

IMA 系统软件集成过程的相关考虑

王 羽, 洪 沛, 王 震

(中国航空无线电电子研究所, 上海 200233)

[摘 要] IMA 架构在军机民机中的应用越来越成熟, 其灵活的架构思想推动了软件往更高的可重用性方向发展, 同时也增加了软件系统的集成难度。因此, 在基于 IMA 软件系统开发过程中, 软件集成活动越来越重要, 软件集成能力的要求越来越高。本文根据已有项目经验, 结合通用集成开发环境的应用, 对 IMA 软件系统集成过程提出了一系列注意事项及相关实践说明。

[关键词] IMA; ARINC 653; 集成

[中图分类号] TP311.1

[文献标识码] A

[文章编号] 1006-141X(2021)03-0037-06

Considerations for Software Integration Process in IMA Systems

WANG Yu, HONG Pei, WANG Zhen

(China National Aeronautical Radio Electronics Research Institute, Shanghai 200233, China)

Abstract: The application of Integrated Modular Avionics System(IMA) architecture in military and civilian aircraft is becoming more and more sophisticated. The flexible architecture promotes the development of software towards higher reusability, and it also increases the difficulty of software system integration. So, in the development process of software system based on IMA, software integration activities become more and more important, and needs higher and higher software integration ability. Several considerations are presented for the software integration process based on the experience of practical projects and the application of General integrated development environment.

Key words: IMA; ARINC 653; integration

综合模块化航空电子 (IMA: Integrated Modular Avionics) 体现了一种开放式航电系统设计思想, 系统由一系列灵活可组的应用功能软件和通用硬件资源组成, 驻留于同一硬件处理模块上的应用软件在满足其功能、性能以及安全性级别等要求的基础上, 共用处理器、内存、输入输出 (I/O: Input / Output) 等底层资源, 在减少硬件模块和设备线缆的同时, 降低了系统开发和生产的成本。

基于 IMA 架构的系统软件开发, 在要求提高可重用性的同时, 对软件的集成能力也提出了更高的要求, 其软件开发过程中的集成角色和集成活动也显得尤为重要。本文结合对 IMA 系统的软件开发过程进行分析, 根据已有项目的经验, 针对软件集成过程中的重要环节进行思考, 为后续项目的实践工作提供指导。

收稿日期: 2020-05-10

引用格式: 王羽, 洪沛, 王震. IMA 系统软件集成过程的相关考虑 [J]. 航空电子技术, 2021, 52(3): 37-42.

1 软件架构

现有比较成熟的 IMA 系统方案中, 一个通用处理模块上驻留的典型软件系统基于 ARINC 653^[1] 标准架构进行设计, 主要由板级支持包、内核操作系统、系统配置表、分区操作系统、APEX 接口^[2] 和应用软件组成, 如图 1 所示。IMA 系统的软件集成过程

指的是利用集成开发环境, 对模块资源合理性划分, 以及对各独立功能模块进行资源和功能的协调综合、测试验证过程。这里主要考虑针对基于 ARINC 653 标准实现的操作系统(如 Vxworks 653^[3]、天脉 II^[4] 等)的软件集成过程, 对于其他框架(如 Vxworks 6.X), 使用 APEX 接口组件^[5] 可以达到相应的 IMA 架构设计和软件开发集成目的。

