

ATE 中的故障诊断系统设计与实现

翟颖焯¹, 周越文¹, 梁学明², 张 斌³

(1.空军工程大学航空航天工程学院,陕西西安,710038;2.北京航空工程技术研究中心,北京,100076;
3.94162 部队,陕西西安,710614)

摘要 针对目前基于 ATE 的故障诊断系统存在的问题,在类似研究的基础上,给出了一种通用故障诊断系统软件设计方案。首先建立了故障诊断系统功能模型,以此为基础建立了基于 XML 语言的故障诊断系统框架及故障诊断过程中所需及产生的各类信息模型,研究了基于 ATE 的采用粗糙集方法和故障树分析法的故障诊断方法,并以某型飞机电子对抗设备综合测试与故障诊断系统为例建立了软件系统总体框架,介绍了各功能模块设计方案,最终完成了故障诊断软件程序的编写。

关键词 故障诊断;自动测试设备;软件系统;XML 语言

DOI 10.3969/j.issn.1009-3516.2014.01.006

中图分类号 V221 **文献标志码** A **文章编号** 1009-3516(2014)01-0024-05

A Design and Implementation of Fault Diagnosis System in Automatic Test Equipment

ZHAI Ying-ye¹, ZHOU Yue-wen¹, LIANG Xue-ming², ZHANG Bin³

(1.Aeronautics and Astronautics Engineering College, Air Force Engineering University, Xi'an 710038, China; 2.Beijing Aeronautical Engineering Technology Research Center, Beijing 100076, China; 3.Unit 94162, Xi'an 710614, China)

Abstract: In order to solve the problem that the use of the existing automatic test equipment can not conduct the task of fault diagnosis very well, a design plan of fault diagnosis software system is proposed. Firstly, a fault diagnosis system framework and information model are given out. Then, taking electron warfare equipment complex test and fault diagnosis system in a certain type of aircraft as example, the function module design is described and a software system framework is built up. And this design plan is applied successfully to an electron warfare equipment complex test and fault diagnosis system in a certain type of aircraft, and good results are achieved, which shows that the plan is effective.

Key words: fault diagnosis; automatic test equipment(ATE); software system; XML language

目前国内市场上的 ATE(Automatic Test Equipment, 自动测试设备)软件系统中,测试功能开发已经十分完备,故障诊断系统也有开发成果,如钟

宇等提出了基于 Labview 编程的神经网络故障诊断模块^[1],黄鑫等人提出了数字电路板的测试及故障诊断系统设计^[2]。以上方法大都不同程度存在系

收稿日期: 2013-06-08

作者简介: 翟颖焯(1989—),男,陕西咸阳市人,硕士生,主要从事航空装备检测自动化及智能化方向的研究. E-mail: zhai2344@sina.com

引用格式: 翟颖焯,周越文,梁学明,等. ATE 中的故障诊断系统设计与实现[J]. 空军工程大学学报:自然科学版,2014,15(1):24-28. ZHAI Yingye, ZHOU Yuewen, LIANG Xueming, et al. A design and implementation of fault diagnosis system in automatic test equipment[J]. Journal of air force engineering university: natural science edition, 2014, 15(1): 24-28.

统较为封闭,信息交互性、故障诊断系统可靠性、通用性及可扩展性较差的问题。本文在类似研究基础上,给出了一种采用通用信息模型、模块化的软件设计思想的故障诊断系统设计方案,使系统拥有了良好的数据交互性、诊断系统可扩展性及通用性。

1 功能模型及信息模型的建立

1.1 功能模型

故障诊断系统需求要求该系统能够通过综合复杂装备的多方面信息,在进行信息汇总,数据预处理及数据分析后,利用信息融合进行故障诊断,将结果输出并对维修方案进行决策。其中,结果及维修方案被存储到数据库,为下次诊断提供历史数据,并进行传输与共享。根据系统框架,建立 ATE 中的故障诊断系统功能模型见图 1。

