更新概要设计说明书

负责人：安靖
编写人：安靖
系统版本号：OrientX Version 1.5
完成时间：2004/09/08
开发单位：中国人民大学 IDKE 实验室 XML 工作组

1．引言

本说明书介绍 OrientX 系统中更新模块的设计。第 2 部分介绍模块的背景知识，模块的功能，以及此模块在整个系统中的地位和作用。第 3 部分介绍整个模块的处理思想，模块内部由哪些小模块构成，以及它们之间的关系。第 4 部分介绍本系统使用的更新语言。第 5 部分讲述模块内的数据结构的设计和主要成员函数的功能。最后第 6 部分介绍该模块的错误处理机制。

2．概述

背景知识：

数据库应该提供增删改等更新功能，允许用户更新数据库中的内容。目前更新关于 XML 数据更新的研究有：设计更新语言，定义 XML 数据的完整性约束，更新操作的约束检查等。这三方面目前都正在讨论阶段，还没有官方的标准，所以作者借鉴他人的研究成果，从可能的应用出发，针对本系统的具体情况，粗略地实现了对 XML 数据的更新。

OrientX2.0 通过对 XQuery 的扩展定义丰富易学易用的更新语言。针对越来越多的应用使用有模式约束的 XML 文档这一事实，提出了快捷先验性地检查 XML 更新约束的方法，能过滤不符合模式约束的更新请求。针对本系统的存储特征，实现 XML 文档的更新。

模块功能：

向用户提供更新语句，实现对 OrientX 数据库中存储的数据文档的增删改操作，包括增加、删除、修改元素、属性和值。增加的元素可以是同一文档中符合某些谓词的元素，也可以是其他文档中的元素。

此模块在整个系统中的地位和作用：

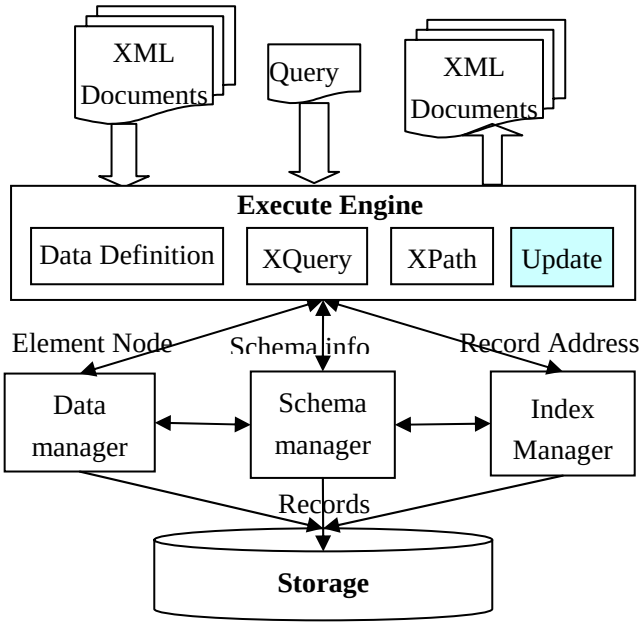
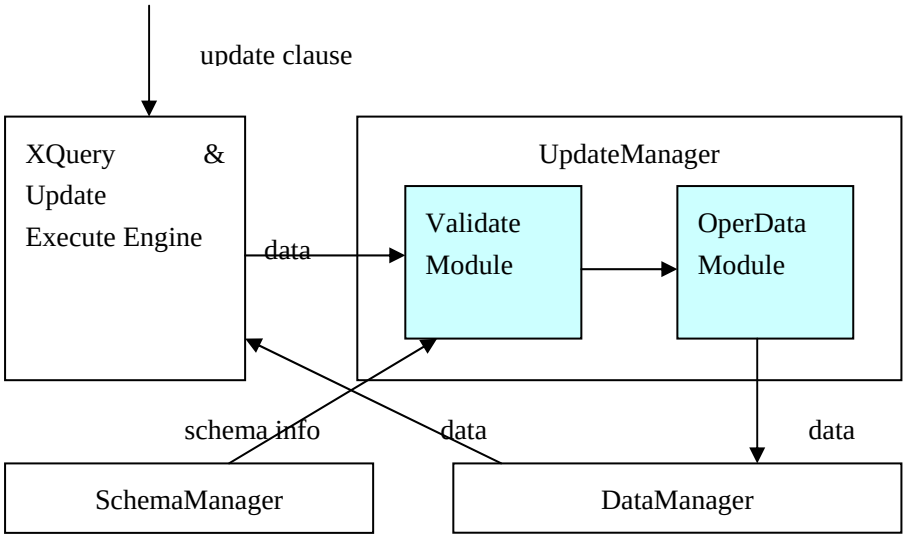


