一种综合化航电系统配置数据的生成方法

阮 婷[1] 王 璞[1] 陈 聪[1]
RUAN Ting Wang Pu Chen Cong

摘 要 航空电子系统是现代飞行器的核心关键组成，承担导航、通信、传感及信息处理等重要功能。随着机载任务日趋复杂化，综合化模块化航空电子系统（IMA）凭借灵活可复用、软硬件资源共享、低成本与低功耗等优势，已成为主流架构，可为机载应用提供高安全、高性能的标准化服务支撑。文章通过梳理IMA航电配置数据的核心内容，重点研究配置数据生成技术，提出一种基于中间文件的配置数据生成方法。实验结果表明，所提方法能够有效缩短配置数据生成耗时，优化数据组织结构，显著提升IMA航电配置数据的编制与处理效率。

关键词 IMA；配置数据；生成方法；FACE；中间文件

doi：10.3969/j.issn.1672-9528.2026.05.005

0 引言

航空电子系统作为飞机上所有电子设备和子系统的集合，负责了导航、通信、传感和信息处理等多种功能，是现代飞机的关键组成部分。随着大型飞机所需执行的任务日益增多和复杂，航空电子系统逐渐采用综合化模块化航电系统架构（integrated modular avionics, IMA）。IMA是一组灵活、可重用和支持互操作的软、硬件的共享资源集合，形成了一个综合化的平台。具有低成本、低功耗等显著优势[1-2]。该综合化平台向驻留其上的执行飞机功能的应用提供经过设计和验证、具备预定的安全性级别、满足应用性能需求的服务。随着微电子行业技术的快速发展，IMA实现了多个子系统共享的几种标准模块，这些模块由机载网络相互连接，形成信息处理与交互的统一综合化支撑平台。

航空电子系统经历了独立式、联合式系统架构的阶段，发展为综合化、模块化架构[3]。相对于独立式系统或者联合式系统，IMA具有资源高度共享化的显著特点。子系统的软硬件资源从独立式的专用演变为一组以网络为核心、以支持分区隔离的安全操作系统为基础的，高度共享的IMA系统资源。IMA定义的各个分区应用支持执行不同任务且互不干涉，具备独立的地址空间。为了支持IMA架构上各个分区运行的应用软件开发，国际航空电子应用软件接口标准ARINC653（avionics application software standard interface）[4]定义了多分区的运行环境及一套通用的航空电子应用软件标准接口。对于各个分区的应用软件开发者，需配置符合ARINC653应用的配置数据。同时，资源的高度共享也使得飞机的研制过程更加复杂，要考虑不同的研制需求，加大了系统方进行资源分配管理的难度。各研制方作为不同的角色，由于软件功能和运行平台等原因，对配置数据存在不同需求，导致配置数据的复杂度和数据量都有了较大提高，这给配置数据的开发带来了挑战。综合化资源配置技术作为贯穿IMA系统整个开发和集成过程的重要支撑技术，是航电系统的重要研究对象。张旻等[5]从计算、网络、应用等多个维度进行分析，对IMA系统软硬件资源统一规划问题进行研究。胡军等[6]在文章中提出了从ARINC653系统配置信息到AADL模型元素的映射规则。唐园园等[7]阐述了一种构型管理方案的设计，利用清单文件进行配置匹配性检查。邱宝等[8]提出了一种面向IMA的AADL多范式建模及代码自动生成方法。分别对不同软件构件进行了模型定义，对模型与代码的转换方式进行了分析。王佳明等[9]从民机角度考虑，提出一种符合D0-297标准的RDCU安全关键配置数据生成方法。大多数研究者从应用或总体的角度对综合化资源配置进行分析，或从建模角度对软硬件资源部署进行研究。对于配置数据的生成过程及数据的处理方法未进行深入研究。为了将系统的顶层需求更好、更快地转化为各个模块的所需资源，本文对配置数据的生成方法及数据处理进行了进一步研究。

