基于 XML Schema 的航运领域本体构建方法①

袁逸涛, 王晓峰

(上海海事大学 信息工程学院, 上海 201306)

摘 要: EDI 电子数据交换作为一种信息技术, 目前已经广泛应用在航运领域. 然而由于各个国家乃至各个航运公司所用的EDI报文格式和系统都不相同, 导致数据流通性差. 为了解决航运领域信息共享存在的语义异构问题, 本文将本体的概念引入到了航运领域之中, 并且提出了一种基于XSLT转换技术和XPath路径语言的本体构建方法, 实现了将航运业务的 XML Schema 结构文档中半自动化的转换成 OWL 语法的本体文档, 建立了航运领域本体. 实验表明, 该方法能够大大提高本体的构建效率, 并在一定程度上保证了本体的正确性.

关键词: 航运信息共享; XSLT; XML Schema; OWL; 本体构建

Shipping Domain Ontology Construction Method Based on XML Schema

YUAN Yi-Tao, WANG Xiao-Feng

(College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China)

Abstract: EDI (electronic data interchange), as a kind of information technology, has been widely used in the field of shipping. However, since various countries and various shipping companies use different EDI packet formats and systems, the ability of data circulation is poor. In order to solve the problem of semantic heterogeneity in shipping field information sharing, this paper introduces the concept of ontology into the field of shipping and proposes an ontology construction method based on the XSLT and XPath technology. It realizes the semi-automatization conversion from the XML Schema document to the OWL syntax ontology document and builds the maritime domain ontology. The experimental results prove that this method improves the efficiency of ontology construction, and ensures the correctness of the ontology.

Key words: shipping information sharing; XSLT; XML Schema; OWL; ontology construction

近年来, 随着信息时代的到来, 各行各业的数据量呈现一个爆发性增长的态势. 而在航运领域, 由于航运业务涵盖地域的广泛性以及业务时间的漫长, 建立高效的信息共享平台是十分必要的. 目前在航运领域主要是使用 EDI(Electronic Data Interchange)报文的方式来进行信息交流, 它实际上是通过建立有效数据流, 利用计算机网络来实现相关部门的数据在计算机应用系统之间的快速和可靠的切换以及自动处理, 针对异构 EDI 的问题, 各大航运公司在使用 EDI 系统的同时结合了 XML 的技术[1]. 在 EDI 系统中引入了 XML 平面文件的概念. 但是 XML 平面文件的技术只能够解决数据在系统上的异构, 然而在航运领域, 数据所展现出的语义含义往往很重要, 它对航运数据在语义和知识上的异构不能进行处理.

在异构数据共享领域, 一些学者从哲学领域引入了本体(Ontology)的概念. 本体方法的提出解决了在信息共享中的语义异构的问题. 通过语义分析的方法, 本体可以对数据进行分类, 把相同的概念或语义数据提取, 并进行相关数据的整合, 挖掘出数据的深层语义. 并且在语义分析中所建立的本体可以重用, 避免了统一领域往复的领域知识分析. 我们希望将本体技术运用到航运领域的信息共享平台之中来[2].

目前在本体领域主要有以下两种构建方法[3]:

① 手工构建: 在行业专家的指导下利用诸如 Prot

① 基金项目:上海市科委重点项目(14511107402)

收稿时间:2016-04-17;收到修改稿时间:2016-05-08 [doi:10.15888/j.cnki.csa.005514]

égé 等编辑工具, 在构建工程中遵循一定的结构方法, 采用手工编辑的方法进行领域本体的构建. 这种方法准确性最高, 但是要耗费大量的时间和人力物力.

② 本体学习: 利用机器学习和统计等技术, 以自动或半自动的方法, 从已有的数据资源中获取期望的本体. 同时本体学习根据不同结构的数据源分为了三种类型, 分别是基于非结构化数据、半结构化数据和结构化数据的本体学习. 与手动进行本体构建相比, 本体自动学习的方法虽然节省了效率, 但是没有达到本体方法语义正确度的标准, 也就是说最终构建的本体与人工理解的本体存在语义差异.

根据在航运公司的调研结果, 目前很多航运公司的 EDI 系统中使用了 XML 技术. EDI 报文会先转换成为 XML 格式的平面文件再存入内部数据库之中. 同时, 为了对 XML 平面文件进行校验定义了 XML schema 文档(即 XSD 文件), 它作为一种标准文件定义了 XML 文件的结构和内容. 由于 XSD 文件在制定的过程中参照了领域专家的意见, 所以我们把 XSD 文件作为航运本体的主要概念来源.

