

引用格式:董勤,沈晶晶,张海军,等.模型驱动的座舱显示软件一体化开发及验证方法研究[J].电光与控制,2018,25(2):92-97. DONG M, SHEN J, ZHANG H J, et al. Model-driven cockpit display software: development and verification[J]. Electronics Optics & Control, 2018, 25(2):92-97.

模型驱动的座舱显示软件一体化开发及验证方法研究

董勤, 沈晶晶, 张海军, 刘晓宏, 刘俊锋, 郭晓光
(中国航空工业集团公司洛阳电光设备研究所, 河南 洛阳 471000)

摘要:为适应座舱人机交互显控技术的发展,基于现有的模型开发思想,结合座舱显示软件的开发需求,提出一种模型驱动的座舱显示软件一体化开发及验证方法,对软件架构分层设计,清晰地分离出了底层驱动和模型业务,实现软件全生命周期的图形化建模、集成与仿真,构建可复用、可配置的松耦合软件架构,建立一体化的模型开发与验证方案。通过多个软件项目实践,开展与传统“以代码为中心”开发模式的对比,证明该方法能够更为高效、高质量地实现显示软件的开发与验证,可广泛应用于座舱显示控制技术领域。

关键词:座舱显示软件;模型驱动;一体化;开发及验证;航空电子系统

中图分类号: V243 **文献标志码:** A **doi:**10.3969/j.issn.1671-637X.2018.02.019

Model-Driven Cockpit Display Software: Development and Verification

DONG Meng, SHEN Jing-jing, ZHANG Hai-jun, LIU Xiao-hong, LIU Jun-feng, GUO Xiao-guang
(Luoyang Institute of Electro-Optical Equipment, AVIC, Luoyang 471000, China)

Abstract: Considering the development of the cockpit human-computer interaction display technology and the development requirements of the cockpit display software, we proposed a development-and-verification method of the model-driven cockpit display software based on existing model development ideas. Hierarchical design was used in the software architecture to clearly separate the bottom driver from the model operation. The graphical modeling, integration and simulation of the software were realized in its full life circle to construct a reusable, configurable and loosely-coupled software architecture, so as to build an integrated model development and verification scheme. Through the practice of several software projects, and the comparison with the traditional “code-centric” development model, it has been proved that the proposed method can achieve high-quality development and verification of the display software more effectively, which can be used widely in the area of cockpit display.

Key words: cockpit display software; model-driven; integration; development and verification; avionic system

0 引言

近年来,随着航空技术的发展,飞机座舱航电系统的功能日益复杂,座舱显示软件的复杂程度也越来越高。为满足现代战争多机协同、多样化飞行任务、多传感器分布显示、多角度信息融合的实战需求,应提供直观、大视场、大信息量、综合智能化的座舱显示,以增强飞行员的态势感知能力,这些给座舱显示软件的开发与验证带来了巨大挑战^[1]。

由于显示控制业务、显示内容复杂程度的提高,

传统“以代码为中心”的开发模式,在软件实际开发过程中,往往面临诸多问题:人工编码工作量大,容易引入错误;效率低下,开发周期较长;不能灵活适应需求的频繁变更,需要做大量重复的工作;验证工作对于硬件平台的依赖性高,缺少仿真验证平台,验证测试困难、验证工作量大;画面和符号设计不直观,后续维护、复用困难。

因此,在较短的研制周期内,面对较为复杂的设计需求,传统的开发模式已经很难适应和满足。特别是对于新一代触控式座舱人机交互显控技术的发展与应用^[2-3],迫切需要引入全新的软件开发方式,模型驱动^[4-5]的座舱显示软件一体化开发及验证方法研究应运而生。

