

基于中标麒麟的测发控类软件一体化设计

孙宝亮, 王毅, 薛凯, 纪祖焱

(空间物理重点实验室, 北京 100071)

摘要:针对新形势下, 软件研制快速迭代、自动化要求高以及国产化要求, 设计并实现了一种跨平台、可复用的测发控类软件一体化设计框架, 解决了航天器地面测试工作自动化程度低、人员介入频繁、误操作率高的问题, 提高系统自动化程度的同时, 满足了软件国产化要求; 所提出软件框架具有扩展性强、稳定性高、使用灵活等特点, 试验表明软件能够适应大部分航天器地面测试流程, 具有跨型号、跨平台使用特性, 具有一定的实用和推广价值。

关键词: 中标麒麟; QT; 软件框架; 设计方法

Integrated Design of Test, Launch and Control Software Based on NeoKylin

SUN Baoliang, WANG Yi, XUE Kai, JI Zubi

(Key Laboratory of Space Physics, Beijing 100071, China)

Abstract: In view of the rapid iteration of software development, high automation requirements and localization requirements under the new situation, a cross platform and reusable integrated design framework of test, launch and control software is designed and implemented, which solves the problems of low automation, frequent personnel intervention and high misoperation rate of spacecraft ground test, and improves the degree of system automation. It meets the requirements of software localization. The proposed software framework has the characteristics of strong expansibility, high stability and flexible use. The test shows that the software can adapt to most spacecraft ground test processes, has the characteristics of cross model and cross platform, and has certain practical and popularization value.

Keywords: Kirin; Qt; software framework; design method

0 引言

随着技术的发展, 航天器作为我国航天技术的集中体现, 有着系统复杂、研制难度高、测试检验流程多、质量要求高的特点^[1-2]。当前形势下, 对航天器软件系统的研制提出了更高的要求, 如何实现航天器的快速装备已成为当前工作的重点。面向装备使用的航天器软件系统的研制、生产相较传统航天型号的研制, 除具有更高质量要求外, 还有面向使用、兼顾平时训练、国产化的使用需求。航天软件系统的研制侧重测试数据的有效性、完整性, 用户多为工程研究人员; 而面向装备的航天器软件系统研制则更侧重可靠性、易用性以及使用的长期性, 用户多为一线操作人员。在当前形势下, 航天软件研制模式已不能适应装备自动化程度高、使用状态多、长期值守的要求。如何实现航天研究研制向装备型号研制的快速转变, 成为当下亟待解决的问题。

针对国产操作系统下, 软件产品化研制要求, 本文提出一种面向测试流程自动化的软件系统一体化设计方法, 实现了在中标麒麟环境下基于 QT 的测发控流程自动控制软件设计与实现。方法以兼顾顶层设计与底层实现、提高系统自动化程度为原则, 开展软件系统设计工作, 通过系统顶层设计、信息流设计, 逐层下沉到软件配置项设计,

识别配置项级功能点, 确保软件更改影响域最小化; 通过软件接口模块化、通用化设计, 降低软件测试、维护成本, 有效缩短软件研制周期、提高软件质量。此外, 通过信息流设计识别关键控制节点、提取流程控制判据, 实现了基于信息流的测发控系统自动控制技术, 有效提高了测发控系统的自动化程度, 降低了人员操作与判读的介入程度, 保障了系统运行的实时性、可靠性。本文以航天器某监控类软件研制过程为例详细说明本文所提出方法的实施过程。

1 软件系统架构设计

1.1 软件系统功能划分

传统航天器软件设计方法往往在电气系统方案确定后, 以电气系统功能需求为输入, 反复迭代软件配置项功能, 该方法的缺点在于系统接口不明确、软件功能不清晰, 软件功能设计强烈依赖系统方案设计, 导致软件设计滞后、设计反复。本文所提出方法, 通过从系统级到配置项级、再到功能模块级逐级多维统一化设计方式, 实现了软件系统各功能点的解耦, 降低了由于系统变更引起的软件维护成本。同时, 由于采用统一化设计, 保证了软件系统研制质量, 降低了软件系统研制成本。本文以系统顶层需求为输入、兼顾电气系统方案, 通过软件系统功能需求开发、配置项功能划分、静态信息流设计、动态信息流设计, 明

收稿日期:2021-10-28; 修回日期:2021-11-08。

作者简介:孙宝亮(1987-),男,河北沧州人,博士,工程师,主要从事软件系统分析与设计方向的研究。

引用格式:孙宝亮,王毅,薛凯,等.基于中标麒麟的测发控类软件一体化设计[J].计算机测量与控制,2021,29(12):145-149,156.

