doi: 10.3969/j.issn.1006-1576.2012.08.022

基于 XML 的通用数据接口测试工具

苏春梅，何剑伟，奚宏明，茅文浩
（中国人民解放军 63680 部队技术部，江苏 江阴 214431）

摘要：针对传统测控软件的接口测试的不足，设计一套 XML 文件配置的通用数据接口测试工具。分析测控软件数据接口特点，设计 XML 数据接口配置模板；描述数据接口测试的基本流程，提出采用“分层架构 + 构件模式”设计组装软件，脱离代码编写，通过文件配置组装各种正常、异常的接口数据，满足测控软件接口测试的各种需求；并给出 XML 的编程实现和需求变化的处理方法。结果表明：该设计能提高软件测试的自动化水平和软件质量，满足测控软件接口的各种需要。

关键词：XML；XSD；数据接口测试；DOM；SAX
中图分类号：TJ06 **文献标志码**：A

General Data Interface Testing Tool Based on XML

Su Chunmei, He Jianwei, Xi Hongming, Mao Wenhao
(*Dept. of Test Technology*, *No.* 63680 *Unit of PLA*, *Jiangyin* 214431, *China*)

Abstract: In this paper, the lack of traditional tracking and control software interface testing, design a set of extensible markup language (XML) files to configure general data interface testing tool. First analysis tracking and control software data interface characteristics, the design of XML data interface configuration template; And then describe the basic flow of data interface testing, and proposed the ‘layered architecture + component model’ design assembly software, from the coding, file configuration to assemble a variety of normal and abnormal interface data to meet the needs of a variety of tracking and control software interface testing; Finally, given the approach of XML programming and changes in demand. The results show that the design can improve the level of automation of software testing and software quality, to meet the various needs of the tracking and control software interface.

Key words: XML; XSD; data interface testing; DOM; SAX

0 引言

软件数据接口测试是软件测试中非常普遍而重要的测试类型，是软件质量保证的重要手段。数据接口测试属于灰盒测试，是功能测试的一种，是介于黑白盒之间的测试技术。数据接口测试要求测试工程师了解软件的设计及接口的定义，编写相应的测试代码进行测试，提高测试的效率。对于测控软件这类基于数据驱动的实时软件系统来说，数据接口是其系统的关键部分。为了保证这些接口的功能正确性和有效性，可进行接口测试，通过模拟产生各种接口数据，测试其接口的正确性。

以往对测控软件的接口测试，均是采取修改软件代码的方式产生各种测试数据，注入被测系统进行测试，测试准备时间长、测试效率低，不能适应高密度测控任务准备的要求。同时，由于当前不同型号任务测控软件交互的数据接口均不相同，种类繁多，且结构复杂，对于上述复杂多变的交互接口科学高效地进行测试变得十分复杂和紧迫。因此，笔者通过研究测控软件数据接口特点和测试难点，提出设计一套基于 XML 文件配置的通用数据接口测试工具(以下简称工具软件)，可满足测控软件接口测试的各种需求，提高软件测试的自动化水平和软件质量。

1 XML

XML 是一种结构化描述语言，能有效地表示各种数据，为信息的查询、交换和计算提供新的载体。XML 具有以下优点：1) XML 是纯文本，可阅读的，语法简单，易于掌握和使用；2) 采用树状结构，层次清晰，文档内容的增加、删除及修改简单易行；3) XML 格式的配置文件具有开发性和通用性；4) XML 文档具有可验证的特性[1]。

XML 文档的验证由 XML 模式来完成。XML 模式用来定义和描述对应的 XML 文档的结构、内容和语义。XML 模式与 XML 文档的关系类似于数

收稿日期：2012-03-08；**修回日期**：2012-04-16
基金项目：军队专项建设经费资助项目“实战软件接口测试工具设计”(2011JS316)
作者简介：苏春梅(1979—)，女，宁夏人，硕士，工程师，从事软件开发及评测技术研究。

