

基于 MVC 架构的通用型数据处理软件的设计与实现

崔北鹏, 张宝欣, 关凯元, 李坤龙, 陈昕

(上海机电工程研究所, 上海 201109)

摘要: 原始数据的解析处理是科研生产的重要环节, 也是试验分析与决策的重要依据; 针对型号研制数量、研制阶段层级、系统设备种类不断增多的现状, 文章基于 MVC 架构, 使用 C# 语言平台, 设计并实现了一种通用型数据处理软件, 能够分析处理不同来源的原始数据, 极大地提高了软件的通用化程度, 并减少了软件版本的数量; 同时, 增加了软件维护更新的速度, 降低了平台管理开发的成本; 目前已成功应用在多个型号任务中, 有力地支撑与推进了型号任务的发展。

关键词: MVC 架构; 数据处理; 翻译器; 数据库; C#

Research and Realization of Universally Data Processing Software Based on MVC Architecture

CUI Beipeng, ZHANG Baoxin, GUAN Kaiyuan, LI Kunlong, CHEN Xin

(Shanghai Electro-Mechanical Engineering Institute, Shanghai 201109, China)

Abstract: The analysis and reduction of raw data is an important part of scientific research, it is also the important basis of test analysis and decision. Aiming at the fast growth situations of the project quantity, project stage and equipment category, a universal data processing software based on MVC architecture and C# language is designed and realized, analyzes and processes the raw data from different universal sources, greatly improves the level of software generalization, and reduces the number of software versions, meanwhile increases the speed of software maintenance and update, decreases the cost of platform management. The result shows that the software powerfully supports and promotes the development of scientific research, which has been successfully applied in many tasks.

Keywords: MVC architecture; data processing; translator; database; C#

0 引言

科研生产各环节中产生的原始数据通过传感器被遥测终端接收到的实时数据, 反馈参试设备在各类试验的状态, 是科研决策分析的重要数据依据, 属于高成本价值数据资源; 但是原始数据不仅数量庞大而且十分耗费存储空间, 单个原始文件一般在百兆字节左右存储空间, 单机版解析软件生成的各参数文件一般在一至十几字节存储空间; 该类数据一般不能够直接用于科研分析, 需要经过一定的处理和解析; 然而, 随着型号业务的拓展、产品系列化快速发展、型号研制的数量激增、历史试验数据分析利用的需求越来越强烈, 不同型号、不同阶段、不同设备的数据文件若采用不同的软件进行处理, 将会大大增加软件版本的数量, 这样的处理方式给软件使用、管理和维护带来了极大的困难; 我们迫切需要一种通用化的、容易维护的、简

捷快速的、灵活部署的软件以适应当今和未来日趋紧迫数据处理任务形势。

虽然各类软件根据相同的通信协议进行编码, 但是由于不同软件解包方法、处理步骤和实现平台均不相同, 为了处理各个设备采集到的数据, 不仅需要部署不同的使用环境, 而且需要协调多个单位和部门的工作, 同时, 也增加了软件发布和维护的工作量。当前原始数据数据处理软件常见问题如下:

- 1) 现有原始数据及解析数据库没有统一有效的管理和规范, 历史数据不能正确的还原和解析;
- 2) 没有统一的数据分析工具, 存在分析对象不对等、数据溯源不便捷等情况;
- 3) 各类产品数据处理软件通用性差, 软件功能简单, 只能完成基础的参数曲线显示和数据文本输出, 不具备参数相关性计算;

收稿日期: 2022-07-12; 修回日期: 2022-08-22。

作者简介: 崔北鹏(1990-), 男, 安徽合肥人, 硕士研究生, 工程师, 主要从事导弹武器测试方向的研究。

引用格式: 崔北鹏, 张宝欣, 关凯元, 等. 基于 MVC 架构的通用型数据处理软件的设计与实现[J]. 计算机测量与控制, 2023, 31(3): 223-227.

