

文章编号:1008-1542(2016)04-0416-07

doi:10.7535/hbkd.2016yx04015

基于抽取规则和本体映射的领域 XML 语义集成

李华昱,张培颖,肖 晗

(中国石油大学(华东)计算机与通信工程学院,山东青岛 266580)

摘 要:油气井工程领域中存在大量的 XML 文档,传统的 XML 集成方案无法提供面向语义的信息查询,导致数据利用率不高。针对油气井 XML 文档 WeXML 语义集成与查询应用需求,提出一种基于抽取规则和本体映射的语义集成方法。首先定义一系列类、属性抽取规则,分别将 WeXML Schema 中的元素、属性映射为 WeOWL 本体中的类和属性;然后,利用实例转换算法将 WeXML 文档转换为本体实例数据;由于 WeOWL 提供有限的局部语义模型,需要在两者之间建立语义映射,并借助 WeOWL 中的术语对全局语义模型中的类和属性进行解释,进而提供面向领域全局本体的语义查询。通过构建 WeXML 数据语义集成原型系统,对提出的转换规则、转换算法和映射规则进行了验证。

关键词:计算机信息管理系统;抽取规则;本体映射;领域 XML;语义集成

中图分类号:TP391

文献标志码:A

Domain XML semantic integration based on extraction rules and ontology mapping

LI Huayu, ZHANG Peiying, XIAO Han

(College of Computer and Communication Engineering, China University of Petroleum, Qingdao, Shandong 266580, China)

Abstract: A plenty of XML documents exist in petroleum engineering field, but traditional XML integration solution can't provide semantic query, which leads to low data use efficiency. In light of WeXML(oil&gas well XML data) semantic integration and query requirement, this paper proposes a semantic integration method based on extraction rules and ontology mapping. The method firstly defines a series of extraction rules with which elements and properties of WeXML Schema are mapped to classes and properties in WeOWL ontology, respectively; secondly, an algorithm is used to transform WeXML documents into WeOWL instances. Because WeOWL provides limited semantics, ontology mappings between two ontologies are then built to explain class and property of global ontology with terms of WeOWL, and semantic query based on global domain concepts model is provided. By constructing a WeXML data semantic integration prototype system, the proposed transformational rule, the transfer algorithm and the mapping rule are tested.

Keywords: computer information management system; extraction rules; ontology mapping; domain XML; semantic integration

收稿日期:2015-10-13;修回日期:2015-11-24;责任编辑:陈书欣

基金项目:山东省自然科学基金(ZR2014FQ018);中央高校基本科研业务费专项资金(14CX02030A)

作者简介:李华昱(1977—),男,山东寿光人,副教授,博士,主要从事语义 Web 与数据集成方面的研究。

E-mail:lhyzj@upc.edu.cn

李华昱,张培颖,肖 晗.基于抽取规则和本体映射的领域 XML 语义集成[J].河北科技大学学报,2016,37(4):416-422.

LI Huayu, ZHANG Peiying, XIAO Han. Domain XML semantic integration based on extraction rules and ontology mapping[J]. Journal of Hebei University of Science and Technology, 2016, 37(4): 416-422.

XML 是领域数据重要的信息表示和存储形式,也是应用系统之间进行数据交换的主要标准。然而,由于 XML 缺乏足够的语义信息,传统的 XML 集成方案无法提供基于领域术语的语义查询,而利用语义 Web 与本体技术,对 XML 文档进行语义集成,是一种有效的解决方案,也是领域数据集成的研究热点之一。

