

利用 D2R 实现关系数据库与关联数据的语义模式映射 *

白海燕 梁冰

(中国科学技术信息研究所 北京 100038)

【摘要】关系型数据库和关联数据的概念模型都基于现实世界的实体、属性及其关系而构建，具备建立映射和实现转换的可能性。两者语义模式映射的重点是关联关系的构建与表达。开源软件 D2R 的映射语言，通过核心要素 ClassMap、PropertyBridge 及其属性，支持关系数据库的 SQL 实现，支持将关系数据库不同实体之间、同一实体内部及与外部数据的关联关系构建为 RDF 链接。

【关键词】D2R 关联数据 语义模式映射

【分类号】G250

Semantic Pattern Mapping Between RDBMS and Linked Data Based on Open Source Software

Bai Haiyan Liang Bing

(Institute of Scientific & Technical Information of China, Beijing 100038, China)

【Abstract】The concept model of RDBMS and linked data builds on basement of real world entity, property and their relationships. So it is possible to build mapping between them. The core of semantic pattern mapping is to construct and express the linking relationships. The language of open source software D2R supports to execute SQL of RDBMS and transfers relationships between different entities, inside same entity and among outside data sets into RDF linkage through core language element ClassMap, PropertyBridge and their properties.

【Keywords】D2R Linked data Semantic pattern mapping

1 引言

关系型数据库管理系统(RDBMS)是计算机数据管理发展史上一个重要的里程碑，具有数据结构化程度高、冗余度低、较高的程序与数据独立性、易于扩充、易于编制应用程序等优点。自 20 世纪 80 年代以来，关系型数据库的商用产品进入了大规模普及应用的成熟期，时至今日，绝大多数的信息服务和信息系统仍以关系型数据库为技术和服务基础，存储和管理了海量的结构化信息^①。

近年来，随着语义网技术的发展，特别是关联数据运动的推动，出现了众多的开源软件，如 Drupal^②、D2R^③、

收稿日期：2011-05-03

修改稿日期：2011-06-13

* 本文系国家社会科学基金“图书馆资源组织培文化研究”(项目编号：10CTQ009)的研究成果之一。

① http://drupal.org/project/linked_data

② <http://sites.wiwiss.fu-berlin.de/suhl/bizer/d2r-server/index.html>

Virtuoso¹、Triplify²等,致力于将关系型数据库的结构化数据转换为具有语义链接能力的关联数据。这些软件工具的通用工作模式是将对关联数据的请求,包括URI(Uniform Resource Identifier)参引、SPARQL查询或DUMP下载等,转换为RDBMS的SQL语言,并将SQL查询结果转换为RDF三元组。而这类软件工具的核心是实现关系数据库与关联数据的语义模式转换。

关系型数据库的组织模式基础是二维逻辑表,而RDF三元组是用最简单的二元关系来表达各种复杂的关系。虽然两者的语义组织模式并不相同,但其概念模型都是基于现实世界的实体、属性及其关系而构建的,因此具备建立映射和实现转换的可能性。

本文以目前广为流行的、专门用于关系数据库的关联数据发布开源软件D2R Server为例,在归纳分析语义组织模式转换关键需求的基础上,重点介绍了以D2RQ Mapping语言为基础的语义映射和模式转换实现,意义在于为相关应用提供可借鉴的方法,为评价其他开源软件的转换和映射能力提供依据,促进关联数据的发布与构建。

2 RDBMS与Linked Data的组织模式

将现实世界中的具体事物抽象、组织为某一数据模型,是将现实世界抽象为信息世界,进而转换为机器世界的必然过程。信息世界中涉及的基本概念主要包括:

(1) 实体:客观存在并可相互区别的事物。实体可以是具体的人、事、物,也可以是抽象的概念或联系;

(2) 属性:实体所具有的某一特性。一个实体可以由若干个属性来描述;

