

doi: 10.7690/bgzdh.2015.11.016

软件可靠性测试平台研究

谢 敏, 许丽星, 林文凤, 刘 志

(中国工程物理研究院电子工程研究所软件开发中心, 四川 绵阳 621000)

摘要: 基于可靠性测试需求设计并实现了一种可靠性测试的平台, 阐述了平台的硬件设计方案与软件设计方案, 该平台硬件层、模型层、数据层、协议层与应用层均可以根据测试需求现场定义, 平台的通用性较强; 平台能够基于软件剖面定义实现测试用例自动生成、测试数据自动收集与可靠性测试数据的收集, 为软件可靠性测试与评估提供了有力的支持。

关键词: 软件可靠性; 测试平台; 测试自动化; 实时平台

中图分类号: TP311.55 **文献标志码:** A

Research on Software Reliability Testing Platform

Xie Min, Xu Lixing, Lin Wenfeng, Liu Zhi

(Software Development Center, Institute of Electronic Engineering, China Academy of Engineering Physics, Mianyang 621000, China)

Abstract: Based on the requirements of software reliability tests, the article designs and implements a new testing platform and expounds the hardware architecture and software architecture in detail. The testing platform which has an ability of being generally accepted, since the hardware level, module level, data level, protocol level and application level can be defined locally according to different testing requirements. The testing platform can also automatically generate software test cases basing on the definition of the software profile, and automatically collect the testing data, so the testing platform can support the software reliability testing.

Keywords: software reliability; testing platform; testing automation; real-time platform

0 引言

随着电子系统中软件的复杂度越来越高, 软件的规模越来越大, 如何提高软件的可靠性, 已成为电子系统工程化研制过程中的关键问题^[1-2]。目前解决该问题的有效手段一方面是加强软件的优化设计, 另一方面则是加强软件测试^[1-2]。测试作为软件质量控制的关键环节, 通过将软件置于预期环境中运行, 能有效地考察软件对需求的满足情况。

传统的测试平台通常基于通用仪器或专用设备开发, 采用仪器监测被测系统运行, 实现对被测件的测试^[3-6]。这种平台开发周期长、成本高、复用性差, 无法实现对多源数据的耦合与在线分析。基于此, 笔者兼顾软件可靠性测试的需求, 基于主流的上/下位机模式设计并实现了一种软件可靠性测试平台的方案^[7], 上位机负责流程的编辑与管理, 下位机模拟仪器进行实时监测与分析。

1 软件可靠性及测试的概述

1.1 软件可靠性定义

IEEE 软件可靠性 (software reliability) 定义为:

在规定条件下, 在规定的时间内, 软件不引起系统失效的概率。IEEE 软件可靠性通常用软件的平均失效时间 MTTF (mean time to failure)、故障率 λ 与软件可靠度等指标进行描述。

1) 软件可靠度: 软件在一定时间内完成指定功能的能力; 假设软件可靠度为 R , 指定的工作时间为 T , P 为软件不失效的概率, 则软件可靠度可以描述为

$$R(t) = P(t < T).$$

可见软件可靠度可以描述为时间的概率函数, 可以通过软件运行的失效时间对软件的可靠性指标进行评估。

2) 故障率 λ : 假设软件从 $t=0$ 时刻开始工作, 软件中总共有 N 个故障, 第 n ($n \subset N$) 个软件故障发生在 T 时刻的概率 $F(t)$ 为指数分布, 则软件的故障率可以描述为

$$\lambda = \sum_{i=0}^N \lambda_i = \sum_{i=0}^N (F_i(T > t)).$$

3) MTTF: 软件在规定的时间内随机工作, 软件的工作总时间与失效时间之比, MTTF 也与软件

收稿日期: 2015-07-04; 修回日期: 2015-08-29

基金项目: 中国工程物理研究院质量与可靠性共性技术研究课题 (S2014ZK)

作者简介: 谢 敏 (1981—), 男, 四川人, 工学硕士, 助理研究员, 从事软件测试技术及其半实物仿真测试平台相关的研究。

的故障率成反比：

$$MTTF = \frac{1}{\lambda}。$$

1.2 软件可靠性测试及平台要求

软件故障率符合指数分布的前提是可靠性测试激励按照软件任务剖面随机生成，有效地记录软件的运行时间与失效时间，为可靠性分析提供数据；基于对软件可靠性度量指标的分析，进行软件可靠性测试对软件测试平台的基本要求如下：