据库中数据表结构和数据表的关系。利用 XML 模式，可以验证对应 XML 文档中数据在结构上的正确性和标记使用的规范性，这样，在数据形成阶段，而不是数据使用阶段就能对数据的有效性和正确性进行部分检验，提高了数据的可靠性和可用性，同时减少编写程序代码验证的繁琐。当前使用的 XML 模式有 XML Schema 和 DTD，笔者选择使用 XML Schema，原因有：1) XML Schema 本身就是 XML 文档，实现了 XML 文档与其描述方式的一致性，解析起来十分方便，而 DTD 则有自己特殊的语法，解析困难；2) XML Schema 比 DTD 拥有更多的内置数据类型，可以充分满足文档的可理解性和数据交换的需要；3) XML Schema 将逐步取代 DTD，成为 XML 体系中正式的类型语言[2-3]。

在工具软件的数据接口设计中，由于数据接口层次复杂，对于不同的被测系统数据接口变化较大，存在系统默认的数据格式难以表述的情况。为提高软件的通用性和可扩展性，采用 XML 与 XML 模式文档 XSD 相结合来定义描述并验证接口数据，同时基于 XSD 来定制界面配置和显示交互。

2　通用数据接口测试工具设计

通用数据接口测试工具软件主要功能是适应各型号、各平台航天器测控软件的数据接口测试需求，通过 XML 文件配置、组装各种接口数据和通信模式，自定义各种正常、异常的测试数据，注入被测软件，对其相关指标进行充分测试，以较大程度提高软件测试效率和测试的自动化水平。

2.1　基于 XML 的接口配置模板设计

工具软件设计的关键是自定义接口数据的描述定制方法。通过对历次型号任务测控软件数据接口的特点梳理，总结主要有以下几点：

1) 遵循总联文件规定接口信息交互传输协议符合 TCP/IP 协议模型分层结构。对于应用软件来说，主要关注应用层和传输层功能，如图 1 所示。应用层有应用头和应用数据的组合作为传输层的数据域，传输层的传输头为 TCP 或 UDP 协议，对于 UDP 协议又细分为单播、组播和广播，不同协议传输方法不同。

图 1　应用层和传输层功能

2) 不同型号任务软件接口要求不同，应用头、应用数据和传输协议存在多种组合模式。某些接口数据要求应用头和应用数据部分字段赋值为给定值，且不同赋值对应不同处理方法，给接口测试验证带来一定难度。

3) 高密度任务同时多发任务测控软件并行开发维护，要求测试人员需要并行完成同时多个型号任务测控软件的测试，时间极为紧迫。

如何充分高效测试软件数据接口的容错性和可靠性，通过编写和修改代码的方式进行测试显然不太现实。因此，需要研究脱离代码的快速接口数据定制方法，设计接口数据定制模板，基于软件框架通过文件配置和接口数据组装实现自动化测试，缩短测试准备时间，提高测试效率。

接口数据定制模板的结构设计是通过分析数据接口的特点，将可变参数提取出来，形成层次化的结构。接口数据定制模板由以下基本元素组成：

1) 测试描述。

① 设计方法：无效类、有效类、边界值等；② 测试环境描述等。

2) 网络传输方式。

① 网发类型：TCP、UDP(广播、单播、组播和指定源组播)等；② 网发目的地址；③ 网发端口；④ 网发主机地址；⑤ 网收类型：TCP、UDP(广播、单播、组播、指定源组播)等；⑥ 网收地址；⑦ 网收端口；⑧ 网收源地址。

3) 数据头结构。

① 数据头编号；② 数据头标识：对数据头进行的唯一标识；③ 数据头描述；④ 数据头参数定义：参数标识、参数类型、参数处理方法、参数赋值、参数取值范围、参数描述等。

4) 数据结构描述。

① 数据结构编号；② 数据结构标识：对数据结构进行的唯一标识；③ 数据结构描述；④ 数据结构参数定义：参数标识、参数类型、参数处理方法、参数赋值、参数取值范围和参数描述等。

