

基于 LoadRunner 自动化应用程序接口的舰船装备软件测试系统设计与实现

韩新宇, 何伟, 张凯

(中国船舶工业集团公司软件质量与可靠性测评中心, 北京 100081)

摘要: 针对舰船装备软件接口交互信息量大, 利用现有动态测试技术对此类软件开展测试工作时效率较低, 自动化测试水平亟待提高的现状, 归纳分析了舰船装备软件测试系统的主要能力需求, 研究了接口协议形式化、测试用例自动生成、测试用例自动驱动、通信接口转换等关键技术, 并以上述技术为基础给出了舰船装备软件测试系统的体系架构设计方案, 进而实现了一种基于 LoadRunner 自动化应用程序接口 (LoadRunner Automation API) 的舰船装备软件测试系统, 为舰船装备软件提供了一种可行的自动化测试手段; 该系统主要包括测试管理模块、测试用例生成模块、测试用例驱动模块和数据收集与分析模块; 采用该测试系统能够有效提高舰船装备软件的测试效率。

关键词: 测试用例; 自动化测试; LoadRunner 自动化应用程序接口; 组件

Automated Warship Software Testing System Based On LoadRunner Automation API

Han Xinyu, He Wei, Zhang Kai

(China Shipbuilding Software Quality & Reliability Testing Center, CSSC, Beijing 100081, China)

Abstract: In order to test warship software which has lots of information to communicate, main requirements of a general warship software testing system are analyzed, key issues including interface protocol formalization, automated test case generation, automated test case driving, and hardware interface conversion are investigated. Then, the architecture of an automated warship software testing system is proposed accordingly, and the system is implemented based on the LoadRunner Automation API. This system is comprised of four parts, which are test management module, test case generation module, test case driving module, data collection and analysis module. The practical results show that this system can greatly improve the efficiency for testing warship software.

Keywords: test case; automated software testing; LoadRunner automation API; component

0 引言

随着信息集成需求的日益增加, 舰船装备软件接口交互信息量越来越大。对于此类软件而言, 在动态测试过程中, 大量工作作用于测试数据的准备和驱动。传统的做法是, 测评机构根据接口协议说明利用软件研制单位提供的模拟器产生报文格式, 然后手工准备测试数据, 再利用通用的测试工具 (如 LoadRunner、各种调试助手) 作为数据驱动工具实施测试。由于研制单位提供的模拟器只能产生固定格式的正常报文, 无法产生异常、边界等测试数据, 且调试助手等难以模拟多个接口逻辑关系^[1], 测试人员往往需要自行开发数据生成和驱动工具^[2]。对于以软件测评为主要业务的专业软件测评机构而言, 这种做法费时费力, 难以形成通用的测试能力, 而人工编写测试数据, 工作量繁杂, 影响测试效率和测试质量, 因此迫切需要研制一种通用的舰船装备软件测试系统^[3]。根据本文研究内容设计并实现的舰船装备软件测试系统解决了接口协议形式化、测试用例自动生成、测试脚本的自动生成、接口转换等多项技术难题, 可针对被测对象做出快速的测试设计, 对整个测试过程提供统一的管理、设计、驱动和分析能力。

1 测试系统的需求

针对目前舰船装备软件测试过程中存在的问题, 测试系统的主要能力需求如下:

1) 接口协议形式化能力。测试系统应具备可视化接口协议编辑器, 实现对被测软件接口协议的形式化建模和编辑功能, 包括字段的数据类型、长度、最低有效位 (LSB)、范围等信息。

2) 测试用例自动生成建模能力。测试系统应具备可视化测试剖面模型编辑器, 能够结合接口协议形式化结果, 实现对测试步骤的定义。

3) 测试用例自动生成能力。测试系统应能够自动生成测试用例所需的测试数据和测试步骤, 通过遍历测试剖面中所有可能的测试路径, 结合路径上测试步骤的定义, 生成满足测试覆盖要求的测试用例, 且支持典型的执行逻辑 (顺序、周期、延迟、反馈、同步)。

4) 测试用例驱动能力。测试系统应能根据测试用例自动生成测试脚本和可驱动测试数据, 并完成多种接口逻辑关系下的信息交互, 覆盖舰船装备常用的接口类型 (如 RS232、RS422、RS485、CAN 总线以及以太网接口)。

