第 38 卷 第 4 期
Vol.38 No.4

计 算 机 工 程
Computer Engineering

2012 年 2 月
February 2012

• 软件技术与数据库 •

文章编号：1000—3428(2012)04—0076—03 文献标识码：A 中图分类号：TP311

数据交换中基于本体的语义冲突消解方案

王 倩，王 辉

(河南科技大学电子信息工程学院，河南 洛阳 471003)

摘 要：为解决数据交换过程中的语义冲突问题，提出一种基于本体的语义冲突消解方案。利用 ER 模型实现关系模式到 XML 模式的语义映射，采用本体对经过初步语义转换的 XML Schema 进行语义标注。实验结果表明，该方案能减少由自然语言或符号不同引起的歧义，在一定程度上消除语义冲突。

关键词：数据交换；语义异构；语义冲突；冲突消解；本体；语义标注

Semantic Conflict Resolution Scheme Based on Ontology in Data Exchange

WANG Qian, WANG Hui

(College of Electronic Information Engineering, Henan University of Science and Technology, Luoyang 471003, China)

【Abstract】To resolve the semantic conflicts in the process of data exchange, an improved semantic conflict resolution scheme is presented. It uses ER model to realize the semantic mapping from relational schema to XML Schema, and uses ontology to make semantic annotations on XML Schema which is translated semantically and preliminary. Experimental results show that this scheme can reduce the ambiguity caused by the difference of natural language or symbol, semantic conflicts between the heterogeneous data sources are resolved to some extent.

【Key words】data exchange; semantic heterogeneity; semantic conflict; conflict resolution; ontology; semantic annotation

DOI: 10.3969/j.issn.1000-3428.2012.04.025

1 概述

在现今的信息系统和电子商务领域，异构系统间的信息交换变得日益频繁。分析各部门的相关数据，可以发现数据异构性主要集中在以下 3 个方面：系统异构，结构异构，语义异构。系统异构指数据所依赖的应用系统的差别，如硬件平台、操作系统、数据库管理系统、并发控制、通信能力、和访问方式的不同。结构异构指数据在存储模式上的差异。语义异构指数据源在信息资源的描述上存在着语义上的区别。这些语义上的不同可能引起各种冲突，从简单的命名冲突，如同义异名、同名异义，到复杂的结构语义冲突，如不同的模式表达同样的信息，语义冲突使信息交换变得复杂化。

目前，异构性问题已在不同程度上得到解决。其中，系统异构和结构异构的处理技术已经日渐成熟。中间件技术(如 CORBA、DCOM、OGSA-DAI)、Web Service 技术已经较好地解决了不同平台及软件系统间的互操作问题；XML 及其相关技术(如 XML Schema)克服了语法与数据模式的异构问题，成为 Internet 上互换信息的关键技术。但 XML Schema 主要用来确定 XML 文档的结构，不能用来确定元素的具体含义以及元素之间的语义联系。虽然它用一种层次方式组织元素，但这种层次方式并不能提供元素之间的关系，而仅仅提供了一种语法，复用一些简单的结构以构造更复杂的结构，不能表达这些元素之间的语义关系。为了给 Schema 增加语义，必须将它的结构和能够表达语义信息的载体联系起来。本体(Ontology)中的概念和属性可以成为这一载体，因为本体能够提供对某一领域概念的语义描述。

综上所述，本文提出一种语义冲突消解方案，通过关系模式到 XML 模式的语义映射以及引入本体对映射文件进行语义标注来解决语义冲突的问题。

2 研究基础

2.1 语义冲突

语义冲突是指不同信息资源之间存在着语义上的区别而引起的各种冲突。例如，一个系统内用“author”代表书刊作者，另一个系统用“writer”表示作者，这 2 个概念属于异名同义。在一个系统内用“on”表示“开”这种状态，而另一个系统内则表示一个实体位于在另一个实体之上，这属于同名异义；对日期类型的属性，它可以用“DD/MM/YYYY”和“YYYY-MM-DD”2 种格式表示；人的姓名在一个库中是姓名，在另外一个数据库中则采用姓、名分开定义等。