目前,XML 语义集成主要采用 2 类方法:1)定义一系列映射规则,从 XML 文档或 XML Schema 中抽取本体,并利用映射信息将 XML 数据转换为本体实例数据,再通过构建基于语义的查询访问接口,实现语义集成。针对此类方法,刘显敏等^[1]提出了一种基于 XML 键规则语义实体抽取方法,以此提高 XML 查询效率;XML2OWL 系统首先采用 XSLT 描述语义抽取规则建立 OWL 本体,再通过实例转换过程生成本体实例数据^[2];FERDINAND 等^[3]提出了 XML Schema 映射为 OWL 本体和将 XML 转换为 RDF 数据的 2 种独立方法;XU 等^[4-5]提出了一种基于映射规则的领域本体构建方法,该方法能够从 XML Schema 抽取语义信息并构建 XML 本体。乔卫^[6]设计了 XML 语义信息抽取模型,通过 4 个功能模块解决了 XML 的语义信息抽取问题,特别是 XML 结构所隐含的语义。2)在 XML Schema 与已有本体之间建立语义映射,利用本体模型描述 XML 数据并通过实例转换或者查询重写,提供面向本体模型的语义查询。例如,WEESA^[7]和 XMLTOWL^[8]采用 XML 和 XSLT 定义 XML Schema 与本体之间的映射规则,自动将 XML 文档转换为本体实例;文献^[9]采用 2 个子过程,借助 XSLT,分别将 XML Schema 和 XML 文档映射为本体模型和实例数据;LEHTI 等^[10]提出的本体映射方法中,不提供实例数据转换功能,而是利用查询重写,将基于语义的查询转换为面向 XML 的 Xquery 查询。CRUZ 等^[11-12]和 XIAO^[13]提出了一个基于本体的 XML 语义数据集成框架,该框架首先将局部 XML 本体进行合并生成全局 RDF 本体,再构造一个模式映射表,记录全局和各个局部本体之间的映射,进而实现语义集成。

石油、材料领域中存在若干个 XML 模型^[14],如 PetroXML,WellLogML,WITSM^[15]和 MatML^[16]等,针对油气井 XML 文档 WeXML,基于文献^[17]和文献^[18]中的 MatML 语义集成方法及相关领域数据集成方案^[19-20],提出了语义解决方案,首先通过定义抽取规则从 WeXML Schema 中抽取油气井局部本体 WeOWL,并通过实例转换部件将 WeXML 文档转换为 WeOWL 本体实例数据;由于 MatOWL 是从 WeXML Schema 中抽取而来,仅提供油气井的基本语义信息,需要利用逻辑规则,在 WeOWL 与领域本体之间建立语义映射,并借助语义映射信息和逻辑规则,实现面向领域语义概念模型的油气井 XML 数据的语义集成与语义查询。

1 WeXML 语义集成框架

1.1 WeXML 基本结构

WeXML(well XML)是一种描述油气井日常生产相关数据的 XML 文档,通过 XML 数据类型、complexType 和 simpleType,WeXML Schema 定义了 Dept, Well 和 Geology 3 类一级元素和 75 种其他子元素和属性,WeXML 基本结构如图 1 所示。

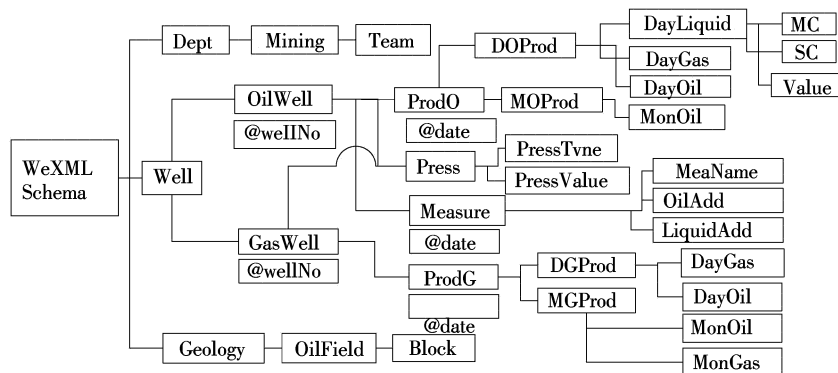


图 1 WeXML 基本结构
Fig. 1 Basic structure of WeXML

图 1 中,实线矩形框表示元素类型,以@开头的矩形框表示属性。Dept, Well 和 Geology 是 3 类一级元素类型:Dept 描述油气井单位隶属关系,包括矿区和小队;Well 是 WeXML 中的核心类型,包括油井和气井

