语义 Web 粗糙本体:定义、模型和存储方法*

王红运, 黄映辉
(大连海事大学 信息科学技术学院, 辽宁 大连 116026)

摘　要: 在讨论语义 Web 粗糙本体定义、模型的基础之上,鉴于关系数据库是目前最通用的存储介质,提出了基于关系数据库的语义 Web 粗糙本体存储方法。重点讨论了语义 Web 粗糙本体的存储模式,并通过构建一个粗糙本体并其存储到关系数据库中。实例验证了所提出存储方法的有效性和可行性。

关键词: 语义 Web 粗糙本体; 关系数据库; 粗糙本体存储
中图分类号: TP391　　**文献标志码**: A　　**文章编号**: 1001-3695(2012)07-2520-04
doi:10.3969/j.issn.1001-3695.2012.07.031

Semantic Web rough ontology: definition, model and storage method

WANG Hong-yun, HUANG Ying-hui
(*College of Information Science & Technology*, *Dalian Maritime University*, *Dalian Liaoning* 116026, *China*)

Abstract: Based on the discussed definition and model of the semantic Web rough ontology, and considering the relational database was one of the most common storage media, this paper put forward the semantic Web rough ontology storage method based on relational database. It mainly discussed the storage model. And it constructed a rough ontology and then stored the rough ontdogy into a relational database, the validity and feasibility of the storage method proposed is tested and verified through the practical example.

Key words: semantic Web rough ontology; relational database; rough ontology storage

哲学术语本体(ontology)被借用于信息科学领域源于 Gruber 在 1993 年给出的定义:"An ontology is an explicit specification of a conceptualization.",其中强调"明确的(explicit)"是本体的属性规定。可见,本体在被引入信息科学领域伊始,就被界定为精确性模型。然而,人类认知客观世界的过程与成果却是以不精确性为本质特征的。为了更好地反映这种不精确性,Ishizu 等人[1]用粗糙集理论对本体进行扩展,提出了粗糙本体(rough ontology)的概念,弥补了本体对不精确信息描述的空白;Halang 等人[2]提出了粗糙本体在现实世界中的应用意义;Keet[3]对基于粗糙概念和粗糙实例的本体工程进行了深入阐述;Hu 等人[4]讨论了基于粗糙本体的信息检索方法。

语义 Web 粗糙本体(semantic Web rough ontology)是通常意义上的语义 Web 本体的一种扩展形式,是经由本体描述语言(RDF/OWL)表示而应用于语义 Web[5]的粗糙本体。

语义 Web 本体存储(semantic Web ontology storage)在语义 Web 应用研究中极为重要,因为在基于语义 Web 本体的应用系统中,语义 Web 本体是应用系统的骨架与核心。语义 Web 本体存储的方法不仅决定了应用系统的设计思路与实现方案,而且直接影响着应用系统运行的整体性能以及后期的维护代价。语义 Web 粗糙本体是语义 Web 本体的扩展与延伸。由于存储语义 Web 本体的方法只能存储精确信息,并不适用于语义 Web 粗糙本体的存储。因此,寻求适当的方法对其进行存储是很有意义的。

本文在讨论语义 Web 粗糙本体的定义、模型的基础上,重点探讨了基于关系数据库的语义 Web 粗糙本体的存储方法、设计其存储模式;并根据所设计的存储模式将所构建的粗糙本体存储到关系数据库中,给出了具体的存储结果,验证了该存储模式的有效性和可行性。

1　语义 Web 粗糙本体

1.1　定义

粗糙本体(rough ontology)是对应于不精确信息的一类本体,由粗糙概念以及粗糙概念之间的关系组成,是在本体理论的基础上引用了粗糙集方法[6]的一种改进。粗糙本体的表示可以采用本体的通用形式 $O=\langle C,R\rangle$,但此处的 C 为粗糙概念集,R 为粗糙概念关系集。前者为三元组 $C=(U,L,Y)$。其中:U 为粗糙概念 $c(c\in C)$ 的上近似外延,表示可能满足内涵 Y 中全体属性的对象集;L 为粗糙概念 c 的下近似外延,表示肯定满足内涵 Y 中全体属性的对象集;Y 为粗糙概念 c 的内涵,表示该粗糙概念所具有的属性。