2.2 本体

本体最早是一个哲学上的概念，后来被引入到人工智能领域，文献[1]给出了定义：Ontology 是共享概念模型明确的形式化规范说明。本体的广泛应用需要一种描述本体并使得它能够进行信息交换的标准语言。本体可以用自然语言来描述，也可以用框架、语义网络等来描述。具体描述本体的方法很多，目前，使用最普遍的方法是 RDF、RDFS、DAML+OIL、OWL 等，本文选用 OWL 语言[2]来描述本体。

本体与 XML Schema 的显著不同在于：XML Schema 只

基金项目：国家自然科学基金资助项目(61070247)；河南省教育厅自然科学研究计划基金资助项目(2010A520017)；河南省科技攻关计划基金资助项目(112102210186)

作者简介：王 倩(1986－)，女，硕士研究生，主研方向：数据集成；王 辉，教授

收稿日期：2011-07-18 **E-mail：**xerstudio@163.com

隐式地包含了对概念的描述，用结构之间的包含关系表示结构之间存在某种关系，而本体则显式地描述了概念及它们之间的关系。本体就是利用这些关系进行逻辑推理来获取数据的内在语义信息，生成语义映射关系。

3　模型设计

本文从 2 个层面逐步消解异构数据源间的语义冲突。语义冲突消解模型如图 1 所示。在实现关系模式向 XML 模式转换时，采用间接模式转换方法，即不直接从关系模式向 XML 模式进行逻辑转换，而是借助关系模式的 ER 图，由 ER 图映射出 XML Schema 图，由 XML Schema 图通过语义映射最终生成 XML Schema。

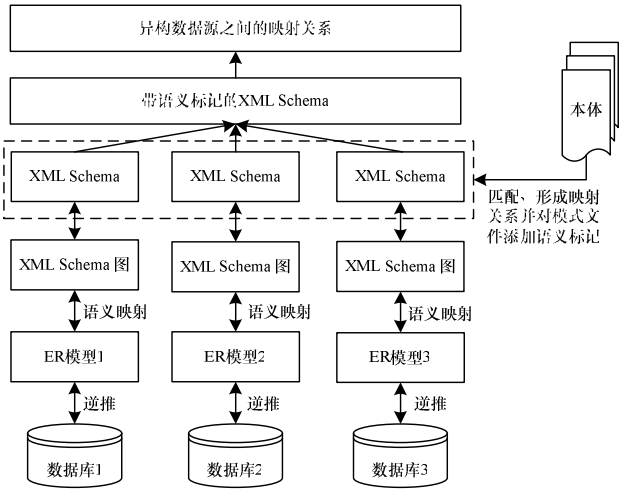

图 1　语义冲突消解模型

为了更进一步地增强 XML Schema 的语义表达能力，首先构建一个本体来实现知识的共享和重用。各个异构数据源的 XML Schema 文件可以通过与本体的映射为相应的元素加上语义标记。这样，加上语义标注的 XML Schema 文件之间便产生了映射关系，异构数据源之间也产生了带有语义信息的映射关系。

4　语义冲突消解方案

4.1　关系模式到 XML 模式的语义映射

在建立数据库系统时，首先要对它进行形式化结构抽象，用数据模型构建数据表示和数据操作的构架。ER 模型是经典的概念数据模型，它用简单的图形方式(ER 图)描述数据，再通过一定规则将 ER 图转换成关系模式集，最终完成对数据库的概念设计。在实现关系模式向 XML 模式转换时，本文提出间接模式转换方法，即不直接从关系模式向 XML 模式进行逻辑转换，而是借助关系模式的 ER 图，由 ER 图映射出 XML Schema 图，由 XML Schema 图最终生成 XML Schema。关系模式到 XML 模式的语义映射过程如图 2 所示。

