第 34 卷 第 9 期 计 算 机 工 程 2008 年 5 月
Vol.34 No.9 Computer Engineering May 2008
·软件技术与数据库· 文章编号：1000—3428(2008)09—0073—03 文献标识码：A 中图分类号：TP391

XML 和 RDF 异构数据源的语义集成和检索

严玮峰，李生琦
(宁夏大学数学计算机学院，银川 750021)

摘 要：提出一种基于 Schema 的数据管理框架。该框架利用分层式的体系结构和全局视图(GAV)的集成方法，对分布式的异构数据源 XML 和 RDF 进行语集成。讨论了分层式体系结构的组成、映射过程和查询处理。实验结果表明了该框架的可行性。
关键词：语义集成；语义网；模式匹配

Semantic Integration and Retrieval of Heterogeneous XML and RDF Data Sources

YAN Wei-feng, LI Sheng-qi
(School of Mathematics and Computer Science, Ningxia University, Yinchuan 750021)

【Abstract】 This paper describes a schema-based data management framework which uses a layered architecture and a Global-As-View(GAV) approach to semantically integrate distributed heterogeneous XML and RDF data sources. Composition of the layered architecture, mapping process and query processing are discussed. The experimental results show that the framework is feasible.
【Key words】 semantic integration; semantic Web; schema matching

1 概述

Internet的飞速发展使网络成为信息传播和交换的重要手段。但是Web上信息的语义无法被计算机理解，语义网(semantic Web)应运而生，其层次结构[1]如图 1 所示。

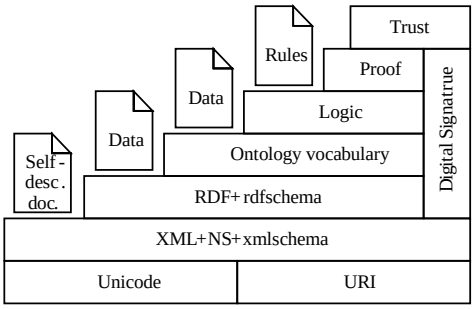

图 1 语义网的层次结构

语义网提出在Web内容上增加语义，使异构数据源具有交互性。虽然RDF和XML都能用来表示Web上的信息，但两者差异很大：RDF数据具有域结构(概念层次上的关系)，而XML数据具有文档结构(元素层次上的关系)[2]。Web环境下的异构数据源集成，特别是基于RDF/XML的数据集成问题，已成为数据集成研究领域及相其关领域的重要课题。

2 分层体系结构框架系统

异构数据源的语义集成通常采用概念模型(conceptual model)，例如E-R模型、本体模型。本文系统使用基于Schema的数据集成技术来集成异构的XML和RDF数据源，其设计采用混合型体系结构(hybrid architecture)和全局视图(Global-As-View, GAV)的方法[3]。

考虑到系统的自治性，数据库维护的便利性，还有异构数据源之间的交互性等诸多问题，为使此数据集成管理系统可以像传统的数据库那样管理自己的本地数据源，同时又能为网络中的其它结点用户系统提供服务和使用服务，本文提出了如下分层的体系结构，如图 2 所示。

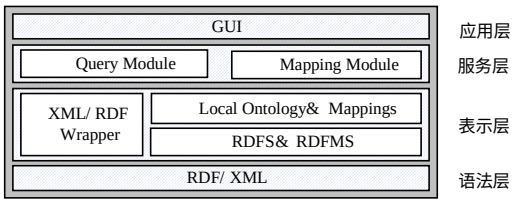

图 2 异构数据语义集成分层体系结构框架

该体系结构框架分为 4 层，每个下层功能的实现都是其上层功能实现的前提，它们相互依附，在每个自治节点内部达到了紧耦合的要求。该系统自上而下的结构如下：

(1)应用层(application layer)，提供统一的图形用户接口(Graphic User Interface, GUI)，为用户查询数据提供便利。在网络的每个自治节点上，用户直接通过这个统一的图形化接口来与本地数据源集成管理系统和网络上其他数据源的集成管理系统进行交互，使人机交互界面更友好。

(2)服务层(service layer)，包括查询模块(query module)和映射模块(mapping module)。通过查询模块，可以与网络上其他自治节点交互，形成一个基于网络的多数据库(multi-database)检索系统，实现在一个在相对广泛的数据源中进行查询的模型。通过映射模块，可以与网络中其他自治系统进行语义交互，建立一个混合型体系结构。