(3) 关系:在现实世界中,事务内部以及事务之间是有联系的,这些联系在信息世界中反映为实体内部的联系和实体之间的联系。实体内部的联系通常是指组成实体的各属性之间的联系^[1]。

2.1 RDBMS的关系模型

关系模型是关系数据库的数据模型,在关系模型中,无论是实体还是实体之间的关系均由单一的结构关系,即二维表来表示。每一个表由行和列组成,表的每一行为一个元组,每一列为一个属性。在实际应用中,通常将每一个实体设计为一个表,例如:5个表分别存储和管理期刊、期刊论文、作者、专利、主题词5

类不同的实体。每个表的列,代表了这个实体的属性特征,如期刊的属性特征包括题名、ISSN号、出版者等,作者的属性特征包括姓名、专业、单位等。属性的值只有相应的数据类型和取值范围。

客观世界中,事务内部及事务之间存在联系,表现为实体内部的联系和实体之间的联系,实体内部的联系通常是指组成实体的各属性之间的联系,实体之间的联系可以分三种:一对联系,一对多联系和多对多联系。实际上,一对一联系是一对多联系的特例,而一对多联系又是多对多联系的特例,这些联系不仅存在于两个实体之间,也存在于两个以上的实体之间^[1]。

2.2 关联数据的一阶谓词模型

关联数据的基本思想是:

- (1) 用URI来标识事物;
- (2) 一切万维网上标识的事物,具体的或抽象的,统称为“资源”;
- (3) 用属性和属性值描述资源以及资源与资源之间的关系^[2]。

关联数据的数据模型是RDF一阶谓词(First-order Logic)模型。RDF模型可以用三元组(Triple)的方法来表示,即主语(Subject)、谓语(Predicate)、客体(Object)构成,称为RDF陈述(Statement)。如果把主语和客体看作节点,属性看成是一条边,则一个简单的RDF陈述可以表示成一个RDF有向图(Graph)^[3]。RDF实质上是一种二元关系的表达,由于任何复杂的关系都可以分解为多个简单的二元关系,属性和属性值类似于E-R模型,因此RDF数据模型可以描述任何复杂的关系。实际上,属性和属性值都可以包含URI,也同样是资源,通过URI可以参引任何可以被标识的事物,因此,RDF可以链接到万维网上的任何地方,描述万维网上的一切资源。在关联数据中,强调通过丰富的RDF链接,构建资源的“语境”^[4,5]。

RDF使用特定命名和值来表达与资源有关的简单声明。但很多情况下,用户希望能够自定义一些词汇,然后用这些词汇来描述资源,即需要定义一些类和特性,例如定义Person类来描述人,定义Book类来描述图书,定义Author特性来描述图书的作者等。RDF本身并不能定义这些类和特性,它们需要用RDF的描述

¹ <http://virtuoso.openlinksw.com/wiki/main/Main>

² <http://triply.org/Overview>

语言 RDF Schema 来定义¹

2.3 两者的映射关系及转换要求

两者概念模型都是基于现实世界的实体、属性及其关系而构建的，因此具备建立映射和实现转换的可能性。在 RDBMS 与 RDF 的映射实现研究方面，Imai 等²提出了新型的元数据类型——关系元数据，即基于 ER 实体关系模式的 XML 数据，用于实现了 RDF 相兼容的目的；Korotkiy 等³使用 RDF/SQL 作为将关系数据与预定义领域本体映射的定义和实现机制，提出了一种映射实现的方法；Yanhui 等⁴通过抽取关系模型的语义并映射到 RDF 模型，提出了一种具备灵活性的映射机制。关系模型和三元组之间的映射转换关系可描述为概念映射和数据映射两个方面，具体如表 1 所示：

表 1 关系模型与三元组之间的映射关系

| 概念映射 | | 数据映射 | |
|-------------|-------|-------------|-----------|
| Linked Data | RDBMS | Linked Data | RDBMS |
| 类 | 个体 | 个体、客体 | 表 |
| 数据属性 | 个体属性 | 谓词 - > 本值 | 列 - > 列 |
| 对象属性 | 个体关系 | 谓词 - > 客体 | 关系行 - > 行 |

