# 从 OWL 本体到关系数据库模式的转换

许卓明,黄永菁

(河海大学计算机及信息工程学院,江苏 南京 210098)

**摘要** 给出了 OWL 本体和关系数据库模式的形式定义,在分析了 OWL 本体和关系数据库模式之间概念对应的基础上,定义了从 OWL 本体到关系数据库模式的转换规则.基于 J2SE 平台的算法实现和典型实例表明,从 OWL 本体到关系数据库模式的转换是可行的.

**关键词** 模式转换;本体;关系数据库模式;OWL;语义 Web

**中图分类号**:TP311 **文献标识码**:A **文章编号**:1000-1980(2006)01-0095-05

语义 Web(Semantic Web)[1]是当前 Web 的一种扩充.Web 信息将赋予机器可处理的语义,使自动工具(和人)能更有效地发现、处理、集成和重用 Web 内容和服务.基于本体(ontology)[2]构造语义 Web 站点(site)[3]或门户(portal)是将当前 Web 提升为语义 Web 的关键环节.早期,构造语义 Web(门户)站点采用自底向上(bottom-up)的方法,将数据密集型 Web(门户)站点"迁移"为语义 Web(门户)站点[3].但真正意义上的语义 Web(门户)站点①的构造应采用自顶向下(top-down)的方法,即先建立领域本体,再在领域本体的驱动下设计站点底层的数据库模式以及站点的内容(如 Web 页和 Web 服务等).本体是对共享概念化的一种显式的形式规约,它通过类、属性以及类和属性之间的约束来表示静态的领域知识[4].因此,一个本体可认为是一种描述某个领域的概念模式.据此,笔者认为,从本体到关系数据库(relational database, RDB)模式的转换在理论上是可行的.当前,OWL②已成为标准的 Web 本体语言,因此,本文研究从 OWL 本体到关系数据库模式的转换方法.这样的转换是实现自顶向下构造语义 Web(门户)站点的基础.

## 1 OWL 本体和关系数据库模式

一个 OWL 本体由一个词汇表(vocabulary)和一组公理(axiom)组成.OWL 语言有两种语法格式:(a)交换语法(exchange syntax),即 RDF/XML 语法,将一个 OWL 本体表示成 RDF 三元组的集合,以便在 Web 上发布和共享本体;(b)抽象语法(abstract syntax),即框架风格的语法,关于一个类或属性的一组信息用一个大的构造子来表示,以便本体用户理解和评价本体.两种语法格式表示的同一个 OWL 本体的语义是等价的,仅由其底层的 RDF 三元组唯一决定[5].定义 1 是 OWL 本体的简化的形式定义.

**定义 1** 一个 OWL 本体 $O$ 是一个二元组 $O=(ID, Axiom)$.其中:

**a.** 标识符集 $ID = CID \cup DPID \cup OPID \cup DTID$ 是一个有限集,$CID$,$DPID$,$OPID$ 和 $DTID$ 两两不相交.$CID$ 是类标识符集,包含本体中定义的所有类的标识符;$DPID$ 是数据类型属性标识符集,包含本体中定义的所有数据类型属性的标识符;$OPID$ 是对象属性标识符集,包含本体中定义的所有对象属性标识符;$DTID$ 是数据类型标识符集,包含本体中使用的所有预定义的 *XML Schema* 数据类型标识符③.

**b.** 公理集 $Axiom = CAxiom \cup PAxiom$ 是一个有限集,$CAxiom$ 和 $PAxiom$ 不相交.$CAxiom$ 是类公理集,包含

**收稿日期**:2005-09-07

**基金项目**:国家自然科学基金资助项目(60573098);江苏省自然科学基金重点资助项目(BK2003001)

**作者简介**:许卓明(1965—),男,江苏苏州人,教授,主要从事数据库、信息检索及语义 Web 技术研究.

① STOLLBERG M, LAUSEN H, LARA R, et al. Towards Semantic Web Portals//Proc of the WWW2004 Workshop on Application Design, Development and Implementation Issues in the Semantic Web. CEUR Workshop Online Proceedings, 2004. http://CEUR-WS.org/Vol-105/.

② MCGUINNESS D L, HARMELEN F V (eds). OWL Web Ontology Language Overview (W3C Recommendation). http://www.w3.org/TR/2004/REC-owl-features-20040210/, 10 February 2004.

③ BIRON P V, PERMANENTE K, MALHOTRA A (eds). XML Schema Part 2: Datatypes Second Edition (W3C Recommendation). http://www.w3.org/TR/xmlschema-2/, 28 October 2004.

