航 空 学 报
Acta Aeronautica et Astronautica Sinica
Apr. 25 2012 Vol. 33 No. 4 722-733
ISSN 1000-6893 CN 11-1929/V
http://hkxb.buaa.edu.cn hkxb@buaa.edu.cn
文章编号:1000-6893(2012)04-0722-12

综合模块化航电软件仿真测试环境研究

周庆[1], 刘斌[1,*], 余正伟[1], 冯时雨[2]

1. 北京航空航天大学 可靠性与系统工程学院,北京 100191
2. 北京航空工程技术研究中心,北京 100076

摘 要:伴随着综合模块化航空电子(IMA)软件在新一代飞机上的应用,其高复杂性、高度综合的特点以及分层的健康监控和故障管理模式给软件测试提出了挑战。传统的仿真测试环境在应对 IMA 软件测试中难以满足 RTCA DO-178B 中规定的对验证过程结果的验证的要求。本文在分析 IMA 软件特点的基础上,根据 DO-178B 的要求,综合国外的发展情况和国内的研究进展情况,研究综合模块化航电软件仿真测试环境需求,提出了基于软件故障注入的综合模块化航电软件灰盒仿真测试环境方案,并给出优势分析。该仿真测试环境方案以 IMA 软件为测试对象,应用软件故障注入技术和代码插装技术满足测试规范文件的要求。其具有通用灵活、适配性强、强实时性等特点,为中国新一代航电软件的系统验证和测试奠定了基础。

关键词:综合模块化航空电子;软件;灰盒测试;仿真测试环境;软件故障注入;DO-178B

中图分类号:V247;TP311.5 **文献标识码**:A

随着信息技术的飞速发展和用户要求(军用和民用)的不断提高,综合模块化航空电子(Integrated Modular Avionics,IMA)已成功运用在国外新一代航空器的机载设备中,例如波音 757、波音 777、空客 A380、F-22 和 F-35 等。与此同时,航电设备软件化的趋势日趋明显,F-22 上由软件实现的航电功能高达 80%[1],并且在一些安全关键的系统中,例如飞控计算机(Flight Control Computer, FCC),其应用软件代码占整个源代码的 1/3,而支持系统余度管理和故障检测的代码部分超过整个代码的 55%(例如波音 757 的 FCC 软件)[2]。这一部分软件代码的开发和测试是极其复杂的,因此给系统的开发和综合增加了不少成本,尤其是为软件测试提出了巨大的挑战。

国外开展 IMA 系统和软件的开发和验证工作起步早,并且制定了技术规范,如 RTCA DO-297《综合模块化航空电子设计指南与合格审定要求》[3]、DO-178B《机载系统和设备合格审定的软件考虑》[4]、SAE-ARP4754《对高度综合或复杂系统的合格审定考虑》等。在软件测试领域也在探索和开展形式化的验证方法[5-6]和行之有效的仿真测试环境工具,来保证软件质量。例如洛克马丁公司为 F-22 研制了 LM-STAR 仿真测试环境[7]用于航电设备的验证、测试和地面维护。尽管如此,在 2007 年还是发生了 F-22 经过国际日期变更线时航电系统崩溃[8]的事故。事后分析得知,软件设计和验证过程中均没有考虑到国际日

收稿日期:2011-06-22;**退修日期**:2011-08-30;**录用日期**:2011-11-29;**网络出版时间**:2011-12-09 17:25
网络出版地址:www.cnki.net/kcms/detail/11.1929.V.20111209.1725.002.html **DOI**:CNKI:11-1929/V.20111209.1725.002
基金项目:国防预研项目(513190802)
* **通讯作者**.Tel.:010-82339950 E-mail:liubin@buaa.edu.cn

引用格式:*Zhou Q, Liu B, Yu Z W, et al. A framework of simulation testing environment for integrated modular avionics software. Acta Aeronautica et Astronautica Sinica, 2012,33(4):722-733.* *周庆,刘斌,余正伟,等. 综合模块化航电软件仿真测试环境研究. 航空学报,2012,33(4):722-733.*

期变更线经纬度的特殊情况,从而导致系统失效。此事件从一个侧面体现出 IMA 软件验证和测试的重要性,以及仿真测试环境对测试充分性的支撑能力。

随着国内航空事业的不断进步和深化发展,各个机构也在积极开展对 IMA 系统以及软件的研究和工程实践工作。但是大多数机构还停留在消化吸收国外相关 IMA 技术标准和规范的前提下,开展 IMA 的研制。文献[9]对 IMA 软件体系结构研究进行了综述。国内已制定了《航空电子应用软件接口》标准[10-11],并且在中国下一代飞机研发的关键技术中综合应用 IMA 规范标准[12]、虚拟化技术[13]和可信性技术[14],同时提出了发展思路。

在嵌入式测试领域,以白盒测试和黑盒方法为主,形成了相对成熟的测试技术和工具。白盒测试工具主要有 C++Test、TestBed、VectorCAST 和 Cantata++等。比较成熟的嵌入式软件仿真测试环境均是基于外总线的黑盒仿真测试环境[15-18]。其中北航研制的通用嵌入式软件仿真测试环境平台[15],已广泛应用于联合式航电软件仿真测试中。

随着 IMA 研究的进一步深入,传统的软件仿真测试环境难以满足 IMA 的测试要求,主要体现在:①现有仿真测试环境难以满足 DO-178B 中对验证过程结果的验证要求;②传统的黑/白盒测试工具(环境)难以适应 IMA 中以分区操作系统、软硬件隔离为特征的模块化结构;③现有仿真测试环境难以满足以面向故障处理为特征的 IMA 测试性设计、健康监控和余度管理等功能验证和测试。

因此,本文分析 IMA 软件仿真测试环境需求,在通用嵌入式软件仿真测试环境的基础上,提出基于软件故障注入的灰盒仿真测试环境方案,并对其优势进行分析,为中国新一代航电软件的验证和测试奠定基础。