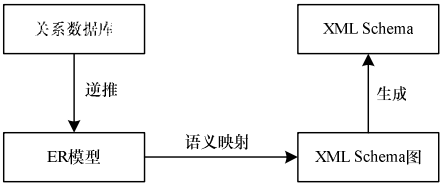

图 2　关系模式到 XML 模式的语义映射过程

由 XML Schema 图向 XML Schema 的转换规则如下：

(1)将 XML Schema 图中根结点直接映射为 XML Schema 的根元素。

(2)将关系模式中的关系表映射为 XML Schema 的元素。

(3)将实体属性映射为 XML Schema 中的属性。在实际映射时，当扫描到多值属性时，可将该属性作为当前元素的子元素映射，减少部分数据冗余。

4.2　本体语义标注过程

为 XML Schema 添加语义的过程为：将 XML 中的个体与本体模型中的概念、关系、属性等本体中的各要素建立连接，使之具有明确的语义。这个过程也称为基于本体的语义标注。本文采用 XML Schema 的扩展机制使用与 Schema 元素映射的本体中的概念和属性作为标记添加到 XML Schema 中。采用对元素进行注解的方法表达其语义，并使用 XML 命名空间提供的扩展机制，将语义注解以 semantic 前缀来声明。本文在已有研究[3-5]的基础上，提出利用本体为 XML Schema 添加语义标注，步骤如下：

(1)对 XML Schema 中 ComplexType、SimpleType 类型的结构采用本体中的概念进行标记；概念标记完成之后，对概念具有的属性进行标记，概念的匹配为其属性的匹配建立了一个上下文环境，在这个上下文环境中可以对其属性进行准确度比较高的自动标记。

(2)不同的 XML Schema 通常会在被本体标记的过程中采用不同粒度的概念标记，在 XML Schema 采用较粗粒度概念时，如果在其结构中没有特定元素进行细粒度概念的区分，则可直接用该粗粒度概念标记；如果使用了其他元素进行细粒度的区分，则用 classifier 标记这个元素。

(3)对 XML Schema 中的 Restriction、Extension 扩展机制可以使用本体中的父子类进行标记，主外键则可使用本体中的属性进行标记。

5　实验与分析

为了更好地说明本文所提出模型进行语义冲突消解的过程，本文以教务管理系统为例，抽取关系模式如下：

Department(dno, dname, address)

Teacher(tno, firstName, lastName, sex, birthday, zipCode)

Student(sno, sname, dno, major)

从关系模式中推导出 ER 模型，数据库实例的 ER 图如图 3 所示。将 ER 模型映射成的 XML Schema 图，过程如图 4 所示。

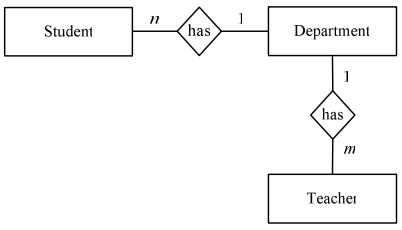

图 3　数据库实例的 ER 图

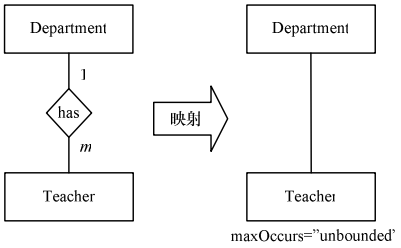

图 4　ER 图映射为 XML Schema 图的过程

根据 XML Schema 图向 XML Schema 的转换规则，得出转换后的 XML Schema 代码片段如下：

```
<xsd:element name="Department">
```

```
    <xsd:complexType>
      <xsd:attribute name="dno" type="ID"/>
        …
      </xsd:complexType>
    </xsd:element>
  <xsd:element name="Teacher">
    <xsd:complexType>
      <xsd:sequence>
          <xsd:element ref="Department">
      </xsd:sequence>
      <xsd:attribute name="tno" type="ID"/>
      <xsd:attribute name="dno" type="IDREF"/>
        …
    </xsd:complexType>
  </xsd:element>
```

可以看出，转换后的 XML Schema 文件在一定程度上体现了 2 个关系模式之间的语义关系，实现了关系模式向 XML Schema 的语义转换。

从校园人事库抽取关系模式如下：

Personnel(pno, name, sex, birthday, postCode)

该关系模式与教务库中抽取的关系模式：

Teacher(tno, firstName, lastName, sex, birthday, zipCode)

存在一定程度的语义冲突。2 种异构模式元素比较如表 1 所示。2 种异构模式的元素映射关系如表 2 所示。通过表 1 和表 2 的分析可知，两者之间存在属性异构冲突、数据格式冲突、数据类型冲突。