(3)表示层(representation layer)，通过包装器(XML/RDF

基金项目：国家自然科学基金资助项目(60663003)
作者简介：严玮峰(1981－)，男，硕士研究生，主研方向：信息集成，知识工程；李生琦，教授
收稿日期：2007-05-23 **E-mail：**gennyie@163.com

wrapper)，将本地数据源中的模式(schema)和数据(data)统一转化为局部本体。分别使用 RDFS(RDF Schema)和 RDFMS (RDF Mapping Schema)描述局部本体和映射。

(4)语法层(syntax layer)，位于系统底层，为局部本体和数据实例提供统一的语法，为语义交互打下基础。

综上所述，这种分层式的体系结构框架，为目前异构信息系统集成面临的异构性、分布性和自治性挑战[4]，提供了比较有效的解决方案，而且充分考虑了由Tim Berners-Lee提出的在语义网上实现异构数据源的语义交互性。

3 映射过程

Web中的每个节点所提供的数据源是XML或RDF。XML数据使用XML Schema来描述，而RDF数据的类和属性则使用RDF Schema来描述。局部本体和全局本体间的映射通过模式匹配(schema matching)[5]来建立。保护好RDF数据源的域结构(domain structure)和XML数据源的文档结构(document structure)是映射过程的关键。

3.1 局部本体的映射

本文方法使用 RDFS 将局部元数据(local metadata)表示成局部本体；关系型数据也用 RDFS 表示；表关系表示成 RDF 类；各表属性表示成 RDF 属性。在转换 XML 数据为 RDF 数据时，将复杂类型(含有嵌套子元素)的元素表示成 RDF 类，将简单类型(不含嵌套子元素)的元素和属性表示成 RDF 属性。分别将异构的数据源映射为 RDFS 表示的局部本体，如图 3～图 5 所示。

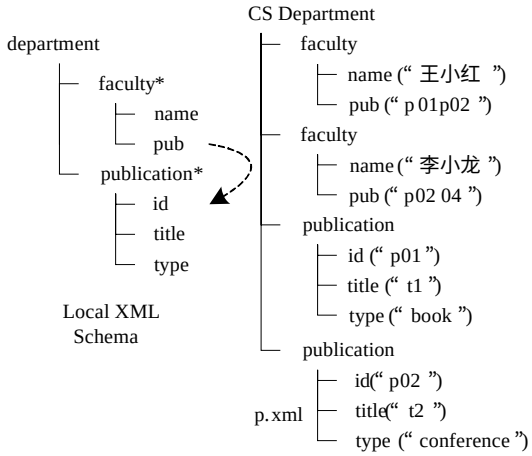

图 3 异构的数据源 *S*1(SML)

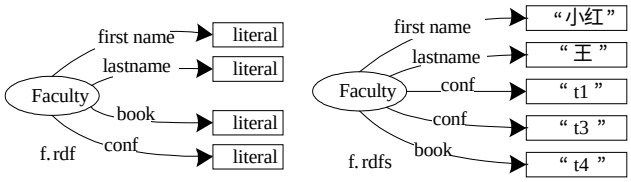

图 4 异构的数据源 *S*2(RDF)

Proceedings

pid	title	year
001	t1	2003
002	t2	2002

Author

aid	name	affiliation
001	李小龙	北京大学
002	王小红	清华大学

Author- proc

aid	pid
001	001
002	002
003	001

图 5 异构的数据源 *S*3(RDB)