本体中定义的所有类公理. $PAxiom$ 是属性公理集,包含本体中定义的所有属性公理. 特别地,子类约束 $\mathrm{subClassOf}(c,d) \in CAxiom$ 表示 $c$ 是 $d$ 的子类,即 $c$ 的实例集是 $d$ 的实例集的子集;属性的定义域约束 $\mathrm{domain}(p) \in PAxiom$ 表示,若 $p \in DPID \cup OPID$ 的定义域是 $c \in CID$,则 $\mathrm{domain}(p) = c$;属性的值域约束 $\mathrm{range}(p) \in PAxiom$ 表示,若 $p \in DPID$ 的值域是 $dt \in DTID$,则 $\mathrm{range}(p) = dt$,若 $p \in OPID$ 的值域是 $c \in CID$,则 $\mathrm{range}(p) = c$ 函数属性约束 $\mathrm{functional}(p) \in PAxiom$ 表示 $p \in DPID$ 是函数属性,即 $p$ 在 $\mathrm{range}(p)$ 中的取值唯一;属性互逆约束 $\mathrm{inverse}(p,q) \in PAxiom$ 表示 $p, q \in OPID$,$\mathrm{domain}(p) = \mathrm{range}(q)$ 且 $\mathrm{range}(p) = \mathrm{domain}(q)$.

一个关系数据库模式由一组关系模式组成,其中包含数据库的基表结构和完整性约束两部分[6]. 基表结构定义了关系(表)的结构、属性(列)及其数据类型与长度等;完整性约束定义了语义施加在数据上的约束,在此,仅考虑主键/唯一列和外键约束. 定义 2 是关系数据库模式简化的形式定义.

**定义 2** 一个关系数据库模式 $D$ 是一个四元组 $D = (Name, PK, Unique, FK)$. 其中:

**a.** 名集 $Name = ET \cup RT \cup DT$ 是一个有限集,$ET$,$RT$ 和 $DT$ 两两不相交. $ET$ 是实体表名的集合,$RT$ 是联系表名的集合,$DT$ 是数据类型名的集合,每个数据类型名是 DBMS 预定义的类型名,如 integer.

**b.** $\forall t \in ET \cup RT$ 有一个非空的列的集合 $U(t)$,且每个列 $c \in U(t)$ 有一个相关的预定义数据类型 $\mathrm{datatype}(c) \in DT$ 作为它的取值范围.

**c.** $\forall t \in ET \cup RT$,$t$ 有且仅有一个主键 $PK(t)$,其值唯一地决定了 $t$ 中每行实例数据,且要么 $PK(t) \in U(t)$(此时称它为单主键,只有实体表有且仅有单主键),要么 $PK(t) \subseteq U(t)$(此时称它为复合主键,只有联系表有且仅有复合主键).

**d.** $\forall t \in ET$,若 $t$ 存在一个主键 $PK(t)$,则 $t$ 可能有 $0 \sim n$ $(n \geqslant 1)$ 个唯一列 $\mathrm{Unique}(t) \in U(t)$ 且 $\mathrm{Unique}(t) \neq PK(t)$,其值唯一地决定了 $t$ 中每行实例数据.

**e.** $\forall t \in ET \cup RT$,$t$ 可能有 $0 \sim m (m \geqslant 1)$ 个引用其他实体表 $r \in ET$ 单主键的外键 $\mathrm{FK}(t,r) \in U(t)$,满足 $\mathrm{value}(\mathrm{FK}(t,r)) \subseteq \mathrm{value}(PK(r)) \cup \{\mathrm{null}\}$,$\mathrm{PK}(r) \in U(r)$. 其中 $\mathrm{value}(*)$ 表示"$*$"的值域.

## 2 从 OWL 本体到关系数据库模式的转换

### 2.1 两者的概念对应

实体-联系(ER)模式到关系数据库模式的转换规则基于它们之间的概念对应[6]. 文献[7]已抽象出 ER 模式与 OWL 本体间的概念对应. 根据这些对应关系,很容易给出 OWL 本体与关系数据库模式间的对应关系(表 1 和表 2).

