基于ATML/STD标准的通用导弹测试软件平台开发

刘 昕
北京泛华恒兴科技有限公司

摘 要：通用软件平台，是测试系统实现通用化的核心。对于导弹测试，测试软件开发运行平台的标准化是军事装备测试系统接口兼容、测试程序及测试数据共享的保障。基于ATML/STD标准的通用导弹测试软件平台，保证了系统体系结构的开放性、运行稳定性、升级维护的可维护性和可扩展性。本系统采用成熟的底层测试执行引擎，提供便捷的测试流程管理和高效的执行功能，保障了测试程序集执行的安全稳定；针对导弹测试的特点，按照ATML标准体系结构对TPS研发和测试等角色进行分工；并根据ATML模式定义TPS开发过程的数据类型和内容，降低了测试过程对人员的素质要求，最大程度的提高了自动测试系统信息及TPS内容信息的开放性。实现ATS信息共享、交换、互操作和TPS跨平台重载移植。

关键词：ATML；STD；自动测试系统

1 引 言

长期以来，装备测试系统需求逐渐增长，测试行业出现了大量各种软件格式的测试软件，这些测试软件中，一部分由于专有设备的特性，可移植性和可维护性很低。另一部分通用测试软件中，部分通用软件虽然在特定的平台上具备一定的通用性，但由于测试程序描述格式、输入输出数据格式以及接口特性没有统一的标准和要求，在自动测试系统（ATS）规模不断扩大时，系统内不可避免的集成各类测试软件，导致了这些软件之间无法互相兼容，大量的测试数据和测试程序无法实现共享，直接导致了装备测试系统维护成本过高。

为解决上述问题，并满足ATS开放性、通用性的要求，在ATS的研发中，面向仪器的测试程序开发由于存在仪器互换性差，测试程序集（TPS）跨平台移植差的缺陷，逐步被面向信号的测试程序开发所取代，测试系统资源和测试执行也采用自动测试标记语言（ATML，automatic test markup language）规范进行严格定义，下一代自动测试系统的架构主体框架就是以ATML/STD为主体。

自动测试标记语言，即IEEE1671标准，是一种测试信息存储和传递的标准。它以XML技术为基础，在此基础上开发了测试领域适用的文件模式(XML Schema)，从而方便的组织和描述测试领域中出现的信息，标准化了测试领域信息传递的格式。引入自动测试领域的XML就称为ATML。

2 ATML体系结构

ATML规范包括ATML框架、组件及相关标准[1]。ATML的框架用3种完全不同的方式来定义，包括外部接口，内部模式，以及服务。所有的外部接口和内部模式反映了ATS的硬件平台，包括UUT及其适配器。尽管服务没有在ATML标准中明确定义，但它希望提供产生、销毁、操作信息的服务。外部接口反映了两个或更多不同组件间交互的信息。

作为测试系统内测试资源定义的规范簇，ATML规范必须为测试程序集生存期内所需要的各项信息提供

支持。包括测试描述、仪器描述、被测对象描述、测试配置描述、测试适配器信息、测试站信息在内的6个子规范。除此之外， ATML规范还包含了对其他规范的引用，包括诊断信息(IEEE1232，AI-ESTATE)、测试结果描述 (IEEE1636，SIMICA)、可测性与可诊断性分析(IEEE1522，testability and diagnosability characteristics and metrics)以及信号及测试定义(IEEE1641，signal and test definition)。ATML规范，以可扩展标记语言XML为数据交换载体，充分继承了XML的优势，通过提供模式检查文档进行开放式的描述方式定义，并对文档的节点进行规范性检查。除每个子规范对应的模式检查文档以外还存在通用性的文档规范性模式检查文档，以及所引用规范的模式检查文档。ATML规范中每个子规范都需要有ATML总规范对应模式检查文档，以及各子规范对应的模式检查文档。所涉及的模式检查文档随规范一并发布。这6个子域将最大化节省开发成本，ATE或TPS变动后，只需替换相应组件即可[2]。

