

基于黑盒测试的航天嵌入式软件用例设计方法

林琳

(北京控制工程研究所, 北京 100190)

摘要: 为了提高测试用例设计的效率, 简化测试用例数量, 提出了基于黑盒测试的航天嵌入式软件用例设计方法。阐述了确认测试过程中实用的测试用例设计流程, 提出了在设计测试输入数据的基础上加强测试逻辑设计的用例设计方法, 并结合实际工程经验, 给出了测试用例的总体设计思路。最后, 对测试遗漏问题及其原因进行了分析, 为今后进行更有效的星载软件测试提供了参考。

关键词: 嵌入式软件; 确认测试; 测试用例; 用例设计; 测试逻辑

中图分类号: TP311.5 **文献标识号:** A **文章编号:** 1000-7024 (2012) 06-2272-05

Design method of aerospace embedded software test cases black-box test based

LIN Lin

(Beijing Institute of Control Engineering, Beijing 100190, China)

Abstract: To increase the efficiency of test case design and reduce the numbers of test case, a design method of aerospace embedded software test cases based on Black-box test is presented. The test case design process applied during the validation test is discussed. A way of enhancing test logic design based on input data design of test case is pointed out. And the whole concept of designing the test cases is presented in combination with author's practical project experiences. Finally, the test omission problems and their causes is analyzed, which provides the reference for the further more effective test of spacecraft software test.

Key words: embedded software; validation test; test case; test case design; testing logic

0 引言

由于卫星等航天工程不仅涉及大量的投入, 而且担负着重要的使命, 因此运行其上的星载嵌入式软件必须具备非常高的可靠性和安全性。在这种情况下, 星载嵌入式软件的质量直接决定了整个卫星产品的质量, 它已成为航天系统质量的重要构成因素^[1]。

嵌入式系统的发展趋势是: 硬件逐渐向通用性平台过度, 依靠软件来完成系统的各种任务。软件已成为系统成败的关键性因素^[2]。为了保证系统的稳定性, 避免由于其可能出现的失效而导致灾难性的后果, 要求对嵌入式系统, 包括嵌入式软件进行严格的测试、确认和验证^[3]。实时嵌入式软件在时间和空间上的约束比较严格, 被测软件一般具有实时性、并发性等特点^[4], 测试此类软件是计算机软件测试中比较困难的问题。

软件测试是软件开发流程中的重要环节。而测试用例是软件测试的核心, 其组织性、功能覆盖性、重复性的特点能够保证测试功能不被遗漏。由于测试用例往往涉及多重选择、循环嵌套, 不同的路径数目可能非常大, 所以必

须精心设计使之能达到最佳的测试效果。设计测试用例需要测试人员花费大量精力去熟悉需求, 并在需求变化时更新用例, 占了测试周期很大一部分时间。本文希望能够总结以往的经验, 提出一个行之有效的测试用例设计流程以及一些基于黑盒测试的用例设计方法, 以提高测试效率, 从而降低软件缺陷遗漏率。

1 确认测试

航天型号软件特别是嵌入式控制软件目前一般采用瀑布式或增量开发生存周期模型。该模型包括的测试活动有: 单元测试、集成测试、确认测试和系统测试^[5]。其中确认测试作为软件开发方的测试, 是由相对独立于设计编码组的软件测试组进行的, 是软件交付系统的重要保证。

确认测试属于动态测试范畴。确认测试内容包括: 在模拟环境(开发环境)下, 运用黑盒测试方法, 验证所测软件是否满足需求规格说明列出的功能和性能需求^[6]; 保证软件配置所有成分齐全, 各方面质量都符合要求, 具有维护阶段所必须细节。通过准则为满足软件需求规格说明中规定的所有功能、性能等要求。

收稿日期: 2011-06-21; 修订日期: 2012-01-28

作者简介: 林琳 (1979-), 女, 浙江金华人, 硕士, 工程师, 研究方向为嵌入式软件测试技术、故障诊断等。E-mail: linhuada@163.com

目前，我们的确认测试是在一套自主研发的星地联试设备（实时半实物的仿真环境）上进行的，其核心任务分为需求分析、用例设计和用例执行。软件测试的效率和效果主要取决于用例设计^[7]，有效的测试用例可以定位以往未发现的软件错误，从而提高测试质量。

