

面向适航认证要求的机载软件模块化分层架构设计方法

杨 扬 毛 威 周 靖

陕西飞机工业有限责任公司 陕西 汉中 723200

摘 要: 为满足机载软件DO-178C、ARINC653等适航认证严苛要求, 解决传统架构故障蔓延、可验证性差、取证效率低等问题, 本文提出模块化分层架构设计方法。构建五层层级架构体系, 明确模块拆分标准与接口规范, 通过隔离设计、追溯优化提升合规性。经飞控工程案例验证, 该方法可强化故障隔离能力, 缩减适航取证与测试成本, 可为高安全机载软件研发提供技术支撑。

关键词: 适航认证要求; 机载软件; 模块化分层架构; 设计方法

引言: 机载软件是航电系统安全运行的核心, 其架构合规性直接决定适航认证成败。传统平铺式软件架构存在耦合度高、故障易跨域传播、层级管控混乱等缺陷, 难以适配高等级适航的安全性及可验证性要求。为破解机载软件设计与适航标准脱节难题, 本文结合模块化与分层架构理论, 搭建适配适航约束的标准化设计体系, 规范架构搭建、模块拆分及验证流程, 助力国产机载软件高效合规取证。

1 机载软件适航认证规范与架构设计基础理论

1.1 机载软件适航认证核心标准体系

(1) DO-178C作为机载软件适航认证的核心依据, 将软件按失效影响程度划分为A至E五个等级, 等级越高, 目标覆盖与验证要求越严格。该标准强调安全性目标的分解与分配, 要求全生命周期内建立从需求、设计、编码到测试的完整可追溯链, 并规定基于需求的测试验证方法, 确保软件实现与高层需求一致。(2) ARINC653定义了综合化航电系统的分区架构, 从操作系统层面规定时空分区、调度策略及分区间通信机制, 强制实现应用间的故障隔离与资源隔离, 为多层级软件集成提供确定性运行环境。(3) DO-254针对硬件设计提出开发与验证要求, 在硬件适配场景下, 软件架构需与硬件容错机制、故障检测覆盖率协同匹配, 确保软硬件接口数据的一致性及错误响应的协调性。

1.2 模块化与分层架构核心理论

(1) 机载软件模块化设计遵循高内聚低耦合原则, 要求功能职责单一、接口标准化, 支持独立开发与替换, 同时通过定义明确输入输出契约提升复用性, 降低变更影响范围。(2) 分层架构采用自底向上的纵向分层机制, 将底层硬件驱动、操作系统服务、应用功能逐级解耦, 同时横向按功能域分块管理, 形成独立故障边界, 阻止错误跨层蔓延。(3) 该架构模式适配机载场景

的实时性、安全性与可靠性需求: 分层调度可保障关键任务的时间确定性, 模块隔离有效控制单点故障风险, 标准化接口与清晰层次结构便于开展安全性分析与符合性验证。

1.3 适航导向的架构设计约束指标

(1) 安全性指标要求架构具备故障隔离与容错能力, 通过分区机制与冗余设计实现故障containment, 并采用危害分析量化风险可控性, 确保单故障或多故障条件下系统仍处于安全状态。(2) 可验证性指标强调模块应具备独立测试能力, 层级间的依赖关系可追溯至系统需求, 接口数据格式与协议可校验, 从而支持需求覆盖分析与变更影响评估, 降低适航审查风险。(3) 工程性指标关注架构的可扩展性与模块复用率, 通过接口标准化和功能抽象提高资产重用效率, 同时优化适航认证适配效率, 减少重复验证成本, 加速项目取证进程。

2 面向适航认证要求的机载软件模块化分层架构设计方法

2.1 机载软件整体分层架构体系构建

(1) 分层原则。依据适航安全等级(DO-178CA~E级)、功能属性(关键控制/数据处理/状态监控/管理支持)及运行权限(内核/系统/用户态)三维度决策。高安全等级功能置于受保护底层, 低等级应用部署上层隔离区, 权限自底向上逐级收窄, 底层服务可被上层调用而反向受限, 从架构源头实现错误遏制。(2) 五层架构。自底向上: 硬件抽象层(封装处理器、存储、I/O差异, 提供统一访问接口); 操作系统适配层(对接ARINC653或POSIX, 实现分区管理、任务调度及时钟服务); 平台服务层(提供日志、配置、健康监控、数据分发等公共服务); 功能应用层(承载导航、通信、显示等航电功能模块); 业务任务层(面向飞行阶段的任务编排与状态管理)。各层职责清晰, 仅通过标准化接口交互, 跨