收稿日期:2017-05-16

修回日期:2017-06-15

作者简介:董勤(1988—),男,河南驻马店人,硕士生,研究方向为瞄准显示软件设计。

1 模型驱动的显示软件设计方法

1.1 显示软件生命周期模型

如图 1 所示,模型驱动的显示软件开发流程采用 V 模型,基于所见即所得的模式,具备灵活、高效、易仿真和易维护的优势。通过系统需求分析、软件需求分析、模型化设计、自动生成代码、模型仿真验证等一系列技术,提高了座舱显示软件设计的工程化水平,缩短了设计及验证周期,为保证软件质量,提高系统安全性提供全面支持。

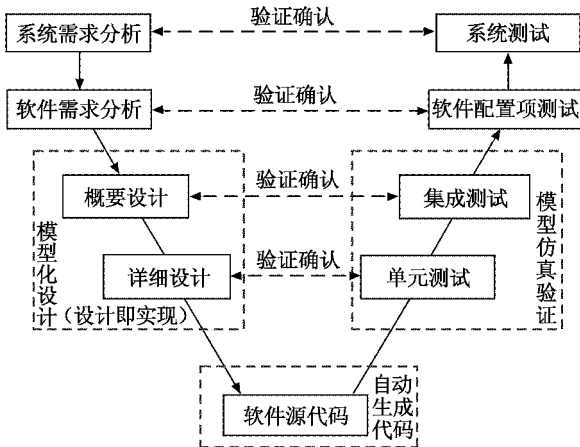


图 1 软件生命周期 V 模型

Fig. 1 V model of software life cycle

通常,模型驱动的显示软件开发系统主要包括:座舱显示人机接口 (Human Machine Interface, HMI) 设计工具,2D/3D OpenGL/ES 图形高性能显示开发环境,如 VAPS XT,SCADE DISPLAY,IData 等软件;显示画面模型开发平台及仿真终端,用于开发及仿真画面图形库,以及显示给定的导航、飞行、作战画面信息;目标机是支持 OpenGL/ES 标准接口的图形引擎硬件平台,如 Arm i. Mx6Q,ATI M54, M9, 国产 5400 图形显卡等;画面显示终端则用于接收显卡视频信号,输出显示画面。

在设计阶段,座舱显示 HMI 设计工具可运行于不同的开发平台,如 Windows PC 端、服务器端或 Linux 平台等,并在仿真终端生成演示画面。验证后,将图形软件 Rehost(主机转换)到目标机图形发生器上。模型驱动座舱显示软件的开发,覆盖了软件需求、软件设计、软件实现和软件测试的全生命周期。

在工程化实施过程中,基于桥接工具可以与其他系统或模型建立彼此的关联:一方面能够重用已有的模型和设计,减少工作量;另一方面可以开展系统级的全局仿真和分析,确定系统行为、分析模型覆盖率、进行形式化验证等,实现软件状态在各开发阶段的高效管理和控制。

目前,主流的桥接工具包括针对软件配置、需求和

文档的管理、数据追踪和编译测试等多个种类,如与 DOORS 桥接实现模型的需求管理,与 CM 桥接实现配置管理,与 Simulink 桥接实现对原有工作的继承和复用^[6],与 CVK 桥接验证编译器能否正确编译模型生成的代码,与 SCADE LIFECYCLE 桥接实现文档的自动化生成、自动化测试环境配置,以及软件全生命周期的数据追踪管理,保证需求、设计和文档的一致性,如图 2 所示。

