

关联数据映射语言:R2R

陶 俊 孙 坦 刘 峥

摘 要 关联数据的目标是使客户端应用通过跟踪 RDF 链接发现新的数据源,实现对不同数据集的平滑整合,但现有方法创建的 RDF 链接十分有限,为此德国柏林自由大学于 2010 年提出面向 RDF 数据集映射的 R2R 语言。该语言包括 `r2r:prefixDefinitions`, `r2r:sourcePattern`, `r2r:targetPattern`, `r2r:transformation`, `r2r:mappingRef` 等映射属性,字符串函数、列表函数、算术函数、XPath 函数及 URI、文字、语言、数据类型等四种变量修饰符,提供 `r2r:hasMapping`, `r2r:mapsTo`, `r2r:dependsOn`, `r2r:sourceDataset`, `r2r:targetDataset` 等映射元数据,映射语句遵循“资源—映射属性—三元组文字”结构及 Turtle 语法等规则。在总结以上语法框架的基础上通过实例阐释其用法,并与 SPARQL 语言进行比较。研究认为 R2R 语言在支持 RDF 表示、变量和修饰符定义、名称空间前缀表示、空白节点定义变量等方面与 SPARQL 存在语法一致性特征,在功能和句法表示方面二者存在一定差别。推广 R2R 语言有助于从词汇和数据集实例两个层面提升 RDF 链接创建能力,进而为创建关联数据集成服务环境提供支持。图 2。表 7。参考文献 21。

关键词 R2R 语言 关联数据映射 RDF 链接 元数据 SPARQL

分类号 G352. 2

Linked Dataset Mapping Language:R2R

Tao Jun, Sun Tan & Liu Zheng

ABSTRACT The goal of linked data is to enable client application to discover new data sources by following RDF links and to smoothly integrate data from those sources. R2R language is an RDF dataset mapping language developed by Free University of Berlin to address the issue that RDF links created by existing methods are extremely limited. R2R includes mapping properties such as `r2r:prefixDefinitions`, `r2r:sourcePattern`, `r2r:targetPattern`, `r2r:transformation`, `r2r:mappingRef`; R2R functions consists of string, list, arithmetic and XPath functions; varname modifiers includes URI, literals, language, datatype modifiers; mapping metadata includes `r2r:hasMapping`, `r2r:mapsTo`, `r2r:dependsOn`, `r2r:sourceDataset`, `r2r:targetDataset`; R2R syntax follows grammatical structure of “Resource-Mapping Property-Literal Value of Triples” and Turtle grammar. This paper summarizes the framework and demonstrates the usage with typical examples. Finally, we compare R2R with RDF-oriented query language SPARQL, and it shows that R2R shows some degree of grammatical consistency with SPARQL in supporting RDF representation, defining varnames and modifiers, expressing prefixes of namespaces, and stating varnames with blank nodes. On the other hand, there are differences between R2R and SPARQL in functionality and syntax. In conclusion, R2R language will be helpful in creating more RDF links from terms of vocabularies and linked datasets, and supporting integrated services of linked data. 2 figs. 7 tabs. 21 refs.

KEY WORDS R2R language. Linked data mapping. RDF links. Metadata. SPARQL.

1 引言

2010年11月,关联开放数据(Linking Open Data, LOD)项目组发布了关联数据规范,其核心规范/协议模型见图1^[1]。

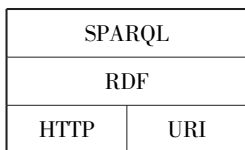


图1 关联数据规范/协议

由上述规范/协议结合关联数据发布规则可知,关联数据的本质是将本体和相关数据资源按照 RDF 格式的标准链接起来,同时要求支持 HTTP URI 访问和 RDF 查询语言 SPARQL 检索。根据 W3C SWEO 研究组的统计,截至 2011 年 9 月,关联数据云(LOD Cloud)含有 310 亿个 RDF 三元组,5 亿个 RDF 链接^[2](见图 2)。由此推算,涉及关联关系的三元组与三元组总量之比不到 3.5%,96% 的 RDF 三元组有建立 RDF 链接的潜力却无 RDF 关联。随着关联数据云中实例数据的快速增长,这一矛盾将进一步加剧。