层调用受限,保障独立验证^[1]。(3)层级边界与交互。硬件抽象层仅处理寄存器读写与中断响应,不涉业务逻辑;操作系统适配层提供分区调度与时间管理,不处理应用数据;平台服务层提供通用中间件,不感知应用语义;功能应用层实现独立算法,不得直接访问硬件或操作系统原语;业务任务层按飞行阶段组织功能组合,调用下层服务但不得修改下层配置。层级间采用同步请求/响应、异步事件通知及数据订阅发布三种交互模式,协议明确参数范围、超时处理与错误码定义,跨层调用记录日志以支撑适航追溯。

2.2 适配适航要求的软件模块化拆分方法

(1)模块粒度划分标准:依据安全等级差异化拆分,A/B级模块须拆至最小功能单元,C/D级可适当放宽;同时遵循功能独立性(单一职责、无隐藏副作用)与适航验证单元(每模块对应独立测试用例集)准则。跨等级模块必须物理分离,同一等级可合并管理。(2)核心功能模块拆分:通信模块负责总线数据收发与协议解析,细分为链路管理、消息路由、差错检测子模块;调度模块管理周期/非周期任务的时间与事件触发逻辑,包含任务编排、时限监控、超时恢复子模块;监控模块采集系统健康状态与故障日志,分为数据采集、阈值判断、告警上报子模块;数据处理模块执行导航解算等算法密集型操作,按算法类型拆解为滤波、变换、融合等独立单元。各子模块通过消息队列或共享接口交互,不直接依赖内部实现。(3)模块接口标准化设计:统一采用IDL描述模块输入输出参数、数据类型与取值范围;交互协议固定格式(消息头含模块ID、序列号、时间戳、校验和,消息体采用TLV结构);调用规范明确同步超时阈值、异步回调优先级及标准错误码枚举。所有接口定义入版受控,确保需求—设计—代码—测试四方追溯链完整^[2]。

2.3 架构适航合规性关键设计优化

(1)层级隔离设计:时空分区隔离参照ARINC653,每分区分配独立存储空间与时间配额,分区切换由操作系统强制实施,确保分区失效不相互影响;故障横向隔离方面,同层不同模块运行于独立任务上下文,采用看门狗与心跳监测活性,单模块故障仅复位该模块,不横向蔓延。(2)模块化可追溯设计:构建从系统需求→软件需求→模块设计→代码→测试用例的垂直追溯矩阵,每模块对应唯一需求编号、设计文档节号及测试用例集;同时建立水平追溯记录模块间接口依赖关系。所有追溯关系存储于配置管理工具,配套自动化脚本定期检查覆盖完备性^[3]。(3)实时性与可靠性优化:设计双

层级调度协同机制——操作系统适配层负责分区时间轮转,功能应用层内采用优先级抢占调度,按速率单调算法分配时间预算;可靠性方面,关键模块部署热备冗余副本,通过三模冗余表决输出结果,健康监控模块实时检测超时与计算错误,触发降级或重构策略,确保时限超限时维持安全降级。

2.4 架构设计流程与规范化实施步骤

(1)适航需求拆解与架构指标映射步骤:从系统适航要求提取软件质量属性需求,逐条转化为量化架构指标,包括分区数量与资源配额、模块最大圈复杂度(≤ 15)、接口响应上限、故障检测覆盖率($\geq 98\%$)等,建立需求—指标映射表经评审后作为设计基线^[4]。(2)分层架构搭建与模块化迭代设计流程:采用自顶向下逐层分解与自底向上逐步集成相结合的迭代模式。先确定五层边界与接口契约,再按安全等级从高到低依次细化各层模块,每完成一层即进行接口仿真测试;集成时按硬件抽象层至业务任务层顺序逐层组装,每层集成后执行冒烟测试与接口一致性检查,确认无误后叠加后续层。(3)架构初步合规性自查与迭代优化流程:依据DO-178C合规性检查单开展自查,包括安全性目标分配完整性、模块测试独立性、接口文档与代码一致性、故障隔离机制有效性(通过故障注入实验验证)。对不符合项分类记录并制定改进计划,每项经回归验证后纳入基线,通常迭代2~3轮直至全部通过,最终形成架构设计符合性报告存档备审。

