

基于 MVC 模式的嵌入式软件测试开发环境设计

殷永峰,王轶辰,刘 斌

YIN Yong-feng, WANG Yi-chen, LIU Bin

北京航空航天大学 工程系统工程系, 北京 100083

Department of System Engineering of Engineering Technology, Beihang University, Beijing 100083, China

E-mail: yyf@buaa.edu.cn

YIN Yong-feng, WANG Yi-chen, LIU Bin. Design of embedded software testing development environment based on MVC. Computer Engineering and Applications, 2007, 43(7): 117-119.

Abstract: ESTDE(Embedded Software Testing Development Environment) is an important part of ESSTP(Embedded Software Simulation Testing Platform) and the foundation to effectively implement the real-time, close-loop testing of embedded software. As a typical architecture pattern for developing interactive software, MVC(Model-View-Controller) is applied to the design of ESTDE by analyzing the function, the composition and principles of ESTDE, not only ensuring the successful development of ESSTP, but also improving the reusability and the adaptability of the system.

Key words: software architecture; embedded software; software testing; pattern

摘 要: 嵌入式软件仿真测试开发环境(ESTDE, Embedded Software Testing Development Environment)是嵌入式软件仿真测试平台(ESSTP, Embedded Software Simulation Testing Platform)的重要组成部分,是有效实现嵌入式软件实时、闭环测试的基础。MVC(Model-View-Controller)是一种开发交互式软件系统的典型体系结构模式,在分析和研究 ESTDE 功能、组成和工作原理的基础上,将 MVC 应用于 ESTDE 的设计当中,不仅保证了 ESSTP 的成功开发,而且提高了系统的可复用性和可适应性。

关键词: 软件体系结构;嵌入式软件;软件测试;模式

文章编号:1002-8331(2007)07-0117-03 文献标识码:A 中图分类号:TP311

现代软件工程理论中,软件测试是满足软件需求、保证软件质量的重要手段。随着嵌入式软件在军用和民用高科技项目中的广泛使用,在其发挥巨大作用的同时,其可靠性也引起了人们的关注,针对嵌入式软件的测试越来越受到重视。对嵌入式软件进行系统测试,目前国内外公认的、行之有效的、具有广泛应用前景的方法就是软件仿真测试环境。

为了对嵌入式软件进行自动的、实时的闭环测试,嵌入式软件仿真测试平台一般包括宿主机 HOST 系统和目标机(TARGET)系统。运行于宿主机 HOST 系统即测试开发环境是整个仿真测试环境的重要组成部分,其作用是提供面向测试的软件系统,便于测试人员组织自己的测试。

在面向对象软件的分析设计方法中,模式的应用及其研究成为当前研究的热点,因为模式能够处理软件系统设计或实现中某些特殊的、重复出现的问题。嵌入式软件测试开发环境本质上是一个交互式系统,而 MVC(Model-View-Controller)模式是针对交互式系统的体系结构设计的比较典型的一种,它为交互式系统的软件系统的开发的应用框架提供了基础。

本文采用 MVC(Model-View-Controller)模式对嵌入式软件测试开发环境进行了分析、设计,为软件系统的开发提供了强有力的支撑,提高了软件系统的可复用性和适应性。作为仿

真测试环境的重要组成部分,有效的测试开发环境为实现针对嵌入式软件的实时、闭环的系统测试提供了保证。

1 MVC 体系结构模式简介

在面向对象的软件设计领域,模式的研究为那些特定设计环境中重复出现的问题提供了有效的解决方案。模式可以用来构建具有特定属性的软件体系结构。

模式的表达如图 1 所示,这种表达可以提供对模式进行描述的模板,是对模式进行描述的基础。

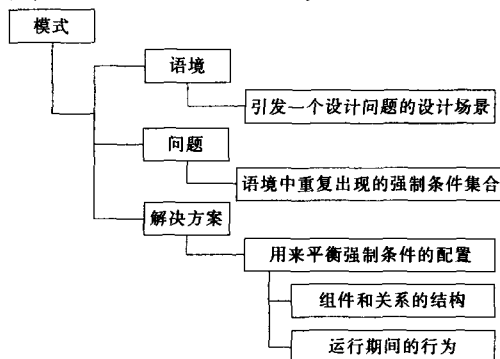


图 1 模式的表达

基金项目:“十五”国家部委预研课题(the Pre-Research Project of the “Tenth Five-Year-Plan” of China)。