从以上接口数据定制模板内容可以看出：描述的结构具有层次化的特点，逐级细化，这正符合了 XML 技术对数据描述的特点。因此，将 XML 引入工具软件设计，使用 XSD[4-5]文件描述接口数据的结构和数据类型，使用 XML 文件保存接口配置数据，不仅能够有效地对接口信息进行充分表述和验证，而且能够方便地根据接口的变化改变配置，脱离代码编写，极大地提高了软件系统的健壮性和可复用性。在数据接口测试准备中，经常出现直接基

于历史 XML 配置文件部分修改复用，虽然可节省时间，但文件的正确性和有效性难以保证，通过引入 XSD 文件可以同步检查验证 XML 文件的有效性，提高了测试可靠性和效率。

基于测试的接口数据组合模式为“通信＋数据头＋接口数据”，不同的测试要求不同的通信方式、数据头和有效数据的组合。图 2 所示 define_script.xsd 文件描述了基于模板的接口数据结构和类型定义，图 2 中可以看出该文件为树形等级结构，根节点只有一个命名为<Script>，一级子节点有 3 个为：<NetSet>、<DataHead> 和 <DataStruct>，一级子节点下有二级子节点和属性。图 2 中 define_script.xml 文件为基于 XSD 的具体接口数据定义。