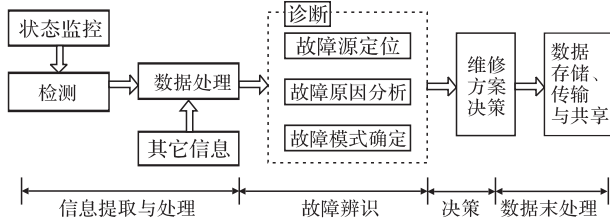


图 1 故障诊断系统功能模型

Fig.1 Functional model of fault diagnosis system

该模型功能流程如下:首先通过各种传感器采集复杂装备的各类信号,再将所有信息进行汇总后进行数据预处理及信息融合。利用得到的复杂装备信息,完成复杂装备的故障诊断,根据诊断得出的结论,提出修理、更换等维修活动决策,最后将测试信息、故障诊断信息及决策信息存储到数据库,以便传输、共享及为下一次诊断提供支持。

1.2 信息模型

信息模型采用 XML(eXtensible Markup Language, 可扩展标记语言)描述。XML 是一种元标记语言,具备良好的通用性和可扩展性,可以作为多个不同种类数据源的全局描述语言来使用^[3]。选择 XML 作为集成异构数据源的全局描述语言。

采用 XML 语言对 UUT(Unit Under Test, 被测件)的测试信息、装备设计信息、环境信息、维护保障信息、诊断信息、故障信息等进行描述。以诊断信息为例,诊断信息存储诊断过程中的动态信息^[4]。

诊断信息的 XML 模型数据见图 2。其中诊断项目指被诊断的设备及系统的名称;诊断描述指故障诊断推理的过程描述;诊断级别指故障诊断的定位层级;故障描述包括故障现象、故障位置等;相关测试信息描述包括测试项目编码、被测件名称、测试

项目名称、测试内容、测试点、测试输入(激励信号)、测试输出等;诊断结果信息包括故障标识、故障描述、诊断可信度、诊断价值度等。

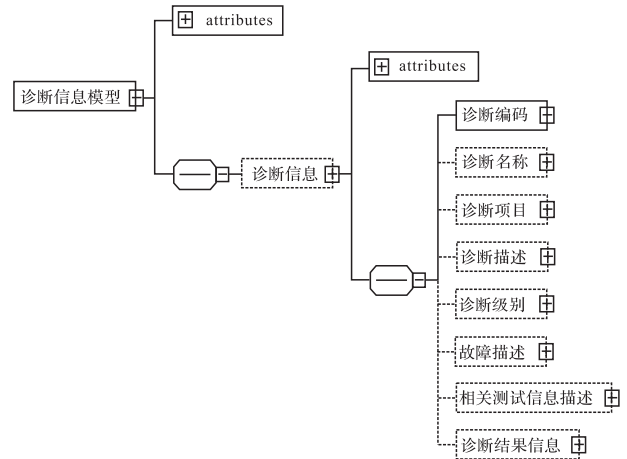


图 2 诊断信息模型

Fig.2 Diagnosis information model

2 故障诊断

复杂装备系统的故障突出表现为以下 4 个特征:层次性、传播性、相关性及不确定性^[5]。因此,在决策规则获取及故障诊断时,故障点知识获取、故障诊断方法的选择及对不协调、不完备测试信息的处理尤为重要。为了确保故障诊断的快速有效进行,本文选择了故障树及粗糙集 2 种故障诊断方法。

2.1 故障知识获取

对测试样本作如下划分:在一个部件中,可能有 n 个故障源,并设 $D = \{d_1, d_2, d_n\}$ 为故障源集合,其中 d_i 表示一个可能的故障源, $i = 1, 2, \dots, n$; 共有 m 个可测信号源, $C = \{c_1, c_2, \dots, c_m\}$ 为可测信号源集合,其中 c_j 为一个可测信号源信号测试结果(通过为“0”或不通过为“1”), $j = 1, 2, \dots, n$; $U = \{e_1, e_2, \dots, e_k\}$ 为测试过程中获取的测试样本集合。

利用粗糙集进行知识获取的步骤如下:

步骤 1 选择可测最底层故障源作为决策属性,可测信号测试结果作为条件属性,对 UUT 进行测试,得到测试结果并进行记录,得到表 1;

表 1 获取知识决策表

Tab.1 Access to knowledge decision tables

U	c_1	c_2	\dots	c_m	D
e_1	$c_1(1)$	$c_2(1)$	\dots	$c_m(1)$	d_1
e_2	$c_1(2)$	$c_2(2)$	\dots	$c_m(2)$	d_2
\vdots	\vdots	\vdots	\dots	\vdots	\vdots
e_n	$c_1(n)$	$c_2(n)$	\dots	$c_m(n)$	d_n

步骤 2 处理冗余及不相容样本、不完备样本;