本文主要探讨确认测试中的用例设计。

2 测试用例的设计

以航天嵌入式软件中的星载控制软件为例，它的主要功能是实现对卫星的姿态和轨道控制、故障诊断，以及其他分系统的通讯等，数据流/控制流复杂。对于太阳同步中低轨道小卫星来说，由于可测控弧段限制，对星载控制软件的自主性要求很高。为了确保卫星的安全可靠，必须在测试中尽量覆盖可能出现的所有情况，而不仅是做到功能点的覆盖。众所周知，由于软件需求更改频繁、软件逻辑路径的组合性、输入数据的大量性及结果多样性等因素，哪怕是一个极简单的程序，要想穷尽所有逻辑路径、所有输入数据和验证所有结果都是非常困难的。因此，进行合适的测试用例选取是做好测试工作的关键。

2.1 用例设计流程

测试用例是为某个特殊目标而编制的一组测试输入、执行条件以及预期结果，以便测试某个程序路径或核实是否满足某个特定需求^[8]。其具体内容详见参考文献^[9]。

确认测试用例的设计是基于需求的，主要依据软件需求规格说明，并参考用户需求、任务书、接口协议等文档。用例设计的前提是进行详细的测试需求分析，吃透功能点，才能保证测试用例的覆盖性和正确性。测试用例的设计是测试需求细化的过程。具体的测试用例设计流程如图 1 所示。

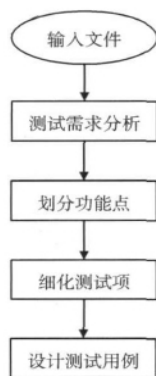


图 1 测试用例设计流程

首先，在输入文件到位后，开始进行测试需求分析。初步了解软件任务书、用户需求及相关文件，对软件功能进行合理划分，列出被测软件的主要功能大项，如初始化、模式切换功能、姿态控制率计算等。其次，对功能大项进行逐步细化和分解，直到每个被测试的功能点可以详细定

义和组合。最大限度地增加发现更深层次问题的可能。进而，根据细化后的测试项采用各种方法进行用例设计。然后建立详细的跟踪矩阵，以便对测试覆盖性进行检查，有效避免测试中的功能点遗漏。

按照这一思路可以进行测试复用。在测试实施过程中，直接复用成熟的测试用例所付出的代价要远远小于重新开发测试用例^[10]。对于不同平台的星载软件，可从功能点级别进行提取，找出相同或相近的功能点；对于同平台不同卫星的星载软件，可复用到测试项或测试用例；对于继承型号的星载软件，可直接复用具体的测试用例。如果将大量的测试用例存入测试用例库中，经过合理的分类，供测试人员选择使用，将极大地提高软件问题的发现率^[11]。该思路可以为开展测试用例复用工作，搭建通用测试用例库提供用例分类依据。

2.2 用例设计方法

测试用例体现了完整的测试过程，因此测试用例的设计方法依据测试类型分类。确认测试是以黑盒测试为主的测试类型，因此在测试用例设计时，必须选择适合的用例设计方法，才能保证在有限时间内，尽量多发现软件问题，提高软件质量。

设计确认测试的测试用例主要按照黑盒方法，具体方法见参考文献^[8]。但是，把这些方法统称为测试用例设计方法是不准确的，这些方法只是用例设计中如何确定测试输入数据的方法，不能包含用例设计的全部内容。测试用例设计不仅是确定输入数据的过程，还包括如何根据测试需求、设计规格说明等文档确定测试用例的设计策略、设计用例的表示方法和组织管理形式等^[12]。特别是对于航天嵌入式的星载软件来说，由于实现的控制逻辑非常复杂，仅考虑输入数据的设计是不够的，还必须针对实际飞行任务中可能发生的各种情况以及输入文件的具体要求设计不同的测试用例。只有全面考虑各个逻辑分支，才能保证测试覆盖性。

确认测试中的用例设计主要是设计测试输入数据和测试逻辑。

2.2.1 测试输入数据设计

关于测试输入数据的设计方法，本文中就不逐一列举了。在实际测试中，要从大量的输入数据中精心挑选出少数有代表性的测试数据，使得采用这些测试数据能达到最佳的测试效果，并高效地把隐藏的故障揭露出来是软件测试的核心和关键。不同测试数据对发现软件缺陷的能力差别很大^[13]，常需综合使用各种方法以有效提高测试效率和测试覆盖率。