4) 原始数据较为分散且缺失相应的分析工具, 设计师无法便捷地对多次试验、不同产品的同类参数数据进行横向、纵向比对分析, 数据共享性差;

5) 缺乏数据处理分析手段, 不能充分挖掘数据资源的价值;

6) 数据显示耗时较长, 不能满足设计师大批量、高频率的数据查询及分析需求。

1 MVC 架构及原理

MVC (即 model-view-controller) 是 Xerox PARC 在 20 世纪 80 年代为编程语言 Smalltalk-80 发明的一种软件设计模式, 提供了一种按功能对软件进行模块划分的方法。早期的 MVC 可以单独开发出数据库访问系统, 即在 Model 层就包含有数据库的读写操作。这种模式最大的优势就是可以为一个模型在运行时同时建立和使用多个视图, 变化—传播机制可以确保所有相关的视图及时得到模型数据变化, 从而使所有关联的视图和控制器做到行为同步^[1]。

MVC 架构的含义是将一个软件分为模型层、视图层和控制层 3 个相对独立的层次^[2]; 模型层主要包含数据处理的具体逻辑及算法、用于数据处理的类及方法, 视图层主要包括数据处理结果的显示、接收及响应用户的不同请求, 控制层主要用于控制软件的整体运行^[3-4], 从视图层获知用户的需求, 送入模型层解算, 再将解算的结果反馈在视图层中^[5-8]; 总之, 模型层是软件的运算中心, 视图层是软件的交互中心, 控制层是软件的总控中心, 各层次之间的关系见图 1。

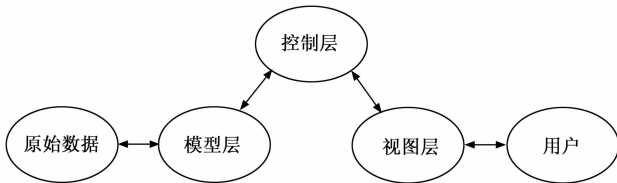


图 1 MVC 架构

通用数据处理软件采用 MVC 架构, 以控制层 Controller 为软件核心, 负责调配视图层 View 和模型层 Model 以实现软件的各个功能。其中视图层主要包含软件界面和数据显示模块; 模型层主要包括对数据文件的建模, 文件的参数数据库、特性关系数据库的建模 (数据库并非唯一选项, 亦可采用其他配置方式) 以及数据处理模块的建模; 控制层主要包括对上述的模块进行有效地调度和资源分配, 以及为提升用户体验所必须的处理过程的时间上的合理配置。

通过 MVC 架构可以将软件的功能最大限度的解耦开来, 每个层级专注于自身的任务, 针对不同的场景, 最大程度的复用软件的各个层级, 比如视图层可以使用 C#、QT、HTML 等不同环境开发, 模型层不需要任何改变, 更重要的是, 不同型号的原始数据, 通过整理与总结, 可以

化归为几种模型进行解算, 而不需要改变视图层, 如此, 大大提高了软件的扩展性、降低了维护的成本^[9-14]。

2 系统设计

2.1 总体设计

虽然科研生产各环节中产生的原始数据量较大, 但是各类产品测量的参数类别、参数特性等共通性较大, 具有较好的参考价值和借鉴性, 采用 MVC 架构可以高效的处理原始数据、管理解析数据库和灵活的显示交互。

基于 MVC 架构, 可以将模型层和视图层继续细分为翻译器、数据库连接器、关联器和绘图器, 如图 2 所示。从用户的角度看, 关联器是软件的输入, 绘图器是输出的结果, 进一步的, 关联器主要用于接受用户的关联指令, 明确用户需要分析哪些数据及需要的输出形式, 绘图器将分析的结果以图片的形式显示出来, 同时, 响应用户对显示曲线的缩放、对比、标记、统计等操作。