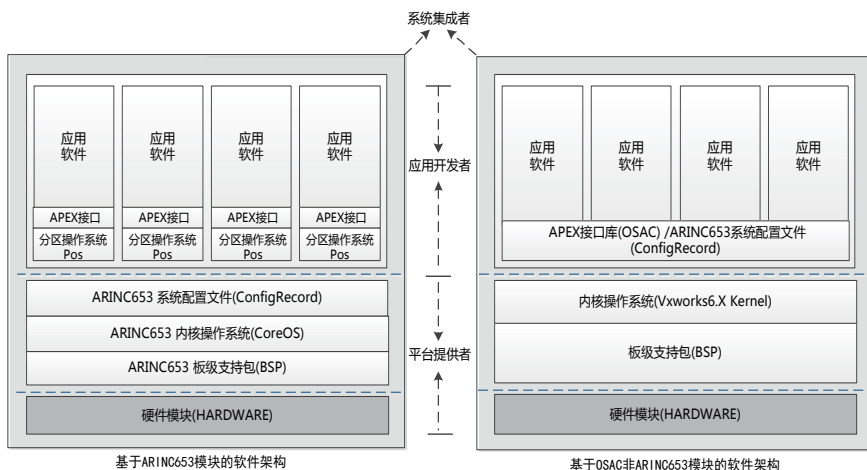


图 1 ARINC 653 架构设计及角色分布

为了与 IMA 系统模块化、综合化的特点相匹配, 在基于 ARINC 653 架构的软件系统开发过程中, 主要的参与人员可归纳为三大类: 平台提供者、应用开发者和系统集成者^[6], 如图 2 所示。其中, 系统集成者在系统开发过程中承载着重要角色, 主要职责包括:

- (1) 根据系统需求对软件系统做一个合理的资源分配, 包括应用任务和资源的划分;
- (2) 规划平台提供者和应用开发者的职责范围, 包括交付物的相关参数说明;
- (3) 在平台提供者和应用开发者的参与下, 完成系统集成配置;
- (4) 对软硬件系统进行集成, 并对软件系统进行集成测试和验证评估。

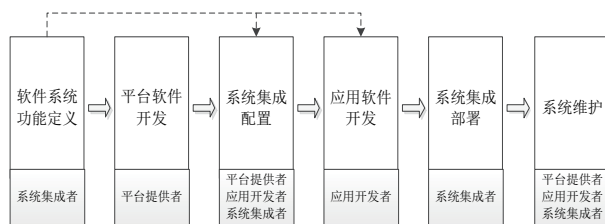


图 2 ARINC 653 架构的软件开发过程及角色分布

2 软件集成过程考虑

根据项目经验, 在一个 IMA 系统软件的研发过程中, 软件集成过程贯穿始终, 有几个环节需要重点关注, 包括: 集成需求、集成策划、集成设计、集成验证和测试、集成维护。如图 3 所示。

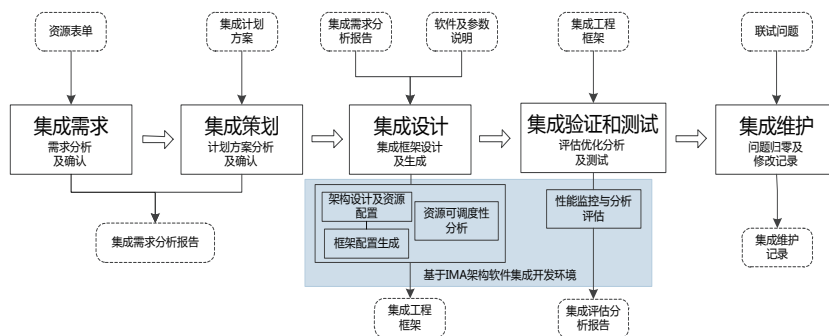


图 3 软件系统集成过程中的关键环节

2.1 集成需求

IMA 系统中软件的模块化程度高, 资源需求相对独立, 但在项目策划前期很难通过专门的方法/工具对所有系统功能的资源需求进行统一的估计。软件系统集成者可以参考 DO-297 的理念^[6], 对 IMA 硬件模块指定一份对应的“资源表单”, 向平台提供者 and 应用开发者搜集软件的模块资源, 指明模块的能力和特性, 作为软件系统前期集成需求的输入。

具体表现为:

(1) 在软件功能定义阶段, 根据整个系统的功能要求和硬件资源能力的配置进行应用分区驻留和数量的划分;

(2) 与应用开发者完成应用资源需求的评估搜集, 包括应用占用的存储和 CPU 资源、任务属性及负载, 甚至是分区健康监控的相关安全性需求;