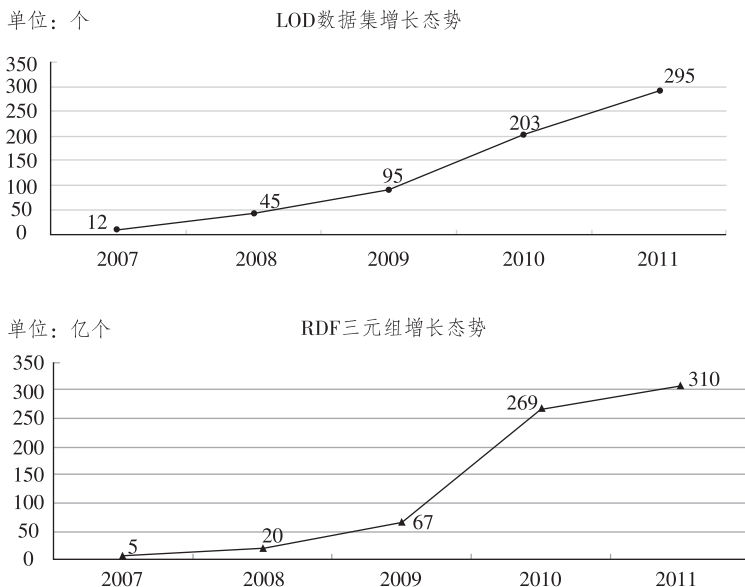


图2 关联数据云 2007—2011 年增长态势

针对发布后的关联数据集,不同数据源用多种本体词汇描述相同实体类型,RDF 关联表现为对上述相同实体创建 RDF 类型关联。创建更多 RDF 关联受词汇异构和实例属性值关系的制约。基于此,国外学者初步形成了发现关联和映射关联两种思路。其中,发现关联是利用已创建的直接关联发现更多不同数据集间的间接关联,从而达到创建更多 RDF 链接的目的。近年来研究集中在运用一定的方法发现实例记录的等同关系,代表成果包括机器学习方

法^[3-5]、各种相似度算法应用^[6-8]、实例关联工具的开发^[9-11]等。与上述思路不同,映射关联从描述数据集的本体词汇出发,消除因不同数据集对相同属性采用多种本体词汇进行描述导致的类或属性异构。由于关联数据规范允许数据发布者自由选择多种本体词汇混合使用以支持实例表示的灵活性,因此,这种自描述方式^[12]形成的词汇异构在 LOD 中十分普遍,克服词汇异构有助于推动更多实例记录关联的创建。

由于缺乏细粒度的映射机制对描述关联数

据集的本体词汇进行深层整合,RDF 关联主要限于实例层,例如通过传统描述方法如 owl:sameAs、skos:exactMatch 等建立实例等同关联^[13],此外也有用 rdfs:subClassOf、rdfs:subPropertyOf、owl:equivalentClass、owl:equivalentProperty 揭示不同本体词汇的等同或包含关系。受词汇异构的制约,更广泛的 RDF 关联无法得到揭示。面对上述困境,德国柏林自由大学学者比泽和舒尔茨提出关联数据映射框架 R2R^[14],该方法拓展了 RDF 关联思路,丰富了映射关联机制,为 LOD 构建面向领域应用的集成服务环境提供了新途径。

R2R 语言是比泽和舒尔茨于 2010 年夏季发布的 RDF 指令语言,目前还处于发展阶段。尽管如此,R2R 语言的简洁性、易扩展性、描述的细粒度以及与语义网联系紧密等特点已逐渐引起人们的重视。2009 年以来,关联数据的研究已引起国内图书情报界的广泛兴趣,但尚未发现运用 R2R 语言探索关联数据映射的成果,

R2R 研究尚未与国际同步。为此,本文在研究 R2R 框架的基础上,结合在 Eclipse HELIOS(集成 JDK 1.6)平台中通过映射处理程序接口调用 R2R 映射实例的体会,向国内同行解读 R2R 语言在关联数据映射中的用法。

2 R2R 语言

R2R 语言的特色在于一方面能够实现不同 RDF 数据集术语映射的创建,另一方面可使生成的映射直接发布为关联数据。作为 R2R 框架的组成部分,R2R 语言包括映射属性、表示规则、函数与修饰符、映射元数据等内容^[15]。