明确了软件系统中各软件配置项的功能要求，同时，依据变更影响最小化原则，指导软件配置项设计，以某监控软件为例，说明设计过程。

某监控软件主要实现多个下位目标设备的监控，同时与多个外系统设备通过网络实现通信。软件通过接收外部设备指令并按指令要求自动控制下位目标设备完成指令动作，同时周期性将监测信息反馈给外部监控设备以图表化方式实时输出。软件功能分解如图 1 所示。

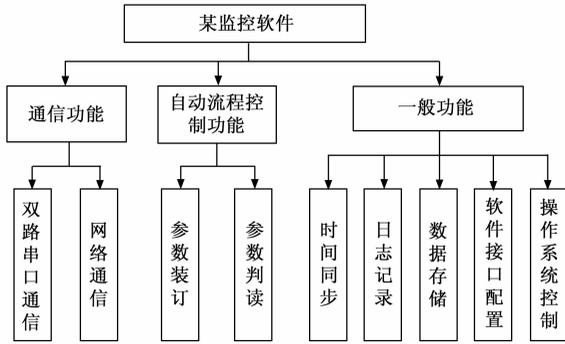


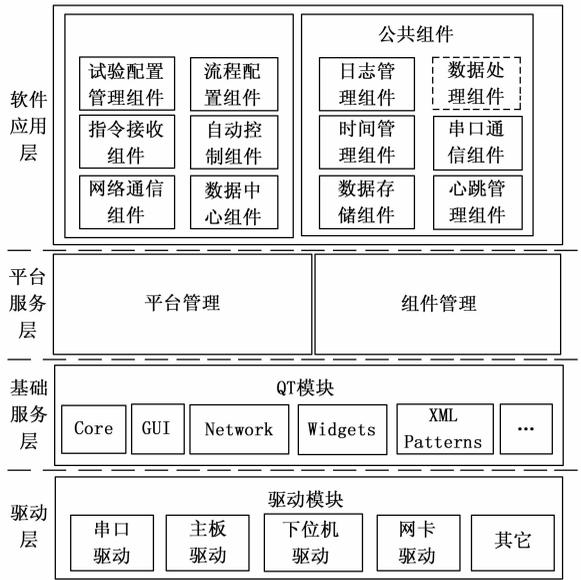
图 1 某监控软件功能框图

本文所设计软件架构为“平台+数据总线+组件”架构。其中，平台完成软件初始化、总线初始化、组件初始化、数据记录、日志记录以及其它公共基础功能初始工作；数据总线为全软件共享数据接口，设计了标准的数据交互接口；组件为具体业务模块。软件中，各功能模块以标准接口接入总线，完成模块间的数据交互的同时避免了模块间资源的竞争与调用，避免了软件跨线程调用和线程竞争的风险。软件结构如图 2 所示。

1.2 软件系统功能划分

软件系统接口设计工作依据软件配置项功能分解和检测一体化系统的要求，对软件系统内各个配置项之间的通信接口进行了设计。

依据软件系统功能划分，基于接口统一化、标准化设计



原则，设计了系统间、子系统内接口关系，实现了软件配置项级的接口与实现分离，确保了软件配置项的松耦合，同时针对各子系统设计了配置项级的逻辑架构图。某监控软件静态信息流如图 3 所示。