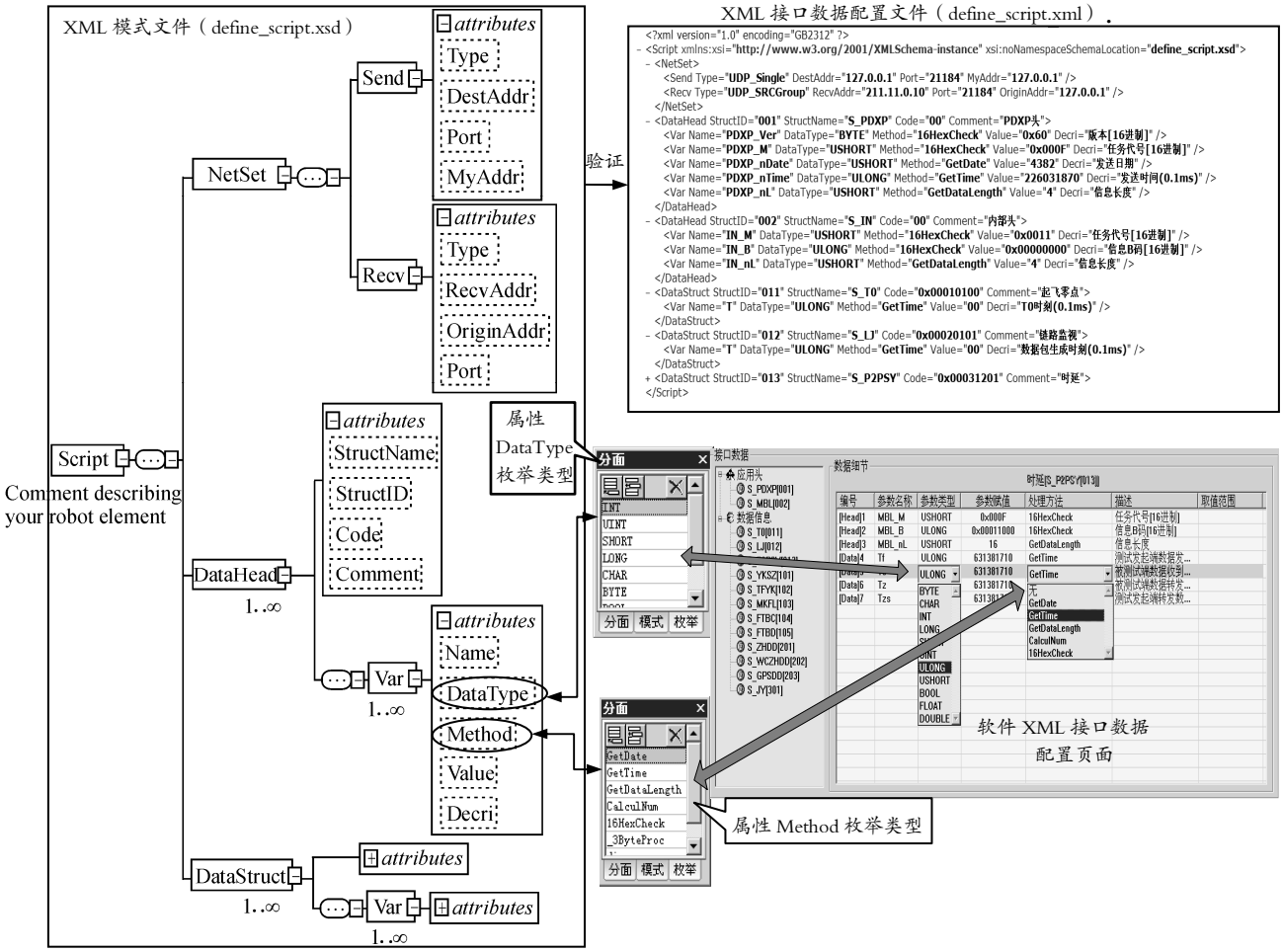

图 2　基于模板的 XML 配置文件、XSD 模式文件与界面配置同步示例

2.2　数据接口测试流程

基于工具软件进行数据接口测试的基本流程如图 3 所示。

1) 数据接口测试需求分析和测试策略定义。根据被测试软件需求、接口信息交换规定等或软件代码梳理出被测软件的接口测试需求并定义接口测试策略。

2) 接口测试用例定义和审核。根据测试需求和所定义的测试策略，定义接口测试用例，通过评审或专家审查等手段审核测试用例，并进行修改完善。

3) 接口测试数据准备。基于工具软件并依据接口测试用例，并逐一配置制作接口测试配置文件并保存。

4) 执行测试用例。搭建测试环境，基于工具软件加载 XML 接口配置文件，逐用例逐步骤地执行测试，通过界面操作向被测软件注入接口测试数据，接收被测软件发送和接收应答等信息，实时显示并记录。在执行过程中，认真观察并详实地记录测试过程、测试结果和发现的问题，形成测试记录。