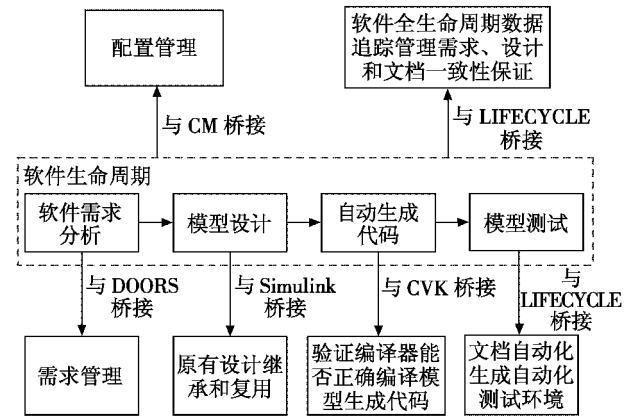


图 2 软件工程化桥接方法

Fig. 2 The bridging method for software engineering

1.2 显示软件架构分层设计

座舱显示软件架构的设计,可借鉴未来航空器能力环境 (Future Airborne Capability Environment, FACE) 架构标准,对软件架构进行分层,实现软件各模块、各组件的可配置、可移植、可扩展和可复用。由于采用了模型驱动的设计方法^[7],软件实体将包括模型设计(控制模型和显示模型)和人工编码(硬件驱动、总线通讯、系统调用等)两部分。

根据分层复用的原则,需要将模型设计与驱动代码分离、将控制逻辑和画面显示分离,从而满足软件架构各层级之间松耦合、强内聚的特性。模型驱动的显示软件将划分为驱动层、调度层、模型控制层和模型显示层四层架构,如图 3 所示。

驱动层为硬件平台相关层,面向各种硬件设备和硬件架构,提供各类硬件驱动,如 CPU 寄存器访问、RS-485,RS-422,1553B,FC,1394 总线通讯接口等,其目的是建立底层驱动库,屏蔽底层硬件和驱动细节,为调度层提供统一的接口服务。

调度层是用于衔接驱动层与模型层的中间架构层,调用驱动接口操作底层的硬件平台和设备,调用模型接口完成画面和符号的显示控制,并通过调度融合架构使得底层驱动与上层模型应用之间实现融合,形成完整运行的系统软件状态。

模型控制层用于实现数据的处理、模型业务的计算,以及显示控制逻辑的处理等,建立基于数据驱动的可配置模块和组件,为模型显示提供标准的显示数据

和可移植的功能模块。

模型显示层则是利用模型控制层输出的显示数据,驱动画面和符号完成显示绘制,其与模型控制层分离,从而满足控制逻辑和画面显示的松散耦合要求。

采用分层设计的方式,各层次之间通过接口调用,完成逐层之间的数据交互、功能和性能的集成,实现了模型与驱动的融合、控制与显示的融合,极大提升了软件的可配置、可移植能力,形成了灵活、可复用的松耦合软件架构。

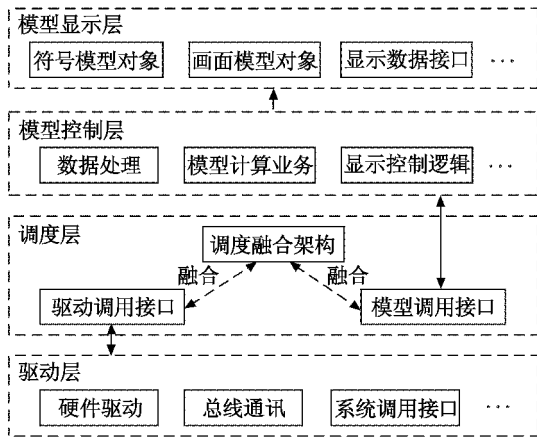


图 3 软件架构层次划分

Fig. 3 Hierarchy of software architecture

1.3 显示软件模型设计流程

座舱显示软件模型设计的流程关键在于:基于 HMI 平台将软件需求转化为画面图形化建模,以及面向目标机的应用程序 Rehost 技术,如图 4 所示。

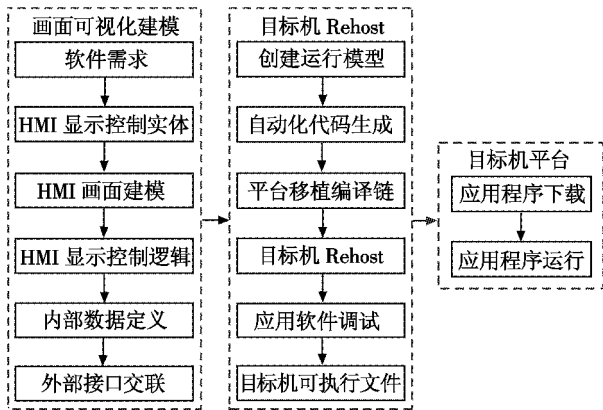


图 4 软件开发流程

Fig. 4 The process of software development

1.3.1 显示需求的图形化建模

从软件需求到画面和符号的显示建模包括如下步骤:将软件需求中各画面和符号的显示,转换为可用 HMI 显示元素表达的控制实体;利用 HMI 建模工具,生成仪表盘、符号对象和画面容器等,完成基本画面显示属性的定义;利用 HMI 工具状态编辑器,使用类 UML