3 架构验证、工程应用与总结展望

3.1 架构适航合规性验证方案设计

(1)验证维度聚焦四大核心方向:安全性维度验证故障隔离机制的有效性、容错触发的正确性及风险控制措施的覆盖率;模块化维度量化评估模块内聚度、耦合度及接口标准化程度;分层解耦维度检验层级依赖方向的合规性与跨层调用的受控性;可验证性维度确认每项适航需求是否具备可测试的架构实现,保障审查证据的完整可采信。(2)验证方法采用四类互补手段:静态架构分析自动扫描代码结构,检查循环依赖、跨层违规及接口一致性;动态仿真测试在目标硬件或全数字环境中注入典型故障与极限负载,验证运行时隔离与调度行为;模块单元测试针对各模块开展接口契约一致性及分支覆盖率验证;适航追溯性校验借助自动化工具检查需求-设计-代码-测试的双向追溯链完整性,杜绝断裂与遗漏。(3)设定量化指标体系:故障传播深度 ≤ 1 层、模块间耦合度 ≤ 0.3 、层级依赖合规率100%、追溯覆盖率100%、分区切换时延抖动 $\leq \pm 5\%$ 、最坏执行时间实测值

不超过预算值的80%。所有指标经适航专家评审确认阈值,任何单项不达标须回溯整改并复测,全部通过后方可判定为合格。

3.2 工程案例应用与结果分析

(1)选取某型飞行控制系统为验证载体,系统涵盖飞控计算机、传感器采集、作动器控制及人机显控交互,软件规模约12万行源代码,安全等级涵盖A级至D级,目标硬件平台基于PowerPC处理器,操作系统符合ARINC653分区规范。(2)按五层架构实施重构,共划分42个核心模块,其中A/B级安全关键模块均独立部署于受保护分区,确保故障隔离。项目历时6个月经历三轮迭代优化,模块单元测试通过率达98.7%;故障注入实验表明单模块失效被有效遏制于本层内,无跨级传播;分区调度时延抖动实测控制在 $\pm 3\%$ 以内,满足实时性要求^[5]。(3)与传统单体平铺架构进行对比:适航证据准备时间缩短42%,审查问题响应周期减少35%;故障隔离成功率由72%提升至96%;单模块变更影响的模块数量由平均5.3个降至1.2个,回归测试工作量缩减40%,综合证明本方法在适航适配效率、系统可靠性与长期可维护性方面具有显著优势。

3.3 研究总结

(1)全文系统梳理了DO-178C、ARINC653、DO-254等适航标准的核心约束,结合模块化与分层架构理论,提出了一套涵盖五层架构体系构建、模块化拆分策略、适航合规性关键设计优化及规范化实施流程的完整架构设计方法,并通过工程案例完成了方法有效性的验证。(2)所提方法的核心价值在于建立了适航要求到架构设计之间的系统化映射路径,将安全性、可验证性等抽象适航目标转化为可量化、可执行的架构约束与指标;通过层级隔离与模块化解耦,实现故障域的精准控制与独立验证,有效降低了适航审查风险,为国产高安全机载软件架构设计提供了可复用的方法论参考与工程实践范式。

3.4 不足与未来展望

(1)当前架构设计方法在模块粒度划分与接口协议生成环节仍依赖架构师经验判断,缺乏自动化辅助决策工具支撑;针对多核处理器架构下的分区缓存一致性、核间干扰及中断嵌套处理等机制尚不完善;验证环节对动态并发行为的覆盖仍存在一定盲区。(2)未来需构建可配置的架构模板库,支持固定翼、旋翼、无人机等多机型平台的快速实例化适配;针对AI/ML类智能化机载软件,研究其非确定性功能在分层分区架构中的安全部署策略与运行时监控机制。(3)长远研究方向包括研发架构合规性自动化验证工具链,实现约束规则自动检查、追溯链自动生成及验证证据自动打包;探索基于模型的架构迭代优化与形式化验证方法,提升关键安全属性的证明能力,推动架构设计从经验驱动向模型驱动升级,为更高自主等级的飞行系统奠定架构基础。

结束语

本文梳理机载软件适航核心标准,构建了一套完整的适航导向模块化分层架构设计与实施体系,经工程实践证实可显著提升系统可靠性与取证效率。同时总结出当前方法存在自动化程度不足、多核适配不完善等短板。后续将搭建标准化架构模板库,研发自动化验证工具链,探索智能化机载软件的合规部署方案,推动机载软件架构向模型化、智能化合规方向迭代升级。

参考文献

- [1]王康星.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,2025,57(1):146-154.
- [2]孙毅刚.IMA架构的ACARS通信管理单元设计及验证[J].计算机应用与软件,2025,42(1):58-64.
- [3]王文浩,毕文豪.面向适航认证基于模型的FADEC软件研发[J].航空发动机,2024,50(6):112-118.
- [4]李鑫,刘畅,陈勇.适航合规的机载嵌入式软件模块化架构优化设计[J].航空计算技术,2023,53(4):89-93.
- [5]肖国梁.基于ARINC653标准的机载软件分层隔离架构设计[J].电子技术与软件工程,2022,11(18):45-48.