语义 Web 粗糙本体是经由本体描述语言表示而应用于语义 Web 的粗糙本体。作为语义 Web 本体的一种扩展形式,同样是支持语义 Web 实际运行的知识库;作为其构成的粗糙概念和粗糙关系,分别由精确概念和精确关系经粗糙集中的上下近似集理论[7]的处理获得。因此,它不仅能表示现实世界中的精确信息,还可以挖掘出现实世界中隐含的不精确信息。

1.2　模型

目前语义 Web 粗糙本体的构建主要采用概念格的方法。Yao 等人[8]讨论了将粗糙集近似理论与形式概念分析相结合

收稿日期: 2011-12-13; **修回日期**: 2012-02-02　　**基金项目**: 国家自然科学基金资助项目(60972090)

作者简介:王红运(1987-),女,辽宁大连人,硕士研究生,主要研究方向为智能信息处理(hywhappy@163.com);黄映辉(1955-),男,教授,博士,主要研究方向为智能信息处理.

的问题。黄冬梅等人[9]研究了在形式背景下基于属性幂集方法进行粗糙概念格的构建,进而转换为粗糙本体,但是并没有给出粗糙本体的具体模型和 OWL 表示方法。

根据 Saquer 等人[10]对形式概念分析中概念近似的讨论,定义粗糙本体中的类、属性、关系如下:

a)类 PreciseClass 为所有精确类的顶层父类,是 Thing 的直接子类。

b)类 RoughClass 为所有粗糙类的顶层父类,是 Thing 的直接子类。

c)属性 up_extent 为粗糙概念的上近似外延,属性 low_extent 为粗糙概念的下近似外延,两者均为对象属性,定义域为粗糙类,值域为精确类。

d)关系 subClassOf 为粗糙概念之间的继承关系,是基于内涵的包含关系;关系 low_Rinclusion 为粗糙实例之间的下粗包含关系;关系 up_Rinclusion 为粗糙实例之间的上粗包含关系;关系 Rinclusion 为粗糙实例之间的粗包含关系;关系 low_Requivalent 为粗糙实例之间的下粗等价关系;关系 up_Requivalent 为粗糙实例之间的上粗等价关系;关系 Requivalent 为粗糙实例之间的粗等价关系。它们都满足传递性。对于两个粗糙概念 $RC_1(U_1,L_1,Y_1)$ 和 $RC_2(U_2,L_2,Y_2)$,上述关系的定义如下所示:

若 $Y_1 \supseteq Y_2$,则 RC_1 继承 RC_2;

若 $U_1 \supseteq U_2$,则 RC_1 上粗包含 RC_2;

若 $L_1 \supseteq L_2$,则 RC_1 下粗包含 RC_2;

若 $U_1 \supseteq U_2$,且 $L_1 \supseteq L_2$,则 RC_1 粗包含 RC_2;

若 $U_1 = U_2$,则 RC_1 上粗等价 RC_2;

若 $L_1 = L_2$,则 RC_1 下粗等价 RC_2;

若 $U_1 = U_2$,且 $L_1 = L_2$,则 RC_1 粗等价 RC_2。

根据以上定义,结合语义 Web 本体模型[11],提出语义 Web 粗糙本体模型,如图 1 所示。

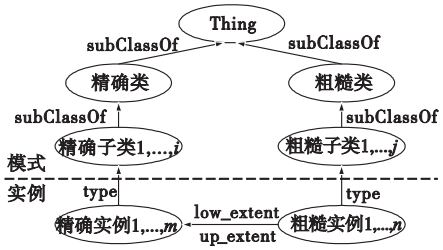

图1　语义Web粗糙本体模型

由图 1 可以看出,其与语义 Web 本体的共同点为:均由模式和实例组成,并都包含精确类和精确实例。与语义 Web 本体的区别为:语义 Web 粗糙本体中不仅包含精确类和精确实例,而且包含粗糙类和粗糙实例;此外重新定义了粗糙类之间的关系、粗糙实例与精确实例之间的关系以及粗糙实例之间的关系。

