,是本文研究内容的基础。而基于机载软件故障模型,实现故障原因及对应安全性需求的分析,则是本文研究内容的目标。

2 基于静态架构与动态框图的机载软件故障建模

2.1 基于功能失效的机载软件故障识别方法

1) 软件工作阶段与功能识别

首先,依据软件需求,识别机载软件运行过程中的飞行阶段以及工作状态;然后,依据软件需求,从“黑盒角度”来确定功能清单。即功能输入数据要“从外部系统中来”,且功能输出数据要到“外部系统中去”。

2) 功能失效状态分析

功能失效状态表示软件功能输出的异常情况,可将其视为软件故障。本文从下述角度分析功能失效状态。

①数据取值:针对功能输出数据的取值情况、精度/分辨率等需求要素进行失效状态识别。例如,功能输出数据的取值大于值域上限等。

②时序约束:针对功能输出数据的周期、时刻、持续时间等需求要素进行失效状态识别。例如,功能持续N个周期,输出取值不变的数据等。

③通信过程:针对功能输出数据的通信数据帧、通信过程等需求要素进行失效状态识别。例如,功能输出的数据帧长度错误等。

④目的设备:针对功能输出数据对应的目的设备,从工作状态、存储容量等角度进行失效状态识别。例如,执行机构未响应控制指令等。

所识别的软件功能失效状态即为软件故障。

3) 功能失效状态影响分析

针对每项软件功能失效,可以从系统任务以及运行安全两个角度进行影响后果分析。同时,可参考ARP 4761、DO-178C、GJBZ 102A等标准,对机载软件功能失效影响后果进行等级评估(例如,DO-178C中将软件故障影响后果从高到低分为A、B、C、D四个等级)。

2.2 基于静态体系架构的机载软件故障层次关系建模方法

目前,已有的机载软件故障树等故障模型的建立方法研究,并未明确该如何依据软件系统需求来确定故障树的顶事件、中间事件以及底事件等组成项,安全性人员在构建故障树模型时,对于不同级别组成项的建模存在随意性、主观性、不一致性。针对此问题,本文借助系统静态体系架构这一客观对象,提出机载软件故障层次关系模型的建立方法。针对软件故障(FC),基于系统静态体系架构,自上而下明确与其相关的“功能→子功能(交联硬件)→模块/单元”及其之间的层次关系。在此基础上,明确软件故障树模型的组成项(顶事件、中间事件以及底事件)及层次关系,形成机载软件故障层次关系模型。

1) 故障树模型的顶事件确定

考虑到建立故障树模型的成本和有效性,相关标准[2,4]推荐选取影响后果较为严重的故障开展建模。因此,本文选取影响等级较高(A或B级)的软件故障作为顶事件,开展故障层次及逻辑关系建模。

2) 故障树模型的中间事件确定

依据软件故障相关的功能以及子功能(交联硬件),确定故障树模型的中间事件。即从功能以及子功能的输出数据、时序约束、通信过程、设备状态、系统交互等角度,识别相关功能以及子功能的故障模式(例如,功能无输出、交联硬件设备下电等),作为故障树模型的中间事件。

3) 故障树模型的底事件确定

依据软件故障相关的软件项,确定故障树模型的底事件。即从模块/单元的输出数据、时序约束、软硬耦合、状态场景等角度,识别相关模块/单元的故障模式(例如,控制模块未输出控制指令、处理单元延迟输出目标数据等),作为故障树模型的底事件,也即识别导致软件故障的基本原因。