5) 测试总结。测试人员对所有测试结果进行分析总结，对于测试不充分之处基于上述流程进行补充测试，对于发现的问题，待软件更动后基于上述流程进行回归测试。

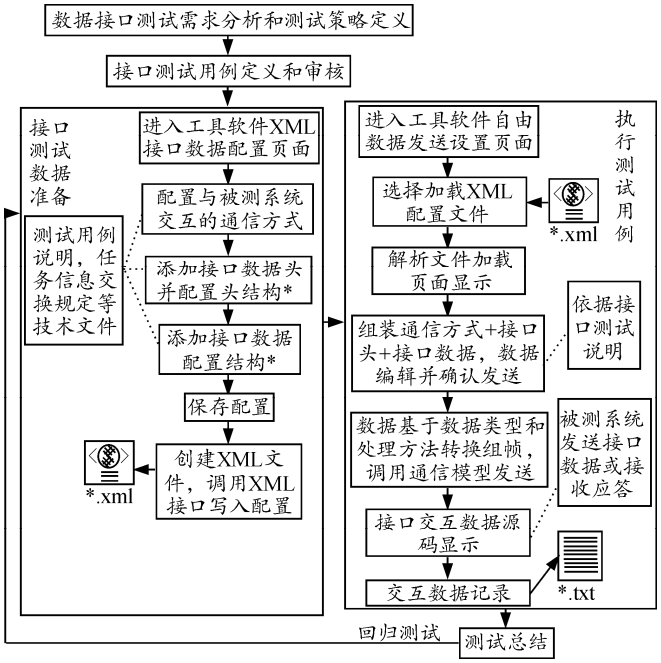

图 3　数据接口测试工作流程

接口测试时通过配置接口数据注入实现对被测软件接口的灵活充分测试，以提高测试效率。其中接口数据配置有 2 种方法：1) 修改复用历史 XML 配置文件，定义正常的或异常的接口数据，通过软件操作注入被测软件；2) 基于软件界面实时定义或修改产生不同要求的接口数据注入被测软件。2 种配置方法各有利弊，对于测试用例明确的接口测试，事先配置接口测试文件的方式，测试效率更高；也经常存在测试用例实时补充的情况，这样，基于界面实时修改定义接口数据的方式更方便。

2.3　软件体系结构

该软件作为被测软件数据接口测试驱动的测试工具，软件自身必须功能正确，可靠性高。为了保证操作及使用的友好性和软件功能的通用性和可扩展性，满足被测软件各种数据接口测试需求，同时适应软件功能基于测试频繁变动和扩展的需要，采用“分层架构＋构件模式”设计组装系统，提高软件的通用性和可靠性，其软件的体系结构如图 4。

采用“分层架构＋构件模式”，即将软件系统分离为多个具有清晰、内聚职责的模块或构件，采用基于表示层、业务层和接口层的 3 层体系架构，把系统大粒度的逻辑结构组织到不同的层中。每一层都有独立和相关的职责，使得较低的层作为低级和通用的服务，较高的层更多的为特定应用，从较高的层到较低的层进行协作和耦合，避免从底层到高层的耦合。每一层都依据其下一层来构建，同时为上一层提供实现基础；每一层内本着高内聚低耦合的原则又细分为若干构件或模块，通过各层构件不同的封装和组合，实现系统的功能扩展。

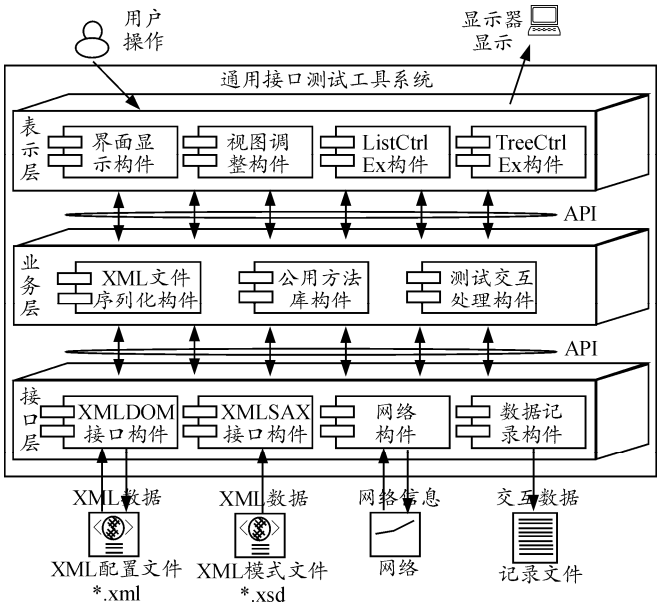

图 4　软件体系架构

表示层主要处理用户请求，将用户指令翻译、传递给业务层；业务层根据其请求返回信息并显示给用户，为用户提供数据显示和人机交互界面，包含视图调整、ListCtrlEx 和 TreeCtrlEx 等构件。

业务层主要处理用户提交的信息，或者将信息下传给接口层，或者调用接口层功能函数读取数据。应用业务规则，实现应用程序逻辑并完成应用程序运行处理，包括 XML 文件序列化、公用方法库和测试交互处理等构件服务。其中，公用方法库构件封装了通用的数据处理方法，对于接口数据结构参数定义中，涉及到日期、时间、变长结构和数组等复杂参数的赋值和组帧，通过调用方法库构件接口来实现。

