

## XML DOM 在自动排课系统中的应用

朱创录, 钟 东

(西北大学 计算机科学系, 陕西 西安 710069)

**摘 要:** XML作为一种数据交换的标准,已经贯穿于 Internet 应用的各个领域之中。基于 XML 的文档对象模型(DOM)是一种与平台无关、语言无关的标准接口,是 XML 文档操作的基础。文中讨论了通过 XML DOM 为主要数据处理技术来进行自动排课系统设计中的一些关键技术及解决方法。

**关键词:** 文档对象模型;XML 模式;自动排课

**中图分类号:** G434

**文献标识码:** A

**文章编号:** 1005-3751(2006)01-0190-03

## Application of XML DOM in System of Automated Courses Scheduling

ZHU Chuang-lu, ZHONG Dong

(Department of Computer Science, Northwest University, Xi'an 710069, China)

**Abstract:** As a standard of data exchange over the Web, XML has penetrated virtually all fields of Internet applications. Document object model (DOM) based on XML is a set of platform-independence and language-independence interfaces, and it is the basis of the implementation of XML document. Talked about much main technology in automated courses scheduling system which is based XML DOM dealing with data.

**Key words:** DOM; XML schema; automated courses scheduling

## 0 引言

众所周知,虽然 Internet 的发展不过十多年,但它对人们工作、学习、生活的方方面面却产生了深远的影响。一时间,电子商务、网上出版、网络通信,许许多多的新名词如潮水般涌来。在这个网络大潮之下,XML(Extensible Markup Language),这一以第二代网页发布语言而著称的新标准,凭借着它的勃勃生机与强大优势,为网络应用注入了新的活力。XML 是一种描述结构化数据的语言,它支持自定义的面向特定领域知识的标签。大多数的 XML 应用程序可归为以下 3 类:文档化、数据交换和数据库连接<sup>[1]</sup>。在以往已经有很多方法来处理自动排课系统中的数据,但普遍存在数据描述不精确、处理数据时繁琐等特点,而文中的自动排课系统就是以 XML 作为信息的主要描述方式,克服了以上的缺点,使整个设计、编程过程更加条理和简化,其总体设计如图 1 所示。

## 1 XML 数据模式与信息建模

XML 技术是互联网下一个发展阶段的关键技术,它已成为互联网上表现结构化和半结构化数据的标准格式和数据交换的主要标准。在这种背景下,把 XML 作为一

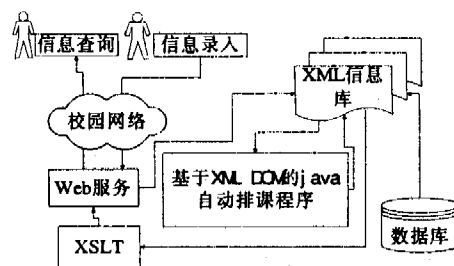


图 1 总体设计

种数据库模型并从中提取信息在数据库领域逐渐受到注意。而且,XML 技术和关系数据库技术的融合能使数据更通用,能被异构系统接受,因此成为很自然的趋势<sup>[2]</sup>。大量的 XML 文档由于不同应用之间交换数据而临时产生,而且也存在像数据库一样需要系统可靠地管理永久性 XML 数据的需要。为此,已经有许多的研究工作致力于融合 XML 应用和关系数据库。特别是关于信息模型的分析对系统设计起着至关重要的作用,直接关系到整个系统在开发过程是否顺利。对于 XML 文档型数据库的设计主要是对 Schema 的设计,XML Schema 本身也是 XML 文档的一种,所以 XML Schema 具有 XML 所有的特点。XML 是典型的树形结构,然而这种结构既然包含了元数据和数据,目的是为了能让机器更好地理解,它的物理表现也就不那么直接地利于人的理解,但它对数据的表现具有良好的灵活性和确定性<sup>[3]</sup>。因此 XML Schema 数据模式就成为整个开发过程中的重点和难点。

信息建模是设计结构合理的 XML Schema 的基础,在信息建模的过程中,通常人们希望在对信息精确定义的同时又要求被定义的信息之间能够获得良好的通信,但往往在实际的设计过程中这两方面存在着冲突,即表现在静态信息模型和动态信息模型之间的冲突,因此,要设计一个合理的信息模型,就必须综合考虑静态信息模型与动态信息模型,以及之间的相互作用关系。在信息建模的过程中,有些人喜欢从静态模型着手进行信息建模;有些人喜欢从动态模型着手进行信息建模,但不管从静态模型开始还是从动态模型开始,都必须把两种模型进行详细地分析,而不能忽略其中的一种。以下就自动排课系统中的静态和动态信息模型分别加以介绍。