作者简介:殷永峰(1978-),男,讲师,研究方向为软件可靠性,嵌入式软件测试等;王轶辰(1977-),男,在读博士,研究方向为软件可靠性,实时系统设计等;刘斌(1978-),男,副教授,博士后,研究方向为软件可靠性,嵌入式软件测试等。

当前大多数的桌面软件系统是交互式系统,这类软件系统通过图形用户接口来达到与用户的高度交互,便于增强系统的可用性,通过较便利的访问方法,能使得用户快速地学习应用程序并产生结果。

交互式系统应用内核通常是稳定的,而用户接口往往经常发生改变,即通常必须不断调整与用户业务过程的接口,因此内核独立于用户接口是交互式系统设计的重点。MVC(Model-View-Controller)模式,即模型-视图-控制器模式将一个交互式系统分为三个组件,即模型、视图和控制器:

(1)模型包含系统的核心功能和数据,它独立于特定的输出表达或输入方式;

(2)视图向用户提供了丰富的显示信息,它可以从模型获得数据,每个模型可能有多个视图;

(3)控制器用来处理用户的输入,视图和控制器共同构成了用户接口;

(4)通过组件之间的变更传递机制保证了用户接口和模型之间的一致性。

MVC 体系结构模式为交互式系统的软件系统的开发的应用框架提供了基础,它通过模型与用户接口的分离,可实现数据的多视图显示,同步化显示视图的更新,同时可实现运行期间用户接口对象的替换,为交互式系统的设计与实现提供了有效的解决方案。

2 嵌入式软件测试开发环境分析

为实现对嵌入式软件进行自动的、实时的闭环测试,嵌入式软件仿真测试平台分为两层,分层体系结构如图2所示。

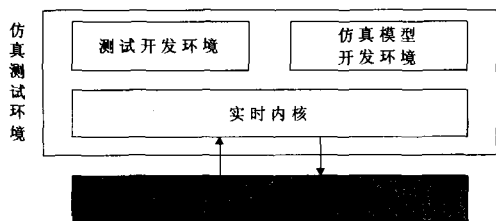


图2 嵌入式软件仿真测试平台体系结构

测试开发环境运行于宿主机 HOST,是面向软件测试的图形化、交互式的软件系统,测试人员可以根据被测软件的要求,通过测试前对系统的各种资源进行配置,组织被测软件的输入,来驱动被测软件运行,同时实现测试过程中测试数据的实时监控、显示以及与被测软件的交互,测试结束后能够自动保存被测软件的输出结果,从而对嵌入式软件进行自动的、实时的闭环测试。测试开发环境是仿真测试环境的重要组成部分。

仿真模型开发环境完成对目标系统交联设备的仿真建模。目的是建立被测软件的设备交联环境,模拟被测软件的真实运行环境完成必要的内部数据解算,通过接口仿真实现各交联设备模型与被测单元之间的数据交换。

实时内核是处理测试时实时数据的程序模块。主要任务是完成对测试任务的实时调度,测试数据的实时传递等功能。

通过分析,可以确定测试开发环境的主要功能应当包括如下几个方面:

- (1)测试文档管理;
- (2)测试脚本开发环境;
- (3)测试面板的搭建;
- (4)测试环境配置;

(5)测试数据分析;

(6)日志管理。

3 基于体系结构的嵌入式软件测试开发环境设计

基于以上分析,嵌入式软件测试开发环境是一个面向测试的交互式软件系统,下面给出基于 MVC 模式的测试开发环境的设计。

3.1 系统模型创建

模型的创建重点在于通过分析软件应用领域将核心功能从设想的输入和输出行为中分离出来,通过模型组件的设计封装内核所需的数据和功能。

根据对嵌入式软件测试开发环境的分析,可以得到整个环境的类图(如图3所示)。

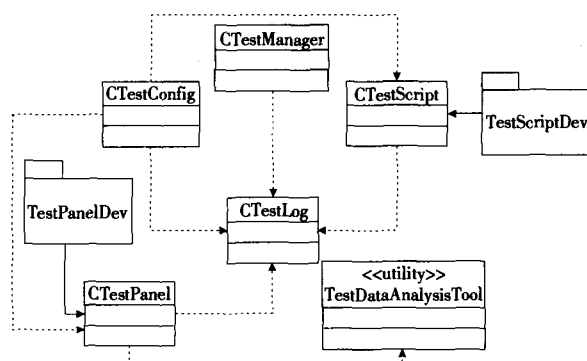


图3 嵌入式软件测试开发环境类图

测试文档管理类(CTestManager)的作用是完成对测试环境文档的分级管理,管理内容包括仿真模型接口文档、日志文件、测试配置文件、测试脚本文件、测试面板文件、测试结果数据文件等,便于测试人员组织和管理自己的测试。