5) 数据收集与监测能力。测试系统应提供可视化的图形用户界面, 完成测试数据收集方案配置功能, 确定数据收集的要求、方法、时机以及数据的分析功能 (包括数据的筛选、判

收稿日期: 2016-03-16; 修回日期: 2016-04-18。

作者简介: 韩新宇 (1980-), 女, 高级工程师, 主要从事软件可靠性与软件测试方向的研究。

断与趋势分析)。

2 关键技术研究

2.1 接口协议形式化技术

接口协议形式化用于建立接口协议的基本信息模型、元素信息模型和执行逻辑模型。基本信息模型由接口类型、协议名称、发送方地址、接收方地址和数据长度等信息构成。元素信息模型包含了元素名称、元素类型、字节数和有效值信息。其中元素类型信息较为复杂,其模型见表 1。执行逻辑模型用以描述报文收发的执行逻辑,特征属性有:1)收发标识;2)执行条件标识(0-错误、1-顺序、2-周期、3-延迟、4-反馈、5-同步);3)周期特性:周期间隔、停止条件、停止时间、停止周期数、停止消息;4)反馈特性:反馈条件、反馈变量名称、反馈变量数据特性、反馈消息;5)同步特性:同步步骤、同步时间;6)是否有返回值。

测试系统采用 XML 语言对接口协议进行形式化描述,具体的操作和数据信息分别记录到 XML 文件的子结点中,一棵 XML 数据树就能够表示一个完整的接口协议。

2.2 测试用例自动生成技术

在测试用例自动生成阶段,首先需要进行测试用例建模。本研究提出了一种基于软件任务的测试用例生成模型(以下称软件任务模型),其组成包括系统的使用者和参与者、功能模块、功能点、操作和输入输出数据。构造该模型包括以下 4 个步骤:

- 1) 构造系统使用视图:确定系统的使用者、参与者和被测系统,利用使用视图描述被测对象,即从使用的角度看待和描述被测系统;
- 2) 构造功能模型:根据软件需求规格说明,获取系统的功能并逐层分解,将其分解为最小功能点的集合;
- 3) 构造操作模型:根据软件需求规格说明,将每一个功

能点的实现分解为具体的操作;

4) 构造任务模型:遍历操作模型,将其中的每个操作序列单独提取出来作为一个任务流,所有任务流的集合称为任务模型。生成任务模型时主要可以遵循 4 种覆盖准则。

准则 1 (节点覆盖准则):遍历完成后,所有的操作节点至少被执行一次,这是最基本的覆盖准则。

准则 2 (迁移边覆盖准则):遍历完成后,所有的迁移边至少被覆盖一次。

准则 3 (逻辑路径覆盖准则):遍历完成后,各逻辑路径至少被执行一次。采用此覆盖准则时,可对模型中循环体的执行次数进行限定,例如只覆盖两种情况,即不进入循环体和进入循环体至少 1 次。

准则 4 (代表值覆盖准则):对于各逻辑控制节点上的每个控制变量,依据指定的测试策略分别选定一个有限的输入值集合,称作代表值集合,则遍历完成后各逻辑控制节点的每个代表值集合中所有取值至少被覆盖 1 次。

测试用例生成包括测试步骤生成和测试数据生成。测试步骤生成的总体策略是:根据软件任务模型,采用图的遍历技术,生成被测件及外围设备的交互流程,即测试用例所包含的各步骤。测试步骤生成包括 4 个阶段,如图 1 所示。异常测试步骤生成考虑:1)正常操作流程中缺少步骤;2)正常操作流程中某些步骤的输入时序不符合要求(包括提前或滞后);3)正常操作流程中某些操作的输入变量不正确。

测试数据生成主要考虑不同测试类型的需要,如功能测试采用等价类划分、错误推测、因果图等方法给出相应的测试数据,边界测试采用边界值分析法给出所需测试数据,接口测试根据接口的特性给出相应的测试数据,性能测试根据性能需求的测试场景生成所需的测试数据。