接口层主要用于屏蔽软件系统与网络、文件等外界环境软硬件接口的差异性，为软件提供通用的接口服务，包含 XMLDOM 接口构件、XMLSAX 接口构件、数据记录构件和网络协议转换构件。XMLDOM 接口构件提供 XML 文件 DOM 解析和保存的相关服务；XMLSAX 接口构件提供 XML 文件 SAX 解析相关服务；数据记录构件提供数据记录定制等相关服务；网络协议转换构件提供网络数据接收发送服务、TCP/IP 协议服务、UDP 协议服务等。

2.4　测试需求变化的通用化处理

对于工具软件来说，基于数据接口测试需求的软件主要集中于数据处理方法和数据类型的更新。

由于该软件体系架构为基于构件的分层模式，高内聚低耦合，上述需求变化仅对应软件业务层公用方法库构件的更新、拔插、替换，以及表示层基于界面的 XSD 文件配置元素数据类型和处理方法等的添加、更新。XSD 文件描述了接口数据的结构元素框架和元素及属性类型定义，对于网络收发类型、接口数据参数数据类型、数据处理方法类型详细定义了枚举，数据处理方法类型与公用方法库构件 API 对应。由于笔者采用 XSD 文件初始化界面显示和配置元素，对于表示层基于上述需求的更新，无需修改代码，仅更新 XSD 文件即可实现同步，简化了开发和维护的工作量，缩短了测试准备时间，提高了测试效率。如图 2 所示，对于数据类型和数据处理方法的添加，通过 XML 编辑工具 XMLSpy 等对 XSD 文件进行操作，为属性 DataType 和 Method 枚举类型添加新的元素，工具软件启动时通过读取 XSD 文件即可实现界面配置元素同步。

3　基于 XML 的接口编程实现

为同时适应航天远洋测量船的特殊要求，系统采用 Visual C++进行编程开发。利用 Visual C++良好的编程环境和 MFC 强大的界面编程功能，结合微软提供的 XMLCOM 库 MSXML 4.0 实现 XML 编程解析，以提高系统的开发效率和通用性。

XML 的编程实现涉及到 XML 和 XSD 文件的解析。由于 XSD 文件本身就是 XML 文档，所以解析方法与 XML 文件相同。通常使用的 XML 解析器有 2 类：1) DOM，文档对象模型，是基于树或基于对象的，使用内存保存对象结构，即 DOM 规范的核心是树模型。DOM 的优点是编程容易，其缺点是对内存的占用空间大。2) SAX，简单应用程序编程接口，是一种面向事件的 API 接口。它的核心是基于事件的串行处理机制，与 DOM 解析器相比，不需要在内存中建立和 XML 文件相对应的树形结构数据，占有的内存容量少。SAX 的缺点是编程困难，很难同时访问同一个文档中的多处不同数据[6]。

下面简要分析工具软件对 XML 和 XSD 2 个文件的解析要求。对于 XML 文件的访问存在并行实时处理和动态修改保存操作，编程需要建立对应文件树形结构，XML 的 DOM 访问接口具有随机访问 XML 文档中元素的功能，相比 SAX 的顺序模式，能够更好地支持数据接口编辑和生成；对于 XSD 文件的访问即基于界面显示配置需求读取部分元素和属性，接口对象创建即销毁，对象生命周期短，不需要在内存中建立文件树形结构，使用 SAX 解析器占用内存少，处理 XML 的效率更高。作为测试工具必须小而精，可靠性高，占用资源少，为提高程序可靠性和效率，采用 DOM 和 SAX 结合的方式优化 XML 编程开发[7]。

MSXML4.0 开发包同时支持 DOM 和 SAX。基于 MSXML DOM 和 SAX 接口进行二次开发封装，编写了为工具软件提供 DOM 和 SAX 解析通用支持的 XMLDOM 和 XMLSAX 接口构件，实现 XML 和 XSD 文件的解析和保存。