我们提出了一种 XSD2OWL 的方法, 先建立 XSD 文件到本体概念的映射关系, 再利用 XSLT 转换技术实现 XML 结构文档到 OWL 本体语言的转换, 为最终的利用本体实现异构 EDI 交换奠定基础.

1 XML Schema向OWL 本体转换方法

1.1 XML Schema 与本体语言 OWL

XML Schema 是用来定义 XML 文档的数据和结构的, 它指定了 XML 文档的一系列规则以确保 XML 文档的一致性和有效性.

某航运公司一份舱单的 XSD 文件 Manifest_Arrival_Air.xsd 如下所示, 我们会基于这份 XSD 文件来实现 XML Schema 向 OWL 本体转换的方法:

```
<xs:complexType name="Manifest">
    <xs:annotation>
        <xs:documentation>空运出口预配舱单修改</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Head" type="Head">
            <xs:annotation>
                <xs:documentation> 报 文 头 (循环次数: 1)</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Declaration">
            <xs:annotation>
                <xs:documentation> 报 文 体 (循环次数: 1)</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:complexContent>
                    <xs:extension base="Declaration"/>
                </xs:complexContent>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

根据上述的文档内容, 我们发现其中 XML Schema 主要组成元素有:

① 简单元素(Element): 简单元素指那些仅包含文本的元素.

② 复合类型(ComplexType): 复合元素指包含了其他元素或属性的 XML 元素.

③ 属性(Attribute): 属性在一定意义上就是元素的语义, 它用基本的数据类型描述了元素的意义.

④ 注释(Annotation): 对于复合元素的中文语义描述.

⑤ 特定顺序(Sequence): 定义复合元素中元素的出现顺序, 具有不可改变性.

XML Schema 是由 XML 编写的, 所以具有很强的扩展性, 另外 XML Schema 最重要的能力之一就是对数据类型的支持, OWL 中内置的数据类型是和 XML Schema 一样的.

目前在本体研究领域, 本体的描述语言有很多, 有 RDF 和 RDF-S、OIL、DAML、OWL 等. 其中目前应用最广的就是 OWL 语言, 它最大的优势是它是基于 XML 语法的, 这个特性也决定了它可以与所有基于 XML 语法的文档通过一定的转换规则进行转换.

我们通过分析一份航运业务的 XSD 文档, 初步设

计一个获取航运领域知识的 OWL 本体的主要有以下三个组成元素[4]:

① 个体(Individual): 个体代表某个知识领域中的一个需要被概念化提取的对象, 在 OWL 中同一个个体可以拥有不同的名称, 在这种情况下, 个体之间的关系必须表达清楚, 标注它们是否相同. 个体是本体之中最基本的组成单位.

② 属性(Property): 属性表述了不同个体之间的一个二元关系. 比如说是否相同, 或者表达从属关系等. 属性在不同的个体之间建立了一种可靠的数据联系. 在OWL中有两个类型的属性: 对象属性和数据类型属性, 分别建立了个体到个体的关系和个体到数值的关系.

③ 类(Class): 类似于面向对象方法中类的概念. 本体中的类就是具有相同概念的个体的集合, 这些个体共享了某些相同的属性. 同时类与类直接也存在着一个用属性表述的二元关系.

1.2 XSD2OWL 转换规则

从以上描述可以看出, OWL 本体与 XML Schema 文档不仅在语法上有着相似之处, 同时它们的组成元素以及结构上存在着一定的对应关系, 我们可以根据语义先对航运业务进行本体建模, 再制定相应元素之间的转换规则[5-7]. 具体规则如下:

① 数据类型: OWL本体是由是由XML语法编写, 所以直接使用了XML Schema文件中的内置数据类型.

② XML Schema 文档中的类型与 OWL 本体中的类: XML Schema 文档中主要有简单元素和复杂类型, 在这里, 根据实际业务需求, 我们只研究复杂类型. 复杂类型主要有简单元素(Element)构成, 而在 OWL 语法中, 类(Class)是个体的集合, 所以我们建立一个对应的映射关系. 把复杂类型作为概念提取出来, 作为本体的类.

③ XML Schema 文档中的简单元素以及属性(Attribute)和 OWL 中的数据类型属性(Datatype Property):通过研究上述文档我们发现, 在航运业务中的基本上所有的简单元素都可以理解为表述复杂类型的一个属性. 那么, 相对应的我们可以把简单元素转换为 OWL 本体中的数据类型属性. 在现有的 XSD 业务文档中, XML Schema 属性(Attribute)使用的很少, 不过为了系统的兼容性, 我们也将它定义转换为OWL本体中的数据类型属性.

④ XML Schema 中的简单元素和复杂类型的嵌套关系表述为 OWL 本体中的对象属性(Object Property). 在XML Schema文档中主要使用的是属性与类的关系, 属性与其子属性的关系以及类与子类的关系[8].

⑤ XML Schema 文档中的 Sequence(特定顺序)用来定义复杂类型的语境, 它表示了在该复合类型中的元素是有序的, 我们在转换为 OWL 文档的时候不能改变元素的顺序. 在 OWL 本体中没有与之相对应得组件存在, 所以我们在OWL本体中定义一个类(Class)来完成相应的功能(可以将这个类命名为 Sequence).

最后, 我们定义一个如表 1 所示的映射关系.

表 1 XSD2OWL 映射关系

XML Schema	OWL
Complex Type 复合类型	Class 类
Element 简单元素	Datatype Property 数据类型属性
Attribute 属性	Object Property 属性
Annotation 注释	Comment 描述
Sequence 特定顺序	自定义 Class 类(squence)
嵌套关系	Object Property 对象属性

2 航运本体建模及映射