### 1.1 自动排课系统中的静态信息

静态信息类似于数据库中的信息,信息被长期保存,并可以用于多种用途。在自动排课系统中,分别有以下静态信息:班级、课程、教师、教室。这些信息的特点是,一开始就已经确定其信息库中的内容,在系统的运行期间始终保持不变。以下我们以 XML Schema 的层次关系图来表示这些静态的信息。

(1) 班级:其中包括班级名称、班级总人数及班级代号,每个班级名称又具有系名、年级、专业等元素。各元素之间的层次关系如图 2 所示。

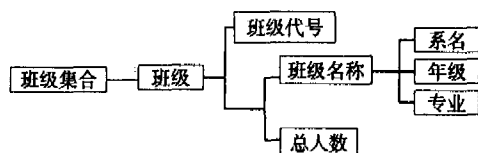


图 2 班级模式的层次关系

(2) 课程:其中包括课程名称、课程优先级、有无多媒体要求、总课时及课程代号。各元素之间的层次关系如图 3 所示。

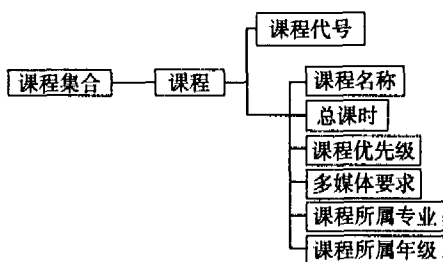


图 3 课程模式的层次关系

(3) 教师:其中包括教师代号、姓名、可带的课程及课程评价价值。各元素之间的层次关系如图 4 所示。

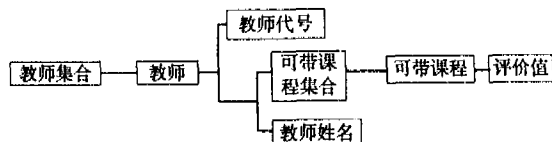


图 4 教师模式的层次关系

(4) 教室:其中包括教室代号、可容纳人数、有无多媒体。各元素之间的层次关系如图 5 所示。

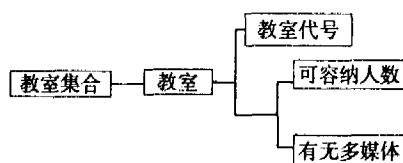


图 5 教室模式的层次关系

### 1.2 自动排课系统中的动态信息

动态信息是直接与设计相关的,信息存在的时间非常短暂,而且用途专一。在自动排课系统中,动态信息也可以通过 XML 文档的形式进行表达,但这些信息作为程序在处理过程中的中间信息,而且随着处理过程在不断的变化。在自动排课系统中,这些动态的信息之间的关系如图 6 所示。

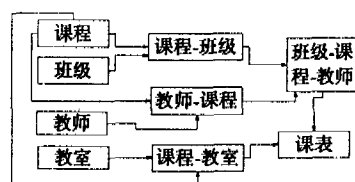


图 6 动态信息关系图

在图 6 中,可根据课程和班级信息库中的关联产生班级与课程的相关联的 XML 文档,根据课程和教师的信息库产生课程和教师相关联的 XML 文档,然后根据班级与课程的相关联的 XML 文档及课程和教师相关联的 XML 文档产生班级、课程、教师相关联的 XML 文档;另外,由课程和教室的信息库可产生课程与教室相关联的 XML 文档;最后,根据班级、课程、教师相关联的 XML 文档及课程与教室相关联的 XML 文档产生所需要的课表。例如,课程教室关联的 XML 文档如下所示:

```
<? xml version="1.0" encoding="UTF-8"? >
<roomCourses>
  <classroom id="教师代号">
    <第一节>
      <课程></课程>
    </第一节>
    <第二节>
      <课程></课程>
    </第二节>
    <第三节>
      <课程></课程>
    </第三节>
  </classroom>
</roomCourses>
```

在进行教室和课程相关联处理时,经过课程信息库和教室信息库的搜索对比可将关联后的课程信息插入到课程元素的子元素处。

## 2 DOM 与数据处理

XML 的层次化信息表示方法,灵活地表示了人们所需要的系统的信息模型,要在这些信息模型中把数据抽象

出来,DOM 就是可选方法之一。当使用 DOM 对 XML 文本文件进行操作时,它首先要解析文件,将文件分解为独立的元素、属性和注释等。然后,它以节点树的形式(在内存中)创建 XML 文件的表示。此后,开发人员可以通过节点树访问文档的内容,并根据需要修改文档。事实上,DOM 执行了更进一步的操作,它将文档中的每个项目看作节点—元素、属性、注释、处理指令,甚至构成属性的文本。