ATML能力主要用于软件智能的判断一个给定的系统能否执行一个给定的测试任务。实现目标需要下列数据：测试需求，仪器能力，测试系统拓扑结构，任何因素引入的测试错误或误差。所以ATML能力信息是分布式的，分布在几个不同的ATML子标准族中，包括：ATML测试描述、ATML仪器描述、ATML仪器实例描述、ATML测试站和测试适配器描述[3]。ATML线表主要反映了测试系统中硬件端口间的互联信息，为测试需求映射到执行仪器提供支持。

ATML标准的定义和输出文件是可以纵向分层为支撑XSD、标准XSD、输出文件，如图1所示。

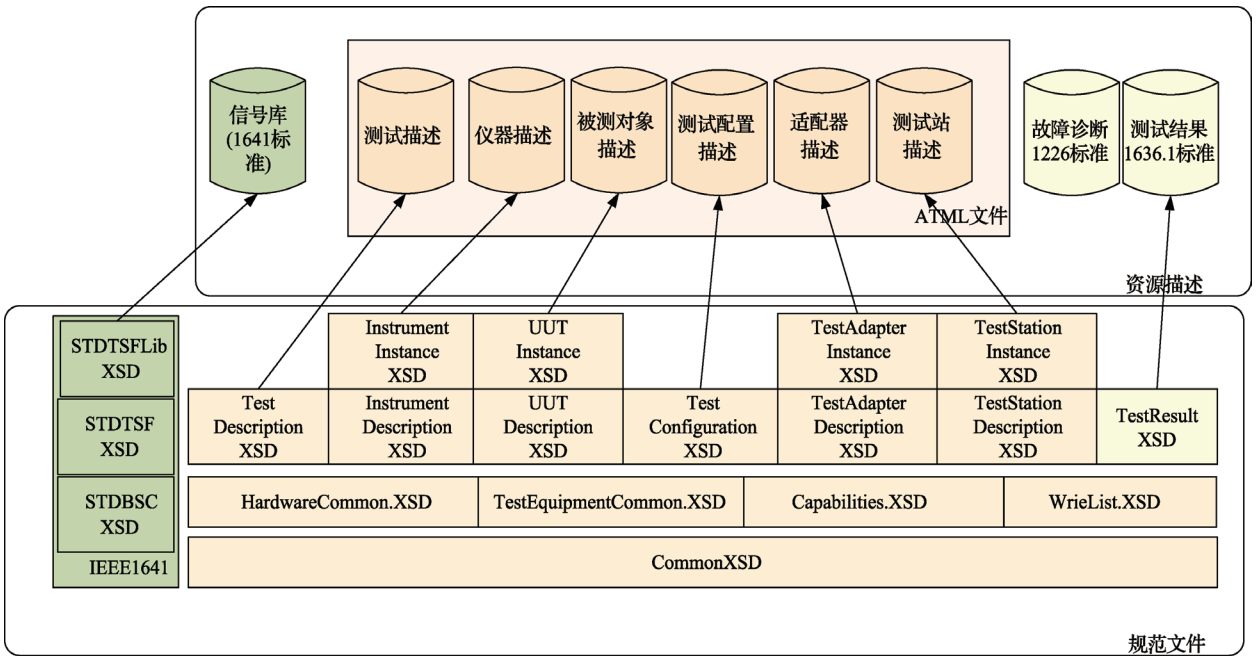

图1 ATML框架调用结构

针对ATS的分解，对ATS操作的工程师，按照职责和需求区分为以下几种类型：TPS开发者、ATE开发者、测试执行、检修维护。这几种类型的人对ATS的操作有特定的范围。操作的区域相互覆盖。如图2所示为各类人员的主要职责。