表 1 接口协议元素类型模型

特性标识	特性介绍	特性说明		
Num_UInt	无符号整数	属性(Property)	1:最低有效位;2:最高有效位;3:单位;4:最小值;5:最大值;6:精度。	
		内容(Content)	字符串,长度 50。	
Num_Int	有符号整数	属性(Property)	1:最低有效位;2:最高有效位;3:单位;4:最小值;5:最大值;6:精度。	
		内容(Content)	字符串,长度 50。	
BCD	BCD 码	属性(Property)	长度	
		内容(Content)	字符串,长度 50。	
IP		IP 地址	字符串,长度 50。	
Time		时间	字符串,长度 50。	
Num_Char		字符串	/	
Control_Word	控制字	字段名(FieldName)	字符串,长度 50。	
		长度范围(Byte)	字符串,长度 50。	
		类型(Type)	1:编码;2:备用。	
		数值(Value)	数值,长度 3。	
		编码或意义(Mean)	字符串,长度 50。	
		关联(Connection)	1:关联有效;-1:关联无效。	
Num_Array	数组	复合数组 (Compound)	元素名称(Name)	字符串,长度 50。
			元素类型(Type)	1:无符号整数;2:有符号整数;3:BCD 码;4:IP 地址;5:时间。
		单一数组 (Single)	字节数(BYTE_NUM)	数组长度,字符串,长度 50。
			元素类型(Type)	1:无符号整数;2:有符号整数;3:BCD 码;4:IP 地址;5:时间。
			字节数(BYTE_NUM)	数组长度,字符串,长度 50。
BackUp		备用	/	

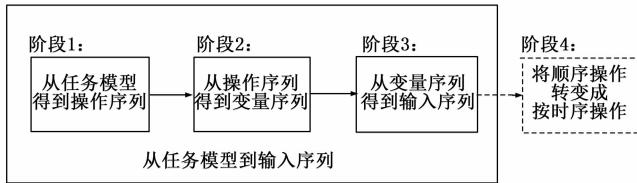


图1 测试步骤生成过程

2.3 测试用例自动驱动技术

考虑到舰船装备软件测试用例的输入以接口报文居多,且字段取值具有不确定性,测试系统采用脚本驱动技术,即:首先根据需求生成测试脚本,实现各种测试场景,然后通过执行测试脚本,检查被测软件在不同场景下运行结果是否正确。本研究主要利用LoadRunner测试脚本的稳定驱动能力,将测试输入数据存储在数据文件里,测试脚本则存放用例步骤信息,执行测试时从文件中读取测试输入数据,以便复用同一个脚本执行不同的测试场景。其中测试数据和测试脚本的自动生成由测试系统来实现,脚本的解析和执行由LoadRunner工具来实现。

2.3.1 测试脚本的自动生成

测试脚本的生成方法包括建立测试用例脚本框架集、定义脚本关键字和关键字实例化3个技术要点,而且要符合LoadRunner工具对测试脚本的语法要求。

测试用例脚本框架是指根据测试用例执行行为预先定义的具有一定流程控制功能的脚本单元。测试用例执行行为与测试脚本的执行逻辑密切相关,因此可以根据对测试用例行为的分类结果建立测试用例脚本框架集。对测试行为的分类主要从以下几个方面考虑:

- 1) 按照数据传输方向分:只有输入行为、只有输出行为、既有输入行为也有输出行为;
- 2) 按照时间特性分:如周期行为、事件行为(如反馈行为、中断行为等);
- 3) 按照传输条件分:无条件行为、有条件执行行为。

脚本关键字^[4]是根据测试用例模型要素,在测试框架集中定义的变量、函数、语句等信息。脚本关键字包括两类:一类是固定关键字,标识了脚本框架的类型、关键分支点,此类关键字是选择与调整脚本框架的依据;另一类是可变关键字,标识了用例要素描述信息,需要替换为具体的用例要素值。

关键字实例化就是用测试用例要素值替换测试脚本关键字的过程。在测试用例模型完备且关键字定义清晰的基础上,根据用例要素的描述在脚本框架中查找与之对应的关键字,并完成替换。

2.3.2 驱动数据的自动生成

驱动数据是指符合LoadRunner工具buf数据格式的测试数据,其生成过程包括数据驱动需求要素的获取、接口报文数据的生成和转换3个步骤:

- 1) 数据需求要素获取:对形式化测试用例的抽象描述进行解析,识别出对所需测试数据的名称、长度、类型、取值范围、最低有效位等要素的描述信息,统称为数据驱动需求要素信息。
- 2) 接口报文数据生成:利用测试用例的数据驱动要素生成所需接口报文数据。对于离散变量可从预先设定的常量池中