2 种类型,记为 OilWell 和 GasWell,每种类型定义产量、压力、泵工艺等子元素,分别描述日产油、日产液、日产气、月产油、月产液、月产气、工艺措施、含水、含砂、压力类型、泵压力、泵排量、泵径、泵深等相关数据;Geology类型描述油田、区块和储层等信息。

1.2 相关定义

定义 1 WeXML Schema=(CT, ST),ST={st|st is xsd:simpleType},CT={ct|ct is xsd:complexType}=E∪ATT,其中:E={e|e is a xsd:Element},ATT={att|att is a xsd:Attribute}

定义 2 XSDType 为 XML Schema 类型集合,XSDType={string, date, dicimal, integer,...}

建立函数集合 FSchema = {getEles, getAttrs, getXSDType}:

- 1) getEles(ct), getAttrs(ct) 分别获取 ct 的子元素和属性;
- 2) getXSDType(att/st) 分别获取 att/st 的名称、类型和对应的 XML Schema 数据类型。

定义 3 WeOWL=(C, OP, DP, I, A^o)

WeOWL (well OWL ontology)是从 WeXML Schema 抽取而来,其中,C 是概念集合,OP 是对象属性集合,DP 是数据类型属性集合,I 是实例集合,A^o 是公理集合。针对 WeOWL 定义,建立函数集合 F_{WeOWL}={hasDomain, hasRange, hasSuperclass}:

- 1) $\forall p \in OP \cup DP, \forall x, y \in C \cup XSDType, p.hasDomain(x)$ 表示属性 p 的 Domain 中包含类 x, $p.hasRange(y)$ 表示属性 p 的 Range 中包含类 y;
- 2) $\forall c, x \in C, x=hasSuperclass(c)$ 用于表示类 x 为类 c 的一个父类。

由于 WeOWL 是从 WeXML Schema 抽取而来,无法表达比 WeXML Schema 更丰富的语义信息,需要将 WeOWL 与领域本体建立映射关联,才能提供更为有效的语义集成服务,为此,引入油气井工程领域本体 WeDOWL(well engineering domain OWL ontology)。

定义 4 WeDOWL=(C_d, OP_d, DP_d, I_d, A_d^o)

WeDOWL 是领域全局本体,描述地质、日常生产、工艺措施和隶属单位相关概念和概念之间的语义关联。其中,C_d 是概念集合,OP_d 是对象属性集合,DP_d 是数据类型属性集合,I_d 是实例集合,A_d^o 是公理集合。

定义 5 $y=f_{C/P}(x)$ 为语义抽取函数: $x \in ST \cup CT \cup XSDType, y \in C \cup OP \cup DP, x$ 为 WeXML Schema 中的类型或者属性; $y=f_{C/P}(x)$ 表示 y 是由 x 抽取生成的 WeOWL 类、对象属性或者数据类型属性

1.3 语义集成框架

WeXML 语义集成框架如图 2 所示,包括 WeXML 数据层、本体实例转换层、语义集成业务层和应用层。

1) WeXML 数据层:包括由各个采油生产单位提交的 WeXML 文档。

2) 本体实例转换层:参照语义抽取规则,通过实例转换算法,将 WeXML 中的油气井数据转换为 WeOWL 实例数据,即填充 WeOWL 实例集合 I。

3) 语义集成业务层:针对 WeXML Schema 结构,定义语义抽取规则,将 WeXML Schema 中元素和属性映射为 WeOWL 中的类、属性以及层次关系并建立 WeOWL 本体;在 WeOWL 与 WeDOWL 之间建立语义映射,提供基于领域概念模型的语义查询。

4) 应用层:基于 WeOWL 语义概念模型,提供可视化查询构造界面,能够根据查询需求构造 SPARQL^[21] 查询语句;通过读取规则库中的逻辑规则并借推理机,实现面向 WeOWL 实例数据进行语义查询。

