复杂航电架构的开放式系统标准研究

高 翔[1]，李 辰[2]
（1.中国航空无线电电子研究所，上海 200241;2. 海军驻上海地区航空军事代表室，上海 200233）

[摘 要] 随着越来越多的功能被综合到航空电子系统中，系统的规模日渐庞大且复杂程度也越来越高，这对新平台的设计及现有平台任务能力的扩展提出了新的挑战。然而，目前由于在平台架构设计上缺乏通用化解决方案，以及软硬件专利化导致的重用性受阻，系统的开发和综合成本居高不下，开发周期也较长。因此必须为未来平台任务能力的发展开发出一种先进的复杂系统架构或环境。开放式系统架构是一种最佳的解决方案，它的概念也在不断得到发展。本文以开源组织提出的未来机载能力环境（FACE）架构标准为例，通过深入研究其相关内容和技术，分析了开放式系统架构的未来发展方向。

[关键词] 开放式系统架构(OSA);未来机载能力环境(FACE);通用性;重用性
[中图分类号] V243 [文献标识码] A [DOI 编码] 10.3969/j.issn. 1006-141X.2015.02.06
[文章编号] 1006-141X（2015）02-0026-06

Research on the Open System Standard of Complex Avionics Architecture

GAO Xiang[1], LI Chen[2]
(1.China National Aeromautical Radio Electronics Research Institue, 200241, China; 2. Aeronautical Military Representative Bureau Resident in Shanghai Region for The Naval Force, Shanghai 200233, China)

Abstract: As more and more function has been integrated into avionic system, the scale of the system and its complicacy also increase, which impose new challenge on platform design and enhancement on the existing platform ability. However, the lack of general solution on platform architecture design as well as reusability hindrance caused by software and hardware patent, the overall cost of system development and integration is relatively high, accompanied by a rather long developing period. Thus, an advanced complicated system architecture or environment has to be developed for the future platform task. In this context, open system architecture emerges as the best solution. This paper utilizes Future Airborne Capability Environment (FACE) architecture as the standard, and it analyzes the future development of open system architecture by studying on related content and technologies.

Key words: open system architecture (OSA); future airborne capability environment (FACE); commonality; reusing

1 引言

计算能力和网络带宽等硬件技术的发展将越来越多的功能综合到航空电子系统中，从而导致了系统规模和复杂程度都大大增加，由此带来了系统的开发、验证、优化及维护的巨大挑战。若系统架构特征在设计之初就不合理，如缺乏扩展性、新技术无法注入或软硬件不可移植等，那么要实现一个

Recommend
火龙果软件

复杂且健壮的系统会很难，甚至还要对系统做重大修改。对于关键接口开放式系统架构设计方法采用的是广泛应用、开放的接口标准，并对接口进行全面定义，使得软硬件可移植，系统架构更灵活，同时缩短了系统开发周期，它因此成为系统设计与开发领域研究的一个发展方向。随着对开放式航空电子系统架构标准的深入研究，对开放式系统的概念也赋予了更多的要求，即要求它是一种开发新系统和改进现有系统的商业和技术综合策略。作为一种商业策略，它将打破供应商之间的封锁，鼓励使用多种来源的商用产品以便快速、经济地构建和升级系统；作为一种技术策略，它更关注于一种接口标准开放、模块化、硬件独立化及软件通用化的系统设计。

2 未来机载能力环境（FACE）架构描述

2.1 架构开发背景

军用航空电子系统的开发通常根据特殊的应用需求来采办系统，这就导致了系统基础结构构建机制、功能分区和接口必须都是专用的，软件在不同的硬件平台间移植困难，各项目之间很少出现共享。即使通过重用和采用COTS组件插入产品线的方法，也不能使这种“烟囱式”系统产生规模经济，其最大的通用性也只限于同一领域的产品。随着复杂任务架构和电子系统的出现，将新的硬件或者软件集成到航空电子系统中的成本居高不下，平台任务能力很难得到扩展，开发周期不断延长。