2　存储方法

存储本体的方法目前主要有基于内存[12]、基于文件[13]和基于关系数据库[14]三种。它们的共同特点是,被存储的信息都是精确的。三种存储方法比较分析如表 1 所示。

由于目前上述方法只能存储精确信息,所以不适用于语义 Web 粗糙本体的存储,需要予以改进。由表 1 可以看出,关系数据库具有结构灵活性、逻辑独立性、物理独立性、数据完整性等优势,因此本文研究了基于关系数据库的语义 Web 粗糙本体存储方法,而语义 Web 粗糙本体的存储模式是其中的关键部分,故将此作为研究重点。

表 1　三种存储方法优缺点

存储介质	优缺点
内存	存储效率高;只适合小规模本体数据
文件	易于实现;适合本体的构建与编辑;比较适用于规模较小的本体;本体规模越大,存储效率越低
关系数据库	技术发展成熟,易于建立查询;易于事务处理,便于备份;充分的关系技术支持;本体规模大时,可保持高效率

3　语义 Web 粗糙本体存储模式

根据第 1.2 节提出的语义 Web 粗糙本体模型,结合语义 Web 本体现有的存储模式,同时兼顾 OWL 的构词特征,提出适合存储语义 Web 粗糙本体的存储模式。该存储模式的设计思路如下:

a)粗糙本体中所有资源都置于表 Concepts 表中。为每个资源分配一个 ID,保证资源的唯一性。因为 OWL 语言使用 URI 来唯一地标志本体中的资源,而 URI 由命名空间和标志符两部分组成,因此资源名有 namespace 和 localname 两个字段。另外,使用 type 字段来区分资源类型,主要包括类、属性、实例三种。语义 Web 粗糙本体中,类又分为精确类(PreciseClass)和粗糙类(RoughClass),实例分为精确实例(PreciseInstance)和粗糙实例(RoughInstance)。

b)两个类之间的结构关系置于 Class_Relation 表中。其中,包含字段 cID1、cID2 和 relationType。对于精确类,relationType 主要为 subClassOf 和 equivalentClass,分别表示两个精确类之间的继承和等价关系。

c)两个属性之间的结构关系置于 Pro_Relation 表中。其中,包含字段 pID1、pID2 和 relationType。目前属性关系类型包含 subPropertyOf、equivalentProperty、inverseOf,分别表示属性之间的继承关系、等价关系和互逆关系。

d)属性的定义域和值域置于表 Pro_Field 中。由于 OWL 语言用 domain 和 range 分别对属性的定义域和值域进行描述,将这两类信息提取出来存储于 Pro_Field 表中。对于表达粗糙性的 low_extent 和 up_extent 属性而言,其定义域为粗糙类,值域为精确类。

e)属性的描述置于表 Pro_Character 中。OWL 用 TransitiveProperty、SymmertricProperty、 FunctionProperty、InverseFuntionalProperty 等描述属性的传递性、对称性、函数类型、反函数类型等特性,存储于 Pro_Character 表中。

f)属性的约束置于表 Pro_Restriction 中。OWL 用 AllValuesFrom、SomeValuesFrom、MinCardinality、MaxCardinality、Cardinality 等描述属性的约束,存储于 Pro_Restriction 表中。

g)精确实例和粗糙实例分别置于不同的表中。精确实例和粗糙实例所隶属的类置于表 Instanceof 中;精确实例具有的属性及属性值置于表 Instancevalue 中;粗糙实例的 low_extent 和 up_extent 属性及其取值置于表 RI_Extentvalue 中,包含字段 riID、epID、iID;粗糙实例内涵中的属性及属性值置于表 RI_Intentvalue 中,包含字段 riID、ipID、value。

h)精确实例之间的关系置于表 Instance_ Relation 中。OWL 引入 DifferentFrom、AllDifferentFrom、sameAs 等来描述精