1 综合模块化航电软件概述

综合模块化航电软件是综合模块化航空电子软件的简称(注:国内也一般称为综合航电软件[9]),是运行在综合模块化航空电子系统中的软件资源。

1.1 综合模块化航空电子系统

IMA 系统本质上是一个分布式实时计算机网络,其主要目标是将分布式体系结构的灵活性扩展到对不同关键级别的功能程序的支持上[19]。其具有系统综合化、结构层次化、功能软件化、网络统一化等特点。

IMA 与传统的联合式航电系统最大的区别在于:模块化的分系统综合在统一的软/硬件资源共享平台,突破了联合式航电系统中各分系统通过外总线(MIL-STD-1553B、ARINC 429 等)进行数据交换的独立处理模式,使得航电系统进入了标准化、模块化和通用化的新时代。它大大节约了系统的物理空间和重量,同时也使得以往的三级维护变成二级维护,由外场可更换单元(Line Replaceable Unit, LRU)变成了外场可更换模块(Line Replaceable Modular, LRM)。

同时 IMA 在体系结构上除了模块化和通用化有着明显的改进之外,还在高速航电总线技术、背板总线以及测试与维护总线上有了新的要求和突破,并且在满足飞机飞行功能需求的同时,对系统的健康监控、余度管理和测试性能力提出了更高的要求。

1.2 综合模块化航电软件结构

IMA 作为软件密集型系统,在统一的平台上实现信息的综合处理,联合式航电系统结构中面向特定硬件处理环境的软件设计理念在 IMA 中不能得以沿用。因此,IMA 采用开放式软件体系结构,并积极吸收民用航电标准和商用货架产品(Commercial Off-The-Shelf,COTS)技术,推进产品的标准化、模块化。在具体实现上,IMA 采用软件分层策略,层与层之间通过标准接口进行访问,旨在实现应用软件与硬件的相互隔离,有利于软硬件产品的升级换代。同时应用程序面向功能进行设计,通过分区加以隔离。文献[9]阐述了航空、航天领域产生了 4 种典型的 IMA 软件体系结构,分别是 ARINC653、ASAAC、GOA 以及 F-22 通用综合处理机(Common Integrated Processor, CIP)上的软件体系结构。

1.2.1 软件体系结构

4种IMA软件体系结构最主要的特点是采取了分区的并行处理策略,改变了以往高可靠、高安全嵌入式软件单任务串行处理的传统结构,在保证实时性的同时兼顾多任务间的"隔离"。

以ARINC653为例,体系结构分为3层(如图1):应用软件层、操作系统核心层和硬件模块支持层[20]。应用软件层是指包括航空电子系统中所有应用软件的功能模块。操作系统核心软件层是指提供了实时操作系统的一般服务,主要包括调度、通信、同步与异步操作、存储管理、异常/中断处理等服务。硬件模块支持层由满足操作系统模块接口规范的专用硬件模块支持软件组成。硬件模块支持层主要提供上下文切换、高速缓冲存储器处理、内存管理单元、总线控制、中断控制、异常处理、实时时钟、BIT、控制台、看门狗、加载服务等功能。

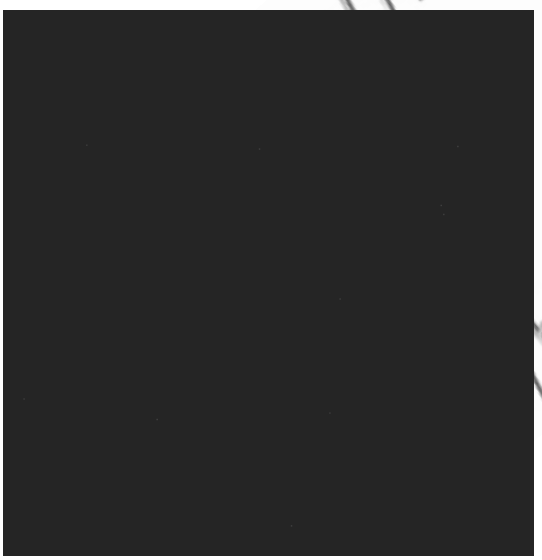

图1 IMA软件体系结构示意图

Fig. 1 IMA software architecture

1.2.2 软件主要特点

IMA软件与联合式航电系统软件相比,其脱离了对硬件的强依赖性,具有如下特点:

① 层次化结构

与联合式航电系统软件必须紧密结合特定硬件的特性相比,层次化结构的IMA软件能够更加有效地解决IMA复杂性的问题,实现硬件资源的综合共享。

② 统一化平台

由于规定了应用软件与操作系统之间的接口关系,IMA应用软件运行在满足IMA要求的统一化的操作系统平台上,这也进一步体现了层次化的结构。同时也进一步弱化甚至隔离了IMA应用软件对硬件系统的依赖关系,使得软件功能单纯化。

③ 配置化管理

IMA中各应用软件通过运行时蓝图[21](也有称之为蓝印[1])在操作系统的统一管理和调度下运行。并且各应用软件以分区进程的形式存在,具有相互的独立性,进程间的通信通过统一的运行时蓝图形式进行配置。这样既增强了系统的灵活性和适应性,又改进了系统的容错性[1]。

④ 健康监控和故障管理

IMA软件中具有通用的管理模块完成对IMA系统中各层次的故障检测、定位和隔离,并且实时地进行健康监控,同时根据运行时蓝图的要求实现余度管理和系统重构。

⑤ 动态重构

在发生故障时,IMA系统中会根据实时健康监控的状态,结合运行时蓝图的信息,进一步整合软硬件资源,实现系统的动态重构。

综合以上几点,相比联合式航电系统,IMA软件在面向故障处理方面的可靠性设计具有明显优势,同时也给其验证和测试提出了难题。

2 IMA软件仿真测试环境需求分析

在IMA软件测试方面,国外NASA、FAA以及RTCA等机构均在开展相关的研究,并且制定了较为成熟的规范,如DO-297和D0-178B。随着国内民航飞机的发展,以DO-178B为主的软件适航认证的实施,国内的相关机构正开展基于DO-178B的软件测试的研究。同时IMA的深入研究必将促使软件测试的方法和技术的改进。本小节将以两个规范文件为基础阐述IMA软件对仿真测试环境要求。