测试配置类(CTestConfig)的作用是对整个测试开发环境进行配置。测试配置主要包括对被测目标系统的配置、仿真测试环境的时钟周期、测试中各种 I/O 的配置等,使得测试环境具有一定的通用性和可扩充性。

日志管理类(CTestLog),它贯穿于测试的全过程,其作用是通过测试后台运行的监控进程对测试前、测试中和测试后事件的自动捕捉,记录测试人员在测试各个阶段的活动,如各个配置、开发活动发生的时间、具体的细节描述等。

测试脚本类(CTestScript),封装了测试脚本中所支持的各种测试任务的链表,表中测试任务都包括测试任务的执行时间、任务的优先级特性和周期特性,以及测试人员所施加的测试动作序列的集合。测试人员根据不同测试用例要求,生成不同的测试脚本动作序列,这些测试动作序列的描述,可以由相应的处理程序编译为相应的测试指令,再通过编译器的实时解析就可以完成对被测软件自动化测试的驱动。

测试脚本开发程序包(ScriptDev),是为了测试人员更方便地生成测试脚本而提供的测试脚本开发程序包,可以支持文本和图形两种方式来生成测试脚本。它依赖于测试配置类,在生成测试脚本时需要测试配置的信息。同时,支持语法的高亮显示,以及提供对生成的测试脚本的语法进行检查。

测试面板类(CTestPanel),提供了用户在测试开发过程中,为便于监视、查看测试过程,采用图形方式建立的测试面板。内容包括数据的显示(如趋势图显示、LED 显示、信号灯显示、开

关量按钮等),同时也可以允许用户在测试过程中,实时地添加一些测试命令(如在线脚本),完成用户与被测软件的交互。

测试面板的开发程序包(PanelDev),是为了测试人员更方便地搭建测试面板而提供的开发程序包。通过在测试开发环境中提供一组封装的控件,允许用户采用图形化拖拽(配置)的方式来搭建测试面板。它依赖于测试配置类,在搭建面板时需要测试配置的信息。

应用程序(TestAnalysisTool)主要提供了对测试收集的结果数据进行分析的应用程序,可以实现测试过程的事后回放(依赖于测试面板类);测试结果数据的过滤、查询、比较;测试结果数据的图形化显示。可以减轻测试人员的负担,提高了测试结果数据分析的效率和准确性。

3.2 视图控制及变更传递机制的设计

从上述模型的创建可以看到,测试开发环境中的动态数据主要包括测试配置数据、测试脚本数据、测试面板数据、测试结果数据等。因此测试开发环境中主要视图应包括测试配置视图、测试日志视图、测试脚本视图、测试面板视图及测试(结果)数据视图。视图设计的原则在于独立于它们所要显示的数据,能够准确无误地显示模型的相关数据,同时还要响应控制器的用户输入,通过变更传递机制可以引起相关模型数据和显示视图的更新。