首先使用边界值分析方法，此类测试用例往往更易发现程序错误。其次，进行等价类划分，针对输入和输出条件给出正常等价类和异常等价类，从而大大降低用例个数。第三，如果程序的功能说明中含有输入条件的组合情况，

可选用因果图法。第四,采取功能图法,通过不同状态下条件的有效性设计不同的测试数据。另外,根据以往的经验,可针对容易出现的错误追加一些测试用例(即错误推测法)。在功能性测试用例设计中,我们常需要把 80% 的精力投入到那些边界情况和失效数据作为输入的测试情况^[14]。

2.2.2 测试逻辑设计

在确定了测试输入数据的基础上,需要进一步进行测试逻辑设计。本文中测试逻辑是指在设计测试用例中的执行步骤时,应考虑的逻辑关系。通常,我们可以根据功能点的复杂程度按功能和路径分析混合模式设置用例。

在确认测试中,对简单的功能点来说,按功能测试最简捷,即按需求规格说明定义的功能点遍历测试每一个功能点。对于复杂的程序模块,其数个功能的作用是相互影响、紧密相关、环环相扣的,可以演变出数量繁多的变化。没有严密的逻辑分析,产生遗漏是在所难免。路径分析是一个很好的方法,其最大的优点在于可以避免漏测试。这里的路径分析法与白盒测试中的路径测试不同,它侧重于功能点的路径,而不关注程序的具体实现,路径数据相对较少。如模式控制功能,对于中低轨道小卫星来说,控制的自主性很强,一个工作模式包含多种定姿和控制率算法,以及选用不同敏感器和执行机构的组合情况,在设计这类功能点的用例时,测试逻辑就是至关重要的了。

所以,我们在设计用例时应更多关注测试逻辑设计。

首先,可以按照时间顺序如星上时或者事件驱动顺序来设计测试步骤,在模飞(模拟实际飞行程序)测试中通常采用这种方法;其次,根据用户需求中相应模块的逻辑分支来进行测试逻辑设计,通过路径分析的方法尽量覆盖所有分支;一般,主分支(即正常情况下运行的分支)被测试的概率很大,错误更有可能存在于那些程序运行时很少进入的异常分支。另外,还要考虑时序因素,数据的时间安排以及处理数据的任务并发性。例如执行某功能时被其他事件打断的情况,以及不同功能之间的相互影响等。

2.2.3 用例设计思路

下面作者结合工作中的经验,阐述一些在实际型号工作中的用例设计思路以及用例设计中的注意事项。进行确认测试用例设计时,首先应该有一个总体的设计思路作为指导思想,然后逐层细化。主要应遵循的原则总结如下:

(1) 自顶向下

测试用例要覆盖需求规格说明中提出的所有功能点,包括功能需求、性能需求、接口需求等。同时考虑充分性,然后按照 2.1 节中的用例设计流程展开用例设计。

举一个简单的例子说明,如表 1 所示。通过测试需求分析得到“重要数据保护”这个功能点,对该功能点做进一步分解,可细化为 2 个测试项,每个测试项可以再分解出具体测试用例的设计要点,最后根据用例设计要点设计出完整的测试用例。

表 1 “重要数据保护”功能的测试用例设计过程

功能点名称		重要数据保护	
编号	测试项	用例设计要点	对应用例
①	重要数据三取二功能	1. 修改三取二变量中的一项 2. 同时修改三取二变量中的两项	略
②	重要数据保存	1. 正常工作时,定时保存重要数据的功能 2. 在各工作模式中发生计算机复位,取回重要数据时的处理 3. 在各工作模式中发生计算机复位,未取回重要数据时的处理	略

(2) 各有侧重

针对不同功能点和测试项,在用例设计时应有所侧重。对于不同的控制模式关注点也不同:如消初偏模式,应关注消除初始大角速度的能力;而轨控模式则更多地关注喷气时间的准确度以及进入或退出条件。作为嵌入式软件,在测试时还应关注与硬件接口的关系等。

