

研
究
开
发CMD 系统地面综合应用
软件验证浅析杨 颖^{1, 2}, 金炜东¹(1. 西南交通大学 电气工程学院, 四川 成都 610031;
2. 株洲南车时代电气股份有限公司 技术中心, 湖南 株洲 412001)

作者简介: 杨 颖 (1977-), 女, 硕士研究生, 工程师, 从事机车产品质量控制方面的研发管理工作。

摘 要: 软件验证是提升软件产品可靠性、使软件满足用户需求的重要方法。按照CMD系统地面综合应用软件的高可靠性要求, 阐述了软件验证执行过程与软件验证用例设计方法的两方面内容, 回答了如何保证软件验证工作质量的问题。

关键词: 软件验证; 用例设计; 验证流程; 可靠性

中图分类号: TP311.52; U292.911⁺4 **文献标识码:** A **文章编号:** 1000-128X(2011)02-0009-03

Verification Analysis of the Ground Comprehensive Application Software
for CMD SystemYANG Yin^{1,2}, JING Wei-dong¹

(1. School of Electrical Engineering, Southwest Jiaotong University, Chengdu, Sichuan 610031, China;

2. Technology Center, Zhuzhou CSR Times Electric Co., Ltd., Zhuzhou, Hunan 412001, China)

Abstract: Software verification was an important way to enhance the reliability of software products and to assure software to fulfill customer's needs. In accordance with the high-reliability requirements of ground comprehensive application software for CMD system, the paper described the process of software verification implementation and the specific methods of designing test case in order to answer the question how to guarantee the effect of software verification.

Key words: software verification; test case design; verification process; reliability

0 引言

软件验证是软件生命周期中的重要阶段, 是软件质量保证(SQA)的重要子域。在软件验证需求和验证环境确定以后, 软件验证用例设计和流程控制就成了决定软件验证是否达到目的的最重要因素。由于CMD系统地面综合应用软件的高可靠性要求, 对软件验证提出了高要求。能否采用正确的验证流程, 同时深入、透彻地理解运用各种验证方法, 设计出好的验证用例是车载地面软件质量评测的关键。

1 CMD系统地面综合应用软件特点

CMD系统地面综合应用软件主要功能是把通过无线网络发到铁路内网的动车状态参数、动车安全信息

和综合监测信息交给数据处理中心负责存储和管理, 再由应用服务平台统一展示, 并实现系统维护, 因此, 系统可划分为数据处理中心和应用服务平台2部分, 前者负责数据的解析和入库环节, 后者负责基于数据库的各种地面终端应用。数据处理中心按照接受并解析的数据源的不同, 划分为实时数据处理和非实时数据处理2个模块。数据库的设计语言是基于ORACLE的PLSQL语言。应用服务平台根据功能可划分为电子地图、司机主屏、运行状态、设备状态、故障报警、专家诊断、在线统计、离线统计、预设置和用户管理10个模块, 设计语言为基于Eclipse平台的JAVA。

本软件数据处理量大, 响应时间要求严格, 系统承载负荷大, 用户日常操作频繁, 因此要求软件有较高的可靠性、可用性及安全性, 对软件验证也提出较高要求。

收稿日期: 2010-10-29; 收修改稿日期: 2011-02-24

2 软件验证设计

软件验证属于黑盒测试，完全不考虑软件内部执行过程和逻辑结构，只根据软件需求分析说明书中的外部功能及接口要求来设计软件验证用例，通过模拟最终客户使用，在尽量短的时间内尽量多地找出软件的错误，确保软件最终可交付。

2.1 流程简述

良好的验证流程是验证工作顺利开展的基础。按产品开发V模型，实际的验证流程如下:软件需求分析→软件验证计划及用例的编制→文档评审→搭建验证环境→第一轮验证执行→验证记录→出具缺陷报告→验证错误跟踪管理→设计修改缺陷→第二轮验证执行→出具验证报告→验证文档归档。

软件需求分析是由设计人员主导完成、验证人员参与的一个过程，是一个很重要的环节，它直接决定后期验证开展的细致程度，一般项目的软件开发需要逐步完善细化，因此验证计划与用例设计也是随着需求的细化而逐步细化的。

验证计划是验证的总纲，需要仔细描述验证依据、验证目标、验证约束、验证方式、验证资源及系统连接关系等内容。

验证用例的文档编写应按模板要求来进行，并要符合规范要求，每个具体测试用例都将至少包括下列详细信息：用例编号、需求编号、用例类别、验证目的、前置条件、验证步骤、期望结果和判断依据、验证级别和验证结果说明等。软件用例的操作步骤要简洁明了，具有很强的可操作性，便于非用例编写人的操作。