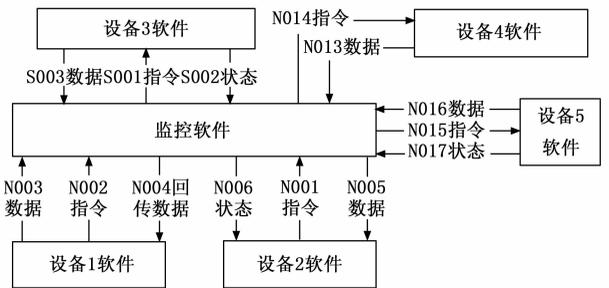


图 3 监控软件静态信息流设计

其中，静态信息流说明如表 1 所示。

表 1 某监控软件静态信息流说明

序号	信息标志	信息名	信源	信宿	触发条件	监听者	周期性	停止条件	备注
1.	N001	指令	设备 1 软件	监控软件	用户操作	—	—	—	—
2.	N002	指令	设备 1 软件	监控软件	用户操作	—	—	—	—
3.	N003	数据	设备 1 软件	监控软件	N002	—	—	—	—
4.	N004	回传数据	监控软件	设备 1 软件	N003	—	—	—	—
5.	N005	数据	监控软件	设备 2 软件	N001	—	周期	—	—
6.	N006	状态	监控软件	设备 2 软件	N001	—	周期	—	—
7.	N014	设备 4 指令	监控软件	设备 4 软件	N001	—	—	—	—
8.	N013	设备 4 数据	设备 4 软件	监控软件	—	—	周期	—	—
9.	N015	设备 5 指令	监控软件	设备 5 软件	N001	—	—	—	—
10.	N016	设备 5 数据	设备 5 软件	监控软件	N015	—	周期	—	—
11.	N017	设备 5 状态	设备 5 软件	监控软件	N015	—	周期	—	—
12.	S001	流程指令	监控软件	设备 3 软件	N001	—	—	—	—
13.	S002	状态	设备 3 软件	监控软件	S001	—	周期	—	—
14.	S003	数据	设备 3 软件	监控软件	S001	—	周期	—	—

1.3 自动化流程控制设计

针对测控系统自动测试及控制需求, 进一步的, 梳理系统关键动作时序关系, 开展软件系统动态信息流设计, 进一步细化软件功能点与系统时序关系, 本文针对系统间的逻辑关系、时序动态、数据判读设计了子系统动态信息流如图 4 所示。

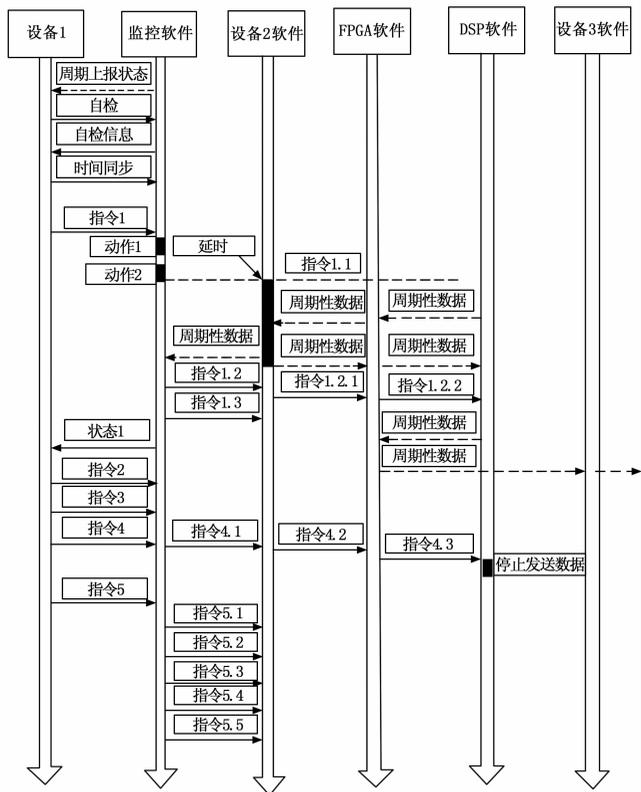


图 4 某子系统自动化控制信息流

依据系统信息流设计流程判据文件, 判据文件为系统自动化执行的依据, 考虑到系统通用性和使用需求, 本文设计了适用于测发控系统自动控制流程的判据管理模块、流程配置模块、分支执行模块、消息通知模块, 软件能够根据测发控需求灵活配置, 实现跨型号测发控系统流程自动控制技术, 提升测发控系统的自动化程度。其中, 流程配置模块针对测控系统常用判读逻辑一与、或、非、大于、等于、小于等基础逻辑及其组合形成通用化流程控制逻辑配置文件, 能够适应大部分测控系统流程控制需求。

经过以上软件系统功能分解、时序设计、功能点拆解, 明确了软件系统的工作模式和运行状态, 实现了系统数据流控制流动态设计, 确保了接口设计匹配性, 保证了系统设计无死角、无歧义, 进而为系统的自动化控制技术提供了技术和方案支撑。在此基础上进一步开展测发控自动控制技术研究。本项目通过梳理系统信息流间的关系结合电气系统方案, 设计了基于信息流的自动化控制方案, 软件自动控制流程如图 5 所示。