定义映射规则之后, 我们根据《中华人民共和国海关进出境运输工具舱单管理办法》和《海关总署关于调整及新增进出境水运和空运运输工具货运舱单等电子数据格式的公告》(海关总署公告 2010 年第 70 号), 根据中国海关进出境舱单报文 XML Schema 文件定义了航运业务本体的数据模型.

本文初步把航运本体可定义为:

Shipping-ontology=<C, H, P, P^R, I>

本体中的各参数的描述如下[9]:

C 代表类(Class), 本体中个体元素的集合, 在航运业务中可以表示为目的地信息, 出发地信息等;

H 代表类之间的层次关系(Hierarchy), 主要是指父类与子类关系, 描述航运业务流程中的层次结构, 在航运业务信息中, 由于各业务的交互, 需要将层次关系清楚描述, 比如说提单数据段中往往包含货物数据段;

P 代表属性(Property), 在航运业务中绝大部分都是数据属性, 描述个体信息;

P^R 代表对属性的限制(Restriction of Property), 主要是对属性取值的类型、范围以及最多最少个数等的

限制, 如发货日期属性只能为“*年*月*日”格式的日期, 且到货日期的值必须要在发货日期的值之后;

I 代表个体(individual), 即某个概念(类)的具体表达.

定义本体之后, 我们结合一份实例文件演示本体建立的完整过程.

我们根据航运舱单的 XSD 文档内容可以从中提取出如图 1 的结构.

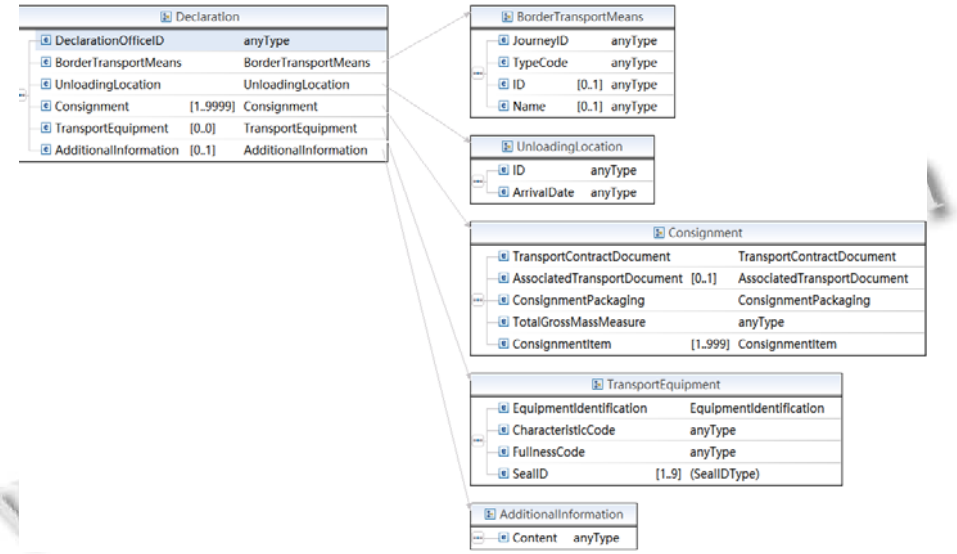

图 1　某公司 XSD 业务文档结构

根据这个文档结构, 我们在航运业务专家的指导下, 定义一些概念数据模型, 并与源数据中的定义建立如图 2 所示的映射关系.

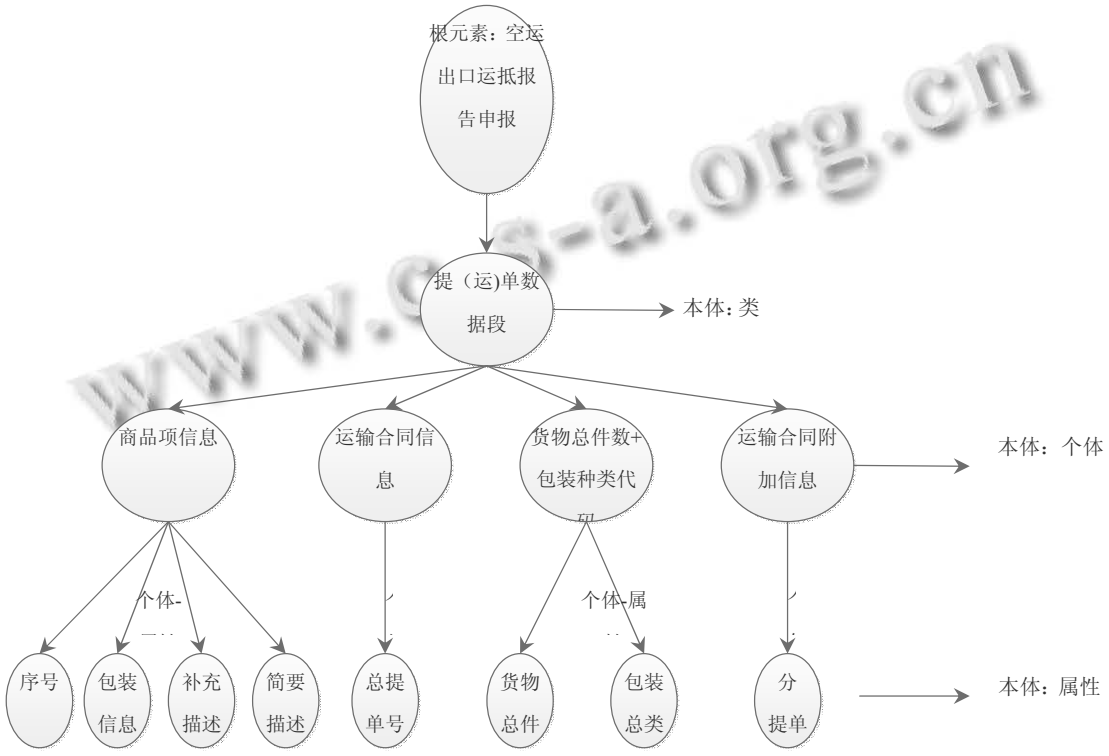

图 2　业务本体数据模型

3 转换工具的设计与实现

基于上述的映射规则和数据模型, 本方案的实现基于 java 平台, 因为 java 平台中具有丰富的 XML 文件处理的 API, 根据实际业务需求, 建立航运业务数据模型, 编写 XSLT 文件来实现 XSD 文档到 OWL 文档的转换, XSLT 使用 XPath 在 XML 文档中查找信息. XPath 被用来通过元素和属性在 XSD 文档中进行导航, 最后生成航运领域本体.

主要使用的开发 IDE 是 MyEclipse, 所用框架技术为 Servlet, 基于 B/S 架构[10,11]完成一个开源的 JAVA WEB 项目.

开发技术介绍:

XSLT 用于将一种 XML 文档转换为另外一种 XML 文档. 由于本体描述语言 OWL 和航运 EDI 系统的平面文件都是基于 XML 语法的, 我们可以利用 XSLT 转换技术完成两种文档的转换. 在使用 XSLT 技术中, 我们需要制定一个模板, 模板的内容基于上文所描述的转换规则. 在模板文件中, 利用 XPath 进行元素的导航与匹配, 并定义用于转换的指令元素, 用这些指令书写的文档称作样式表, 其本身也是一个文档. 我们把源文件 XSD 文档和制定的样式表文件输入 XSLT 处理器, 生成航运本体的 OWL 文档[12,13]. 该过程如图 3 所示.

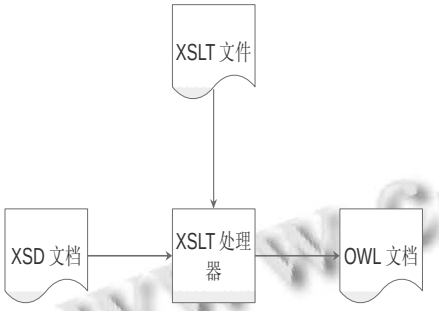

图 3 XSLT 处理模块

主要算法流程如图 4 所示.

方法实现(XSLT 文件实例):