取值作为接口报文数据;对于连续型变量可采用随机测试数据生成方法,根据变量的取值范围随机生成接口报文数据;如果软件开发方提供被测软件的源代码,还可以利用启发式搜索算法生成接口报文数据^[5]。

3) 接口报文数据转换:在测试数据生成过程中需要解决两类数据转换问题。一类是实际物理量的转换,将接口报文中使用的实际物理量映射为LoadRunner可直接驱动的十六进制数据;另一类是状态数据的转换,将接口报文中使用的“状态位”信息按要求进行组合并转换成LoadRunner可直接驱动的十六进制数据。

2.4 硬件接口转换技术

由于LoadRunner工具只支持以太网数据传输接口,要想满足舰船装备软件其他常用接口(RS232、RS422、RS485、CAN总线)的信息交互需求,就需要实现接口转换,实现以太网与CAN总线接口、以太网与串口(RS232/RS422/RS485)的双向透明数据传输。本研究中采用ARM CORTEX-M3处理芯片进行编程以实现接口转换,总体方案如图2所示,其软件部分采用模块化的设计思路,主要分为接口初始化、串口通信、CAN通信、以太网通信、缓冲区管理、设备管理等模块,具备网络中断后自动断开连接、以太网/CAN总线/串口传输参数设定、兼容CAN2.0A和CAN2.0B标准、串口数据分帧设置、串口硬件隔离、支持TCP/IP、UDP工作模式等能力。

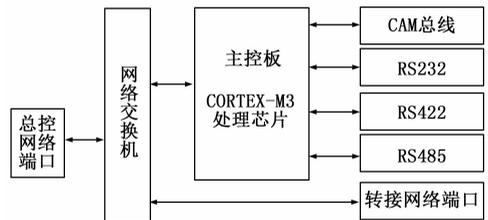


图2 接口转换方案

3 测试系统的设计与实现

3.1 测试系统的体系结构

本研究以上述技术为基础开发舰船装备软件测试系统。该测试系统包括操作显示层、业务逻辑层、数据存储层、通信服务层和物理层共五层结构。操作显示层提供人机交互界面接口;业务逻辑层实现系统的各项核心功能;数据存储层负责存储系统运行过程中产生的各类数据;通信服务层和物理层负责测试过程中与被测件的数据交互和管理,其软件体系结构如图3所示。

3.2 测试系统的设计

舰船装备软件测试系统硬件部分包括:1)已安装Windows操作系统和.NET Framework 4.0的测试计算机1台;2)接口转换设备2台,每台设备对外提供2路CAN总线接口、2路RS232串口、2路RS485串口、2路RS422串口和1路以太网接口。接口转换设备通过交换机与测试计算机连接,如图4所示。

舰船装备软件测试系统软件部分主要包括测试管理模块、测试用例生成模块、测试用例驱动模块和数据收集与分析模块,其组成如图5所示,采用C++/C#语言基于LoadRunner