为不同类人员分配不同的系统操作权限，是不同职责工程师共同开发ATS的必要条件。所以首先在系统的顶层设计中，根据权限设计用例。如图2所示。

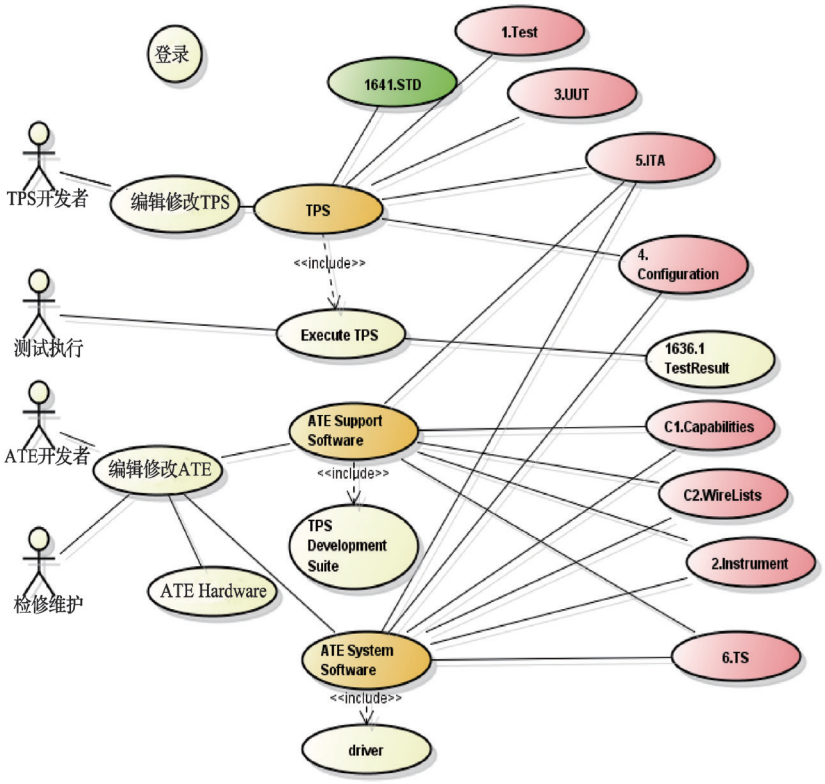

图2 ATS TPS操作权限用例

注：粉色部分为1671规范，绿色部分为1641规范，黄色部分为ATS组成。

测试人员通过登录获取为该用户名分配的操作权限，操作权限分为如下4类：

（1）TPS开发者：该类型人员主要进行TPS的编辑和修改。在修改TPS的过程中，涉及到对ATML中文件的修改：1671.1测试描述、1671.3UUT描述、1671.5ITA即适配器描述、1671.4测试配置描述，以及对信号库的读写、扩充、编辑、删除操作。

通过在ATML环境中，对与TPS硬件信息相关项配置：UUT描述、适配器描述（与UUT连接关系和适配器内部连接关系）、系统配置信息；与TPS执行相关的测试程序;测试描述和信号编辑。

（2）ATE开发者：该类型人员主要进行ATE的编辑和修改。该类人员的职责主要是：①配置系统硬件，②配置系统软件，与本系统相关的驱动等，③ATE支持软件，按照规范的划分，TPS的开发软件也属于这部分，例如本系统的编译平台。

ATE支持软件操作的ATML文件有：1671.5适配器描述，1671支持文件capabilities（引用的信号定义），1671支持文件WireLists接线表（各个XML元素间的硬件接口连接）。

（3）测试执行：操作TPS执行软件，不涉及对文件的修改，只生成结果输出。

（4）检修维护：该类型人员主要是进行测试系统维护和标检、设备更换等操作。在仪器更换和标检的同时，要负责更新仪器、适配器、连线、能力描述内的信息。不涉及更改的部分不需修改。

3 软件平台架构

系统软件平台在设计上分为5个层级，从上至下分别是数据管理层、应用层、信号层、驱动层、物理设备层，如图3所示。

数据管理层通过X-Designer平台的Data on Demand来实现对测试数据的管理。

应用层基于Test On Demand（简称TOM）进行开发管理，实现用户界面的操作、硬件资源的配置，以及通过TOM引擎来实现TPS的编辑及执行功能。

信号驱动层在X-Designer平台中集成了ATML/STD标准和ATLAS规范。

驱动层主要是对驱动的管理，在系统中用到的硬件驱动主要以仪器、数据采集设备及专用仪器3类为主，通过X-Advisor来进行管理，仪器类驱动通过IVI标准来统一，统

一的类别包括IVI标准规定的十二类仪器：分别为示波器类(Scope)、数字多用表类(Dmm)、函数发生器/任意波形发生器类(FGen)、直流电源类(DCPwr)、开关类(Swtch)、功率计类(Pwrmeter)、频谱分析仪类(SpecAn)、射频信号发生器类(RfSigGen)、计数器类（Counter)、下变频器类(Downconverter)、上变频器类(Upconverter)、数字化仪类(Digitizer)[4]；数据采集设备(DAQ设备)驱动可以定制统一的驱动接口，目前系统中所采用的数据采集设备驱动接口具有一致性；专用仪器类也可以通过定制统一的驱动接口，便于后期的升级及维护。

