

# 基于依赖关系图模型的本体演化影响分析

尚 晋, 张 睿, 吕 帅, 刘 磊  
(吉林大学 计算机科学与技术学院, 长春 130012)

**摘要:** 基于描述逻辑子语言 SHOIN( $D$ ) 定义本体中各元素间的依赖关系, 再通过依赖关系建立本体依赖关系图模型, 并给出了变化影响频度的概念及相应的计算方法。

**关键词:** 本体演化; 本体变化; 描述逻辑; 影响频度

**中图分类号:** TP311    **文献标志码:** A    **文章编号:** 1671-5489(2012)01-0089-06

## Impact Analysis of Ontology Evolution Based on Dependency Graph Model

SHANG Jin, ZHANG Rui, LÜ Shuai, LIU Lei  
(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

**Abstract:** Studying the semantics of the description logic SHOIN( $D$ ), we clarified the dependency relations between the elements in the evolving ontology and proposed a dependency graph model for the ontology. Based on this model, the concept of impact frequency was defined and an algorithm was proposed to compute it.

**Key words:** ontology evolution; ontology change; description logic; impact frequency

本体是概念模型的明确规范说明<sup>[1]</sup>, 本体为语义驱动的应用系统提供了共享的知识模型. 在语义 Web 中, 基于本体的 Web 应用程序很大程度上依赖于支撑此应用的领域本体, 随着应用领域的变化或用户需求的改变, 本体也在不断变化<sup>[2]</sup>. 本体演化是本体对变化及时地调整及对变化一致性的管理<sup>[3]</sup>. 由于本体的普遍性和重要性, 本体演化已成为一个重要的研究领域<sup>[4]</sup>. Stojanovic 等<sup>[5-6]</sup> 为保持本体一致性提出两种方法: 过程式方法和描述式方法. 在过程式方法中, 提出了演化策略, 本体工程师需要理解并参与本体演化的过程, 该方法过分依赖于人工干预, 因而在描述式方法中简化了变化需求的说明, 使本体工程师只需关注本体演化的结果即可, 从而减少了人工干预.

由于本体元素间存在依赖关系, 因此当一个元素变化时, 会对其他元素产生影响, 这种影响范围的量化可作为启发式搜索本体一致性状态的一种启发函数, 同时这种量化也对减小依赖于本体的应用程序影响有重要作用. 基于启发式搜索算法, 刘晨等<sup>[7]</sup> 提出一种可减小变更影响范围的本体演化算法, 该算法分析了本体实体和服务间的依赖关系, 并提出了量化变更影响范围的数学公式, 给出了新的启发搜索算子, 通过搜索一条变更影响值最小的执行路径减轻变更执行过程对现有服务产生的影响. 但对于变化影响的量化仅考虑了相邻节点的影响, 没有对更深层次的依赖关系进行分析. 基于本体定义元模型, 金龙飞等<sup>[8]</sup> 提出了本体图模型的概念并对波及效应进行了分析和量化界定. 但此本体图模型节点和边的定义较繁琐, 并且没有给出形式化的表示模型. 为判断本体实例的有效性, 文献[9]定义了若干依赖规则, 但提到的依赖规则只局限于属性之间的依赖关系.

收稿日期: 2011-01-05.

**作者简介:** 尚 晋(1986—), 男, 汉族, 硕士研究生, 从事语义网与本体工程的研究, E-mail: jseanj@126.com. 通信作者: 刘 磊(1960—), 男, 汉族, 博士, 教授, 博士生导师, 从事软件形式化、语义网与本体工程的研究, E-mail: liulei@mail.jlu.edu.cn.

**基金项目:** 国家自然科学基金(批准号: 60873044).

基于本体元素间的依赖关系,元素的变化会产生更深层次的影响.为了量化这种影响,同时简化本体图模型及形式化地表示变化,本文首先定义了本体元素间的依赖关系,同时提出一种依赖关系图模型,该图模型基于描述逻辑子语言 SHOIN( $D$ )建立,然后给出本体变化的定义和变化影响频度的定义,最后基于依赖关系图的可达矩阵给出计算变化影响频度的算法.