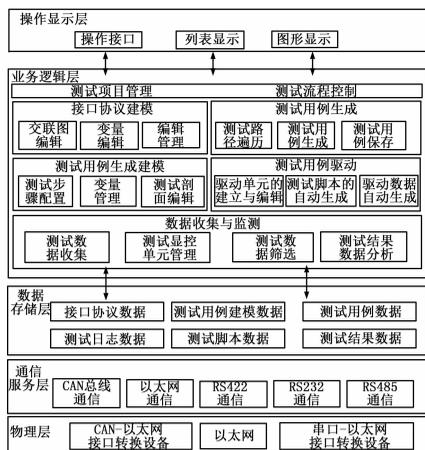


图 3 舰船装备软件测试系统软件体系结构

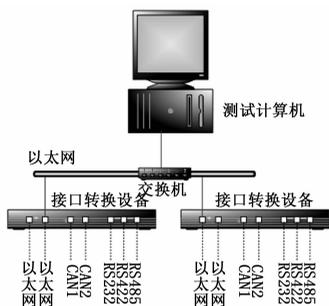


图 4 测试系统硬件组成

自动化应用程序接口 (LoadRunner Automation API) 开发实现。

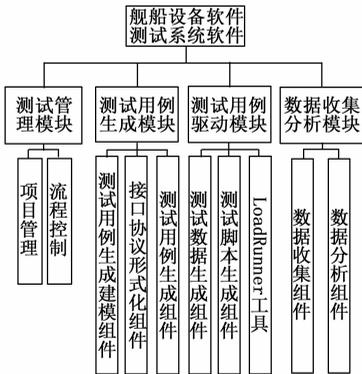


图 5 舰船装备软件测试系统软件组成

为了提高测试执行效率,通过编写测试后台程序调用 LoadRunner 自动化应用程序接口,实现测试场景的自动构建、负载生成器的自动配置、测试脚本的自动执行以及运行日志的自动记录。通过编写挂接外部模块对 LoadRunner 测试脚本功能进行扩展,实时监控和采集测试响应数据、测试用例执行状态以及测试结果等信息。

4 结束语

软件测试自动化技术一直是软件工程领域研究的热点。本文依据舰船装备软件报文信息交互测试的需要,研究接口协议形式化、测试用例自动生成、测试用例自动驱动、接口转换等技术,然后以 LoadRunner 工具为依托,给出了测试系统的实现方法,相对于手工测试本测试系统主要优点在于:

- 1) 便于回归测试。利用已有的测试脚本能够很方便的执行回归测试,回归测试是自动化测试最主要的任务。
- 2) 缩短测试时间。由于自动测试采用脚本技术,可能只需要做少量的甚至不做修改实现在不同的测试过程中使用相同的用例,可多次重复执行,提高测试的效率。
- 3) 自动输入大量数据,可以执行很多手工测试困难或不可能做的测试,比如性能测试、大数据量输入测试等。
- 4) 节约人力资源,可实现无人化测试执行,测试人员可投入更多的精力到测试用例设计工作。
- 5) 保证测试的一致性和可重复性,利用自动测试重复多次相同的测试,在手工测试中是很难得到保证的。

实践证明,本测试系统为舰船报文交互软件提供了一种可行的自动化测试手段。然而,受人机交互存在不确定性的约束,测试系统未考虑人机交互界面测试的情况,这也是后续重点研究方向。

参考文献:

[1] 陈 策,郭久武,胡 勇. 装备通用嵌入式软件综合测试平台研究 [J]. 军用软件测评实验室, 2014, 44.

[2] 刘娟娟. 基于 Robot 框架的软件自动化测试技术的研究与应用 [D]. 杭州: 浙江大学, 2013.

[3] 张 峰,斯中毅,王 斌. 信息化装备软件测试 [J]. 舰船电子工程, 2006 (5): 23-26.

[4] 王云鹏. 一种关键字驱动的 UI 自动化测试框架的设计与实现 [D]. 北京: 北京大学, 2010.

[5] 何 伟,韩新宇,唐龙利,等. 舰船装备软件测试数据自动生成方法及其应用研究 [J]. 计算机测量与控制, 2015 (8): 2633-2636.

(上接第 157 页)

参考文献:

[1] 殷希梅,王占操,张运兵,等. 地面区域封锁弹药综述 [J]. 兵工自动化, 2014, 33 (7): 79-82.

[2] 薛雪东,徐 兵,洪 光,等. 基于 Creator/Vega Prime 的某导弹发射设备模拟训练系统 [J]. 弹箭与制导学报, 2008, 28 (6): 307-310.

[3] 李京伟,张利萍. 基于虚拟现实技术的飞行视景仿真 [J]. 计算机工程与设计, 2005, 26 (7): 1936-1937.

[4] 孙崇波,唐永刚,宁建国. 基于 Creator/Vega Prime 的虚拟航母漫游设计 [J]. 计算机仿真, 2007, 24 (1): 222-224.

[5] 孔中武,石林锁,王 涛. 基于 Creator 和 Vega Prime 的导弹飞行视景仿真系统设计与实现 [J]. 四川兵工学报, 2014 (10): 111-113.

[6] 卢卫平. 基于 Vega Prime/Multigen Creator 的飞行复现视景仿真系统设计 [D]. 南京: 南京航空航天大学, 2011.

[7] 王 乘,周均清,李利军. Creator 可视化仿真建模技术 [D]. 武汉: 华中科技大学出版社, 2005.

[8] 李亚臣,罗永峰,黎远忠,等. 基于 Vega Prime 的火箭视景仿真 [J]. 弹箭与制导学报, 2007, 27 (4): 222-225.