通过表 1 可以看出，在语义模式的映射中，类、数据属性与实体、实体关系的映射比较清晰。类与实体的映射体现为表与 RDF 类的映射，即表映射为 RDF 三元组中的主体或客体，表的主键映射为主语或对象的 URI；数据属性的映射，即二维表的列对应为三元组中的谓词，其行数据映射为文本对象。

而对象属性的映射，则相对比较复杂，原因在于个体内部及个体之间的关系，在关系数据库中有不同的表达和构建方式，并且基于实际需求，有一定的应用设计特点。而在关联数据中，对象之间的语义关系通过 RDF 链接来表达，因此，对象属性的映射，是关系数据库与关联数据语义组织模式映射的关键。

基于数据组织的实际应用，本文分析和归纳了关系数据库中关联关系的不同构建方式，确定了核心的语义映射和模式转换需求。

(1) 不同表之间的关联关系

不同的表代表不同的实体，而表间关系即实体之间的关系，需要在关联数据的 RDF 中，通过关联链接来进行表达。通常在关系数据库中，不同表之间的关系通过以下几种方式进行构建：

¹ 基于外键引用的表间关系，即一个表中的非主属性，来自于另一个表的主属性。例如期刊论文表中，期刊论文的

subject 属性，引自主题词表的 subject_id，期刊论文的母体即 journal 属性，引自期刊表的 journal_id，对这些数据进行关联数据发布时，应表达这些在数据库中显性构建的关联关系。

² 基于属性关联的表间关系，即一个表中的属性，基于一定的限制条件，与另一个表的相关属性相等。例如期刊论文表和专利表，都有 author 属性，表明两者存在一定的关系，当使用 SQL 语句，select * from 期刊论文表, 专利表 where 期刊论文表 author = 专利表 author，则可以得到期刊论文与专利文献基于相同作者的关联关系，同时，可增加限制条件，在满足一定条件下，动态获得某些关联关系。对这些数据进行关联数据发布时，应显性表达这些隐含在数据中的关联关系。

(2) 同一表内部的关联关系

³ 一个表及其列代表一个实体及其属性之间存在的一对多的关系，即表 1 中的数据属性及其实现。例如期刊表与其诸多列之间，存在一对多的内在联系，对这些数据进行关联数据发布时，应表达这些在数据库中显性构建的关联关系。

⁴ 一个表及其别名表，通过某一属性的等值关系，可构建两个不同实体的逻辑关系。例如一个主题表中，每一个主题词 subjectid，都有一个上级主题词 parentid，通过 subjectid = parentid 可以构造出主题词及其上级主题词的等级关系。这种关系可以通过 SQL 语句中别名的运用，将一个表逻辑上拆分为两个表，基于属性相等的条件，得到两个虚拟表的关联关系，从而获得了同一表（主题表）内部，主题及其上位主题的等级关系。对这些数据进行关联数据发布时，应显性表达这些隐含在数据中的关联关系。

(3) 与外部数据的关系

在将关系型数据库转换为关联数据的过程中，如已知与外部数据（如 RDF 词表、某些常量等）存在某些关系（如 rdf:seeAlso, owl:sameAs 关系等），则需要与已知的 URL 建立关联关系。

3 D2R Server 的语义模式映射

3.1 语义模式映射机制

D2R 是一个将关系型数据库发布为关联数据的专业工具，支持主流关系型数据库如 Oracle、MySQL、PostgreSQL、Microsoft SQL Server、Microsoft Access 等，将表数据发布为关联数据。目前使用 D2R 进行关联数据发布的应用有：Gene Ontology Annotations¹、Berlin DBLP Bibliography Server²、Hannover DBLP Bibliography Server³。