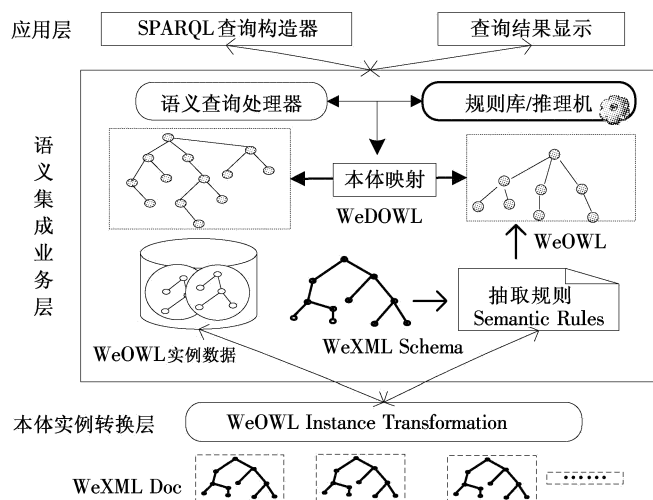


图 2 WeXML 语义集成框架

Fig. 2 Semantic integration framework of WeXML

2 WeOWL 构建

2.1 语义抽取规则

通过对 WeXML Schema 结构进行分析,并参照 HUNTER 等^[22]提出的方法,分别针对类、层次结构、对象属性、数据对象属性和基数定义 5 项抽取规则,从 WeXML Schema 中抽取语义信息并建立 WeOWL 本体,规则如下。

规则 1 Class 抽取规则

$$\forall t \in CT \cup ST_R \rightarrow (\exists c \in C) \wedge (c.id = getName(t))$$

规则说明:对于任意 CT 或 ST_R 类型 t ,它将被转换为 WeOWL 中的一个类 c ,并将 t 的名称作为 c 的标志符。该规则简记为 $c = f_C(t)$ 。

规则 2 Class Hierarchy 生成规则

规则 2.1 $\exists c1 \in C \wedge (c1.id = 'Well') \wedge \exists c2 \in C (c2.id = 'OilWell' \vee c2.id = 'GasWell') \rightarrow c1 = \text{hasSuperclass}(c2)$

规则 2.2 $\exists c1 \in C \wedge (c1.id = 'ProdO') \wedge \exists c2 \in C (c2.id = 'DOProd' \vee c2.id = 'MOProd') \rightarrow c1 = \text{hasSuperclass}(c2)$

规则 2.3 $\exists c1 \in C \wedge (c1.id = 'ProdG') \wedge \exists c2 \in C (c2.id = 'DGProd' \vee c2.id = 'MGProd') \rightarrow c1 = \text{hasSuperclass}(c2)$

规则说明:2.1 将 'Well' 类设置为 'OilWell' 和 'GasWell' 的父类,即将“油井”和“气井”设置为“井”的子类;2.2 将 'ProdO' 类设置为 'DOProd' 和 'MOProd' 的父类,即将“日产油”和“月产油”设置为“产油”的子类;2.3 将 'ProdG' 类设置为 'DGProd' 和 'MGProd' 的父类,即将“日产气”和“月产气”设置为“产气”的子类。

定义子类集合 $C_{child} = \{OilWell, GasWell, DOProd, MOProd, DGProd, MGProd\}$, C_{child} 包含上述 6 个子类。

规则 3 ObjectProperty 抽取规则

$$\forall ct \in CT \wedge (\forall x \in \text{getEles}(ct) \cup \text{getAttrs}(ct)) \wedge (x \notin C_{child}) \wedge (\text{getType}(x) \in CT \cup ST_R) \rightarrow (\exists op \in OP) \wedge (op = f_{OP}(x)) \wedge (op.id = getName(x)) \wedge (op.hasDomain(f_C(ct))) \wedge (op.hasRange(f_C(\text{getType}(x))))。$$