确实例之间的关系。粗糙实例之间的关系置于表 RI_Relation 中。对于粗糙实例而言,本文定义了粗糙实例之间的下粗包含(low_Rinclusion)关系、上粗包含(up_Rinclusion)关系、粗包含(Rinclusion)关系、下粗等价(low_Requivalent)关系、上粗等价(up_Requivalent)关系、粗等价(Requivalent)关系。

综合上述思路,本文设计的存储模式包含 12 个关系模式,具体结构如下:

Concepts (ID, namespace, localname, type);
Class_Relation (cID1, cID2, relationType);
Pro_Relation (pID1, pID2, relationType);
Pro_Field (pID, domain, range);
Pro_Character (pID, character);
Pro_Restriction (pID, cID, restriction, value);
Instanceof (iID, cID);
Instancevalue (iID, pID, value);
Instance_Relation (iID1, iID2, relationType);
RI_Extentvalue (riID, epID, iID);
RI_Intentvalue (riID, ipID, value);
RI_Relation (riID1, riID2, relationType)。

实现语义 Web 粗糙本体存储的关系数据库模式共包含 12 张表,表之间使用 Concepts 表中 ID 字段相关联,表的主键用下画线标出,表之间的箭头表示约束关系,如图 2 所示。

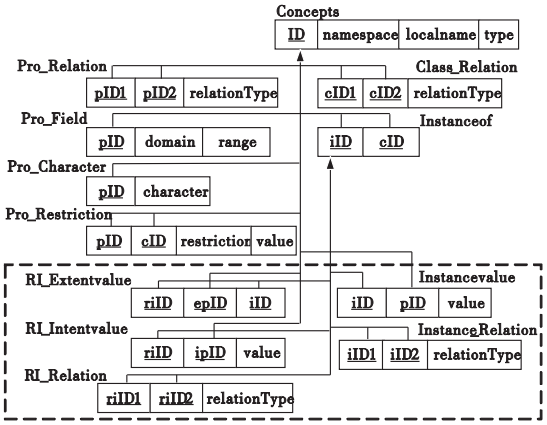

图2 语义Web粗糙本体的存储模式

4 示例

4.1 语义 Web 粗糙本体示例

本文选用概念格方法构建语义 Web 粗糙本体。首先获取疾病领域的一个形式背景,其次通过属性幂集方法[9]抽取粗糙概念,构建粗糙概念格;然后利用 OWL 对粗糙概念进行表示,最终得到疾病领域的语义 Web 粗糙本体。

根据疾病领域的信息获得一个形式背景。其中患者集合为$\{O_1,O_2,O_3\}$;条件属性集为$\{c_1$(高烧),c_2(咳嗽),c_3(咽痒),c_4(呼吸困难)$\}$,值域为$\{0,1\}$;决策属性集为$\{D$(疾病种类)$\}$,值域为$\{d_1$(感冒),d_2(咽炎),d_3(肺炎)$\}$,如表 2 所示。

表 2 形式背景

O	c_1	c_2	c_3	c_4	D
O_1	1	1	1	0	d_1
O_2	0	1	1	0	d_2
O_3	1	1	0	1	d_3

粗糙本体中的精确概念只限于形式背景中的初始概念。与表 2 相对应的精确概念如表 3 所示。

从表 2 的形式背景中,根据属性幂集方法计算得到的粗糙概念(保留上近似外延为空的粗糙概念)如表 4 所示。

表 3 精确概念表

精确概念	外延	内涵
PC1	$\{O_1\}$	$\{c_1,c_2,c_3\}$
PC2	$\{O_2\}$	$\{c_2,c_3\}$
PC3	$\{O_3\}$	$\{c_1,c_2,c_4\}$

