

基于 ATML 标准的 SCATS 在机载电子系统通用自动测试设备中的应用

吴国宝¹, 宋帆¹, 曾俊华¹, 游洪顺²

(1. 中国直升机设计研究所, 江西 景德镇 333001; 2. 成都天奥测控技术有限公司, 成都 611731)

摘要: SCATS 是一款由成都天奥测控技术有限公司开发的, 基于最新 AMTL 标准的面向信号的国产自动测试程序集开发软件; ATML 标准的显著特征, 便是采用了可扩展标记语言 (XML), 以实现测试信息描述的标准化和测试信息的可交互, 通过生成的基于 ATML 标准的描述文档, 实现对被测对象的特征信息、测试信号、测试路径、测试策略、故障诊断等内容的表述, 进而实现通用测试设备的通用化设计, 有效克服了传统面向信号的测试设备测试程序集开发软件存在的语言形式复杂、信号模型少、软件架构开放性差等不足; SCATS 软件采用纯中文开发界面, 测试程序集采用无代码开发, 软件界面友好、可读性强, 大大降低了测试程序集开发人员的技術门槛, 内置信号模型丰富可满足现代机载电子系统测试需求; 通过在机载电子系统通用测试设备中的应用表明, 使用 SCATS 软件开发的机载电子系统测试程序集具有开发速度快、故障定位准确、通用性和可移植性好等特点。

关键词: ATML; SCATS; 机载电子系统; 测试设备

Application of SCATS Based on ATML Standard in General Testing Equipment for Airborne Electronic Systems

WU Guobao¹, SONG Fan¹, ZENG Junhua¹, YOU Hongshun²

(1. China Helicopter Research and Development Institute, Jingdezhen 333001, China;

2. Chengdu Spaceon T&C Technology Co., Ltd., Chengdu 611731, China)

Abstract: SCATS is a domestic automatic testing program set development software for signals based on the latest AMTL standard developed by Chengdu Spaceon T&C Technology Co. Ltd. The notable feature of the automated test markup language (ATML) standard is that the extensible markup language (XML) is adopted to achieve the standardized description and information exchange of testing information. The description document based on the ATML standard is established, it achieves the sharing and exchanges of the signal feature information, test results, test path, test strategy, fault diagnosis, thereby achieving the universal design of universal testing equipment. It effectively overcomes the shortcomings of the signal-oriented test equipment and test program set development software, such as PAWS, this shortcomings include complex language form, few signal models, poor openness of software architecture, etc. The SCATS software adopts a pure Chinese development interface, with the test program set no code. The software interface is friendly and readable, greatly reducing the technical threshold for test program set developers. The built-in signal model is rich and can meet the testing needs of modern airborne electronic systems. The application shows that in the universal testing equipment for airborne electronic systems, the test program set for the airborne electronic systems developed by using SCATS software has the characteristics of fast development speed, accurate fault location, good universality and portability.

Keywords: ATML; SCATS; airborne electronic systems; test equipment

0 引言

现代飞机机载电子系统功能性能先进、集成度高、系统内外信号交联复杂^[1-2]。机载电子系统在装机使用过程中, 当发生故障时, 仅靠机内测试 (BIT) 无法实现故障 100% 检测和 100% 隔离 (隔离到单个外场可更换单元 (LRU) 或外场可更换模块 (LRM)), 这给飞机的外场维修保障带来了不小挑战, 将直接影响飞机的出勤率^[3]。

机载电子系统通用自动测试设备, 主要用于辅助机载

电子系统的外场故障检测^[4-6], 其作用为: 当装机的机载电子系统发生故障, 且 BIT 无法实现故障检测或将故障隔离到单个 LRU/LRM 时^[7], 使用该设备能够将故障检测并将故障定位到单个 LRU/LRM, 以便指导维护人员在外场对故障件进行快速更换, 确保飞机的出勤率。

传统针对机载电子系统研制的通用自动测试设备, 其测试程序集 (TPS) 以面向仪器为核心^[8], 存在一定的不足, 主要表现在 TPS 开发周期长^[9], 必须在完成硬件资源的选型后才能开始 TPS 开发, 且 TPS 开发人员必须对所选

收稿日期: 2023-04-05; 修回日期: 2023-05-30。

作者简介: 吴国宝 (1988-), 男, 硕士研究生, 工程师。

引用格式: 吴国宝, 宋帆, 曾俊华, 等. 基于 ATML 标准的 SCATS 在机载电子系统通用自动测试设备中的应用[J]. 计算机测量与控制, 2024, 32(3): 340-345, 352.