模式,完成显示控制逻辑的判断与计算等工作;定义内部数据通道,实现显示数据或内部变量由画面到符号、符号到属性的逐级传递;定义外部通讯接口,实现显示画面模型与外部构建之间的数据交联,可利用外围程序控制模型的显示内容、显示属性和显示状态。

1.3.2 模型 Rehost 技术

模型驱动的目标机 Rehost 技术,能够在 Windows 上开发显示画面,并在不修改模型的前提下,方便移植到不同的硬件平台上,使得画面模型具备较强的重用性和移植能力,避免大量的重复工作,缩短项目开发周期,提高开发效率。

为了适应各种嵌入式系统目标机平台^[8],HMI 工具软件是将画面控制逻辑和显示绘制进行了分层设计,将与硬件平台相关的构件(主要是图形系统绘图代码)嵌入到代码自动生成框架中,并对代码自动生成器进行配置,定义所使用的跨平台编译工具,使其能够面向不同目标平台的显示驱动进行移植。其自动化代码生成的方式分为两种:1)生成平台无关的代码(如 SCADA DISPLAY),依靠平台自身的编译链完成程序的编译和配置;2)直接绑定平台移植层构件(Porting Layer),生成平台运行代码(如 VAPS XT),并基于 HMI 工具完成工程编译,产生目标文件。

目标机 Rehost 技术需要针对硬件本身,对通用的源代码、实例和模板进行移植和适配,如 CPU 寄存器设置、显卡初始化等,并将硬件驱动、总线通讯、系统调用与 HMI 自动生成的代码进行融合,完成目标软件的开发。软件联调过程需要对 HMI 工具进行下载配置,并针对目标机的软硬件接口进行集成开发、调试,利用跨平台编译工具,与目标机系统库文件建立调用关联,生成目标机可执行文件,达到下载到目标机平台运行和测试的目的,如图 5 所示。

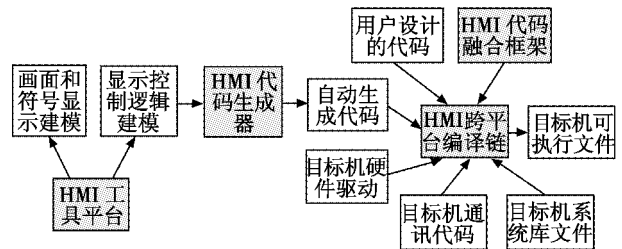


图 5 模型 Rehost 技术原理

Fig. 5 The principle of model Rehost

2 模型驱动显示软件验证方法

2.1 显示软件验证流程

模型驱动显示软件验证包含了模型验证、代码验证和系统验证,如图 6 所示。

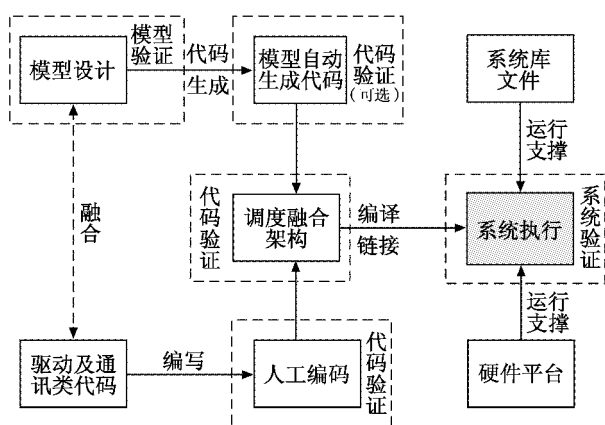


图6 软件验证流程

Fig. 6 The process of software verification

图中:模型验证主要针对模型设计,包括基于HMI平台的画面和符号显示建模和显示控制逻辑建模;代码验证包括模型自动生成代码、人工编写代码(包含硬件驱动、通讯代码等),以及两者融合的代码架构;系统验证则是依赖于硬件平台、系统库文件,对软件系统的执行进行验证。