步骤3 进行信息约简。

2.2 故障诊断方法

2.2.1 基于粗糙集的正反向混合故障诊断方法

优先采用诊断精度较高的基于粗糙集的正反向混合故障诊断方法,其基本过程如下:

步骤1 在测试过程中存在不通过项,认为装备运行状况发生异常,转入下一步;

步骤2 对系统软件运行状况进行诊断。在 h 个测试样本 $(\theta_1, \forall \theta_2, \forall \theta_h)$ 中, $\varphi_d = (d_j, 1)$ 如果发现符合故障诊断决策规则的,则认为装备在系统软件运行上发生了故障,否则转入下一步;

步骤3 在关于故障源信号的故障诊断决策规则中,寻找 h 在种状态下 $(\theta_1, \forall \theta_2, \forall \theta_h)$ 符合的决策规则,如果有,认为所符合的决策规则指向的 d_j 发生了故障,否则转入下一步;

步骤4 搜索所有可能造成 c_j 故障的 d_j , 并寻找当 $\varphi_d = (d_j, 1)$ 同时 $e_n \notin (\theta_1, \forall \theta_2, \forall \theta_h)$ 的测试样本,如果有测试样本符合包含的规则,认为该规则所指向的 d_j 发生了故障,否则转入下一步;

步骤5 判定粗糙集故障诊断失效,开始采用基于故障树的二叉树 DFS 故障诊断方法。

2.2.2 基于 BDD 化简的二叉树故障诊断方法

步骤如下:

步骤1 遍历测试样本及故障树模块集(为了提高故障诊断效率,故障树二叉树模型为分模块存储,每一模块代表一个部件的故障树,在故障诊断过程中,通过上级模块诊断结果判断进行下一步诊断的下级模块),寻找所有包括未通过测试项目的故障树二叉树模型;

步骤2 调出图3的被测件二叉树模型,被测件二叉树模型通过 BDD 技术^[6]建立,是对故障树模型的简化,仅显示了对底事件判断,可以有效提高故障诊断效率。图中, D_i 表示故障树底事件,即测试设备获取的底层信号,不通过为“1”,通过为“0”,代表一个部件发生故障或未发生故障;

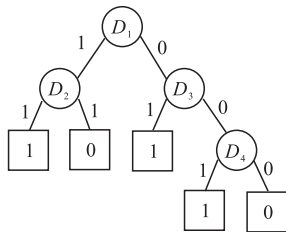


图3 二叉树模型

Fig.3 Binary tree model

步骤3 对二叉树进行搜索,并遵守以下规则:

① 搜索从项节点开始;② 在搜索到达的节点(故障

树底事件),若该底事件在测试中通过,搜索到达该节点右下节点(通过“0”路径);若该底事件在测试中不通过,搜索到达该节点左下节点(通过“1”路径);③ 当搜索到达底部“1”或“0”时结束。重复步骤3直到完成所有二叉树的搜索;

步骤4 进行搜索结果分析。

2.3 不协调及不完备信息处理

针对测试信息不协调的情况,基于测试设备的故障诊断信息系统可以重复提取样本信息,并且在测试过程中可以积累大量样本信息以供故障诊断系统利用,因此,可以选择多数优先原则,即选择覆盖多数样本的规则,完成对不相容信息的处理。

在测试过程中,通常不会出现频繁和大量的数据遗漏现象,一旦出现这种问题,一般认为出现遗漏现象的系统(部件)已经发生了故障或测试设备本身存在故障。因此,针对测试信息不完备的情况,在系统软件中一般采用以下2种:① 进行重复测试以补齐缺失数据;② 对大量数据遗漏现象并无法补齐的,判断发生数据遗漏的部件故障。

3 故障诊断系统实现

本文以某型飞机电子对抗设备综合测试与故障诊断系统为例,介绍 ATE 中的故障诊断系统设计与实现。某型飞机电子对抗设备具有综合化程度高、系统复杂、系统之间信息交流多等特点。因此,为了实现快速、准确的故障发现、隔离、定位与排除,必须在 ATE 中建立有效的故障诊断机制。