```
//Mapping from named complexTypes to owl:Classes 复合类型转换为类
<xsl:for-each select="dyn:evaluate(concat('/', $xsdPrefix, 'schema/', $xsdPrefix, 'complexType'))">
    <xsl:if test="@name != ''">

        <xsl:if test="$deb &gt; 0">
            <xsl:message>Class found, converted from named complexType: '<xsl:value-of select="@name"/>'</xsl:message>
        </xsl:if>
        <xsl:call-template name="createClass">
            <xsl:with-param name="class" select="@name"/>
        </xsl:call-template>
    </xsl:if>
</xsl:for-each>
```

原始 XSD 文件与生成后的 OWL 文件片段:

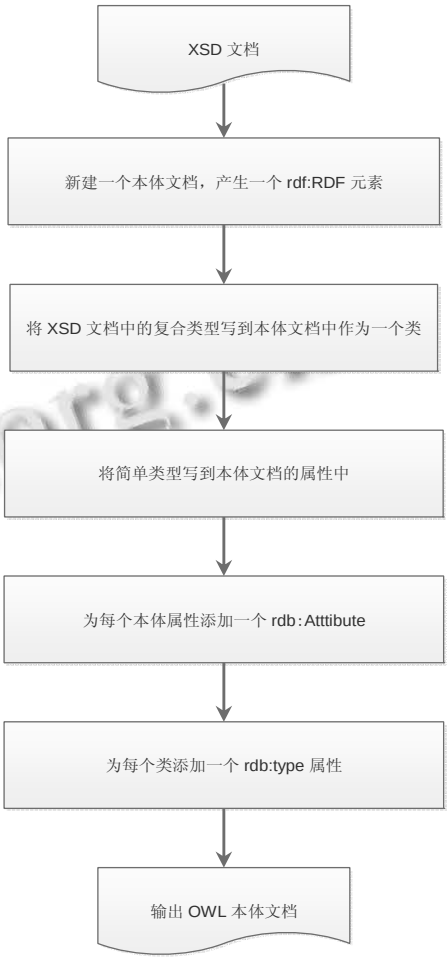

图 4 算法流程图

<xs:complexType name="BorderTransportMeans"> <xs:annotation> <xs:documentation>运输工具数据段</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="JourneyID"> <xs:annotation> <xs:documentation>航次航班编号</xs:documentation></xs:annotation> </xs:element> <xs:element name="TypeCode"> <xs:annotation> <xs:documentation>运输方式代码</xs:documentation> </xs:annotation> </xs:element> <xs:element name="ID" minOccurs="0"> <xs:annotation> <xs:documentation>运输工具代码</xs:documentation> </xs:annotation> </xs:element>	<Class rdf:about="Manifest#BorderTransportMeans"> <rdfs:subClassOf rdf:resource="/ontologies/Manifest#Declaration"/> <rdfs:comment> 运输工具数据段 </rdfs:comment> </Class> <DatatypeProperty rdf:about="Manifest#JourneyID"> <rdfs:comment> 航次航班编号 </rdfs:comment> <rdfs:domain rdf:resource="Manifest#BorderTransportMeans"/> </DatatypeProperty> <DatatypeProperty rdf:about="Manifest#TypeCode"> <rdfs:comment> 运输方式代码 </rdfs:comment> <rdfs:domain rdf:resource="Manifest#BorderTransportMeans"/> </DatatypeProperty>

