1.为下面的xml文件写一个schema文档

<?xml version="1.0"encoding="GB2312"?>

<学生名册 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="studentschema.xsd"> (最后需要加入的部分)

<学生 学号="1">

<姓名>张三</姓名>

<性别>男</性别>

<年龄>20</年龄>

</学生>

<学生 学号="2">

<姓名>李四</姓名>

<性别>女</性别>

<年龄>19</年龄>

</学生>

<学生 学号="3">

<姓名>王二</姓名>

<性别>男</性别>

<年龄>21</年龄>

</学生>

<学生 学号="4">

<姓名>王二</姓名>

<性别>男</性别>

<年龄>21</年龄>

</学生>

</学生名册>

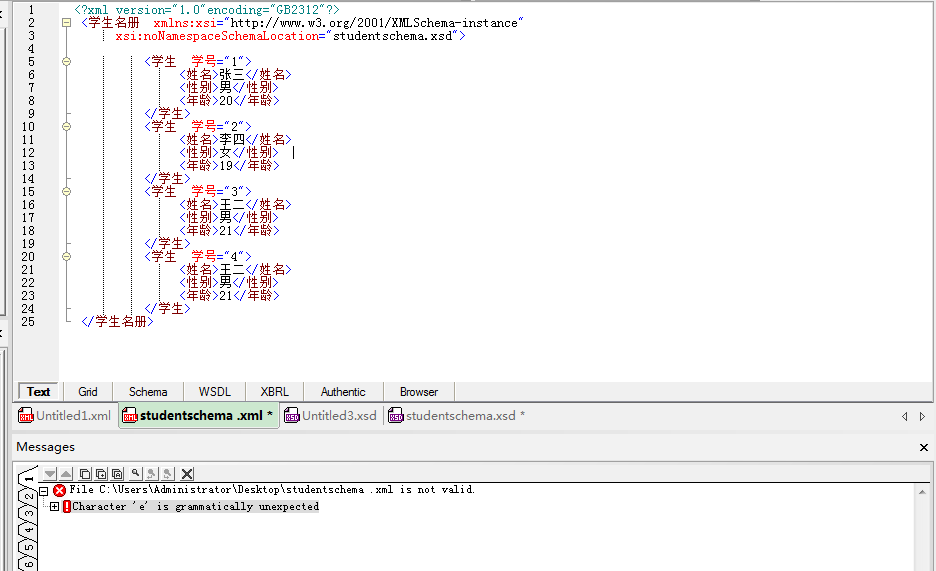
并用XML spy进行校验

schema文档

1. <?xml version="1.0"encoding="GB2312"?>
2. <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"
3. targetNamespace="http://www.hexinli.com"
4. xmlns="http://www.hexinli.com">
5. <xs:element name="学生名册">
6. <xs:complexType>
7. <xs:sequence>
8. <xs:element name="学生"type="我的类型"maxOccurs="unbounded"/>
9. </xs:sequence>
10. </xs:complexType>
11. </xs:element>
12. <xs:complexType name=" 我的类型">
13. <xs:sequence>
14. <xs:element name="名字"type="xs:string"/>
15. </xs:sequence>
16. <xs:attribute name="学号"type="xs:int"/>
17. </xs:complexType>
18. </xs:schema>
19. <xs:element name="学生">
20. <xs:complexType>
21. <xs:sequence>
22. <xs:element ref="姓名"/>
23. <xs:element ref="性别"/>
24. <xs:element ref="年龄"/>
25. </xs:sequence>
26. <xs:attribute name="学号"use="required"typr="xs:int"/>
27. </xs:complexType>
28. </xs:element>
29. <xs:element name="姓名"type="xs:string"/>
30. <xs:element name="性别">
31. <xs:simpleType>
32. <xs:restriction base="xs:string">
33. <xs:enumeration value=" 女"/>
34. <xs:enumeration value=" 男"/>
35. </xs:restriction>
36. </xs:simpleType>
37. </xs:element>
38. <xs:element name="年龄"type="xs:int"/>

截图





浅谈基于XML的数据交换

1 XML优点及相关技术

XML是World Wide Web联盟（W3C）的一个开放标准，它是一组规则和准则的集合，由于以无格式文本（而不是二进制格式）来描述结构化数据，因此它具有良好的数据存储格式、开放性、可扩展性、自描述性、高度的结构化、便于网络传输等特性。XML所关心的主要是数据，而其它的因素像数据结构和数据类型、表现和操作都由其它的以XML为核心的相关技术来完成。其中XSLT(Extensible Stylesheet Language Transformations)用于将XML文档转换成其它类型的文档或其它格式的XML文档。Xpath(Xml Path Language)用于定义如何在XML文档中查找和定位数据。XML Schema用XML描述了一类XML文档的数据结构和数据类型。通过使用上述几种技术，我们便可方便地把需要交换的数据转换成XML文档，在互联网和企业内部的不同应用程序间方便地提交数据、处理数据，完成数据的交换。

2 基于XML的数据交换模型

2.1数据交换的实现模式

（1）面向数据的交换模式 面向数据的数据交换发生在不同系统的数据库之间，通过将数据从一个数据源移植到另一个数据源来完成数据的交换。其优点是它直接在数据访问层作应用开发，无须对应用逻辑与数据结构作任何改变。缺点是因为要直接对数据库进行操作，编程人员需要对原有数据访问层作大量修改同时又必须保持数据的完整性；在跨越互联网进行数据交换时，因为系统需要了解双方数据库的实现方式，需要承担很高的安全风险；而且，对于不同的应用，这种数据交换方式需要作不同的设计，系统的可重用性很低。

（2）面向应用接口的交换模式 面向应用接口的数据交换是指按各个应用接口所需数据格式在接口间转换数据、传输数据。这样，开发者就能够将现有应用捆绑在一起，允许它们共享商业逻辑和信息，这种策略的局限性一般是由接口的特征和功能所决定的。

（3）面向方法的交换模式 面向方法的数据交换是指在网络环境中的跨平台应用程序之间，建立一个可供多方共享的方法，实现数据共享和交换，采用面向方法的数据交换成可以实现Internet环境下的企业应用的松散耦合和集成，使企业可以方便地集成现有的应用并开发新的应用。

实例见网址https://wenku.baidu.com/view/8a61e804eff9aef8941e065d.html?from=search