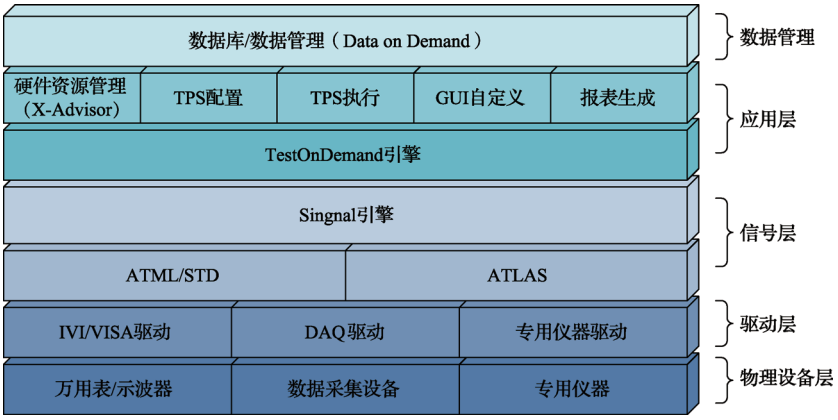

图3 软件平台分层

3.1 驱动层(X-Advisor)

硬件资源的接口标准化是测试程序集(TPS)与硬件无关化的首要要求——包括硬件接口和驱动接口。X-Designer平台在软件架构的纵向分层上，通过X-Advisor软件管理系统中的仪器及驱动。

X-Advisor软件支持系统自带驱动和用户自导驱动两种模式，前一种适合于IVI类仪器驱动及泛华、NI等常见厂商驱动；后一种通过灵活的“模块向导”功能，将自定义驱动导入系统。

在驱动层先后出现了SCPI、VISA、IVI等标准。这些规范伴随自动测试系统发展以来技术较为成熟，各大仪器厂商提供的仪器驱动均符合上述标准。采用这些标准化的驱动，可以在仪器驱动层完成对仪器物理接口的统一控制。所以，X-Advisor为了使工程师更好地搭建测试系统，在软件中内置了IVI驱动模块及SCPI/VISA指令收发模块，便于进行仪器控制和测试。

3.2 信号层(ATML/ATLAS)

信号层不仅仅是STD规范规定的信号及信号组成,还包括了引用信号的测试描述和仪器描述等ATML规范规定的要素，以及基于ATLAS的测试需求描述。

在信号层中，分为ATML和ATLAS两大主体。ATLAS包含了ATLAS解析引擎和运行引擎，在此不作详述；ATML包括依赖的STD规范以及IEEE1636、1232等规范。

在ATML规范的架构实现中，分层为基础数据类型定义部分、顶层协议部分。

（1）基础数据类型定义

包括Common、Hardware-Common、TestEquipment三种。这三种定义模式的特点是不直接输出对应的XML文件，而是通过引用，以节点的形式输出并包含在顶层协议输出的XML中。这三者间，Common又是最为基础的数据类型定义，包含一些数据类型、操作运算符等基础。

（2）顶层协议部分

这部分除了包括与ATML子协议同名的XSD文件，还包含其中硬件部分的实例化文件（Instance），Capabilities（在仪器实例等文件中将引用该文件）、WrieLists的XSD文件，引用规范对应的XSD文件。

3.3 应用层(Test On Demand)

应用层的执行模式包括并行顺序执行。在并行测试模式下，各个UUT是相互独立的——与引擎间的命令、数据及运行状态相互独立。在UUT运行模型中，包括对界面响应和硬件触发的响应。UUT的测试序列先后经历测试开始、运行、测试结束的运行状态。如图4所示。