控制器的设计主要是通过即时捕捉用户界面所发生的事件(如鼠标、键盘等),即时响应并对相应的事件进行处理。

MVC模式中的变更传递机制是维护交互式软件系统的关键技术,是实现模型、视图和控制器之间数据同步的基础。

图4给出了测试开发环境中比较典型的测试脚本视图及变更传递机制的动态描述。

4 应用与结论

本文研究的嵌入式测试开发环境已在“嵌入式软件仿真测试平台”产品中得到了验证。嵌入式软件仿真测试单机平台已经成功地对几个航空某型号的航电系统嵌入式软件进行了测试。

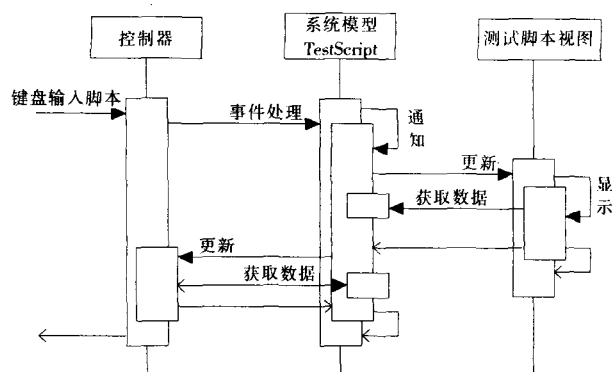


图4 典型变更传递机制的动态描述

本文提出的基于MVC(Model-View-Controller)模式的嵌入式软件测试开发环境的设计,为嵌入式软件仿真测试平台的成功开发提供了有力保证,提高了软件系统的可复用性和适应性。进一步的工作包括加强对测试开发环境中模型的封装,增强测试开发环境的可视化配置功能,以降低软件使用的复杂度,使测试人员更容易开发、配置自己的测试。

(收稿日期:2006年5月)

参考文献:

- [1] Fontoura M, Pree W, Rumpe B. The UML profile for framework architectures[M]. [S.l.]: Pearson Education Limited, 2001.
- [2] Tsai W T, Na Y, Paul R, et al. Adaptive scenario-based object-oriented test frameworks for testing embedded systems[C]//26th Annual International on Computer Software and Applications Conference, Oxford, England, 2002: 321-326.
- [3] 刘斌, 高小鹏, 陆民燕, 等. 嵌入式软件可靠性仿真测试系统研究[J]. 北航学报, 2000, 26(4): 490-493.
- [4] 殷永峰. 嵌入式软件仿真测试脚本技术研究[D]. 北京: 北京航空航天大学工程系统工程系, 2003.
- [5] Buschmann F, Meunier R, Rohnert H, et al. 面向模式的软件体系结构 卷1: 模式系统[M]. 贾可荣, 郭福亮, 赵超, 等译. 北京: 机械工业出版社, 2003.

参考文献:

- [1] Yamamoto Y, Yun Xiao-ping. Coordinating locomotion and manipulation of a mobile manipulator[J]. IEEE Transactions on Automatic Control, 1994, 39(6): 1326-1332.
- [2] Dong Wen-jie. On trajectory and force tracking control of constrained mobile manipulators with parameter uncertainty[J]. Automatica, 2002, 38: 1475-1484.
- [3] Block A. Control and stabilization of nonholonomic dynamic systems[J]. IEEE Transactions on Automatic Control, 1992, 37(11): 1746-1757.
- [4] Hu Y M, Chao H M. High order sliding mode control of nonlinear control systems with application to mobile robots[M]//Advances in Variable Structure Systems: Analysis, Integration and Applications. Singapore: World Scientific, 2000: 125-134.
- [5] Young K D, Zguner O U. Variable structure systems, sliding mode, and nonlinear control[M]. New York: Springer, 1999.
- [6] 吴玉香, 胡跃明. 二阶动态滑模控制在移动机械臂输出跟踪中的应用[J]. 控制理论与应用, 2006, 23(3): 411-415.
- [7] 吴玉香, 胡跃明. 轮式移动机械臂的建模与仿真研究[J]. 计算机仿真, 2006, 23(1): 147-151.

(上接22页)

$k_r=0.02 \text{ V}/(\text{rad/s})$, $k_T=0.191 \text{ N}\cdot\text{m}/\text{A}$, 齿轮减速比分别为 $\beta_1=71$, $\beta_2=51$ 。各设计参数分别取为: $K=\text{diag}[50 \ 50 \ 150 \ 1500]$, $W=\text{diag}[60 \ 60 \ 250 \ 2500]$, $m_1=\text{diag}[35 \ 35 \ 40 \ 450]$, $m_2=\text{diag}[40 \ 40 \ 45 \ 550]$ 。系统惯性参数的扰动满足: $\Delta M=10\%M$, $\Delta V=10\%V$, $\Delta G=10\%G$ 。仿真结果如图2所示。

从仿真结果可以看出, 本文所设计的鲁棒跟踪控制器成功地实现了轮式移动机械臂对给定轨迹的跟踪。即使在初始误差较大的情况下, 跟踪效果也很好。

5 结论

本文首先讨论并推导了考虑驱动电机动态特性的轮式移动机械臂简化三阶动态模型, 接着给出了系统惯性参数扰动时轮式移动机械臂简化动态模型, 然后采用滑模控制为其设计了鲁棒输出跟踪控制器并给出了具体的设计过程。仿真结果表明, 该控制方法能很好地实现轮式移动机械臂的鲁棒输出跟踪。(收稿日期: 2006年11月)