2.1 映射属性

映射属性是 R2R 语言的基础,通过映射属性可创建不同的映射语句,为不同类型 RDF 词汇表(本体)类或属性构建映射。其类型有六种,如表 1 所示。

表 1 R2R 映射属性

类型	映射属性	用法
映射声明	r2r:ClassMapping、 r2r:propertyMapping	定义映射 URI,声明 r2r 映射类型,属强制类型。一般约定类(Class)用大写开头,属性(property)用小写开头表示
前缀定义	r2r:prefixDefinitions	定义本体空间的前缀,一个映射中可有多多个前缀定义
源模式	r2r:sourcePattern	一般定义一个源模式,属强制类型
目标模式	r2r:targetPattern	定义目标模式,可定义一个乃至多个目标模式,属强制类型
转换策略	r2r:transformation	运用函数或修饰符等定义从源模式到目标模式的转换规则
映射调用	r2r:mappingRef	通过映射 URI 调用其他 r2r 映射

R2R 定义了 ClassMapping 和 propertyMapping 两种映射类型的类。类在 R2R 映射语句中充当宾语,声明创建映射的类型,而映射属性为谓语。映射声明、源模式和目标模式语句属强制类型,任何 R2R 映射均要采用;其他映射属性属非强制类型,其应用根据需求适时采用。例 1 反映了上述属性在具体环境中的用法。

例 1:
@ prefix mp: < http://www4. wiwiss. fu-berlin. de/R2Rmappings/ > .

```
mp: DBpediaToFoafMapping rdf: type r2r:
ClassMapping;
r2r: prefixDefinitions " foaf: < http://xmlns.
com/foaf/0.1/ >. dbpedia: < http://dbpedia. org/
ontology/ >";
r2r: sourcePattern "?SUBJ rdf: type dbpedia:
Person";
r2r: targetPattern "?SUBJ rdf: type foaf: Per-
son".
```

这里构建数据集 dbpedia 中 Person 类到本

体 foaf 中 Person 类之间的映射, 第一行声明映射的名称空间前缀; 第二行定义映射 URI 并声明映射类型为类映射; 随后定义源模式和目标模式中用到的前缀; 第四行和第五行分别定义源模式和目标模式, 除空节点外, 一般要使用 ?SUBJ 符号, 这是 R2R 映射规则的基本要求。

2.2 表示规则

R2R 语句在创建过程中需要遵循映射创建规范、函数调用规范、修饰符使用规范等多种表示规则。本节主要介绍创建规范, 其他规范将在 2.3 节阐述。笔者通过运行 R2R 映射实例总结得出 R2R 存在以下表示规则:

(1) R2R 映射语句由“资源—映射属性—三元组文字”结构组成。以例 1 的源模式语句 mp: DBpediaToFoafMapping r2r: sourcePattern "?SUBJ rdf: type dbpedia: Person" 为例, 其“资源”是映射 URI mp: DBpediaToFoafPersonMapping, 映射属性是 r2r: sourcePattern, "?SUBJ rdf: type dbpedia: Person" 则是三元组结构, ?SUBJ, rdf: type 和 dbpedia: Person 分别充当主谓宾, 其整体又是作为映射语句中的文字值。三元组文字表示与 SPARQL^[16] 查询的主谓宾结构一致。与此相同的还有目标模式语句和转换语句; 而前缀定义格式则遵循“资源—映射属性—字符串文字”规则。值得指出的是, 结构中的三元组为文字定义, 应与资源定义区别开来, 故表示时须加引号。

(2) 映射声明语句、源模式语句和目标模式语句是构建映射的基础语句, 其他语句依应用情形的需要选择表示。

(3) 源模式和目标模式语句有两个约束: 其一, 变量不允许在三元组文字的谓语位置; 其二, ?SUBJ 是 R2R 专门的变量, 在源模式和目标模式中均要用到, 其目的是用于标识数据集的主语 URI。

(4) R2R 支持 Turtle 语法^[17] 规则。R2R 构建 RDF 数据集间的映射需选择一种 RDF 序列化方式实现信息建模, 除支持 RDF/XML 语法以 RDF 树结构表示外, R2R 语句也支持 Turtle 语法的 RDF 序列化, Turtle 较 RDF/XML 和 N-Tri-

ple^[18] 序列化格式更为简洁, 目前在基于语义网的各类应用中倍受青睐。例如使用 a 代替 rdf: type, 通过分号和逗号分别实现 RDF 陈述的主语省略和主谓省略, 以 # 字符表示注释。

(5) 各语句的先后顺序无严格要求。例如 r2r: prefixDefinitions 语句置于源模式语句前或目标模式后对程序运行结果均无影响。

(6) 一般将映射声明语句 (例 1 中的第二句) 拆分成两段表示。表示映射语句的主语占一行, 谓语和宾语占一行。突出映射 URI 便于其他映射语句如 r2r: mappingRef 调用该映射。