表 4 粗糙概念表

粗糙概念	上近似外延	下近似外延	内涵
RC0	$\{\}$	$\{\}$	$\{\}$
RC1	$\{O_1,O_3\}$	$\{O_1,O_3\}$	$\{c_1\}$
RC2	$\{O_1,O_2,O_3\}$	$\{O_1,O_2,O_3\}$	$\{c_2\}$
RC3	$\{O_1,O_2\}$	$\{O_1,O_2\}$	$\{c_3\}$
RC4	$\{O_3\}$	$\{O_3\}$	$\{c_4\}$
RC5	$\{O_1,O_2,O_3\}$	$\{O_1,O_3\}$	$\{c_1,c_2\}$
RC6	$\{O_1,O_2,O_3\}$	$\{O_1\}$	$\{c_1,c_3\}$
RC7	$\{O_1,O_3\}$	$\{O_3\}$	$\{c_1,c_4\}$
RC8	$\{O_1,O_2,O_3\}$	$\{O_1,O_2\}$	$\{c_2,c_3\}$
RC9	$\{O_1,O_2,O_3\}$	$\{O_3\}$	$\{c_2,c_4\}$
RC10	$\{O_1,O_2,O_3\}$	$\{\}$	$\{c_3,c_4\}$
RC11	$\{O_1,O_2,O_3\}$	$\{O_1\}$	$\{c_1,c_2,c_3\}$
RC12	$\{O_1,O_2,O_3\}$	$\{O_3\}$	$\{c_1,c_2,c_4\}$
RC13	$\{O_1,O_2,O_3\}$	$\{\}$	$\{c_1,c_3,c_4\}$
RC14	$\{O_1,O_2,O_3\}$	$\{\}$	$\{c_2,c_3,c_4\}$
RC15	$\{O_1,O_2,O_3\}$	$\{\}$	$\{c_1,c_2,c_3,c_4\}$

按照第 1.2 节定义的粗糙概念之间的继承关系,得到粗糙概念格,如图 3 所示。

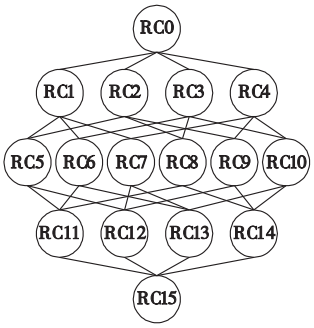

图3 粗糙概念格

将该粗糙概念格中的粗糙概念(顶节点除外)用 OWL 代码表示,即可得到语义 Web 粗糙本体。

举例说明:对于粗糙概念 RC6,它的意义是肯定包含属性集$\{c_1,c_3\}$的所有属性的精确实例集为$\{O_1\}$,至少包含一个属性的精确实例集为$\{O_1,O_2,O_3\}$,因此根据症状$\{c_1,c_3\}$可以推断该病人患有感冒的可能性最大,而患有咽炎和肺炎的可能性相对较小。用 OWL 对粗糙概念 RC6 进行表示:

```
〈owl: class rdf: ID = "#PreciseClass"/〉
〈owl: class rdf: ID = "#RoughClass"/〉
〈owl: ObjectProperty rdf: ID = "low_extent"〉
〈rdf: domain rdf: resource = "#RoughClass"/〉
〈rdf: range rdf: resource = "#PreciseClass"/〉
〈/owl: ObjectProperty〉
〈owl: ObjectProperty rdf: ID = "up_extent"〉
〈rdf: domain rdf: resource = "#RoughClass"/〉
〈rdf: range rdf: resource = "#PreciseClass"/〉
〈/owl: ObjectProperty〉
〈RoughClass rdf: ID = "RC6"〉
〈low_extent rdf:resource = "#O1"/〉
〈up_extent rdf:resource = "#O1"/〉
〈up_extent rdf:resource = "#O2"/〉
〈up_extent rdf:resource = "#O3"/〉
```

```
〈hasFever rdf:datatype = "&xsd;boolean"〉T〈/hasFever〉
〈throatItchy rdf:datatype = "&xsd;boolean"〉T〈/throatItchy〉
〈/RoughClass〉
```