硬件资源的使用必须很了解。面向仪器的测试程序 (TP), 当新增测试需求或对通用自动测试设备硬件进行新增/改型时, 需 TPS 开发人员对已完成鉴定的所有 TPS 进行适应性改进、调试、验收, 整个开发周期将大大增加, 非常不利于通用自动测试设备的升级维护。

通用自动测试设备测试程序集由面向仪器的开发, 过渡到面向信号的开发, 是自动测试领域的一次重要转变^[10], 面向信号的测试程序集开发, 其优势主要体现在测试程序集开发人员只需关注被测对象的测试逻辑及特性信号本身, 而无需关心测试工作站中仪器的规格、型号、性能参数及具体操作, 是保证测试程序集具备通用性和跨平台可移植性的核心特征^[11]。

早期流行的面向信号的测试程序集开发软件, 有美国 TYX 公司研发的 PAWS 软件, 该软件使用 ATLAS 语言的进行驱动及测试程序集的开发。ATLAS 作为一种面向信号的标准化测试编程语言, 其在国内外机载电子设备自动测试领域得到了广泛应用^[12], 但也存在开发语言结构复杂、信号模型少、软件架构开放性差等不足。基于此, 电气与电子工程师协会 (IEEE) 于 2002 年提出了一种新的面向信号的自动测试标记语言 (ATML) 标准, ATML 标准的显著特征, 便是采用了可扩展标记语言 (XML), 以实现测试信息描述的标准化和测试信息的可交互^[13], 通过生成的基于 ATML 标准的描述文档, 实现对被测对象的特征信息, 测试信号, 测试路径, 测试策略, 故障诊断等的表述, 测试结果记录、显示, 故障诊断结果记录^[14]、显示等信息的交互, 从而实现通用自动测试设备的通用化设计。

ATML 标准定义了公共信息、测试信息描述、仪器信息描述、适配器信息、测试工作站信息、被测对象特征描述、测试信息配置、测试结论、故障诊断共 9 个子组件标准^[15], 囊括了通用自动测试设备研制的每一个环节, 提供了一套完整的面向信号测试的测试信号交互架构^[16]。基于此 ATML 可实现不同通用自动测试设备之间的测试信息的交互, 现实 TPS 在不同硬件平台上的移植, 支持故障诊断结果跨平台调用^[17]。ATML 标准与通用测试设备的对应关系如图 1 所示。

SCATS 是一款由成都天奥测控技术有限公司开发的国产面向信号的 TPS 开发软件, SCATS 软件基于最新 AMTL 标准 (IEEE 1232-2010、IEEE 1641-2010、IEEE 1671-2010), SCATS 采用了多种国内领先的测试软件开发技术, 包括标准信号库构造技术; 纯中文界面的无代码测试程序集开发、调试技术; 基于标准模型的测试程序运行