基于静态体系结构的机载软件故障层次组成模型构建示意如图2所示。

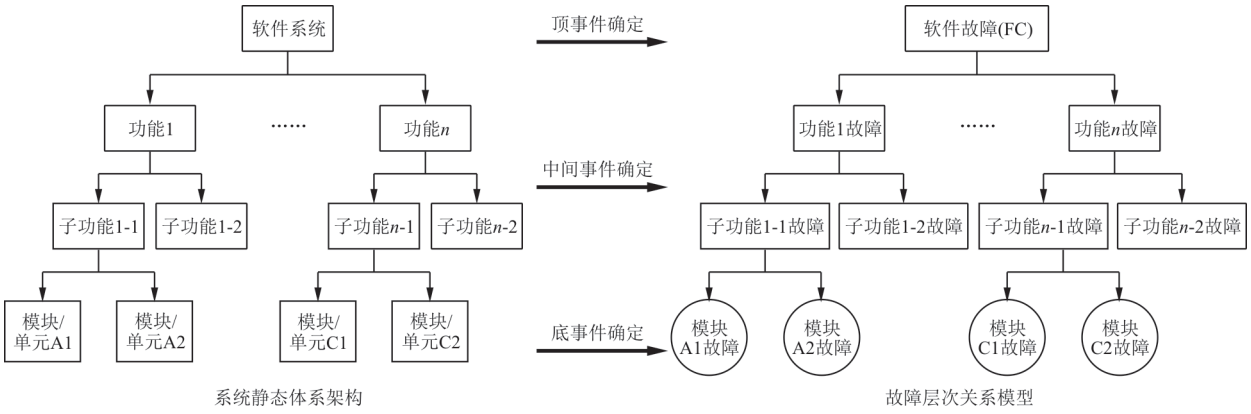

图 2　基于静态体系架构的机载软件故障层次关系模型构建方法

Fig. 2　The fault hierarchy model of the airborne software based on the architecture

2.3　基于动态运行框图的机载软件故障逻辑关系建模方法

目前,已有的机载软件故障树等故障模型的建立方法研究,对于组成项之间逻辑关系(即与门、或门、与或门等)的构建多是依据人的主观经验,缺乏规范统一的确定方法。针对此问题,本文借助软件系统的动态运行框图,提出机载软件故障逻辑关系模型的建立方法。即依据软件故障树模型组成项(顶事件、中间事件以及底事件)之间的串联、并联、串并联、并串联等动态运行关系,确定故障模型中间事件/底事件之间的逻辑关系,形成机载软件故障逻辑关系模型,具体方法如下:

1) 或门构建方法:如果组成项之间的运行关系是串行,即任一组成项发生故障,都将导致顶事件“软件故障”的发生,则对应的故障树逻辑关系是“或门”。

2) 与门构建方法:如果组成项之间的运行关系是并行,即全部组成项发生故障,才会导致顶事件“软件故障”的发生,则对应的故障树逻辑关系是“与门”。

3) 或与门构建方法:如果组成项之间是先串联后并联的关系,则对应的故障树逻辑关系是“或与门”。

4) 与或门构建方法:如果组成项之间是先并联后串联的关系,则对应的故障树逻辑关系是“与或门”。

5) 表决门构建方法:如果组成项之间是表决框图,则对应的故障树逻辑关系是“表决门”。

综上,构建基于静态架构与动态框图的机载软件故障模型如图 3 所示。

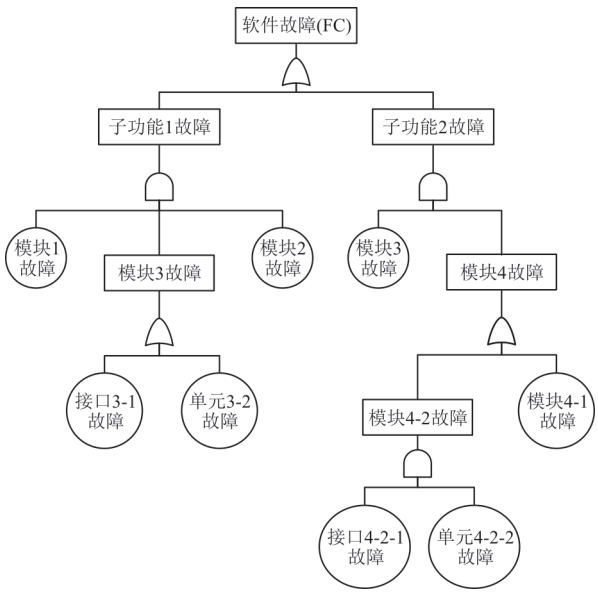

图 3　基于系统架构与运行框图的机载软件故障模型

Fig. 3　The airborne software fault model based on the architecture and block diagram

