文章编号:1674-8190(XXXX)XX-001-14

关于“软件定义飞机”发展的若干思考

孙智孝,温晓玲,艾笑天,韩卓言,姜梦岑
(中国航空工业集团公司沈阳飞机设计研究所,沈阳 110031)

摘　要: 随着航空技术的迅猛发展,传统飞机设计模式正面临深刻变革。现代飞机80%的功能依赖软件实现,硬件平台趋于标准化,而通过软件迭代实现装备能力升级已成为主流趋势,“软件定义”理念应运而生。通过软硬件解耦、功能模块化及服务化架构等特性,飞机的升级与维护更具灵活性和可扩展性,使其能够像汽车和智能手机一样快速响应用户需求,实现定制化能力生成。本文回顾了软件定义飞机的发展历程,并分析了各系统软件定义的特点;从新兴技术创新、设计模式转变及生态体系构建三个维度,结合实践案例,探讨了“软件定义”相关新技术;指出“软件定义”已成为生产方式升级、生产关系变革及新型产业发展的重要引擎,是驱动未来发展的核心力量。未来需进一步释放其创新潜力,突破关键领域瓶颈,助力现代化航空工业体系建设。

关键词: 软件定义;开放式架构;复杂系统工程;软件工厂;智能软件研发
中图分类号: V221　　**文献标识码:** A
DOI: 10.16615/j.cnki.1674-8190.XXXX.XX.01

Some thoughts on the development of "software defined aircraft"

SUN Zhixiao, WEN Xiaoling, AI Xiaotian, HAN Zhuoyan, JIANG Mengcen
(AVIC Shenyang Aircraft Design & Research Institute, Shenyang 110031, China)

Abstract: With the rapid development of aviation technology, traditional aircraft design is undergoing significant transformation. Modern aircraft rely on software for 80% of their functionalities, with hardware platforms becoming standardized, while software iteration drives capability upgrades, giving rise to the concept of "software-defined aircraft". Through features such as hardware-software decoupling, modular functionality, and service-oriented architecture, aircraft upgrades and maintenance have become more flexible and scalable, enabling rapid response to user needs and customized capability generation, akin to automobiles and smartphones. The development of software-defined aircraft is reviewed, and the characteristics of software-defined systems are analyzed. From three dimensions, emerging technological innovation, design paradigm shifts, and ecosystem construction, the related technologies with practical cases are explored. It pointed that the "software-defined" has become a key driver for upgrading production methods, transforming production relations, and fostering new industrial development. Future efforts should focus on unleashing its innovative potential, overcoming bottlenecks, and advancing modern aviation industry development.

Key words: software definition; open architecture; complex systems engineering; software factory; intelligent software development

收稿日期: 2025-03-03;　**修回日期:** 2025-06-22
通信作者: 艾笑天(1994—),男,硕士,工程师。E-mail: 240468702@qq.com
引用格式: 孙智孝,温晓玲,艾笑天,等. 关于"软件定义飞机"发展的若干思考[J]. 航空工程进展, XXXX, XX(XX): 1-14.
SUN Zhixiao, WEN Xiaoling, AI Xiaotian, et al. Some thoughts on the development of "software defined aircraft"[J]. Advances in Aeronautical Science and Engineering, XXXX, XX(XX): 1-14. (in Chinese)

0 引 言

“软件定义”概念自20世纪90年代出现以来，目前已在无线电[1]、卫星[2]、汽车[3]等多个领域得到充分应用，“软件定义”已成趋势。随着科学技术的发展与变化，传统飞机的设计方式面临重大变革，从早期飞机依赖硬件定义功能到软件辅助硬件，再到软件定义功能和系统高度集成后的软件定义飞机能力，软件的重要性与规模与日俱增，“软件定义飞机”的概念也随之应运而生。

本文从“软件定义”的概念与发展出发，介绍“软件定义飞机”的内涵、面临的挑战，并给出所做的思考，展望如何以“软件定义飞机”为契机，在关键领域实现突破，将“软件定义”赋能于飞机设计当中，使其成为未来。

1 软件定义飞机的认识

1.1 软件定义技术发展

“软件定义”概念自20世纪90年代出现以来，目前已在无线电、卫星、汽车等多个领域得到充分应用，大幅提升了行业的生产灵活性、研发响应力，促进了行业的创新发展。

1992年5月，软件定义无线电的概念由美国麦特公司提出。软件定义无线电可以使相同的硬件可以通过“软件定义”来完成不同的功能[1]。

21世纪初，在软件定义无线电基础上发展出软件定义卫星的概念，软件定义卫星使相同的载荷硬件设备实现不同的功能[2]。以天基超算为基础的开放架构的智能卫星系统，拥有丰富的星上应用软件，可多用户共享使用。该技术可极大提升通信卫星有效载荷的灵活性。一方面，改变传统地球静止轨道通信卫星在工作期内无法及时更新通信技术的劣势；另一方面，软件定义卫星将改变目前的通信卫星制造模式，制造商不必在获得订单之后再进行制造，可按照一定的规格实现通信卫星的批量制造和预先制造，用户可在卫星发射入轨后进行能力配置，从而缩短卫星研制周期并降低研制成本。同时用户也可根据业务需要和卫星的健康状态，最大程度地利用卫星能力，实现卫星制造商和用户的“双赢”。

2012年，特斯拉Model系列完成了首次整车OTA(Over-The-Air)，软件定义汽车的概念开始萌芽。近年来，汽车生产从基于硬件的产品向以软件为中心不断转变，为汽车行业提供新的机遇并加速多个功能领域的创新。2020年12月21日，中国成立软件定义汽车委员会，目标是协同汽车产业制造单位，共同定义软硬件标准化接口，降低研发复杂度，加速发展[3]。

综上所述，“软件定义”的核心是从面向专用功能定制化软件设计，转向更小单元的标准化可复用软件设计，使软件可面向需求快速迭代升级。

1.2 软件定义飞机

软件定义飞机是一种以软件为核心的设计理念，软件定义飞机示意图如图1所示，在硬件一次到位的情况下，机载软件持续开发、部署与集成，目的是通过强大的软件系统来实现飞机功能的灵活配置和优化，从而提升飞机功能、性能、安全性等指标，使飞机在全生命周期中能够更智能、更高效地适应不同任务需求。

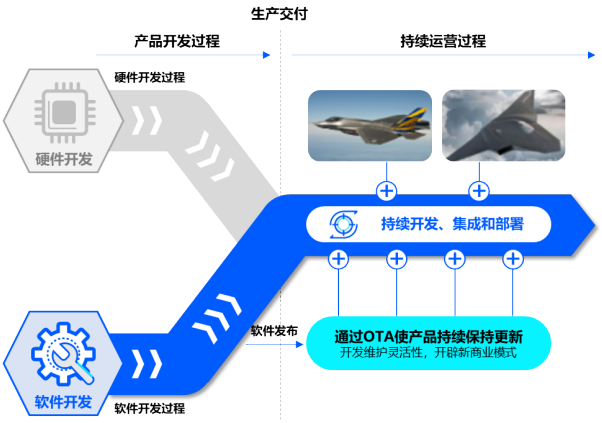

图1 软件定义飞机示意图

Fig. 1 Schematic diagram of software-defined aircraft

软件定义飞机应具备以下特性：

1）模块化：飞机的硬件和软件分离，允许软件独立于硬件进行开发和升级，模块化结构使得每个模块可以独立开发和替换，满足不同任务需求，增强飞机的适应性和更新速度；

2）开放性：通过开放式架构，系统能够整合不同厂商开发的模块，确保互操作性。这种架构允许灵活的模块替换和更新，支持多样化的功能集成，降低了研发成本并促进了创新；

3）自动化：通过自动化测试和部署流程，开发团队快速响应需求和技术变化，确保软件始终保持在前沿。同时可快速修复漏洞并优化性能，增

强系统的稳定性和安全性。

4) 智能互联:智能互联使飞机能够实时共享数据,与其他系统和设备协同工作,提高作战效率和决策能力。

1.3　软件定义飞机的发展历程

软件定义飞机是航空领域的重大技术变革,总结来看,从早期飞机依赖硬件性能,到未来软件定义制胜力,软件定义飞机共经历了5个主要发展阶段。

早期萌芽阶段:该时期的主要特征可以概括为硬件定义功能,即核心关注硬件,软件概念尚在萌芽,飞机的功能都是由硬件实现。

初步发展阶段:该时期的主要特征可以概括为软件辅助硬件,这个阶段的飞机逐步呈现出电子系统复杂化的特征,软件定义作战效能初显。此阶段飞机已开始装载应用软件,代码行数一般在80万行左右。