4.2　语义 Web 粗糙本体存储示例

根据第 3 章所提出的存储模式,对所构建的语义 Web 粗糙本体进行存储。

将所有精确类、粗糙类、精确实例、粗糙实例、属性等存储在 Concepts 表中,存储结果如表 5 所示。

表 5　Concepts 表

ID	namespace	localname	type
c_1	http://www.sw.org/ontologies//RoughOntology	Thing	class
c_2	http://www.sw.org/ontologies//RoughOntology	PreciseClass	class
c_3	http://www.sw.org/ontologies//RoughOntology	RoughClass	class
c_4	http://www.sw.org/ontologies//RoughOntology	Disease	class
c_5	http://www.sw.org/ontologies//RoughOntology	PreciseClass1	class
c_6	http://www.sw.org/ontologies//RoughOntology	PreciseClass2	class
c_7	http://www.sw.org/ontologies//RoughOntology	PreciseClass3	class
p_1	http://www.sw.org/ontologies//RoughOntology	hasDisease	property
p_2	http://www.sw.org/ontologies//RoughOntology	low_extent	property
p_3	http://www.sw.org/ontologies//RoughOntoogy	up_extent	property
p_4	http://www.sw.org/ontologies//RoughOntoogy	hasFever	property
p_5	http://www.sw.org/ontologies//RoughOntology	hasCough	property
p_6	http://www.sw.org/ontologies//RoughOntology	throatItchy	property
p_7	http://www.sw.org/ontologies//RoughOntology	breathDifficult	property
i_1	http://www.sw.org/ontologies//RoughOntology	O_1	instance
i_2	http://www.sw.org/ontologies//RoughOntology	O_2	instance
i_3	http://www.sw.org/ontologies//RoughOntology	O_3	instance
i_4	http://www.sw.org/ontologies//RoughOntology	cold	instance
i_5	http://www.sw.org/ontologies//RoughOntology	pharyngitis	instance
i_6	http://www.sw.org/ontologies//RoughOntology	pneumonia	instance
ri_1	http://www.sw.org/ontologies//RoughOntology	RC1	instance
…	…	…	…
ri_15	http://www.sw.org/ontologies//RoughOntology	RC15	instance

将类之间的关系存储在 Class_Relation 表中,存储结果如表 6 所示。

表 6　Class_Relation 表

cID1	cID2	relationType
c_2	c_1	subClassOf
c_3	c_1	subClassOf
c_5	c_2	subClassOf
c_6	c_2	subClassOf
c_7	c_2	subClassOf

将所定义的属性及其定义域、值域存储在 Pro_Field 表中,存储结果如表 7 所示。

表 7　Pro_Field 表

pID	domain	range
p_1	c_1	c_4
p_2	c_3	c_2
p_3	c_3	c_2
p_4	c_1	boolean
p_5	c_1	boolean
p_6	c_1	boolean
p_7	c_1	boolean

将类与实例的隶属关系存储在 Instanceof 表中,存储结果如表 8 所示。

表 8　Instanceof 表

iID1	cID2	iID1	cID2
i_1	c_2	i_6	c_4
i_2	c_2	ri_1	c_3
i_3	c_2	…	…
i_4	c_4	ri_15	c_3
i_5	c_4		

将精确实例及其具有的属性、属性值存储在 Instancevalue 表中,以精确实例 i_1 为例,存储结果如表 9 所示。

将粗糙实例及其具有的 low_extent 和 up_extent 属性存储在 RI_Extentvalue 表中,这里以粗糙实例 RC6 为例,存储结果如表 10 所示。