3　机载软件故障原因分析与安全性需求制定

3.1　需求与准则驱动的故障原因分析

参考 DO-178C、GJB/Z 102A 等标准要求以及历史失效数据,制定如下的故障原因分析准则:

1) 非法接口数据准则

针对软件的外部接口数据,识别非法数据取值、无效通信过程、非法操作行为、异常设备状态等,作为故障原因,例如取值超出值域范围、数据帧报文格式异常、设备处于故障状态等。非法接

口数据准则示例如表1所示。

表1 非法接口数据准则
Table 1 Illegal interface data rule

序号	故障原因分析准则
1	对具有连续型值域的数据进行检查，分析数据取值为“大于值域上限、小于值域下限”等异常值的情况下输出的正确性
2	对具有离散型值域的数据进行检查，分析数据取值为“有效值域外未定义的异常值”等情况下输出的正确性
3	……

2）无效功能逻辑准则

针对软件功能处理逻辑，识别非法判定条件、异常分支路径等，作为软件故障原因，例如，功能判定条件由成立变为不成立、功能分支路径不可达等。无效功能逻辑准则示例如表2所示。

表2 无效功能逻辑准则
Table 2 Invalid function logic rule

序号	分析准则
1	对功能的执行条件进行检查，分析功能执行过程中执行条件“不满足、再次满足”等情况下功能处理的正确性
2	对功能的逻辑判断条件和逻辑分支进行检查，分析“逻辑恒假、逻辑恒真、逻辑判断条件错误、逻辑分支遗漏”等情况下功能处理的正确性
3	……

3）异常时序约束准则

针对功能和接口的时序约束，识别功能执行超时、数据采集超时、非法操作顺序等，作为软件故障原因，例如，功能滞后开始执行、数据采集周期大于规定周期等。异常时序约束准则示例如表3所示。

表3 异常时序约束准则
Table 3 Abnormal sequence constraint rule

序号	分析准则
1	针对关键功能的持续时间进行检查，分析功能持续时间大于规定时间等情况下，功能处理的正确性
2	针对功能的执行条件中的输入数据取值进行检查，分析功能执行条件“提前满足、滞后满足”等情况下，功能处理的正确性
3	……

4）功能组合冲突准则

针对软件功能组合关系，识别功能并发冲突、串行异常、设备调用冲突等，作为软件故障原因，例如，多项功能同时执行导致控制冲突、多功能操作序列异常等。功能组合冲突准则示例如表4所示。

表4 功能组合冲突准则
Table 4 Function combination conflict rule

序号	分析准则
1	分析由于多个功能的执行条件满足、导致并发执行的情况下多功能输出的正确性
2	多个功能同时对相同数据进行读写操作，分析当出现数据读写冲突等情况下，多功能处理的正确性
3	……

5）异常状态场景准则

针对软件的工作状态与任务阶段，识别状态异常切换、非法时序、未定义场景等，作为软件故障原因，例如，状态运行超时、功能执行过程中，状态异常切换等。异常状态场景准则示例如表5所示。

表5 异常状态场景准则
Table 5 Abnormal state scene rule

序号	分析准则
1	对系统状态、软件状态进行检查，分析“不存在对应软件状态、软件与系统不一致”等情况下，功能执行的正确性
2	对状态进入条件进行检查，分析状态执行过程中进入条件“满足、不满足、再次满足”等情况下，功能执行的正确性
3	……

限于篇幅，本文所阐述的几类故障原因分析准则仅是个示例。在开展故障建模分析工程应用时，需要结合机载软件特点（例如，实时控制类、数据处理类等），依据下述主要来源来制定具体的准则。