快速发展阶段:该时期的主要特征可以概括为软件定义功能,即硬件功能软件化,软件可以提升飞机性能,这个时期的飞机软件代码行数为600~800万行。

高度集成阶段:该时期的主要特征可以概括为软件定义能力,通过软硬件解耦,软件架构走向服务化,该阶段的飞机软件代码规模已达到2000万行。

智能开放阶段:该时期的主要特征可以概括为软件定义智能,即软件智能化,助力飞机效能提升。飞机利用软件定义智能水平,实现算法赋能与自演进,并且可以快速增智,该阶段的飞机软件代码规模预计可达到5000万行。

1.4　飞机各系统软件的特点

飞机作为复杂的工业产品,各系统软件特点差异大,一般分为航电系统、飞控系统、机电系统和指控系统。

航电系统:航电系统是典型的复杂巨系统,软件配置项数量多、功能逻辑复杂、升级改动频繁。对数据处理、任务处理能力要求高,一般系统架构多采用平台、中间件技术,且架构复杂,多为分层结构[4]。所谓一代飞机一代航电,纵观世界飞机发展史,航电系统发展变化巨大。早期飞机采用分离式航电架构,设备功能独立,部分设备通过串口进行简单信息交互,部分设备有了专用的用汇编语言编制的功能软件。这样的软件所发挥的作用就是辅助硬件功能实现,其对飞机性能影响小,后续飞机航电架构又经历了联合式架构,综合化架构,先进综合化架构等发展历程,未来机型航电系统架构的重要方向之一是开放式架构[5],同时软件功能及接口走向服务化、标准化[6]。

飞控系统:飞控系统具有多余度、强实时、高安全、高可靠的特点,复杂性略低于航电系统,软件规模一般,但是对安全性和实时性要求极高[7]。飞控系统一般采用分布式系统架构,软件架构强实时分时分区,考虑安全性及效率,一般不采用复杂平台技术。随着系统功能复杂度提高,飞控系统从机械式、模拟式走向数字式,软件研发流程也从瀑布式向敏捷式演进,随着代码自动生成、自动测试[8-10]、持续集成等方法的应用,飞控软件研发周期大幅缩短,软件质量明显提升,软件缺陷进一步降低,飞控系统逐步向自主化、智能化发展。软件定义飞控系统、保证飞控安全、提高智能化水平,已成为发展主流。

机电系统:机电系统是典型的油、液、气、电、热多物理域特性混合的安全关键系统,软件规模较小,对于安全性与实时性的要求仅次于飞控系统,同样不会采用复杂平台技术[11]。一般采用综合化管理系统架构,以逻辑解算为主,软件架构较简单。随着机电系统技术发展,未来通过分布式控制和综合化管理,将极大改善机电系统的任务可靠性和维护保障特性,进而减轻空地勤监控操作压力[12]。

2　“软件定义飞机”发展思考

随着飞机的不断发展,目前各机载软件系统已经演进成复杂、规模化的嵌入式软件巨系统,如何解决传统模式带来的开发慢、验证慢、部署慢、管理难等问题已成为机载软件研发方探索的关键问题,美军提出了“敏捷压倒一切”的战略方向并积极探索“软件定义飞机”的新模式,目前已经吸收商用软件公司成功的工具与方法,建设18家空军软件工厂,同时整合现有资源建立面向整个国防部的软件工厂生态系统,以实现“以相匹配的速度交付弹性软件功能”这一愿景。

同样为应对机载软件现如今面临的软件规模大，系统复杂度高，专业特点差异大，功能扩展难，部署升级慢分等挑战，本文从新兴技术创新、设计模式转变和生态体系建立3个维度进行思考以破解机载软件难题。从新兴技术创新角度，思考了如何将应用开放式系统架构、软件OTA升级技术、智能应用上装飞机、虚拟仿真验证等方面技术应用于机载软件开发的过程中，进而思考如何进行设计模式的转变，将其他领域的新兴设计模式应用于机载软件生命周期中，包括：采用复杂系统工程[13-14]、软件工厂研发模式[15-16]、复杂系统权衡设计、智能辅助软件研发等新型研发模式，思考了如何建立软件工厂生态与跨行业生态移植等软件协同生态，通过以上思考探索实现软件从需求到部署的敏捷循环，从而实现软件定义飞机的途径。

2.1 新兴技术创新

关于将新兴技术应用于机载软件的思考，主要包括应用开放式架构，采用软件OTA升级，智能应用上装以及虚拟仿真验证等技术。

2.1.1 开放式架构

以往飞机大多采用单体式封闭架构，软硬一体化开发，从“软件定义飞机”的角度出发，可以尝试采用开放式架构，以实现“功能＋服务＋资源”的解耦，解决机载软件复杂度高，功能扩展难等问题，尝试实现敏捷能力生成，跨平台灵活接入和在线快速升级的需求[17]。

2.1.1.1 开放式架构概念

开放式架构是一种系统设计理念，强调不同组件和服务的相互操作性和灵活性。这种架构的核心在于模块化和标准化，使得系统能够轻松集成来自不同供应商的产品，从而满足不断变化的业务需求和技术进步[18]。

开放式架构的优势在于它能够降低系统开发和维护的复杂性。开发者可以在不影响整体系统的情况下，替换或升级单个组件。这种特性对于快速变化的技术环境尤为重要，因为它允许企业迅速响应市场需求和技术创新。例如，在软件开发中，微服务架构就是一种开放式架构的体现，允许各个服务独立部署和扩展，从而提升了系统的灵活性和可维护性。

开放式架构还为开发者提供了更大的创新空间。由于平台是开放的，开发者可以根据用户的需求，快速开发和发布新的功能。然而，开放式架构也面临一些挑战。由于涉及多个供应商和技术标准，确保设备之间的兼容性和安全性是一项复杂的任务。此外，开放式架构可能会导致系统的复杂性增加，管理和维护的成本也相应上升。因此，在设计和实施开放式架构时，需要综合考虑这些因素，以确保系统的稳定性和安全性[19]。

开放式架构为技术的进步和业务的灵活性提供了强有力的支持。对于飞机软件的开发而言，如果采用开放式软件架构，开展服务化设计，可以实现敏捷能力生成，跨平台灵活接入和在线快速升级的需求。

未来在机载软件上借助硬件资源虚拟化、软件应用服务化以及功能服务编排三种手段进行功能、服务、资源的解耦更好的实现开放式架构。

2.1.1.2 硬件资源虚拟化

硬件资源虚拟化是开放式架构应用落地的典型手段，通过容器技术实现硬件资源虚拟化，在原有软件架构的基础上，添加容器引擎、容器管理平台，同时将相关应用软件操作系统进行适配工作，实现对各硬件模块资源的统一管理，具有利用率高，硬件资源灵活可配的优点[20]。

硬件资源虚拟化通过容器技术实现，容器提供与宿主操作系统完全一致的能力视图，本质上是一种应用打包方式。面向多层解耦设计思想快速实现多种容器运行时和容器公共服务层的组合，采用CRI、CNI和OCI标准符合性设计，具备充分的技术兼容性（兼容K8S、K3S）[21]。车载系统的硬件资源虚拟化示意图如图2所示，未来机载系统可以参考车载系统，研究基于嵌入式实时操作系统的容器技术，结合DevOps工具环境实现安全、自动化、高效率的应用发布与机上部署[22-23]。