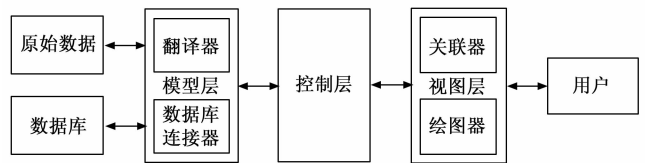


图 2 总体架构

模型层是软件的运算中心, 除了一些基础算法与数据结构, 它的核心是种类多样的翻译器与数据库连接器。数据库连接器是配合翻译器工作的模块, 它可以针对不同的翻译器, 找到对应的数据库, 并正确的提取其中的信息, 接着, 翻译器借助数据库中的信息对原始数据进行拆分、还原, 得到用户需要的结果。

控制层是软件的中枢, 它从关联器中得到了用户的需求, 送入对应的翻译器进行解算, 解算的参数从数据库连接器中获取, 最后将解算的结果反馈在绘图器中, 如此, 在用户、原始数据、数据库 3 个元素之间搭建起了结构清晰、适应性广、扩展性强的软件桥梁^[15-16]。

2.2 翻译器设计

根据目前的数据来源, 翻译器主要分为 422 翻译器、遥测翻译器、记录仪翻译器、CAN 翻译器等类型, 如图 3 所示。翻译器主要实现了数据预处理 (含时码修正、数据剔野等功能)、模拟量还原、数字量还原、解析结果标准化输出等功能, 可以适应不同来源的数据, 同种来源的数据若处理方法略有区别, 可以通过继承方法, 大大减少软件开发的代码量, 提高代码的复用性及维护的速度。

针对数据来源多样、硬件资源差异较大的情况, 采用抽象类的技术进行翻译器的设计, 根据 422 翻译器、遥测翻译器、记录仪翻译器、CAN 翻译器等具体翻译器抽象出基础翻译器; 基础翻译器主要实现 4 个功能: 数据提取、数据清洗、数据还原和标准输出; 其中, 数据提取是基础翻译器的一个抽象的方法, 具体实现在各类翻译器中,

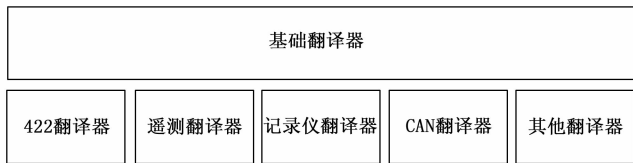


图 3 翻译器类图

提取好数据利用基础翻译器中通用方法进行数据清洗和数据还原, 最后再根据用户需求进行标准化输出。

2.3 数据库连接器设计

为了覆盖科研在用的数据库类型, 主要设计了 ACCESS 数据库连接器、MYSQL 数据库连接器和 XML 数据库连接器等, 如图 4 所示。



图 4 数据库连接器类图

在数据库连接器的设计过程中主要采用了标准数据库技术, 通过对模型层的进一步分析, 可以将模型层进一步细分为解析方法和解析参数两大类, 针对不同型号、不同阶段、不同设备的原始数据, 通过设计统一的标准数据库, 可以将模型层所需的解析参数细致的区分并存储下来, 这样一来, 在解析方法完备的情况下, 只需根据通信协议更新数据库, 便可以实现软件的快速维护。

标准数据库主要分为 7 个部分如表 1 所示; 其中, 设备表包含原始数据所含的设备类别, 浮动波道帧格式表定义了所有设备浮动波道的帧格式, 浮动波道占位表罗列了所有设备浮动波道在主帧中的位置, 数字量参数表和模拟量参数表分别存储了数字量与模拟量的还原系数、设备类型、数据类型等一系列信息, 帧结构表存储了主帧帧格式、码速率等信息, 帧标识表罗列了所有帧标识的类型和定义, 用于解析具有分时复用功能的复杂数据帧。设备收录下来的帧数据会存储在 .dat 文件中, 格式基本按照遥测设备研制任务书中关于子帧波道表的描述^[17]。

表 1 标准数据库

序号	表名	说明
1	Device	设备表
2	FloatFormat	浮动波道帧格式表
3	FloatPos	浮动波道占位表
4	FloatPara	数字量参数表
5	FramePara	模拟量参数表
6	Skeleton	帧结构表
7	SN	帧标识表