**表 1 OWL 本体与关系数据库模式之间元素的对应关系**

**Table 1 Element correspondence between OWL ontology and RDB schema**

| OWL 本体中的元素 | 关系数据库模式中的元素 |
|---|---|
| 类 | 表 |
| 以一个类为定义域的数据类型属性及其 XML 模式数据类型 | 表的非外键列及其相应的预定义数据类型 |
| 以一个类为定义域的一个或多个数据类型属性如果是函数属性 | 其中一个列声明为定义域类所对应的表(实体表)的主键,其他列声明为唯一列 |
| 类 1 与类 2 之间的一对互逆的对象属性,类 1 与类 3 之间的一对互逆的对象属性,且类 2 和类 3 中都有函数属性 | 类 1 对应的表为联系表,其中添加一个引用类 2 对应的表(实体表)主键的外键以及一个引用类 3 对应的表(实体表)主键的外键,并将这两个外键组成这个联系表的复合主键 |
| 类 1 与类 2 间的子类-超类关系,即类 1 是类 2 的子类 | 在类 1 对应的表中添加一个主键列,且这个列同时成为引用类 2 对应的表的主键的一个外键 |

### 2.2 转换规则与算法

基于以上对应关系,可导出从 OWL 本体到关系数据库模式的转换规则(即如下规则 1 ~ 6). 为形式化表示转换规则,引入以下辅助函数:

**a.** $\mathrm{identifier}(id)$ 表示取 OWL 本体 $O$ 中标识符的字面量,即若 $id \in ID$,则 $\mathrm{identifier}(id) = id$.

**b.** $\mathrm{c2tMapping}(c)$ 表示取 OWL 本体 $O$ 中的类所对应的关系数据库模式 $D$ 中的表,即若 $c \in CID$ 所对应的表是 $t \in ET \cup RT$,则 $\mathrm{c2tMapping}(\mathrm{c}) = t$.

**c.** dtMapping($dt$):表示取 OWL 本体 $O$ 中数据类型所对应的关系数据库模式 $D$ 中的数据类型,即若 $dt \in DTID$ 的所对应的关系数据库模式中的数据类型是 $dtype \in DT$,则 dtMapping($dt$) = $dtype$.

**规则 1**　将 OWL 本体中的每个类转换为关系数据库模式中的一张表,且表名与类名同名. 即:$\forall\ c \in CID \rightarrow t = \text{identifier}(c) \in ET \cup RT$.

**规则 2**　将 OWL 本体中的每一个数据类型属性及其值域转换为其定义域所对应的表中的列及其相应数据类型. 即:$\forall\ dp \in DPID \rightarrow a = \text{identifier}(dp) \in U(\text{c2tMapping}(domain(dp))) \wedge \text{datatype}(a) = \text{dtMapping}(range(dp)) \in DT$.

**规则 3**　将 OWL 本体中函数属性的定义域所对应的表声明为实体表,并且,若此表尚不存在主键,则将这个属性所对应的列声明为此表的主键. 即:$\forall\ dp \in DPID \wedge functional(dp) \wedge \exists\ t = \text{c2tMapping}(domain(dp)) \in ET \cup RT \wedge \neg \exists\ PK(t) \in U(t) \rightarrow t \in ET \wedge PK(t) = \text{identifier}(dp) \in U(t)$.

**规则 4**　若 OWL 本体中一个函数属性的定义域所对应的实体表已存在主键,则将这个属性所对应的列声明为此表的唯一列. 即:$\forall\ dp \in DPID \wedge functional(dp) \wedge \exists\ t = \text{c2tMapping}(domain(dp)) \in ET \wedge \exists\ PK(t) \in U(t) \rightarrow Unique(t) = \text{identifier}(dp) \in U(t)$.

**规则 5**　若 OWL 本体中类 $c$ 是类 $d$ 的子类,则将类 $c$ 所对应的表声明为实体表,并在此表中添加一个与类 $d$ 所对应的实体表的主键同名的列. 此列既作为此表的主键,又作为引用类 $d$ 所对应的实体表主键的外键. 即:$\forall\ c, d \in CID \wedge subClassOf(c, d) \wedge \exists\ t = \text{c2tMapping}(c) \in ET \cup RT \wedge \exists\ r = \text{c2tMapping}(d) \in ET \rightarrow t \in ET \wedge a = \text{identifier}(PK(r)) \in U(t) \wedge PK(t) = a \wedge FK(t, r) = a$.

**规则 6**　若 OWL 本体中类 $d$ 和类 $e$ 所对应的表均是实体表,且它们与类 $c$ 之间均有一对互逆的对象属性,则将类 $c$ 所对应的表声明为联系表,且在此表中添加两个列,并将这两列分别作为外键引用类 $d$ 和类 $e$ 所对应的两个表的主键,同时将这两列的组合作为此表的复合主键. 即:$\forall\ c, d, e \in CID \wedge \exists\ t = \text{c2tMapping}(c) \in ET \cup RT \wedge \exists\ r = \text{c2tMapping}(d) \in ET \wedge \exists\ s = \text{c2tMapping}(e) \in ET \wedge \exists\ p, q, u, v \in OPID \wedge domain(p) = domain(q) = c \wedge range(p) = d \wedge range(q) = e \wedge inverse(p, u) \wedge inverse(q, v) \rightarrow t \in RT \wedge a = FK(t, r) \in U(t) \wedge b = FK(t, s) \in U(t) \wedge PK(t) = \{a, b\} \subseteq U(t)$.