规则说明:对于任意 CT 类型 ct ,如果其包含的元素或属性 x 是 CT 类型或是 ST 类型,并且 x 不包含在 C_{child} 集合,则 x 将被抽取为对象属性 op 。同时,类 $f_C(ct)$ 被添加到 op 的 domain 中,类 $f_C(\text{getType}(x))$ 被添加到 op 的 range 中。该规则简记为 $op = f_P(ct, x)$ 。

规则 4 DataTypeProperty 抽取规则

$\forall ct \in CT \wedge (\forall x \in \text{getElements}(ct) \cup \text{getAttributes}(ct)) \wedge (\text{getType}(x) \in ST) \wedge (\text{getXSDDType}(x) \in XSDDType) \rightarrow (\exists dp \in DP) \wedge (dp = f_{C/DP}(x)) \wedge (dp.id = getName(x)) \wedge (dp.hasDomain(f_{C/P}(ct))) \wedge (dp.hasRange(\text{getXSDDType}(x)))。$

规则说明:对于任意 CT 类型 ct ,如果其包含的元素或属性 x 为 ST 简单类型或为 XML Schema 数据类型, x 将被抽取为一个数据类型属性 dp 。同时,类 $f_{C/P}(ct)$ 将被添加到 dp 的 domain 中,类 $f_{C/P}(\text{getXSDDType}(x))$ 将被添加到 dp 的 range 中。该规则简记为 $dp = f_P(ct, x)$ 。

规则 5 PropertyCardinality 生成规则

$\forall ct \in CT \wedge (\forall x \in \text{getEles}(ct) \cup \text{getAttrs}(ct)) \wedge (\exists c = f_C(ct)) \wedge (\exists p = f_P(ct, x))$, 定义如下子规则:

规则 5.1 $\neg \text{hasAtt}(x.minOccurs) \wedge \neg \text{hasAtt}(x.maxOccurs) \rightarrow p.Cardinality(c) = 1$

规则 5.2 $(x.minOccurs = m) \wedge \neg \text{hasAtt}(x.maxOccurs) \rightarrow p.MinCardinality(c) = m$

规则说明:如果元素 x 没有定义 $minOccurs$ 和 $maxOccurs$ 属性,则 p 对于类 c 的 Cardinality 设定为 1;如果 x 的 $minOccurs$ 数值为 m ,并且没有定义 $maxOccurs$ 属性,则 p 对于类 c 的 Cardinality 设定为 m 。

基于 1.2 定义的函数和 2.1 定义的抽取规则,通过“本体抽取”模块从 WeXML Schema 中获取结构信息并建立 WeOWL 本体,WeOWL 结构如图 3 所示,其中 WeXML 是 WeOWL 的 root 类。

在利用抽取规则建立 WeOWL 本体过程中,设置映射表 $MapTab(\text{mapping table})$ 描述 WeXML Schema

中元素、属性和 WeOWL 中类、属性之间的对应关系。

2.2 WeOWL 实例数据转换

生成 WeOWL 后,需要利用本体实例转换部件(WeOWL instance transformation)并参照语义抽取规则,通过转换算法对 WeXML 进行遍历,依次读取元素、属性和数值信息,生成 WeOWL 实例数据。实例转换算法如表 1 所示,输入为 WeXML 文档 wexml.xml, WeOWL 本体模型 weowl.owl,输出结果为包括本体实例数据的 weowl'。

TraverseXML 主要包括 Transform 和 TransfromST 2 个函数。其中,Transform 采用递归形式遍历 wexml.xml 文档,针对 CT 和 ST 类型元素分别进行转换;TransformST 专门处理 ST 类型元素的转换。