国外提出的所有模块化开放式系统方法都在研究开发一种通用架构，旨在降低航电的开发和集成成本。

军用航电系统的开发是一种软件密集型开发过程。软件第一次构建、测试和验证的能力是一个重要的储备，可以被反复利用。尽管商用软件上使用了“通用操作环境”标准，但因其缺乏严格的安全性需求，所以该标准不适用于军用航空电子领域，而且其软件也很难在不同平台间实现复用。2010年由美国海军航空系统司令部发起、开源组织提出了未来机载能力环境（FACE：Future Airborne Capability Environment）策略。FACE策略是在已安装好硬件的军用航电平台上建立软件通用操作环境，使FACE组件应用在不同平台上可被重新部署，从而实现跨平台的可移植性和重用性。

2.2 架构设计特点

2.2.1 “分段式”架构

早前提出的模块化开放式系统方法关注模块化的、接口定义明确的广泛支持使用的行业接口标准的系统设计，其设计的五大原则是：建立有利的开发环境、采用模块化设计、采用开放式标准、管理关键接口和验证标准的一致性。其中，在关键接口设定方面，模块化开放式系统设计方法并没有控制和管理系统内部和系统之间的所有接口，而是将接口分为关键接口和非关键接口来进行管理，对关键接口采用开放式标准。FACE解决方案扩展了MOSA的设计思想，它将一些基于软件的功能“分段”以组件形式开发。该组件通过定义好的接口向其他组件开放，并且对特定关键接口及“分段”接口之间的差异进行了定义。这些功能段由上至下分为（如图1所示）：操作系统段、I/O服务段、平台特定服务段、传输服务段和可移植组件段。

（1）操作系统段

FACE架构假设了硬件是由第一段操作系统段抽象得到，因此不用直接进行硬件处理。所有FACE分段都是在操作系统段的基础上搭建的，它是所有分段的底层，操作系统段使用适当的处理器控制机制来管理控制其他分段访问的计算平台资源、运行时间和框架接口。操作系统段内的软件代码由供应商提供。

（2）I/O服务段

I/O服务段为没有FACE标准应用程序接口（API）的I/O设备提供了一个通用接口，它用于从操作系统架构驱动向平台特定服务段传输数据。由于I/O接口卡供应商众多，各供应商都提供有不同的接口卡驱动，且各驱动接口和操作系统对应的接口可能都各不相同。为了实现平台特定服务段中FACE组件的重用性和可移植性，同时减少I/O接口卡更新和驱动架构的同步更新带来的影响，FACE技术标准引入了I/O服务段。FACE I/O服务组件使与驱动架构的通信更加抽象化。I/O服务段

Recommend 火龙果软件

的代码采用了适配器设计方案，在转化为符合FACE 标准的接口后，I/O 服务组件可单独连接供应商提供的驱动。由于更新带来的影响全被封装在单个适配器中，因此若发生卡更新或驱动架构更改，则该段中的适配器会随架构驱动器的变化得到更新。适配器的设计方案解决了 FACE 系统中遗留的软件组件兼容问题，它使非标准接口能适应标准接口。若遗留的软件组件在绑定适配器后能改进组件间的通信及与操作系统之间的通信，那么它们仍可保留。I/O 服务段包含了 I/O 管理能力模块，其负责初始化并配置特定的 I/O 服务。

（3）平台特定服务段

平台特定服务段将平台硬件设备数据提供给可移植组件段和 I/O 服务段，还能提供某些设备的专用服务，如初始化、BIT 管理、状态路由、信息节流和滤波等。在传统的软件架构中平台特定软件通常都会与平台特定架构的接口控制文件紧耦合。对于一个特定的外围架构而言，平台特定服务组件的功能可以隔离可移植组件段与平台特定服务段上的组件，并将那些与底层软件架构相连的操作逻辑进行建模或封装。这些服务获取到的数据将被用作平台数据源，从而避免了因外围模型改变而造成的影响。平台特定服务段分为三个子服务段：平台专用设备服务、平台专用通用服务和平台专用图像服务。

（4）传输服务段

传输服务段由组件和服务组成。它提供了兼容多种数据传输方式的服务，使 FACE 组件在无需更改就能在不同传输需求和不同传输能力的平台上支持可移植组件段与平台特定服务段之间的数据传输，还能支持可移植组件段与平台特定服务段之间的数据传输。传输服务为组件之间的数据传输提供了抽象接口，可处理数据的路由、分配和转换，本质上是提供了一种数据搬移的模式，从而使移植组件无需关注远程数据端的位置及部署方式也能实现传输。传输服务段包含了数据分发功能、配置功能、服务质量管理功能、数据转换功能、范例转化功能及信息关联功能。