2.1 IMA软件测试的新需求

伴随着IMA与传统联合式航电系统在设备形式、软件形态上以及开发过程和方法上的差异,

IMA 软件测试需求也存在一定差异:

① 被测对象的不同

联合式航电系统软件必须紧密结合特定硬件,在开展软件系统测试时,必须构建系统仿真测试环境完成外部数据激励。而 IMA 软件具有从应用程序到模块平台再到系统的层次化结构。DO-297 规定了在 IMA 合格审定中的 6 大任务,包括:模块平台验收、应用程序验收、IMA 系统验收、航空器系统综合验收、模块或应用程序的更改验收和重用验收。因此,被测对象的不同对 IMA 软件测试提出新的过程要求和仿真测试环境要求。

② 高度综合的新特点

IMA 软件具有高度综合的特点,其健康监控和故障管理的功能成为软件测试的难点,尤其是动态重构的功能。如何达到并且满足"故障覆盖率"是 IMA 软件测试相对于传统航点软件测试而言新的需求。

③ 对验证过程的验证的新要求

IMA 软件的复杂性给测试带来风险,作为对安全性要求极为苛刻的 DO-178B 标准,规定除了对开发过程要进行验证之外,还要有指标来证明这些验证工作结果真实可靠,这就是验证过程结果的验证过程。即如何通过软件结构覆盖率来保证软件需求的覆盖。

2.2 仿真测试环境的新需求

IMA 系统在软件测试方面提出了新的要求,因此仿真测试环境也必须有相应的改进和完善。为保证 IMA 软件测试的充分性,把 IMA 模块和应用软件作为测试对象,就必须考虑到交联环境的复杂性、数据输入的实时性和有效性(如单次测试输入的代码覆盖率信息),以及如何覆盖系统中的健康监控需求的特殊性等因素。

对于模块平台和应用程序分别要开展验收测试。分别以模块平台和应用程序为测试对象,这是 DO-297 为 IMA 软件测试提出的最基本的要求。因此测试人员必须围绕着 IMA 平台和应用软件的特征来考虑仿真测试环境的需求。

现有的面向联合式航电软件仿真测试环境难以满足 IMA 软件(模块平台和应用程序)测试的需求。如何产生测试数据,如何激励被测件(模块平台或应用程序)的运行,如何收集测试结果数据等等问题均给仿真测试环境提出了难题。

DO-178B 中还对仿真测试环境提出了明确的要求:①具有高保真度,被测软件最好能在目标机环境中运行;②精确的控制和监视测试输入和代码执行情况;③能够辅助提供需求覆盖和结构覆盖的信息。

结合 IMA 软件的特点,满足 DO-178B 要求的软件测试过程除了在对测试过程和技术有着进一步的要求之外,难点在于如何在真实目标机环境下满足对验证过程结果的验证。

文献[15]中研究的分布式仿真测试环境(DSTE)具有通用灵活的特点,已广泛应用于国内的多个型号装备的软件测试工程实践中。DSTE 具有如下功能特点:

a) 可视化交联环境建模;

b) 仿真模型代码自动生成;

c) 强实时的数据处理;

d) 多协议的外总线级接口适配(MIL-STD-1553B、ARINC-429、AD/DA、DIDO、RS422 等总线或者接口);

e) 灵活可控的类 C 语言实时测试脚本;

f) 丰富的数据显示控件。

然而,IMA 软件的自身特点与 DO-297、DO-178B 对机载软件测试的要求相结合给 DSTE 提出了新的挑战。因此,对 DSTE 提出新的需求,主要包括:

① 进程/分区的交联环境建模需求

被测对象从以往的嵌入式设备到现有的 IMA 模块中的进程/分区。建模对象的粒度发生明显的变化。

② 模型重用需求

IMA 软件相比联合式的航电软件更加复杂,交联环境建模中的仿真模型在软件测试中必须具有复用的功能,才能有效应对 DO-297 中对"更改"验证和"重用"验证的要求。

③ 运行时蓝图仿真需求

被测对象中的 IMA 中的进程/分区是在运行时蓝图的统一管理和调度下运行的,因此仿真测试环境必须提供运行时蓝图的仿真接口。

④ 软件故障注入需求

IMA 软件具有故障检测、定位和隔离,以及健康监控、余度管理和动态重构的功能,为激励这

些功能模块运行,因此必须开展故障注入。

⑤ 覆盖率的需求

为满足 DO-178B 对软件仿真测试环境提出的结构覆盖要求,因此仿真测试环境中必须能够实时获取被测代码的覆盖率信息,用以保证测试的充分性和有效性。

⑥ 测试和维护总线(MTM-Bus)的需求

IMA 在系统测试性设计中引入了 MTM-Bus,并且其直接服务于系统的健康监控模块,因此仿真测试环境必须有效地适配该总线并且结合故障注入的需求完成对 IMA 系统测试性的验证和测试。

⑦ 机载网络升级的需求

与联合式航电系统相比,IMA 具有实时性更强、可靠性和容错能力更高的机载网络[1]。因此,仿真测试环境支持传统航电总线 MIL-STD-1553B 基础上,需要进一步扩展并且支持以光纤通道(FC)和航空电子全双工交换式以太网(AFDX)为代表的新一代机载网络总线接口需求,以达到接口适配性的要求。

