

分布式信息集成中 XUL-SQL 问题的研究

赵 媛¹, 吕 红¹, 杨阿桃², 王丽娜¹, 张 杰¹

(1 海军航空工程学院, 山东 烟台 264001; 2 贵州师范大学, 贵州 贵阳 550001)

摘 要: 在已构建的分布式信息集成平台中, 通过分析平台固有特点, 提出一种基于 XML 视图的远程信息修改策略 XUL-SQL, 并介绍了它的转换实质、转换算法, 最后给出了 XUL-SQL 在信息集成平台中的具体应用流程。

关键词: 分布式信息集成; XML; XUL; SQL; FLWU; XUpdate

中图分类号: TP393. 09

文献标识码: A

文章编号: 1000-7180(2011)08-0056-03

Research on the Problem of XUL-SQL in Distributed Information Integration

ZHAO Yuan¹, LV Hong¹, YANG A-tiao², WANG Li-na¹, ZHANG Jie¹

(1 Naval Aeronautical and Astronautical University, Yantai 264001, China;

2 Guizhou Normal University, Guiyang 550001, China)

Abstract: On the basis of analyzing the specialty of the distributed information integtation platform, a technique 'XUL-SQL' for updating the remote integrated information is proposed. Also, its transfer principle and mathematic are introduced. Finally, the concrete application process of XUL-SQL in information integrated platform is given.

Key words: distributed information integration; XML; XUL; SQL; FLWU; XUpdate

1 引言

在传统应用程序访问数据库的过程中, 例如应用程序服务器上的 Java 应用程序, 通过方法 DriverManager . getConnection (JDBC_URL, JDBC_USER, JDBC_PASSWORD) 来获得与远程 RDB 数据库的连接, 对于应用程序服务器上的 Java 应用程序, 可以说远程 RDB(Relational Database) 数据库是可见的, 即可以通过提交 SQL 语句来直接实现对于远程 RDB 的修改. 而在项目所搭建的基于 web service 的分布式信息集成平台中(图 1), 为解决各类异构的信息资源集成问题, 将信息从各类数据库中抽取出来包装成统一的 XML 信息进行发布. 当信息请求方访问信息时就可以得到 XML 格式的信息, 最终用户可以通过浏览器等客户端工具查看所请求的信息. 这里可以看到, 对于信息请求者来说,

远程 RDB 是不可见的, 因为信息在发布的同时, 信息描述细节也被封装, 只有 XML 视图可见. XUL (XML Update Language) 正是面向 XML 视图的修改语言, 而 SQL 是面向 RDB 的修改语言, 进而得出通过 XML 视图修改远程分布式 RDB, 就是要研究 XUL 到 SQL 转换的问题, 这正是本文所提出的 XUL-SQL.

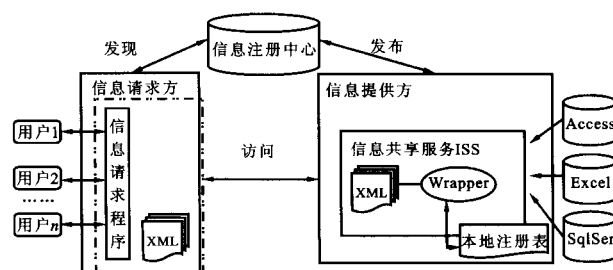


图 1 分布式信息集成平台的整体框架

收稿日期: 2011-05-11; 修回日期: 2011-06-23

基金项目: 海空航空工程学院基础研究基金(HYJC201020)

2 XUL

XUL 从逻辑方面能够表达用户修改 XML 的意图,但是要使其成为一种真正意义上的 XML 修改语言,仍然需要“语言化”的处理.借鉴 SQL 语言既能查询又能修改 RDB 这一成功范例,将 XUL 与 Xquery^[1]进行融合,则形成广义的 XQuery,扩展了 XQuery 在修改方面的功能.语法融合的意义^[2-3]是使 XQuery 既具有查询又具有修改的功能.当用户基于 XML 视图用 XQuery 操作底层 RDB 时,只需要进行一次语法解析,就像 SQL 在 RDB 上的执行一样,不需要对检索和修改分别进行语法解析.

XML 修改并不需要返回修改的结果,而是需要表达修改意图.而基于 FOR、LET、WHERE 语句的在 XML 文档中的节点定位、变量绑定、条件过滤是需要的.进行语法融合后的一般形式,如下:

```
FOR $ bindings IN XpathExp1,.....
LET $ bindings := XpathExp1,.....
WHERE Predacat1,.....
UPDATE(Operator1,.....)
```

其中,XML 修改的一般形式由表示节点定位的 FOR 子句、表示变量绑定的 LET 子句、表示条件过滤的 WHERE 子句以及表示修改操作的 UPDATE 子句组成,称之为 FLWU (FOR LET WHERE UPDATE)语句. Operator 表示修改意图,用 XUL 中的基本操作语句表示.