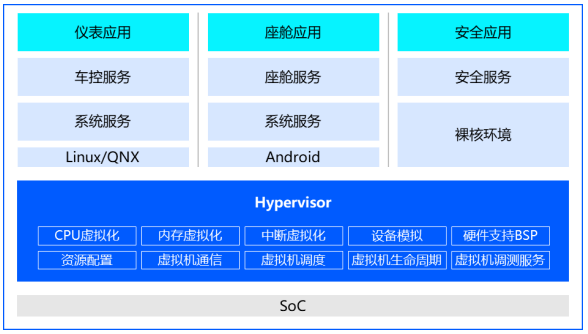

图2 车载系统硬件资源虚拟化示意图

Fig. 2 Schematic diagram of hardware resource virtualization for on-board systems

2.1.1.3　软件应用服务化

SOA(Service-Oriented Architecture)服务化架构是一种设计理念,旨在通过将应用程序分解为可重用的服务,提升系统的灵活性和可扩展性[24]。在这一架构中,每个服务都是一个独立的功能模块,通过标准化的协议进行通信,软件应用服务化示意图如图3所示,多个服务组件可以跟据使用者的编排来实现流程重组。这种方法允许不同的系统和应用程序之间进行高效的集成,同时增强了业务逻辑的独立性。

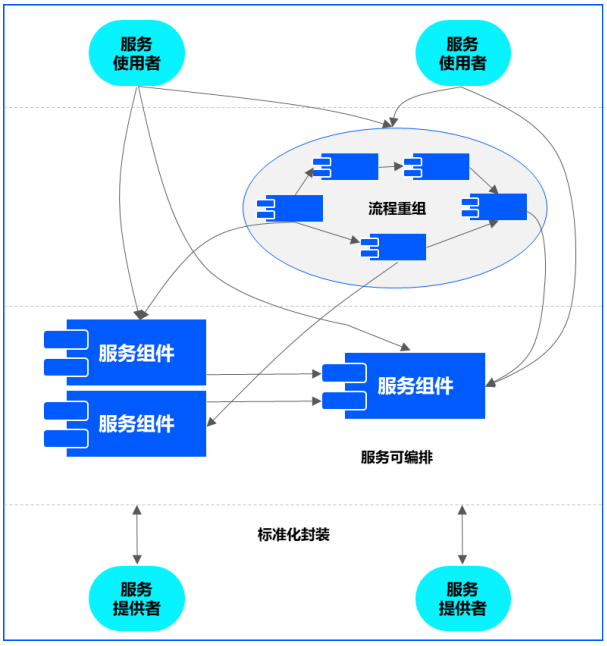

图3　软件应用服务化示意图

Fig. 3　Schematic diagram of software application servitization

在服务化架构中,服务通常具有明确的接口和契约,能够独立部署和更新。这意味着开发团队可以在不影响整体系统的情况下,更新某一特定服务。此外,服务化架构支持多种技术栈的共存,使得组织能够根据具体需求选择最适合的技术解决方案。

此外,服务化架构也为微服务的实施奠定了基础,微服务是一种进一步细分服务的架构风格,强调服务的单一责任和极高的独立性[25]。然而,服务化架构并非没有挑战。服务之间的通信管理、数据一致性、服务监控和治理都是需要关注的重要问题。为了有效地管理这些挑战,组织需要引入API管理、服务发现和负载均衡等技术。

对于未来航空器,服务化架构可以应用于软件设计当中,可以对功能进行模块化拆分,实现功能标准化分类和定义,使用服务化运行架构实现机内/机间服务发现、注册、调用等功能,可以大幅提升系统运行效率。通过定义系统数据字典模型体系和功能统一接口标准,可以实现全机与多机应用/服务数据共享。采用组件化/容器化技术实现应用程序封装,支持机上软件平台间快速移植与OTA敏捷部署。

2.1.1.4　功能服务编排

功能服务编排从注重实现功能的产品间时序/信息交互的面向过程设计转变为注重产品内可提供服务能力的面向对象设计,优化产品服务间边界,分层次提升功能间关联关系的隔离度。

通过服务治理框架根据用户使用需求进行服务编排和动态部署,能够实现能力环的快速编排与精确闭合。通过隔离虚拟化操作系统、数据传输中间件及平台级系统管理能力,为应用软件提供标准化服务化运行时环境。软件的服务化封装示意图如图4所示,非服务化软件外部进行服务接口封装,以形成适合服务化框架调用的组件。

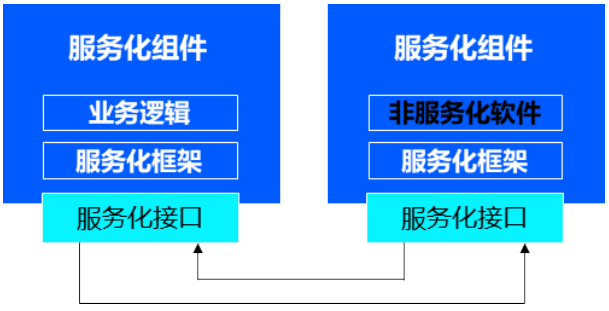

图4　软件的服务化封装示意图

Fig. 4　Service package diagram of software

抽象设备服务软件示意图如图5所示,开发抽象设备服务软件对非标准应用进行消息转换处理形成可被服务化组件识别的调用机制。抽象设备服务软件通过FACE TSS接口或FACE IO接口与非标准应用进行数据交互,并通过服务化接口与其它服务化组件交互[26]。

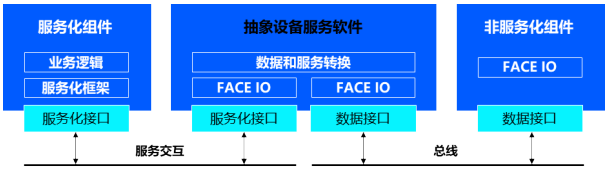

图5　抽象设备服务软件示意图

Fig. 5　Schematic diagram of abstract device service software

在未来飞机软件设计时开展原子服务操作，进行服务化工具开发，可以实现机载软件的敏捷研发，实现即插即用快速升级，并增加飞机能力的动态生成。

2.1.2 软件OTA升级

过去的飞机往往采用物理方式进行接线电缆升级，时间周期长，人力成本高，在“软件定义飞机”的背景下，可以采用OTA（Over-The-Air）技术对飞机进行升级，OTA技术为通过移动通信的空中接口实现对系统和应用程序进行远程管理的技术[27]。对于飞机来说，可以通过5G基站等方式实现从软件研发中心到飞机的无线升级，缩短软件研发到部署的时间，并为飞机提供多种升级途径与选择，解决“软件定义飞机”目前面临软件部署升级慢的困境，并给予飞机多环境下的服务保障。机载软件OTA示意图如图6所示.