从上述文件片段看出，XSD 文件中的 complexType 类型“运输工具数据段”被转换成了 OWL 文件中的 class 类型, XSD 文件中的 element 类型“运输方式代码”等被转换成了 OWL 文件中的 DatatypeProperty 类型.

我们将所生成的 OWL 文件在本体编辑工具 Protégé 中打开，得到了如图 5 所示的本体模型图，说明生成的本体满足了 OWL 本体的数据规则，具有可用性，我们在接下来的研究工作中可以利用所生产的本体去做进一步的数据共享研究.

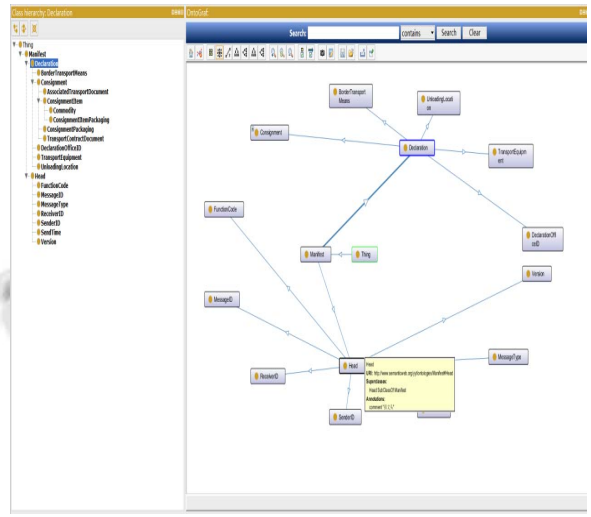

图 5 Manifest_Arrival_Air.OWL 本体模型图

4 结语

本文首先分析了航运领域异构数据交换的研究现状以及现存的一些问题，引入了本体的概念. 基于航运公司提供的 XML Schema 文件，重点分析了 XML Schema 语法和 OWL 语言的组成结构，在此基础上进行模式匹配，再建立航运业务概念模型，通过 XSLT 和 XPath 的技术将 XSD 文档转换为 OWL 预言的本体文档，实现了从半结构化数据 XSD 文档中建立航运领域本体的过程.

在方法实现之后，下一步工作是完善航运领域本体的数据库，将所建立的本体应用到航运异构信息共享平台中，以本体作为数据语义的依托，利用本体对不同公司的 XML 文档进行语义标注[14]，建立不同文档之间的映射关系，生成它们之间的转换文件，实现不同语义的业务文档的自动映射和转换.

参考文献

1 虞祺.基于 XML 的 EDI 在航运企业的实施研究[学位论文]. 上海:上海交通大学,2004.

2 杜小勇,李曼,王珊.本体学习研究综述.软件学报,2006, 17(9):1837–1847.

3 徐红升,张瑞玲.基于粗概念格模型的电子商务领域本体的

构建研究.计算机工程与科学,2014,36(3):530–535.
4 胡鹤,刘大有,王生生.Web 本体语言 OWL.计算机工程,2004,30(12):1–2.
5 靖争.XML/Schema 到 OWL DL 本体映射的研究[硕士学位论文].沈阳:东北大学,2008.
6 李为冲.XML 到 OWL 文档生成方法研究[硕士学位论文].青岛:中国石油大学,2008.
7 谭介平.XML 数据到 OWL 本体的转换方法的研究[硕士学位论文].南昌:华东交通大学,2011.
8 许卓明,顾华建,倪玉燕,等.UML 类图向 OWL 本体转换工具的设计与实现.河海大学学报(自然科学版),2007,35(4):477–482.
9 李鹏.面向地质勘查的多源异构数据集成关键技术研究[博士学位论文].北京:中国地质大学,2013.
10 Tudorache T, Nyulas C, Noy NF, et al. Web Protégé: A collaborative ontology editor and knowledge acquisition tool for the Web. Semantic Web, 2013, 4(1): 89–99.
11 王艳敏,谢强,丁秋林.基于本体和 Web Services 的数据交换平台.计算机技术与发展,2010,20(5):112–116.
12 Kay M, Fernández MF, Boag S, et al. XML path language (XPath) 2.0. Efficient & Flexible Search in Large Scale Distributed Systems, 2007(January): 505–506.
13 刘红,王晔,潘晨,等.基于 XML 和 XSLT 的通用报表系统的设计与实现.计算机应用与软件,2011,28(2):142–144.
14 黄洋.基于 SSH 架构与本体的异构数据集成技术研究[硕士学位论文].北京:北京邮电大学,2015.