1）标准要求：即针对DO-178C、GJB/Z 102A等标准中的故障识别要求，结合机载软件系统特点，进行对标分析，形成分析准则。例如，可以将GJB/Z 102A中的“接口故障：应充分估计接口的各种可能故障，并采取相应的措施”这条要求，可以落地为1项“非法接口数据准则”，即：对具有连续型值域的数据进行检查，分析数据取值为“大于值域上限、小于值域下限”等异常值的情况下输出的正确性”，并给出相应的设计策略。

2）历史失效数据：可以针对机载软件的测试问题、外场问题等数据中的问题原因，进行提炼分析，形成可复用的故障原因分析准则。例如，一个

历史问题的原因是:软件接收到“飞行阶段”数据为数值5(未定义),而软件无法处理未定义的“飞行阶段”,则可将其提炼成为1项“非法接口数据准则”,即:对具有离散型值域的数据进行检查,分析数据取值为“有效值域外未定义的异常值”等情况下输出的正确性。

综合上述分析,故障原因分析准则制定的充分性是通过“覆盖标准要求”“总结历史经验”这两个方面来保证的。同时,本文主要依据GJB 438B规定的软件需求(即工作状态方式、功能项、外部接口项),将故障原因分析准则分为5大类别,确保故障原因覆盖通用的软件需求类型,再一次确保故障原因分析准则的充分性。

3.2　基于故障原因的安全性需求获取策略

针对所识别的软件故障原因,参考GJB/Z 102A等标准要求,从“事前预防”或“事后控制”等角度,制定软件安全性需求获取策略,确保导致故障的原因都已得到控制或消除,构建机载软件故障分析验证的确认与验证(V&V)闭环。

1) 针对接口故障的软件安全性需求获取策略

可针对数据、时序、通信、设备等接口故障原因,制定安全性需求,示例如表6所示。

表6　针对接口故障的软件安全性需求

Table 6　Safety requirement for interface fault

类别	安全性需求获取说明	安全性需求获取示例
针对数据故障的安全性需求	针对接口数据的取值、变化等异常情况进行检查,提出处理措施,确保软件正确处理数据故障,不产生异常输出数据	针对连续型数据取值进行检查,若取值超出有效区间(例如大于区间上限,或者小于区间下限),给予相应处理(例如置为区间上限或下限值)
针对时序故障的安全性需求	针对数据的取值周期、时刻、持续时长等时序约束的异常情况进行检查,提出处理措施,确保软件正确处理时序异常的数据	对接口数据的取值持续时长进行判断,若持续时长大于或小于规定的时长,则给予提示,并进行相应处理
针对通信故障的安全性需求	针对接口通信协议的格式、内容等各类异常情况进行检查,提出处理措施,确保软件能够正确处理通信异常的接口数据	若出现通信异常情况(例如线路中断、开路、短路等),应给出对应处理策略,例如切换余度或者输出告警提示
针对设备故障的安全性需求	对交联源/目的设备进行检查,分析设备处于下电、故障等异常情况,提出处理措施,确保软件正确与设备的数据交互	对源设备的工作状态进行检查,若处于下电模式或者初始化模式,则不处理源设备发送的输入数据