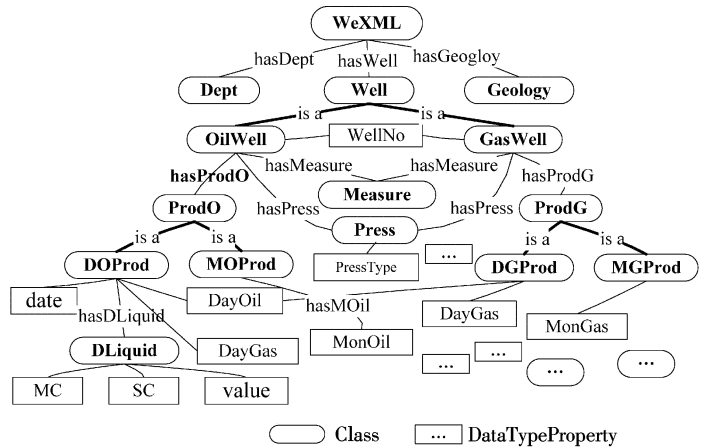


图 3 WeOWL 本体结构
Fig. 3 Structure of WeOWL ontology

表 1 WeOWL 实例转换算法

Tab. 1 WeOWL instance data transform algorithm

Algorithm: TraverseXML()
Input: wexml.xml, weowl.owl
Output: weowl'
<pre> 1 rootNode = parser.getDocument("wexml.xml"); 2 nodes = rootNode.getChildNodes(); 3 for (int i = 0; i < nodes.count; i++) 4 transform(nodes[i]); 5 Procedure void Transform(node){ 6 if (node is CT) { 7 c=getClassFromMapTab(node.name); 8 attrs=getAttrs(node); eles=getEles(node); 9 for (int i=0;i<attrs.count;i++) { 10 dtp=getDTPFromMapTab(attrs[i].name); 11 I.addProperty(c,dtp,attrs[i].value); } 12 for (int j=0;j<eles.count;j++) { 13 if(eles[j] is CT){ 14 Transform(eles[j]); I.addOP(c,eles[j]);} 15 if(eles[j] is ST){ 16 TransformST(eles[j]); I.addDTP(c,eles[j]); } 17 }else 18 TransformST(node); } 19 Procedure void TransformST(node){ 20 if (eles[j] is ST & eles[j] is Eumeration) 21 for (int k=0;k<eles[j].count;k++) 22 I.addDTProperty(c,dtp,eles[j][k].value); 23 if (eles[j] is ST & eles[j] is ! Eumeration) 24 I.addDTProperty(c,dtp,eles[j].value); } </pre>

算法首先读取 wexml.xml 文档并将文档根节点记为 rootNode;然后,针对 rootNode 每一个子节点执

行 Transform 函数(1-4 行), Transform 函数根据传入的 node 参数类型,分别进行处理:如果为 CT 类型,从 MapTab 映射表中获取 node 对应的类记为 c(5-7 行),并遍历 node 的每一个属性和子元素(8 行),其中,属性直接转换为数据类型实例数据(9-11 行),子元素如果为 CT 类型,递归执行 Transform 函数并转换为对象类型实例数据(13-14 行),如果为 ST 类型,执行 TransformST 函数(15-16 行)。TransformST 函数根据 ST 是否为枚举类型,分别处理 ST 类型的实例数据转换(19-24 行)。

3 本体映射

由于 WeOWL 是从 WeXML Schema 中抽取而来,仅提供局部、有限的语义信息,需要在 WeOWL 和 WeDOWL 之间建立语义映射,进而提供面向领域全局语义模型的语义查询。

本体映射由用户通过逻辑规则进行定义,定义的逻辑规则将存入到规则库中。通过参照逻辑规则,推理引擎能够利用 WeOWL 术语对 WeDOWL 中的类和属性进行解释,自动地将 WeOWL 本体实例数据归类为 WeDOWL 实例。逻辑规则定义形式如下所示:

$$\text{weowl:Oilwell}(? x) \wedge \text{weowl:hasMeasure}(? x, ? y) \wedge \text{weowl:hasMeasureDate}(? y, ? z) \rightarrow \text{wedowl:MeaOilWell}(? x)。$$