符合要求的验证环境有利于准确地测出软件存在的问题，并对问题做出正确的判断。需根据不同软件的运行环境要求，搭建符合该软件运行的环境。

验证执行及记录由2个人组成，一人进行验证操作，另一人进行验证记录。每日以验证日报的形式将验证内容及发现的缺陷记录下来，以邮件形式反馈给设计人员，对存在的问题进行及时沟通与确认，以便于修改缺陷。最终的验证报告可采用统计图表的方式，汇报用例执行情况、按功能点分布统计的缺陷及关键问题，使验证数据一目了然，如表1、图1。

表1 缺陷按功能点分布统计

序号	功能名称	缺陷个数	A类	B类	C类	D类
1	数据解析入库（模拟） - 主故障信息	8	0	8	0	0
2	数据解析入库（模拟） - 实时运行信息	23	0	23	0	0
...	...	...	...	...	...	...
统计	...	...	...	...	...	...

2.2 验证用例设计

一般软件验证设计方法有正-反向功能逻辑验证法、边界值测试法、等价类测试法、因果图、判定表驱

动分析、正交试验设计和功能图分析7种。本文主要针对前3种方法在工程实践中的运用情况进行简述。

1)正-反向功能逻辑验证法

这种方法列举软件功能点的逻辑路径，分别从正向和反向2个方向进行覆盖验证，以找出设计的缺陷或遗漏点。比如软件需求中提出了用户管理功能，能对系统用户、部门及权限角色进行管理。以“新增用户”功能为例，建立功能逻辑如图2。

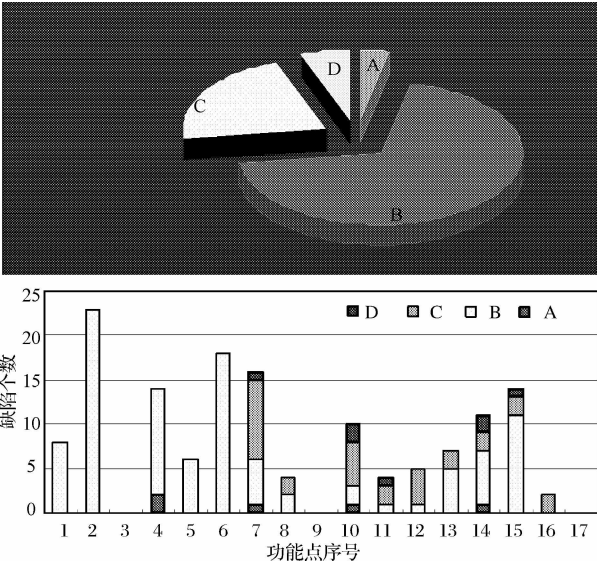


图1 缺陷在各个功能点中的严重等级分布

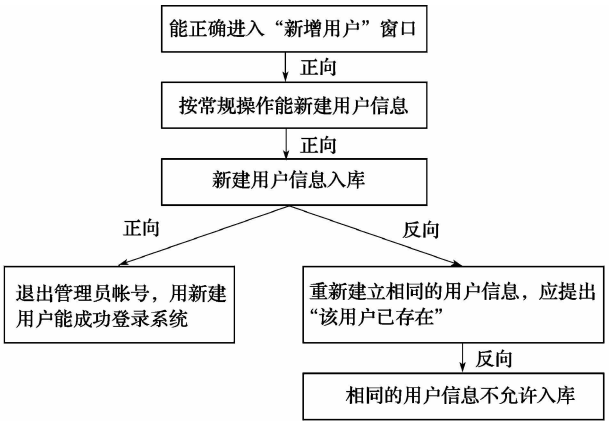


图2 功能逻辑验证图

按图2所示逻辑树，共编写5条验证用例，经过验证，正向逻辑没有问题，但在反向逻辑上发现2条B类缺陷：当重新建立已有的用户信息时，系统未报错，且该信息能入库。

2) 边界值分析法

本方法是考虑软件输入与输出真、假交界处的数值，选取边界的内、外、上多个临界值，进行黑盒验证，本方法具有很强的发现程序错误的能力，数值选取一般采用3点法：选取略小于、正常和略超过边界值的3个值。