### 2.1 DOM 内部逻辑结构

DOM 文档中的逻辑结构可以用节点树的形式进行表述。通过对 XML 文件的解析处理,XML 文件中的元素便转化为 DOM 文档中的节点对象。DOM 的文档节点有 Document, Element, Comment, Type 等节点类型,其中每一个 DOM 文档必须有一个 Document 节点,并且为节点树的根节点。它可以有子节点,或者叶子节点如 Text 节点、Comment 节点等。任何的格式良好的 XML 文件中的每一个元素均有 DOM 文档中的一个节点类型与之对应。利用 DOM 接口将 XML 文件转化成 DOM 文档后,就可以自由地处理 XML 文件了。

### 2.2 使用 DOM 处理 XML 文档的优越性

#### (1) DOM 能够保证正确的语法和格式正规性。

由于 DOM 将文本文件转化为抽象的节点树表示,因此能够完全避免无结束标记和不正确的标记嵌套等问题。使用 DOM 操作 XML 文档时,开发人员不必担心文档的文本表示,只需要关注父子关系和相关的信息。另外,DOM 能够避免文档中不正确的父子关系。例如,一个 Attr 对象永远也不能成为另一个 Attr 对象的父对象。

#### (2) DOM 能够从语法中提取内容。

由 DOM 创建的节点树是 XML 文件内容的逻辑表示,它显示了文件提供的信息,以及它们之间的关系,而不受限于 XML 语法。例如,节点树蕴含的信息可以用于更新关系数据库,或者创建 HTML 页面(开发人员不必纠缠于 XML 语法规则)。

#### (3) DOM 能够简化内部文档操作。

就修改 XML 文件的结构而言,使用 DOM 比使用传统的文件操作机制更加简单。利用 DOM 在文档中增加、删除、修改元素非常简便。另外,你可以通过几条命令执行全局性操作(例如:从文档中删除具有特定标记名称的所有元素),而不必采用繁琐的方法——首先对文件进行扫描,然后删除相关的标记<sup>[4]</sup>。

(4) DOM 能够贴切地反映典型的层次数据库和关系数据库的结构。

DOM 表示数据元素关系的方式非常类似于现代层次型和关系型数据库表示信息的方法。这使得利用 DOM 在数据库和 XML 文件之间移动信息变得相当简单。

DOM 以上的优越性表现了它在 XML 文档数据处理方面的能力,但就 DOM 而言,它的处理能力也仅仅局限在 XML 文档中的数据的抽取、数据的保存,在 DOM 操作

中表现为元素及属性节点的查找、修改、删除、插入等操作,对于自动排课系统,因为本身数据量并不是特别大,所以采用 DOM 的方式来处理数据,这样会使系统的实现更加容易。

### 3 自动排课系统中的 DOM 处理

在自动排课系统中,数据的抽取、数据的插入、数据的删除都要用到 DOM 进行。由于整个系统的程序庞大,所以此处以课程信息库及教室信息库关联产生课程及班级相关联 XML 为例来说明 XML DOM 在自动排课系统中的应用<sup>[5]</sup>。

```
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();
Document Cdoc = builder.parse("course.xml");
Document CRdoc = builder.parse("classroom.xml"); //教室信息库
Document RCdoc = builder.parse("roomCourse.xml");
int i = 10; //优先级
int j = 0; //课程顺序
int k = 0; //教室顺序
Element rootC = Cdoc.documentElement;
Element rootRC;
for(; i < 0; i--)
{
    Node coursePre = rootC.childNodes.item(0).childNodes.item(j).
        childNodes.item(2); //取得课程优先级元素
    for(; j < rootC.childNodes.item(0).childNodes.getLength(); j++)
    {
        rootRC = RCdoc.documentElement;
        if(coursePre.nodeName == i)
        {
            int CRi = 0; //设置循环标志
            for(int m = 0; m < rootRC.childNodes.getLength() * 3; m++)
            {
                Node courseInsert = rootRC.childNodes.item(CRi).
                    childNodes.item(j); //要插入的课程
                Node RCcourse = rootRC.childNodes.item(0).childNodes.
                    item(n); //要插入的位置
                RCcourse.appendChild(courseInsert); //插入课程
                if(m == rootRC.childNodes.getLength() - 1)
                    CRi++;
            }
            j++;
        }
        i--;
    }
    FileOutputStream outStream = new FileOutputStream("room-
```

(下转第 195 页)

述系统,最后由程序员进行编程实现。

交互设计的结果不仅仅影响人机交互部分,还可能影响到系统的其它设计部分。因为交互设计是由用户的目标决定功能的,所以交互设计师对单个 use case 或 use case 与 use case 之间的协作的描述可能与建模人员的描述不一致。这种不一致需要通过协商解决。一般情况下,交互设计的结果是交互设计师通过分析用户的目标得到的,设计结果更符合最终用户的需求,所以交互设计师通常对整个软件项目负责,拥有更大的发言权。