3 IMA 软件仿真测试环境框架

通过对 IMA 软件仿真测试环境需求的分析,结合 IMA 软件的特点以及嵌入式软件测试的基本方法和相关测试规范文件的要求,测试人员需要构建一个自动的、实时的、侵入式的闭环仿真测试环境,本文称之为基于软件故障注入的 IMA 软件灰盒仿真测试环境(以下简称仿真测试环境)。

3.1 基于软件故障注入的灰盒仿真测试环境

为有效地验证和测试 IMA 软件的功能,尤其验证代码量超过 50%的与故障处理相关的功能模块,仿真测试环境必须具有软件故障注入和探测代码覆盖率的典型特征,这是有别于传统的黑/白盒仿真测试环境的显著特点之一。

灰盒测试是指结合白盒测试和黑盒测试的测试方法[22]。白盒测试[23]又称为结构测试,在测试过程中测试者可以看到被测的源程序,通过分析程序的内部结构,根据其内部结构设计测试用例。黑盒测试[23]又称为功能测试,在测试过程中被测程序被视为黑盒,测试者在完全不考虑程序内部结构和内部特征(或对于上述信息无从获知)的情况下,根据需求规格说明书设计测试用例和推断测试结果的正确性。

显然,这两类测试方法是从完全不同的角度出发对软件进行测试的。两类方法各有侧重,在测试的实践中都是有效和实用的,不能指望一类方法能够完全代替另一类方法。但应对 IMA 软件的测试需求,二者又各具缺点,这些缺点不是通过在各自的测试方法内部进行完善就能够解决的。只有将二者有效地结合,即进行所谓的"灰盒测试",才能弥补任何一种方法的不足,使测试方法的机理更完善。

基于上述考虑,在分布式仿真测试环境体系结构[5]的基础上结合软件故障注入和代码覆盖率探测的需求,提出了基于软件故障注入的 IMA 软件的灰盒仿真测试环境方案。其主要功能如下:

a) 一体化测试开发环境;

b) 图形化交联环境建模;

c) 分布式半实物实时仿真;

d) 动态软件故障注入,包括故障模型配置、故障注入触发以及信息收集;

e) 代码插装功能;

f) 实时测试脚本,主要包括周期型脚本和任务定时型脚本任务;

g) 实时任务调度;

h) 实时数据收集;

i) 内存管理;

j) IMA 目标平台管理,包括测试监控进程和背板总线数据通信管理;

k) 统一的通讯协议栈管理;

l) 测试管理功能。

3.2 仿真测试环境框架结构

通过对仿真测试环境的功能需求的分析,图 2 给出了仿真测试环境框架组件结构图。仿真测试环境框架主要包括 3 大部分:测试开发环境、实时处理内核和 IMA 目标平台。测试开发环境与实时处理内核通过以太网连接,而实时处理内核为多节点的分布式系统,其通过真实物理接口(背板总线、MTM-Bus、MIL-STD-1553B、ARINC-429、FC、AFDX 等)完成与 IMA 目标平台连接。

测试人员利用测试开发环境提供的方便灵活

的开发、配置和管理图形界面,并调用实时处理内核各项测试服务,构建被测对象的交联环境,并利用测试脚本实现测试数据的输入以及故障注入控制,完成与被测 IMA 模块或者应用软件的数据输入输出通信,以及相关覆盖率信息获取,最终得到被测件的运行情况并分析出测试结果。

图 2 IMA 软件仿真测试环境框架示意图

Fig. 2 IMA software testing environment framework

3.2.1 测试开发环境结构

测试开发环境主要是为软件测试人员提供方便易用的图形用户界面(GUI)服务,其中主要包括一体化测试管理组件、仿真模型开发组件、测试脚本开发组件、测试结果分析组件和数据监控图形界面组件,以及目标代码产生组件、软件故障注入组件、代码插装组件和蓝图仿真开发组件。

一体化测试管理组件,作为整个仿真测试环境的图形界面框架,实现对测试开发环境中其他各个组件的统一管理和配置,并且提供灵活的、可扩展的组件管理功能,能灵活应对用户在使用界面上的个性化定制。

仿真模型开发组件,主要是提供被测件交联环境仿真模型接口数据的配置开发,以及模型重用和引用外部模型的管理和控制工作[17]。建模的对象不但包括外部交联设备,例如联合式航电系统中的飞控设备、惯导设备等,还可以对 IMA 系统中模块和分区进行交联环境建模。其具有图形化配置、代码自动生成、模型动态加载的特点。

测试脚本开发组件,提供测试脚本任务的开发、配置和编辑功能,以及相应的语法检查、辅助编写的功能。实时测试脚本[24]以一种类 C 语言语法的形式为测试人员提供测试输入控制的手段,并且其与软件故障注入组件相结合,提供故障注入控制手段。

测试结果分析组件,主要提供对测试输入输出数据的图表显示,以及代码的覆盖率显示等功能。

数据监控图形界面组件,主要是为测试人员构建灵活多样的输入输出监控界面,提高易用性和信息表述的准确性。

目标代码产生组件,主要提供对目标代码的链接、编译以及加载的功能。其是为软件故障注入和代码插装服务的。

软件故障注入组件,主要结合软件故障模型,

利用程序变异技术对被测应用软件开展动态故障注入,以达到激励 IMA 系统中健康监控的各项功能运行的目标,并由此来验证健康监控功能的有效性。该组件作为整个仿真测试环境中重要的组成部分,在测试开发环境中提供程序变异的控制、配置,并且生成在 IMA 目标平台系统分区中运行的"故障监控程序"。

代码插装组件,主要是对被测应用程序的源代码进行插装,并且完成与插装库文件的连接工作,经过目标代码产生组件完成编译后加载到 IMA 目标平台系统中。

蓝图仿真开发组件,主要提供 IMA 蓝图的配置工作,以保证被测程序在 IMA 目标平台中的正确有效运行。

3.2.2 实时处理内核结构

实时处理内核主要是为仿真测试环境提供实时数据通信保证。其通过嵌入式实时操作系统的统一调度和管理,为测试开发环境提供实时处理服务,同时对交联环境中的各硬件接口资源进行统一管理。其主要分为应用层、系统层、实时操作系统层和硬件驱动层 4 层结构。