实例：要求“累计运行距离”数据入库。数据协议中定义“累计运行距离”为4个字节，数据范围为

0~0XFFFFFFFF (0~4 294 967 295), 小于数据库字典允许范围 [number (10) 0-9 999 999 999], 数据入库时不会溢出。分别选取以下 6 个数进行验证: 0X00000000、0X00000001、0X8FFFFFFF、0X8FFFFFFF、0XFFFFFFFE、0XFFFFFFF, 发现当输入数据 $\leq 2\ 147\ 483\ 647$ (7FFFFFFF) 时, 输出正确; 当输入数据 $> 2\ 147\ 483\ 647$ (7FFFFFFF) 时, 输出为 -1。经过设计者检查, 是程序中该变量类型定义有误, 导致数据溢出。

3) 等价类测试方法及运用

本方法是把软件的输入数值划分为多个子集合, 每个子集合即为一个等价类, 在每个等价类中选取少数具有代表性的数据作为测试用例, 可分为有效等价类和无效等价类 2 种。

实例: 按协议传送的开机自检信息如果在正常范围内 (年份 ≥ 1970 , 月份为 1~12, 日期为 1~31) 应直接入库; 如果不在正常范围内, 数据库中记录为“ 1970-1-1 1:1:1 ”。确定输入和输出状态, 并将其进行等价类划分。输入变量包括年-月-日, 其等价类集合划分见表 2。

表 2

输入	有效等价类	无效等价类
年	1970~9999	$< 1970 ; \geq 10000$
月	1~12	$\leq 0 ; \geq 13$
日	1~31	$\leq 0 ; \geq 32$

3 总结

通过 CMD 系统地面综合应用软件验证, 发现软件存在的缺陷并进行修正, 有效保证软件可靠性。目前验证工作尚处于几种常见验证方法的简单运用阶段, 今后还需加强对验证技术的学习, 总结验证经验和方法, 通过设计科学合理的验证用例, 达到最佳验证效果, 同时也需遵循技术中心产品研发体系流程要求及相关验证原则, 确保验证过程规范高效, 提高软件可靠性。

参考文献:

[1] 刘 雅. 星载软件测试用例设计方法的研究 [J]. 航空计算技术, 2010 (5).

(上接第 4 页)

以同时查看主从车的数据。正常情况下, 主车一般可直接看仪表显示。二级显示下, 增加了与调试、检修相关的信息和交互界面, 如轮径设置、历史故障信息查看等。

6 几个关键的设计考虑

6.1 停车换端功能

TM3 机车为推挽式动车组编组模式, 列车折返运行时, 需要换端操作。为提高机车运用效率并保障机车安全, 列车网络系统设计了停车换端功能。司机在进行换端时只需按下停车换端按钮, 网络系统先通过 MVB 总线发送停放制动施加命令给 LCU, LCU 输出停放制动, 在接收到列车停放制动施加后, 系统进入停车换端工况, 此时主控端司机关断司机钥匙后显示屏换端界面开始倒计时, 同时受电弓、列供、劈相机等设备保持在正常工作状态。换端过程中在规定的 3 min 时间内, 任意一端重新闭合司机钥匙并闭合受电弓、列供、劈相机等琴键开关则完成换端, 再次按下停车换端按钮缓解网络系统因换端而输出的停放制动。网络停车换端逻辑控制功能在保证机车安全的前提下实现了机车无钥匙情况下保持相关设备正常工作, 减小了列车到达终点站后更换操作端的时间, 提高了机车折返的效率。

6.2 系统冗余处理

TM3 机车微机网络控制系统在三层总线的通信、

各控制设备硬件及相互协同工作等方面均做了完备的冗余设计考虑, 对防止机破、保证机车按运行图正常运行是必不可少的。首先, 列车级 WTB 网络、车辆级 MVB 网络均有冗余热设计备份。TCU 单元采用 A、B2 组冷备, 功能控制级总线及功能单元均有备份。LCU 等控制设备硬件也采用 A、B2 组冷备方式。这样保证了通信功能及控制设备的可用性。此外, CCU 单元中 DXM 模块进行了主要操作命令的冗余采集, 这样, 当主车 LCU、TCU 单元彻底瘫痪故障时, 司机还可以操纵从车进行降级运行。同时保证了在主从车均可以实施紧急制动的功能。

7 结语

TM3 型机车微机网络控制系统已在 2009 年底完成了施工设计及全面试验, 并完成了全部 42 套产品的制造交付。在 2010 年已全面投入运用, 产品运行非常稳定。这是基于公司构建的适应多需求最优控制系统平台进行高效组合应用开发的成功范例, 推动了电力机车控制技术的发展, 满足了铁路运输运载设备的需要。

参考文献:

[1] 周桂法. SS8 型机车微机控制和部件诊断系统 [J]. 机车电传动, 1995(4).
[2] 严云升. TM1 型机车的微机控制系统 [J]. 机车电传动, 1997 (6).