模型自动生成代码的验证过程,可根据软件项目的实际需要进行简化,特别是对于取得了适航DO-178B标准符合性证据的HMI代码生成器,如SCADE KCG已通过军工及航空业DO-178B标准A级软件鉴定和DO-178C/DO-330的TQL-1级别工具鉴定^[9],无需在源代码级开展大量的验证工作,即无需代码的单元测试,能够节约大量的验证时间。

基于模型的验证遵循严格的模型设计语法和语义(由HMI平台保证),并采用形式化或黑盒验证方法,能够在模型设计阶段完成软件需求的确认,验证设计的正确性、与需求的一致性,从而大幅提高系统的安全性和可靠性。

2.2 显示软件的模型验证

模型驱动的开发方式,采用精确的模型描述软件实体,能够进行仿真测试和验证。对于复杂的座舱显示系统,验证的内容分为显示控制的逻辑处理和画面符号的显示绘制两部分。

针对控制逻辑的验证,要解决系统应用中的逻辑处理和运算,一般采用有限状态机模型,分离软件系统的运行状态,以及各状态之间的数据流和控制流。采用状态机的描述,使得数据流和控制流信息的传递更为准确、清晰,能有效避免需求传递时的二义性。

针对画面符号显示的验证,要解决显示元素的显示有效性和准确性,一般采用离散状态模型,利用搭积木式的方式完成画面或符号模型的搭建,以快速调整画面或符号的布局,修改符号的显示参数,替换符号的

显示策略,从而快速适应画面显示需求的验证。

模型验证^[10]分为模型级验证和代码级验证两部分。其中:代码级验证是针对源代码和目标码的验证,包括软件的单元测试、配置项测试和集成测试;模型级验证则是针对模型设计的验证,包括模型覆盖率的获取、模型与需求的一致性分析,以及模型输出结果的对比等。

由于软件测试的差异性和完备性要求,模型级验证无法完全代替目标环境级的验证,模型生成的代码仍需要完成基于代码的测试。而对于取得了适航DO-178B标准认证的代码生成器,代码测试过程可以简化,无需单元测试,只需要完成配置项测试和集成测试,其测试用例可以复用模型测试的用例集。

系统级的需求则需要基于软硬件综合平台完成验证,其目的不仅是针对软件模型的验证,更是对硬件平台、硬件电路及FPGA逻辑与软件实体所构建的完整的系统功能、系统性能进行的综合测试和验证。

3 基于HMI的显示软件模型设计

目前,针对座舱显示软件的开发,典型的HMI设计工具包括美国昆腾公司的IData,加拿大Presigas公司的VAPS XT,以及法国ESTEREL公司的SCADE。本文以SCADE为例进行介绍。

3.1 基于SCADE的模型设计

SCADE产品线由4个主体模块构成:SUITE, DISPLAY, SYSTEM和LIFECYCLE。其中:SUITE用于控制逻辑的设计和验证;DISPLAY用于人机界面的开发、仿真和验证;SYSTEM用于系统建模和验证;LIFECYCLE用于软件生命周期数据维护。基于SCADE的模型设计^[11],通常采用SUITE对显示控制逻辑建模,采用DISPLAY对画面和符号显示建模,两者能够无缝集成和对接。

