

XML 在嵌入式系统中的应用

钟 华,李海燕,马家宇

(南华大学 计算机科学与技术学院,湖南 衡阳 421001)

摘 要:阐述了 XML 在嵌入式系统中使用的背景,重点分析了嵌入式环境中数据存储和数据交换过程中的 XML 的验证机制、数据封装、数据打包解包过程,根据实验测试数据,给出了 XML 技术在实时数据交换中的性能分析报告,测试结果表明实时数据交换平台的性能符合应用要求。

关键词:XML;嵌入式系统;数据存储;数据交换

中图分类号:TP311 **文献标识码:**B

Application of XML in the Embedded System

ZHONG Hua, LI Hai-yan, MA Jia-yu

(School of Computer Science and Technology, University of South China, Hengyang, Hunan 421001, China)

Abstract: This paper firstly presets the background of XML used in the embedded system. Then provides an analysis for the data validation mechanism, encapsulation, data packing and unpacking of XML in the process of storage and transform. Finally, it gives a performance report of XML technology in real-time data transform according to the experimentation, and indicates the performance of real-time data transform platform is required for application.

Key words: XML; embedded system; data storage; data transform

XML(eXtensible Markup Language,可扩展标记语言)是由 W3C(World Wide Web Consortium,互联网联合组织)为克服 HTML 的局限,于 1998 年 2 月发布的一个标准^[1],它是标准通用标记语言(SGML)的一个子集,具有跨平台、结构清晰、支持动态修改、可移植等特性。在嵌入式环境中,XML 用统一的接口来管理不同的设备接口及其相关操作,提高了设备访问的透明性及相关平台的交互能力^[2],具有很强的扩展性,因而在异构

多变的嵌入式环境中的数据存储和数据交换中具有广泛的应用。

1 在嵌入式系统中使用 XML 技术的背景

XML 是一种元数据语言,取用了 SGML 的核心部分,具有明显的灵活性和简洁性。XML 既是一种语言,又是一种标准,它提供一种格式来描述结构化数据,允许开发者以统一的标准在应用程

序中描述和传输结构化数据,不同类型的嵌入式终端设备能从同一格式信息中提取数据,进行不同的处理和使用。

1.1 嵌入式系统中的数据存储问题

在嵌入式系统中,经常需要使用文本文件来存储数据,特别是配置数据,但是这些系统并不像 Windows 中那样有一个通用的配置文件格式(.ini),需要自行设计文本数据的存储格式,造成了文本数据文件格式的不统一,可移植性很差。同时,大部分自行设计的文本文件都不支持“写”操作,原因是实现较为复杂。而使用 XML 文件作为文本数据文件的格式,则不仅可以统一系统中的文件格式为一种开放的标准格式,而且借助于 XML 语言强大的数据描述能力,还可以使系统更加灵活方便,同时,XML 能够以一种标准的方式很好地支持“写”操作。

1.2 嵌入式系统中的数据交换问题

嵌入式系统中的各个不同模块经常需要交换数据,如果这其中的数据格式使用直接的数据结构,则会造成这些软件模块之间的依赖关系太紧密,降低了系统的灵活性,如果一个模块修改了,往往另外的模块也需要做对应的修改。虽然可以引入许多其它的数据通信格式来解决,例如 TLV 格式、文本格式等,但这些格式大部分也不是一个开放的标准,通用性不是很强。若使用 XML 格式的数据包,则不仅可以很好地解决通用性问题,而且也是一种比较标准的解决方案。

2 XML 技术在数据存储中的应用

文本数据文件大多是人为编辑的,所以很难保证它一定是有效的。其它格式的文本文件基本没有有效性验证机制,而 XML 文件则可以通过 DTD(Document Type Declaration)或者 Schema 机制来验证文件的有效性。DTD 是一种保证 XML 文档格式正确的有效方法,可以通过比较 XML 文档和 DTD 文件来看文档是否符合规范,元素和标签使用是否正确。DTD 是 XML1.0 的默认验证机制。而 Schema 则是新一代的 XML 验证机制,它比 DTD 更优越,提供了比 DTD 更强大的功能,多个 Schema 可以复合使用 XML 名称空间,可以详细定义元素的内容及属性值的数据类型。另外 XML 支持 XPath 语言查询 DOM 树的节点集。DOM(Document Object Model)是一种在浏览器内编程的模型^[3],其定义了一组与平台和语言无关的接口,它是基于树的 API,它把所有的数据以父子的

节点层次结构装入内存构成一棵树,这些节点的类型可以是元素、文本、属性、注释或其它。它允许开发者读取、创建、删除和编辑 XML 数据,以便程序和脚本能够动态访问和修改 XML 文档内容、结构及样式,从而很方便的进行数据存储和查询。

3 XML 技术在数据交换中的应用

3.1 XML 数据封装

嵌入式系统中的不同模块之间往往需要交换数据,用于封装这些交换数据最常见的格式有以下四种:

1)直接的数据结构封装。直接交换要求其数据结构是不需要打包和解包,发送端直接发送需要交换的数据结构,而接收端则可以直接将收到的数据包作为指定的数据结构即可,这种方法的效率最高。缺点是发送和接收两模块之间的依赖性特强。

2)使用 RPC(Remote Procedure Call)之类的抽象通信方式进行数据交换。这种数据交换的优点是上层程序无需关心底层的数据交换细节,仅仅把一个远端的过程调用当作本地的过程调用即可,RPC 封装了所有数据交换的细节^[4]。缺点是 RPC 两端软件的依赖性仍然很强。