（5）可移植组件段

可移植组件段是由一组可移植 FACE 组件和通用服务所构成。在该段架构中，由于软件组件和基础结构硬件之间的每一段都设有隔离，将软件与硬件的变化隔离开就可使不同供应商都能提供各段的组件，只要那些组件符合该段的接口要求即可。FACE 标准将可移植性单元定义为一个符合 FACE 接口标准的软件集，这些软件集包括语言运行时间、框架或操作系统适配器，它们在设计和封装上都应考虑到最大可重用性和可移植性。

2.2.2 数据结构

数据结构有助于更有效、灵活地利用和复用架构描述，同时也为决策者和流程所有者提供更广阔的应用空间。FACE 数据结构是一组相关的模型、标准和管理措施，旨在支持通过传输服务接口在可移植单元间的数据传输。FACE 数据结构包括：

一种针对基本元对象机制的元模型和对象约束的数据模型语言；

共享数据模型：由数据模型配置控制板管理的数据模型，可确保新的数据模型的一致性；

通过 FACE 共享数据模型管理方案建立配置控制板主体和运行参数；

可移植单元补充模型：由软件供应商提供的数据模型记录可移植单元交换的消息；

一个语言捆绑协议：描述了传输服务接口上获取的数据模型中的元素如何与编程语言数据结构相映射。

为了通过传输服务接口在可移植单元间进行数据传输，数据模型语言可以捕获在可移植单元间交换的信息。这个结构确保了消息定义，它能够保证系统集成商能够决定不同消息间的消息域转换的语义信息是否合适。除了一般的消息打印，为必要时采取转换，还定义了正式的测量系统和转换机制，这样就不需要可移植单元来进行数据转换，因此也减少了为了支持不同数据集成而修改可移植单元的可能性。

FACE 架构数据支持消息中包含的语义内容的描述，它采纳了共享数据模型机制来实现消息中语义数据的编制。共享数据模型包含了基于配置控制板管理元素和能够由配置控制板或可移植单元供

Recommend 火龙果软件

应商定义的扩展单元。扩展单元中包含了基于配置控制板管理单元的映射，确保了所有独立开发的可移植单元使用的传输服务端消息都能被追溯到通用单元。

每一个应用实现都包括了FACE数据结构中的建模概念、逻辑和可移植单元间数据交换的平台细节。FACE数据结构实现过程依次为：概念到逻辑；逻辑到平台；平台到可移植单元；可移植单元到代码，如图2所示。FACE数据模型在该过程中的任何时候都可以被扩展。最初的共享数据模型很小，但是该数据模型将被每一个新系统不断地重用和扩展。随着它不断趋于成熟，FACE架构数据扩展开发也将逐渐减少。