其中, 流程中主要包含外部指令、执行机构、数据判

读等关键动作外部指令为系统输入, 结合外系统使用及系统设计方案完成关键检查点控制; 执行机构为系统内部控制单元, 根据指令执行相应动作; 数据判读为流程关键数据、指标的判读, 只有符合预置数据包络范围内, 方可自动执行控制流程。测控系统自动控制需求源自系统方案设计, 其中数据判读结合系统信息流开展设计, 综合系统时序设置合理的判据包络支撑后续控制流程的自动化。

2 研制流程

软件研制过程中, 打破了系统、配置项的界限, 建立以系统信息流、控制流为导向的设计模式。然而, 方法的效率提升离不开软件产品化设计, 反过来细粒度软件产品化设计能够拓展方法的灵活性和标的性。以可重用构件为原材料, 以自主研发的通用化平台框架为基辅, 实现型号软件的快速研发和生产范式。在软件项目管理方面, 软件采用集成项目管理模式开展统一管理, 充分优化管理成本。

2.1 软件平台

在软件研制阶段, 软件以功能组件为基本单位, 进行“积木式”开发。软件通过平台功能扩展, 实现软件应用层功能。平台提供标准公共接口, 可灵活加载/卸载功能组件, 且不影响软件主体功能实现。该技术能极大地提高软件产品黑盒复用率, 提高代码质量, 同时能够有效配合本文所提出方法的设计思想, 实现对成熟构件的可靠性和安全性的继承。

2.2 数据总线

数据总线为软件内部各功能组件进行数据交互的共用数据访问模块。该模块设计有标准数据访问接口, 各功能组件按需访问总线数据, 同时与其它组件实现隔离, 避免数据读写操作造成的错误。同时, 数据总线维护总线数据结构, 各组件可根据实际需要完成对数据的新建、删除、修改等操作, 总线维护数据的介入保护, 避免不同线程访问数据时造成的不稳定问题。

2.3 功能组件划分

功能组件在软件系统功能设计阶段完成后, 各软件配置项功能既定的情况下, 开展各软件配置项需求分析, 进而实现功能划分, 识别其中可重用功能, 通过组件标准化实现并扩充至组织资产。可重用构件应尽可能地只规定所需的功能和性能指标, 详细描述构建的接口、配置参数、功能范围等, 允许开发者决定操作上和实现上的细节, 提高构件的可重用性。

2.4 功能组件实现

构件实现需以设计阶段输出为标的, 对标功能、性能指标开展实现, 并考虑提高每个构件的可复用性。重用构件的质量是开展过程中的重要一环, 每个构件作为独立的产品进行测试。对构件的测试也作为构件的一部分来管理。可重用构件实现完成后必须进行较通常软件开发更加严格的代码审查、单元测试、组装测试和确认测试, 必要时开展第三方测试。