图 1　**系统框架图**

3. 总体设计

本模块包括更新操作的有效性检查模块，数据操作模块。



有效性检查模块

有效性检查模块负责检查更新请求的合法性，过滤不合法的更新请求，执行合法的更新请求。经过检查的有效的更新请求才会被继续处理，会被数据操作子

模块处理。

XML 文档分成有模式约束和无约束两种，更新操作执行完后的文档如果仍然符合模式约束的话，则此更新操作是合法的更新操作，否则是非法的。有效性检查的作用就是在更新操作执行前，判断该更新请求是否合法。有效性检查分两种：元素出现次数的检查和元素结构的检查。本系统提出了利用编码和模式树高效准确的进行更新的有效性检查的方法。

数据操作模块

4. 更新语言

OrientX2.0 系统使用的更新语句基于对 XQuery 的扩展，从总体上来说可以分成两部分，一部分是 fw(for-where)语句，用来定位要更新的元素；另一部分是更新子句，表述 insert、delete、update 的功能，即：

fw clause

insert/delete/update clause

对元素、属性、值进行增删改操作，一共有 9 种情况：

1	INSERT INTO/AFTER/BEFORE ele1 ELEMENT VALUES ele2	给指定的元素 ele1 增加新的子元素、兄弟元素 ele2
2	INSERT ele1 ATTRIBUTE attr-name VALUES attr-val	给指定的元素增加属性
3	INSERT ele1 TEXT VALUES text-val	给指定的元素增加值
4	DELETE ele1	删除元素
5	DELETE ele1 ATTIBUTE attr-name	删除元素的某个属性
6	DELETE ele1 TEXT	删除元素的值
7	UPDATE ele1 ELEMENT VALUES ele2	用元素 ele2 替换 ele1
8	UPDATE ele1 ATTRIBUTE attr-name VALUES attr-val	用 attr-val 替换属性 attr-name 的值
9	UPDATE ele1 TEXT VALUES text-val	用新值替换元素 ele1 的值

补充说明：

ele1 和 ele2 是 XPath 语句绑定的 element

ele2 可以是某个乘法、加法等数学函数，可以是 XML 格式的片断，可以是函数 datasource(“filename”)。如果要增加或者更改的元素比较简单，可以直接按照 XML 语法写在更新语句中，如果内容很多比较复杂，或者有现成的某个文档要全部插入进来，则用函数 datasource(“filename”)比较方便，filename 是上述的文档的名称。

5．数据结构设计

5．1 逻辑结构设计

主要的数据结构 CxqeUpdateNode

继承自	CxqeXAlgebraNode	
功能	更新语句中更新子句在内存中的表示，在分析更新语句时产生，是语法分析树的一部分	
数据成员	UpdateOp m_op	说明更新操作的类型，是 Insert、Delete 或 Update
	UpdateDir m_dir	Insert 和 Update 元素时说明插入元素相对于参考元素的方向关系：子元素、左兄弟或右兄弟
	CxqeXAlgebraNode* m_val	要更新的参考元素
	CxqeXAlgebraNode* m_expr	新插入的元素、属性值或元素值
	char* m_AttrName	更新属性时，指示要更新的属性名