(3) 要求平台提供者在交付底层平台时, 应明确内存、FLASH、NVRAM、IO 设备等空间的大小和地址划分, 以及应用软件的固化启动方式。同时, 要求其完成相应的功能测试以确保平台的基本功能, 并提供诸如分区/进程切换延迟、系统调用时间、存储器读写速度等性能数据作为系统配置时的参考;

(4) 明确集成过程中所需的基于 IMA 架构软件集成开发环境;

(5) 分析确认平台提供者和应用开发者提交的资源需求, 形成集成需求分析报告, 为后续集成过程做铺垫。

2.2 集成策划

软件系统集成是一个耗时耗力的重复性过程, 系统集成者应从全局角度出发, 确保各软件的集成效率。软件系统集成者需要根据集成条件做出集成策划, 确定集成顺序, 形成集成计划。具体表现为:

(1) 确定平台软件的就绪程度, 重点关注操作系统基于当前板级支持包 (BSP: Board Support Package) 的移植情况、设备驱动的组件功能以及性能测试情况;

(2) 其次, 需要确定各应用软件的就绪程度, 如软件代码的编译情况, 是否已在仿真运行环境中完成基本逻辑功能 (访问底层硬件设备功能除外) 的开发;

(3) 对已就绪的应用和平台软件按照软件功能的复杂程度进行划分, 规划集成顺序。因为功能复

杂的软件往往依赖较多的平台资源 (如驱动接口、通信资源等), 集成难度和人力成本相对较高。平台软件作为基础平台操作软件将首先被考虑集成, 为各应用提供可运行环境;

(4) 针对计划的分析和确认, 系统集成者最终形成集成方案, 可纳入集成需求分析报告的一部分, 指导后续集成活动的有序进行。

2.3 集成设计

根据集成需求分析报告, 系统集成者可以在集成设计活动中进行有序实施, 将集成需求逐一转换为实体, 主要包括系统集成配置和系统集成框架生成, 如图 4 所示。具体表现为:

(1) 接收平台提供者和应用开发者的交付物, 包括可执行软件 (或源码) 和必要的参数 (对应的保障类手册等), 作为集成的组件输入;

(2) 根据集成需求分析报告的资源信息, 通过基于 IMA 架构软件集成开发环境进行软件系统架构设计及资源的整合 (如应用配置、分区配置、端口配置以及分区调度策略配置);

(3) 通过基于 IMA 架构软件集成开发环境对分区调度策略做理论上的可行性检测和调整;

(4) 根据当前的项目阶段 (联试阶段、装机阶段) 确定最终软件映像的生成和部署方式;

(5) 根据集成策划, 通过基于 IMA 架构软件集成开发环境对各软件进行部署, 并生成软件系统集成框架, 作为后续集成验证测试及迭代的基础。

2.3.1 系统集成配置

为了提高软件代码的可移植性, 降低适航认证的成本, 基于 ARINC 653 操作系统开发的软件系统中将很大一部分的系统可变信息以配置文件的形式从代码中分离, 作为系统内核和应用软件配置、开发和文档编写的公开标准, 可以进行独立编辑和开发, 提高软件集成开发的并行度。

图 5 为 vxWorks 653 操作系统下的系统配置, 采用可扩展的树状结构设计, 构成了 XML 配置文件的全部内容, 最终经过 DO-178B^[7] 验证的 XML 编译器生成二进制文件加载到系统中^[8]。

图6为一个分区调度策略的配置, 实际项目中的运用场景可能会复杂得多, 需要考虑的因素也较多, 系统集成者在前期可以基于处理器、分区、分区任务等调度属性作为输入, 对资源可调度性进行