## 1 本体元素依赖关系与依赖关系图模型

描述逻辑<sup>[10]</sup>(description logic,简称 DL)作为一阶逻辑的子集,是知识表示的一种形式化机制.描述逻辑是以个体、概念(个体的集合)、角色(关联两个个体)为基本元素的概念描述语言.通过使用构造算子,构造复杂的概念和角色.由于构造算子集合的不同,形成了不同表达能力的描述逻辑子语言.

本文只考虑描述逻辑子语言 SHOIN( $D$ ),因为 Web 本体语言 OWL DL 的语义可通过描述逻辑的语义定义,OWL DL 对应于描述逻辑 SHOIN( $D$ )<sup>[11]</sup>.SHOIN( $D$ )描述逻辑提供了丰富的构造算子,包括角色的逆算子、数量值约束、枚举构造算子等.

在本体中,一个本体元素(如概念、属性)的修改,常会影响到本体中的其他元素,这些元素由于受到影响,需要修改才能减少依赖于本体服务受到的影响.

### 1.1 本体元素间的依赖关系

给定一个本体  $O$ ,  $E$  是本体  $O$  的元素集合,  $E = N_C \cup N_R \cup N_I$ ;  $E'$  是本体  $O$  中概念与角色的集合,  $E' = N_C \cup N_R$ . 其中:  $N_C$  是本体的概念集合;  $N_R$  是本体的角色集合;  $N_I$  是本体的个体集合.

**定义 1** 对于  $E$  中的两个元素  $e_1$  和  $e_2$ , 如果针对  $e_1$  的变化影响到  $e_2$ , 则称  $e_2$  依赖于  $e_1$ , 记为  $e_1 < e_2$ .

本文根据 SHOIN( $D$ ) 知识库中 TBox 和 ABox 的公理集合和断言集合以及构造算子给出本体元素间依赖关系的定义.

**定义 2** 对于  $E'$  中的元素  $e_1, e_2, \dots, e_n$ , 如果  $e_1 \subseteq e_2 * e_3 * \dots * e_n$ , 其中  $*$  是构造算子, 则  $e_2 < e_1$ ,  $e_3 < e_1, \dots, e_n < e_1$ , 这种依赖关系称为蕴涵依赖.

例如, 知识库中存在公理  $\text{Researcher} \subseteq \text{Person}$ , 则  $\text{Person} < \text{Researcher}$ .

**定义 3** 对于  $E'$  中的元素  $e_1, e_2, \dots, e_n$ , 如果  $e_1 = e_2 * e_3 * \dots * e_n$ , 其中  $*$  是构造算子, 则  $e_2 < e_1$ ,  $e_3 < e_1, \dots, e_n < e_1$ ,  $e_1 < e_2 * e_3 * \dots * e_n$ ,  $e_2 * e_3 * \dots * e_n < e_1$ , 这种依赖关系称为等式依赖.

**定义 4** 给定一个概念  $C$  和个体  $a$ , 如果  $C(a) \in \text{Abox}$ , 则  $C < a$ , 这种依赖关系称为断言依赖.

例如, 知识库中存在公理  $\text{Researcher}(\text{peter})$ , 则  $\text{Researcher} < \text{peter}$ .

由于角色  $R$  直接依赖于  $R$  的定义域和值域, 而个体  $a$  和  $b$  分别依赖于角色  $R$  的定义域和值域, 因此定义域和值域的变化既会影响角色  $R$  也会影响个体  $a$  和  $b$ , 但角色  $R$  与个体  $a$  和  $b$  间不存在依赖关系.

**定义 5** 给定一个概念  $C$  和一个角色  $R$ , 如果存在实体  $\exists R. C$  或  $\forall R. C$ , 则  $C < R$ , 这种依赖关系称为角色依赖.

SHOIN( $D$ ) 公理、实体和断言与本体元素依赖关系间的映射列于表 1.