¹ <http://spade.lbl.gov:2020>

² <http://www4.wiwiiss.fu-berlin.de/dblp/>

³ <http://dblp13s.de/d2r>

EuroStat Countries and Regions Server¹ 等^[6]。

D2R 语义模式映射机制主要包括两大部分：

(1) 构建映射规则，即利用映射语言，以 RDF 格式描述映射关系，形成映射文件，该文件的作用是在访问关系型数据时将 RDF 数据的查询语言 SPARQL 转换为关系数据库的 SQL 数据查询式，并将 SQL 查询结果转换为 RDF 元组，该部分的实现脚本为 generate.bat。

(2) 构建关联数据服务，使用映射文件对关系型数据进行转换，并提供多种访问模式，包括关联数据的 URI 参引、SPARQL 查询和 DUMP 下载，其生成脚本为 d2r-server.bat。

基于映射语言编写的映射文件是 D2R 实现语义模式映射的核心，如图 1 所示。

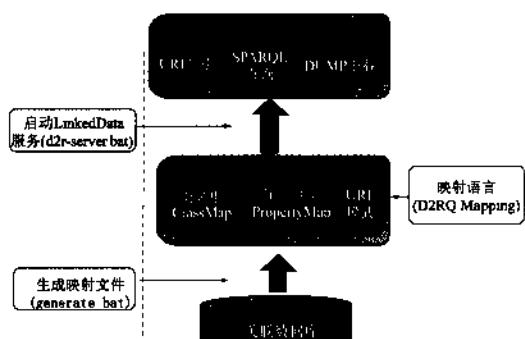


图 1 D2R 的语义模式映射机制

本文利用 SQL Server 建立了一个学术文献数据库，针对期刊、期刊论文、专利等不同类型文献，分别设计了相应的期刊（Journal）、期刊论文（JournalPaper）、专利（Patent）等数据表，并建立了与之相关的作者（Author）、主题词（Subject）等数据表。通过运行 D2R 的 generate.bat 脚本，生成了一个映射文件 outfile.n3；基于该映射文件，构建了学术文献数据库的关联数据的浏览与查询服务。

3.2 语义映射语言

D2RQ Mapping 是 D2R 的映射语言，作为一种声明性语言，用于描述关系型数据模式与 RDFS Vocabularies 或者 OWL 的映射关系。每一个由 generate.bat 脚本生成的映射文件，都是用 D2RQ 语言编写的 RDF 文件。这种语言由 D2RQ RDFS Schema² 定义。D2RQ 的命名空间为 <http://www.wiwiss.fu-berlin.de/suhl-bizer/D2RQ/0.1#>，该语言的语义映射实现有两大核心。

(1) URI 模式定义

在关联数据中，任何一个资源都使用 HTTP URI 来标识，以通过 HTTP 协议进行访问，实现基于 Web 的访问与互联^{[10][11]}。

D2RQ 映射语言提供了 URI 生成的描述模板机制，用来指导生成实例化资源的真实 URI，具体包括 4 种方式：绝对 URI 模式（URI Pattern），相对 URI 模式（Relative URI Pattern），URI 列模式（URI Columns Pattern）和空节点模式（Blank Nodes Pattern），具体介绍见文献[12]和文献[14]。

本文采用绝对 URI 模式，即使用 d2rq:uriPattern 属性对每个表对象进行配置，形如 JournalPaper @ @ JournalPaper_PAPER_ID@ @，“@前面的部分为表名，后面“@ @”之间的部分为表的列