初步的验证和调整, 后续应结合具体系统运行数据做进一步的调整。分区应用周期和负载数据的准确性都是影响最终方案有效性的关键因素, 需要重点关注。

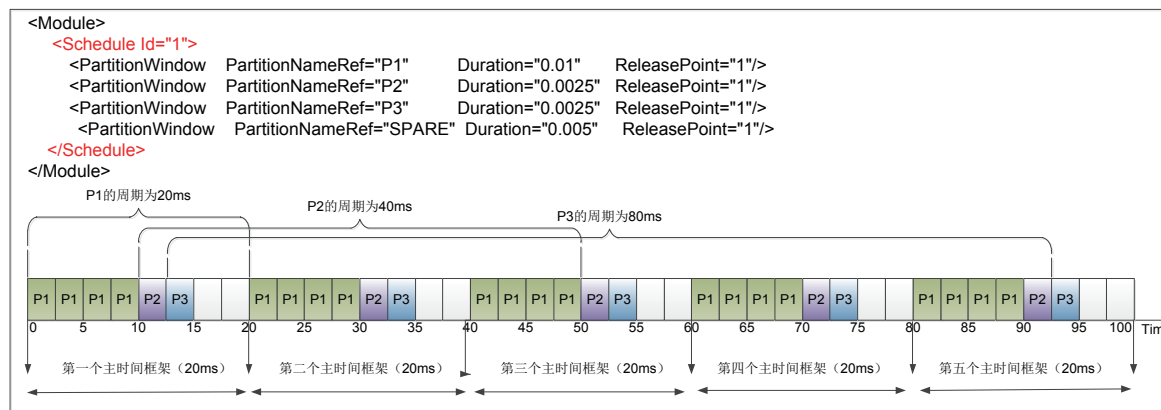


图6 分区调度策略示意图

(5) 健康监测配置

ARINC 653 标准定义了一个对故障进行分区管理、逐级报告和响应的框架, 提供了可配置的故障处理方式^[9]。如 VxWorks653 操作系统下健康监测配置可分为四类: 系统级、模块级、分区级、进程级。不同的集成角色可根据场景需要进行健康监测的配置和处理, 一般的系统级、分区级和模块级健康监测主要由系统集成者配置, 进程级健康监测由应用开发者通过创建故障处理进程来实现。

某些情况下, 系统默认的故障处理能力不能满足项目实际的需要, 系统集成者可要求平台提供者自定义的故障处理句柄(故障记录到 NVRAM 等), 由系统集成者配置到需要的健康监测机制中, 提供产品的问题排查手段。

2.3.2 系统集成框架

集成设计活动的输出为一个可编译的工程框架, 作为后续在处理模块上运行的软件系统实体, 也是后续平台提供者和应用开发者进行系统功能联试的代码工程。在 IMA 系统开发过程中很可能会遇到意外的资源需求, 在集成活动中也会对资源资源配置进行调整, 所以框架中的 XML 配置文件以及对应生成的二进制镜像应享有与应用软件同等的地位, 同样需要进行配置管理。

2.4 集成验证和测试

软件系统的集成框架作为一个软件开发实体, 需要进行验证测试, 确保整合的各应用软件能正常运行, 并通过识别、记录和采取措施解决集成验证

过程中的问题。相对于功能性需求, 非功能性需求的验证较为困难, 如性能、安全性及鲁棒性等。我们可以借助集成开发环境, 通过搜集运行时系统状态信息完成集成评估和优化迭代, 甚至可以为平台提供者和应用开发者提出相关建议。系统集成者前期可以在仿真环境中完成基本的集成验证测试工作, 后续需要在真实的硬件模块上进行。具体表现为:

(1) 对集成框架进行编译, 确保代码工程的完备性。因为在集成过程中, 特别是在编译链接过程中产生的错误都会影响到下一步的集成过程;

(2) 根据当前项目阶段选择一种映像加载部署方式(如表1所示);

(3) 根据集成需求, 完成平台和应用间接口完备性校验, 包括共享内存、系统调用、端口通讯、健康监测等接口或配置;

(4) 配合平台提供者和应用开发者完成各软件的功能测试以及系统的基础能力的验证;

(5) 通过基于 IMA 架构软件集成开发环境对应用时间和空间资源等进行动态监控, 并做出评估, 为系统配置的调整提供参考;

(6) 根据监控数据分析集成优化的必要性并主导实施优化(如应用软件优化、资源优化等);