3 XUL-SQL 的转换实质

XUL^[4]到 SQL 转换的实质就是 FLWU 语句到 Select-From-Where 语句的转换问题.与 XQuery 到 SQL 的转换相比,两者的共同之处在于都需要根据路径表达式进行定位,确定 FLWU 语句操作的节点.路径表达式是通过一个 document 操作来首先获得指定文档的根节点,然后由导航操作“/”在该文档内部进行节点导航,在文件结构中进行路径匹配,再通过条件表达式进行过滤,“@”后面跟属性名字,表示要获取的特定属性的节点.两者不同之处在于 XUL 到 SQL 的转换需要改变 XML 文件、不需要加标记返回数据结果.

面向 XML 视图的查询转换成面向 RDB 的查询时,XQuery 查询语句 FLWR^[1]转换为 SQL 中的 Select-Where-From 语句.而面向 XML 视图的修改转换成面向 RDB 的修改时,XUL 修改语句 FLWU 转换为 SQL 中的 Update-Set-Where 语句,而 UP-

DATE 这种数据库的操作语言有个基本的特点是,一次只能操作一个表^[2],故目前本文的研究仅限于单源信息.

4 XUL-SQL 的转换算法

本文将“通过对 XML 视图的修改来修改远程的 RDB 数据库信息”引入到分布式信息集成中,设计时因为用户对于数据库信息的访问接口是以数据表的形式呈现的,各数据库中关系数据表的字段是固定的,呈现给用户时也是以表头的形式呈现的.所以在设计信息修改的用户接口时,只提供相应数据表字段值的修改操作,字段是不允许进行增加、删除、修改的.于是用户的修改,提交到信息请求方的信息修改程序,就不会对 XML 视图的 XMLSchema 产生变化^[5].也就是说,在信息请求方,XMLView()所接收的修改意图,只会修改 XML 的元素值,不会修改 XMLSchema.

toSQL()方法接收到的参数是<xul id="id 值">FLWU 语句</xul>,实际上 FLWU 到 SQL 的转换,即 For-Let-Where-Update 到 Update-Set-Where 的转换,主要是根据 FLWU 语句和本地注册表 Info_reg.xml 进行分析和处理.

读取<xul>标签中的 FLWU 语句进行处理,在 XML 文档中更新某个节点,FOR 子句的功能是定位节点,LET 子句的功能是变量绑定,WHERE 子句的功能是条件过滤,UPDATE 的功能是实现更新节点的操作. FLWU 操作的 XML 对象有可能是子节点,也可能是子树,而本文所提出的算法限制了 FLWU 操作的 XML 对象只可能是子节点,所以不考虑 XMLSchema 的变化.通过实验也证明,修改只是改变了 XML 内容,对于其文件结构是没有影响的.这里对于输入的定义^[6]是:输入为 FLWU 语句,结构如下:

```
FOR $p IN
document(r_id# t_id.xml)/root/element[@Col_key]
//变量绑定到要更新的 element 元素.
LET $q = $p/field //变量绑定到要更新的最小元素
UPDATE $p { UPDATEVALUES ( $q, "
fieldvalue" ) }
```

具体算法如下:

Step1: (RID, TID) = DecideInfoID(F), 即根据 FLWU 语句的 FOR 子句,能得到信息的 resource_id 和 table_id,分别标记为 RID、TID.

Step2: $T = \text{Red_reg}(\text{RID}, \text{TID})$, 根据 Step1 得到的 RID、TID, 来读本地注册表 Info_reg. xml, 进行匹配得到 $\langle \text{xmlfile_name} \rangle$ 的元素值, 为要进行修改的基本表名, 记为 T. 另外, 读 Info_reg. xml 还得到 T 对应数据源名称 $\langle \text{datasource} \rangle$ 、驱动 $\langle \text{jdbc} \rangle$, 以及信息的实际物理位置 $\langle \text{resource_phyloc} \rangle$, 分别保存, 用于后面和 RDB 的连接.

Step3: $\text{Col_keyValue} = \text{DecideTuple}(F)$, 由 FOR 子句, 得到 element 元素的属性, 根据前面的包装规范可以确定 element 元素的属性对应的就是基本表里的主键, 进而确定进行修改的 XML 节点在由 Step2 得到的基本表 T 中的主键及其值 Col_keyValue.

Step4: $(A, A') = \text{UpdateValues}(L, U)$, 根据 FLWU 语句的 LET 子句, UPDATE 子句, 来确定在基本表 T 中要修改的字段名 A, 以及修改后的字段值 A'.