应用层主要是为测试开发环境提供实时服务,包括测试控制、仿真模型、测试脚本执行、测试显示监控、数据传输路由和数据收集服务等。其中每项服务均对应于测试开发环境中某个或者多个组件。尤其是数据传输路由服务,它为作为黑盒仿真测试环境的外部交联数据提供路由服务,确保测试数据通过匹配的真实物理总线与被测 IMA 模块完成数据通讯。

系统层主要是为应用层提供封装过的系统服务,包括任务调度和管理模块、内存管理模块、通讯协议栈管理模块、实时事务处理模块、错误处理模块和分布式节点通信处理模块。任务调度和管理模块主要是对周期任务和定时任务进行统一注册、管理和调度。内存管理模块则是对仿真模型变量存取、数据收集等内存存储相关的功能进行统一管理,包括内存的统一申请和释放、缓存区的开辟等。通讯协议栈管理模块主要是针对仿真测试环境中各硬件接口协议进行统一的管理。实时事务处理模块则是系统层的辅助模块,包括对测试流程的控制以及相关信号量资源的管理模块。错误处理模块则是向测试开发环境上报错误代码,并且在实时处理内核中按照出错处理原则实施报错、停止运行、重启等操作。分布式节点通信处理模块,则是在多节点运行下,完成各节点间的数据共享和交互,形成透明的分布式处理机制。

内核层则是实时操作系统,它为系统层提供基本服务。

驱动层则是各硬件接口的驱动程序集合,同时也是包括与实时操作系统匹配的板级支持包(BSP)。

3.2.3 IMA 目标平台结构

IMA 目标平台在此仿真测试环境中具有一定的特殊性。IMA 软件中的应用程序与硬件的耦合性较低,IMA 软件装载在 LRM 产品中,而对 LRM 成品无法实施故障注入和代码插装操作。为确保 IMA 软件测试的充分性,在测试其健康监控、余度管理、测试性设计等与故障相关的操作时,必须通过软件故障注入与外总线数据相结合的方式开展测试。因此,需要对被测软件实施侵入式操作。同时,为收集代码覆盖率信息,也必须对源代码实施侵入式的插装工作,再经过编译后重新生成新的目标文件。

因此需要一个 IMA 目标平台,该平台既能保持被测 LRM 所有的硬件特征,尤其保证背板总线数据的真实性,又要具有程序加载接口,能加载经侵入式操作(代码插装和程序变异)后的目标文件。并且该目标平台的 IMA 系统分区中还应驻存有故障监控程序、测试控制程序和蓝图仿真控制程序,分别完成对故障注入的监控、覆盖率信息的处理和蓝图配置的仿真控制功能。

3.3 关键技术

仿真测试环境继承了 DSTE 在交联环境仿真建模、实时调度、统一总线接口协议管理等方面的优势并重点集成了 IMA 平台下的程序插装和基于程序变异的软件故障注入技术。

3.3.1 IMA 平台下代码插装技术

在 IMA 体系结构下,结合其操作系统实现插装技术是本仿真测试环境的关键技术之一。插装

技术也正是“灰盒”测试技术中表现“白盒”测试特征的关键点,并且在 DO-178B 中关于“验证的验证”的测试思路,也是通过插装技术来体现的。在此技术下,需要考虑以下几个方面的内容:插装哪些信息、什么位置插装、如何插装、插装数据的编码和解码,还有插装对原代码的影响分析和插装与软件故障注入的关联关系等。

仿真测试环境中通过对被测源代码的静态分析(包括预处理、语法分析和语义分析等),结合插装策略,生成插装库探针函数,实施对源文件代码的插装工作,基本原理示意图如图 3 所示。

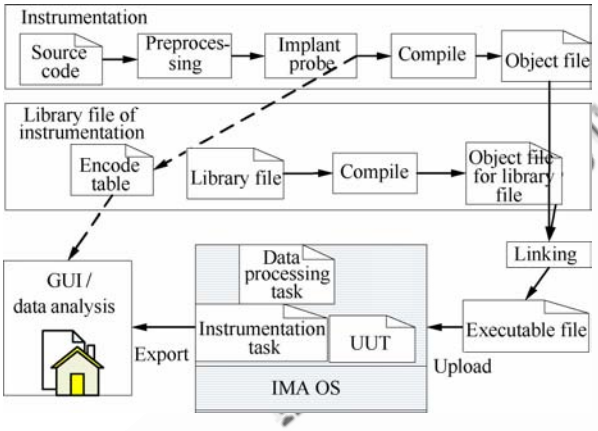

图 3 IMA 平台下代码插装技术示意图

Fig. 3 IMA software code instrumentation diagram

插装技术的具体实现,仿真测试环境可以借鉴或者集成市场白盒测试工具的功能模块,如 C++TEST、TestBed、CodeTest 等,形成对代码覆盖率信息的获取和分析。

3.3.2 基于程序变异的软件故障注入技术

程序变异是一种面向软件缺陷的测试方法,主要有代码变异和数据状态变异两类。其中代码变异是直接修改源代码,从而改变程序执行状态;数据变异是指程序运行时修改程序的内部状态,如内存、全局变量及时间等。仿真测试环境中的程序变异必须与代码插装技术相结合,达到既能注入故障又能监控故障的目的,实现灰盒测试目标。在 IMA 平台架构下,受到硬件体系结构、机载软件复杂性以及代码插装技术的影响,为满足机载软件在“故障”覆盖率测试方面的需求,程序变异需要解决如下几个问题:程序变异与代码插装的异同;变异的位置;变异的方式方法,即变异算子与故障模型的结合;故障恢复问题的考虑;软件故障注入脚本技术。