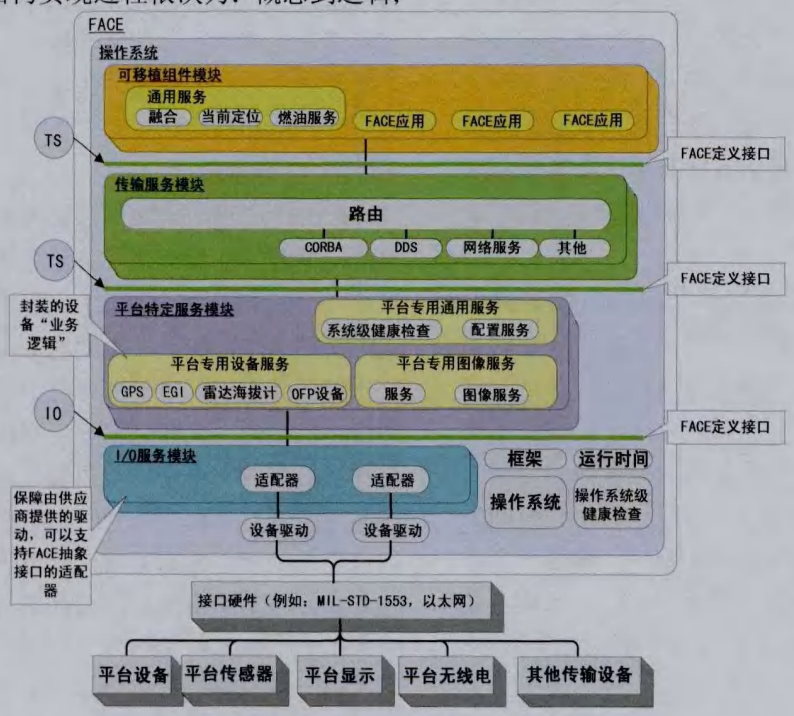

图1 一种典型的FACE架构示例

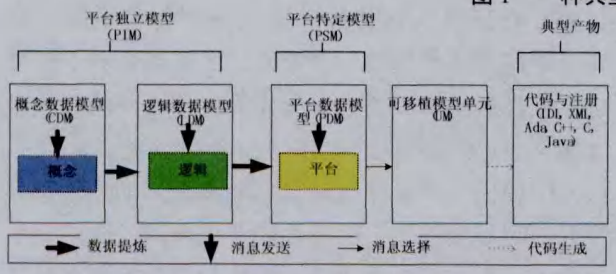

图2 FACE架构数据实现过程

2.2.3 配置服务

重用性通常会较多地出现在平台特定服务段和可移植组件段中。由于FACE在设计上采用了“隐藏”的设计规则，“隐藏”设计规则包含了对系统无影响模块的设计决策，传输服务段对具体的系统组件会隐藏系统的组成和拓扑结构的变化，因此传输服务段的重用性可以采用高度配置的方式平衡各平台的基本路由和传输机制差异来实现。

为了处理航空电子安全领域中不同的应用程序和服务开发需求，在FACE标准中采用了配置文件方法。配置文件定义了可用应用程序的数量和种类。FACE组件支持在编译、构建、初始化和运行时的配置；安全关键分区一般在编译时进行配置；通用分区在运行时进行配置。编译时采用XML文件来配置系统。其中，通用配置文件是最严格的配置文件，它提供了最大的一组操作系统应用程序接口，解决了无需严格安全审批和认证的服务和应用程序。

配置功能块可支持静态配置和动态配置。采取静态配置的话，就需要传输服务段在编译时就进行配置工作，从而在程序运行时使用的代码和资源都能得到优化。采取动态配置则需要传输服务段在初

Recommend 火龙果软件

始时刻就进行配置工作，通过调用 Initialize()函数来实现，并把组件链接的配置文件名传递给传输服务段。

2.2.4 操作系统分区

FACE 架构设计采用在时间或空间上的操作系统分区原则，系统开发人员使用分区能隔离执行功能所需的资源，同时隔离在同一个 FACE 环境下运行的软件组件。操作系统分区可以区分不同的安全级别，能将不同的应用程序分隔开。在空间分区中采用存储分区机制，确保分区能分配到配置好的资源配额。在多个分区上采纳 FACE 解决方案能使架构分段的组件被分隔得更小，可实现这些小组件在不同的分区上的部署。

多个分区的物理存储空间共享通过建立虚拟地址空间来实现。操作系统段控制了每个分区的调度存储及设备的访问，且每个分区都有针对这段共享内存唯一的访问许可。在时间分区中，操作系统段会确保分区能分配到配置好的时间配额。典型的时间分配方法基于 ARINC 653 规范，将执行时间划分为一系列重复出现的分区时间窗口，通过一个或多个分区时间窗口来分配时间。FACE 方案要求操作系统段能支持两个以上配置好的组件在同一分区时间窗口中运行。

3 FACE 架构分析

FACE 架构的设计特点是在计算硬件内建立一个软件环境，而该软件环境能让由 FACE 组件组成的应用在不同平台上部署时对 FACE 应用的影响最小，甚至没有影响。其中，重用性和可移植性则是实现这种系统架构开发的重要性能。下文分析 FACE 设计对开放式系统架构发展做出的贡献以及其目前还存在的一些问题。

1）重用性

FACE 采用的是可移植性单元，通过适配器来解决遗留的软件组件兼容问题，但 FACE 关注的只是软件，包括基础架构（传输机制和操作系统）、应用程序的平台服务以及数据处理应用，它只解决了软件问题，未考虑硬件模块。由于软件重用的内涵既包括在现有项目产品中的重用，也包括在新设计应用或未来应用中的重用，所以 FACE 架构面临着一些应用与之前发布的标准一致、也可能与最新发布的标准不一致的问题。FACE 认为，随着平台或硬件抽象化程度的提高，每段上都将会设置可移植单元，因此重用和可移植性数量都有望得到增加。但在 FACE 演进过程中仍需要解决诸如编程语言版本过时、产品设计和编码不一致、健康监视和容错管理的接口和维护不一致、在不同平台间的应用需求差异等技术问题，并寻求 FACE 架构的进一步分解策略和模块化策略。