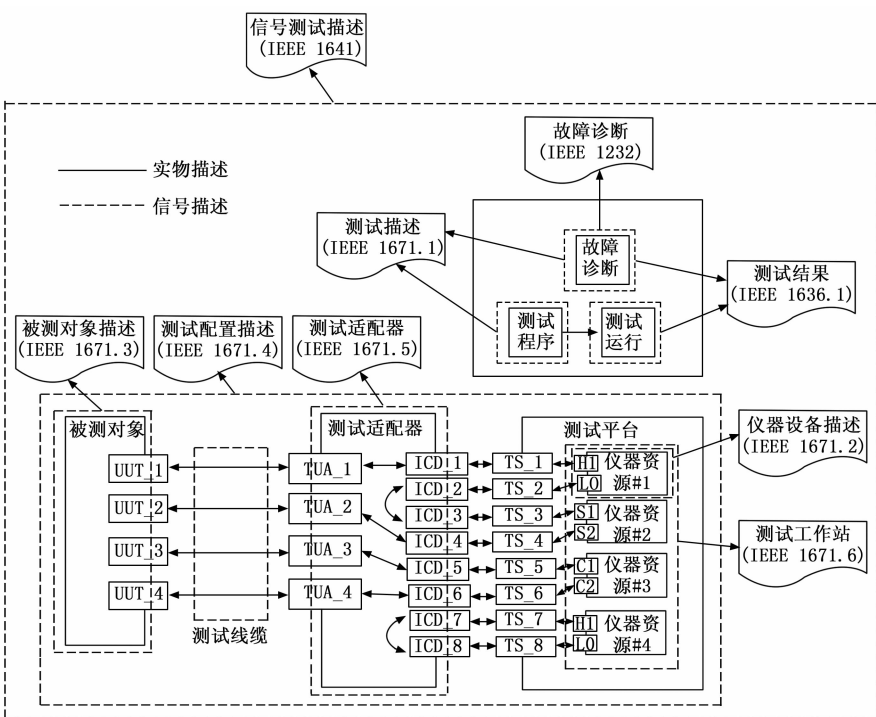


图 1 ATML 标准与通用测试设备的对应关系图

引擎技术等。SCATS 软件采用纯中文开发界面, 测试程序集采用无代码开发, 软件界面友好、可读性强, 大大降低了测试程序集开发人员的技术门槛, 内置信号模型丰富可满足现代机载电子系统测试需求。通过在机载电子系统通用自动测试设备研制中的应用表明, 使用 SCATS 软件开发的被测机载电子系统 TPS 具有开发速度快、故障定位准确、通用性和可移植性好等特点。

1 SCATS 架构及原理

1.1 SCATS 软件架构

SCATS 软件, 采用层次化体系架构, 自下至上分为物理层、驱动层、标准层、系统层、开发层和管理运行层, 层次结构示意图如图 2 所示。

物理层: 包含软件运行所依赖的所有硬件资源基础, 如各类标准总线下的模块化仪器, 通用或专用的台式仪器 (包括模拟器等)、控制器、适配器和测试线缆等。

驱动层: 主要面向驱动开发人员, 基于 Python 或 C 语言进行驱动程序的开发、调试, 其中包括面向信号的驱动框架生成工具, 基于 CVI 或 Visual Studio Code 的驱动开发环境, 生成的驱动动态库。

标准层: 包含最新 ATML 标准, 这些标准构成了软件平台的数据结构基础, 具体如下。

- 1) IEEE 1671-2010: 自动测试标记语言 (ATML) 的主标准, 其中定义了各个子标准中可能用到的公共类型^[18]。
- 2) IEEE 1671.1-2017: 测试描述标准, 其中主要定义了完整描述一个测试需求的必要信息。
- 3) IEEE 1671.2-2012: 仪器描述标准, 其中主要定义了

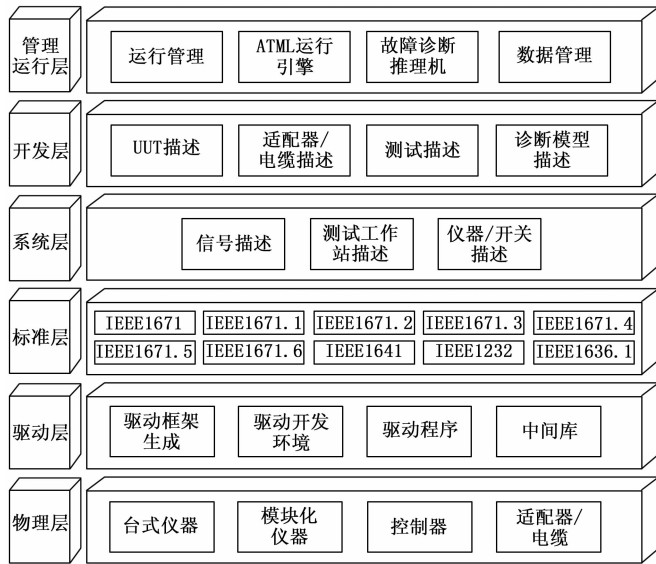


图 2 SCATS 层次结构框图

具相对应，管理运行层，主要提供了用户权限管理功能、测试程序集的运行功能、测试程序集管理功能（增加、删除、分组）、平台硬件资源管理功能、故障诊断推理功能以及测试数据管理功能等。