具体而言,SCADE SUITE提供了基于数据流图或安全状态机两套图形化建模机制,分别适用于连续系统和离散系统的建模。其建立在严格的数学模型基础上,保证模型设计的精确性、完整性、一致性、可验证性和无二义性;模型编辑器集成了多种视图的显示方式和各类帮助文档,支持访问已发布的典型案例;支持模型覆盖率分析,能够定量评估模型功能的完备程度;能够在模型设计时发现需求错误、模型设计错误、测试用例不足等问题;并具备形式化验证能力,保证复杂逻辑控制系统的安全性。同时,SUITE能够基于模型产生的真实代码进行可视化仿真,根据特定条件设定调试断点、检查变量、记录和回访场景等。

SCADE DISPLAY内置丰富的基本图元、行业标准控件,以及交互式人机界面扩展控件库,用于特定的图形页面设计;支持开发人员对自定义图形库和控件的

封装和重用,提高了库对象和 Widgets 的复用性;集成了数据流控制和基于 UML 的状态机图形编辑,实现了 HMI 模型的逻辑设计;支持多种语言和字体,并能与第三方的 OpenGL 程序叠加集成。同时,DISPLAY 对 OpenGL 标准接口进行了扩展,增强了矢量绘图、遮罩、位图、纹理贴图、光晕及抗锯齿等特效;并支持模块化设计、图形的协同并行开发;提供所见即所得的图形界面,实现人机交互效果的快速动态仿真。

SCADE LIFECYCLE 提供需求捕获、管理需求变更、生成可定制文档的能力,能够在不同的对象间建立可追溯联系,最终使得系统需求、SUITE 和 DISPLAY 模型、测试计划以及其他项目数据保持同步,能够生成一系列的文档报告,包括可追溯矩阵、覆盖率分析、影响分析等,通过 LIFECYCLE 展示板,软件项目经理可以轻松获知需求定义信息、需求实现情况、需求验证结果及模型覆盖率等信息,从而使项目的研制始终保持在高效、可控的状态之下,适用于军用软件研制能力成熟度模型项目工程化实践。

3.2 基于 SCADE 的模型集成仿真

SCADE DISPLAY 作为图形界面开发工具,可与控制逻辑设计工具 SCADE SUITE 联合进行模型无缝集成,实现画面和逻辑的联合设计、联合仿真、联合报告生成和联合代码生成等。这些集成功能都是模型级别实现的,可以在软件设计早期进行设计调试和验证。

此外,SUITE 和 DISPLAY 建模器均提供了将 SCADE 模型以 FMU (Functional Mock-up Unit) 形式导出的功能,能够基于黑盒或白盒的形式,在 Simplorer 多学科建模器的物理仿真系统中实现自动化集成,也使得 SCADE 仿真器能够与外部任何支持 FMI 标准的第三方工具进行集成和仿真。

在 SUITE 和 DISPLAY 模型集成仿真过程中,两者均是基于生成的代码来执行,这样的仿真所看到的是真实行为,也是最终目标系统的行为。因此,集成仿真既是调试两者模型本身的正确性,也是调试目标系统数字化原型的视觉效果。

同时,由于 SUITE 和 DISPLAY 的代码生成器 KCG 均通过了 DO-178B 适航标准认证,能够自动生成产品级标准 C 代码,具备很强的跨平台移植能力,集成仿真工作能够省去大量繁琐的编码过程和验证活动。

3.3 基于模板库的模型优化设计

在基于 HMI 平台进行模型设计时,可将座舱显示画面中常用的图形、符号开发成符号模板,构建显示软件的通用符号模板库。该通用模板库中的符号模板,可直接用于导入应用程序的开发,以减少重复性工作,提高开发效率。

通用符号模板库主要是根据座舱飞行仪表常用的图符进行开发,包括机身基准符、速度矢量符、航向刻度带、速度刻度带、高度刻度带和俯仰梯等典型符号。其中,符号模板的开发,是将各图符作为基础对象进行设计,对象的属性不仅包括符号本身的物理参数,如航向、速度、高度参数等,还包括对象的外观属性,如坐标位置、尺寸、线条、颜色、字体、是否填充、是否闪烁等。基于符号模板的设计可以通过图形化的方法,使用鼠标拖拽添加模板,将符号模板加入到新开发的显示软件中,并通过对象外观属性的修改,适应不同座舱飞行仪表画面的显示需求。