3.1 系统硬件设计

本文利用和参考了现有的测试系统设计成果,主要选择 COTS(Commercial Off The Shelf,货架产品)进行研发。

硬件原理见图4。

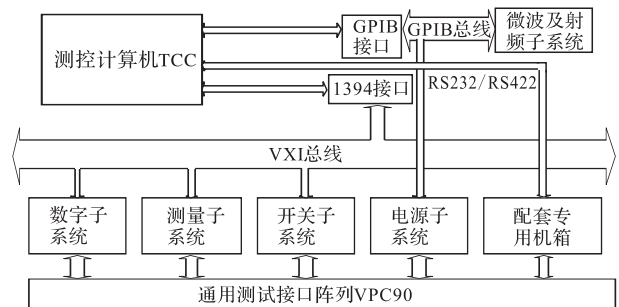


图4 ATE 硬件原理图

Fig.4 Automatic test system hardware pinciple

3.2 软件系统总体框架建立

基于硬件系统建立软件系统总体框架,见图 5。

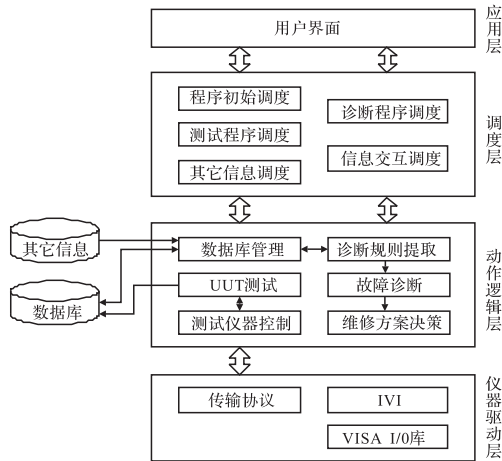


图 5 软件系统总体框架

Fig.5 Software system overall framework

- 1)应用层:为用户直接进行测试及故障诊断过程操作的程序界面。
- 2)调度层:主要接收用户操作命令,并根据命令控制测试及故障诊断系统实现基本功能动作。
- 3)动作逻辑层:完成了测试及故障诊断系统最基本动作的封装,基本动作通过与数据库和其它输入信息的交互,供调度层使用。
- 4)仪器驱动层:主要由所使用的 COTS 产品驱动程序及自研设备的驱动程度组成。

系统开发环境为 Visual Studio 2010,采用面向对象框架技术^[7-8]开发系统软件。该软件程序具有开发和维护迅速、可移植性强、复用性好等优势。

系统的文档视图类图见图 6。系统采用 MDI (多文档视图)类型的文档视图框架,系统软件需要 4 类用户界面:测试界面、诊断界面、信息交互界面和历史纪录界面,故采用 4 个文档模板。