2.4 关联器设计

通过对用户的需求进行深入分析, 对于原始数据, 科研人员主要有数据检查、图线查看、批量输出等需求, 进一步的, 用户既需要软件具有原始数据品质检查的功能, 又需要针对部分参数能够快速的绘制曲线进行分析, 也需要软件能够一键化、标准化输出全部处理后的数据, 方便输入其他平台进行更加深入的分析。

为了满足用户的 3 种需求, 关联器实现了设置关联、预处理、批量输出、曲线显示 4 个主要功能, 其中, 设置关联明确原始数据和翻译器的对应关系, 预处理实现了原始数据的零点修正、时码修正和数据剔除, 同时生成预处理报告, 详细分析了数据的品质供用户参考, 批量输出则完成了选定数据的一键解析, 根据用户的选择生成对应的 TXT、CSV、DAT 等格式的文件, 最后, 曲线显示完成了参数的快速查询及曲线绘制。

为了后续型号、阶段提供可扩展配置的接口, 在关联器的设计过程中采用了反射技术。

反射是以自述方式表示系统的状态和行为。通常一个反射系统由两个逻辑空间组成基空间和元空间, 其中基空间是被反射的系统运行的空间, 元空间是对它进行反射的空间, 通过“具体化”, 系统从基空间映射到元空间, 通过“反射”, 系统从元空间映射到基空间, 利用反射技术在运行时重新配置系统的行为可以实现系统运行时对环境 and 需求变化的自适应^[18]。

模型层解析方法的构建是一个不断完善迭代的过程, 随着设备的新研、技术的更新, 软件可以快速的继承或者新研更多的解析方法, 而针对不同来源的原始数据, 通过反射技术, 可以从程序外部注入部分代码并灵活的配置软件所需要使用的解析方法, 这样一来, 和标准数据库组合使用, 就实现了软件的型号通用性、阶段通用性以及设备通用性, 即一个软件适应多个型号、多个阶段、多种设备, 并为后续型号、后续阶段、新研设备留有扩展配置接口。

2.5 绘图器设计

绘图器主要用于选定参数的曲线绘制, 同时满足用户的曲线交互分析需求, 基于事件响应互锁结构和数据压缩显示技术, 主要设计了鼠标滚轮缩放、鼠标拖曳放大、鼠标悬停、鼠标捕捉、自动幅值、数据统计、屏幕截图、屏幕标记、屏幕数据截取等功能, 功能全面, 操作灵活, 响应迅速, 极大的提高了软件的交互性及数据分析的效率。

由于原始数据的数量庞大, 现有数据处理软件的显示方法响应很慢, 不能满足各专业设计师的使用需求, 所以需要一种数据压缩算法进行数据的高效交互, 现有的数据压缩算法主要有: 王鹤等人提出基于分布式压缩感知和边缘计算的配电网电能质量数据压缩存储方法, 利用同步正交匹配追踪算法进行配电网电能质量数据压缩, 通过 K-SVD 字典学习算法对配电网测量值进行分析, 实现电能质量数据的高精度压缩; 赵会群等人提出基于密度划分的数

数据存储方法，通过密度区域划分算法降低高度重复数据的冗余度，利用缩 LZW 算法有效减少数据存储空间；常会丽等人提出的基于多维矢量矩阵的 MVC 多维数据压缩方法通过压缩编码处理变化后的系数矩阵数据，提升多维数据压缩效果；常会丽等人还提出的基于关系数据库的多维数据存储方法，是利用关系数据库存储压缩后的多维数据^[19]。

参考研究人员的成果，结合原始数据的特征，在绘图器的设计过程中创造性的采用了数据压缩显示技术，大大提高了显示的效率，提升了人机交互的效果。

如果直接进行折线图的绘制，由于实时采集的数据量很大，参数分离解析后，可能会有百万级别的数据量，这将给软件前端带来很大压力，显示刷新的速度会非常慢；经过分析可以发现，实际上真正参与绘图的点数不会超过显示的分辨率，进一步的，往往与显示的横向分辨率呈正相关，故当点集密度大于显示分辨率时，按照像素排序进行绘图，反之，按照点集排序进行绘图，由此，采用这种基于像素的图像压缩显示技术，可以大幅增进了视图层显示刷新的速度，也就是说，软件的反应速度、人机交互的效率得到了提高^[20-22]。