表 1 SHOIN( $D$ ) 公理、实体和断言与本体元素依赖关系间的映射

Table 1 Matching of SHOIN( $D$ ) axioms, entities, assertions and dependency relations between ontology elements

SHOIN( $D$ )中的公理、实体和断言	依赖关系
$C_1 \subseteq C_2$	$C_2 < C_1$
$R_1 \subseteq R_2$	$R_2 < R_1$
$C \equiv D$	$C < D, D < C$
$\exists R. C$	$C < R$
$\forall R. C$	$C < R$
$C(a)$	$C < a$

除了上述依赖关系中出现的构造算子, 描述逻辑中还包括其他构造算子, 如  $\sqcap$  等, 但这些构造算子连接的元素之间不存在依赖关系, 因而这里不定义此类构造算子的依赖关系.

### 1.2 依赖关系图模型

将本体元素视为图的节点, 它们之间的依赖关系视为图的边, 本体模型即被抽象为一个图模型.

**定义 6** 一个本体  $O$  的依赖关系图模型是一个有向图  $G = (V, E)$ , 其中: 节点集  $V$  中的每个元素  $V_i$  对应本体  $O$  中的概念、属性或实例; 边集  $E$  中的每个元素  $\langle V_i, V_j \rangle$  是一个有序对, 当且仅当  $V_i < V_j$  时,  $\langle V_i, V_j \rangle \in E$ .

**例 1** SHOIN( $D$ ) 描述逻辑语言表示本体模型, 其中 TBox 为  $\{ \text{Researcher} \subseteq \text{Person}, \text{Student} \subseteq \text{Person}, \text{Article} \subseteq \text{Publication}, \top \subseteq \forall \text{author. Publication}, \top \subseteq \forall \text{author. Person} \}$ , ABox 为  $\{ \text{Article}(\text{anArticle}), \text{Researcher}(\text{peter}), \text{Student}(\text{peter}), \text{author}(\text{anArticle}, \text{peter}) \}$ .

根据上面定义的 4 种依赖关系, 可以给出本体中元素间的依赖关系, 图 1 为依赖关系图模型. 由图 1 可见, 该本体的依赖关系图模型由 8 个节点和 8 条边组成. 节点表示本体元素:  $\{ V_1 = \text{Publication}, V_2 = \text{Person}, V_3 = \text{author}, V_4 = \text{Article}, V_5 = \text{Researcher}, V_6 = \text{Student}, V_7 = \text{anArticle}, V_8 = \text{peter} \}$ . 边表示本体各元素间的依赖关系.

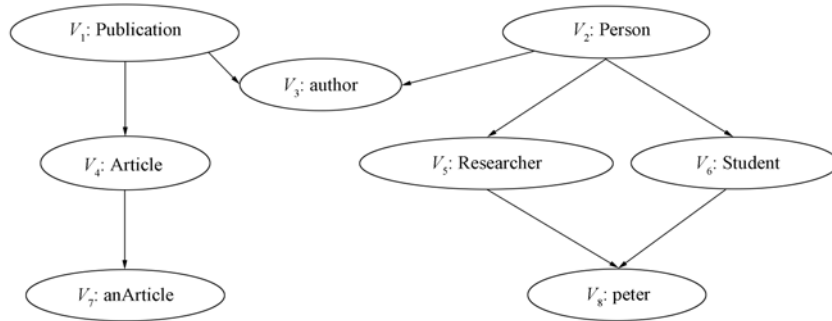


图 1 依赖关系图模型

Fig.1 Dependency graph

该依赖关系图与文献[9]提出的本体图模型相比有 3 点不同:

- 1) 该依赖关系图的节点用本体元素名称表示, 而本体图模型中的节点用包含节点类型和属性集合的二元组形式表示, 这种二元组表示方式显得繁琐;
- 2) 该依赖关系图中元素间的依赖关系是基于描述逻辑建立的, 依赖关系有明确的形式化定义, 而本体图模型只通过举例说明元素间的依赖关系, 不具有普遍适用性;
- 3) 该依赖关系图的节点类型包含了基本的本体元素类型, 而本体图模型中却未考虑个体对本体中其他成分的影响.

由于将本体元素间的依赖关系表示为依赖关系图模型的形式, 因而可在分析变化影响时使用图论的方法进行量化计算.