2）平台重构性

FACE 通过配置文件来实现平台的重构能力。目前，平台管理和部署任务已经从应用软件和平台设备/服务功能中被分离出，从而无需改变平台部署就可实现重用，能最大限度地利用资源，支持不断变化的任务需求。由于航空电子系统有严格的安全和保密性问题，大部分配置将在构建时进行，因此很难在机载航空电子环境内验证动态重构能力。而在通用配置文件内，FACE 不禁止动态平台重构，但也没明确规定实施方案。此外，由于存在异构系统分解及新架构标准所产生的具体要求，实现平台的重构性仍然具有挑战性。

3）功能分离原则

耦合程度表示架构中单元间的相互依赖程度。计算平台硬件与这些硬件通信的软件模块间的耦合是应用可移植性的一个难点，而紧耦合对于硬件可移植性更是一个难点，因为为支持不同的架构单元需求，应用修改的工作量将是巨大的。因此，FACE 提出了“关注点分离”的概念，即功能必须分开，应用软件与平台硬件的变化隔离。这是一个关键的分离，它能支持软件模块化。如果平台易于配置或分离，对这些平台的修改可以在应用软件中实现，此外应用软件低成本移植可减少平台转换带来的变化。FACE 组件考虑了与系统架构 I/O 和人机接口紧耦合的部分应用，并将这部分应用放在通用服务部分和平台特定服务段内，以此达到 FACE 组件的可移植性。

4）架构建模

FACE 采用了模型驱动架构方法及标准建模语

Recommend
火龙果软件

言来解释其架构。FACE 通过分解式数据建模方式将数据分为概念数据模型、逻辑数据模型和平台数据模型，对数据模型层层递进并逐渐细化。同时，这种建模方式能与基于模型的架构和 DoDAF 元模型紧密结合。最重要的是，这种分解式数据建模方式支持在FACE技术标准中所定义的关键互操作性需求，并且降低了使旧组件符合FACE标准的成本。

5）架构层定义

FACE 在其标准内考虑了基础结构定义和架构定义，并且解决了组件之间的应用程序接口，FACE 除了关注应用软件之间水平的应用程序接口之外，还关注垂直操作环境堆栈中的垂直应用程序接口。FACE 可移植单元规定了需要和提供的信息，但是不固定信息的实际来源和目的地，由平台集成商建立规则。FACE 从根本上采用了更面向消息，甚至面向数据的通信。

6）传输机制

FACE 采用了灵活的传输机制，允许模块开发商根据平台需求选择一种或多种组合的传输机制来实现模块内部通信、内部组件的配置和部署。FACE 还允许系统集成商选择平台相应的传输机制，并在可移植组件的规范中对每个预计的交互（发布-订阅，请求-应答等）都指定了通信协议的类型。

4 开放式系统架构未来的发展

未来开放式系统架构的开发将是一种集技术和商业为一体的综合策略。作为一种技术策略，系统的开发必须顺应技术的变化，能实现未来技术升级和新技术注入。作为一种商业策略，系统应能支持开源的商用产品，能更快和更经济地构建、升级系统和为系统提供支撑。FACE 标准的开发给出了十分有价值的方法和经验，使我们清晰地看到了未来开放式系统架构的发展方向：

1）建立政府或标准化组织可控制的开放式架构

改进商业流程，调整采购体系。通过广泛征求系统集成商、组件开发商和工具开发商的意见，创建可以用于多个产品甚至跨项目的政府或标准化组织组件共享库，每个组件都应采用相似的架构原则、功能和质量要求，并且整个组件采办、技术开发和系统综合过程应融入到开放式系统架构设计中。