2.3 R2R 函数与修饰符

R2R 语言提供的函数和修饰符在映射属性 r2r: transformation 的支持下可满足词汇修饰和复杂属性值转换的需求。作为 R2R 语言颇具特色的功能之一, R2R 函数可有效改善 R2R 应用性能, 拓展关联数据映射的范围。R2R 函数和修饰符主要应用于变量, 函数一般应用于 r2r: transformation 语句中, 修饰符支持变量的数据类型、语言标记、变量 URI 与文字值之间的相互转换等修饰情形。

截至 2011 年 9 月, R2R 语言提供字符串函数 (6 种)、算术函数 (6 种)、列表函数 (6 种) 和 XPath 函数 (29 种以上) 四种类型供在映射转换中使用。这些函数综合了 Java 语言函数、XML 路径语言 XPath 函数的重要功能, 并与它们有一定的相关性。其中算术函数较为简单, 包括加、减、乘、除、求整和求余六类运算, 与 Java 中的二元算术运算一致。字符串函数 (见表 2) 在应用中最为广泛, XPath 函数是 R2R 语言于 2011 年 7 月推出的新内容, 尚处于发展中。有关 XPath 函数的功能见文献 [19]。

由表 2 可知, R2R 的字符串函数有六类, 主要包括插入、连接、拆分、替代等字符串处理功能。

例 2:

```
mp: concatFirstAndLastNameMapping
rdf: type r2r: PropertyMapping;
r2r: sourcePattern "?SUBJ foaf: firstName ?f.
?SUBJ foaf: lastName ?l";
```

表 2 字符串函数

函 数	功 能	举 例
join(infix, arg1, arg2, . . . , argN)	将字符 infix 插入字符串 arg1, arg2, . . . , argN 之间。	?a = join(‘-’; 2011, 12, 25)
concat(arg1, arg2, . . . , argN)	连接字符串 arg1, arg2, . . . , argN	?p = concat(‘Hello’, ‘Java’)
split(regex, stringarg)	从符合正则表达式 regex 的地方拆分字符串集合 stringarg	?splitdate = split(‘-’; ?date)
listJoin(infix, list)	将字符 infix 插入数组之间并将数组连接起来	?sameAsDate = listJoin(‘-’; split(‘-’; ?date))
regexToList(regex, stringarg)	按照定义的正则表达式返回字符串列表 stringarg	?Strings = regexToList(‘\.(*) - (. *) - (. *)’; ?date)
replaceAll(Regex, withThatString, String)	在变量 String 中用字符串 withThatString 替代正则表达式 Regex	?formattedDate = replaceAll(‘-’; ‘/’; ?date)

r2r: targetPattern "?SUBJ v: n ?name";
r2r: transformation "?name = concat(?1, ‘;’; ?f)";
r2r: prefixDefinitions " foaf: < http://xmlns.com/foaf/0.1/ > . v: < http://www.w3.org/2006/vcard/ns# > ".
例 2 的应用情景是构建词汇 foaf:firstName, foaf:lastName 和词汇 v:n 的映射, 且要实现姓

(last Name) 和名(first Name) 的交换。这里采用连接函数 concat(), 通过转换策略语句 r2r:transformation 构建变量 name 的转换规则?name = concat(?1, ‘;’; ?f)。这样原数据集的 2n 个三元组记录经转换后在新数据集中生成了 n 个三元组, 新三元组的本体词汇由 v:n 替代。
列表函数与 Java 语言有类似之处, 表 3 总结了列表函数的功能。

表 3 列表函数

函 数	功 能	举 例
list(arg1, arg2, . . . , argN)	创建一个参数列表 arg1, arg2, . . . , argN	list(1, 2, 3) → [1, 2, 3]
subList(listarg, from, to)	截取数组 listarg 第 from 位到第 to - 1 位的数组	subList(list(5, 6, 7, 8, 9), 1, 4) → [6, 7, 8]
subListByIndex(listarg, i1, i2, . . . , iN)	从数组列表 listarg 的输出中, 截取位于 i1, i2, . . . , iN 号位置的数组	subListByIndex(list(5, 6, 7, 8, 9), 3, 1) → [8, 6]
listConcat(listArg1, listArg2, . . . , listArg3)	连接所有数组为一个新的数组	listConcat(list(1, 2), list(3, 4)) → [1, 2, 3, 4]
getByIndex(listArg, index)	输出数组 listArg 中位于第 index 号的元素	getByIndex(list(4, 5, 6), 2) → 6
length(arg)	返回数组的长度	length(5) → 1, length(list(5, 6)) → 2