在基于交联环境建模的基础上,研究仿真测试环境框架中基于程序变异的软件故障注入技术。首先,针对源代码进行静态分析(与插装技术过程类似),完成对源代码的模式识别,再结合故障模型,手动/自动在故障敏感点处完成程序变异工作,包括代码编译和数据状态变异,图 4 为程序变异技术的原理图。

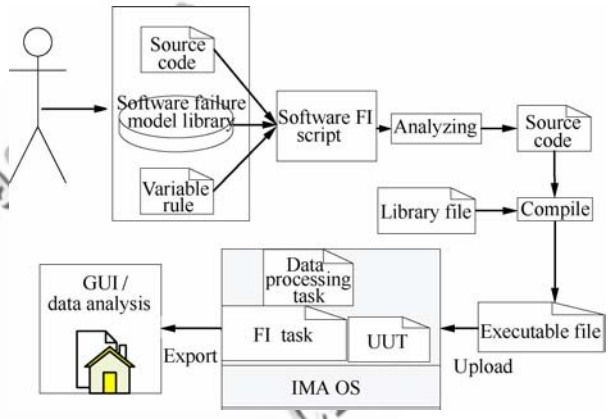

图 4 基于程序变异的软件故障注入技术

Fig. 4 Software fault injection technology based on program mutation

3.4 仿真测试环境实现

继承 DSTE 在工程应用中的优点,本文设计实现了 IMA 软件仿真测试环境(Simulation Testing Environment for IMA Software, STE-IMAS)原型系统。其中对于新增的外部总线接口需求(如 AFDX、FC、MTM-Bus 等),可在 DSTE 提供的接口扩展体系框架满足,并且统一纳入到可视化交联环境仿真建模组件中实现。

图 5 为仿真测试环境硬件部署图,按照其测试开发环境、实时处理内核和 IMA 目标平台 3 部分的划分,该环境部署有测试控制和目标平台控制两台主机,以及 3 台分布式实时处理和目标码运行的 IMA 目标平台。

测试主控机在一体化测试管理组件框架下完成仿真模型开发、测试脚本开发以及数据监控控件的配置和测试管理等功能,而目标平台主控机主要是完成软件故障注入配置、代码插装以及蓝图仿真开发和目标码加载的功能,并且两者通过

以太网完成数据的共享。

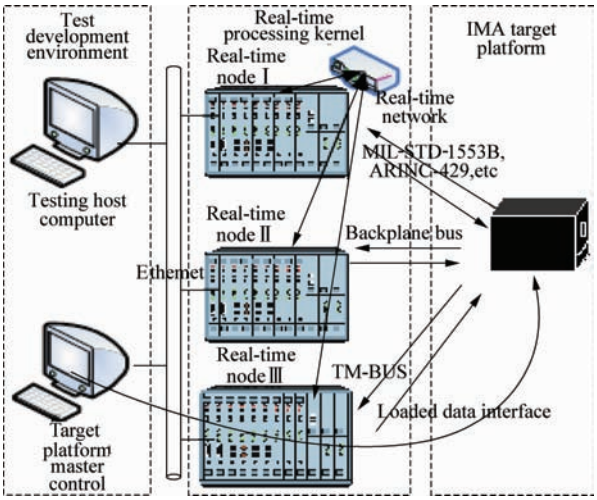

图 5 IMA 软件仿真测试环境硬件部署示意图

Fig. 5 IMA software testing environment hardware deployment diagram

3 个实时节点具有相同的软件体系结构，只是配置不同的硬件接口，其通过实时内存完成分布式数据通信，并达到透明实时处理的目标。

IMA 目标平台则需要根据被测 IMA 软件的情况动态配置，在本原型系统中提供了一个通用化基于风河 VxWorks653 的 IMA 硬件仿真平台。通过目标平台主控机的故障注入和代码插装组件的配置，把目标程序以及相关辅助程序加载到 IMA 目标平台中运行。

4 实例应用

本文研究的测试环境适用于 IMA 模块平台测试和应用程序测试。STE-IMAS 是在 DSTE 的基础上进行扩展而成的。

结合 DO-297 中对 IMA 模块平台测试的任务要求，在 STE-IMAS 中对某 IMA 模块平台进行测试，通过对外部数据交联环境数据仿真，以及脚本技术实现数据间的逻辑交互，从而完成对某 IMA 原型模块的测试工作；同时通过风河 VxWorks653 的仿真工具完成对被测软件代码的插装和加载，同步实现代码覆盖率指标的获取。图 6 为交联环境建模、脚本编辑控制、测试数据显示以及监控界面的综合图例。

DSTE 良好的接口可扩展性能使得 STE-IMAS 中无需修改通讯协议组件就可以集成 AFDX、实时光纤网络、MTM-BUS 等与 IMA 相匹配的外部总线接口。同时其实时处理精度也从 DSTE 的 1 ms 级，提升到了 0.1 ms 级的粒度，保证了数据的实时通信。

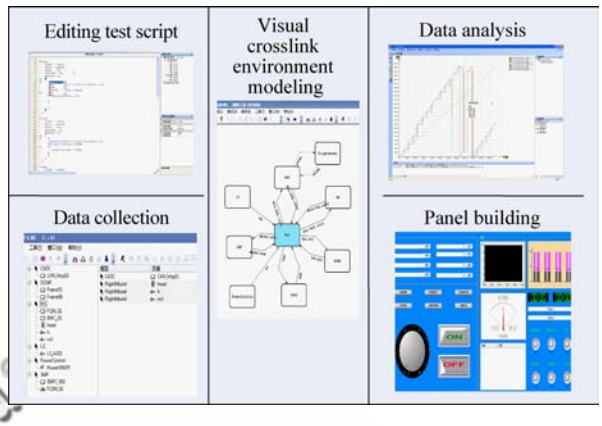

图 6 IMA 模块平台测试综合图例

Fig. 6 IMA platform testing diagram

被测对象通过 JTAG 与 IMA 仿真器连接，通过风河 VxWorks653 的仿真组件以及软件故障注入脚本完成对代码的插装和变异，以保证测试的覆盖率（如语句覆盖、分支覆盖等）。