表 1 2 种异构模式的元素比较

Personnel	Teacher
pno	tno
name	firstName, lastName
sex	sex
birthday	birthday
postCode	zipCode

表 2 2 种异构模式的元素映射关系

Personnel	Teacher
pno	tno
name	connect(firstName, lastName)
sex	sex
birthday	birthday
postCode	zipCode

将 2 种异构模式的 Schema 结构与本体中的概念进行匹配，参考已经成熟的模式匹配技术产生 XML Schema 与本体之间的映射，提高本体映射精确度的方法可参考文献[6]。

首先对关系模式进行分析，发现模式 Personnel 中的 name 属性是模式 Teacher 中的属性 firstName、lastName 的组合。在本体知识片断中对属性 firstName、lastName 进行并运算，即 owl:unionOf；2 种模式中对“邮编”采用了不同的表示方式，两者是等价的关系，在 OWL 中用 owl:equivalent Class 描述 2 个完全相同的实例。从实验中抽取的本体 OWL 描述片段如下：

```
<owl:Class rdf:ID="name">
<owl:unionOf rdf:parseType="Collection">
  <owl:Class rdf:about="#firstName" />
  <owl:Class rdf:about="#lastName" />
</owl:unionOf>
</owl:Class>
<owl:Class rdf:ID="postCode">
<owl:equivalentClass rdf:resource="&zipCode;postCode"/>
</owl:Class>
```

根据上面构建的本体文件 ontol.owl，对产生的 XML Schema 中的结构进行语义标记，下面是为 2 个模式文件加了语义标记的代码片段。本文限于篇幅，不列出全部实验代码。

关系模式 Personnel(pno, name, sex, birthday, postCode)经过语义标记后的 XML 模式代码片段如下：

```
<xs:element name="name" type="xs:string"
            semantic:type="&ontol;# name"/>
<xs:element name="sex" type="xs:string"
            semantic:type="&ontol;# sex"/>
<xs:element name="postCode"type="xs:string"
            semantic:type="&ontol;# postCode" />
```

关系模式 Teacher(tno, firstName, lastName, sex, birthday, zipCode)经过语义标记后的 XML 模式文件片段如下：

```
<xs:element name="firstName" type="xs:string"
            semantic:type="&ontol;# name"/>
<xs:element name="lastName" type="xs:string"
            semantic:type="&ontol;# name"/>
<xs:element name="sex" type="xs:string" minOccurs="0"
            semantic:type="&ontol;# sex"/>
 <xs:element name="zipCode" type="xs:string"
            semantic:type="&ontol;# postCode" />
```

各数据模式加上语义标记后，利用模式中的语义标记，各模式的元素和结构之间就可以通过语义标记的匹配以及本体的参与，实现模式之间的映射。映射关系的建立是通过 2 种不同数据模式通过匹配技术分别与本体进行匹配产生的，通过用户的参与进行语义标注，然后在本体的参与下，2 种带有语义标注的数据模式之间进行匹配。加上语义标记后 2 个 XML Schema 之间的映射关系如图 5 所示。

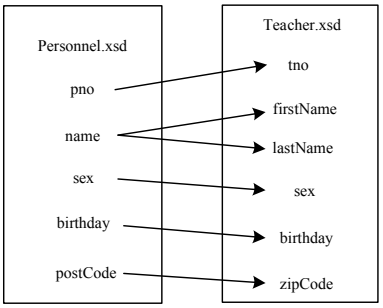

图 5 加上语义标记后的 XML Schema 映射关系

上述实验最后得出 XML Schema 文件，由于对存在语义冲突如属性异构冲突、数据格式冲突、数据类型冲突的部分做了标记，因此在进行数据交换过程中减少了由自然语言或符号不同引起的歧义，从而在一定程度上消除了语义冲突。

6 结束语

本文提出一种基于本体的语义冲突消解方案。实验结果证明，该方案使得异构数据源间的语义交互成为可能，在一定程度上解决了语义异构问题。下一步的工作是将现有方案应用到信息系统的实际数据交换中，根据实际问题进一步完善本文方案。 (下转第 81 页)