5．2 物理结构设计

主要类成员函数的实现（算法）
模式有效性检查的算法：

数据操作的算法：
InsertElement()算法：

1）检查插入操作的有效性，如果是合法操作则转到 2），否则退出
2）生成新的子树 CopySubTreeForUpdate() 和 createEBRecord ()
3）存储新的子树 StoreSubTree()
4）重新存储父结点 StoreAgain()

DeleteElement()算法：

1) 检查删除操作的有效性，如果是合法操作则转到 2），否则退出
2) 删除指定元素，调用 DeleteNode()
3) 重新存储父记录

UpdateElement()算法

1) 检查更新操作的有效性，如果是合法操作则转到 2），否则退出
2) 生成新的子树 CopySubTreeForUpdate() 和 createEBRecord ()
3) 存储生成的新子树 StoreSubTree()

4) 删除指定元素，调用 DeleteNode() 5) 重新存储父记录

CopySubTreeForUpdate(DTDNodeCode& parentEID,void* dtdTree)算法

目的	得到本节点的一个拷贝，为 Insert 和 Update 提供新节点，根据插入位置确定拷贝节点的 EID	
整体说明	本函数是多态函数，ElementNode、RecordNode、CComputeElementNode、TextNode 重载 NxdbNode 类的同名函数。	
算法	NxdbNode	提供虚函数
	ElementNode	生成新元素，其 EID 或者和本元素的相同，或者根据函数参数确定； 拷贝属性 AID 和值，或者指定 AID； 递归调用同名多态函数，拷贝子节点； 如果本节点是 CComputeElementNode 子节点，则要生成相应的 RecordNode 作为本节点的父结点；
	RecordNode	如果对应的元素内容未读入到内存，则读入相应信息； 生成新记录，其 EID 或者和本元素的相同，或者根据函数参数确定； 递归调用同名多态函数，拷贝子节点；
	ComputeElementNode	生成新的元素和新的记录节点，其 EID 根据本节点的名称和函数参数确定； 递归调用同名多态函数，拷贝子节点；
	TextNode	得到一个一样的拷贝，包括值、类型、长度

createEBRecord(DTDTree* dtdTree,char* fileName,DTDNodeCode entryEid)算法

目的：	将 fileName 指定的 XML 文件中的所有节点生成子树，为 Insert 和 Update 提供新节点，根据插入位置确定拷贝节点的 EID	
总体说明	利用 SAX2 的 parser 生成内存中适合存储的数据格式 如果新节点在出现次数和结构上不符合模式约束，则抛出异常	
算法	startElement()	生成元素（包含属性）和记录，根据参数 dtdTree，entryEid 确定新节点的 EID
	endElement()	回到父结点
	characters ()	生成 text 节点

StoreSubTree()

为子树分配 oid，赋值 poid； 分配存储空间并存储 AssignSpace()

AssignSpace（）

目的	为子树分配存储空间，并存储到缓冲页	
总体说明	本函数是多态函数，	
算法	NxdbNode	提供虚函数
	ElementNode	递归调用同名虚函数
	RecordNode	打成字节流； 从文档尾寻找存储空间并保存记录； 将记录地址保存到 oid 表中； 递归调用同名虚函数

DeleteNode()算法

目的	从内存和缓冲页中删除元素	
总体说明	本函数是多态函数，ElementNode、RecordNode 重载 NxdbNode 类的同名函数。	
算法	NxdbNode	提供虚函数
	ElementNode	收回分配给它的 oid 值，删除子节点； 递归调用同名多态函数，删除子节点
	RecordNode	如果它的子节点没有读入内存，则扩展子节点； 删除缓冲页上的数据内容，即该记录对应的字符流； 递归调用同名多态函数，删除子节点； 删除内存（cache）中的相应内容，包括 element、attribute、text

6. 出错处理

当更新请求不符合模式约束的时候，会抛出异常，由上层捕捉