基于上述架构，SCATS 的组成如图 3 所示，其中包含两个主要工具以及 15 个子工具，可覆盖用户及系统需求。

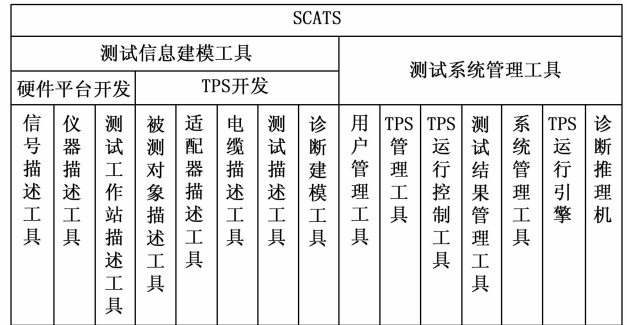


图 3 软件组成示意图

一个仪器的资源、能力和端口，以及它们之间的映射关系。

4) IEEE 1671.3-2017: 被测对象描述标准，其中主要定义了一个被测对象的识别信息和接口信息。

5) IEEE 1671.4-2014: 测试配置描述标准，用于配置一个 TPS 所需的软硬件资源。

6) IEEE 1671.5-2015: 测试适配器描述标准，其中主要定义了一个测试适配器（或线缆）的接口信息和连线信息。

7) IEEE 1671.6-2015: 测试工作站描述标准，其中主要定义了通用测试设备硬件平台所包含的仪器、功能模块、开关资源，以及对外的测试接口，此外还包含这些资源和端口的互连关系。

8) IEEE 1641-2010: 信号和测试定义标准，给出了针对一个所需的信号进行建模的通用方法，并且提供了许多有用的预定义信号，是面向信号测试的根本遵循^[19]。

9) IEEE 1232-2010: 测试相关的人工智能服务信息交换 (AI-ESTATE) 标准，给出了与故障诊断相关的数据结构，并给出了若干实用的诊断模型，例如 D-矩阵模型^[20]。

10) IEEE 1636.1-2018: 测试结果信息描述标准，其中定义了测试结果相关的信息，如测试时间、测试项目、测试判据、测试数据、测试结论等。

系统层：主要是面向通用自动测试设备的开发人员，是一种测试信号的建模工具，用于对测试所需的各类信号，各类仪器、开关以及集成各类仪器、开关的测试工作站进行建模。每台通用自动测试设备都含有这些信息，这些信息是通用自动测试设备的特有信息，一般是不可移植的。

开发层：主要是面向测试程序集的开发人员，是一种测试逻辑的建模工具，针对具体的被测机载电子系统，描述该被测机载电子系统本身的一些信息（产品名称、型号、接口、LRU 组成等）、相应的测试适配器、测试线缆信息（针脚、端口）、以及故障诊断模型（D-矩阵）信息等。

管理运行层：主要是面向用户的，与测试系统管理工

具相对应，管理运行层，主要提供了用户权限管理功能、测试程序集的运行功能、测试程序集管理功能（增加、删除、分组）、平台硬件资源管理功能、故障诊断推理功能以及测试数据管理功能等。

3 个硬件平台开发子工具分别为：
1) 基于 IEEE 1641-2010 标准的信号描述工具，主要提供标准信号的查询以及自定义信号的管理功能（即查、增、删、改等）。

2) 基于 IEEE 1671.2-2012 标准的仪器描述工具，能够描述仪器的接口信息、信号能力和资源，以及它们之间的映射关系和连线等信息。支持创建仪器驱动程序开发框架。

3) 基于 IEEE 1671.6-2015 标准的测试工作站描述工具，通过引用仪器描述模型将其组织称为一个测试设备，支持测试设备接口（ICA）和连线信息的图形化建模。

5 个 TPS 开发子工具分别为：

1) 基于 IEEE 1671.3-2017 标准的被测对象描述工具，能够提供被测对象设备的外观展示、支持被测对象接口的图形化建模，并提供方便的针脚、端口批量编辑功能。

2) 基于 IEEE 1671.1-2017 的测试描述工具，能够对测试项目和测试流程进行图形化建模。以面向信号的方式定义一个测试项目中具体的测试操作信息，支持标准中规定的所有信号操作类型以及若干实用的扩展信号操作。

3) 基于 IEEE 1671.5-2015 的适配器描述工具，能够提供适配器设备的外观展示、支持适配器接口的图形化建模，并提供方便的针脚、端口批量编辑功能。能够将测试工作站描述中的 ICA 信息导入为适配器的背板接口信息。

4) 基于 IEEE 1671.5-2015 的电缆描述工具。与适配器使用同一个标准，并提供相似的功能。该工具支持直接导入被测对象接口信息和适配器前面板接口信息作为电缆两端的接口信息。

5) 诊断建模工具，支持经典的 D-矩阵模型。能够从被测对象描述中导入 SRU 信息（这些 SRU 是最小故障隔离单元）和故障信息以及从测试描述中导入用于识别故障的测试结果信息。