3)封装成 TLV 之类的格式。TLV 格式的特点是将数据按照 T(Type)、L(Length)、V(Value)进行封装,发送端将需要交换的数据结构的各个字段进行打包,而接收端则按照类型和长度分别从数据包中取出各个字段,重新组合成需要的数据结构。这种方法大大提高了数据的兼容性。但它的缺点是需要打包和解包,效率上有一定程度的降低,且数据的可读性和可调试性还是比较差。

4)封装成文本数据流。其宗旨就是将需要发送的数据都打包成对应的字符串然后再进行发送,而接收端则作相反的转换,重新组合成需要的数据结构。这种方式不再需要考虑数据的字节序等一系列问题,并且使数据的可读性和可调试性都大大提高。其缺点是数据的打包解包效率进一步降低。

在嵌入式数据交换中使用 XML 技术,则充分利用了 TLV 和文本流方法各自的优点,既大大提高了软件模块的兼容性,又摒弃了字节序转换等一系列繁琐的细节问题,并且使所交换的数据的可读性有明显改善,大大提高了系统的可调试性。更为重要的是 XML 作为一种开放的标准技术,只要提供数据交换的 Schema 验证规则,在发送端可

方便地为本系统编写模块.在接收端,则可使用一个标准的XML解析器将收到的XML文本包进行分析,生成一棵DOM树,然后再遍历该DOM树或者使用XPath语言等访问其中的节点数据,并将这些数据重新组合成所需要的数据结构.

3.2 XML 格式数据打包

在嵌入式系统中,经常使用的数据无非是整数、字符串以及由它们组合而成的结构或者数组,打包技术略有所区别.

1)整数的打包.将整数转换为对应的字符串,作为对应元素的内容,例如:`int data = 100;`打包成如下元素:`<data>100</data>`

2)字符串的打包.直接将该字符串作为对应元素的内容,例如:`char * data = "hello,test";`打包成如下元素:`<data>hello,test</data>`

3)结构的打包.将该结构的各成员作为该结构元素的子元素,将各成员的值作为对应子元素的内容,例如:

```
struct S
{ struct S data = { 100, "hello,test" };打包
成如下元素:
```

```
<data> <a>100</a> <b>hello,test</b> </data>
```

4)数组的打包.将数组名称作为父元素,数组的各元素作为该父元素的子元素,子元素名称统一为'v',数组各元素的值作为各个元素'v'的内容,例如:

```
int data[4] = { 100,200,300,400 };打包成
如下元素:
```

```
<data> <v>100</v> <v>200</v>
<v>300</v> <v>400</v> </data>
```

3.3 XML 格式数据解包

在数据的接收端,接收到的是XML格式的数据包,必须要首先使用XML分析器将数据分析生成一棵DOM树,然后再遍历该树,访问其中的节点,具体过程基本上是打包过程的逆过程,但是考虑到模块之间的兼容性,应该注意以下几点:1)遇到自己认识的元素名称把其中的数据取出来,存入解包数据结构的对应位置;2)遇到自己不认识的元素名称,则直接忽略;3)在遍历DOM树完成后,如果发现有一些元素在该DOM树中不存在,则认为这些元素不存在,按照实际情况将它们置成初始值或者默认值或者不要修改它们.

4 XML 在实时数据交换中的性能测试分析

由于使用XML作为实时交换数据包的格式需要首先在发送端打包,然后在接收端解包,所以与直接的二进制数据交换相比,该过程需要额外的性能开销.如果该开销太大,而实际的运行环境又对性能要求十分严格的话,那么XML的打包解包过程可能成为整个系统的瓶颈,从而导致整个系统无法正常运行,因此需要对XML数据包的打包解包过程进行性能测试.由于数组的打包是比较占用空间,测试XML的性能瓶颈时应该选择一个带有较多数组元素的结构进行测试.选择不加任何空格和换行字符数据打包后大小为2038字节的数据包,分别在Windows、Linux和Vx-Works平台下循环10 000次,记录所花费的时间,结果见表1(单位:ms).

表1 性能测试分析表

Table 1 Analysis table of performance testing

测试平台	打包方法		
	直接打包	DOM树方法打包	XML分析器解包
Windows 平台	1 250	3 860	3 828
Linux 平台	1 250	5 020	5 020
Vxworks 平台	2 210	8 690	8 680

由测试结果分析可见,XML打包解包确实增加了系统额外的开销,但在此实时性要求不太高的嵌入式环境中,且存储空间和网络性能都非常优越,XML的有效应用不是此系统的性能瓶颈,可以不考虑这些额外的开销,实验表明实时数据交换平台的性能符合应用要求.

参考文献:

- [1] Frank S, Torsten K. Towards XML oriented internet management[C]//Marz. Proc. of the Eighth IFIP/IEEE International Symposium. Colorado, IEEE/IFIP, 2003: 505-518.
- [2] Gutleber J, Magrans I, Nafria M. Uniform management of data acquisition devices with XML[J]. IEEE Transactions on Nuclear Science, 2004, 51(3): 903-908.
- [3] 乔磊, 李曦, 龚育昌, 等. 一种支持XML的文件系统构建模型[J]. 小型微型计算机系统, 2008, 29(8): 1431-1436.
- [4] 袁非, 陆阳. XML在嵌入式环境下的应用[J]. 合肥工业大学学报(自然科学版), 2007, 30(4): 445-448.