## 2 变化的影响

在本体演化中, 变化既包括对本体元素的修改、添加或删除, 也包括对本体元素间关系的修改、添加或删除. 由于任何一种变化都可通过添加和删除这两种基本变化的组合完成<sup>[8]</sup>, 因此, 在描述逻辑表示的本体模型中, 变化可视为向描述逻辑知识库中添加或删除实体.

**定义 7** 给定一个描述逻辑表示的本体  $O$ , 一个本体变化被定义为一个函数:  $f(e) = e'$ , 其中:  $e$  为描述逻辑知识库中的实体或公理;  $e'$  为  $O$  的依赖关系图模型变化后的结果.

由本体变化的定义可见, 当本体发生变化时, 与其对应的依赖关系图模型也发生变化. 如向描述逻辑表示的本体知识库中添加一条公理时, 公理中的实体(概念、角色或个体)首先会作为节点添加到依赖关系图模型中; 然后根据本体元素依赖关系的定义, 将表示这些实体间依赖关系的有向边添加到依赖关系图模型中. 这时, 一个本体变化才执行完毕.

在例 1 中, 删除概念 Person 和 Researcher 间继承关系的变化被定义为  $\text{RemoveAxiom}(\text{Researcher} \subseteq$

Person) = { }}, 集合为空表示依赖关系图模型中已经将边  $\langle \text{Person}, \text{Researcher} \rangle$  删除. 添加概念 Person 和 Researcher 间继承关系的变化被定义为  $\text{AddAxiom}(\text{Researcher} \subseteq \text{Person}) = \{ \langle \text{Person}, \text{Researcher} \rangle \}$ .

在本体变化时, 因为本体元素间都有依赖关系的存在, 因此, 当一个元素变化时, 会对其他元素产生影响, 同时也会对依赖于这些本体元素的应用程序产生影响. 文献[7]提出一种可减小变更影响范围的本体演化算法, 该算法中, 一个本体元素的变化只会影响到相邻实体, 影响未继续传播. 文献[9]提出的一种波及效应分析方法中虽然在量化影响范围时考虑了变化影响传播的特性, 但没有考虑本体变化的实时性. 而在实际 Web 环境下, 由于变化影响的传播特性, 使影响的范围可能很大, 且由于变化的实时性及依赖关系图模型中可能包含环, 使本体元素在变化执行过程中可能受到多次影响. 因此, 本文给出变化影响频度的定义, 并通过一个算法计算依赖关系图模型中变化的影响频度.

**定义 8** 设依赖关系图模型  $G = \langle V, E \rangle$ , 给定一个图变化  $g(e)$ , 其中  $e$  是  $G$  中的一个实体, 则有:

1) 如果  $e \in E$ , 则变化的影响频度为

$$\text{IF}(e) = \sum \text{NumPath}(V_n) - i + 1,$$

其中:  $e = \langle V_m, V_n \rangle$ ;  $\text{NumPath}(V_k)$  是计算以节点  $V_k$  为起点的有向路径上节点的个数;  $i$  是以节点  $V_k$  为起点的有向路径的个数;

2) 如果  $e \in V$ , 则此变化的影响频度为

$$\text{IF}(e) = \sum_{\langle V_m, V_j \rangle \in E} \text{IF}(\langle V_m, V_j \rangle) + \sum_{\langle V_i, V_m \rangle \in E} \text{IF}(\langle V_i, V_m \rangle),$$

其中  $e = V_m$ .

在例 1 中, 假设变化需求为删除概念 Person, 则首先删除边  $\langle \text{Person}, \text{author} \rangle$ , 删除这条边会影响角色 author; 再删除边  $\langle \text{Person}, \text{Researcher} \rangle$ , 这时会影响概念 Researcher 和个体 peter; 最后删除边  $\langle \text{Person}, \text{Student} \rangle$ , 会影响概念 Student 和个体 peter. 当这 3 条边都被删除后, 再将概念 Person 删除. 因此, 个体 peter 会被影响两次, 即删除概念 Person 的变化影响频度为 5.

根据变化影响频度的定义, 下面给出计算依赖关系图模型中变化影响频度的算法.