7 个测试系统管理子工具分别为：

1) 用户管理工具, 用于管理软件操作人员的信息, 并对其权限进行限定。只有合法用户才能登陆软件, 且当前用户只能管理本人创建的 TPS 和测试结果等信息。

2) TPS 管理工具, 提供 TPS 的安装、卸载和附加信息设置等功能。此外, 为方便查找 TPS, 还提供按关键词过滤、排序和分组显示功能。

3) TPS 运行控制工具, 为当前 TP 提供测试流程选择、启动、暂停、继续和停止测试控制功能, 支持测试流程节点的状态可视化以及测试过程和结果信息的即时显示。

4) 测试结果管理工具, 用于保存测试结果信息, 对历史测试结果信息进行查看、删除、生成测试报告和故障诊断, 将数据库中的测试结果记录导出为符合 IEEE 1636.1-2018 标准的 XML 文档。

5) 系统管理工具, 查看测试工作站中的仪器, 可设置仪器的可用状态。提供仪器资源的自动化自检功能。

6) TPS 运行引擎, 解析并加载所有必要的 ATML 描述文件, 并按给定的测试流程或测试项目解释执行测试描述中包含的信号操作。具有动态资源匹配和信号路由搜索功能, 支持面向信号的自动测试和指定仪器的测试。

7) 诊断推理机, 结合当前测试结果和诊断模型信息, 对潜在故障进行推断, 给出故障诊断结论, 支持生成故障诊断报告。

1.2 测试信息建模工作原理

测试信息建模的工作原理如图 4 所示, 首先根据被测对象测试需求, 视情使用信号描述工具扩展当前信号库, 以支持仪器和测试描述中对信号的需求。

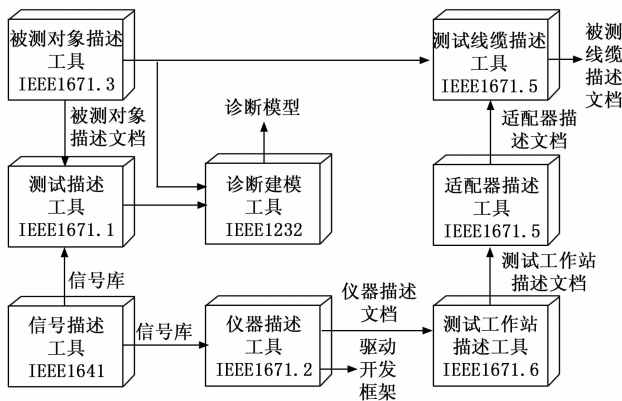


图 4 测试信息建模原理示意图

使用仪器描述工具对每一个仪器 (开关) 资源进行建模, 在描述仪器的能力时需要从信号库中获取期望的信号类型。根据仪器能力的构成, 生成驱动程序开发框架, 然后由驱动开发工程师完成驱动功能的实现, 最终生成可调用的驱动程序库。

测试工作站描述工具引用已有的仪器描述, 将其纳入测试工作站的组成部分, 此外对对外接口信息进行描述, 以方便适配器描述工具导入背板接口信息。

采用被测对象描述工具对被测对象的基本信息、接口

信息、内部组成和故障信息进行描述, 生成被测对象描述文档。

测试线缆描述工具导入适配器和被测对象的接口信息, 然后再将两端接口上的端口或针脚连接起来。

测试描述工具对测试需求的细节进行建模, 在此期间可能需从被测对象接口信息中获取连接针脚信息以及从信号库中获取描述信号需求的信号类型。

诊断建模工具分别从被测对象描述和测试描述文档中读取与故障诊断相关的故障定义和测试结果或测试指标等内容, 然后建立它们的依赖性关系, 生成诊断模型描述, 以备诊断推理机使用。

1.3 测试系统管理原理

测试系统管理的工作原理如图 5 所示。首先在测试服务器上启动自动测试服务, 此时 TPS 运行引擎处于待命状态。

用户启动客户端, 输入用户名和密码, 验证送过后登录主界面: 此时 TPS 运行引擎通过系统配置中指定的测试工作站信息, 加载相应的信号库、测试工作站和仪器描述。

用户通过 TPS 管理工具选择一个要执行的 TPS, 同时将当前 TPS 相关的被测对象描述、适配器描述、测试线缆描述和测试描述信息传递给 TPS 运行控制工具。