分别将图 3～图 5 中的 *S*1, *S*2, *S*3 转换成用 RDFS 表示的局部本体，如图 6 所示。

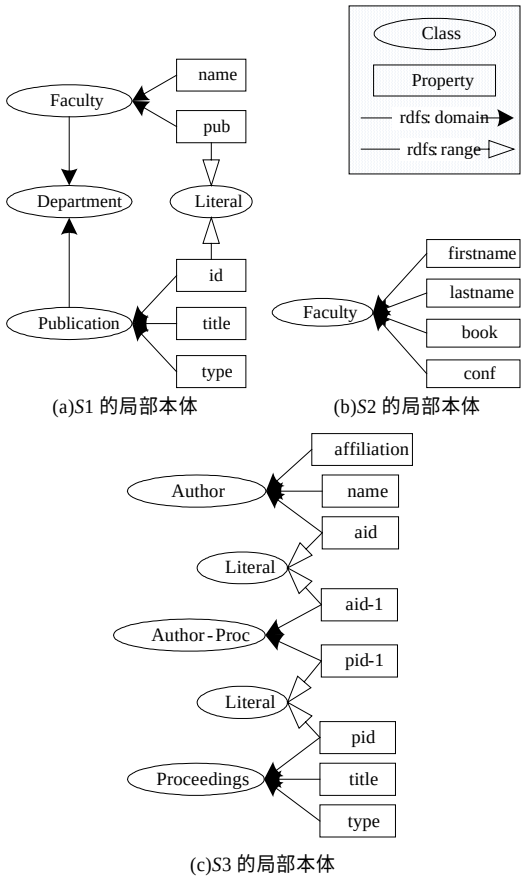

图 6 RDFS 描述的局部本体

3.2 全局本体的映射

通过模式匹配集成 RDFS 描述的各个局部本体，形成全局本体。模式匹配的功能是把全局本体 *G* 和分布的各个局部本体 *R* 作为输入，返回元素 *G* 和 *R* 之间的映射 *M* 作为输出。同时，全局本体通过从局部 RDF Schema 中归并或增加元数据得到更新。

RDF Schema 中的元素包括类和属性。用本地 RDF Schema 同全局 RDF ontology 匹配时，对于本地 RDF Schema 中的每个元素 P_L，如果在全局 ontology 中已经存在与之语义等价的元素 P_G，则这两个元素归并；否则，将元素 P_L 拷贝到全局 ontology 成为 P_G。映射表包含了局部 RDF 本体和全局 RDF 本体之间的映射信息。通常，如果全局本体 P_G 中的各个类、类的属性或它们之间的关系，由分布不同的数据源 P_i 和 P_j 通过模式匹配归并而得，则产生形如 (P_G, P_i, P_j) 的映射信息。如果全局 ontology P_G 中的类和属性是拷贝局部本体 P_i 而产生的，则产生形如 (P_G, P_i) 的映射信息。

4 查询处理

对XML数据源的查询，可用PXQuery(Partial XQuery)表示，它遵循XQuery[6]规则，是XQuery的一个子集，包含 4 个子句：for, let, where, return。而对RDF数据源的查询，可使用RDQL[7]，它采用类似SQL语法的表达，由以下子句构成：SELECT, FROM, WHERE, AND和USING。本文用三元组 (V_{Q^R}, V_{Q^W}, C_Q) 表示一个PXQuery的查询Q，其中，V_{Q^R} 和 V_{Q^W} 分别是包含在return子句和where子句中所有XML路径表达式的集合；C_Q 是一些限制，其项由形如 vRc 的表达式构成，$v \in V_{Q^W}$，R是一个比较操作(例如：$=, \prec, \succ, \leq, \geq, \neq$)，c代表常

量。用三元组 (P_{Q^S}, P_{Q^W}, C_Q) 表示一个RDQL的查询，其中，P_{Q^S} 和 P_{Q^W} 是两个集合，分别是SELECT子句和WHERE子句中所有RDF路径表达式的集合。

4.1 源查询的分析与转换

将分析源查询从字符串类型转化为三元组，如果在 XML 源上，则使用 PXQuery 查询，将它转换成三元组 $(V_{Q_{in}^R}, V_{Q_{in}^W}, C_{Q_{in}})$；如果在 RDF 源上，则使用 RDQL 查询表达式，将它转换成三元组 $(P_{Q_{in}^S}, P_{Q_{in}^W}, C_{Q_{in}})$。因为两者的查询过程相似，所以只须进行相应变换就能得到。

4.2 源查询的分解

使用查询重写算法 RDQL2RDQL 或 RDQL2PXQuery，将用户源查询重写为各个异构数据源上的目标子查询，这要用到生成的映射表信息。查询重写可以用函数表达式 $Q_2 = f(Q_1, M)$ 来表示，其中，Q_1 是源查询；M 是映射信息，通过查询重写算法 $f(Q_1, M)$ 就能生成需要的目标子查询。

例如，查找由作者“*a2*”出版的所有图书，代码如下：

```
SELECT ?title
WHERE (?book,<go:title>,?title), //go 代表全局本体的命名空间
       (?book,<rdfx:contains>,?author),
       (?author,<go:name>,?name)
AND (?name eq “a2”)
```

首先，可得到：

$$P_{Q_{in}^S} = \{\text{Book.title}\}$$

$$P_{Q_{in}^W} = \{\text{Book}, \text{Book.title}, \text{Author}, \text{Author.name}\}$$

$$C_{Q_{in}} = \{\text{Author.name}, \text{eq}, \text{"a2"}\}$$

再使用映射表信息 M 更新源查询为语义等价的三元组，如下：

$$P_{Q_{in}^S} = \{\text{Article.title}\}$$

$$P_{Q_{in}^W} = \{\text{Article}, \text{Article.title}, \text{Writer}, \text{Writer}, \text{Wholename}\}$$

$$C_{Q_{in}} = \{\text{Writer.wholename}, \text{eq}, \text{"a2"}\}$$

然后，将源查询重写为各异构数据源上的目标子查询，如下：

```
SELECT ?title
WHERE (?article,<lo:title>,?title), //lo 代表局部本体的命名空间
       (?article,<rdfx:contain>,?writer),
       (?writer,<lo:wholename>,?wholename)
AND (?wholename eq “a2”)
```