图6 机载软件OTA示意图

Fig. 6 Schematic diagram of airborne software OTA

软件OTA升级技术在物联网（IoT）、智能手机、汽车电子等领域已经得到了广泛应用，旨在提高设备的性能、修复漏洞和提供新功能[28]。2000，日本手机厂商通过OTA技术对手机上的移动数据及应用进行远程管理。21世纪初，本田等车企开始利用OTA技术对汽车进行更新，提供远程车况查询、远程控制车门等服务，2012年，Model S上市特斯拉实现整车OTA应用，5年时间内，通过25次大版本OTA升级，实现了人机交互、动力和电池系统等功能升级[29]。OTA升级的优势主要体现在便利性和效率上，用户无需手动操作，升级过程可以在后台自动完成。

OTA升级的基本流程包括以下几个步骤：

升级包准备：开发团队在服务器上准备好包含新软件版本的升级包。该包通常包括完整的系统镜像或增量更新，以减少下载时间和带宽消耗；

设备检测：设备定期或实时检测可用的更新。通常，设备会向服务器发送请求，检查新版本的可用性；

下载升级包：一旦检测到更新，设备将从服务器下载升级包。在此过程中，可能会进行一些验证，以确保下载内容的完整性和安全性；

安装升级包：下载完成后，设备会在合适的时间点（如空闲状态）启动安装过程。此过程可能需要重启设备；

升级验证：安装完成后，设备会进行验证，以确认新软件正常运行。如果升级失败，设备能够自动回滚到之前的版本。

特斯拉是OTA升级技术应用的一个典型案例。自2015年以来，特斯拉开始通过OTA技术为其电动汽车提供软件更新。用户无需前往服务中心，便可通过Wi-Fi直接从特斯拉的服务器下载最新的软件版本。这些更新不仅包括安全补丁和bug修复，还常常添加新功能，例如提升驾驶体验和增强自动驾驶能力[30]。

特斯拉的OTA系统能够实时收集车辆的数据，分析用户的使用情况，以便更好地定制更新内容。此外，特斯拉还通过OTA技术在电池管理和能量使用方面进行优化，帮助用户提高续航里程。特斯拉在全球范围内的成功应用证明了OTA升级技术的潜力，不仅提升了用户体验，也减少了维修和维护成本。

将OTA技术应用于未来飞机的保障等场景，在地面借助无线无线通信网络实现批量飞机软件无线升级、数据加载，空中借助数据链路实现软件快速升级。从而提升飞机服务保障能力与用户需求响应能力。但同样也面临一些挑战。安全性是主要问题之一，攻击者可能会利用漏洞进行恶意软件植入或数据窃取。因此，加强数据加密和身份验证机制是确保OTA升级安全的关键[31]。此外，用户的网络环境、设备兼容性等也是影响OTA成功的因素。未来，随着5G等通信技术的发展，OTA升级的速度和效率将进一步提升，应用场景也将更加丰富。人工智能的结合将使得升级过程更智能化，可以根据用户行为预测和优化升级内容，从而实现个性化服务，成为软件定义飞机的基础。

2.1.3 智能应用上装

随着人工智能的发展，飞机也向智能化逐步

演进,飞机上上装了越来越多的智能软件辅助飞行员进行决策,智能软件相比传统软件设计数据、训练与学习,具有显著的成长性,将智能软件在飞机上应用上装,可使飞机具有更高的自主性与协同性,并且获得远超传统飞机的感知能力、认知能力以及行动能力。

以智能驾驶软件为例,智能软件存在“端到端”和“模块化”两种开发模式[32],“模块化”与“端到端”智能驾驶软件对比图如图7所示。

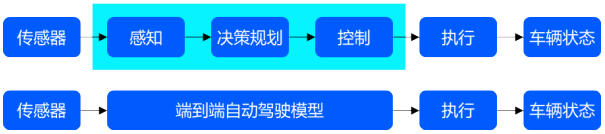

图7　“模块化”与“端到端”智能驾驶软件对比图

Fig. 7　Comparison of "modular" and "end-to-end" intelligent driving software

模块化的定义是将智能任务分解为感知、预测、规划和控制等多个独立模块,通过系统集成实现自动驾驶功能。其优势在于可以简化任务,且具备高可解释性。但是其挑战包括代码量大;泛化能力不足;级联误差多[33]。

端到端的定义是通过深度神经网络模型实现从感知到控制的完全映射[34]。这种模式的优势在于能够减少误差传递;集中资源;灵活性高。但是挑战包括对数据的需求量大;算力要求高;黑盒子问题。

因此,目前量产的智能驾驶软件大多采用模块化结构,但端到端的架构因其高效性和灵活性逐渐成为行业关注的热点。以CNN、Transformer为代表的算法升级是迭代的核心驱动力。无论采用哪种研发模式,智能软件相比于传统软件都涉及数据、学习和训练,具有显著的成长性。

以“特斯拉”端到端的智能驾驶软件为例,智能软件研发涉及数据采集、算法研发和模型训练三个环节。通过多路摄像头、GPS、温度传感器等设备,覆盖不同场景、天气和道路条件进行数据收集,并通过后台将数据传到云端服务器,同时使用影子模式收集边界数据,增加多样性;通过人工标注、自动标注和数据仿真三个途径[35],进行数据标注工作与工程师紧密结合保证质量。

在算法方面以BEV+Transformer核心基础:提供不依赖高精地图的感知和定位能力;通过HydraNets进行视觉感知,俗称“九头蛇网络”,共享主干网络,通过分支成多个头,同时输出多个预测张量,每个头负责不同的任务或对象类别,如信号灯检测、行人检测等;最后通过Vector Space进行路径规划,迅速产生最优解,约束迭代,轻量生成网络辅助规划。

在算力方面使用Dojo高性能计算系统,用于处理和计算自动驾驶系统产生的海量数据。Dojo最小组成单元是“Training Node”,每个节点负责一部分计算,多个节点组合一起构成超算平台[36]。

相比汽车,对于飞机来说,智能软件应用上装存在训练样本稀缺、实装迁移难,仿真天地不一致的问题[37],且对于上装来说技术的难点在于智能算法如何适配机载环境,未来飞行器的机载智能软件跨平台适配可以通过研发面向深度学习的嵌入式智能计算框架和设计面向机载嵌入式的AI芯片的异构并行架构来实现,研发面向深度学习的嵌入式智能计算框架,基于FPGA的高效能可配置算子库并适配,实现软硬协同优化和高效推理;并考虑设计面向机载嵌入式的AI芯片的异构并行架构,对算法结构进行优化,在运行精度和实时性进行测试,充分利用AI芯片大算力和资源,同时降低AI芯片运行对软件生态的依赖[38]。

2.1.4　虚拟仿真验证

虚拟仿真验证是将虚拟仿真技术应用于软件测试于集成验证试验中,将物理资源抽象为逻辑资源,关联多系统,搭建“持续集成+全数字仿真”平台,将此技术用于机载软件的测试中,可以解决传统飞机由于硬件资源受限导致的集成难,缺陷多,发现晚的问题,并缩短软件研制周期。

虚拟仿真技术(Virtualization)是一种将物理资源(计算机CPU、存储、网络等)抽象为逻辑资源的技术手段。虚拟资源不受现有架设方式、地域或物理配置所限制,在航空航天、汽车、机器人等领域被广泛采用,在汽车领域虚拟仿真技术的实现示意图如图8所示,汽车软件基于数字孪生/vECU(虚拟电子控制单元)进行测试验证方法以在业内共识[39],ETAS,SYNOPSYS、DSPACE等工具厂商可提供完整商用数字仿真模型开发平台。其允许研究人员和工程师在虚拟环境中测试和验证设计方案,从而降低成本和风险。该技术结合了计算机图形学、物理建模、人工智能等多个学科的理论和方法,能够实现高精度、高效率的模拟。

图 8 汽车软件虚拟仿真技术示意图

Fig. 8 Automotive software virtual simulation technology schematic

虚拟仿真验证技术的核心在于建立准确的模型，这些模型需要反映实际系统的物理特性和动态行为。例如，在航空航天领域，飞行器的飞行性能验证可以通过虚拟仿真技术进行，工程师可以在设计阶段对不同飞行状态下的飞行器表现进行测试，避免昂贵的实机试飞。

一个典型案例是 NASA 的“阿尔忒弥斯”计划[40]。在这一计划中，NASA 利用虚拟仿真验证技术进行火箭发射和载人登月的各项实验。通过高保真的模拟环境，工程师可以预测火箭在不同条件下的飞行轨迹、结构强度和系统性能。在项目初期，NASA 开发了名为“可视化飞行仿真系统”(VFS)，这个系统不仅能模拟火箭的动力学，还能重现复杂的外部环境影响，如气象变化和地球引力等。

在这个案例中，虚拟仿真验证技术显著提高了设计效率，降低了测试成本[41]，使得 NASA 能够在实际发射前发现潜在问题并进行调整。例如，在一次模拟中，发现了火箭在特定高度会遭遇不利气流的情况，工程师及时修改了设计方案，从而确保了实际发射的安全性和成功率。