2) 针对功能故障的软件安全性需求获取策略

可针对功能时序、逻辑、并发、软硬耦合等功能处理故障原因,制定安全性需求,示例如表7所示。

表7　针对功能故障的软件安全性需求

Table 7　Safety requirement for function fault

类别	安全性需求获取说明	安全性需求获取示例
针对功能时序故障的安全性需求	针对功能运行时间、启动时刻等时序约束的异常情况,进行检查并提出处理措施	对功能运行时间进行检查,若功能运行时间大于规定时间或者小于规定时间,则给予相应处理(例如停止执行功能)
针对功能逻辑故障的安全性需求	针对判断条件、数据操作、算法迭代等逻辑的异常情况,进行检查,提出处理措施	针对功能逻辑中的迭代过程进行检查,若其超时收敛或者无法收敛,应给予相应处理(例如退出迭代过程)
针对功能并发故障的安全性需求	针对多项功能并发时的冲突异常情况,进行检查,提出处理措施	若多项功能同时对同一接口数据进行取值操作,则应规定多项功能之间的优先级,避免多项功能对同一接口数据进行取值出现冲突
针对软硬耦合故障的安全性需求	针对软硬件之间的数据耦合或者控制耦合关系进行检查,并给出相应处理,避免耦合冲突导致软件故障	若目的设备未在规定时间内响应软件输出的控制指令,则应重新发送控制指令。若连续三次重新发送指令后,目的设备仍未响应,则应进行告警,控制系统进入安全状态

3) 针对状态故障的软件安全性需求获取策略

可针对功能时序、逻辑、并发、软硬耦合等功能处理故障原因，制定安全性需求，示例如表 8 所示。

表 8 针对状态故障的软件安全性需求

Table 8 Safety requirement for state fault

类别	安全性需求获取说明	安全性需求获取示例
针对工作状态失效的安全性需求	针对软件或者系统工作状态运行过程进行检查，提出处理措施	若初始化、维护等状态的运行时长超出规定阈值，则给予告警提示，并进行相应处理
针对状态转移的安全性需求	针对状态之间的转移条件和路径进行检查，提出处理措施	若当前工作状态向多个其他工作状态之间的转移条件同时成立，则应规定优先级，避免软件同时进入多个工作状态

4 典型工程应用示例

选择某型机载刹车控制软件，进行工程应用示例分析，验证机载软件故障识别方法的可行性和规范性。

1) 基于功能失效分析的机载软件故障识别

依据 2.1 节中的方法，识别刹车控制软件故障示例，如表 9 所示。

表 9 机载刹车控制软件故障识别示例

Table 9 The fault identification of the airborne brake control software

序号	功能	功能失效状态(即软件故障)	所属阶段	影响等级
1	防滑刹车控制	FC-01:进入起动状态后，软件未输出刹车控制指令	起动	B
2	制动刹车控制	FC-02:软件输出的制动指令大于值域上限	运行	A

2) 基于静态体系架构的故障层次组成建模

选择影响等级为 A 的软件故障 FC-02，作为故障模型的顶事件。依据机载刹车控制软件体系架构，针对“制动刹车控制功能”FC，构建软件故障层次组成模型，如图 4 所示。