3 功能实现与验证

3.1 总体功能实现与验证

通用型数据处理软件主要在 C# 平台上开发实现，通过接口定义，明确了控制层、模型层和视图层之间传输的数据，利用工厂模式，在对应的接口下定义了控制层、模型层和视图层的父类，具体的逻辑与流程，通过继承与重写的方法，在子类中详细实现。

软件工作流程是：首先进行设置关联，添加数据库和对应的原始数据文件；其次进行预处理，对主帧序列进行校验、清洗和补正，并将异步嵌入的数字量帧从主帧中分离出来，并从主帧中提取对应时刻的时间；接着选择需要显示的参数；最后进行曲线绘制，在画图之前，程序会自动进行二次分离，将需要显示的参数从对应数字量帧中分离出来，并将提取对应时刻的时间；画图完成后，可以根据需要选择文本、图像等格式的输出，用于数据的进一步提炼与分析。

软件开发完成即在多个型号中使用，通过型号的具体运用很好的验证了软件的可用性与可靠性；其次，采用新的设计提高了数据解析的效率，增快了科研任务的进度；同时，采用统一的数据处理软件，大大减少了软件及数据库的维护时间；此外，在遇到实际使用的问题时，可以快速的进行举一反三，软件的健壮性得到很大的提高。

3.2 关联器功能实现与验证

关联器主要利用 C# 的 WinForm 平台设计实现，选用了选项卡、按钮、文本框、数值框、复选框、树形显示等前端控件，设计了按钮单击事件、树形显示单双击事件、复选框数值变化事件、选项卡切换事件等用户响应，原始

数据和翻译器的对应关系存储在树形结构中，既方便增减与遍历，又方便存储与读取，关联器的界面如图 5 所示。

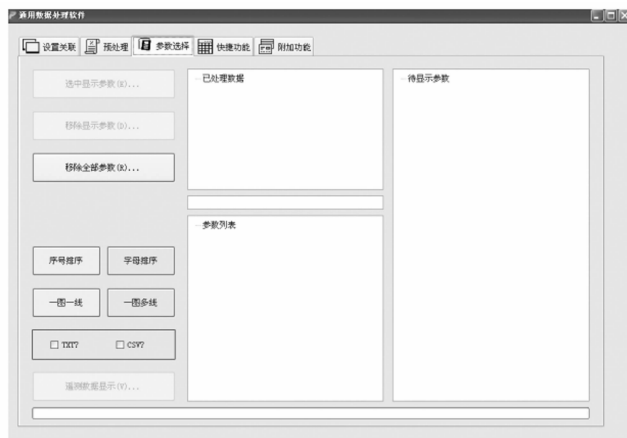


图 5 关联器界面

关联器开发完成后采用遍历方法对其功能进行验证，首先，针对一组数据，选择不同的排序方法、绘制方法和输出选项，进行图线显示与输出，显示结果、输出内容与预先设置符合，测试结果合格；其次，对快速检索栏进行测试，输入不同的关键字，参数列表可以快速有效的进行筛选显示，删除关键字，参数列表立即恢复初始状态，测试结果合格；最后，对添加删除参数进行测试，选择两组数据不少于 10 项的模拟量和数字量参数，分别通过按钮点击和鼠标控制两种方式进行操作，参数可以正常添加和删除，测试结果合格。

用户打开软件后，首先进入设置关联选项卡，添加数据库与原始数据，添加完成后可以直接进入快捷功能选项卡，这里可以进行原始数据的批量处理，在预处理选项卡中，可以检查原始数据的品质，并得到预处理报告，原始数据品质检查合格后，便能在参数选择选项卡中，查询并选中部分参数，点击曲线显示按钮后，关联器界面自动隐藏，绘图器界面弹出并绘制对应的图线，关联器工作流程见图 6。