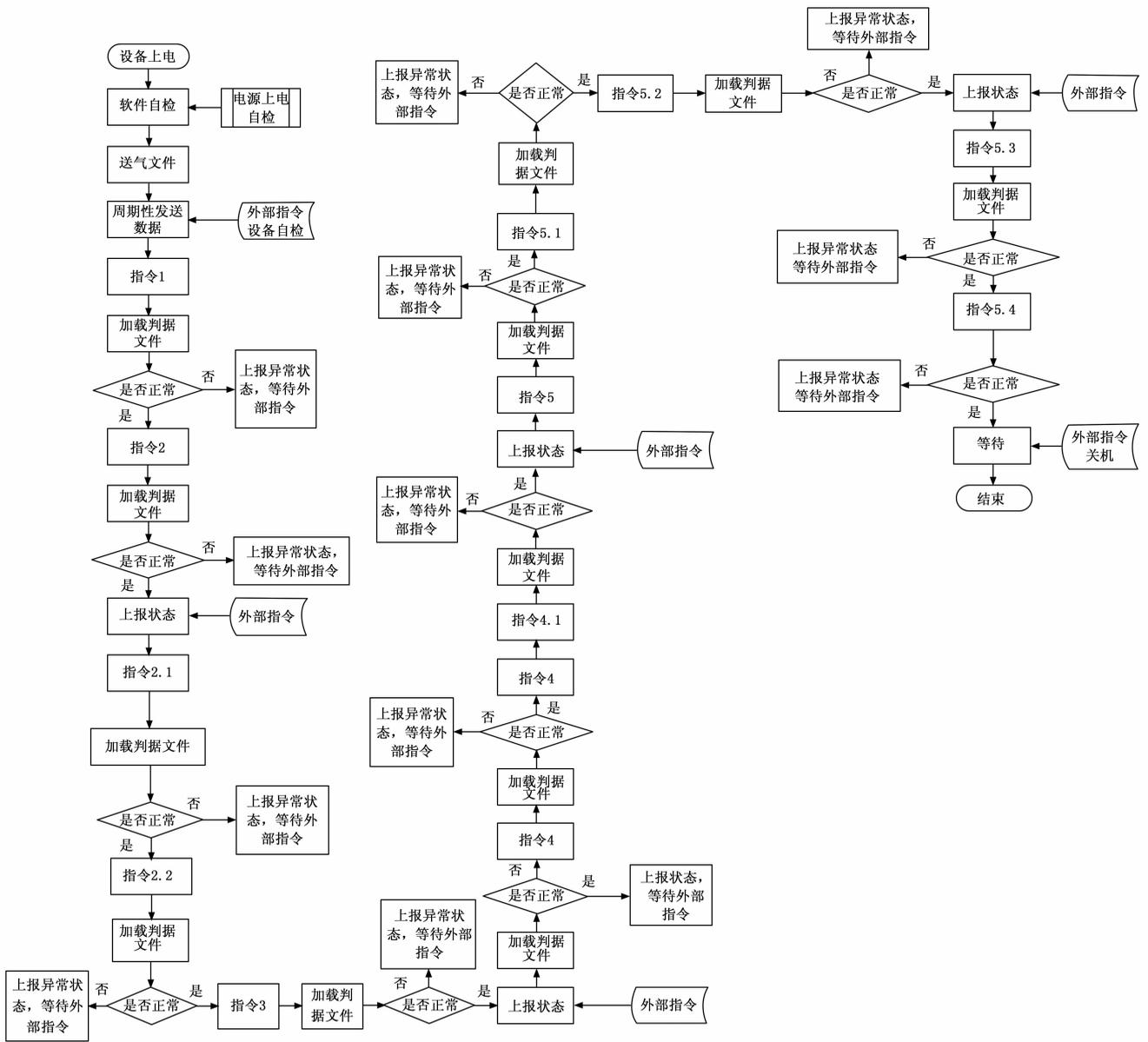


图 5 某监控子系统自动化控制流程

其中, 新研构件开发流程, 如图 6 所示。

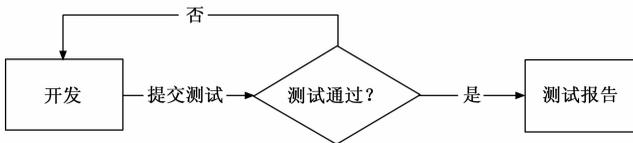


图 6 新研构件开发流程

重用构件研制流程, 如图 7 所示。

2.5 功能组件集成

在软件各功能组件研制完成后, 软件进入集成封装阶段。组件完成所有研制流程后进行入库, 入库前需准备详细的构件配套文档。如果构建研制随项目进行, 那么在项



图 7 重用构件研制流程

目结束时从项目工程文档中提取构件相关内容, 完成构件配套文档编写, 构件需根据接口规范, 生成标准文件。其中构件需维护标准接口如图 8 所示。

3 实验结果与分析

3.1 开发环境

当前, 主流国产操作系统以中标麒麟^[3-5]、银河麒麟为

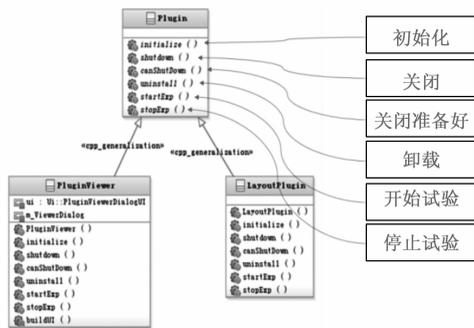


图 8 构件标准接口

代表。其中, 中标麒麟是国内首款支持龙芯 64 位平台的高安全等级可信的、自主可控的操作系统操作系统, 兼容主流的硬件和自主 CPU 平台, 如飞腾、龙芯、申威^[6-7]。与 Vxworks^[8-9]等操作系统不同, 中标麒麟兼容主流的硬件和自主 CPU 平台, 如飞腾、龙芯、申威^[10]。