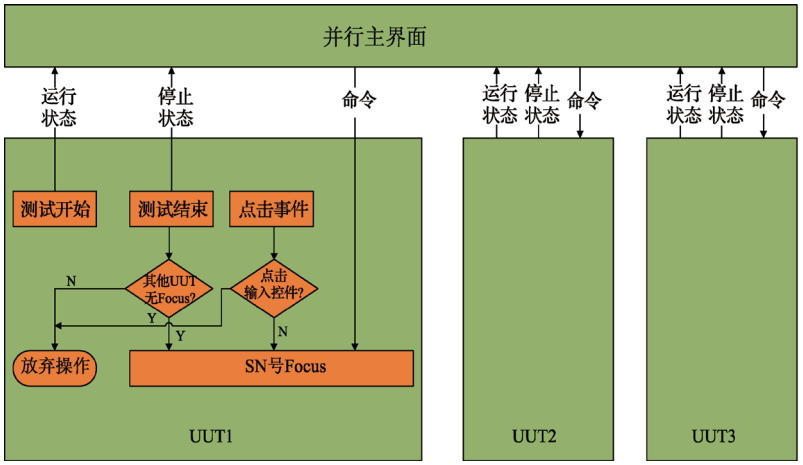

图4 TOM对被测对象的控制执行模型

3.4 数据管理层(Data On Demand)

下一代自动测试系统，海量数据处理将成为制约其发展的一个瓶颈。现有测试数据管理系统难以满足用户对海量测试数据的综合利用，数据挖掘。

暴露的现状是：实验数据分散独立，缺乏完整的管理体系；试验信息缺乏完整性，试验过程及测试数据之间缺乏集成；试验数据缺乏标准化管理，不能适应多种不同型号测试任务的需求；大通道数/解码量时：数据存储性能差；大数据量条件下，数据查询效率低下，数据曲线显示缓慢；对MATLAB等常用软件支持差，导致存储后的数据利用率低。

X-Designer平台针对这些现状分析，通过一系列优化技术：通过对测试数据的全生命周期管理，建立以对象关系型数据库为核心的测试数据管理系统等优化技术，推出Data On Demand数据库管理软件，对X-Designer平台上的测试系统产生的测试数据、结果和文件进行数据库上传、下载，备份恢复等操作。并对测试结果通过数据挖掘、大数据、云计算等技术进行统计分析和数据处理[5]。测试结果统计如图5所示。