以这些转换规则为核心,笔者设计了一个从 OWL 本体到关系数据库模式的转换算法 Onto2RDB. 算法的输入为 OWL 本体(用抽象语法表示),输出为关系数据库模式的 DDL 语句. 依据以上转换规则,分别对 OWL 本体中的类、数据类型属性、对象属性及类/属性公理进行处理.

Onto2RDB 算法基于 J2SE v1.4.2 平台实现,并用开放源码关系数据库系统 MySQL 4.0 的 DDL 语句作为输出. MySQL 数据库基本上与 ANSI/ISO SQL 标准兼容,但在外键约束上有其独特之处. 例如:外键约束只能在 InnoDB 类型的表上定义,引用表的主键列上和被引用表的外键列上都要有索引. 因此,Onto2RDB 算法实现时将每个本体类转换成一个 InnoDB 类型的表,主/外键列上添加了索引定义.

表 2　OWL 本体与关系数据库之间数据类型的对应关系

Table 2　Datatype correspondence between OWL ontology and RDB

| 数据类型 | OWL 本体中数据类型属性的值域(XML 模式数据类型) | 关系数据库中列的数据类型(MySQL) |
|---|---|---|
| 数值型 | xsd:decimal | NUMERIC/DECIMAL/DEC,转换时取 NUMERIC |
| | xsd:integer | INTEGER/INT,转换时取 INT |
| | xsd:short | SMALLINT |
| | xsd:float | FLOAT/REAL,转换时取 FLOAT |
| | xsd:double | DOUBLE PRECISION |
| 字符型 | xsd:string | CHAR/VARCHAR/TEXT,转换时取 TEXT |
| 日期时间型 | xsd:dateTime | DATETIME |
| | xsd:date | DATE |
| | xsd:time | TIME |
| 布尔型 | xsd:boolean | BOOLEAN (MySQL v4.1 + 支持) |

## 3　实　例

以下面的 OWL 本体作为 Onto2RDB 算法的输入,

```
Ontology(studentsTakeCourse Annotation(VersionInfo 1.0)
Class(Staff)
DatatypeProperty(staffNo domain(Staff) range(xsd:short) Functional)
DatatypeProperty(staffName domain(Staff) range(xsd:string))
DatatypeProperty(staffAge domain(Staff) range(xsd:integer))
```

DatatypeProperty(jobTitle domain(Staff) range(xsd:string))
DatatypeProperty(emailAddr domain(Staff) range(xsd:string))
Class(AcademicStaff)
subClassOf(AcademicStaff Staff)
DatatypeProperty(specialty domain(AcademicStaff) range(xsd:string))
Class(Student)
DatatypeProperty(studentNo domain(Student) range(xsd:short) Functional)
DatatypeProperty(studentName domain(Student) range(xsd:string))
DatatypeProperty(major domain(Student) range(xsd:string))
DatatypeProperty(enrollmentDate domain(Student) range(xsd:date))
Class(Course)
DatatypeProperty(courseNo domain(Course) range(xsd:short) Functional)
DatatypeProperty(courseName domain(Course) range(xsd:string))
DatatypeProperty(creditHour domain(Course) range(xsd:integer))
Class(CourseLearning)
ObjectProperty(takenBy domain(CourseLearning) range(Student))
ObjectProperty(inv_takenBy domain(Student) range(CourseLearning) inverseOf(takenBy))
ObjectProperty(takesCourse domain(CourseLearning) range(Course))
ObjectProperty(inv_takesCourse domain(Course) range(CourseLearning) inverseOf(takesCourse))
Class(CourseTeaching)
ObjectProperty(taughtBy domain(CourseTeaching) range(AcademicStaff))
ObjectProperty(inv_taughtBy domain(AcademicStaff) range(CourseTeaching) inverseOf(taughtBy))
ObjectProperty(teacherOf domain(CourseTeaching) range(Course))
ObjectProperty(inv_teacherOf domain(Course) range(CourseTeaching) inverseOf(teacherOf))