1) 任务剖面的构建：是进行可靠性测试用例自动生成、对实现对任务剖面有效覆盖的前提；

2) 数据定义：数据定义是对软件任务剖面进行序列化数据描述的基础，通过有条件的约束实现对剖面数据域的有效覆盖；

3) 测试用例的自动生成：可靠性测试需要执行海量的随机生成的测试用例，自动测试用例生成是可靠性测试必须借助的手段；

4) 测试自动执行：执行海量的测试用例并有效记录软件测试的过程数据，尤其是失效时间数据，是软件可靠性数据分析的基础。

2 软件可靠性测试平台设计

2.1 可靠性测试平台硬件设计

为方便系统的扩展与升级，本平台采用主从式架构。平台包含测试主机、实时主机与被测设备，测试平台的硬件结构如图 1 所示，测试平台实物图如图 2 所示。

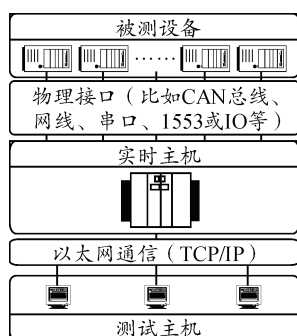


图 1 测试平台硬件结构

测试主机：基于通用计算机平台开发，运行非实时操作系统，用于测试平台资源管理与测试管理。

实时主机：运行实时操作系统，接收、解析与执行测试主机的配置文件、模型文件与脚本文件，进行实时通信、控制或数据采集。

测试主机与实时主机间采用以太网通信，测试

用例脚本由测试主机下发到实时主机执行，并将执行的中间数据与结果通过以太网上传到测试主机分析。

实时主机与被测设备间由真实物理接口互连，实时主机通过物理连接向被测设备发送真实的物理数据。

为了达到通用化的要求，实时主机配备常用的通信板卡，板卡特性见表 1。



图 2 测试平台实物

表 1 测试平台通信板卡列表

板卡类	板卡参数
串口	4 通道，支持 RS-232/422/485
CAN	2 通道 CAN 卡
1553B	双总线、双冗余，支持 BC、BM 与 RT 模式
A/D	64 通道、12 位、采样 1.25 MS/s
D/A	32 通道、16 位、D/A，16 路 VO，16 路 CO，电压范围 -10~10 V，电流范围 0~20 mA
I/O	96 通道，5 V TTL/CMOS 数字 IO 线 (20 mA)
继电器	32 通道，单刀双掷，单通道支持 220 V/5 A
以太网	2 通道以太网