(3) 设计策略

考虑用例的执行顺序与测试时间的约束;在逻辑清晰的前提下,用尽量少的用例覆盖测试需求(一个用例包括尽可能多的逻辑分支)。有些用例可同时执行,例如部分模式切换的用例可以在模飞测试中进行;另外,可以将需要在相同模式中执行的用例统一测试,这样能有效节约测试

时间。从而在有限的时间内进行更多测试。

(4) 关注细节

对于一个较复杂的大型软件来说,测试用例的数量庞大,详细的期望结果有助于测试执行人员更快更准确地完成测试工作。同时,某些细小的地方很容易被遗漏,如剔野、限幅和计数器清零的位置等,都要在用例设计时特别关注。

(5) 安全性可靠性正确性

航天软件对安全性可靠性要求很高,因此在验证软件正确性的基础上,必须包括可靠性测试和安全性测试。可以根据相应的测试需求设计用例,必要时借助相关测试工具。

(6) 有全局观

站在系统的角度思考问题,更容易发现高层次错误。

要做好软件测试, 不能只是简单照搬需求, 关键在于理解。对于星载控制软件, 涉及控制分系统各领域知识, 只有对这些知识都有相当的了解, 才能设计出高效的用例。在确认测试阶段, 常会有一些测试要以系统级行为为线索, 从系统的使用方式来设计测试用例, 强调对使用情况的覆盖。

2.3 用例设计实例

以中低轨道小卫星的正常运行模式为例。假设该卫星安装了惯性陀螺敏感器、红外地球敏感器和数字太阳敏感器, 定姿方式有3种分别为陀螺+红外(A)、红外+数字太阳(B)、陀螺+红外+数字太阳(C); 控制方式为轮控(D)、喷气控制(E)和磁控(F)。在设计模式控制功能的用例时, 首先按照输入文件中正常运行模式下定姿与控制的逻辑分支进行路径分析, 分解出正常及异常分支如图2所示, 然后根据路径分析结果得出用例的测试逻辑。在用例逻辑设计时, 执行步骤应遍历路径分析结果, 以确保功能得到完整的覆盖。

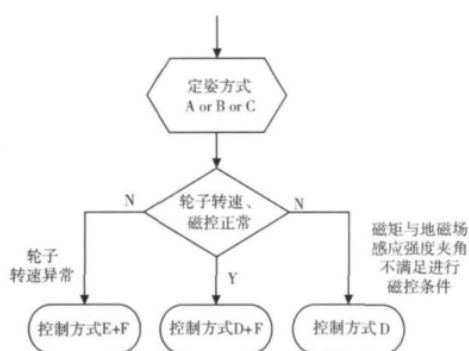


图2 正常运行模式定姿与控制逻辑

3 测试遗漏的原因及对策

我们都知道, 穷尽的测试是不可能的^[15]。软件测试永远不可能发现所有的错误。测试人员的责任就在于尽量避免人为因素造成的测试遗漏(即本该在该项测试中发现而未被发现的软件问题)。只有清楚地认识到测试遗漏发生的原因, 才能提高软件错误发现率和测试可信度, 最终确保软件产品的质量。

由于项目进度等客观条件限制, 测试人员只能有重点地选取测试用例, 不可能进行面面俱到的测试, 这就增加了错误被遗漏的可能性。在实际测试工作中用例选取不当的教训比比皆是。经过对多个型号控制软件的确认测试以及测试问题分析发现, 为了减少测试遗漏, 需重点关注以下几方面:

- (1) 测试中, 不能抱有侥幸心理, 没有测到的分支往往是错误高发的地方, 测试力度决定软件 bug 的发现数量;
- (2) 对于发生更改的软件来说, 错误通常集中在程序修改之处(包括新增功能)或者说是由更改引起的, 所以在测试中要重点关注。回归测试的影响域分析往往做得不

够, 因此在测试时要多考虑相关功能(错误原因可能是编码错误、修改考虑不全面或方案设计问题);

- (3) 加强数据库中试验数据的判读: 某些问题不易在动态测试时发现, 通过查看数据库, 往往能暴露一些意想不到的问题;

- (4) 测试中暴露出问题的代码往往隐藏着多个 bug, 经常出错的模块改错后还会经常出错;