同时,通用符号模板库可以根据显示画面的不同需求进行分类,以形成不同的符号模板主题:根据机型的不同类型形成模板显示主题大类,如歼击机、运输机和直升机等;根据产品或设备的划分形成模板显示主题小类,如平视显示器、头盔显示器、多功能显示器和辅助显示器等,如图 7 所示。

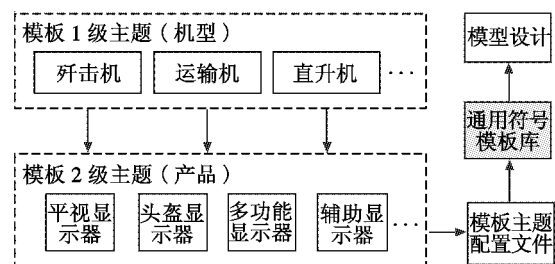


图 7 基于模板库的模型设计

Fig. 7 Template library based model design

其中,模板主题可以采用 XML 格式文件进行配置和定义,提供默认属性和个性化设置的手段,用于快速构建该主题下的座舱仪表的画面布局、符号显示样式、字体及线条样式等,以提升模型库在不同产品或不同应用场景下的复用能力,增强显示软件设计的灵活性,实现显示软件的高效开发。

4 模型驱动与传统模式实践对比分析

基于模型驱动的座舱显示软件一体化开发及验证方法,在 3 个座舱显示软件项目开发中完成实践。分析软件全生命周期各阶段的工作,相比于传统“以代码为中心”的人工编码模式,该方法在软件设计、编码实现、软件测试等环节具备以下优势:

- 1) 模型驱动模式支持控制逻辑建模、画面显示建模的协同并行开发,且图形化的建模方式更为直观和高效;
- 2) 模型化设计即实现,模型可通过代码生成器自动转换为源代码,无须人工编码,较传统模式省略了编码实现阶段;
- 3) 对于取得了 DO-178B 标准认证的 HMI 代码生成器,软件测试过程可以简化,无需做源代

码级别的单元测试;4) 软件集成、配置项测试能够基于 HMI 平台进行仿真,实现开发与验证的所见即所得,而传统模式则需要依赖于产品硬件完成。

分析 3 个软件项目实践的过程数据,形成结论如下:软件设计、编码实现阶段,模型驱动模式效率提升约 44%;同时,免去单元测试环节;软件集成、配置项测试效率提升约 43%。如图 8 所示。

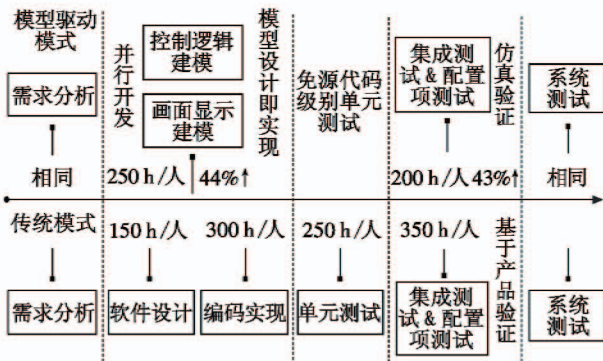


图 8 模型驱动与传统模式显示软件生命周期对比

Fig. 8 Display software life cycle of the two modes

表 1 模型驱动与传统模式显示软件开发对比

Table 1 Display software development comparison of the two modes

对比项	传统模式	模型驱动模式
专业要求	专业技能要求高	可视化设计,技能要求低
培训周期	周期较长,3 个月	周期较短,只需 1~2 周
开发效率	人工编码,效率低	图形化建模,效率高
测试验证	依赖硬件,效率低	免单元测试,支持集成仿真,效率高
文档生成	手工编写,效率低	自动生成设计、测试报告
需求追踪	人工梳理,效率低	自动完成需求追踪
安全性	跟编程水平相关	符合 DO-178B 标准认证
移植性	人工移植工作量大	自动生成跨平台代码
维护性	维护代码,不直观	维护模型、直观易维护
复用性	跟编程水平相关	易于创建图形化复用库