利用 XML 技术描述被测对象 IMA 模块平台中的运行蓝图，并通过 VxWorks653 的仿真组件加载到目标平台上，完成目标平台的资源配置。软件故障注入则通过 GDB 调试运行的方式，利用脚本化的描述实现对被测系统软件的故障注入。目前，两者在 STE-IMAS 中均提供文本方式的人机交互界面，下一步将采用可视化的描述方式。

软件故障注入功能提供了 5 种故障模型：内存泄漏、空指针引用、数组越界、变量未初始化和非法计算。测试人员可根据设计需求应用故障注入脚本进行逻辑组合和扩充。

通过图形化的交联环境建模技术和灵活的脚本技术实现对 MTM-BUS 协议数据的控制（如周期数据的更新、事件性数据的反馈等），再结合软件故障模型实现对被测对象健康监控管理功能的测试和验证。

利用 STE-IMAS 完成某 IMA 原型模块的测试验证工作，在保证外部接口总线适配的前提下，同时保证了软件需求和代码结构的测试覆盖，并且通过软件故障模型实现对软件的故障覆盖，满

足了 DO-178B 对软件验证的目标要求。

5 结 论

本文讨论的仿真测试环境方案,既结合了黑盒与白盒测试工具优势,又满足了测试规范的目标和要求。现已完成原型设计与验证工作,通过工程实践表明其具有如下优势和意义:

1) IMA 软件功能验证过程中实时体现代码覆盖率的信息,更加有效保证了测试的充分性,并满足测试过程中"需求-代码-用例"追踪性验证目标。

2) 灰盒仿真测试环境的一体化应用,为 IMA 系统的综合集成验证提供了有利手段,避免了单纯的黑盒测试工具和白盒测试工具的简单叠加使用造成的信息不同步、激励不真实的缺点,丰富了测试方法。

3) 灰盒仿真测试环境故障注入的特点,有效地保障了 IMA 系统中的健康监控功能的验证和测试,保证了测试的充分性以及故障覆盖率要求,为装备质量的保证奠定的基础。

4)灰盒仿真测试环境既扩展升级了机载网络保证了接口的适配性,又提供了灵活易用的交联环境建模功能,为 IMA 测试验证工作提供便利的技术手段。

随着中国新一代航电系统的深入研究与实践以及型号任务的进一步发展,软件占据着愈来愈大的比例,软件测试的重要性也更趋明显,而仿真测试环境又是软件测试充分性的有力保障。本文所研究的 STE-IMAS 既继承了 DSTE 的优势,又具有较强的针对性,具有较好的实现基础和可行性,为软件的适航认证和验证提供必要的手段,必将成为未来新一代航电系统软件测试的重要工具。

参 考 文 献

[1] Xiong H G, Wang Z H. Advanced avionics integration technologies. Beijing: National Defence Industry Press, 2009: 14-15. (in Chinese)
熊华钢,王中华. 先进航空电子综合技术. 北京:国防工业出版社, 2009: 14-15.

[2] Xie W T. Digital avionics handbook avionics development and implementation (I). Beijing: Aviation Industry Press, 2010: 61-62 (in Chinese)
谢文涛. 数字航空电子技术(上). 北京:航空工业出版社, 2010: 61-62.

[3] RTCA DO-297. Integrated modular avionics (IMA) development guidance and certification considerations. Washington D. C.: Radio Technical Commission for Aeronautics, Inc(RTCA), 2005.

[4] RTCA DO-178B. Software Considerations in airborne systems and equipment certification. Washington D. C.: Radio Technical Commission for Aeronautics, Inc(RTCA), 1992.

[5] Boydston A, Lewis W. Qualification and reliability of complex electronic rotorcraft systems. AHS Specialists' Meeting on Systems Engineering. 2009.

[6] Bartley G, Lingberg B. Certification concerns of integrated modular avionics (IMA) systems. IEEE/AIAA 27th Digital Avionics Systems Conference. 2008: 1. E. 1-1 - 1. E. 1-12.

[7] Mcdonell R, Brackett R. Designing an open test software architecture featuring LM-STAR case study. IEEE Autotestcon Proceedings. 2004: 202-209.

[8] Hill B. Lockheed's F-22 raptor gets zapped by international date line. (2007-02-26). http://www.freerepublic.com/focus/fnews/1791574/posts.

[9] Chu W K, Zhang F M, Fan X G. Overview on software architecture of integrated modular avionic systems. Acta Aeronautica et Astronautica Sinica, 2009, 30(10): 935-942. (in Chinese)
褚文奎,张凤鸣,樊晓光. 综合模块化航空电子系统软件体系结构综述. 航空学报, 2009, 30(10): 935-942.

[10] Xu X G, Ye H. The design and implementation of inter-partition communication in avionics systems. Aeronautical Computer Technique, 2005, 35(1): 45-47. (in Chinese)
徐晓光,叶宏. 分区间通信在航空电子系统中的设计与实现. 航空计算技术, 2005, 35(1): 45-47.

[11] Zhang X H, Sun G X. Research of healthmonitor in high-security real-time operating system. Aeronautical Computer Technique, 2005, 35(4): 65-67. (in Chinese)
张晓红,孙高翔. 实时操作系统中健康监控技术研究. 航空计算技术, 2005, 35(4): 65-67.

[12] Chen Z J, Kong F E, Li W Q, et al. Study on flight control computer systems of advanced fighters. Acta Aeronautica et Astronautica Sinica, 2007, 28(4): 935-942. (in Chinese)
陈宗基,孔繁峨,李卫琪,等. 先进战斗机的飞行控制计算机系统研究. 航空学报, 2007, 28(4): 935-942.