此外，虚拟仿真验证技术的应用也促进了团队的协作。不同领域的专家可以在同一个虚拟环境中进行实时互动，快速交流意见，进一步优化设计。这种跨学科的合作方式，为复杂系统的设计提供了更多的视角和解决方案。

对于航空领域来说，未来基于仿真数字孪生的软件运行平台进行飞机软件集成，随开发随集成，解耦软硬件研制，实现软件测试左移，缩短研制周期、降低更改成本[42]。随着计算能力的提升和算法的进步，虚拟仿真技术将会在飞机领域中发挥重要作用。

2.2 设计模式转变

“软件定义飞机”在设计模式转变的创新手段的思考，主要包括复杂系统工程与软件工厂研发模式。

2.2.1 复杂系统工程

先进航空武器装备系统复杂性与涌现性呈非线性方式增长，90% 的装备功能由软件实现，传统飞机在过往研发时往往采用零散式开发，未来在飞机研发时，可以贯彻复杂系统工程思想进行复杂系统权衡设计，如图 9 所示，从需求捕获、功能分析、架构权衡、设计定义到试验验证的全流程贯通，为复杂机载系统软件的设计与开发提供了完整输入与方法支撑，探索解决“软件定义飞机”目前系统复杂度高，软件规模大带来的软件缺陷多、功能扩展难，变更处理慢等问题的方法[43-44]。

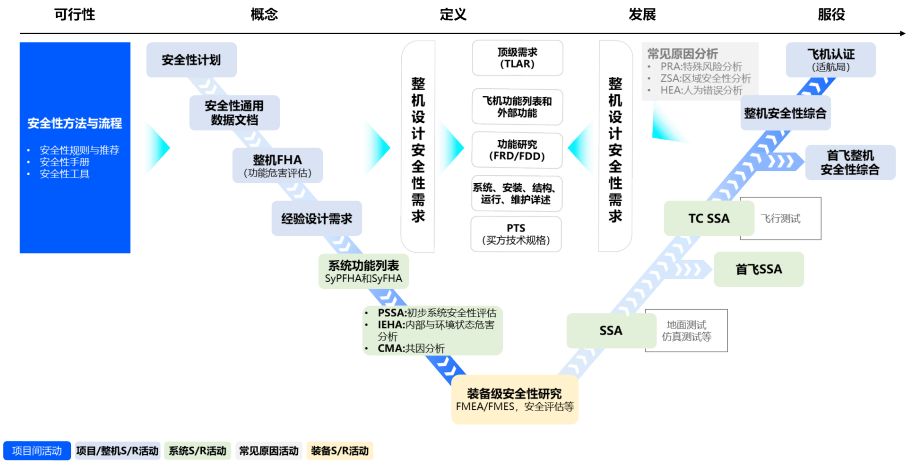

图 9　机载软件系统工程“V”字图

Fig. 9　Airborne software system engineering "V" diagram

在系统需求开发阶段,可以尝试通过用例归类能力需求,具体建模定义系统需求,开展功能逻辑快速迭代验证,实现模型到需求的映射,完成基于模型的系统需求精细化、系统性开发,最终达到的效果是面向实战和全生命周期,构建用例,建立 SysML 模型[45]。

在架构设计阶段,可以尝试开展多要素架构设计权衡。一般构建复杂系统时需要综合考虑业务领域的需求和系统的质量属性,领域驱动设计有助于深入理解业务领域,定义业务边界和核心概念,而属性驱动设计关注系统的非功能型需求,帮助定义和测量系统的质量属性。通过结合着两种方法,架构设计可以更全面地考虑业务和技术方面的需求,从而建立更健壮、可扩展和高质量的系统。通过权衡设计,建立关重要素选取到功能、逻辑再到物理架构的系统架构设计流程,强化基于模型的多架构权衡分析,实现“需求-功能-逻辑-物理”的架构映射。在无人机上设计中开展多要素的权衡架构设计,可以减轻飞机重量与功耗,并可节约单机成本。

在功能逻辑模型的基础上,可以进一步开展跨学科、多物理域联合建模仿真,分析飞机关键技术和关键性能指标。并将功能逻辑模型作为软件开发的输入进行传递,代替软件研制任务书,基于模型开展软件代码自动生成[46],实现软件研发周期大幅缩短,并且减少软件缺陷。

对于未来飞行器上人机交互界面的可视化建模,未来可通过多画面模型联动调度[47],多外部接口数据的数据驱动仿真,模拟真实机上状态,建立人机交互的“设计-开发-仿真”一体化敏捷研制流程,完成自闭环功能逻辑,实现软件设计及开发的统一化、平台化、模块化、标准化及可视化,缩短软件研发周期,提高不同机型的软件移植率。

最后通过构建模型驱动的集成验证体系与环境,从概念验证到试验鉴定,逐渐由模型过渡到实物,实现建造前验证,减少设计返工。

2.2.2　软件工厂研发模式

软件工厂研发模式是一种高效的软件开发方法论,与传统的小作坊式分包工作不同,软件工厂旨在通过标准化和模块化的方式提升软件开发的效率和质量。这种模式受到制造业“工厂”理念的启发,强调生产线式的开发流程[48],通过将软件开发过程视为一个可重复的生产过程,从而实现规模化生产。将软件工厂研发模式应用于机载软件的研发中,可以缩短软件研发周期,减少软件缺陷,使得软件变更速度加快,并提供更好的可维护性与拓展性。

软件工厂的核心在于将开发过程拆解为多个可重复的模块,每个模块都有明确的功能和接口。这种模块化设计允许团队在不同项目中复用现有的模块,减少重复开发的工作量,提高开发速度。同时,标准化的流程确保了每个模块的质量,减少了因人员差异导致的产品不一致性。

软件工厂研发模式通常依赖于几个关键技术:

模型驱动开发(MDD):通过高层次模型描述系统,生成代码,提高开发效率。

持续集成(CI):自动化构建和测试,确保每次

提交的代码都能快速集成,减少集成时的问题。

微服务架构:将系统拆分为多个独立的服务,便于独立开发、测试和部署。

美军的“凯赛尔航线”软件工厂采用了与用户持续交流的模式,组建了研发-用户联合团队,明确应用程序和工作流程的交互关系。凯塞尔航线建设软件项目的初期目标不是提供原型,而是建设用户可以测试并用于支持作战的最小可行产品(MVP),之后的让用户参与测试,以启动开发、测试、学习的循环迭代,建立用户持续参与的机制[49]。“凯赛尔航线”采用自动化的持续集成和持续部署管道技术。平均每3.3 h通过管道部署一次代码。从开发团队准备部署起,平均只需要8.0 h,就能完成开发环境到生产环境的部署,并采取持续跟踪、持续响应的模式,不断跟踪生产环境中的应用程序,并解决使用中遇到的问题。

为提升市场竞争力,互联网公司等企业融合自身研发实践和管理理念,提出了研发数字化解决方案,应用基于软件工厂的研发模式。针对业务需求多且繁杂、业务需求时效要求高等痛点问题,华为(CodeArts平台)、阿里(云效平台)、腾讯(Coding平台)等公司采用DevSecûps等先进研发思想,提出了各自的研发数字化转型解决方案,着力实现研发流程数字化、研发任务集成化、研发过程敏捷化、研发资源共享化等目标,助力企业研发效能提升。

以华为的CodeArts为例,其是一款由华为云推出的一体化云端DevOps平台,专为软件开发的全生命周期设计,包括需求管理、项目管理、构建编译、测试、部署、监控等环节,核心功能主要包括:代码智能辅助、性能分析预测等,该平台强调智能化和自动化,旨在帮助企业实现数字化转型和智能化升级。