**算法 1** 计算本体变化影响频度算法  $\text{ImpactInOnto}(O, \mathbf{M}, \text{ChReqs})$ .

输入:  $(O, \mathbf{M}, \text{ChReqs})$ , 即待演化的本体  $O$ 、本体依赖关系图模型的可达矩阵  $\mathbf{M}$  和待执行的变化操作  $\text{ChReqs}$ , 其中如果  $V_j < V_i$ , 则  $M_{ij} = 1$ , 否则  $M_{ij} = 0$ .

输出: 待执行变化的影响频度.

实现:

```
function ImpactInOnto( $O, \mathbf{M}, \text{ChReqs}$ ) {
  value = 0; //初始变化影响频度值为 0
  switch ChReqs do
    case addAxiom( $D$ ):
      抽取出公理  $D$  中包含的本体元素集合  $\{V_1, V_2, \dots, V_n\}$ 
      查询本体元素依赖关系, 得到  $m$  个元素影响对的集合 EdgeSet, 此集合包含  $m$  条边
      for  $\forall \langle V_i, V_j \rangle \in \text{EdgeSet}$  do
        if  $V_i$  不是数据属性 then
          for  $1 \leq k \leq n$  do
            if  $M_{kj} = 1$  then //如果在依赖关系图模型中存在边  $\langle V_j, V_k \rangle$ 
              value + +;
            value + +; //由于  $V_j$  也受到影响, 将影响频度加 1
          return value;
    case removeAxiom( $D$ ):
      抽取出公理  $D$  中包含的本体元素集合  $\{V_1, V_2, \dots, V_n\}$ 
      查询本体元素依赖关系, 得到  $m$  个元素影响对的集合 EdgeSet, 此集合包含  $m$  条边
```

```

for  $\forall \langle V_i, V_j \rangle \in \text{EdgeSet}$  do
  for  $1 \leq k \leq n$  do
    if  $M_{kj} = 1$  then //如果在依赖关系图模型中存在边 $\langle V_j, V_k \rangle$ 
      value + + ;
      value + + ; //由于  $V_j$  也受到影响, 将影响频度加 1
  return value;
case removeEntity ( C ):
  查询本体图模型, 得到该实体对应的本体元素  $V_m$ ;
  value = DelEntity (  $V_m$  );
   $M$  中删除本体元素  $V_m$ ;
  return value;
case addEntity ( C ):
  查询本体图模型, 得到该实体对应的本体元素  $V_m$ ;
   $M$  中添加本体元素  $V_m$ ;
  return 0;
}
function DelEntity (  $V_m$  ) {
  for  $1 \leq i \leq n$  do
    if  $M_{im} = 1$  then //如果在依赖关系图模型中存在边 $\langle V_m, V_i \rangle$ 
      value + + ;
      DelEntity (  $V_i$  );
    for  $1 \leq j \leq n$  do
      if  $M_{mj} = 1$  then //如果在依赖关系图模型中存在边 $\langle V_j, V_m \rangle$ 
        value + + ;
      value = value +  $\partial$ ; //删除本体元素本身的影响频度定义为常数 $\partial$ ,  $\partial > 1$ 
  return value;
}

```

算法 1 包含两个函数: ImpactInOnto 函数和 DelEntity 函数. 其中 ImpactInOnto 函数的主要功能是计算变化的影响频度. 首先, 算法 1 针对变化种类进行变化匹配, 如果添加或删除公理, 则抽取公理中包含的本体元素, 查询这些本体元素间的依赖关系, 通过这些依赖关系确定有向边的方向, 再针对每条有向边, 确定变化是添加边还是删除边的操作, 并根据依赖关系图的可达矩阵计算每条边变化所影响的节点数; 如果是添加或删除实体, 则首先要查询本体图模型, 得到该实体对应的本体元素, 然后添加或删除该本体元素. 其中, 删除实体时要调用 DelEntity 函数, 此函数通过依赖关系图的可达矩阵计算删除与实体连接的所有边所影响的节点数. 注意删除实体本身的影响比删除包含实体公理的影响大许多, 因此定义删除实体本身的影响频度为常数 $\partial$ , 其中 $\partial > 1$ . 而添加实体所影响的节点数为 0.