在早期的航电系统中，配置数据总量不大，使用源代码为核心开发来承载配置数据。但它存在很多缺点：不支持实时的设计约束；配置数据的错误在设计阶段不易发现，直至系统综合阶段，软硬件联调时才能发现，修改代价大；直观性不好，可读性差，不方便进行数据统计、验证、分析；与应用耦合，生成时间过长等。对于百、千行的配置数据量级，

1. 中国航空工业集团公司西安航空计算技术研究所
陕西西安 710065

这些缺点还可以忍受。IMA 的功能应用多，通信频率高且数据量大，系统状态转换复杂，配置数据量级已激增到万甚至十万行源代码。若仍沿用之前的方法，这些缺点会严重影响项目的进度。因此，本文介绍了一种新的综合化航电系统配置数据的生成方法，实现了与应用的解耦合，在设计阶段提供了约束，增加了可读性，且极大地提高了配置数据的生成速率。

1 IMA 配置数据的内容

IMA 配置数据主要给以下三类人员提供：平台和模块提供者、应用开发者、系统集成者。

（1）作为计算机资源供应商，平台和模块提供者实现 IMA 平台的软硬件资源。需要根据各子系统的计算能力、网络流量、存储能力等需求，分配软硬件平台资源。

IMA 的硬件平台包括各种标准功能模块，如信号处理模块、数据处理模块等。平台提供者需要提供硬件资源配置表，对提供的模块进行定义。IMA 需要采集飞机内外传感器的大量数据，对机载网络有一定要求，典型的机载网络有 AFDX、FC、1394 等。其运行依赖于网络配置表，对通信端点配置、端口映射等进行定义。平台和模块提供者还需要提供基础软件资源，对 ARINC653 配置数据等进行定义。

（2）应用开发者在平台和模块提供的服务和约束下开发应用软件。主要定义了应用构型、应用消息、应用网络文件系统配置、ARINC653 配置数据中的应用、分区、端口和内存等配置数据。

（3）系统集成者作为飞机的系统承包商，需要定义和管理系统接口。一般使用接口控制文件（ICD）管理系统接口数据。ICD 描述 IMA 子系统的逻辑结构和子系统间的数据交互。逻辑结构定义了子系统及各个子系统的功能应用。数据交互定义子系统及各个功能应用间的消息通信。系统集成者还需分配和管理系统资源，将资源分配表转化为可加载配置数据表，精确分配资源到各个子系统。

近年来，为了解决航电系统软硬件紧密耦合、复用性差、升级高昂的问题。FACE 文件标准被逐渐使用。该标准是由美国军方、航空工业界和学术界共同推动的开放式架构标准，旨在实现航电系统的模块化、可移植性和互操作性。该文件标准已经得到了广泛的关注 [10]。该标准定义了一套通用的架构方法，将系统划分为不同的单元和服务区，并规定了数据在这些部分之间流动的规则。符合 FACE 标准的配置文件一般为 XML 格式。在该文件中定义了系统的顶层网络通信、功能设计、模块资源等信息。本文分析设计了一个综合化配置数据生成工具，以 FACE 文件作为输入，输出系统综合者、平台和模块提供者、应用开发者实际使用的配置数据。具体输出的配置文件格式及内容应与系统综合者、平台和模块提供者、应用开发者共同商议确定。这三类人员所需的配置数据内容如表 1 所示。IMA 综合化配置数据生成工具的输入输出图如图 1 所示。

表 1 IMA 开发者的配置数据

人员	配置数据
系统综合者	资源分配、软硬件部署、653 调度配置
平台和模块提供者	硬件资源、网络配置、653OS 配置
应用开发者	应用构型、应用消息、653 应用配置