目前中标麒麟可安装的应用开发环境有 Qt、Eclipse、GCC 和脚本语言^[11-13]。其中, Qt 是由 TrollTech 公司出品的可跨平台图形用户界面应用程序开发框架^[13]。其基于 C++ 语言的面向对象的框架, 既可以开发 GUI 程序, 也可用于开发非 GUI 程序, 目前被广泛应用于桌面应用、嵌入式、军工系统等有跨平台需求的、要与硬件交互的软件系统中^[14]。

本文采用中标麒麟 5.0 版、Qt5.6.2 版完成案例开发。

3.2 软件编译

软件开发过程中, 应完成需求分析, 设计功能组件组成, 区分可重用组件、新研组件; 可重用组件移植, 再识别软件可重用功能组件后, 需要将基于 windows 平台下编写的功能组件, 编写适用于 QT 环境的 .pro 文件, 在目标平台上重新编译, 实现组件的移植。总结如下:

1) 需求分析, 完成软件功能划分, 如有需要开发驱动加载至系统内核;

2) 使用 Qt Creator 等开发环境开发软件界面及逻辑代码实现, 由于中标麒麟目前支持集成开发环境较少, 当前主流集成开发环境为 Qt Creator;

3) 可执行文件或动态链接库 (.so) 文件, 软件完成开发后, 需将各功能组件编译生成可调用执行的文件, 同时生成软件的可执行文件, 打包发布软件。

3.3 软件发布

软件发布阶段, 需将用户程序编译后与所依赖的动态链接库、配置文件等文件一起打包生成可安装程序, 由于中标麒麟在 X86 平台、龙芯平台下编译生成的程序具备广泛通用性, 因此, 需分别在不同目标平台重新编译并分别发布软件安装包, 其主要工作内容 y 有:

1) release 文件文件生成, 复制 release 文件内所有文件至新建文件夹, 重命名, 并建立 pack.sh 脚本文件和 .h 文件。

2) 编写 pack.sh 文件, 主要为需要发布的程序名称和创建文件夹位置等信息, 例如:

```
! /bin/sh
Exe="mynewName"
Des="/mypath/mynewName"
Deplist="(lddexe | awk ' { if (match (3, "/")) { printf ("% s", 3}}')
cp deplist des
```

3) 在终端中运行 pack.sh 指令, 执行文件所依赖的链接库会自动复制到项目文件夹中。在其他龙芯平台上运行项目 .h 脚本文件即可启动软件。

3.4 效果分析

本文所述一体化软件设计方法, 能够在系统设计之初考虑软件配置项功能分配、接口定义, 在缩短系统设计反复、避免冗余迭代的同时, 缩短系统软件研制周期降低软件成本, 本方法在某项目中已成功应用, 相比 WPS 软件设计方法, 在相同功能点的情况下, 本项目中统计数据如表 2 所示。数据来源于其它型号类似软件研制过程数据, 其中, 代码缺陷率以第三方测试检出问题数量统计。

表 2 本项目相关数据统计情况

功能点	其它项目设计周期 (人/日)	本项目设计周期 (人/日)	其它项目代码缺陷率(代码错误/KLOC)	本项目代码缺陷率(代码错误/KLOC)
串口通信	10	6	1	0
网络通信	15	8	1	0
数据监测	15	8	2	0
指令接收	10	5	1	0
数据存储	10	8	0	1
日志存储	10	8	0	0
合计	65	43	5	1

4 结束语

本文所提出方法以系统需求为输入依据, 开展顶层需求分解, 实现了软件系统的功能划分, 通过接口设计、信息流设计进一步细化了软件功能点与系统的关系, 解耦各功能点的耦合关系, 实现了变更影响最小化设计, 提高了软件系统质量。从顶层设计到底层实现的设计思路, 解决了由于系统需求传递不准确导致的设计反复问题, 有效提升了型号软件研制效率。此外, 采用多维逐级分解的设计思路, 在进行系统设计的同时, 形成了软件系统自动化控制依据, 通过软件接口模块化、通用化设计解耦了不同流程判据不同的问题, 实现了测发控系统的自动化控制技术, 同时为丰富自主可控软件生态环境提供技术储备, 为未来基于国产软硬件平台的软件研制提供可借鉴案例。

参考文献:

[1] 白 芳, 郭宝锋, 李启瑞. 指控装备使用记录系统设计与实现 [J]. 计算机测量与控制, 2021, 29 (9): 215-219.

(下转第 156 页)