Step5: $S = \text{DecideSets}(A, A')$, 根据 Step4 得到的字段名 A, 以及修改后的字段值 A' 来确定 SQL 语句中的 SET 语句, 记为 S.

Step6: $\text{DecideSQL}(T, \text{Col_key}, S)$, 即根据上述步骤得到的基本表 T, 要修改的元素的主键值 Col_key, 得到的 SET 语句 S, 来生成面向 RDB 的 SQL 语句, 即 Update T Set S Where Col_keyValue.

5 XUL-SQL 的应用

将 XUL-SQL 的修改策略应用于分布式信息集成平台的信息修改(图 2)中, 具体如下:

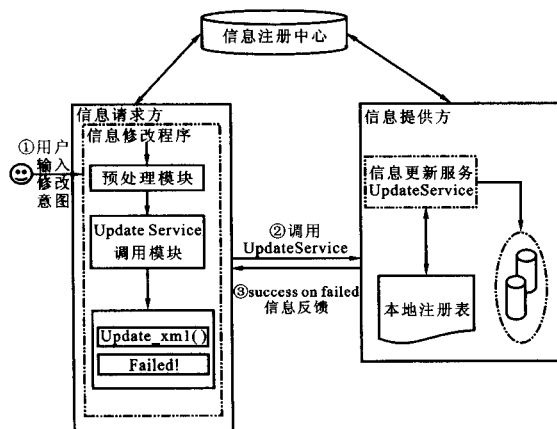


图 2 基于 XUL-SQL 的信息修改流程

①用户在信息提供方提供的基于浏览器的信息修改页面中输入其修改意图; 信息请求方接收到 HTTP 响应后, 启动信息修改程序, 信息修改程序由三个功能模块组成: 预处理模块、UpdateService

调用模块、结果处理模块.

预处理模块的功能是: 将接收到的用户修改意图, 转换成面向 XML 的 XUL 修改语言, 即 FLWU 语句序列, 将此序列加 $\langle \text{xul id} = \text{"id 值"} \rangle \langle / \text{xul} \rangle$ 标签对写入本地的 update. xml 文件, 并且将加标签后的语句序列作为字符串参数传递给 UpdateService 调用模块.

UpdateService 调用模块的功能是: 将预处理模块传递过来的参数, 即 FLWU 语句序列作为参数调用信息提供方的 UpdateService, 并且等待 UpdateService 执行完毕返回的 SOAP 消息结果.

结果处理模块的功能是: 根据 SOAP 消息中的 $\langle \text{xul id} = \text{"id 值"} \rangle \text{Success} \langle / \text{xul} \rangle$ 或者 $\langle \text{xul id} = \text{"id 值"} \rangle \text{Failed} \langle / \text{xul} \rangle$ 来处理本地的 XML 视图, 如果成功则根据 $\langle \text{xul} \rangle$ 的属性 id 读取保存在 update. xml 中的 FLWU 语句, 并且删除这个 FLWU 语句在 update. xml 中的记录, 执行对于 XML 视图的修改, 如果返回的是 Failed, 则根据返回的 $\langle \text{xul} \rangle$ 的 id, 直接去删除 update. xml 中的相关记录, 不对 XML 视图进行任何修改, 返回给用户“修改操作失败”的信息.

②UpdateService 所实现的功能是: 通过接收到信息提供方传递过来的 SOAP 消息, 得到服务调用参数 FLWU 语句, 同时读本地注册表, 进行要修改的 RDB 基本表的定位, 生成 SQL 语句, 执行对 RDB 的修改.

③UpdateService 执行结束, 返回信息请求方一个结果 SOAP 消息, 信息请求方进入结果处理模块.

6 结束语

本文对前期项目所构建的分布式信息集成平台, 进行了远程信息修改的研究, 提出针对单源信息的基础上抽取包装后的 XML 信息, 所以面向 XML 的 XUL、以及对 XML 节点的操作转换后涉及的数据表也单一, 而基于多源信息的复杂 FLWU 到 SQL 转换, 在以后工作中将做进一步探讨.

参考文献:

- [1] W3C. XQuery update facility & XQuery update facility use cases[EB/OL]. [2010-05-16]. <http://www.w3.org/XML/Query>.

(下转第 62 页)

间相差无几,此时 stops 主要通过 DBSCAN 算法得到;随着轨迹数增加,使用 DBSCAN 算法的时间增长很快,而使用 VR-tree 索引的运行时间增长相对较慢.由图 6 可以看出,使用 VR-tree 索引和 DBSCAN 算法发现的 stops 数量基本相等.