图5 测试结果统计（GRR统计）

4 软件设计

4.1 软件设计框架

整个平台的设计，按照面向对象的方式，遵循UML设计规范，从用例、类图、活动图、序列图等方式对平台进行由浅入深的阶段性设计。

按纵向划分，类的层级包括界面类、ATML类及其继承类等，如图6所示。

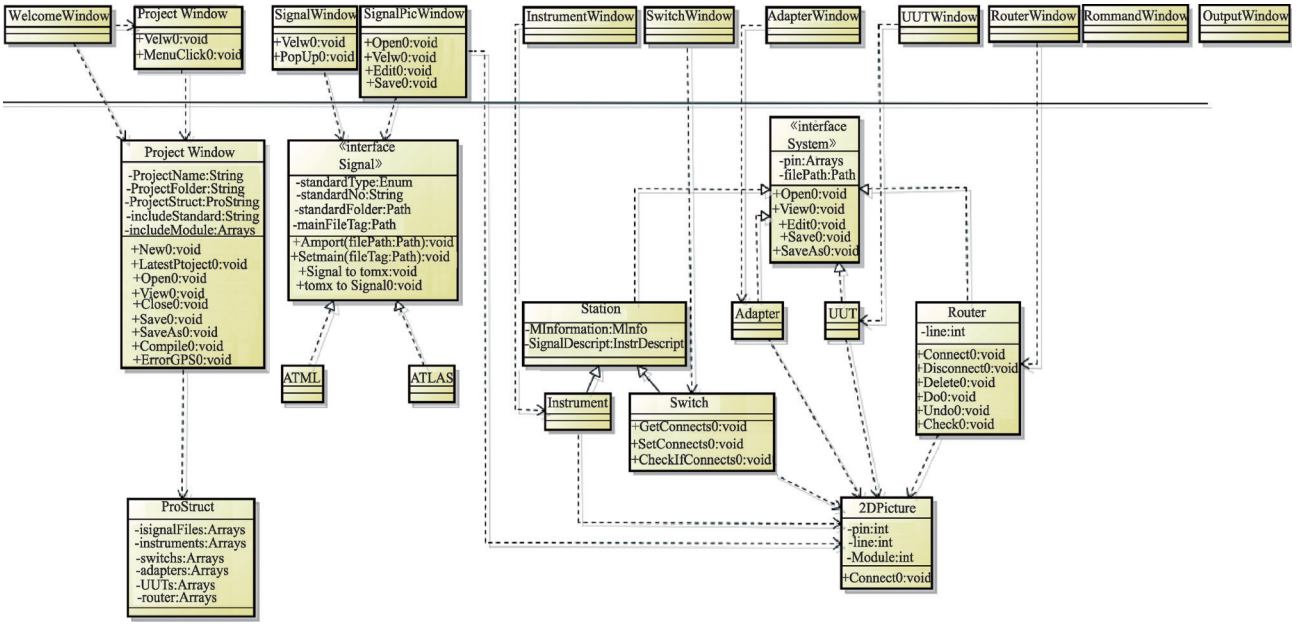

图6 类的层级示意

4.2 信号映射关键点

在该软件平台中，信号描述最终要对应到仪器的驱动。在当前仪器厂商没有提供信号驱动的情况下，只能通过接口映射，将信号及其参数，映射到仪器驱动的具体参数上。这个映射过程包括：对ATML资源进行虚拟资源映射、通道路由和驱动的信号的驱动映射。

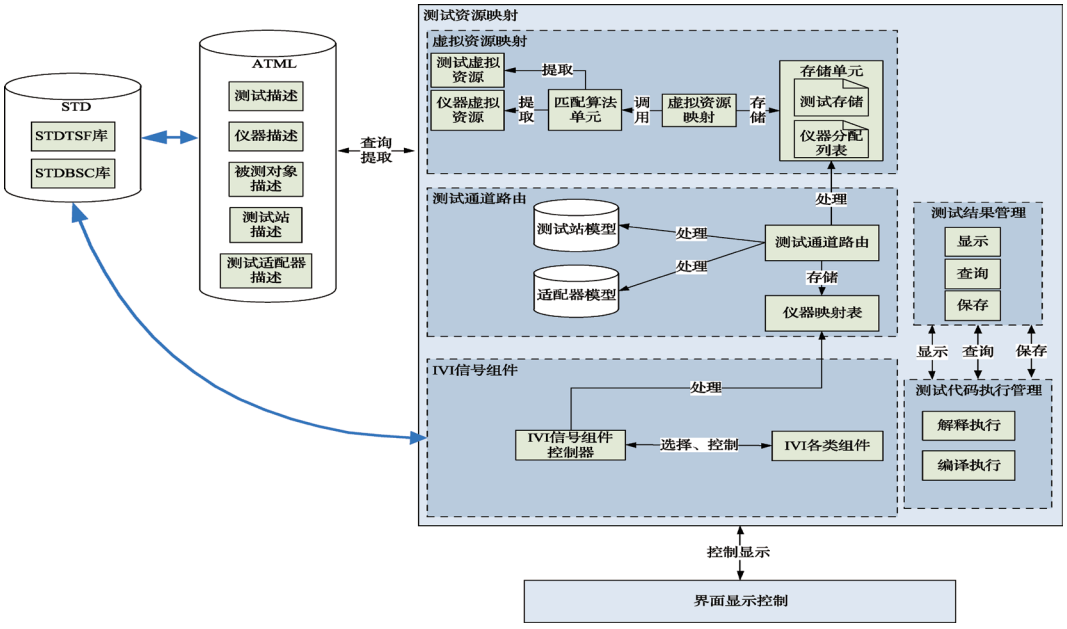

图7 ATML信号映射和驱动结构图

5 实验验证

5.1 开发过程

(1)建立信号模型

按照信号需求，在X-Designer平台的信号描述界面中进行参数的配置和仿真。当信号属于基本信号，即STD规范中的BSC和TSF库中已经有了对应的定义时，可在测试描述界面中直接选择和配置；若信号属于复杂类信号，需要对基本信号进行一定的组合才可形成的信号，需要在图8所示的信号描述界面中进行详细的组合关系描述[6]。为了验证信号的正确性，可通过仿真界面，进行信号的仿真，并根据信号的组成关系，将系统支持的硬件予以提示。当该信号不能被现有硬件库支持，将提醒工程师这个信号可能在该测试系统中存在无法实现的风险。