**经过转换处理,输出的关系数据库模式(MySQL DDL 语句)如下:**

CREATE TABLE Staff (staffNo SMALLINT NOT NULL, staffAge INT, staffName TEXT, jobTitle TEXT, emailAddr TEXT, PRIMARY KEY (staffNo), INDEX (staffNo)) TYPE = InnoDB;

CREATE TABLE AcademicStaff (staffNo SMALLINT NOT NULL, specialty TEXT, PRIMARY KEY (staffNo), INDEX (staffNo), FOREIGN KEY (staffNo) REFERENCES Staff (staffNo) ON DELETE RESTRICT ON UPDATE CASCADE) TYPE = InnoDB;

CREATE TABLE Student (studentNo SMALLINT NOT NULL, studentName TEXT, majorIn INT, enrollmentDate DATE, PRIMARY KEY (studentNo), INDEX (studentNo)) TYPE = InnoDB;

CREATE TABLE Course (courseNo SMALLINT NOT NULL, courseName TEXT, creditHour INT, PRIMARY KEY (courseNo), INDEX (courseNo)) TYPE = InnoDB;

CREATE TABLE CourseLearning (studentNo SMALLINT NOT NULL, courseNo SMALLINT NOT NULL, PRIMARY KEY (studentNo, courseNo), INDEX (studentNo), FOREIGN KEY (studentNo) REFERENCES Student (studentNo) ON DELETE RESTRICT ON UPDATE CASCADE, INDEX (courseNo), FOREIGN KEY (courseNo) REFERENCES Course (courseNo) ON DELETE RESTRICT ON UPDATE CASCADE) TYPE = InnoDB;

CREATE TABLE CourseTeaching (staffNo SMALLINT NOT NULL, courseNo SMALLINT NOT NULL, PRIMARY KEY (staffNo, courseNo), INDEX (staffNo), FOREIGN KEY (staffNo) REFERENCES AcademicStaff (staffNo) ON DELETE RESTRICT ON UPDATE CASCADE, INDEX (courseNo), FOREIGN KEY (courseNo) REFERENCES Course (courseNo) ON DELETE RESTRICT ON UPDATE CASCADE) TYPE = InnoDB;

**此例证明了本文所提出的转换规则的可行性,验证了 Onto2RDB 算法的有效性.**

## 4 结 语

**本文在分析 OWL 本体和关系数据库模式之间概念对应的基础上,给出了从 OWL 本体到关系数据库模**

式转换的形式规则.算法实现和实例表明,从 OWL 本体到关系数据库模式的转换是可行的.

基于本文的转换规则可实现本体驱动的数据库设计,从而为全面实现自顶向下的语义 Web(门户)站点构造奠定基础.

## 参考文献:


[1] BERNERS-LEE T,HENDLER J,LASSILA O. The Semantic Web[J]. Scientific American,2001,284(5):34-43.

[2] HENDLER J. Ontologies on the Semantic Web[J]. IEEE Intelligent Systems,2002,17(2):73-74.

[3] STOJANOVIC L,STOJANOVIC N,VOLZ R. Migrating data-intensive Web sites into the Semantic Web[C]//Proc of the 17th ACM Symposium on Applied Computing. New York:ACM Press,2002:1100-1107.

[4] JACOB E K. Ontologies and the Semantic Web[J]. Bulletin of the American Society for Information Science and Technology (ASIST),2003,29(4):19-22.

[5] HORROCKS I,PATEL-SCHNEIDER P F,VAN HARMELEN F. From SHIQ and RDF to OWL:the making of a Web ontology language [J]. Journal of Web Semantics,2003,1(1):7-26.

[6] 王能斌.数据库系统教程:上册[M].北京:电子工业出版社,2002:22-238.

[7] XU Zhuo-ming,CAO Xiao,DONG Yi-sheng,et al. Formal approach and automated tool for translating ER schemata into OWL ontologies [C]//DAI H,SRIKANT R,ZHANG C. Advances in Knowledge Discovery and Data Mining (PAKDD2004 Proceedings,LNAI Vol.3056). Berlin:Springer,2004:464-475.


# Conversion from OWL ontology to relational database schema


**XU Zhuo-ming, HUANG Yong-jing**

(*College of Computer and Information Engineering, Hohai University, Nanjing* 210098, *China*)



**Abstract**: The formal definitions of OWL ontology and relational database (RDB) schema were given. Based on an analysis of the conceptual correspondence between OWL ontology and RDB schema, the rules for conversion from an OWL ontology to an RDB schema were presented. J2SE platform-based algorithm implementation and a typical example show the feasibility of the conversion.



**Key words**: schema conversion; ontology; relational database schema; OWL; Semantic Web