对于飞机来说,类似软件工厂研发模式通过标准化、模块化和自动化的手段,有效提高了软件开发的效率和质量。其成功实施不仅依赖于技术工具的应用,更需要团队的协作和持续改进的文化[50]。因此,建设现代化软件工厂,推动上下游产业链协同,提升软件研发响应能力,持续交付高质量软件,驱动飞机能力敏捷升级,推动机载软件现代化发展,是实现软件定义飞机的重要手段。

2.2.3 智能赋能软件研发

大模型技术正在催生软件开发走向智能化新阶段。生成式人工智能将重塑软件全生命周期,从代码为中心走向模型为中心[51]。“在软件定义飞机”的研发过程中,通过引入自动化代码生成、智能测试与调试、智能需求分析等技术手段,将极大提高机载软件开发效率和软件质量,打造软件开发的人机协作新范式。

在开发机载软件过程中,应用大模型技术,通过自动化和智能化工具,帮助研发人员更快速、精准地完成代码编写、调试和测试等工作。例如,自动代码生成技术能够根据设计需求自动生成部分代码,从而减少人为错误并缩短开发周期。像GitHub Copilot这样的工具,已经在软件开发中广泛应用,开发者可以通过简单描述功能需求,获取AI自动生成的代码建议,从而加快开发进程[52]。

此外,智能测试与调试是另一项关键技术,通过机器学习算法对历史数据的分析,可以自动生成高效的测试用例,并且能在代码更改时自动检测潜在的错误。比如,Facebook开发的深度学习模型SapFix,能够在检测到bug后,自动生成修复代码,大幅提高了错误修复的效率,在飞机的软件研发中应用类似技术,可以减少软件缺陷率,提高研发效率。

同时,智能辅助软件研发还包括智能需求分析和预期性维护来进一步优化软件开发流程。人工智能可以分析软件使用场景,提前识别潜在的安全风险,并将这些数据转化为软件需求,指导软件的功能设计,减少机上软件返工迭代的次数。

综上,智能辅助软件研发不仅在开发效率和质量上有显著提升,还为机载软件的灵活性和稳定性提供了有力支撑。未来,随着智能技术的进一步发展,智能赋能的深度融合将成为机载软件研发的核心趋势,推动机载软件研发过程智能化、自动化与安全性提升[53]。

2.3 生态体系建立

为解决“软件定义飞机”面临的问题,可以考虑尝试生态创新的实践主要包括软件工厂生态和跨行业移植生态。

2.3.1 软件工厂生态

针对前文提到的软件工厂研发模式,美空军瞄准商业软件头部企业进行设计,建立了多级软件工厂联合生态,对原有软件工厂进行重组[54],美空军的第一个目标是整合希尔、罗宾斯、廷克空军

基地的软件工程团队,并与其他的软件工厂建立联系。不同的工作组,如采购、人员、韧性等由软件工厂生态中不同的参与者组成,并探索软件工厂之间的协同合作。

车企与互联网公司也都进行将自己的软件开发工具链向生态建立方向推进,如汽车行业的AUTOSAR联盟[55],该联盟由9家汽车公司建立,包括宝马,大众、福特丰田等公司,目前该联盟以及拥有了57家高级成员、50家开发成员、142家普通成员,包括了各大主流整车厂、一级供应商、标准软件供应商等等,该联盟的成立可以汽车全行业上下游集成生态,通过重用提高软件质量并降低成本,实现低成本的跨平台和跨汽车制造商的功能重复利用,实现功能之间接口标准化,并给出分布式开发流程的交换标准格式等。

国内航空相关企业可以学习美空军软件工程生态与汽车行业AUTOSAR全行业生态,建立类似的生态模式,为各级软件工厂提供标准的指导规范与牵引,提供逻辑统一、架构开放、安全可信、可拓展、可连通的专业工具链,支撑各级软件工厂之间的研发资源共建共享共用,可以实现各级软件工厂的协同有序发展,解决“软件定义飞机”跨域协同难等问题。

2.3.2　跨行业移植生态

建立跨行业的移植生态,将汽车等其他行业的软件生态进行移植,可以提高研发效率,提高软件复用率,并且使机载软件生态更加开放,可以更好地实现飞机与飞机、飞机与其他设备之间的互联互通互操作。

例如,可以借鉴手机企业相关的嵌入式操作系统的开发经验,将相关架构、代码、开发流程等引入飞机的计算机里,减少开发成本,提高操作系统性能与稳定性;可以借鉴汽车架构生态,构建航空装备通用架构,引入车规级计算模组、车载操作系统、数字化软件开发工具等技术,借鉴高算力通用计算主机、原子化服务理念、时间敏感网络与运维在线持续升级方法,赋能航空装备研制。

3　结束语

从无线电到卫星,从汽车到装备,在飞机领域由硬件为中心到软件为中心。“软件定义”已成趋势,思考如何采用开放式系统架构、复杂系统工程贯穿软件工程、服务化架构设计解决复杂系统设计与软件研发问题,并将软件工厂研发模式与OTA技术结合,引入大模型赋能软件研发、规模化仿真验证技术,有助于未来进一步提升软件开发、验证与部署效率,加速用户需求响应环闭合。

“软件定义”已成为生产方式升级、生产关系变革、新型产业发展的重要引擎,成为驱动未来发展的重要力量,后续需要充分释放“软件定义”的创新活力和驱动力,不断探索,打破瓶颈,在关键领域实现突破,积极助力现代化航空工业体系建设。

参 考 文 献

[1] 刘献科, 张栋岭, 陈涵生. 软件定义无线电及软件通信体系结构规范[J]. 计算机工程, 2004(1): 95-96,131.
LIU Xianke, ZHANG Dongling, CHEN Hansheng. Software defined radio and software communication architecture specification[J]. Computer Engineering, 2004(1): 95-96, 131. (in Chinese)

[2] 陈建光, 王聪, 梁晓莉. 国外软件定义卫星技术进展[J]. 卫星与网络, 2018(4): 50-53.
CHEN Jianguang, WANG Cong, LIANG Xiaoli. Foreign software definition satellite technology progress[J]. Satellite and Network, 2018(4): 50-53. (in Chinese)

[3] 孟天闯, 李佳幸, 黄晋, 等. 软件定义汽车技术体系的研究[J]. 汽车工程, 2021, 43(4): 459-468.
MENG Tianchuang, LI Jiaxing, HUANG Jin, et al. Research on software defined vehicle technical system[J]. Automotive Engineering, 2021, 43(4): 459-468. (in Chinese)

[4] 张红. 新一代综合航电系统的发展趋势[J]. 航空精密制造技术, 2016, 52(2): 1-4.
ZHANG Hong. Development trend of new generation integrated avionics systems[J]. Precision Aircraft Manufacturing Technologies, 2016, 52(2): 1-4. (in Chinese)

[5] 罗锐, 李国柱. F-35航电系统的发展[J]. 舰船电子工程, 2012, 32(8): 21-23,63.
LUO Rui, LI Guozhu. Development of an avionics system F-35[J]. Marine Electronics Engineering, 2012, 32(8): 21-23,63. (in Chinese)

[6] 刘继国, 李江帆. 国外制造业服务化问题研究综述[J]. 经济学家, 2007(3): 119-126.
LIU Jiguo, LI Jiangfan. Review of research on service in foreign manufacturing industry[J]. Economist, 2007(3): 119-126. (in Chinese)

[7] 陈宗基, 秦旭东, 高金源. 非相似余度飞控计算机[J]. 航空学报, 2005, 26(3): 320-327.
CHEN Zongji, QIN Xudong, GAO Jinyuan. Non-similar redundancy flight control computer[J]. Journal of Aeronautics, 2005, 26(3): 320-327. (in Chinese)

[8] 孙金明. 自动化技术的三大革新[J]. 中国科技信息, 2005(14): 18.
SUN Jinming. Three major innovations in automation technology [J]. China Science and Technology Information, 2005(14): 18. (in Chinese)

[9] 唐杰, 牟佳媛. 电气工程中自动化技术的运用[J]. 科技创新与应用, 2013(1): 63.
TANG Jie, MOU Jiayuan. Application of automation technology in electrical engineering [J]. Technological Innovation and Application, 2013(1): 63. (in Chinese)