- (5) 在软件研制过程, 特别是后续的版本升级中, 不少软件错误其实是相关的, 常由于当时分析不到位或修改不彻底, 导致将原本可解决的问题或者新的问题带入下一版本;

世界上不存在没有缺陷的软件, 对于软件测试人员来说, 发生测试遗漏是很难避免的, 重要的是不断提高发现问题的能力, 降低测试遗漏率。

4 结束语

本文对航天嵌入式软件的测试用例设计进行了探讨, 总结出了一套通用的用例设计流程和思路, 有利于在软件研制初期快速进行测试项分解以及开展用例设计。针对航天嵌入式软件的用途和较难维护、逻辑时序复杂等特性, 提出了测试输入数据和测试逻辑结合的设计方法, 能够有效减轻用例设计工作量, 提高测试效率, 有效减少航天嵌入式软件在测试过程中的错误遗漏。用例设计是一个不断改进和日趋完善的过程, 只有掌握了正确的测试用例设计流程并运用适当的用例设计方法, 才能达到事半功倍的效果。

参考文献:

- [1] CHEN Jiayu, XING Zhongbao. Research on on-board embedded software testing process model [J]. Optics and Precision Engineering, 2008, 16 (9): 1654-1659 (in Chinese). [陈佳豫, 邢忠宝. 星载嵌入式软件测试过程模型的研究 [J]. 光学精密工程, 2008, 16 (9): 1654-1659.]
- [2] DU Yan, LIU Congyue. Testing method for embedded realtime system software [J]. Control & Automation, 2006, (9-2): 31-33 (in Chinese). [杜延, 刘从越. 嵌入式实时系统软件测试实践 [J]. 微计算机信息, 2006, (9-2): 31-33.]
- [3] ZOU Yuehe, LIN Maosen, TANG Fei. Discussion of embedded software system test [J]. Electronic Product Reliability and Environmental Testing, 2007, 25 (5): 52-55 (in Chinese). [邹月和, 林茂森, 唐飞. 嵌入式软件系统测试综述 [J]. 电子产品可靠性与环境试验, 2007, 25 (5): 52-55.]
- [4] YANG Guanghua, QI Xuan, SHI Nansheng. Design of embedded software test cases based on scenario pattern [J]. Computer Engineering, 2010, 36 (15): 89-91 (in Chinese). [杨广华, 齐璇, 施寅生. 基于场景模式的嵌入式软件测试用例设计 [J]. 计算机工程, 2010, 36 (15): 89-91.]
- [5] Roger S Pressman. Software engineering-a practitioner's approach sixth edition [M]. Beijing: China Machine

- Press, 2009.
- [6] Andreas Spillner, Tilo Linz, Hanz Schaefer. Software testing foundations: A study guide for the certified tester exam [M]. 2nd ed. Beijing: Posts & Telecom Press, 2009.
- [7] LU Xiaoli, GE Wei, CHEN Xinli, et al. Designing a test case library system of supporting sharing and reusing [J]. Computer Science, 2006, 35 (5): 290-291 (in Chinese). [路晓丽, 葛玮, 陈新丽, 等. 支持共享和复用的测试用例库系统的设计 [J]. 计算机科学, 2006, 35 (5): 290-291.]
- [8] Glenford J Myers. The art of software testing [M]. Beijing: China Machine Press, 2006.
- [9] SHANG Dongjuan, HAO Kegang. A study of test cases and reuse in software testing [J]. Computer Technology and Development, 2006, 16 (1): 60-72 (in Chinese). [尚冬娟, 郝克刚. 软件测试中的测试用例及复用研究 [J]. 计算机技术与发展, 2006, 16 (1): 60-72.]
- [10] MENG Wei, LUO Shengxian. Study on feasibility of functional regression testing [J]. Computer Engineering and Design, 2009, 30 (1): 125-128 (in Chinese). [孟微, 罗省贤. 功能测试可回归性研究 [J]. 计算机工程与设计, 2009, 30 (1): 125-128.]
- [11] LIU Jie. Software testing and reuse technology [J]. Science & Technology Information, 2007, 4 (4): 209-210 (in Chinese). [刘杰. 软件测试与测试中的复用技术 [J]. 科技资讯, 2007, 4 (4): 209-210.]
- [12] LIU Bai, TANG Longli, CHEN Dasheng. The method research of testcase design based on requirement [J]. Electronics Quality, 2007, 28 (10): 61-63 (in Chinese). [刘柏, 唐龙利, 陈大圣. 基于需求的测试用例设计方法研究 [J]. 电子质量, 2007, 28 (10): 61-63.]
- [13] CUI Yingxia, LI Longshu. Integrated black-box test base on input/output relationship [J]. Computer Engineering and Design, 2007, 28 (23): 5581-5584 (in Chinese). [崔应霞, 李龙澍. 基于输入输出关系的综合黑盒测试方法 [J]. 计算机工程与设计, 2007, 28 (23): 5581-5584.]
- [14] ZHANG Yulin, XIE Kanglin. Test case design and reuse technology [J]. Computer Applications and Software, 2008, 25 (1): 100-101 (in Chinese). [张玉彬, 谢康林. 测试用例的设计和复用技术 [J]. 计算机应用与软件, 2008, 25 (1): 100-101.]
- [15] Ron Patton. Software testing [M]. Beijing: China Machine Press, 2008.