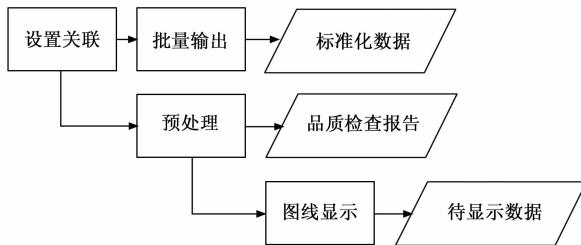


图 6 关联器工作流程

3.3 绘图器功能实现与验证

绘图器主要通过 C# 的 WinForm 平台和 GDI+ 绘图方法设计实现，关联器接收到图线显示指令后，在控制层的指挥下，先利用模型层解析原始数据，再将待显示数据显示在绘图器中，绘图器则用数据压缩显示算法，快速的将

图线显示在界面上, 绘图器界面如图 7 所示。

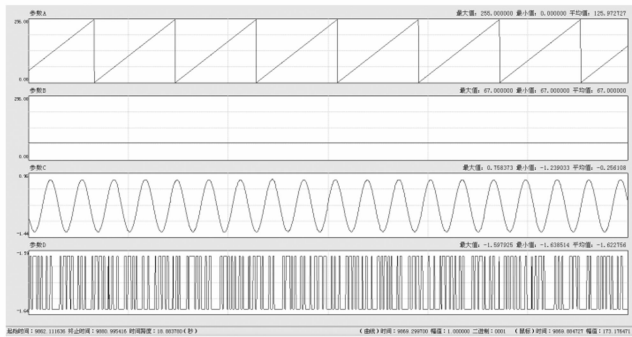


图 7 绘图器界面

绘图器有丰富的人机交互功能, 方便查看分析曲线, 如果科研人员需要缩放功能, 既可以通过鼠标滚轮快速横向缩放、也可以使用鼠标拖曳横向或纵向放大, 此外, 右键菜单中的自动幅值可以根据曲线的幅值范围, 自动将图像的纵轴改变到二八比例, 节省人员操作时间, 如果用户需要获取关键点的信息, 鼠标捕捉可以抓取曲线上离鼠标指针最近的点, 屏幕标记可以记录并和其他点进行比较, 最后, 使用者还可以通过屏幕截图或数据截取, 输出当前图像和点集, 用于报告的编写或深度数据分析。

绘图器开发完成后采用两组不同时间的长航时的飞行数据进行验证, 首先, 在同一组数据中分别选择不少于 10 项的模拟量与各路数字量参数, 依次进行显示验证, 验证结果合格, 可以正常快速显示; 其次, 在同一组数据中交叉选择不同通道的数字量及模拟量参数, 验证结果合格; 接着, 随机选择两组数据的模拟量及数字量参数进行复合显示, 验证结果合格; 最后, 软测人员选择不少于 20 项的模拟量和数字量参数, 通过快速移动鼠标及滚轮等操作进行压力测试, 显示情况正常, 没有出现死机、蓝屏等情况, 同时曲线刷新迅速且正常, 验证结果合格。

4 结束语

基于 MVC 架构的通用型数据处理软件开发完成后, 已成功运用在多个型号任务中。通用数据处理软件, 采用了统一的处理框架, 将复杂多变的数据处理流程化繁为简。同时, 模块化的程序架构也充分保证了对不同的数据有相应的处理方式, 并且通过软件反射技术能够进行迭代升级。一方面, 解决了多重版本所带来的问题, 有效降低了软件维护成本; 另一方面, 也使得数据处理的针对性更强, 更加细致, 能够从更深层次挖掘数据所包含的意义。科研人员利用该平台, 快速的处理与分析复杂多源的原始数据, 极大的提高了数据判读的效率和型号研制的进度, 提升了原始数据的效能, 丰富了设计师的分析手段, 为他们提供了更加全面和多维度的数据视角, 并为大数据技术在数据处理与分析方面的运用做好铺垫准备。同时, 运维人员受益于软件的通用性和易维护性, 节省了大量的人力和物力,