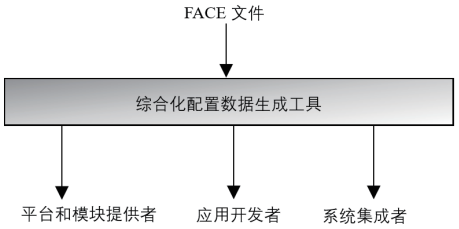

图 1 综合化配置数据生成工具的输入输出图

由于研发过程中需求或资源变化，输入的 FACE 文件会有多次迭代的可能。因此，对综合化配置数据生成效率及正确性提出了要求。

2 IMA 配置数据的生成方法

2.1 IMA 配置数据的形式

IMA 配置数据可总结为三种格式：XML、源文件和二进制格式。其优缺点如表 2 所示。

表 2 三类配置数据格式特点

数据形式	优点	缺点
源文件	访问数据速度快	数据和程序紧耦合；可读性较差
XML 文件	可读性好；数据和程序解耦	访问速度慢；需要 XML 接口库
二进制文件	访问速度较快；数据和程序解耦	不具备可读性

基于上述三种格式的优缺点和项目需要，在与系统综合者、平台和模块提供者、应用开发者沟通后，提出一种阶段式 IMA 配置数据生成方式，将 IMA 配置数据开发划分为两个阶段：开发时配置数据和运行时配置数据。两者关系如图 2 所示。

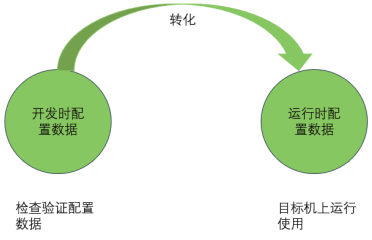

图 2 配置数据阶段示意图

配置数据的来源为 FACE 文件，其中包含了各种人员所需的配置数据信息，需通过综合化配置数据生成工具，将其转换为各种人员所需的符合一定格式的配置文件。这个转换过程分为开发时配置数据和运行时配置数据。其中开发时配置数据主要用于系统综合者修改及检查数据，即确定从 FACE 文件转化到开发时配置数据过程中，数据的正确性。同时需支持对其进行修改，因此需要具备良好的结构化和可读性，采用 XML 格式。可扩展标记语言（extensible markup language, XML）是一种描述型的、表达结构化数据的标记语言。XML 文档由 Schema 文件和 XML 文件构成：Schema 是文档类型定义文件，规定 XML 文件的逻辑结构，包括 XML 文档的元素、元素属性以及元素间关系。XML 文件是具体的结构化数据存储文件。在开发配置数据时，将配置数据验证的规则定义在 Schema 文件中，实现对配置数据的验证和分析。在配置数据不符合预先定义的规则时，及时报错以提示系统综合者进行分析，重新设计 FACE 文件或修正数据。

运行时的配置数据，要具有较快的访问速度且具有一定的可读性。XML 文件的访问速度慢，不利于用户的日常使用。二进制文件具备较快的访问速度，但不具备可读性，不利于用户的修改。因此采用源文件作为运行时的配置数据。

2.2 运行时配置数据生成

开发时配置数据到运行时配置数据转化，需要从 XML 文件转化为 C/C++ 源代码。源文件大小根据软件需求变化，某些大型软件的配置数据源文件可能达到数万甚至数十万行。直接从 XML 文件生成 C/C++ 源代码，涉及大量的字符串处理。系统综合者、平台和模块提供者、应用开发者所需配置文件格式各不相同，涉及不同规则的字符串处理。大量级的源文件可读性较差，出错不利于修改。因此，提出了一种基于中间文件的轻量级的、适应性高的大数据量 C/C++ 源代码文件生成方法，流程图如图 3 所示。先生成各个配置文件的中间文件，再通过生成器将 XML 格式的中间文件转为 C/C++ 源文件。

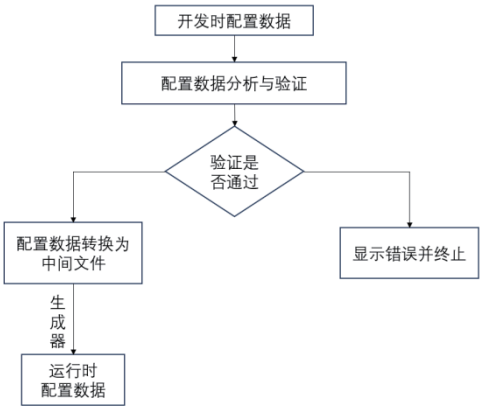

图 3 运行时配置数据生成流程图

中间文件为 XML 格式，有利于用户检查分析。通过生成器将中间文件转化为运行时配置文件，统一了生成规则，减少了对字符串的特殊处理。

2.2.1 分析制定中间文件规范

首先分析运行时配置数据的 C/C++ 源代码文件结构语法以制定中间文件规范。通过统计分析，作为配置数据的 C/C++ 源文件主要包括头文件、变量、数组、宏定义、枚举和注释六类元素。分别根据这六类元素，制定 include、variable、struct 、define、enum 类型的节点。