2）寻找最灵活的系统架构

无论是通过系统组件功能分解和模块化规则，还是通过软件与处理平台功能分离，或是通过域分析和基础架构来定义需求范围，这些标准都是在寻找更为灵活的系统架构解决方案。然而，若这种方案具有广泛、灵活、多变的实施性，则会降低目标实现的可能性。若降低实现的灵活多变性，则会增加目标实现的可能性，但这涉及的领域将变小。因此，在对架构限制进行决策时灵活性的选择和折衷是至关重要的。

3）实现跨领域应用的可重用性和可移植性

通常可移植性的重点在于操作系统的调用，所以移植组件不仅仅是移植到一个新的操作系统上，而是移植到了一个新的整体环境中，它由操作系统和环境中所有其他组件组合而成，所以可移植性应该支持组件在新平台上的切实可用。此外，在架构设计时还必须考虑可重用组件的开发成本以及重用的经济性。

4）提供认证服务

开放式系统架构的演进是一个较长期的工作，因此标准的研究和制定方除了应组织发布相关技术标准和开发指南（如，白皮书、技术研究及资质认证和测试文档）以外，还应建立健全的认证服务以推进开发式标准在行业内的不断成熟和发展。

5 结论

军用航电系统开发过程呈现了软件密集型特点，其开发成本与复杂性的增加推动着对通用解决方案、应用重用以及开放式系统架构应用的需求发展。FACE 策略旨在实现跨平台通用化软件的应用，减少重复开发。随着 FACE 架构的应用和发展，未来军用平台的发展也将在开放式系统架构下实现更大范围的跨领域、跨不同处理资源和应用平台所需的可重用性、可扩展性和技术快速更新性。

（下转第41页）

Recommend 火龙果软件

（3）数据发送延迟。网络可靠性较差会导致重发数据过多，进而无用数据占用一部分网络带宽，影响数据发送速率。若某一数据块需要重复发送，则后面各数据块必须等待本数据块的所有副本都发送完毕后才能发送，会影响网络响应的实时性。

5 结论

本文分析了座舱显控系统中的运行时通信机制，描述了不同类型的通信方式和通信可靠性保证措施。同时，研究一种基于序号的可靠通信协议，将 ARINC 661 运行时数据封装至数据组和扩展数据块，并按照特定协议执行流程进行通信，在保证通信可靠性的同时，使得网络负载保持在可承受范围。本文重点阐述了该通信协议的执行流程和参数选择原则，对于提高座舱显示控制系统的通信可靠性具有一定的参考价值。

参 考 文 献

[1] ARINC 661-4. Cockpit Display System Interfaces to User System [S]. Aeronautical Radio Inc, 2010.

[2] 高忠杰，康介祥，缪万胜. 基于 ARINC 661 的显控系统设计与实现[J]. 航空电子技术, 2011, 42(4): 12-15.

[3] 刘建，刘勤，孙永荣，曹猛. ARINC 661 规范及其应用开发研究[J]. 计算机与现代化, 2010(4): 188-192.

[收稿日期] 2014-10-24

[作者简介] 李为宇（1986-），男，硕士。研究方向：机载嵌入式软件开发。

王 月（1987-），女，硕士。研究方向：机载嵌入式软件开发。

（上接第31页）

参 考 文 献

[1] Mark HENRY, George VACHULA, Gregary B. PRINCE. A Comparison of Open Architecture Standards for the Development of Complex Military Systems: GRA, FACE,SCA NeXT(4.0)[C]. Military Communication Conference, 2012.

[2] Thomas GASKA. Optimizing an Incremental Modular Open System Approach (MOSA) in Avionics Systems for Balanced Architecture Decisions[C]. 31st Digital Avionics Systems Conference, 2012.

[3] Ronald KOONTZ, Dale JOHNSON. Apache Mission Processor Software Architecture: Future Airborne Capability Environment(FACETM)Considerations[C]. AHS 70th Annual Forum, 2014.

[4] The Open Group. Technical Standard for Future Airborne Capability Environment (FACETM), Edition 2.1[M/OL]. [2014-11-20]. http://www.opengroup.org/bookstore.

[5] The Open Group. Reference Implementation Guide for FACETM Technical Standard, Edition 2.0[M/OL]. [2014-11-20]. http://www.opengroup.org/bookstore.

[收稿日期] 2014-12-21

[作者简介] 高 翔（1982-），女，工程师。研究方向：航空电子技术情报研究。

李 辰（1990-），男，学士。研究方向：航空电子。

Recommend 火龙果软件