XML 文件的解析过程：当程序表示层应用调用 XML 文件序列化构件解析 XML 时，XML 文件序列化构件实例化 XMLDOM 接口构件，XMLDOM 调用底层的 COM 生成 IXMLDOMDocument 对象并实例化，IXMLDOMDocument 对象装载 XML 文件，XML 文件序列化构件顺次调用 XMLDOM 解析 API 操作 IXMLDOMDocument 对象，装载节点和属性等生成 std::vector 树结构，整个过程的处理效率是比较高的，具体流程如图 5 所示。

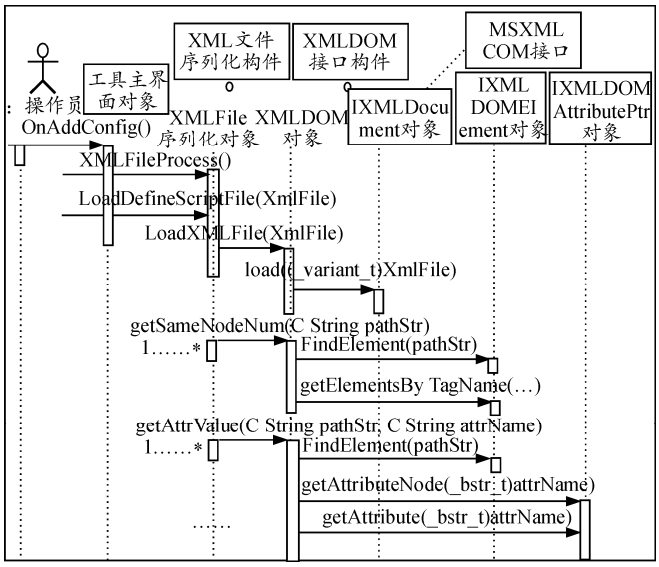

图 5　XML 文件 DOM 解析流程

XSD 文件的解析过程：当程序表示层应用调用 XML 文件序列化构件解析 XSD 时，XML 文件序列化构件实例化 XMLSAX 接口构件，XMLSAX 调用底层的 COM 生成 ISAXXMLReader 对象并实例化，实例化 ISAXContentHandler 对象并关联 ISAXXMLReader 对象，然后事件驱动的调用 ISAXContentHandler 对象 startDocument、startElement、endElement 和 endDocument 等 API 查询并串行解析 XSD 节点数据，加载显示，读取速度快、占用内存小，提高了 XSD 解析效率，达到优

化程序目的。

4　应用实例

不同型号任务对航天远洋测控应用软件系统的数据接口要求均不同。为适应高密度测控任务准备的要求，高质量按时完成测试任务，测试人员使用本工具软件辅助进行系统数据接口测试，根据测试要求和数据特点基于 2.2 节测试流程任意定制发送各种正常或异常的接口数据；充分验证了被测系统数据接口的正确性、容错性及可靠性，达到数据接口的充分测试；同时使测试准备及测试执行时间缩减了 50%。大大提高了测试效率，确保了软件质量，为航天测控任务取得成功奠定了基础。图 2 中右侧界面所示为工具软件界面示意。

5　结论

笔者以测控软件数据接口测试为切入点，提出设计一套基于 XML 文件配置的通用数据接口测试工具，满足测控软件接口测试的各种需求，提高软件测试的自动化水平和软件质量。该研究成果已应用在航天测量船海上测控领域系列软件测试中，证明是有效和科学的，同时也可为同类软件系统的开发应用提供一定的参考。

参考文献：

[1] 魏国志, 高晓光, 魏小丰. 基于 XML 的实时战场环境仿真[J]. 微型电脑应用, 2011, 27(3): 10-13.

[2] 杨大韬, 秦克, 闵绍荣. 基于 XML 的舰艇作战系统接口建模[J]. 船海工程, 2010, 39(5): 164-168.