表 9　Instancevalue 表

iID	pID	value
i_1	p_4	T
i_1	p_5	T
i_1	p_6	T
i_1	p_1	i_4

表 10　RI_Extentvalue 表

riID	epID	iID
ri_6	p_2	i_1
ri_6	p_3	i_1
ri_6	p_3	i_2
ri_6	p_3	i_3

将粗糙实例内涵中的属性及属性值存储在 RI_Intentvalue 表中,这里以粗糙实例 RC6 为例,存储结果如表 11 所示。

表 11　RI_Intentvalue 表

riID	ipID	value
ri_6	p_4	T
ri_6	p_6	T

将粗糙实例之间的关系存储在 RI_Relation 表中,存储结果如表 12 所示。

表 12　RI_Relation 表

riID1	riID2	relationType
ri_1	ri_4	Rinclusion
ri_2	ri_1	Rinclusion
ri_2	ri_3	Rinclusion
ri_2	ri_4	Rinclusion
ri_2	ri_5	Rinclusion
ri_3	ri_6	low_Rinclusion
ri_5	ri_3	up_Rinclusion
ri_5	ri_1	Rinclusion
ri_5	ri_4	Rinclusion
…	…	…

5　结束语

本文参照语义 Web 本体的理论与实践,研究了语义 Web 粗糙本体的定义、模型;根据所提出的语义 Web 粗糙本体模型以及自定义的粗糙关系等,结合 OWL 的构词特征,提出了基于关系数据库的语义 Web 粗糙本体存储方法,重点设计了适合存储语义 Web 粗糙本体的存储模式。通过实例验证了所提出语义 Web 粗糙本体模型以及存储模式的合理性与可用性。本文对语义 Web 粗糙本体的研究尚存不足,例如未弄清如何对语义 Web 粗糙本体存储的性能进行验证。

(下转第 2526 页)

(上接第 2523 页)

参考文献:

[1] ISHIZU S, GEHRMANN A, NAGAI Y. Rough ontology: extension of ontologies by rough sets[J]. Human Interface, 2007, 45(57): 56-62.

[2] HALANG W, ZHAO Yi, WANG Xia. Rough ontology mapping in E-business integration[J]. Studies in Computational Intelligence, 2007, 37(2): 75-93.

[3] KEET C M. Ontology engineering with rough concepts and instances [C]//Proc of the 17th International Conference on Knowledge Engineering and Knowledge Maragement. 2010: 503-513.

[4] HU Jun, LI Zhi-lu, GUAN Chun. A method of rough ontology-based information retrieval[C]//Proc of IEEE International Conference on Granular Computing. 2008: 296-299.

[5] BERNERS-LEE T, HANDLER J, LASSILA O. The semantic Web [J]. Scientific American, 2001(5): 34-43.

[6] 申锦标. 不完备信息系统中一种拓展粗糙集模型[J]. 计算机应用研究, 2009, 26(6): 2101-2103.

[7] 刘财辉, 苗夺谦. 基于矩阵的粗糙集上、下近似求解算法[J]. 计算机应用研究, 2011, 28(5): 1628-1630.

[8] YAO Yi-yu, CHEN Yao-hua. Rough set approxirnations in formal concept analysis[C]//Proc of the 23rd International Meeting of the North Ameriean Fuzzy Inforrnation Processing Society. 2004: 73-78.

[9] 黄冬梅, 朱慧. 粗糙形式概念分析在海洋本体构建中的应用[J]. 计算机科学, 2008, 35(4): 6-8.

[10] SAQUER J, DEOGUN J S. Concept approximations based on rough sets and similarity measures [J]. International Journal Applied Mathematics and Computer Science, 2001, 11(3): 655-674 .

[11] 张斌, 刘增良, 余达太, 等. 基于形式概念分析与统计理论的本体构建模型[J]. 计算机应用研究, 2011, 28(1): 111-113.

[12] KIRYAKOV A, OGNYANOV D, MANOV D. OWLIM: a pragmatic semantic repository for OWL[C]//Proc of the 1st International Workshop on Scalable Semantic Web Knowledge Base Systems. 2005: 182-192.

[13] WOOD D, GEARON P, ADAMS T. Kowari: a platform for semantic Web storage and analysis [C]//Proc of the 14th International World Wide Web Conference. 2005: 24-27.

[14] JEONG D, SHIN H, BAIK D K, *et al.* An efficient Web ontology storage considering hierarchical knowledge for Jena-based applications [J]. Journal of Information Processing Systems, 2009, 5(1): 11-18.