以上规则表示:如果在 WeOWL 中 x 为油井,并且在某一时间进行过工艺措施,那么 x 可以作为 WeDOWL 中的“措施油井”类型。

例如:查询“2015 年 1 月以后所有措施油井井号、增油产量及实施的工艺措施类型”,可以通过设定以下逻辑规则:

$$\text{weowl:Oilwell}(? x) \wedge \text{weowl:hasWellNo}(? x, a) \wedge \text{weowl:hasMeasure}(? x, ? b) \wedge \text{weowl:hasMeasureDate}(? b, ? c) \wedge \text{weowl:measureDate}(? c, '2015.1') \wedge \text{weowl:oilAdd}(? b, ? d) \wedge \text{weowl:hasMeaNName}(? b, ? e) \rightarrow \text{wedowl:MeaOilWell}(? x) \wedge \text{wedowl:wellName}(? x, ? a) \wedge \text{wedowl:oilAdd}(? x, d) \wedge \text{wedowl:MeaType}(? x, ? b)。$$

用户可以通过查询构造器设置如下 SPARQL 语句:

```
Select ? a ? b ? c
```

```
Where { ? a rdf:type MeaOilWell. ? a wedowl:wellName ? b. ? a wedowl:oilAdd ? c. }
```

查询语句提交给语义查询处理器后,处理器将利用规则库和推理机,自动将 WeOWL 中相关实例数据归类为 MeaOilWell 实例,完成语义查询。

4 系统实现

基于 Eclipse 开发平台,开发了 WeXMLSemSys 原型系统。WeXMLSemSys 采用 Java Swing 设计查询操作界面,将 Dom4j 作为 WeXML 文档的解析工具,利用 Jena API 完成 Sparql 语义查询和逻辑推理。该原型系统主要包括 3 个功能:浏览 WeXML 数据内容并将 WeXML 文档转换为 WeOWL 实例数据;通过 Sparql 语句对数据进行查询;利用 Jena 推理机和逻辑规则实现推理并返回语义查询结果。

在实际应用中,WeXMLSemSys 读取了胜利油田某采油厂 2015 年的 103 个 WeXML 文档,每个文档包含 1-2 口油井的生产数据,通过实例转换算法产生了基于领域全局本体的 WeDOWL 的实例数据,能够有效支持面向语义的数据查询,提高数据使用效率。

5 结 语

近年来,很多学者关注 XML 语义集成和 XML 与本体之间映射的相关研究,并取得了一定的应用成果。通过比较,本文提出的方法具有以下特点:1) WeOWL 本体构建方法并不是将 XML 文档直接与特定本体进行映射,而是首先根据抽取规则,自动构建局部本体,然后再与领域全局本体进行语义映射;2) 该方法是针对特定领域的 XML 文档 WeXML,除了与相关工作类似的集成方法之外,还根据 WeXML Schema 和领域特点,加入了特定的抽取规则和映射方法。通过构建原型系统 WeXMLSemSys,对本文提出的方法进行了验证,取得了良好的应用效果。

参考文献/References:

- [1] 刘显敏, 李建中. 基于键规则的 XML 实体抽取方法[J]. 计算机研究与发展, 2014, 51(1): 64-75.
LIU Xianmin, LI Jianzhong. Key-based method for extracting entities from XML data[J]. Journal of Computer Research and Development, 2014, 51(1): 64-75.
- [2] BOHRING H, AUER S. Mapping XML to OWL ontologies[J]. Leipziger Informatik-Tage, 2005, 72: 147-156.
- [3] FERDINAND M, ZIRPINS C, TRASTOUR D. Lifting XML Schema to OWL[J]. Lecture Notes in Computer Science, 2004, 3140: 354-358.
- [4] XU Jiuyun, WANG Hongquan, DUAN Youxiang, et al. Automatically bootstrapping OWL ontologies from domain-specific XML documents[J]. Journal of Computational Information Systems, 2007, 3(3): 1269-1276.
- [5] XU Jiuyun, WANG Hongquan, DUAN Youxiang. Auto-building OWL ontology from XML data sources[J]. Journal of Computational Information Systems, 2006, 2(3): 1051-1057.
- [6] 乔卫. 基于领域本体的 XML 语义信息抽取的研究与实现[D]. 武汉: 武汉理工大学, 2009.
QIAO Wei. Research and Implementation of XML Semantic Information Based on Domain Ontology[D]. Wuhan: Wuhan University of Technology, 2009.
- [7] REIF G, JAZAYERI M, GALL H. Towards semantic web engineering: WEESA-Mapping XML Schema to ontologies[C] // In Workshop on Application Design. New York: IEEE Computer Society, 2004: 105-109.
- [8] KOBEISSY N, GENET M G, ZEGHLACHE D. Mapping XML to OWL for seamless information retrieval in context-aware environments[C] // International Conference on Pervasive Services. [S. l.]: [s. n.], 2007: 349-354.
- [9] 靖争. XML/Schema 到 OWL 本体映射的研究[D]. 沈阳: 东北大学, 2008.
JING Zheng. Research on Mapping XML/Schema to OWL Ontology[D]. Shenyang: Northeastern University, 2008.
- [10] LEHTI P, FANKHAUSER P. XML data integration with OWL: Experiences and challenges[C] // International Symposium on Applications and the Internet. [S. l.]: IEEE Computer Society, 2004: 160-167.
- [11] CRUZ I F, XIAO H, HSU F. An ontology-based framework for XML semantic integration[J]. Ideas, 2004, 26(1): 217-226.
- [12] CRUZ I F, XIAO H, LAB A. The role of ontologies in data integration[J]. International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, 2005, 13(4): 245-252.
- [13] XIAO H. Query Processing for Heterogeneous Data Integration Using Ontologies[D]. Chicago: University of Illinois, 2006.
- [14] 王宏琳. 石油勘探开发信息技术发展态势——数据集成、应用集成和知识集成[J]. 石油工业计算机应用, 2007, 15(1): 6-11.
WANG Honglin. The trend of information technology: Data integration application integration and knowledge integration in oil exploration and development[J]. Computer Applications of Petroleum, 2007, 15(1): 6-11.
- [15] WELLSITE. Information Transfer Standard Markup Language [EB/OL]. <http://www.witsml.org>, 2007-03-16.
- [16] BEGLEY E F. MatML Version 3.1 Schema [EB/OL]. <http://www.matml.org>, 2011-02-23.
- [17] ZHANG Xiaoming, HU Changjun, LI Huayui. Semantic query on materials data based on mapping MatML to an OWL ontology[J]. Data Science Journal, 2009, 8: 1-17.
- [18] 张晓明. 领域科学数据语义集成模型及映射[D]. 北京: 北京科技大学, 2009.
ZHANG Xiaoming. Domain-specific Semantic Model and Mapping for Scientific Data Integration[D]. Beijing: University of Science and Technology Beijing, 2009.
- [19] 刘紫玉, 杨国霞, 李学会. 基于本体的多专业领域知识语义检索系统研究[J]. 河北科技大学学报, 2011, 32(5): 471-476.
LIU Ziyu, YANG Guoxia, LI Xuehui. Research in semantic retrieval system for knowledge of multiple majors domain based on ontology [J]. Journal of Hebei University of Science and Technology, 2011, 32(5): 471-476.
- [20] 张立岩, 张世民. 基于语义相似度的主观题评分算法研究[J]. 河北科技大学学报, 2012, 33(3): 263-265.
ZHANG Liyan, ZHANG Shimin. Algorithm of subjective item marking based on semantic similarity[J]. Journal of Hebei University of Science and Technology, 2012, 33(3): 263-265.
- [21] PRUDHOMMEAU E, SEABORNE A. SPARQL Query Language for RDF [EB/OL]. <http://www.w3.org/TR/rdf-sparql-query>, 2008-01-15.
- [22] HUNTER J, LITTLE S, SCHROETER R. The application of semantic web technologies to multimedia data fusion within science[C] // Semantic Multimedia and Ontologies. Springer: [s. n.], 2008: 207-226.