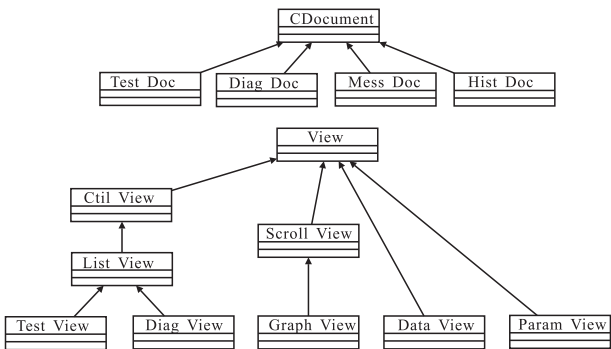


图 6 文档视图类图

Fig.6 Document view class

3.3 功能模块设计

软件功能模块主要包括测试类及故障诊断类,其中测试文档 CTestDoc 的类设计见图 7。故障诊断文档 CDiagDoc 的类设计见图 8。

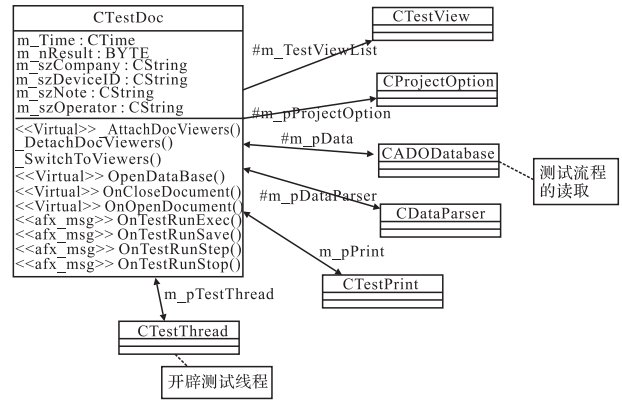


图 7 测试功能类设计

Fig.7 Test function class

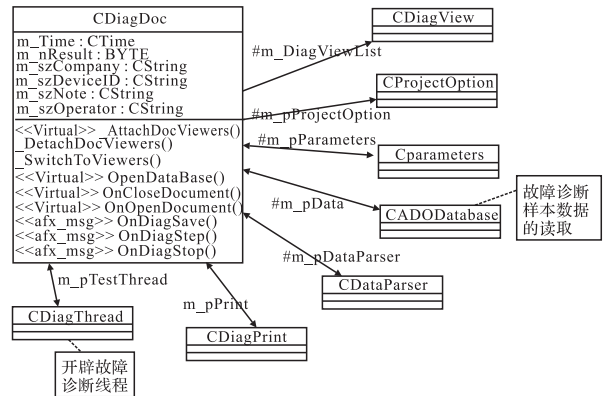


图 8 诊断功能类设计

Fig.8 Fault diagnosis function class

3.4 软件控制界面实现

系统的主界面提供系统测试、故障诊断、信息交互和历史查询的功能选择按钮。

- 1)系统测试主要完成测试功能,并将测试存储进数据库以方便故障诊断程序调用;
- 2)故障诊断主要完成对电子装备的故障诊断,包括基于故障树、粗糙集、测试信息不协调条件下、测试信息不完备条件下的故障诊断;
- 3)信息交互用于提供用户对包括测试信息、故障诊断信息等的处理和修改,包括信息添加、删除、网络共享等;
- 4)历史查询主要用于查询维护保障信息中的测试、故障诊断、使用维护等历史信息。

故障诊断界面见图 9,可以根据不同测试结果选择诊断方法。



图9 故障诊断界面

Fig.9 Fault diagnosis interface

4 结语

本文在现有 ATE 研究基础上,完成了一种基于测试设备的故障诊断系统通用设计方案。在建立通用信息模型基础上,利用粗糙集方法和故障树分析法完成了电子设备的故障诊断。但由于复杂电子设备故障的关联性及其复杂性,可能存在故障覆盖率不足及误诊误报的问题,因此需要在未来使用及研究过程中对诊断规则进行动态完善。所取得的初步成果已经应用于某型飞机电子对抗设备综合测试与故障诊断系统,并实现了快速故障诊断的目的。

参考文献(References):

- [1] 钟宇,白云. 某自动测试系统神经网络故障诊断模块设计与实现 [J]. 航天控制,2011, 29(5): 58-76.
ZHONG Yu, BAI Yun, Design and implementation of fault diagnosis module based on neural network in certain ATS [J]. Aerospace control,2011, 29(5): 58-76.(in Chinese)
- [2] 黄鑫,常天庆. 数字电路板自动测试与故障诊断系统的设计与实现 [J]. 计算机测试与控制, 2010, 18(7): 1512-1514.
HUANG Xin, CHANG Tianqing. Design and realization of automatic test and fault diagnosis system for digital circuits[J]. Computer measurement & control, 2010, 18(7): 1512-1514.(in Chinese)
- [3] Elliott Rusty Harold. XML 宝典[M]. 马云,钟萍,译.北京:电子工业出版社, 2002.
Elliott Rusty Harold. XML handbook[M]. MA Yun, ZHONG Ping, translate. Beijing: Electric industry press,2002.(in Chinese)
- [4] 欧建,李艾华. 复杂机电设备故障诊断专家系统的设计与实现 [J]. 计算机测试与控制,2011, 19(12): 2890-2915.
OU Jian, LI Aihua. Design and implementation of fault diagnosis expert systems for complex mechatronic equipment[J]. Computer measurement & control, 2011, 19(12): 2890-2915.(in Chinese)
- [5] 李伟. 复杂系统的故障诊断技术现状及其发展趋势 [J]. 计算机仿真,2004, 21(10): 4-7.
LI Wei. Advance of intelligent fault diagnosis for complex system and its present situation[J]. Computer simulation, 2004, 21(10): 4-7.(in Chinese)
- [6] Bryant R E. Graph-based algorithm for boolean function manipulation[J]. IEEE transaction on computers, 1986,35(8): 677-691.
- [7] Zakarian A, Kusiak A. Process analysis and engineering [J]. Computer & industry engineering,2001, 41(2): 135-150.
- [8] Coplien J, Hoffman D, Weiss D. Commonality and variability in software engineering [J]. IEEE software,1998, 15(6): 356-362.

(编辑:徐敏)