2.2 软件可靠性测试平台软件设计

测试平台的软件设计方案根据数据路由关系共分为 6 层，如图 3 所示。

1) 应用层：用于测试平台的定义，测试用例生成与执行，测试数据的管理。

2) 协议层 1：运行于测试主机，用于将测试用例的数据按照自定义协议进行数据组帧与解析。

3) 数据层：用于下位机回传数据的缓存。

4) 协议层 2：运行于实时主机，将测试用例的数据按照自定义协议进行数据解析与组帧。

5) 模型层：运行仿真模型与测试模型，执行解析得到的测试数据序列。

6) 硬件层：根据测试序列数据执行物理动作，完成与被测件的交互。

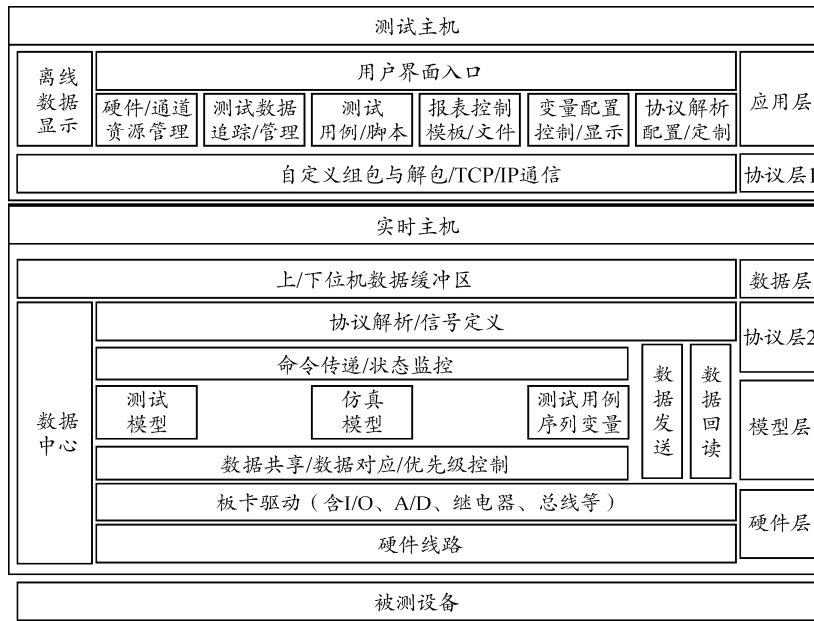


图 3 测试平台软件总体架构

2.3 平台的软件可靠性测试支撑

为了较好地支持软件可靠性测试，测试平台的上下位机软件均有针对性开展软件设计，实现测试工作的高效执行与失效数据的有效采集。平台中物理通道定义、数据定义、协议定义、模型定义与任务剖面定义的约束与绑定关系如图 4 所示。

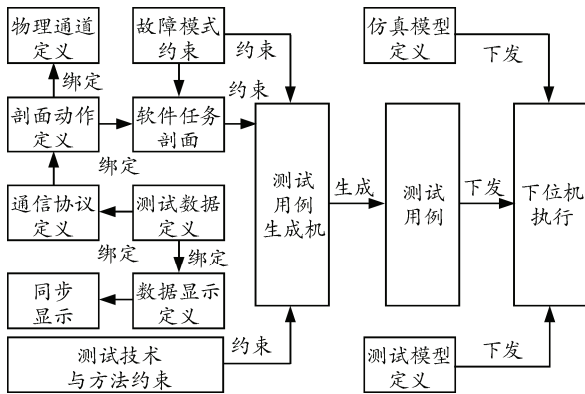


图 4 测试平台中的约束与绑定关系

基于该平台开展软件可靠性测试的流程：

- 1) 物理通道定义：根据与被测件的数据交互要求明确构建软件测试环境需要的硬件资源，并配置硬件的通信协议。
- 2) 数据(域)定义：定义测试平台中的支撑数据，即基于软件剖面的运行要求，各个动作节点需要收发数据或数据约束(当定义为数据约束时，测试用例生成机根据约束条件自动生成基于任务剖面的、覆盖数据约束要求的一组测试用例，实现可靠

性测试对测试数据域的有效覆盖)。

3) 异常/故障模式的定义：为一种特殊的数据定义，明确软件在任务剖面各个节点的故障模式与数据源要求；测试用例生成机能基于故障模式约束生成针对基于软件运行剖面的子剖面的一组测试用例，实现可靠性测试对软件运行多任务剖面与故障状态的有效覆盖；基于平台的测试数据定义与通信协议定义如图 5 所示。



图 5 物理环境、数据与通信协议定义

4) 协议定义：基于数据定义进行通信协议封装，将定义的独立数据与通信协议绑定，通过共享完整且匹配的通信协议实现上位机与下位机、下位

机与被测件之间数据组帧与解析的自动化。

5) 测试技术定义: 明确针对软件流程各节点欲采用的测试方法与技术, 如二分法、边界值法、故障模式法等, 测试用例生成机基于测试技术定义的约束生成软件测试用例。

6) 任务剖面定义: 构建软件任务剖面, 明确系统的运行流程要求; 某软件主要实现对系统初始化状态的检测, 通过发送自检命令, 接收系统的多种回令, 判断系统的状态参数是否正确, 基于平台的任务剖面定义如图 6 所示。