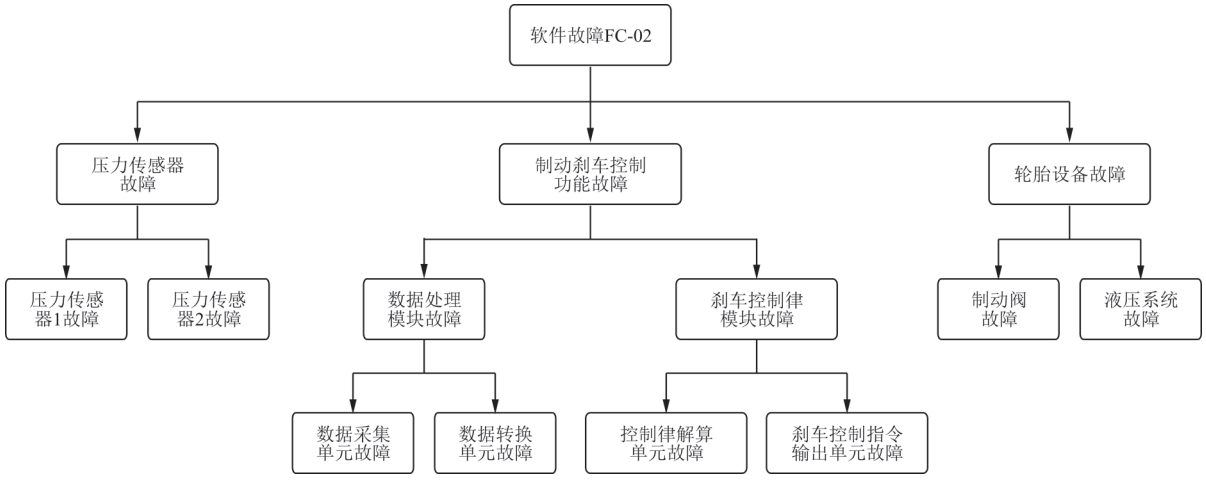

图 4 机载刹车控制软件故障层次组成模型

Fig. 4 The fault hierarchy model of the airborne brake control software

3) 基于动态运行框图的故障逻辑关系建模

基于机载刹车控制软件故障层次组成模型，构建“制动刹车控制功能”下的动态运行框图，如图 5 所示。

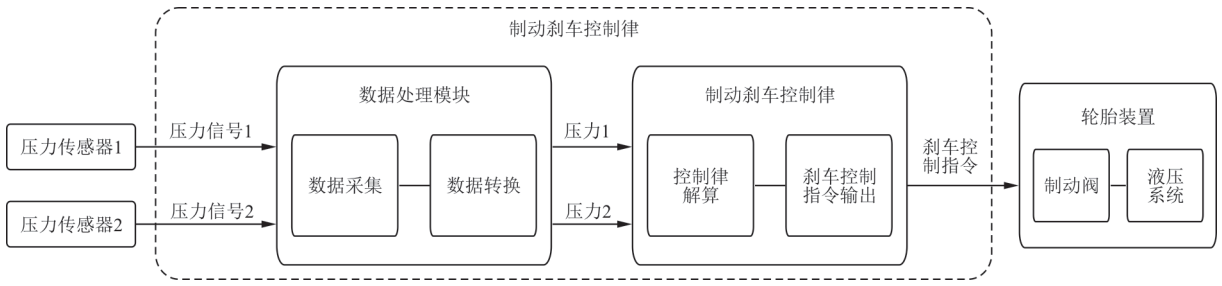

图 5　机载刹车控制软件动态运行框图

Fig. 5　The dynamic operational model of the airborne brake control software

依据图 5,识别故障层次组成模型中间事件和底事件的逻辑关系。

第一层中间事件的逻辑关系确定:“制动刹车控制、压力传感器、轮胎装置”等在实现“制动刹车控制功能”时,整体运行关系是串行关系。因此,三者中间事件的逻辑关系为“或门”。