include 节点描述头文件关系，具有 name 和 type 属性。name 属性定义头文件的名称，type 属性定义头文件类型；variable 节点描述变量或者数组变量，name 属性定义变量名称，type 属性定义变量类型。一个 variable 可包括多个 element 和 composite 节点。一个 element 描述一个简单数据项，一个 composite 描述一个组合类型数据项，可包括多个 element 和 composite 节点；struct 节点描述数据结构类型，其中 type 属性定义数据结构名称。struct 可包括多个 element 子节点。一个 element 描述一个数据项，element 的 name 属性描述数据项名称，type 属性描述数据项类型；define 节点描述宏定义。其中 name 属性定义宏名称，value 属性定义宏值，type 属性定义宏类型，取值范围为 define、ifdef、ifndef；enum 节点描述枚举类型。name 属性定义枚举类型名称，type 属性定义枚举类型宏名称。enum 可包括多个 element 子节点。一个 element 描述一个枚举项，element 的 name 属性描述枚举项名称，value 属性描述枚举项值。include、define、enum、struct、variable、element、composite 都具有 note 属性，可定义注释。除此之外还制定了 line 类型的节点，用于描述一行 C/C++ 源代码，其中 name 属性定义该行代码的所有字符。root 作为根节点，name 属性用于描述 C/C++ 源文件的路径和名字，note 属性定义 C/C++ 源文件文件头注释。

2.2.2 源代码文件生成器

运行时配置数据具有多种类型，包括硬件网络配置、系统管理配置、嵌入式操作系统配置、应用配置等。如果每个类型都单独对应一个从 XML 文件到源代码的转换程序，程序会存在通用性差、代码冗余的问题。生成的源文件也不方便检查。通过中间文件的方法，将不同类型的开发时配置数据都生成 XML 格式的中间文件，这一过程可调用处理 XML 文件的标准库，从而减少对字符串的直接处理。再调用唯一的转换程序生成运行时配置数据。这个转换程序以生成器的方式实现。将生成器遍历节点，根据节点调用相应的处理程序，从 root 节点属性获取目标文件名称和注释头信息；遍历 root 的子节点，分解其类型调用各类子节点的生成程序。其