(2) 类与属性的映射

类与属性的部分映射编码如下：

1 表 JournalPaper 的类映射

```

<file> D: d2r - server outfile n3#JournalPaper > a d2rq:ClassMap;
d2rq: dataStorage map database,
d2rq:uriPattern "JournalPaper" @ @ "JournalPaper_PAPER_ID" @ @",
d2rq: class < http://localhost:2020/vocab/resource/JournalPaper >,
d2rq: classDefinitionLabel "JournalPaper" .,
```

```

<file> D: d2r - server outfile n3#JournalPaper - label > a d2rq:PropertyBridge;
```

```

d2rq: belongsToClassMap < file> D: d2r - server outfile n3#JournalPaper >;
```

```

d2rq: property rdfs:label,
```

```

d2rq: pattern "JournalPaper# @ @ JournalPaper_PAPER_ID" @ @ ",
```

```

map: JournalPaper_PAPER_ID a d2rq:PropertyBridge,
```

```

d2rq: belongsToClassMap < file> D: d2r - server outfile n3#
```

```

JournalPaper >,
```

```

d2rq: property vocab:JournalPaper_PAPER_ID,
```

```

d2rq: propertyDefinitionLabel "JournalPaper_PAPER_ID",
```

```

d2rq: column "JournalPaper_PAPER_ID",
```

```

d2rq: datatype xsd:int,
```

```

map: JournalPaper_TITLE a d2rq:PropertyBridge,
```

```

-----
```

```

1 http://www.wiwiss.fu-berlin.de/eurostat
```

```

2 http://www.wiwiss.fu-berlin.de/bizer/d2rq/0.1
```

```

d2rq: belongsToClassMap < file:D/d2r-server.outfile.n3#
    JournalPaper>,
d2rq: property vocab:JournalPaper_TITLE,
d2rq: propertyDefinitionLabel "JournalPaper TITLE",
d2rq: column "JournalPaper TITLE",
map: JournalPaper_Language a d2rq: PropertyBridge,
d2rq: belongsToClassMap < file:D/d2r-server.outfile.n3#
    JournalPaper>,
d2rq: property vocab: JournalPaper_Language,
d2rq: propertyDefinitionLabel "JournalPaper Language",
d2rq: column "JournalPaper Language";

```

③ 外键引用的关联构建

```

map: JournalPaper_MotherJournal a d2rq: PropertyBridge;
d2rq: belongsToClassMap < file:D/d2r-server.outfile.n3#
    JournalPaper>,
d2rq: property vocab: JournalPaper_MotherJournal,
d2rq: refersToClassMap < file:D/d2r-server.outfile.n3#
    Journal>,
d2rq: join "JournalPaper_MotherJournal => Journal_Journal_ID",

```

类映射(d2rq, ClassMaps)中的 ClassMap 代表一个类或本体中相似的一组类,它与关系数据库中的一个表相映射。类映射有两个重要属性 d2rq: Class 和 d2rq: UriPattern。URI 模式下,d2rq: Class 表示当前的类映射所对应的 Class,其取值可以来自现有的 OWL Ontology 或者 RDFS Schema,也可以根据数据特征定义新的 Class。每一个类映射组成了数据库的映射,如 JournalPaper_ClassMap 和 Journal_ClassMap 等,其中 JournalPaper_ClassMap 的映射编码见映射编码[1][12-14]。

属性映射(d2rq: PropertyBridge)代表 OWL Ontology 或者 RDFS Schema 中 Class 的属性,它与关系数据库中某个表的列相对应。本实例中,JournalPaper_ClassMap 包括了 PaperID_PropertyBridge、Title_PropertyBridge 和 Language_PropertyBridge 三个属性映射,分别映射三个列,其映射编码见映射编码[2]。属性映射可概括为几个基本属性:d2rq: belongsToClassMap 表示该 PropertyBridge 所属的 ClassMap;d2rq: property 表示该 PropertyBridge 所对应 property,其值可以来自现有的 OWL Ontology 或者 RDFS Schema,也可以根据数据特征定义新的 property;d2rq: column 表示该 PropertyBridge 关联的某个具体表中的列[12-14]。

本实例的映射结构^[12-15]如图 2 所示:

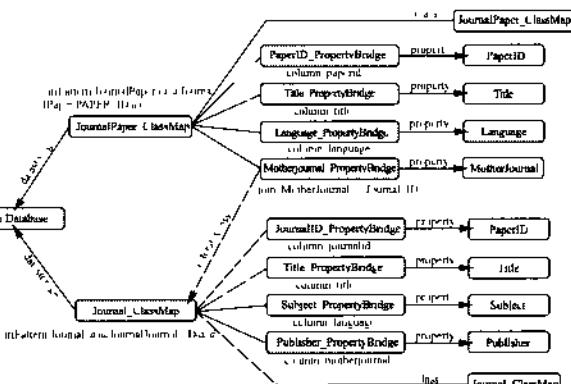


图 2 表映射与属性映射示意图^[12-15]

3.3 关联关系的构建与映射实现

(1) 不同表间关联关系的构建

在图 2 中,期刊论文表的母体属性 MotherJournal,引用了期刊表中的 JournalID。D2RQ 映射会自动为数据库中已构建的显性关系,如基于主键和外键引用的表间关系,建立链接,从而在关联数据浏览中提供跳转链接,如图 3 所示:

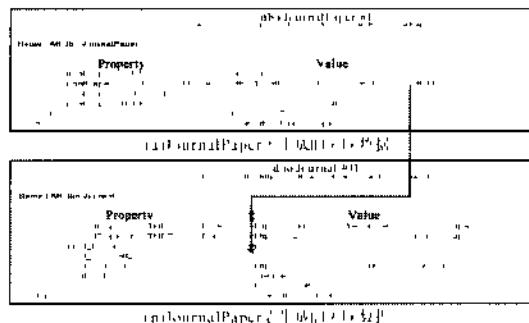


图 3 不同表间关系在关联数据中生成的 RDF 链接

图 3 中,(a)是由表 JournalPaper 生成的关联数据,其中包括指向 Journal 的关联数据(b)的链接。这是通过属性映射 d2rq: PropertyBridge 的相关属性 d2rq:refersToClassMap 和 d2rq:join 来实现的。参见映射编码[3],d2rq:refersToClassMap 说明了关联的对象引用自哪个类,d2rq:join 则描述了关联关系,两个引用字段之间的“=>”符号代表了引用的方向。

如果数据库没有显性地建立这种参引关系,而是通过 SQL 语句动态获得表间关系,则需要手工进行支

持 SQL 语句的映射描述,即手工在每个表的 ClassMap 的 PropertyBridge 增加 join 属性。例如当需要在关联数据中揭示期刊论文和专利文献基于相同作者的关联关系时,可使用 d2rq:refersToClassMap map: 专利 和 d2rq:join "专利表 作者 => 期刊论文表. 作者" 来构建这种关联。而如果一个期刊论文作者对应多个专利文献时,在关联数据中会为期刊论文的一个作者生成指向多个专利文献的 RDF 链接,因此,为了列出多个作者之间可能存在的关系,可以同时使用两个方向的描述,即 d2rq:join "专利表 作者 => 期刊论文表. 作者" 和 d2rq:join "期刊论文表. 作者 <= 专利表 作者",从而描述两个对象(期刊论文与专利文献)之间基于相同属性(作者)的 n:m 的多对多关系,动态表间关系映射编码如下:

```

map PaperAuthor a d2rq.PropertyBridge,
    d2rq.belongsToClassMap map JournalPaper,
    d2rq.property JournalAuthor;
d2rq.refersToClassMap map Patent,
d2rq.join "JournalPaper Author <= Patent Author";
d2rq.join "JournalPaper Author => Patent Author",
d2rq.condition "JournalPaper PubYear > 2000"

```

当这种关系需要满足一定的条件、或在某些条件下才成立时,则需要使用 d2rq:condition 来对这些条件进行描述,从而达到关系的动态构建和多对多关系的转换实现,如其中的 d2rq:condition "JournalPaper. PubYear > 2000",相当于 SQL 中的 WHERE 语句。