(7) 记录集成活动过程中产生的问题、措施、建议或要求以及分析文件, 形成关于系统性能及资源的集成评估报告, 作为集成验证和测试的输出, 便于集成过程的追踪。

表 1 映像加载部署模式

项目阶段	映像加载部署模式	描述
联试阶段	Downloadable 或 Rampayload 模式	便于软件的频繁修改和调试
装机阶段	Rompayload 模式	根据系统配置文件烧录各软件可执行映像

2.4.1 系统集成评估

相对于联合式航电系统, IMA 航电系统引入了更多的共享资源, 但在使用的过程中可能会带来额外的系统消耗, 如输入输出中转、网络传输延迟、分区调度开销等一系列额外的时间损耗。当这些损耗足够大已无法忽略时, 或软件自身的处理强度已无法再调整时, 就需要考虑改变现有的软件系统架构或分区调度策略, 降低额外开销对系统需求的影响。

系统集成者可以通过基于 IMA 架构软件集成开发环境对运行时各应用软件的时空资源、系统事件、IO 性能等进行监控, 分析应用开销, 形成集成评估报告。

2.4.2 系统集成优化

通过系统集成评估, 我们可以摸清系统的性能和开销情况, 分析软件架构, 必要时可作为输入对优化提出合理性的建议, 并由平台提供者或应用开发者配合实施。具体表现为:

(1) 资源优化: 应用分区资源分配得过多或过少, 冗余的资源分配(共享内存、端口通道等)的问题可以对资源配置进行合理的优化布局;

(2) 应用或平台优化: 编译优化、驱动调用、任务堆栈或阻塞性操作带来的过渡资源消耗等问题建议应分别提交给应用开发者和平台提供者, 由其分析优化;

(3) 对优化后的集成框架进行迭代验证和测试, 并做优化前后的性能比对, 作为集成评估报告的一部分。

2.5 集成维护

在实际项目中, 各独立软件由平台提供者和应用开发者各自负责维护, 对于系统集成者尤其需要把控系统资源配置的维护。具体表现为:

(1) 关注集成状态: 当有涉及资源配置问题时, 及时汇总分析确认, 修改系统配置文件, 并完成集成的回归验证和测试;

(2) 做好集成受控: 对最终迭代产生的资源配

置文件及二进制映像进行版本控制;

(3) 反馈集成问题: 在过程中有涉及软件问题的应反馈给平台提供者或应用开发者, 及时归零, 形成集成维护记录, 作为集成维护的输出。

3 结束语

本文介绍了 IMA 系统软件集成过程中的集成需求、集成策划、集成设计、集成验证和测试以及集成维护这几个关键环节的主要工作内容, 针对其中需要重点考虑注意的事项进行了分析说明, 借助软件集成开发环境实现更高效的软件集成过程, 推动软件系统集成的正步走模式, 对后续项目提供一定的指导意义。

参 考 文 献

- [1] ARINC. Avionics application software standard interface part 1-required services: ARINC653-1[S]. 2006.
- [2] 中国人民解放军总装备部. 航空电子应用软件接口要求: GJB 5357-2005[S]. 中国人民解放军总装备部, 2005.
- [3] 张明, 周琳. 基于 VxWorks 653 分区操作系统的 IMA 设计与实现[J]. 火力与指挥控制, 2014, (04): 149-152.
- [4] 吴姣, 戴小氏, 张亦姝. 基于天脉 653 操作系统的航空应用软件开发[J]. 航空计算技术, 2017, (05): 77-81.
- [5] 王震, 王念伟, 崔杰. 跨平台 APEX 接口组件的设计与实现[J]. 测控技术, 2018, 37(9): 151-155.
- [6] Committee R S. Integrated modular avionics (IMA) development guidance and certification considerations[M]. 2005.
- [7] IncR. Software considerations in airborne systems and equipment certification: DO-178B[S]. IncR, 1992.
- [8] Wind River. VxWorks 653 configuration and build guide 2.4[M]. Wind River System INC. 2012.
- [9] 王震, 朱剑锋, 洪沛. 基于分布式 IMA 平台的系统健康管理的设计与实现[J]. 航空电子技术, 2016, 47(2): 11-15.