由表 3 可知, R2R 列表函数主要用于输出数组元素、截取子数组、返回数组长度等。与 Ja-

va 语言数组的表示一致, R2R 列表函数中的数
位从 0 开始, 首位为 a0, 依此类推。例如 sublist

(list(5,6,7,8,9),1,4)表示截取数位在第1位到第4-1位的所有序列,即数位a1,a2,a3,故输出为6,7,8。同理,subListByIndex(list(5,6,7,8,9),3,1)表示数位为a3和a1的输出元素。

此外,R2R 也提供条件三元运算,其表示规则为:
[式1 X 式2 ? 式3 :式4]
规则中各式表示参与运算的表达式,X 是任一比较运算符(= , > , < , < = , > = , ! =),表示“式1”和“式2”之间的关系,该三元运算的含义是先求式1×式2 的值,若为真,则执行式3 并返回式3 的结果;若为假,则执行式4 并返回式4 的结果。例如:

concat([?sex = female ^ ? Ms. ^: Mr.], ?familyname) "

上例在 concat 函数中嵌套条件三元运算构建称呼与姓氏的连接,即若为女性则与 Ms. 连接,反之与 Mr. 连接。

由于 R2R 函数自身的不健壮性,R2R 语言提供了 r2r:importFunction 属性来实现外部函数的导入,以弥补 R2R 函数的不足。可通过配置类路径、本地文件、远程 URL 方式等实现外部函数的装载。

除使用函数功能外,R2R 语言在变量中还可使用以下四类修饰符(见表4)。

表 4 变量修饰符

类型	功能	举例
URI 修饰符	变量值变成 URI,将变量名称打尖括号	?o→? <o >
文字修饰符	将变量名称打引号	?varname→?'varname'
语言修饰符	字符串文字值增加语言标记	? 'varname' @ en
数据类型修饰符	使变量值的数据类型具体化,与文字修饰符联合使用	dc: date ?c^^xsd: dateTime

由表 4 可知,R2R 修饰符与 SPARQL 类似。修饰符均用于三元组文字的宾语,故常用 rdfs:label 属性充当谓语。值得注意的是,URI 修饰符不得用于修饰谓语。正如 2.2 节所述,三元组文字的谓语必须是明确的 URI,而不能由变量充当。例如语句"?SUBJ ?p ?o"在程序运行中会出现异常,因为该谓语是变量?p,不符合 R2R 语法规定。

2.4 映射元数据

如前所述,笔者以映射属性为起点介绍了

如何通过表示规则、函数与变量修饰符创建映射程序。上述映射程序尚未对映射本身进行描述。在 R2R 中,描述映射本身的属性以及提供映射发现的属性统称为映射元数据。映射元数据一方面可构建映射 URI 与映射数据集和本体词汇的链接机制,为用户获得高质量的映射提供参考;另一方面可用于标注 R2R 映射,建立本体词汇与映射 URI 之间的关联发现。简言之,映射元数据用于支撑 R2R 映射的发布和发现机制(见表5)。

表 5 R2R 映射元数据

元数据类型	功能
r2r:mapsTo	链接映射 URI 与目标映射词汇,映射 URI 为主语
r2r:dependsOn	链接映射 URI 与源映射词汇,映射 URI 为主语
r2r:sourceDataset	声明映射中源模式对应数据集的元信息,映射 URI 为主语
r2r:targetDataset	声明映射中目标模式对应数据集的元信息,映射 URI 为主语
r2r:hasMapping	建立映射词汇与映射 URI 之间的关联,映射词汇为主语

其中,前四组元数据用于 r2r 映射的发布,第五组用于构建映射词汇与映射 URI 之间的关联发现。除 R2R 提供的元数据外,在应用程序中通常还使用由 RDF schema, Dublin Core 提供的 rdfs:label, rdfs:comment, dc:date, dc:creator 等常用的元数据进行描述。例 3 阐述了映射元数据的有关用法。

```
例 3-1:
mp: concatFirstAndLastNameMapping
rdfs: label "foaf to vard name property mapping and individuals changed and concated";
rdfs: comment "Name are transformed through changing places with each other";
dc: date "2011 - 09 - 11";
dc: creator < http://www.w3.org/2006/vcard/ns#/name/Bizer >;
r2r: mapsTo v: n;
r2r: dependsOn foaf: firstName;
r2r: dependsOn foaf: lastName;
r2r: sourceDataset mapping: foafVOID;
r2r: targetDataset mapping: vcardVOID.
```

