文章编号:1005-8451(2004)08-0026-03

XML 在铁路确报传输系统中的应用

唐 堃
(铁道科学研究院 电子计算技术研究所,北京 100081)

摘 要:为了提高确报系统中信息的及时性和可扩展性,以及其与 TMIS 中其它子系统的交互能力,确报 2.0 版本采用 XML 作为传输文件格式。该系统在开放源码的 expat XML 库的基础上,实现自己的解析模块,结构合理,易于扩展,可以无缝地提供 WEB 等高级服务,并进一步做到数据读写和现有系统无关。大量的实际应用表明,XML 的使用大大简化确报业务实现的复杂性,增强系统的模块化,提高了系统的可靠性,对于整个 TMIS 建设起到了较好的推动作用。

关键词:确报系统;XML 解析器;文档对象模型;简单接口模型
中图分类号:TP39 文献标识码:B

Application of XML technology to Railway Acknowledgement Transmission System

TANG Kun
(Institute of Computing Technology, China Academy of Railway Sciences, Beijing 100081, China)

Abstract: In order to improve the promptness and the scalability of the acknowledgement system, as well as its interoperability with the other subsystems in TMIS, Acknowledgement System 2.0 adopted XML as the transmission format. Based on the open source expat XML lib, the system achieved the parser method, having a reasonable architecture and pretty goof scalability. Large amount of practical applications suggested that the application of XML module could greatly reduce the complexity, modularize large system and improve its reability. Consequently, it has a good social and economic benefit.

Key words: Acknowledgement System; XML parser; document object model; simple interface model

随着铁路 TMIS 应用的不断深入,各级系统的整合工作已初见成效,信息化为运输生产指挥带来的效果逐步明显。同时,各级系统对基础信息的完整性、及时性和准确性的要求也日益加强。其中,对作为运输指挥主要信息源的确报的要求最为紧迫。由于设计时间较早,目前采用的"运统一文件交换格式"(以下简称交换格式)无论在格式上、内容上都存在一定缺陷,在一定程度上影响了确报质量。粗略地说,目前的格式至少存在下列问题:

(1)交换格式没有提供必要的数据完整性限制,导致一些无效信息的存在,如重车缺少发到站、空车缺少发站等;(2)格式无法识别重报,导致大量重复报文的存在;(3)格式缺少一些必要的内容或格式位数不足,如发车日期、5 位车种等问题,只能通过一些变通的手段迂回解决;(4)格式无法从确报导航到货票、装载清单等和车辆密切相关的其它信息,造成了货运系统实施的困难。

为根除上述弊病,铁道部已决定对确报系统从根本上进行完善,并作出了有关的具体指示。新版确报传输格式的设计方案,就是根据铁道部颁布的《增加新确报传输格式的建议》、《确报升级改造软件开发计划》和《确报软件升级工作可行性报告》的基本精神制订的。为了有更好的可扩展性,新版的确报在传输层上,使用了 XML 技术。

收稿日期:2004-01-10
作者简介:唐 堃,研究实习员。

1 总体框架流程

确报传输格式中采用 XML 作为网络传输格式,其主要流程为:发送端数据从源数据库导出之后由应用程序将其转换成 XML 格式的文件,然后调用 IBM 消息中间件 MQ Series 将该文件发送到接收方。接收方收到该文件之后调用 XML 接口函数,对该文件内容进行解析,并从解析后的内容中得到所有有用信息,然后由数据库导入程序将其装载入接收方数据库中,如图 1。

图 1 中可以看出,需要进行 XML 格式转换的有两部分:a. 发送方:对从数据库导出的数据编码,将其转换成 XML 文件。b. 接收端:从 MQ 接收到的 XML