交互设计文档和其它设计文档作为设计的结果,共同对后续的过程产生影响。在进行面向对象的编程时,程序员必须以此作为编程的参考,要充分实现设计文档的内容。程序员不能单方面地更改设计文档。在测试阶段,测试人员除了要进行面向对象的测试之外,尤其要测试软件与用户的交互是否和交互设计文档中所详细描述的过程一致,并且要检查软件的界面是否与交互设计文档对界面的描述保持一致。

### 3 结 论

软件的可用性是软件成功的重要因素,提高软件可用性是软件向来追求的目标之一。从软件建模角度来看,面向对象的软件开发技术对人机交互部分的设计过多关注于面向对象的抽象,其开发过程基本上是 use case 驱动的,通过分析阶段产生的 use case 来指导人机交互部分的设计,对交互部分的设计和实现也是基于建模和编程人员的逻辑思维,这样使设计出的软件的可用性无法保证。

交互设计从过程上为获得软件的可用性提供了一种

途径,以用户目标驱动功能的开发和界面的设计来满足用户的需求。文中提出了以目标驱动功能的交互的设计思路,并对交互设计师和建模人员的职责进行了探讨,使得交互设计和面向对象设计进行了较好的结合。由于交互设计师是代表用户的,其描述语言和图形都是用户易于理解的,而建模人员运用面向对象的观点对交互设计进行抽象和设计,在保证软件可用性的基础上又提高了软件的可修改性和可维护性。

### 参考文献:

- [1] 邵维忠,杨美清.面向对象的系统设计[M].北京:清华大学出版社,2003.
- [2] Cooper A. 软件开发的创新思维[M].刘瑞挺,刘强,程岩等译.北京:电子工业出版社,2003.
- [3] ISO 9241-11. Guidance on Usability[S]. 1998.
- [4] 孟祥旭,李学庆.人机交互技术:原理与应用[M].北京:清华大学出版社,2004.
- [5] Michael J A, Mazur B. Content and Complexity: Information Design in Technical Communication[M]. Erlbaum: [s. n.], 2003.
- [6] Gould J D, Boies S J, Lewis C. Making usable, useful, productivity-enhancing computer applications[J]. Communications of the ACM, 1991, 34(1): 74-85.
- [7] David R P, Alan M. Graphical User Interface Design and Evaluation[M]. London: Prentice Hall, 1995.
- [8] Dermot B. Structured User-interface Design for Interaction Optimisation[M]. New York, NY: Prentice Hall, 1993.
- [9] 董士海,王 坚,戴国忠,等.人机交互和多通道用户界面[M].北京:科学出版社,1999.

(上接第 192 页)

```
Course.xml");
OutputStreamWriter outWriter = new OutputStreamWriter(outStream);
((XmlDocument)doc).write(outWriter, "GB2312");
outWriter.close();
outStream.close();
```

此算法是根据扫描算法得来的,该算法在此不作详细介绍。DOM 的处理过程如下:首先,将相关的 XML 数据文档以 DOM 树的形式读入内存,在循环体内,从课程优先级最高值(10)开始扫描,逐个找出,然后将优先级高的课程排在效率较高的课时(此处效率由高到低分别是第一节(上午),第二节(上午),第三节(下午)),这样逐一进行排课,最后将向相应的课程的 XML 文档数据插入到所关联的教师当中。当以上信息自动执行完毕后,将存于内存中的教室、课程相关联的 XML 文档写入 roomCourse.xml 文件中,作为下一模块的信息基础。

### 4 结束语

文中以 XML 作为信息描述和存储的方法,并且通过

XML Schema 来定义相关的信息模型,在数据的处理方面采用了 XML DOM 作为信息抽取、修改的基本方法。这样使系统的信息描述精确,使系统的设计更加简洁。但该系统,由于 DOM 本身的特点决定,在操作大的信息文档时候,会造成内存的占用过大。所以,该系统在系统信息库中的数据不是很大的情况下会取得很好的效率。当信息库比较大的时候,可以考虑采用结合 SAX 的方法来提取、修改数据。

### 参考文献:

- [1] Birbeck M, Martin D. XML 高级编程[M].裴剑锋译.北京:机械工业出版社,2002.
- [2] Carlson D. Modeling XML Vocabularies with UML(Part I, II, III)[EB/OL]. <http://www.xmlmodeling.com>, 2001.
- [3] W3 Consortium. XML Schema Part 0: Primer, Recommendation[EB/OL]. <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>, 2001.
- [4] Hunter D, Watt A, Rafter J, et al. Beginning XML[EB/OL]. <http://www.wrox.com/>, 2004.
- [5] Harold E R. 用 Java 处理 XML(影印版)[M].北京:科学出版社,2004.