例 3-1 通过元数据定义映射的元信息。首行是映射 URI,随后四行定义了映射的标签及创建者来源信息,为映射质量评估提供元信息。接着通过 mapsTo 和 dependsOn 定义了 R2R 映射与目标数据集的本体词汇和源数据集的本体词汇。最后两行定义了源数据集和目标数据集的映射 VOID 描述。VOID 词汇表(The Vocabulary of Interlinked Datasets)是描述关联数据集本身的元数据集^[20], W3C 倡议发布关联数据应提供数据集本身的描述信息以促进关联数据集消费,在此背景下 VOID 应运而生,它本质上是一种本体。数据发布者可通过 VOID 对关联数据集进行著录描述、数据集 VOID 存档和 VOID 信息的部署;关联数据消费者则可通过它来发现和重用数据集及其映射本身的信息。由于 VOID 框架描述数据集的全面性,其使用较 Semantic Sitemaps 更为普遍。正因如此,VOID 有望成为描述关联数据集的 W3C 推荐标准^[21]。

例 3-1 通过表 5 的前四组元数据提供了 R2R 映射的发布属性,元数据 r2r:hasMapping 则

提供了 R2R 映射的发现属性。那么二者有何区别呢?映射发布是从数据提供者的角度将映射本身的描述信息部署后发布到 Web 中,映射发现则是从映射消费者角度提供映射词汇与映射本身的关联发现,表示该本体词汇有相应的 R2R 映射可供消费者使用。例 3-2 揭示了 hasMapping 的使用。

```
例 3-2:
v: n rdf: type owl: DatatypeProperty;
rdfs: domain owl: Thing;
rdfs: range xsd: string;
r2r: hasMapping mp: concatFirstAndLastNameMapping.
```

上例首先声明属性 v:n 属于数据属性,区别于对象属性(owl:ObjectProperty), rdfs:domain 和 rdfs:range 分别声明 v:n 的主语和宾语的取值范围是实例和字符串资源类型, mp: concatFirstAndLastNameMapping 是映射 URI, v:n 通过 r2r:hasMapping 元数据建立与映射 URI 的关联。映射消费者通过定向查询 r2r:hasMapping, 可发现映射词汇及关联的映射 URI。通过发现机制共享网上已有的映射文件,可实现映射的有效重用。

从映射作用 RDF 位置的角度可将映射分为词汇层(Vocabulary, V)和实例层(Dataset, D)映射。词汇层用于转换术语属性,实现不同词汇间类的映射或属性的映射;实例层定义实例怎样转换,主要是消除属性值不一致,例如,统一不同的测量单位,或增加属性值的语义标签或数据类型。R2R 语言一方面能够分别构建各层映射(V-V, D-D), 另一方面也能实现彼此间的映射(V-D, D-V), 见表 6。例 1 就是 V-V 模式, 例 2 将 foaf:firstName 和 foaf:lastName 两个本体词汇映射到 v:n, 并通过连接函数调整次序合并生成新的实例, 则属于词汇层到实例层映射的 V-D 模式。

表 6 R2R 映射类型

序号	类型	源模式	目标模式
1	V-V	本体词汇	本体词汇
2	D-D	数据集实例	数据集实例
3	D-V	数据集实例	本体词汇
4	V-D	本体词汇	数据集实例

3 与 SPARQL 语言的比较

R2R 语言是专门针对 RDF 数据集之间的转换而开发的映射语言,其命名(RDF dataset to RDF dataset,缩写 R2R)即体现了上述基本目

标。基于前文发现, R2R 在应用中结合了 SPARQL、XPath、Java 等的有关规则或函数功能。R2R 语言最突出的特点是借鉴了 SPARQL 语言,二者在表示上既有联系又有区别,其主要差异见表 7。

表 7 R2R 与 SPARQL 比较

比较内容		R2R	SPARQL
语法一致性	支持 RDF 三元组表示	√	√
	变量表示方式	√	√
	修饰符表示方式	√	√
	关键字 PREFIX 支持名称空间前缀使用	√	√
	支持 Turtle 序列化	√	√
	支持空白节点([])表示	√	√
语法区别	SELECT, CONSTRUCT, ASK, DESCRIBE 句法		√
	映射属性;映射元数据	√	
	支持通配符(*)查询		√
	? SUBJ 固定变量定义	√	