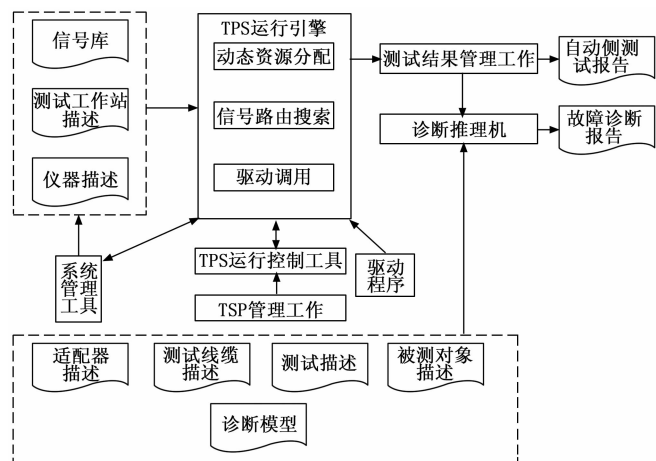


图 5 测试系统管理原理示意图

用户在 TPS 运行控制工具中选择一个测试流程 (或者通过自定义串行流程临时组织一个测试流程), 启动自动测试。TPS 运行引擎根据当前测试流程中的测试项目, 按照面向信号的自动测试原理依次解释执行其中包含的信号操作。

动态资源分配功能根据信号需求查找测试工作站中可用的仪器资源, 将仪器能力与信号需求进行比较, 若其能力包含该需求, 则匹配成功。对于多个仪器匹配当前信号需求的情况, 由资源分配算法根据匹配度、历史经验或测试成本消耗等因素优选一个最佳的仪器。

信号路由搜索算法是根据当前开关状态, 在不造成开关资源冲突并保证安全的前提下, 找到一条可行的信号通路, 将相应的被测对象端口和仪器端口连接起来, 完成实际信号的激励或测量。

驱动调用功能根据当前选择的仪器和开关, 调用相关

的驱动程序函数，完成仪器或开关的实际控制。

测试过程中产生的测试结果信息可由测试结果管理工具保存到数据库中，后期可在测试结果管理工具中对该数据库进行管理和维护。

诊断推理过程结合当前 TPS 的诊断模型以及测试结果，对可能存在的故障进行推断，给出潜在故障的权重。

测试或诊断完成后可由相应工具生成 PDF 格式的报告。

通过系统管理工具能够获取当前测试工作站中所有仪器信息的列表，可以对仪器是否可用状态进行设置。也可以启动仪器的自检，此时需要 TPS 运行引擎配合通过调用相应的仪器驱动函数来完成自检。

2 SCATS 的应用

为方便读者更好地理解 SCATS 软件的使用思路和方法，以如图 6 所示的简化的通用自动测试设备为例，对主要的操作步骤进行进一步说明。尽管是一个简化的设备，但其中包含的思想可以很容易地映射到复杂的真实设备。

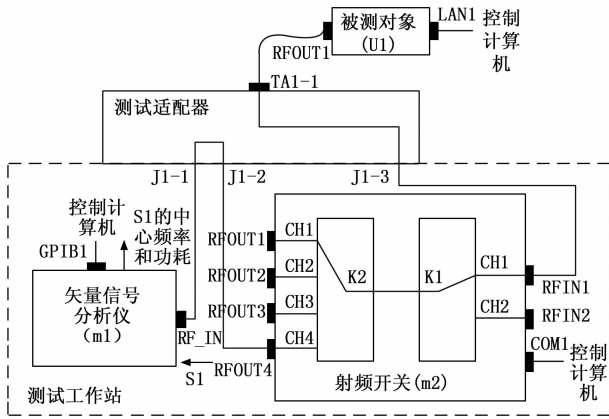


图 6 简化通用自动测试设备

测试配置信息包括：被测对象、测试线缆、测试适配器、测试工作站（包含射频开关和矢量信号分析仪）。被测对象通过测试线缆连接适配器前端，适配器后端连接测试工作站。

SCATS 软件的应用包括：系统建模，信号建模，模块建模，测试工作站建模，被测对象建模，适配器建模，测试线缆建模，系统集成，驱动开发、调试，TPS 开发、调试等几个方面，如图 7 所示。