[3] 胡艮胜, 包战, 颜美卿. 一种基于 XMLSchema 的仿真资源元数据管理方法[J]. 系统仿真学报, 2011, 23(1): 86-88.

[4] 曾春平, 王超, 张鹏. XML 编程从入门到精通[M]. 北京: 北京希望电子出版社, 2002: 48-74.

[5] 南松辉, 田佳, 张海波. 基于 UML 和 XMLSchema 的航天飞行数据建模[J]. 兵工自动化, 2008, 27(1): 46-48.

[6] Fabio Arciniegas. C++XML 高级编程指南[M]. 武磊, 刘拴强, 译. 北京: 北京希望电子出版社, 2002: 126-143.

[7] 文必龙, 关翔瑞, 周凯. 基于 VC 平台的 XML 解析技术分析[J]. 齐齐哈尔大学学报, 2007, 23(1): 35-38.

****

（上接第 71 页）

采用 XML 描述的电子对抗仿真想定是基于 XML Schema 定义的一套庞大的关于军事想定元素的标记集合，XML Schema 技术限定了这些标记的结构与成分，从而实现了军事想定数据格式和结构描述的标准化[3]。首先用 XML Schema 设计好整个电子对抗仿真想定的文件结构，通过 XML schema 中的语法和语义定义，可以实现对于电子对抗仿真想定场景的完整、准确描述，同时 XML 数据提取的方便性有利于实现仿真运行阶段对于想定脚本的装载。电子对抗仿真想定作战编成的格式如图 5。

下面以作战任务和作战行动的想定数据编辑为例，说明协同编辑的过程，作战任务列表和作战行动列表的 XML 描述如图 6 所示。

各个军兵种按照上述格式编辑完各自的想定数据后，提交至服务器；服务器通过 XML 解析将这些想定数据读出，将其写入总的电子对抗仿真想定 XML 文件中，如图 6 所示；其他想定要素与作战行动列表和作战任务列表一样，均经解析后写入到总的 XML 文件中，这样就得到了初步的电子对抗仿真想定 XML 文件。由于初始的想定文件数据可能存在着错误和冲突，故还需对想定数据进行审查并纠错才能得到最终的电子对抗仿真想定文件。作战任务和作战行动数据的融合过程如图 7 所示。

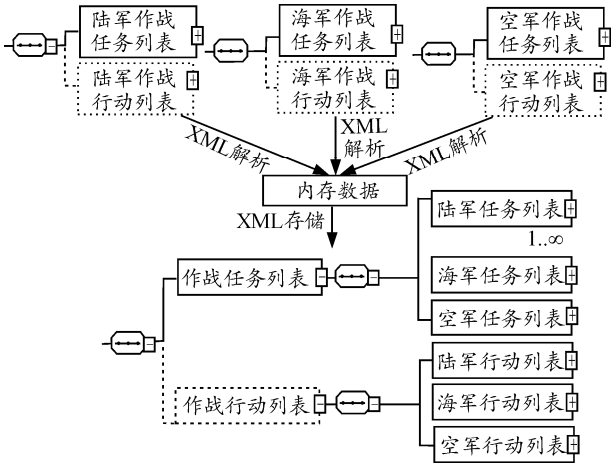

图 7　协同作战任务和作战行动数据的融合

3　结束语

分析结果表明：异步式的电子对抗仿真想定协同编辑方法，能够较好地满足电子对抗仿真想定协同编辑的需求。下一步，将深入实践，完成想定编辑系统的设计与实现。

参考文献：

[1] 蔡志宽, 张洪斌, 唐德金. 电子对抗仿真及其模型的检验[J]. 航天电子对抗, 2000(2): 30-32.

[2] 包金宇, 廖文和. 面向协同设计的冲突自动消解系统研究[J]. 机械科学与技术, 2007, 26(2): 254-256.

[3] 黄智, 邱晓刚. 军事想定定义语言 MSDL 技术研究[J]. 计算机仿真, 2008, 25(8): 9-11.