第二层中间事件的逻辑关系确定:“数据处理模块”和“制动刹车控制律”的运行关系也是串行。因此,中间事件的逻辑关系为“或门”。

重复上述过程,依次识别底事件之间的逻辑关系,形成机载软件故障逻辑关系模型,如图 6 所示。

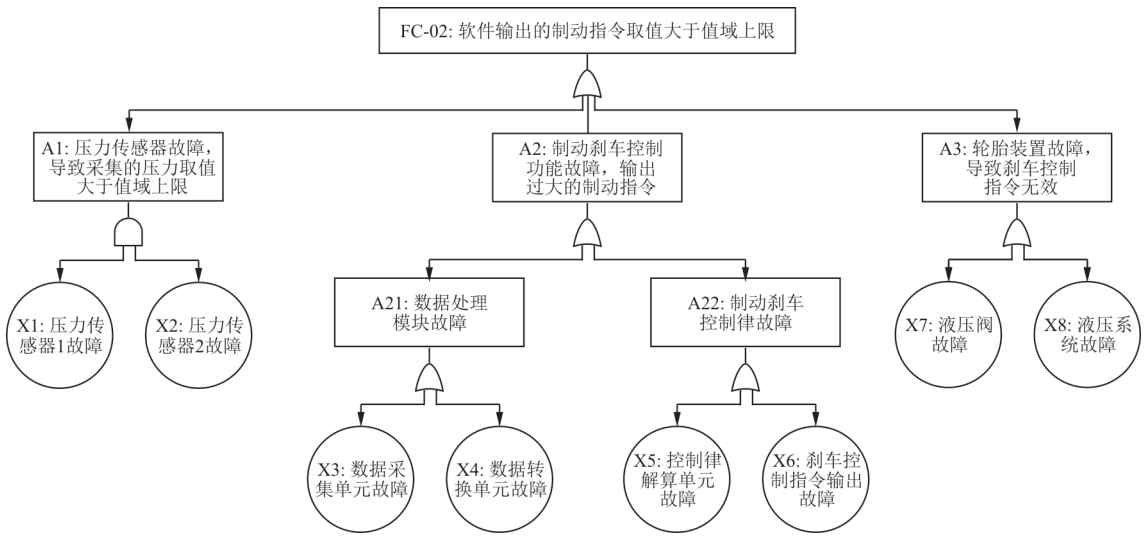

图 6　机载刹车控制软件故障逻辑关系模型+故障层次组成模型

Fig. 6　The fault hierarchy model and fault logic relationship model of the airborne brake control software

4) 基于需求的机载软件故障原因分析

以底事件 X5“控制律解算单元故障”为例,依据“控制律解算单元”需求,其输入数据“轮胎压力”为连续型数据。因此,选择“非法接口数据准则”来识别故障原因,即压力取值超出值域范围。

5) 基于故障原因的机载软件安全性需求获取

故障原因“压力取值超出值域范围”与输入数据取值相关,选择“针对输入输出接口的软件安全性需求获取策略”,形成安全性需求 SAFETY-Q1,如果压力取值超出值域范围,则置为值域的上限值,实现对底事件 X5“控制律解算单元故障”的控制。

重复上述过程,对图 6 中所有的底事件和中间事件进行故障原因分析和安全性需求获取,自下而上逐层实现了对软件故障 FC-02 的控制。

5　结　论

1) 本文形成符合标准、规范有效的机载软件故障建模与分析技术,能够提升故障识别效率,消除薄弱环节。

2) 该技术支撑构建故障树模型,能够有效识别软件故障原因,解决了传统故障树建模规范性较差等不足。

3) 通过构建机载软件故障分析与验证的

V&V闭环，保障机载软件的可靠性与安全性满足型号要求。

参考文献

[1] LEARMOUNT D. Never again: Flight Int'l [R]. US: Anon, 2012.

[2] 中国人民解放军总装备部. 软件安全性设计指南：GJB/Z 102A[S]. 北京：中国人民解放军总装备部，2012.
General Equipment Department of the Chinese People's Liberation Army. Military software safety design guide: GJB/Z 102A [S]. Beijing: General Equipment Department of the Chinese People's Liberation Army, 2012. (in Chinese)

[3] 黄志球，徐丙凤，阚双龙，等. 嵌入式机载软件安全性分析标准、方法及工具研究综述[J]. 软件学报，2014，25(2)：200-218.
HUANG Zhiqiu, XU Bingfeng, KAN Shuanglong, et al. Survey on embedded software analysis standards, methods and tools for airborne system [J]. Journal of Software, 2014, 25(2): 200-218. (in Chinese)

[4] RTCA. Software considerations in airborne systems and equipment certification: DO-178C [S]. Washington DC: RTCA, 2011.

[5] NASA. Software safety guide: NASA-8719 [S]. New York: NASA, 2004.