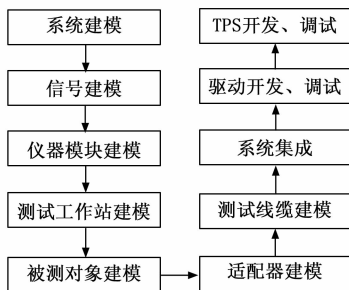


图 7 SCATS 在测试设备中的应用

1) 系统建模：

系统建模是根据相应的 ATMI 标准，对系统中的信号、仪器（射频开关和矢量信号分析仪）、测试工作站、适配器、测试线缆和被测对象（机载电子系统）等要素进行描述。

2) 信号建模：

经过梳理，上述示例中只有一个信号，即一个单载波射频信号 (S1)，由于该信号不是内置信号，因此需要对其进行扩展。

在 ATML 模型资源管理器中双击“ExtTSFLib”这个扩展信号库，然后在其中，

新增一个名为 SPECTRUM_TEST 类型的信号，如下图所示。其中包含若干属性，如中心频率，参考幅度，标记的峰值幅度和标记的峰值频率等。

3) 仪器建模：

示例测试工作站中包含两个仪器，一个频谱仪 (m1) 和一个射频开关 (m2)，其中频谱仪具备的能力是测量上述射频信号的幅度和频率。下面为这两个仪器构建模型。

在 ATML 模型资源管理器中选择“仪器模型”节点，右键单击，在弹出的上下菜单中选择“新建仪器模型”，然后在弹出的对话框中输入仪器的型号，假设为“SA_SIMPLE_1”，点击“新建按钮”后将在工作区中显示一个仪器模型的初始图形，需要注意的是，在编辑仪器能力的时候，能力的角色是“测量”，信号类型是上面定义的射频信号，被测属性是上述信号的标记幅度 (markampl) 和标记频率 (markfreq)。

同理，构建射频开关的仪器模型，其中需要注意的是在开关列表中对其中每个开关单元的定义，主要体现在其端口上，如一个单刀双掷的开关有一个公共端 (COM，只能位于端口表的第一项) 和两个连接端，这些信息至要体现在开关端口表中，此外还要在端口映射中，将各个内部开关单元的端口连接到外部接口上，也支持不同内部开关单元端口的互连，这样可以构建多级开关。

4) 测试工作站建模：

将测试设备所有的仪器模型集成为一个测试工作站，需要为所展仪器实例指定唯一的 ID 和地址（实际物理测试才需要），另一方面就是建立硬件测试平台中仪器端口和硬件测试平台对外接口之间的连接，这个工作在“端口连接”功能中进行。

将上述两个仪器模型集成为一个测试工作站。其中需要为这两个仪器实例指定唯一的 ID 和地址（实际物理测试才需要）。另一方面就是按照示例图中所示建立测试工作站中仪器端口之间和测试工作站接口之间的连接，这个工作在“端口连接”功能中进行。

5) 被测对象建模：

构建被测对象模型，依据被测机载电子系统的特点，为其构建连接器和端口即可。

6) 适配器建模：

构建适配器模型，为其构建后面板测试针脚并包含

适配器前后面板连线关系。

7) 测试线缆建模:

构建测试线缆模型,包含一个从适配器前面板到被测对象的连线。

8) 系统集成:

将上述 1)~7) 自动测试要素集成为一个通用自动测试设备,将各个模型的外部端口进行互连完成系统集成。

9) 驱动程序开发:

生成驱动程序框架的功能访问入口位于仪器建模工具中。驱动框架支持 Python 和 C 语言。SCATS 建议优先选择生成 Python 驱动,因为 Python 驱动程序的开发和调试效率明显高于 C 语言。只有当厂家只提供 C 语言驱动或 Python 语言不能处理时,再考虑使用 C 语言。

使用 VSCode 打开生成的驱动程序框架,在生成的函数里,编写示例中所示仪器频谱分析仪及射频开关的驱动。需要注意的是,除 TP 仿真不调用驱动,驱动仿真、模型仿真、物理测试均会调用驱动程序,但是驱动仿真只是调用驱动里对应的功能函数,不需要对物理仪器进行操作。因此在代码中仪器驱动初始化函数 `def SCATS_XXX_Instrument_Init` 里增加“SimDriver”的判断,录入 `simDriver` 处理方式。

频谱分析仪及射频开关驱动程序代码编写完成后,可以直接运行来调试。将频谱分析仪及射频开关连接上控制计算机后,运行编写好的驱动函数,为其传入给定的参数,观察仪器是否按预期响应。

10) TPS 开发和调试:

进行 TPS 的开发和调试。示例的测试需求就是控制被测对象发射一个射频信号,然后通过频谱仪来测量该信号的峰值幅度和频率。

TPS 开发过程如下:

1) 建立测量信号(信号的引用保存在局部变量 `ls1` 中),这是一个信号需求,需要的信号类型为 `SPECTRUM_TEST`(如上所述),预期的中心频率在局部变量 `cFreq` 中定义,值为 `1E8Hz`,然后信号属性 `centerfreq` 引用了该变量。

2) 连接测量信号,将 `ls1` 信号以单针(Series)的方式连接到被测对象的 X1-1 针脚。执行该步骤后,后端的测试服务会控制开关连接,以便形成测量的信号通路。

3) 通过调用一个外部的 Python 脚本,控制被测对象发出规定参数的射频信号。此时信号通路上就有实际信号在运行了。

4) 等待 1 s,使得信号达到稳定。

5) 读取测量值,此时会调用分配的频谱仪的驱动程序,完成上述信号的测量后获得返回值,这里同时测量两个指标,分别保存给两个测试结果。

6) 断开连接,即断开信号通路(有必要的话需要再之前加一个调用操作,控制被测对象停止输出射频信号)。

7) 重置测量信号,使得仪器恢复到初始状态。

TPS 开发完成后,可以通过设置断点,然后点击“启动

调试”工具按钮,在“模型仿真”模式下进行仿真调试。点击调试信息下的“停止/关闭调试”工具按钮,则当前调试过程停止并且调试信息面板关闭,表明支持停止调试功能。

由以上示例可知,SCATS 软件将 TPS 开发与测试工作站中的仪器进行隔离,真正实现了面向信号的测试目的。TPS 最终呈现为是一个 XML 文件的集合,对于其他基于 ATML 标准定义的 TPS 开发软件具有很好的可移植性和可扩展性,且纯中文代码界面,使得 TPS 代码的可读性强,测试逻辑清晰明了。

3 结束语

结合机载电子系统故障诊断的特点,本文介绍了一款基于 ATML 标准的面向信号的 TPS 开发软件—SCATS。其在机载电子系统通用自动测试设备研制过程中的应用包括系统建模、信号建模、模块建模、测试工作站建模、被测对象建模、适配器建模、测试线缆建模、系统集成、驱动开发及调试、TPS 开发及调试等。通过生成的 ATML 描述文档,实现被测对象信息描述,测试信号描述,测试路径描述,测试策略描述,故障诊断模型描述,测试结果记录、显示,故障诊断结果记录、显示等信息的交互,从而实现通用自动测试设备的通用化设计。

应用结果表明,SCATS 软件采用纯中文开发界面,测试程序集采用无代码开发,软件界面友好、可读性强,大大降低了测试程序集开发人员的技术门槛,内置信号模型丰富可满足现代机载电子系统测试需求,具有开发速度快、故障定位准确、通用性和可移植性好等特点。

参考文献:

- [1] 袁 钢. 故障树分析法在机载电子设备故障诊断中的应用研究 [J]. 电子测试, 2019 (22): 33-36.
- [2] 侯满义, 荆献勇, 范惠林, 等. 基于特征参数分布的机载电子设备故障诊断与预测 [J]. 数学的实践与认识, 2019, 42 (18): 69-75.
- [3] 路亚峰, 张 涛, 张 贤, 等. 机载电子设备故障诊断专家系统的设计与实现 [J]. 电子测量技术, 2018, 33 (1): 118-120.
- [4] 葛名立. 机载设备测试性建模与应用 [J]. 计算机测量与控制, 2022, 30 (12): 36-41.
- [5] 任献彬, 姜志森. ATS 故障树法故障诊断功能的实现 [J]. 测控技术, 2019, 32 (8): 141-144.
- [6] 吴国宝, 宋 帆, 陈德军, 等. 某型直升机机载电子系统原位检测设计与实现 [J]. 计算机测量与控制, 2019, 27 (4): 51-54, 110.
- [7] 王 娟, 潘 卫, 陈 奎. 航电设备测试性与外部自动测试系统一体化平台设计 [J]. 航空计算技术, 2023, 53 (1): 87-90, 95.
- [8] 黄雪明, 潘增寿, 任登娟. ATML 标准在自动测试系统中的应用 [J]. 现代信息科技, 2019, 15 (3): 7-9.
- [9] 钱 锋, 孟晨. 面向信号的 ATS 软件体系结构研究 [J]. 仪表技术, 2009 (3): 56-58.

(下转第 352 页)