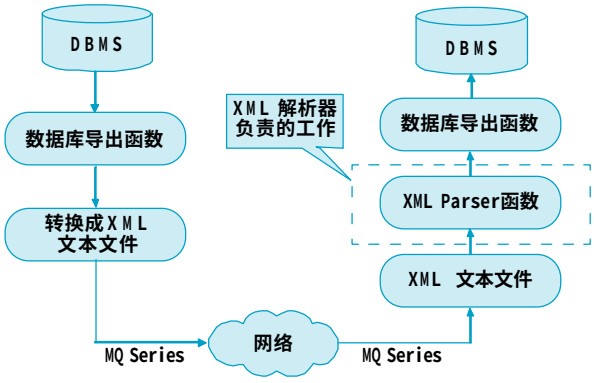

图 1　确报传输总体框架流程

文件中解析出所有信息，得到有用信息，然后由数据库导入程序将其装载入接收方数据库中。

2　XML 编程接口的选型

XML 提供多种编程接口。这些接口为开发人员使用 XML 文档提供了统一的方法，有许多 API 可以使用；一些新的以源码形式发布的与 XML 相关的工具都是采用 Java 编写。尽管 Java 在 XML 竞技场中占有明显优势，许多 C/C++ 程序员还是在进行 XML 的开发，而且现在有许多 XML 工具可供 C 和 C++ 程序员使用。考虑到TMIS 的大部分应用中是基于C/C++，而且铁路系统中正在运行的各种 Unix 并不是都具备 Java 虚拟机的安装条件，所以我们选择了 C/C++ 的 XML 解析器。以下将从解析器和解析器API 模型和验证方式简要介绍我们在选择XML 库时的考虑。

2.1　解析器和解析器 API 模型的选择

拥有了 DTD 或模式以及与其相配的 XML 文档，就需要一个解析器来读取并解释该 XML 文档。目前最流行和广泛使用的 C/C++ 的 XML 解析器API 有2种：文档对象模型（Document Object Model (DOM)）和用于 XML 的简单 API（Simple API for XML (SAX)）。DOM 为 XML 文档的已解析版本定义了一组接口。解析器读入整个文档，构建一个驻留内存的树结构，你的代码就可以使用 DOM 接口来操作这个树结构。可以遍历树以了解原始文档包含了什么，可以删除树的几个部分，还可以重新排列树和添加新的分支等。DOM 的使用是有代价的，它创建了整个文档驻留内存的树，在解析文档之前必须读取整个文档，它的优势也在于对内存树的增、删、改。用于 XML 的简单 API（通常称为 SAX）是基于事件的 API。解析器将事件（譬如，元素的开始或结束）发送给处理信息的事件处理程序。然后，应用程序自己可以处理数据。虽然原始文档保持不变，但 SAX 提供了操纵数据的方法，然后会将该方法导向另一个过程或文档。SAX 是对 XML 文档的流式解析，并不驻留在内存之中，因此也不能对 XML 文档进行改动，所以速度上也快于 DOM。

用于将软件与 XML 解析器结合的两种常用 API 模型是：文档模型和事件模型。文档 API 模型对 XML 数据进行解析以生成一个对象。对象将文档的内容抽象成树结构。应用程序对这个树结构对象进行操作。事件 API 模型使用回调机制向应用程序通报 XML 数据的结构，事件（/回调通常在进行解析时发生。在确报传输系统中，需要的是对文档解析，并不涉及文档的改变，在此我们选用 SAX 接口模型。

2.2　选择验证方式和对API标准支持的考虑

XML 解析器有两种形式：验证和非验证。根据W3C 建议的XML1.0 规范的定义，按照满足XML 规范和进一步的约束的不同，XML 文档被划分成格式良好的（Well-Formed）XML 文档和有效的（Valid）XML 文档两类，两者的严格定义请参见《XML1.0 规范》。其中，格式良好的XML 文档比较容易实现，市场上已经存在很多支持其格式校验的产品，但是其本身不能实现严格的有效性校验。比如对非军运重车的到站内容必须非空的要求，就无法从其内部定义实现。但是我们可以通过额外的校验规则进行控制）；有效的XML 文档要求较严，目前完全满足其校验要求的产品很少，如果采用这种方式，将会大幅度增加开发的进度风险。因此，以格式良好的XML，即非验证方式，作为确报传输的格式比较合适。

一般解析器 API 模型已经被进一步改进成特定API 标准。W3C 已经推荐 DOM 级别 1 和 2，级别 3 正在草拟中）作为标准化文档 API 模型。SAX 虽然不是 W3C 项目，但它已经占有了事实上的标准事件API 模型的地位。

基于以上几点考虑，在确报系统中，我们选定了用 C 语言开发的 SAX 接口模型的 XML 解析器。在目前较成熟的 XML 解析器，最流行的开放源码 XML 库是 Expat、Libxml、Xerces和XML4C。这4者都是跨平台的，每一种都充当 XSLT 库实现的基础，一旦满足了基本 XML 需要之后，它就会给出一条成长途径。经过在多种操作系统（如 Win, Linux, BSD, Solaris, HP-UX和 AIX 等上的测试，Expat 可以满足我们的要求。Expat 是 James Clark 创始的开放源码面向事件的