中子节点的生成程序即通过获取该节点的不同属性值，按照C/C++ 源代码的格式拼成相应的字符串进行输出。其流程如图 4 所示。

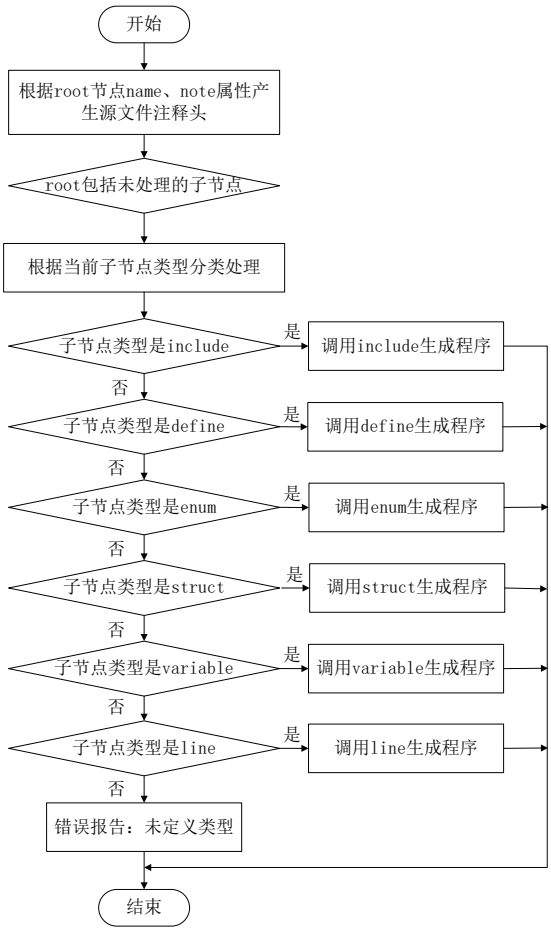

图 4 生成器实现流程图

之前生成运行时配置数据，是通过对每种类型的运行时配置文件分别定义不同的字符串处理程序，调用处理程序拼接生成字符串，再将字符串输出到 C/C++ 源代码文件中。这种方法多次对字符串进行匹配等操作，不仅运行内存大，也带来了时间负担。而提出的基于中间文件的源代码生成方法具有更好的规范性，支持作为数据 C/C++ 源代码文件的全部语法单元类型；生成器模块性结构性好，易于实现和扩展；且通过唯一的生成器进行转化，更易于调试修改。同时，在中间文件生成后，可以对配置数据结构进行整合，存储重复使用的数据，减少生成过程的重复操作，对配置数据生成流程进行进一步的优化。经过对同一项目的配置数据生成时间进行比对，结果显示，新生成方法的时间速率得到了 10 倍的提高。

3 结语

本文介绍了 IMA 航电配置数据的内容并对其生成方法进行了进一步的讨论。将配置数据定义为开发时和运行时两个阶段，把配置数据的验证、分析从项目的集成阶段提前到设计阶段，在设计阶段提供了约束。同时通过中间文件的方式，对配置数据的生成流程进行了优化，增加了可读性，提高了模块的稳定性，更有利于调试修改。根据上述理论和方法，设计优化了 IMA 配置工具软件，有效地提升了开发效率。

参考文献：

[1]Gaska T, Watkins C, Chen Yu. Integrated modular avionics: past, present, and future[J]. IEEE Aerospace and Electronic Systems Magazine, 2015, 30(12): 12-18.

[2] Suo Dajiang, Zhu Jihong, An Jinxia. A new approach to improve safety of reconfiguration in Integrated Modular Avionics[C]//2011 IEEE/AIAA 30th Digital Avionics Systems Conference.Piscataway:IEEE ,2011:16-20.

[3]Wang Hongchun, Niu Wensheng. A review on key technologies of the distributed integrated modular avionics system[J]. International Journal of Wireless Information Networks, 2018, 25: 358-369.

[4] Samolej S. ARINC specification 653 based real-time software engineering[J]. E-Informatica Software Engineering Journal, 2011, 5(1): 39-49.

[5] 张旻，王济乾，田丹 . 综合化航空电子系统资源配置技术研究 [J]. 航空计算技术 ,2024,54(6):102-106.

[6] 胡军，马金晶，袁翔，等 . 一种基于 AADL 的 IMA 系统配置信息的正确性检测方法 [J]. 南京航空航天大学学报，2014, 46(6): 920-930.

[7] 唐园园，杨利宁，李雪源 . 基于 AcoreOS653 的构型管理系统设计 [J]. 电子技术 ,2023,52(1):290-291.

[8] 邱宝，杨志斌，周勇，等 . 面向 IMA 的 AADL 多范式建模及代码自动生成方法 [J]. 小型微型计算机系统，2021, 42(10): 2223-2233.

[9] 王佳明，陈福，吴云 . 民机 RDCU 安全关键配置数据生成技术研究 [J]. 中国新通信 ,2021,23(7):64-65.

[10] 王鹏，曹先泽，张伟，等 . 未来机载能力环境 (FACE) 技术发展综述 [J]. 电讯技术 ,2023,63(8):1268-1276.

【作者简介】

阮婷（1996—），女，陕西商洛人，硕士，工程师，研究方向：机载嵌入式软件。

（收稿日期：2025-09-18 修回日期：2026-05-07）