维护成本大大降低, 更新速度显著提高, 从而, 进一步的支撑了型号的快速推进, 助力型号研制迭代升级。

参考文献:

- [1] 杜成龙. 基于 MVC 模式的三层架构研究 [J]. 软件, 2022, 43 (6): 100-102.
- [2] 徐浩, 高正, 徐曼. 基于 MVC 模式的护理管理系统的设计和实现 [J]. 电脑知识与技术, 2016, 12 (30): 75-76.
- [3] 梁远. 基于 MVC 的地下水监测系统的研究 [D]. 淮南: 安徽理工大学, 2017.
- [4] 张勇军, 熊庆国, 黎学文. 基于 MVC 的项目管理系统设计与实现 [J]. 自动化与仪表, 2015, 30 (11): 58-61.
- [5] 吕猛, 张巍. 基于云平台与 MVC 架构的医院信息化综合管理系统 [J]. 自动化技术与应用, 2022, 41 (6): 148-151.
- [6] 张煜. 基于 MVC 架构模式的服务开通系统设计与实现 [D]. 南京: 南京邮电大学, 2019.
- [7] 尹强. 基于本体的 MVC 架构模式识别及重构技术 [D]. 南京: 东南大学, 2019.
- [8] 时慧晶, 刘凤伟, 陈亮. 水下航行器通用数据处理软件的实现 [J]. 计算机测量与控制, 2020, 28 (11): 131-134.
- [9] 高祝宇, 韩頔雨, 杨明, 等. 基于 Spring MVC 的气象预警信息 Web 系统设计与实现 [J]. 计算机与网络, 2020, 46 (12): 61-63.
- [10] 栾忠祥. 基于 MVC 模式的软件体系开发研究 [J]. 电子技术与软件工程, 2020, (20): 55-57.
- [11] 陈志毅. 大数据技术在遥测数据处理中的应用展望 [J]. 中国管理信息化, 2019, 22 (5): 156-157.
- [12] 朱丽梅, 王裕明, 岳铁林. 遥测数据处理软件的可靠性研究 [J]. 飞航导弹, 2004, (9): 36-38.
- [13] 史伟娜, 马卫东, 刘志, 等. 一种基于在线编译的遥测事后数据处理通用平台 [J]. 兵工自动化, 2017 (2): 61-65.
- [14] 马志刚, 刘文怡, 张文栋. 航天测试系统通用数据处理软件的设计 [J]. 微电子学与计算机, 2014 (11): 98-101.
- [15] 仇颖文. 通用导弹遥测数据处理平台的研究及应用 [D]. 上海: 上海交通大学, 2007.
- [16] 岳铁林. 通用遥测数据处理平台及外弹道估计方法研究 [D]. 哈尔滨: 哈尔滨工程大学, 2011: 15-26.
- [17] 王力文, 张国群. 遥测数据处理软件的通用化设计与展望 [J]. 兵工自动化, 2019, 38 (5): 41-43, 58.
- [18] 颜涛. 基于反射技术的软件动态演化方法 [D]. 吉林: 吉林大学, 2007.
- [19] 常会丽, 周金莲. 基于 MVC 架构的多维数据压缩存储方法研究 [J]. 计算机仿真, 2022, 39 (4): 279-283.
- [20] 苏静, 郑诗庭, 罗慧, 等. 基于 MVC 模式的随访系统设计与开发 [J]. 微电子学与计算机, 2019 (6): 93-95, 100.
- [21] 刘欣. 基于 MVC 模式的 Web 软件系统开发框架设计与实现 [D]. 济南: 山东大学, 2013.
- [22] 马文新, 龚美, 张庆维. 基于 MVC 模式的企业综合管理系统的设计与实现 [J]. 数字技术与应用, 2022 (4): 140-142.