XML 解析库。现在该项已经转让给了 SourceForge 中的一个小组。在 GNU 许多项目中都可以找到 Expat 解析器，如开放源码浏览器 Mozilla、XSLT 处理器 Transformiix 和 RDF 工具Repat。将Expat库编译并链接到已有项目毫不费力，并且 Expat 还包括了 DSP 项目 makefile。对于在 Linux 或 Unix 上运行的项目，大多数情况下，可以将源代码直接解包（untar）到一个空目录中，设置某些选项，然后输入“make”来构建共享库。

3 函数详细说明

应用中解析模块的作用是从输入的 XML 文件中解析出有用信息并将其放入到本地内存结构体中，在此我们通过一个函数int xml_to_struct(char *filename, QBINFO *qbinfo_ptr) 来完成任务。其中char *filename是输入的 XML 文件名，qbinfo_ptr 是输出信息的结构体指针，对应的结构体由调用该函数的应用预先分配。该函数内部采用的是 Expat XML Parser[2]，它是一个 SAX 类型的解析器。并不是在内存中对整个字符流生成一个 DOM TREE 才对其进行解析，而是直接在字符流中捕捉指定的标签和标签内容（），主要涉及的 Expat 函数有以下 6 个：

（1）XML_Parser

XML_ParserCreate（const XML_Char *encoding）；创建解析器，指定该解析器译码方式，默认为 UTF-8，可以设定值为：ISO-8859-1，US-ASCII，UTF-8和 UTF-16。

（2）Void XML_Parser

XML_SetUserData (XML_Parser p, void *userData)；该函数的功能是从将主函数的参数传递到各个回调函数，以及回调函数之间的参数指针传递。

（3）回调函数

XML_SetElementHandler(XML_Parser p，XML_StartElementHandler start，XML_EndElementHandler end)；同时设定开始和结束标签，假如遇到一个 XML 文件中的一个开始标签就执行 start 函数，结束标签就执行 end 函数。

（4）回调函数

XML_SetCharacterDataHandler (XML_Parser p，XML_CharacterDataHandler charhndl)；设定数据内容标签，加入遇到一个 XML 文件中的数据项就执行 charhndl 函数。

（5）XML_Status XML_Parse

(XML_Parser p，const char *s，int len，int isFinal)；将整个字符流 s 中内容送入解析器进行解析，isFinal 表明该字符流 s 是否为最后一个待解析的字符流。

（6）Void XML_ParserFree

(XML_Parser p)释放解析器所占用的内存空间。

xml_to_struct 函数内部实现流程如图 2。

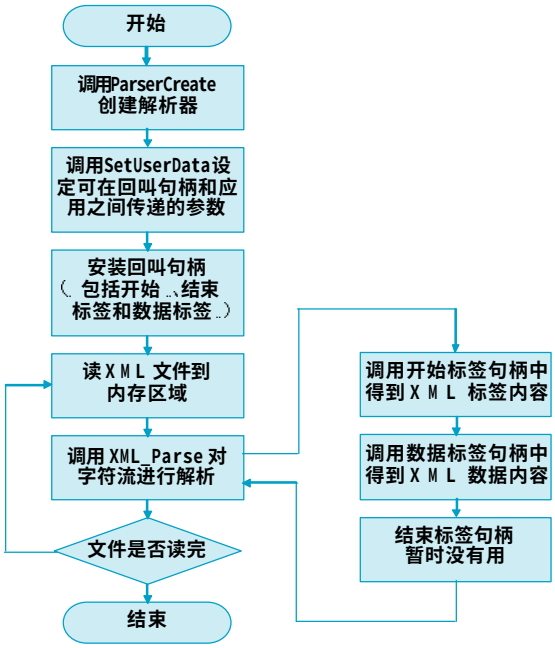

图 2 解析模块函数内部实现

4 结束语

在测试环境发现解析器构建之初会占用大约 20 K 内存，之后测试循环 1 000 次，该内存没有增加，所以认为是初始化加载函数库时分配了内存空间，对运行不会有影响。进程结束之后，内存空间自动释放。在对上述方案的研发和实施的基础上，我们也对该方案的风险进行了充分研究，认为方案结构合理，易于扩展，可以无缝地升级到有效的 XML 格式，并进一步做到数据读写和现有系统无关，对于整个 TMIS 的应用起到了较好的推动作用。

参考文献：

[1] 孙 健. 分布式信息共享技术及其实现[D]. 北京：铁道科学研究院硕士学位论文，2003，5.

[2] C-C++ 开发人员充实您的 XML 工具箱[EB/OL]. http://www-900.cn.ibm.com/developerworks/cn/xml/php-xml-toolkit/index.shtml.