(2) 同一表内部关系的构建

代表实体与其属性之间的一对多的关系,可通过 ClassMap 及 PropertyBridge 的描述来实现,而对于同一表内所包含的内在实体关系,D2RQ 映射语言提供对 SQL 语言中的“别名”的支持,即 d2rq:alias [x] 分不同的逻辑对象,再利用 d2rq:join 进行关系描述。因此,可以通过使用别名,实现同一表内部树状关系的提取,并将其这种关系在关联数据中转换为 RDF 链接,如图 4 所示。

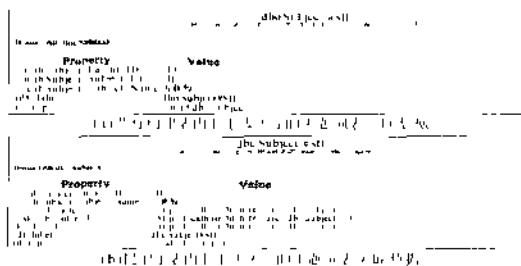


图 4 表内关系的关联数据发布

图 4 中,主题词表 Subject 中的每一个主题词 subjectid 都有一个上位类 subjectparentid,如果采用默认的生成方式(见图 4(a)),只能显示其上位类的概念代码“S10”,但无法得到其链接;而通过建立一个别名表 parentsubject,基于属性相等的条件 subjectid = parentid,则能够得到同一表内部、不同逻辑实体,主题概念及其上位概念之间的等级关系(见图 4(b)),使用以下编码,则可以获得其上位概念 S10 的链接及其下位概念 S112,S113 的链接。

同一表内部关系映射编码如下:

```

map Subject_Parent_ID a d2rq.PropertyBridge,
    d2rq.belongsToClassMap
<file:D> d2r-server document n3#dbo_Subject>,
    d2rq.property skos:broader,
    d2rq.refersToClassMap
<file:D> d2r-server document n3#dbo_Subject<,
    d2rq.join "dbo_Subject_Parent_ID = parentSubject
    Subject_ID",
    d2rq.alias "dbo_Subject AS parentSubject",

```

(3) 与外部数据集的关联构建

关系型数据库的数据内容可能与外部数据之间存在某种联系,如果已知这种联系,可以在发布为关联数据时进行关联创建和描述,从而通过丰富的数据链接,达到构建数据“语境”的目的。在 D2RQ 中,通常支持使用以下两种方式的外部关联应用。

① 与某类常量的关联,即为每一个资源增加一个属性,通过类映射的 d2rq:constantValue,为这个属性关联一个常量

与常量关联的编码如下:

```

map PatentClassMap a d2rq.ClassMap,
    d2rq.class Patent,
map,seeAlsoBridge a d2rq.PropertyBridge,
    d2rq.belongsToClassMap map PatentClassMap,
    d2rq.property rdfs:seeAlso;
    d2rq.constantValue <http://zh.wikipedia.org/wiki/专利>;

```

其中,rdfs:seeAlso 带有一个固定的 URL 对象,连接到每一个 map:PatentClassMap 映射的实例中,即为每一个专利类增加了一个 Wikipedia 的专利概念页。在更早的版本中,对增加常量关联的方法采用增加附加属性的方法 d2rq:additionalProperty 未实现。

② 与通用词表的映射关联 关联数据构建的基本原则之一,是尽可能地使用已有的通用词表^[1]。因此,在关联数据的构建过程中,需要为已有的类或属性名称,建立与通用

词汇的映射，并支持这些通用词汇与 SPARQL 的模式查询。

D2RQ 提供了附加类定义 (d2rq_additionalClassDefinition) 和附加属性定义 (d2rq_additionalPropertyDefinition)，分别与类映射和属性映射相对应，能够为私有的类和属性词汇，建立与通用词表之间的映射。

与通用词表的关联编码如下：

```
map AuthorClassMap a d2rq_ClassMap,
  d2rq_class , Author,
  d2rq_additionalClassDefinitionProperties;
map AuthorEquivalence
map AuthorEquivalence a d2rq_AdditionalProperty,
  d2rq_propertyName owl:equivalentClass;
  d2rq_propertyValue foaf:Person
```