[13] Zhang J, Lu Z X, Hu Y Y, et al. Perspective view of virtualization technologies for avionics system. Journal of Beijing University of Aeronautics and Astronautics, 2010, 36(2): 127-130. (in Chinese)
张炯,吕紫旭,胡彦彦,等. 虚拟化技术在综合化航电系统中的应用. 北京航空航天大学学报, 2010, 36(2): 127-130.

[14] Shen Y L, Cui X N, Ma J F, et al. Trust software technology in integrated avionics systems. Acta Aeronautica et Astronautica Sinica, 2009, 30(5): 938-945. (in Chinese)
沈玉龙，崔西宁，马建峰，等. 综合化航空电子系统可信软件技术. 航空学报，2009，30(5)：938-945.

[15] Liu C, Liu B, Ruan L. Software architecture of simulation testing environment for software in avionics. Acta Aeronautica et Astronautica Sinica, 2006, 27(5): 877-882. (in Chinese)
刘畅，刘斌，阮镰. 航空电子软件仿真测试环境软件体系结构研究. 航空学报，2006，27(5)：877-882.

[16] Ruan L, Liu B, Chen X S. Software reliability test and it's testing environment. Measurement and Control Technology, 2000,19(2):9-16. (in Chinese)
阮镰，刘斌，陈雪松. 软件可靠性测试及其仿真测试环境. 测控技术，2000，19(2)：9-16.

[17] Zhang L, Liu B, Lu M Y. Framework design of embedded software testing development environment. Journal of Beijing University of Aeronautics and Astronautics, 2005, 31(6):336-340. (in Chinese)
章亮，刘斌，陆民燕. 嵌入式软件测试开发环境的框架设计. 北京航空航天大学学报，2005，31(6)：336-340.

[18] Liu B, Gao X P, Lu M Y, et al. Research on embedded software reliability simulation testing system. Journal of Beijing University of Aeronautics and Astronautics, 2000, 26(4): 490-493. (in Chinese)
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. 北京航空航天大学学报，2000,26(4):490-493.

[19] Zhang F M, Chu W K, Fan X G, et al. Research on architecture of integrated modular avionics. Electronics Optics & Control, 2009,16(9): 47-51. (in Chinese)
张凤鸣，褚文奎，樊晓光，等. 综合模块化航空电子体系结构研究. 电光与控制，2009，16(9)：47-51.

[20] Aeronautical Radio, Inc. ARINC specification 653 avionics application software standard interface. Annapolis: Aeronautical Radio, Inc, 1997.

[21] Song L R, He F, Xiong H G. Real-time performance design of avionic blueprint system. Electronics Optics & Control, 2010, 16(9): 5-8. (in Chinese)
宋丽茹，何锋，熊华钢. 航空电子蓝图系统实时性设计. 电光与控制，2010，16(9)：5-8.

[22] Li Q Y, Liu B, Ruan L. Application of grey-box testing method in software reliability testing. Acta Aeronautica et Astronautica Sinica, 2002, 23(5): 455-458. (in Chinese)
李秋英，刘斌，阮镰. 灰盒测试方法在软件可靠性测试中的应用. 航空学报，2002，23(5)：455-458.

[23] Zheng R J. Computer software testing technique. Beijing: Tsinghua University Press, 1992: 44-45. (in Chinese)
郑人杰. 计算机软件测试技术. 北京：清华大学出版社，1992：44-45.

[24] Yin Y F, Liu B, Wang C. Execution engine for real-time embedded software test design and realization. Journal of Beijing University of Aeronautics and Astronautics, 2010, 36(6): 723-727. (in Chinese)
殷永峰，刘斌，王晨. 实时嵌入式软件测试执行引擎的设计与实现. 北京航空航天大学学报，2010，36(6)：723-727.

作者简介：

周庆 男，博士研究生。主要研究方向：软件可靠性、嵌入式软件测试环境、故障注入技术。
Tel：010-51736719
E-mail：zhouqing228@gmail.com

刘斌 男，博士，教授，博士生导师。主要研究方向：软件可靠性、软件故障学、嵌入式软件仿真测试环境等。
Tel：010-82339950
E-mail：liubin@buaa.edu.cn

余正伟 男，博士，工程师。主要研究方向：软件可靠性、嵌入式软件仿真测试环境等。
E-mail：buaayuzhengwei@163.com

冯时雨 男，硕士，工程师。主要研究方向：系统测试性、嵌入式软件测试性等。
E-mail：sharp_fsy@163.com

A Framework of Simulation Testing Environment for Integrated Modular Avionics Software

ZHOU Qing[1], LIU Bin[1,*], YU Zhengwei[1], FENG Shiyu[2]

1. *School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China*
2. *Beijing Aeronautical Technology Research Center, Beijing 100076, China*

Abstract: It is difficult for the traditional software testing environment to meet the requirements of integrated modular avionics (IMA) software testing and verification of the verification process results in RTCA-DO-178B. It is also difficult for the traditional software testing environment to achieve the goals of functional testing and validation of the IMA software, which includes testability design, health monitoring and redundancy management functions. This paper analyzes the characteristics of integrated modular avionics software, and describes the requirements of the testing environment aimed for IMA software testing. Then, it proposes a program of IMA software grey-box testing environment based on software fault injection. This program is more general, flexible and strong real-time than the traditional software testing environment.

Key words: integrated modular avionics; software; gray-box testing; simulation testing environment; software fault injection; DO-178B

Received: 2011-06-22; **Revised**: 2011-08-30; **Accepted**: 2011-11-29; **Published online**: 2011-12-09 17:25
URL: www.cnki.net/kcms/detail/11.1929.V.20111209.1725.002.html **DOI**: CNKI:11-1929/V.20111209.1725.002
Foundation item: National Defense Pre-research Foundation of China (513190802)
* **Corresponding author**. Tel.: 010-82339950 E-mail: liubin@buaa.edu.cn