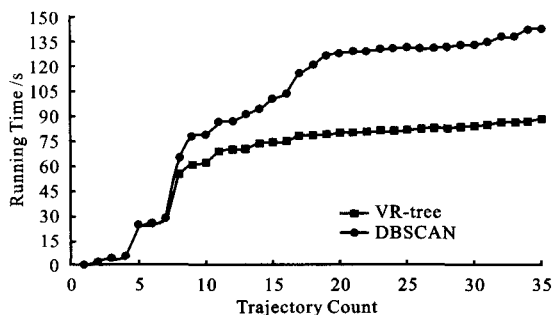


图 5 轨迹数量与运行时间的关系

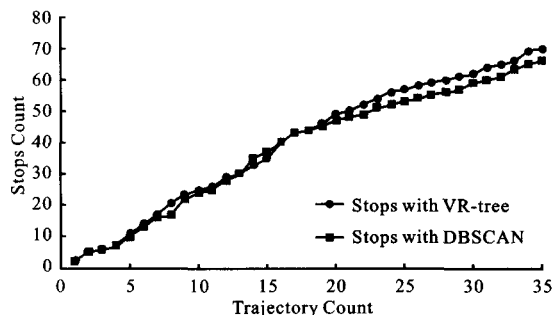


图 6 轨迹数量与生成 stops 个数关系

4 结束语

本文提出基于 VR-tree 索引结构的轨迹 stop 区域发现方法.首先选取一条轨迹使用 DBSCAN 算法发现 stops 并构建 VR-tree;然后对于其余轨迹,先在 VR-tree 中搜索,将与已处理轨迹 stop 区域相交且速度域相近的轨迹段从原轨迹中筛选;使用 DBSCAN 算法得到所有候选 stops;最后按照覆盖率大小合并候选 stops 得到最终 stops.使用麋鹿轨

迹数据进行实验,对本文提出的方法和使用 DBSCAN 算法挖掘轨迹 Stops 区域在运行效果和运行时间方面比较,可知本文方法不但挖掘 stops 效果比较准确,而且效率提高近 60%.

下一步将在此基础上,发现多轨迹的 stop 区域和轨迹间的关联关系,为用户提供个性化的服务.

参考文献:

- [1] Andrey Tietbohl, Vania Bogorny, Bart Kuijpers, et al. A clustering based approach for discovering interesting places in trajectories [C]//SAC 2008. Brazil: Ceara, 2008:863-868.
- [2] Yu Zheng, Xing Xie. Enabling smart location-based services by mining GPS traces [C]//CCCF. Canada, 2010.
- [3] Antonin Guttman. R-trees a dynamic index structure for spatial searching[C]// Proc ACM SIGMOD IntConf on Management of Data. Boston: 1984:47-57.
- [4] 何凯涛,唐宇,廖巍,等. 动态环境下移动对象索引技术研究[J]. 电子与信息学报,2008,30(10):2507-2511.
- [5] Saltenis S, Jensen C S, Leutenegger S, et al. Indexing the positions of continuously moving objects[C]// Proc. of the SIGMOD. USA, New York, 2000: 331-342.
- [6] Lee Jaegil, Han Jiawei, Whang KyuYoung. Trajectory clustering: a partition-and-group framework[C]// ACM SIGMOD 2007. China, Beijing, 2007:593-604.

作者简介:

李雯女,(1988-),硕士研究生.研究方向为知识发现与数据挖掘.

夏士雄男,(1961-),教授,博士生导师.研究方向为智能信息处理.

张磊男,(1977-),副教授.研究方向为语义与本体、数据挖掘.

(上接第 58 页)

- [2] Braganholo V, Heuser C, Vittori C. Updating relational databases though xml views[C]// Proceedings of Third International Conference on Information Integration and Web-based Applications & Services - IIWAS 2001. Austria, Linz,2001:85-94.
- [3] Sun Hongwei, Zhang Shusheng, Zhou Jingtao, et al, Constraints-preserving mapping algorithm from XML-schema to relational schema[J]. Lecture Note in Computer Science, 2002(2480):193-207.
- [4] Manolescu Ioana, Florescu Daniela, Kossmann Donald. Pushing XML queries inside relational databases[R].

Washington, Institute National of Research of Information and Automation,2001.

- [5] 赵圣猛,赵雷. 一种优化的 XML 文档模型映射方案[J]. 微电子学与计算机,2009,26(10):174-177.

- [6] 孙宏伟,张树生,周竞涛,等. 基于模型驱动的 XML 与数据库双向映像技术[J]. 计算机工程与应用,2002,38(4):25-27.

作者简介:

赵媛女,(1983-),硕士,工程师.研究方向为 Web Service、分布式计算.