例 1 至例 3 是 R2R 语言实例,下面例举 SPARQL 语言的实例,通过对比揭示二者语法一致性特征。

```
例 4:
PREFIX rdfs: < http://www. w3. org/2000/
01/rdf-schema# >
PREFIX dbprop: < http://dbpedia. org/prop-
erty/ >
SELECT ?location
WHERE{
?person rdfs: lable“George Washington”@ en.
?location dbprop: namedFor ?person.
}
```

该例用于 SPARQL 端点下的 RDF 查询,其目标是查找英文名为 George Washington 的地方。其中 Where 中的两个子句均用 RDF 三元组表示,说明 SPARQL 支持 RDF 三元组表示;变量?location 表明 SPARQL 同样支持采用“?变量名”的方式定义变量(此外 SPARQL 和 R2R 均可用“\$”表示变量定义);地名限定运用@ en 修

饰字符串 George Washington;通过 RDF 三元组谓词可看出, SPARQL 支持 PREFIX 定义名称空间前缀;整体表示结构则反映了 SPARQL 也运用 Turtle 序列化方式。

尽管均支持 RDF 三元组表示,但 SPARQL 用于查询功能,而 R2R 则以映射为目的,因此二者又表现出多种语法差别。SPARQL 围绕查询提供 SELECT、CONSTRUCT、ASK 和 DESCRIBE 等句法, R2R 围绕映射提供源模式 r2r: source-Pattern、目标模式 r2r: targetPattern、转换策略 r2r: transformation 等多种映射属性来支持映射语句的生成。SPARQL 中的 SELECT 和 DESCRIBE 句法还支持用通配符(*) 返回所有已知变量,而 R2R 通过?SUBJ 这一特有变量定义来标识 RDF 数据集的主语 URI。在目标模式中?SUBJ 所在位置支持使用空白节点,声明主语变量与其他属性的绑定;而 SPARQL 语言同样支持空白节点作为变量。

作为面向 RDF 的两种语言,二者在关联数据应用中互为补充,不可替代。发布到 LOD 中

的 RDF 数据集,其采集除采用 RDF dump 静态获取外,更多通过 SPARQL 端点查询实现动态获取。采集到的 JSON, RDF/XML, N-Triple, N3/Turtle 和 TSV(如 Freebase, Geonames 的 Dump 文件)等多种格式混合的数据源经清洗预处理,随后构建 R2R 映射程序,生成的映射文件供映射消费者在 Jena 语义 Web 框架下通过映射处理程序接口调用,以支持面向领域应用的集成服务环境的创建。

4 结语

R2R 语言是针对当前不同 RDF 数据集仅能构建实例等同关联关系和通过 `rdfs:subPropertyOf`, `owl:equivalentClass` 等定义本体词汇的等同和包含关系,而无法解决 RDF 数据集间的本体词汇异构和属性值转换问题开发的映射语言。作为在因特网上共享数据的方法,关联数据与其他方法的区别在于它是用自描述方式发布的,自描述方式会优先重用业界熟悉的知名本体,但由于知名本体通常不会覆盖发布数据的所有方面,为了提升关联数据的发布能力就要选择其他本体,包括选择自定义的本体词汇进行混合描述,这样就容易形成词汇异构。R2R 语言为解决这一问题提供了程序化的解决方案。同时我们也看到,R2R 语言整体还处于发展阶段,其缺陷与不足还有待我们在实践中进一步发掘。但毋庸置疑,R2R 语言为我们研究关联数据映射提供了可兹参考的新思路。

R2R 框架的应用才刚刚开始。围绕关联数据映射的研究,下一步一方面仍将密切跟踪 R2R 框架的发展,另一方面将开展面向 LOD 数据集的映射整体案例研究。在此基础上,结合本体匹配的最新研究成果,从理论和实践两方面提升 R2R 框架在解决关联数据映射方面的使用效果。

参考文献:

- [1] Linked Data Specifications [OL]. [2011-09-29]. <http://linkeddata-specs.info/>.
- [2] LinkingOpenData [OL]. [2011-09-29]. <http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>.
- [3] Mayfield J, Alexander D, Dorr B, et al. Cross-document coreference resolution: A key technology for learning by reading [C/OL]. Proceedings of AAAI Spring Symposium on Learning by Reading and Learning to Read, 2009. [2011-12-28]. <http://terpconnect.umd.edu/~oard/pdf/sss09.pdf>.
- [4] Sleeman J, Finin T. A machine learning approach to linking FOAF instances [C/OL]. Association for the Advancement of Artificial Intelligence, 2010. [2011-12-28]. <http://ebiquity.umbc.edu/paper/html/id/471>.
- [5] Nikolov A, Motta E. Capturing emerging relations between schema ontologies on the Web of data [C/OL]. Proceedings of the 1st International Workshop on Consuming Linked Data, ISWC 2010, Shanghai, China, 2010. [2011-12-28]. http://ceur-ws.org/Vol-665/NikolovEtAl_COLD2010.pdf.
- [6] Raimond Y, Sutton C, Sandler M. Automatic inter-linking of music datasets on the semantic Web [C/OL]. proceedings of WWW2008 workshop-Linked Data on the Web (LDOW2008). [2011-12-28]. <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-369/paper18.pdf>.
- [7] Halpin H, Hayes P J. When owl:Same as isn't the same: An analysis of identity links on the semantic Web [C]. 3rd Workshop about Linked Data on the Web (LDOW2010), Raleigh, North Carolina, 2010.
- [8] 邓兰兰,李春旺. Web 数据关联创建策略研究 [J]. 现代图书情报技术, 2011(5): 1-6. (Deng Lanlan, Li Chunwang. Study on linkage creation strategies of Web data [J]. New Technology of Library and Information Service, 2011(5): 1-6.)
- [9] Volz J, Bizer C, Gaedke M, et al. Silk-A link discovery framework for the Web of data [C/OL]. Proceedings of WWW2009 workshop-Linked Data on the Web (LDOW2009). [2011-12-28]. <http://www4.wiwiw.fu-berlin.de/bizer/silk>.
- [10] LIMES-Link discovery framework for METRIC Spaces [OL]. [2011-12-28]. <http://aksw.org/Projects/limes>.
- [11] Nikolov A. Fusing automatically extracted annota-

- tions for the semantic Web [R/OL]. [2011 - 12 - 28]. Technical Report KMI, 2006. <http://people.kmi.open.ac.uk/andriy/nikolov-thesis-submitted.pdf>.
- [12] Mendelsohn N. The self-describing Web, W3C TAG finding [OL]. [2011 - 09 - 29]. <http://www.w3.org/2001/tag/doc/selfDescribingDocuments>.
- [13] Miles A, Bechhofer S. SKOS simple knowledge organization system reference; 10. 6 [R/OL]. [2011 - 12 - 28]. <http://www.w3.org/TR/2009/REC-skos-reference-20090818/#L4858>.
- [14] Bizer C, Schultz A. The R2R framework; Publishing and discovering mappings on the Web [C]. Proceedings of the First International Workshop on Consuming Linked Data (COLD2010).
- [15] Schultz A, Bizer C. The R2R framework [OL]. [2011 - 09 - 29]. <http://www4.wiwiw.fu-berlin.de/bizer/r2r/spec/>.
- [16] W3C. SPARQL query language for RDF [OL]. [2011 - 09 - 29]. <http://www.w3.org/TR/rdf-sparql-query/>.
- [17] W3C Team Submission. Turtle-terse RDF triple language [OL]. [2011 - 09 - 29]. <http://www.w3.org/TeamSubmission/2011/SUBM-turtle-20110328/>.
- [18] W3C. RDF test cases [OL]. [2011 - 09 - 29]. <http://www.w3.org/TR/rdf-testcases/#ntriples>.
- [19] XPath, XQuery 以及 XSLT 函数 [OL]. [2011 - 12 - 28]. http://www.w3school.com.cn/xpath/xpath_functions.asp. (XPath, XQuery, and XSLT functions [OL]. [2011 - 12 - 28]. http://www.w3schools.com/xpath/xpath_functions.asp.)
- [20] Cyganiak R, Zhao J, Alexander K, et al. Vocabulary of interlinked datasets [OL]. [2011 - 09 - 29]. <http://vocab.deri.ie/void>.
- [21] W3C Interest Group. Describing linked datasets with the VoID vocabulary [OL]. [2011 - 09 - 29]. <http://www.w3.org/TR/2011/NOTE-void-20110303/>.

陶俊 中国科学院国家科学图书馆 2009 级博士研究生。通讯地址: 北京北四环西路 33 号。邮编: 100190。

孙坦 中国科学院国家科学图书馆研究馆员, 博士生导师。通讯地址同上。

刘峥 中国科学院国家科学图书馆副研究馆员。通讯地址同上。

(收稿日期: 2011-09-29; 最后修回日期: 2012-02-03)