[10] 黄雪芳. 探讨电气工程中自动化技术的应用[J]. 广东科技, 2012, 21(13): 48.
HUANG Xuefang. Exploration of the application of automation technology in electrical engineering[J]. Guangdong Science and Technology, 2012, 21(13): 48. (in Chinese)

[11] 董鹏辉, 车飞锋, 李强, 等. 一种机电管理系统控制的开放式和通用化设计方法[C]// 第六届民用飞机航电国际论坛. 西安: 中国航空学会, 2017: 1-5.
DONG Penghui, CHE Feifeng, LI Qiang, et al. An open and universal design method of electromechanical management system control [C]// The Sixth International Forum on Civil Aircraft Avionics. Xi'an: CSAA, 2017: 15. (in Chinese)

[12] 金欣, 闫晶晶, 赵克俭. 面向联合作战的服务化指控系统软件架构研究[C]// 2014第二届中国指挥控制大会. 北京: 中国指挥与控制学会, 2014: 124-130.
JIN Xin, YAN Jingjing, ZHAO Kejian. Research on the software architecture of service-oriented command control system for united wars [C]// 2014 the Second China Command Control Conference. Beijing: Chinese Society of Command Control, 2014: 124-130. (in Chinese)

[13] 岳超源, 系统工程,决策理论与方法[M]. 北京: 科学出版社, 2003.
YUE Chaoyuan, Systems engineering, decision theory and methods[M]. Beijing: Science Press, 2003. (in Chinese)

[14] 王众托. 知识系统工程[M]. 北京: 科学出版社, 2004.
WANG Zhongtuo. Knowledge systems engineering [M]. Beijing: Science Press, 2004. (in Chinese)

[15] 黄秀娟. "软件工厂"式项目教学的实践研究[J]. 职业圈, 2007(7): 61-63.
HUANG Xiujuan, "Software factory" project teaching practice study[J]. Career Circle, 2007(7): 61-63. (in Chinese)

[16] 马春黎, 卢苇. 软件工厂研究现状综述[J]. 计算机教育, 2006(12): 48-49.
MA Chunli, LU Wei. Overview of the status of software factories research[J]. Computer Education, 2006(12): 48-49. (in Chinese)

[17] 郭兵. 嵌入式软件开放式集成开发平台体系结构研究[D]. 成都: 电子科技大学, 2002.
GUO Bing. Research on architecture of open integrated development platform for embedded software [D]. Chengdu: University of Electronic Technology, 2002. (in Chinese)

[18] 李豪, 付强, 张宇飞. 基于云原生技术的机载软件开放式架构设计研究[J]. 电脑编程技巧与维护, 2023(5): 36-38.
LI Hao, FU Qiang, ZHANG Yufei. Open architecture design study of airborne software based on cloud proprietary technology [J]. Computer Programming Skills and Maintenance, 2023(5): 36-38. (in Chinese)

[19] 韩凝晖, 陈宋宋, 张旭东, 等. 智能家居综合传感装置开放式架构设计[J]. 电力信息与通信技术, 2020, 18(4): 104-108.
HAN Ninghui, CHEN Songsong, ZHANG Xudong, et al. Open architecture design of intelligent home integrated sensing device[J]. Power Information and Communication Technology, 2020, 18(4): 104-108. (in Chinese)

[20] 武志学. 云计算虚拟化技术的发展与趋势[J]. 计算机应用, 2017, 37(4): 915-923.
WU Zhixue. Developments and trends in cloud computing virtualization technology [J]. Computer applications, 2017, 37(4): 915-923. (in Chinese)

[21] 赵维五, 段子腾. 云原生安全之K8s安全实践[J]. 保密科学技术, 2023(6): 48-52.
ZHAO Weiwu, DUAN Ziteng. K8s safety practice for cloud primary security [J]. Secrecy Science and Technology, 2023(6): 48-52. (in Chinese)

[22] 黄璜, 张贺, 邵栋. 自动化工具对中国DevOps实践的影响[J]. 软件学报, 2019, 30(10): 3056-3070.
HUANG HUANG, ZHANG He, SHAO Dong. The impact of automation tools on Chinese DevolOpts practice[J]. Journal of Software, 2019, 30(10): 3056-3070. (in Chinese)

[23] 张军红, 童强. 基于软件虚拟化技术的新一代航空机载软件设计[J]. 南京航空航天大学学报, 2019, 51(6): 772-777.
ZHANG Junhong, TONG Qiang. New generation airborne software design based on software virtualization technology [J]. Journal of Nanjing University of Aeronautics and Astronautics, 2019, 51(6): 772-777. (in Chinese)

[24] 张春霞, 李旭东, 徐涛. 浅谈面向服务架构(SOA)的核心理念[J]. 计算机系统应用, 2010, 19(6): 251-256.
ZHANG Chunxia, LI Xudong, XU Tao. On the core idea of service oriented architecture (SOA)[J]. Computer System Application, 2010, 19(6): 251-256. (in Chinese)

[25] 吴封斌, 李笑瑜, 蒲睿强, 等. Istio:微服务架构服务治理升级研究[J]. 网络安全技术与应用, 2022(7): 1-2.
WU Fengbin, LI Xiaoyu, PU Ruiqiang, et al. Istio: microservice architecture service governance upgrade research [J]. Network Security Technology and Applications, 2022(7): 1-2. (in Chinese)

[26] 朱立平, 陆志东, 张丹涛, 等. 一种基于FACE标准的飞行器管理系统软件架构[J]. 微电子学与计算机, 2019, 36(3): 77-81.

ZHU Liping, LU Zhidong, ZHANG Dantao, et al. A software architecture for aircraft management system based on FACE standard [J]. Microelectronics and Computers, 2019, 36(3): 77-81. (in Chinese)

[27] 孟永琪, 董长清, 夏竞, 等. 基于OTA技术的掌上运维系统建设与应用[J]. 移动通信, 2008, 32(22): 26-28.
MENG Yongqi, DONG Changqing, XIA Jing, et al. Construction and application of pocket operation maintenance system based on OTA technology[J]. Mobile Communication, 2008, 32(22): 26-28. (in Chinese)

[28] 马鑫. 国标落地汽车OTA规范有讲求[N]. 中国汽车报, 2024-09-23(026).
MA Xin. National standard landing vehicle OTA specification[N]. China Auto News, 2024-09-23(026). (in Chinese)

[29] 柳旭, 周时莹, 范玲玲, 等. 智能网联车辆空中下载系统精益化设计[J]. 汽车工程师, 2024(6): 1-7.
LIU Xu, ZHOU Shiying, FAN Lingling, et al. Lean design of air load system of intelligent IoT[J]. Automotive Engineer, 2024(6): 1-7. (in Chinese)

[30] 景意新. 简述空中下载技术的热点应用[J]. 电器工业, 2021(12): 75-78.
JING Yixin. Brief description of hot applications of airborne downloading technology[J]. Electrical Industry, 2021(12): 75-78. (in Chinese)

[31] 陶鸿飞, 赵正德, 王文. 空中下载(OTA)系统安全性的研究[J]. 中国图象图形学报, 2008, 13(10): 1930-1933.
TAO Hongfei, ZHAO Zhengde, WANG Wen. Research on system safety in air download (OTA)[J]. China Graphics Bulletin, 2008, 13(10): 1930-1933. (in Chinese)

[32] 褚端峰, 王如康, 王竞一, 等. 端到端自动驾驶的研究进展及挑战[J/OL]. 中国公路学报, 1-29[2024-11-04]. http://kns.cnki.net/kcms/detail/61.1313.U.20241011.1516.002.html.
CHU Duanfeng, WANG Rukang, WANG Jingyi, et al. Research progress and challenges of end-to-end self-driving [J/OL]. China Road Journal, 1-29[2024-11-04]. http://kns.cnki.net/kcms/detail/61.1313.U.20241011.1516.002.html. (in Chinese)