此外,模型驱动模式在开发环节具有优势:代码生成器通过 DO-178B 标准认证,较传统模式具有更高的代码安全性;自动生成面向不同平台的标准 C 代码,具备更强的跨平台移植能力;避免了人工编码中大量存在的编程不规范、数组越界和内存溢出等低级错误,提升软件质量;图形方式建模,对于设计人员的专业技术要求较低,更容易上手实践;支持设计、测试报告一系列文档的自动生成;可实现需求、设计、模型对象的自动追踪和关联。

综上所述,结合 3 个座舱显示软件项目的实践情况和结果,总结模型驱动和传统模式显示软件开发过程的各方面因素,形成对比结论,如表 1 所示。

5 总结

基于模型开发的思想,结合座舱显示软件的开发

需求,提出一种模型驱动座舱显示软件的开发及验证方法,通过对软件架构分层设计,清晰地分离出了底层驱动和模型业务,形成了可复用、可配置的松散软件体系架构,实现了对复杂控制逻辑的建模、画面显示需求的建模设计,并通过模型的集成与仿真,建立了分层架构下模型驱动的软件一体化开发和验证方案。

在一体化开发环境中,模型设计平台与嵌入式编译环境无缝集成,覆盖了软件开发的全生命周期,为软件需求分析、软件设计、软件实现和软件测试提供了一体化操作和部署的开发平台,从而实现快速、高质量的软件开发与验证。

最后,通过多个软件项目实践,开展与传统“以代码为中心”开发模式的对比,证明该方法能够更为高效、高质量地实现显示软件的开发与验证,可广泛应用于座舱显示控制技术领域。

参考文献

- [1] NORTHROP L. Ultra-large-scale systems: the software challenge of the future [D]. USA: Carnegie Mellon University, 2006.
- [2] BARBONI E, NAVARRE D. A formal description technique for interactive cockpit applications compliant with ARINC specification 661 [C]//IEEE 2nd International Symposium on Industrial Embedded Systems, Lisbon, Portugal, 2007:250-257.
- [3] KAMIYAMA T, SOEDA T, YOO M. A Simulink to UML transformation tool for embedded control software design [J]. International Journal of Modeling and Optimization, 2012, 2(3):197-201.
- [4] 唐沁洋. 基于模型驱动的嵌入式实时系统开发平台的研究与实现 [D]. 成都:西南交通大学,2011.
- [5] 刘畅. 模型驱动的飞机座舱显示控制应用软件开发方法研究与实验 [D]. 上海:上海交通大学,2014.
- [6] JOSHI A, HEIMDAHL M P E. Model-based safety analysis of Simulink models using SCADE design verifier [C]//The 24th International Conference on Computer Safety, Reliability, and Security, Springer Berlin Heidelberg, 2005, 3688:122-135.
- [7] 黄书强. 基于模型驱动的软件开发模式研究 [J]. 微电子学与计算机,2009,26(4):234-236.
- [8] 王文全,宋科璞,王勇,等. 基于模型驱动的机载嵌入式软件应用 [J]. 计算机技术与发展,2013,23(8):145-148.
- [9] 徐丙凤,黄志球,胡军,等. 面向适航认证的模型驱动机载软件构件的安全性验证 [J]. 航空学报,2012,33(5):796-808.
- [10] 杨波,吴际,徐璐,等. 一种软件测试需求建模及测试用例生成方法 [J]. 计算机学报,2014,37(3):522-538.
- [11] 胡钢伟,李振水,高亚奎. SCADE 软件开发方法研究 [J]. 系统仿真学报,2009,20(s):286-288.