(上接第 2181 页)

- [2] WANG Jiangshao, YU Zong, LI Guang. Study of trusted chain technology of computing trusted [J]. Computer Engineering & Design, 2005, 29 (9): 2195-2198 (in Chinese). [王江少, 余综, 李光. 可信计算之信任链技术研究 [J]. 计算机工程与设计, 2005, 29 (9): 2195-2198.]
- [3] PCI-SIG. PCI Firmware specification 3.0 [EB/OL]. [2005-06-20]. http://www.pcisig.com/specification/conventional/pci_firmware/.
- [4] Compaq Computer Corp, phoenix Technologies Ltd, Intel Corp. BIOS boot specs [EB/OL]. [1996-01-11]. <http://www.scs.stanford.edu/nyu/04fa/lab/specsbbs101>.
- [5] Intel Corp. Intel® 64 and IA-32 Architectures software developer's manual volume 3a [EB/OL]. <http://www.intel.com/products/processor/manuals/>, 2011.
- [6] CHEN Wenqin. Bios inside [M]. Beijing: Tsinghua University Press, 2001 (in Chinese). [陈文钦. BIOS 研发技术剖析 [M]. 北京: 清华大学出版社, 2001.]
- [7] UEFI. UEFI specification version 2.3.1 [EB/OL]. <http://www.uefi.org/specs/>, 2011.
- [8] Trusted computing group. TCG. PC client specific TPM interface specification [EB/OL]. [2005-07-11]. <https://www.trustedcomputinggroup.org/home/>.
- [9] SUN Dayong. The applied research in the hard disk protective technology based on PCI bus [D]. Hefei: Hefei University of Technology, 2009 (in Chinese). [孙大勇. 基于 PCI 的硬盘保护技术的应用研究 [D]. 合肥: 合肥工业大学, 2009.]
- [10] Intel Corp. Intel® 64 Architectures X2APIC specification [EB/OL]. <http://www.intel.com/product-s/procesor/manuals/>, 2008.
- [11] Intel Corp. Intel® 64 and IA-32 architectures software developer's manual volume 3b [EB/OL]. <http://www.intel.com/products/processor/manuals/>, 2011.
- [12] Trusted Computing Group. TCG. TCG specification architecture overview version 1.4 [EB/OL]. <https://www.trusted-computinggroup.org/home/>, 2007.
- [13] Paul. Stoffregen. Understanding FAT32 filesystem [EB/OL]. [2009-10-20]. <http://www.pjrc.com/tech/8051/ide/fat32.html>.
- [14] SHEN Changxiang, ZHANG Huanguo. Research and development of the trusted computing [J]. Science China, 2010, 40 (2): 139-166 (in Chinese). [沈昌祥, 张焕国. 可信计算的研究与发展 [J]. 中国科学, 2010, 40 (2): 139-166.]
- [15] YIN Yongsheng, HU Yonghua, ZHOU Ganmin. Implementation of PCI expansion ROM mechanism [J]. Computer Engineering and Application, 2005, 41 (10): 101-103 (in Chinese). [尹勇生, 胡永华, 周干民. PCI 扩展 ROM 机制的实现方法 [J]. 计算机工程与应用, 2005, 41 (10): 101-103.]