[33] 栗慧哲, 刘明扬, 王江波, 等. 智能网联环境下自动驾驶模块化公交服务框架设计[J]. 人民公交, 2024(11): 59-63.
LI Huizhe, LIU Mingyang, WANG Jiangbo, et al. Design of auto-driving modular traffic service framework under intelligent Internet[J]. People's Public Transport, 2024(11): 59-63. (in Chinese)

[34] 高驰. “端到端”能否颠覆自动驾驶?[J]. 汽车与配件, 2024(11): 58-59.
GAO Chi. "End-to-end "can subvert self-driving[J]. Cars and Accessories, 2024(11): 58-59. (in Chinese)

[35] 傅文进, 朱龙. 基于人工智能的用电数据采集策略优化分析[J]. 集成电路应用, 2024, 41(8): 242-243.
FU Wenjin, ZHU Long. Optimal analysis of data acquisition strategy based on artificial intelligence[J]. Integrated Circuits, 2024, 41(8): 242-243. (in Chinese)

[36] 吴中勤, 韩钧宇. 如何研发中国新一代自主可控的AI算法引擎[J]. 人工智能, 2019(2): 112-121.
WU Zhongqin, HAN Junyu. How to develop a new generation of self-controlled AI algorithm engine in China[J]. Artificial Intelligence, 2019(2): 112-121. (in Chinese)

[37] 文鹏程, 白林亭, 高泽, 等. 机载智能计算技术工程实践思考[J]. 航空计算技术, 2021, 51(2): 130-134.
WEN Pengcheng, BAI Linting, GAO Ze, et al. Engineering practice of airborne intelligent computing technology[J]. Aeronautical Computing Technology, 2021, 51(2): 130-134. (in Chinese)

[38] 卢新来, 杜子亮, 许赟. 航空人工智能概念与应用发展综述[J]. 航空学报, 2021, 42(4): 251-264.
LU Xinlai, DU Ziliang, XU Yun. Overview of the concepts and applications of aeronautical artificial intelligence [J]. Journal of Aeronautics, 2021, 42(4): 251-264. (in Chinese)

[39] 陶飞, 张贺, 戚庆林, 等. 数字孪生模型构建理论及应用[J]. 计算机集成制造系统, 2021, 27(1): 1-15.
TAO Fei, ZHANG He, QI Qinglin, et al. Theory and application of digital twin model construction and application [J]. Computer Integrated Manufacturing System, 2021, 27(1): 1-15. (in Chinese)

[40] 陈泽煜, 贺永宁, 宋宁, 等. 美国阿尔忒弥斯载人登月计划进展[J]. 空间电子技术, 2022, 19(6): 75-84.
CHEN Zeyu, HE Yongning, SONG Ning, et al. Progress of Artemis manned landing project in USA [J]. Space Electronics Technology, 2022, 19(6): 75-84. (in Chinese)

[41] 谭莉娟, 郑巍, 刘友林, 等. 面向适航标准的机载软件测试验证方法综述[J]. Journal of Computer Engineering & Applications, 2021, 57(15): 9-22.
TAN Lijuan, ZHENG Wei, LIU Youlin, et al. Overview of airworthiness standards based airborne software test validation method[J]. Journal of Computer Engineering & Applications, 2021, 57(15): 9-22. (in Chinese)

[42] 韩峰岩, 王昕. 机载计算机软件的测试[J]. 航空计算技术, 2004, 34(3): 65-69.
HAN Fengyan, WANG Xin. Test of airborne computer software [J]. Aviation Computing Technology, 2004, 34(3): 65-69. (in Chinese)

[43] 成思危. 复杂科学与系统工程[J]. 管理科学学报, 1999, 2(2): 1-7.
CHENG Siwei. Complex science and systems engineering [J]. Journal of Management Sciences, 1999, 2(2): 1-7. (in Chinese)

[44] 金菊良, 魏一鸣. 复杂系统广义智能评价方法与应用[M]. 北京: 科学出版社, 2007.
JIN Jiuliang, WEI Yiming. Generalized intelligent evalua-

tion method and application of complex system[M]. Beijing: Science Press, 2007. (in Chinese)

[45] 胡军, 吕佳润, 王立松, 等. 一个机载软件需求形式化建模与分析实例研究[J]. 软件学报, 2022, 33(5): 1652-1673.
HU Jun, LYU Jiayun, WANG Lisong, et al. A case study of formal modeling and analysis of airborne software requirements[J]. Journal of Software, 2022, 33(5): 1652-1673. (in Chinese)

[46] 张卫, 丁金福, 纪杨建, 等. 工业大数据环境下的智能服务模块化设计[J]. 中国机械工程, 2019, 30(2): 167-173.
ZHANG Wei, DING Jinfu, JI Yangjian, et al. Modularization design of intelligent services industrial data environment [J]. China Mechanical Engineering, 2019, 30(2): 167-173. (in Chinese)

[47] 刘希, 朱凡, 蔡满意, 等. 飞行控制系统可视化仿真平台设计[J]. 计算机工程, 2011, 37(16): 260-262,266.
LIU Xi, ZHU Fan, CAI Manyi, et al. Visual simulation platform design for flight control systems[J]. Computer Engineering, 2011, 37(16): 260-262,266. (in Chinese)

[48] 焦洪硕, 鲁建厦. 智能工厂及其关键技术研究现状综述[J]. 机电工程, 2018, 35(12): 1249-1258.
JIAO Hongshuo, LU Zhixia. A review of the status of research on intelligent factories and key technologies[J]. Electrical and Mechanical Engineering, 2018, 35(12): 1249-1258. (in Chinese)

[49] 臧飞, 党敏侠. 美军软件工厂发展实践研究综述[J]. 中国军转民, 2024(17): 61-63.
ZANG Fei, DANG Minxia. Review of the practice of software factory development in US Army[J]. China Military Transfers, 2024(17): 61-63. (in Chinese)

[50] 郑军, 黄志球, 徐丙凤. 机载软件适航认证标准新进展及展望[J]. 计算机工程与设计, 2012, 33(1): 204-208.
ZHENG Jun, HUANG Zhiqiu, XU Bingfeng. New development and prospect of airworthiness certification standard for airborne software[J]. Computer Engineering and Design, 2012, 33(1): 204-208. (in Chinese)

[51] 马晓星, 刘譞哲, 谢冰, 等. 软件开发方法发展回顾与展望[J]. 软件学报, 2019, 30(1): 3-21.
MA Xiaoxing, LIU Xizhe, XIE Bing, et al. Review and prospect of software development methods[J]. Journal of Software, 2019, 30(1): 3-21. (in Chinese)

[52] 王东煜, 莫然, 詹文静, 等. 代码自动生成工具Github Copilot代码质量分析[J/OL]. 计算机科学, 1-23[2024-11-04]. http://kns.cnki.net/kcms/detail/50.1075.tp.20241012.1256.038.html.
WANG Dongyu, Mo Ran, ZHAN Wenjing, et al. Code quality analysis of Github Copilot code quality analysis[J/OL]. Computer Science, 1-23[2024-11-04]. http://kns.cnki.net/kcms/detail/50.1075.tp.20241012.1256.038.html. (in Chinese)

[53] 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. Embedded software security analysis standards, methods and tools research summary[J]. Journal of Software, 2014, 25(2): 200-218. (in Chinese)

[54] 曹远, 曾令斌, 黄友, 等. 美军软件工厂:模式,案例及启示[J]. 国防科技, 2024, 45(6): 92-101.
CAO Yuan, ZENG Lingbin, HUANG You, et al. The U.S. Military software factory: models, cases and implications [J]. National Defense Science and Technology, 2024, 45(6): 92-101. (in Chinese)

[55] FENNEL H, BUNZEL S, HEINECKE H, et al. Achievements and exploitation of the AUTOSAR development partnership[R]. US: Society of Automotive Engineers, 2006.

(编辑:丛艳娟)