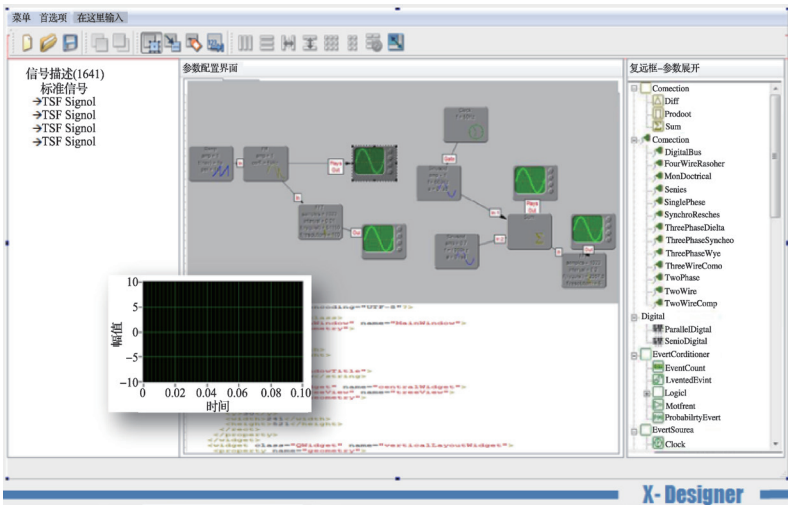

图8 信号编辑和仿真

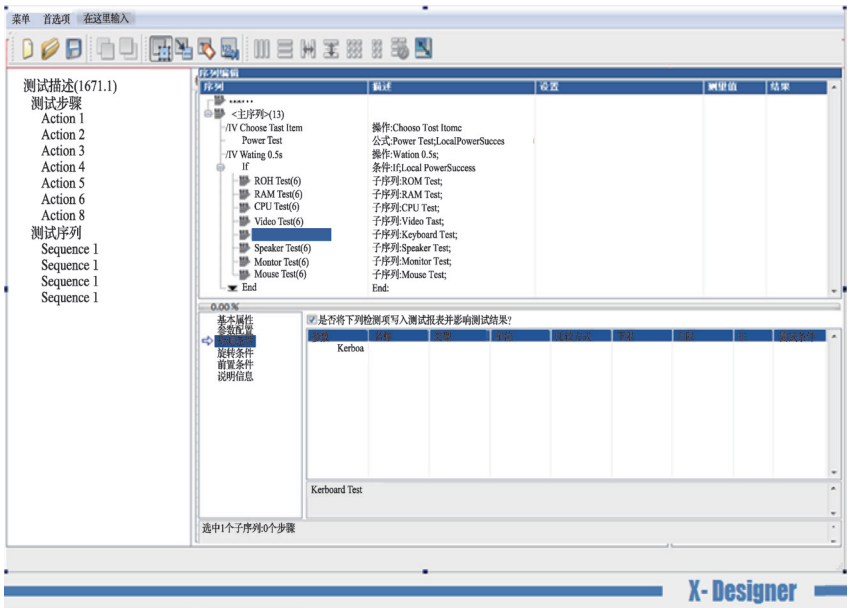

图9 测试序列编辑界面

(2)创建测试序列

测试序列由测试项组成，测试项又包含了若干步骤。在测试序列的创建过程中，包含了对测试序列编辑界面、测试项编辑界面、测试步骤编辑界面的界面操作。对于测试序列、测试项、测试步骤，都有一定的向导操作，引导工程师快速创建一个涵盖了基本测试信息的测试序列。通过导航栏可以快速回到主序列，查看测试序列的全貌，如图9所示。

(3)导入硬件模型

仪器、测试站、适配器、UUT等模型，当成为一定规范进行大批量生产时，通过导入硬件模型的功能，就能快速进行模型信息的配置，如仪器描述。

(4)导入接线表

连接关系以EXCEL格式的电子表格存储，导入系统中，可在界面上进行显示和修正。

5.2 解析运行

在ATML开发完成后，解析运行的过程包括：1)对测试序列的验证工作。2)对仪器的优化选择。3)对执行过程中测试数据的自定义界面显示。4)测试流程监控。5)测试结果显示。下面就比较重要的几个方面着重介绍。