[6] 张杰，徐一初. 故障树分析方法在FADEC控制软件中的应用研究[J]. 测控技术，2018，37(12)：22-25.
ZHANG Jie, XU Yichu. Application and research on fault tree analysis of control software in FADEC [J]. Measurement & Control Technology, 2018, 37(12): 22-25. (in Chinese)

[7] 檀德宾. 公共安全系统的软件故障树构建及应用[D]. 上海：上海交通大学，2014.
TAN Debin. Software fault tree construction and application of public safety system [D]. Shanghai: Shanghai Jiao Tong University, 2014. (in Chinese)

[8] 汪相国，宋涛，杨柳. 故障树在嵌入式控制软件可靠性设计中的应用[J]. 中国高新科技，2021(22)：86-87.
WANG Xiangguo, SONG Tao, YANG Liu. Application of fault tree in embedded control software reliability design [J]. China Innovative Technology, 2021(22): 86-87. (in Chinese)

[9] 张辉，陈世浩. 民用飞机着陆系统安全性评估的故障树分析[J]. 航空工程进展，2021，12(1)：64-72.
ZHANG Hui, CHEN Shihao. Fault tree analysis for safety assessment of civil aircraft landing system [J]. Advances in Aeronautical Science and Engineering, 2021, 12(1): 64-72. (in Chinese)

[10] 林红，杨瀚程. 基于Petri网的软件故障树分析[J]. 火控雷达技术，2013，42(4)：40-43.
LIN Hong, YANG Hancheng. Petri Net-based analysis on software fault tree [J]. Fire Control Radar Technology, 2013, 42(4): 40-43. (in Chinese)

[11] 王思琪，黄志球，黄传林，等. 一种基于状态事件故障树的软件安全性分析方法研究[J]. 小型微型计算机系统，2016，37(1)：12-17.
WANG Siqi, HUANG Zhiqiu, HUANG Chuanlin, et al. Method based on state/even fault tree for safety analysis of software [J]. Journal of Chinese Computer Systems, 2016, 37(1): 12-17. (in Chinese)

[12] 石柱，郑重. 软件故障树分析实例研究[J]. 航天控制，2014，32(6)：67-71.
SHI Zhu, ZHENG Zhong. A case study on software fault tree analysis [J]. Aerospace Control, 2014, 32(6): 67-71. (in Chinese)

[13] 樊茜，何雨昂，刘海山，等. 基于动态故障树的伺服飞控软件故障诊断方法及应用[J]. 电子技术与软件工程，2017(16)：75-76.
FAN Qian, HE Yuang, LIU Haishan, et al. Servo flight control software fault diagnosis and its application based on dynamic fault tree [J]. Electronic Technology & Software Engineering, 2017(16): 75-76. (in Chinese)

[14] 张红林，张春元，刘东. 一种适用于具有相互依赖基本事件和重复事件的动态故障树独立模块识别方法[J]. 计算机学报，2012，35(2)：229-243.
ZHANG Honglin, ZHANG Chunyuan, LIU Dong. An identification method of independent module applying to dynamic fault tree with interdependent basic events and repeated events [J]. Chinese Journal of Computers, 2012, 35(2): 229-243. (in Chinese)

[15] BALAJE T, JEFFERY K. Large-scale fault tree implementation: a software tutorial[C]// 2022 Annual Reliability and Maintainability Symposium. [S.l.]: AIAA, 2022: 24-27.

[16] NATALIYA Y, MORAYO A. Model-based quantitative fault tree analysis based on FIDES reliability prediction [C]// 2020 IEEE International Symposium on Software Reliability Engineering. [S.l.]: IEEE, 2020: 161-162.

[17] JUNG S, YOO J, LEE Y J. A software fault tree analysis technique for formal requirement specifications of nuclear reactor protection systems [J]. Reliability Engineering and System Safety, 2020(23): 103-122.

[18] PHILIPPOV V. Reliability model of air traffic control with IMA onboard data link system [J]. Reliability and Statistics in Transportation and Communication, 2021, 23: 267-275.

作者简介：

温晓玲(1979—)，女，硕士，副总设计师。主要研究方向：软件工程，嵌入式软件开发。

姜梦岑(1985—)，女，硕士，高级工程师。主要研究方向：软件工程，嵌入式软件开发。

艾笑天(1994—)，男，硕士，工程师。主要研究方向：软件工程，嵌入式软件开发。

柳庆宇(1990—)，男，硕士，工程师。主要研究方向：软件工程，嵌入式软件开发。

(编辑：丛艳娟)