其中，通过附加一个类映射 AuthorEquivalence，为 Author 类与通用的 foaf:Person 类建立了映射关联，通过附加一个属性映射 PaperTitleEquivalence，为 Title 建立了与 dc:title 的映射。而 foaf:Person,dc:title 也能够像 Author 和 Title 一样，先与 SPARQL 的模式查询，实现诸如 (additionalName, ANY, ANY), (URI, ANY, additionalName) 和 (URI, additionalName, ANY) 的模式匹配查询。

4 结语

通过 D2R 的研究与应用可以看出，D2R 能够提供对复杂关系结构的灵活映射，而不改变已经存在的数据模式。这种灵活性是通过在映射规则中直接支持 SQL 语句来获得的。因此，映射语言及映射规则，成为语义模式转换的核心。而评价映射语言的关键，最重要的是考察对象属性的提取和映射能力，即支持关系数据库显性或隐性的数据关联关系的提取和表达能力。与其他类似开源软件如 Drupal、Virtuso、Tuple 相比，D2R 提供了独立和专用的映射语言及表达规则，对于关联关系的提取和支持能力更为灵活，语义映射和转换的实现能力更为突出。

参考文献：

[1] 于珊, 陈红. 数据库系统原理教程 [M]. 1 版. 北京: 清华大学出版社, 1998.

出版社, 1998

- [2] Berners-Lee T. Linked Data - Design Issues [EB/OL] [2011-04-25]. <http://www.w3.org/DesignIssues/LinkedData.html>
- [3] RDF Primer [EB/OL] [2011-06-09]. <http://www.w3.org/TR/2004/REC-rdf-primer-20040210>
- [4] 林维民. 语义网信息组织技术与方法 [M]. 上海: 学林出版社, 2008.
- [5] 长伟, 张明. 语义网简明教程 [M]. 1 版. 北京: 高等教育出版社, 2004.
- [6] RDF Vocabulary Description Language 1.0 - RDF Schema [EB/OL] [2011-06-09]. <http://www.w3.org/TR/rdf-schema-introduction>
- [7] Imai A, Yokota S. RDI Model and Relational Metadata [C] In: Proceedings of the 17th International Conference on Advanced Information Networking and Applications, Xiamen, China. 2003: 534-537.
- [8] Korotkiy M, Top J J. From Relational Data to RDPS Models [C] In: Proceedings of the International Conference on Web Engineering, LNCS 3140, 2004: 430-434.
- [9] Lv Y, Ma Z M. Transformation of Relational Model to RDI Model [EB/OL] [2011-06-09]. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4811327>
- [10] Bizer C, Cyganiak R, Heath T. How to Publish Linked Data on the Web [EB/OL] [2011-06-09]. <http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial relationaldata>
- [11] 几个概念 开放数据、关联数据、语义 Web 和 Web3.0 [EB/OL] [2011-04-01]. <http://www.kenewi.name/archives/2114>
- [12] The D2RQ Platform v0.7 - Treating Non-RDF Relational Databases as Virtual RDF Graphs [EB/OL] [2011-04-01]. <http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/specification>
- [13] Bizer C. D2R MAP - A Database to RDI Mapping Language [EB/OL] [2011-06-12]. <http://www.wiwiss.fu-berlin.de/suhl/bizer/d2rmap/www2003-D2R-Map.pdf>
- [14] 张静, 乃春娥. 如何利用 D2R 发布 Linked Data [EB/OL] [2010-04-25]. http://www.ibm.com/developerworks/cn/web/1003_zhangjing_d2r_index.html ea=drs-cn-0325
- [15] 白海波, 乔晓东. 基于本体和关联数据的图书馆组织语义文化研究 [J]. 现代图书情报技术, 2010(9): 18-27.

(作者 E-mail: bhyt@istc.ac.cn)