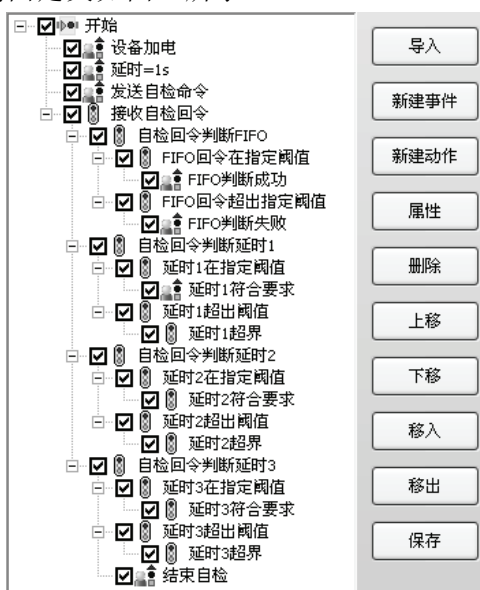


图 6 软件任务剖面定义

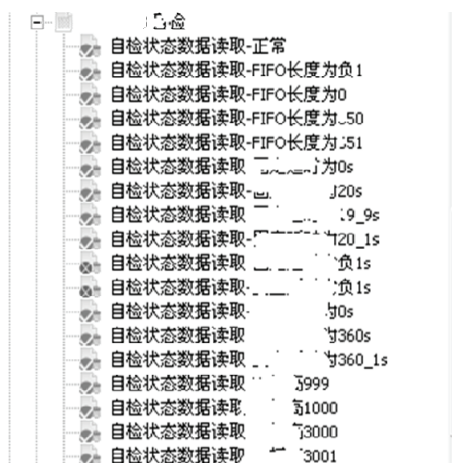


图 7 生成测试用例及执行结果

7) 测试用例生成: 测试用例生成基于软件任务剖面、测试数据定义、通信协议定义域软件测试技术约束, 生成软件测试用例组; 针对图 6 的软件任务剖面, 基于边界值约束生成测试用例如 7 所示, 自动生成用例实现了对几组参数的边界值进行了覆盖, 执行结果显示“×”的测试用例标志自动判断

未通过, 显示“√”标志测试用例自动判断为通过。

8) 软件测试执行与数据分析: 基于测试主机缓存技术, 实时主机能够将测试流程的完整数据上传到测试主机进行分析, 以便对软件测试失效数据进行分析与定位。

3 测试平台的关键技术

1) 开放的自定义数据/协议框架: 可以通过编辑文本的方式对复杂的通信/数据协议进行定义, 实现复杂数据组包/解析流程自动化, 极大地降低了数据源生成与测试数据分析的难度, 为可靠性测试的有约束的数据源奠定了基础。

2) 基于任务剖面、测试数据、故障模式与测试技术约束的测试用例自动生成技术: 兼顾软件需求、数据域与测试技术等多重约束生成测试用例, 为多维度的软件测试设计提供手段, 既可确保测试目的性, 又能兼顾可靠性测试用例的随机性。

3) 测试管理/执行一体化: 软件需求到测试报告的一体化管理, 且实现各个阶段的数据追溯与测试文档自动生成, 极大地简化了软件测试的分析与总结工作。

4 结论

该平台基于测试主机+实时主机+被测设备的方案进行设计, 已应用于多个软件的测试工作。实际应用结果表明: 该平台的硬件与软件均可以基于测试需求实现测试平台的二次现场定义, 通用性较强; 同时, 该平台支持基于约束的测试用例自动生成, 能够有效实现对软件任务剖面与数据域的覆盖, 支持测试数据的自动收集与测试结果的自动判断, 对软件可靠性测试提供了较好的支持。

参考文献:

- [1] 李建军, 邵培南, 段晓峰, 等. 软件可靠性测试及评估平台的设计与实现[J]. 微型机与应用, 2014(33): 10-22.
- [2] 岳晓果, 周瑞. 软件可靠性测试初探[J]. 软件导刊, 2008(11): 23-24.
- [3] 王轶辰, 刘斌. 计算机工程与运用[J]. 计算机仿真, 2005(6): 97-99.
- [4] 李明辉, 刘连生, 曲培树. 基于虚拟仪器的位标器自动测试系统[J]. 电子测试, 2008(3): 37-42.
- [5] 杨成. 基于虚拟仪器的航空电子设备检测系统设计[J]. 指挥控制与仿真, 2008(30): 106-119.
- [6] 谢敏, 罗永红, 张谊, 等. 基于虚拟仪器的可重构测试平台设计[C]. 系统仿真技术及应用学术会议论文集, 2009: 1-12.
- [7] 丁宁, 郑斌. 基于测试管理软件的测试平台软件架构技术研究[D]. 太原: 中北大学, 2014: 63-65.