例 1 中, 以删除概念 Researcher 为例. 在本体编辑工具 Protégé 中, 删除概念后会将所有子概念或个体全部删除, 因此将概念 Researcher 删除后, 会将个体 peter 删除, 变化的序列为 { removeAxiom ( Researcher ( peter ) ), removeAxiom ( Researcher  $\subseteq$  Person ), removeEntity ( Researcher ), removeEntity ( peter ) }, 总的变化影响频度为  $2 + 2\partial$ . 而如果采用另外一种变化策略, 即当父概念被删除后所有子概念或个体重新作为祖先概念的子概念或个体时, 总的变化影响频度将降低. 例如, 删除概念 Researcher 后, 个体 peter 重新作为概念 Person 的个体, 总的影响频度为  $3 + \partial$ , 此变化影响频度小于在 Protégé 中执行变化的影响频度. 因此, 可将变化影响频度作为启发式搜索本体一致性状态的一种启发函数.

综上所述, 在本体演化中, 不同本体演化方法对依赖于本体的应用程序所产生的影响较大, 因为本体中元素的语义在演化过程中可能变得更具体或更宽泛, 且本体元素在不同时刻频繁的变化也会影

响依赖于它的应用. 本文通过分析描述逻辑 SHOIN( $D$ )的公理语义给出了本体元素间的依赖关系, 通过依赖关系建立了本体依赖关系图模型; 提出变化影响频度的概念, 并给出了相应的计算方法, 变化影响频度的提出为启发式搜索本体一致性状态提供了一种启发式函数.

### 参 考 文 献

- [ 1 ] Tomas R G. A Translation Approach to Portable Ontology Specifications [J]. Knowledge Acquisitions, 1993, 5(2): 199-220.
- [ 2 ] Stojanovic L, Maedche A, Stojanovic N, et al. Ontology Evolution as Reconfiguration-Design Problem Solving [C]//Proceedings of the 2nd International Conference on Knowledge Capture K-CAP'03. New York: ACM, 2003: 162-171.
- [ 3 ] Haase P, Stojanovic L. Consistent Evolution of OWL Ontologies [C]//Proceedings of 2nd European Semantic Web Conference 2005. Berlin: Springer, 2005: 182-197.
- [ 4 ] Noy N F, Klein M. Ontology Evolution: Not the Same as Schema Evolution [J]. Knowledge and Information Systems, 2004, 6(4): 428-440.
- [ 5 ] Stojanovic L. Methods and Tools for Ontology Evolution [D]: [Ph D Thesis]. Karlsruhe: University of Karlsruhe, 2004.
- [ 6 ] Stojanovic L, Maedche A, Motik B, et al. User-Driven Ontology Evolution Management [C]//EKAW'02 Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management 2002. London: Springer-Verlag, 2002: 133-140.
- [ 7 ] LIU Chen, HAN Yan-bo, CHEN Wang-hu, et al. MINI: An Ontology Evolution Algorithm for Reducing Impact Ranges [J]. Chinese Journal of Computers, 2008, 31(5): 711-720. (刘晨, 韩燕波, 陈旺虎, 等. MINI: 一种可减小变更影响范围的本体演化算法 [J]. 计算机学报, 2008, 31(5): 711-720.)
- [ 8 ] JIN Long-fei, LIU Lei. A Ripple-Effect Analysis Method for Ontology Evolution [J]. Acta Electronica Sinica, 2006, 34(8): 1469-1474. (金龙飞, 刘磊. 一种本体演化波及效应分析方法 [J]. 电子学报, 2006, 34(8): 1469-1474.)
- [ 9 ] LI Qin, Vijayalakshmi A. Evaluating the Validity of Data Instances against Ontology Evolution over the Semantic Web [J]. Information and Software Technology, 2009, 51(1): 83-97.
- [ 10 ] Baader F, Calvanese D, McGuinness D, et al. The Description Logic Handbook: Theory, Implementation and Applications [M]. Cambridge: Cambridge University Press, 2003.
- [ 11 ] Horrocks I, Patel-Schneider P F. Reducing OWL Entailment to Description Logic Satisfiability [J]. Journal of Web Semantics, 2004, 1(4): 345-357.

(责任编辑: 韩 啸)