4.3 返回的查询结果集成

通过集成本地查询结果和远程查询返回的查询结果，可得到最终的查询结果。这不仅要在移除相似记录的同时，联合来自不同数据源的查询结果，还要使用某些关键属性来关联查询记录，实现查询过程的语义集成。

5 结束语

本文提出了一种集成异构 XML 和 RDF 数据源的方法，使用分层式的体系结构框架，为异构信息系统当前面临的挑战提供了一种解决方案，使异构数据源具有语义交互性。

参考文献

[1] Berners-Lee T. SemanticWeb[EB/OL]. (2000-12-06). http://www.w3.org/2000/talks/1206-xml2k-Tim.tbl/.XML2000.

[2] Halevy A Y. Piazza: Data Management Infrastructure for Semantic Web Applications[C]//Proceedings of the 12th International World Wide Web Conference. [S. l.]: IEEE Press, 2003: 556-567.

[3] Lenzerini M. Data Integration: A Theoretical Perspective[C]// Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. [S. l.]: ACM Press, 2003: 233-246.

[4] 张付志. 信息集成技术在数字图书馆中的应用研究[J]. 计算机工程, 2005, 31(7): 90-92.

[5] Rahm E, Bernstein P A. A Survey of Approaches to Automatic Schema Matching[J].VLDB Journal, 2001, 10(4): 334-350.

[6] Boag S, Chamberlin D, Fernández M F, et al. XQuery 1.0: An XML Query Language[EB/OL]. (2003-12-12). http://www.w3.org/TR/xquery/W3C Working Draft.

[7] HP Labs. RDQL-RDF Data Query Language[EB/OL]. (2003-12-05). http://www.hpl.hp.com/semweb/rdql.htm.

（上接第 72 页）

阈值 *B-threshold* 越高，则召回率 *Recall* 越低，因为阈值降低虽然可能误判一些正常邮件为敏感邮件类别，但减少了漏判的概率。在贝叶斯分类模块，设置 *B-threshold*=0.4，并将基于中文分词的普通贝叶斯算法与基于数据库查询的贝叶斯算法的最终结果进行了比较，如表 5 所示。

表 5 结果比较

	训练时间/s	测试时间/s	召回率	准确率
基于中文分词的贝叶斯分类	5	2 329	0.926	0.941
基于数据库查询的贝叶斯分类	5	478	0.930	0.987

结果显示，与基于分词的分类方法相比，本文提出的分类方法在效率上优势十分明显，召回率相当但准确率略高，更加适合对海量邮件的处理和分析，完全能够满足实际需求。

6 结束语

结合某安全部门的现实需求，本文研究了针对解析后存储在数据库中的海量邮件数据的敏感类别分类技术，提出一种基于数据库编程语言的分类方法，结合 ORACLE PL/SQL 存储过程与贝叶斯算法对邮件进行分类处理。实验结果表明，该方法对海量数据的处理有很高的分类效率和较高的准确率。该方法也同样适用于垃圾邮件的过滤技术，对大量邮件进行自动分类和转发，有效减少了系统分发邮件的工作量。

参考文献

[1] 张培颖, 李村合. 一种中文分词词典新机制——四字哈希机制[J]. 微型电脑应用, 2006, 22(10): 35-36.

[2] Shaffer C A. 数据结构与算法分析[M]. 张 铭, 刘晓丹, 译. 北京: 电子工业出版社, 2001.

[3] 周志军. 中文邮件分类系统的研究及其实现[D]. 苏州: 苏州大学, 2005-04.

[4] 郑 刚, 彭 宏, 郑启伦. 存储过程在嵌入式多功能数据挖掘器中的应用[J]. 计算机应用, 2006, 26(6): 102-104.

[5] 谈竹贤, 王 毅, 赵景亮, 等. Oracle 9i PL/SQL 从入门到精通[M]. 北京: 中国水利水电出版社, 2002-02.

[6] 高朝瑞. GKD-Base PL/SQL 存储过程和包的研究与实现[D]. 长沙: 国防科技大学, 2004.

[7] 毛国君, 段立娟, 王 实. 数据挖掘原理与算法[M]. 北京: 清华大学出版社, 2005.