(1)自定义界面设置

测试数据有总线数据、数值、布尔、数组等数据类型，但表现形式不止这几种。如果把一种数据类型局限于一种表现形式，将使界面不具备具体测量量的物理含义，表达性差。

传统的测试程序，可以为具体

一类被测件设计专门的测试界面，这种设计固然能满足测试便捷性，但由于与测试需求绑定，就大大降低了测试程序的可移植、可维护。

在本设计中，吸取了传统测试程序的缺陷，使用Test On Demand的自定义界面的功能，使之与ATML测试序列相分离，极大地提高了ATML测试序列的复用性[7]。自定义界面包含了测试流程控制与引擎的关联、测试数据与引擎的关联，设计完成的自定义界面如图10所示。

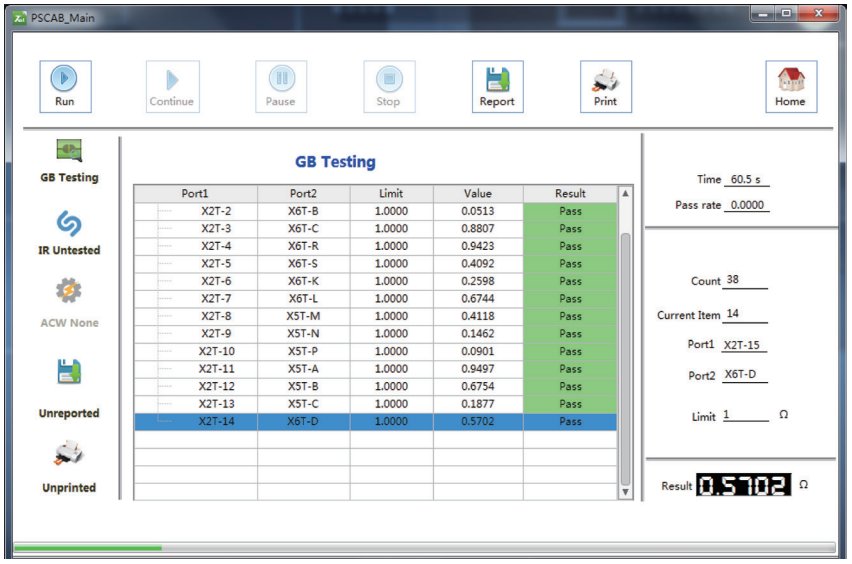

图10 自定义界面示例

(2)测试流程显示

测试流程通常包含了测试的跳转，跳转点处测试项的测试结果信息是很重要的。在该平台的运行过程中，将始终看到测试流程的跳转信息，点击某一步骤或测试项，该步骤或测试项的测试信息甚至中间结果，将立即显示出来，即使是已经测试结束的步骤，信息也将被保留，直至所有测试完成。

(3)测试结果显示

在测试结束后，测试结果将以一定的界面呈现方式进行显示。通过Data On Demand对测试结果进行分析，并与历史测试数据进行对比，这一统计数据将用于对UUT的设计改进。

6 结 论

基于ATML/STD的通用导弹测试软件平台，输入输出信号按照STD规范定义进行描述和组合，通过信号模型仿真工具验证信号的准确性，使用ATML规范对测试资源进行编辑配置，通过对序列流程的控制，在测试过程中，可以通过X-Designer平台实现对仪器的控制和数据的读取、显示，并在测试结束后对数据进行分析显示。通过这一平台，验证了ATML/STD标准在导弹测试中的实际应用价值。

参考文献

[1] 路辉.自动测试系统测试描述语言[M].机械工业出版社，2011.

[2] 刘昕. 面向信号的测试资源映射技术研究[J]. 哈尔滨：哈尔滨工业大学，2011:8-19.

[3] IEEE SCC20. IEEE Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML. 2010:1-10.

[4] IVI-COM Technology Reference. IVI Foundation. Version0.1. 2000.

[5] 北京中科泛华测控技术有限公司，Data On Demand v3.1 设计文档，2013.

[6] IEEE 1641. IEEE Standard for Signal and Test Definition[S]. IEEE,Piscataway,NJ . 2005.

[7] 北京中科泛华测控技术有限公司. Test On Demand v3.1 设计文档，2013.