

一种本体演化波及效应分析方法

金龙飞, 刘 磊

(吉林大学计算机科学与技术学院, 吉林长春 130012)

摘 要: 本文提出了一种本体图模型, 建立了本体邻接矩阵和可达矩阵, 凭借矩阵变换与运算对本体演化中的波及效应进行了深入的分析 and 量化界定; 文中阐述了对演化中的本体变化操作所引起的各种波及效应; 给出了本体元素对本体结构的贡献大小、本体内聚度及本体演化中元素受影响程度的计算方法, 并且分析了依赖未知情况下的本体演化的波及效应问题, 提出了一种支持本体演化的服务模型, 从而为本体演化的管理、控制、利用和评价提供了可靠的依据, 为本体演化的计算机自动处理奠定了基础。

关键词: 本体演化; 波及效应; 可达矩阵

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2006) 08-1469-06

A Ripple-Effect Analysis Method for Ontology Evolution

JIN Long-fei LU Lei

(College of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China)

Abstract An ontology graph model which creates ontology adjacency matrix and ontology reachability matrix is described. Depending on matrix shift and calculation, ripple-effect of ontology evolution can be analyzed and its quantity can be ascertained. Every ripple-effect caused by ontology change operation is described. At the same time, approaches for calculating ontology element contribution, ontology cohesions and effect degree applied to every ontology element during ontology evolution, are provided. Ripple-effect analysis for dependency-unknown ontology evolution is discussed, and a server model for ontology evolution is also provided. All are credible foundation for management, control, usage and evaluation of ontology evolution, and are foundation for ontology evolution automation calculation in computer.

Keywords ontology evolution; ripple-effect; reachability matrix

1 引言

语义网 (Semantic Web)^[1] 是计算机信息处理领域当前研究的热点之一, 其基本思想是对当前的万维网进行扩展, 使得网络中所有信息都是具有语义的, 是计算机能够理解和处理的, 便于人和计算机之间的交互与合作。本体 (Ontology) 是某一领域共享概念模型的明确表示和描述, 是语义网实现的基础和关键。本体和万维网的特点决定了必须允许对本体进行修改订正, 由此产生了本体演化 (Ontology Evolution) 问题。本体的变化对本体所定义的数据、依赖于该本体的其它本体以及使用这些本体和数据的应用程序都会产生较大的影响。

目前本体演化研究的重要性已经得到了充分的认识。但是由于本体演化的复杂性, 致使大多数研究还停留在框

架分析和定性讨论层面。本文提出了一种本体图模型, 建立了图模型的本体邻接矩阵和可达矩阵, 凭借矩阵变换与运算对本体演化中的波及效应进行了深入的分析 and 量化界定; 并对演化中的本体变化操作所引起的各种波及效应给予了阐述; 同时, 给出了本体元素对本体结构的贡献大小、本体内聚度及本体演化中元素受影响程度的计算方法, 并分析了依赖未知情况下的本体演化的波及效应问题, 提出了一种支持本体演化的服务模型。为本体演化的管理、控制、利用和评价提供了可靠的依据, 为本体演化的计算机自动处理奠定了基础。

2 本体图模型及本体邻接矩阵

本体是共享的概念模型的形式化的规范说明^[2]。其定义如下:

定义 1 本体 本体可以用五元组 $O = (C, \leq_c, R, \leq_r, \sigma)$ 来表示, 其中^[3]:

- (1) C 和 R 为两个集合, 分别表示概念集合和关系集合;
- (2) 一个作用于 C 上的偏序关系 \leq_c , 称为概念层次或分类法;
- (3) 一个函数 $\sigma: R \rightarrow C^+$, 称为标识或签名;
- (4) 一个作用于 R 上的偏序关系 \leq_r , 称为关系层次,

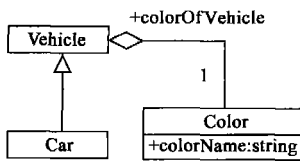
对于 $1 \leq i \leq |\sigma(r_i)|$, 如果 $r_1 \leq_r r_2$, 则意味着 $|\sigma(r_1)| = |\sigma(r_2)|$ 而且 $p_i(\sigma(r_1)) \leq_c p_i(\sigma(r_2))$, p_i 表示作用于 R 上的一个操作.

其中关系层次 (即父子关系) 和概念层次 (父子概念) 密切相关, 例如当 $p_i(\sigma(r))$ 表示 r 定义域 (Domain) 中的概念时, 则父子关系 $r_1 \leq_r r_2$ 对应的定义域中的类 $c_1 \in p_i(\sigma(r_1))$ 和 $c_2 \in p_i(\sigma(r_2))$ 也一定是父子概念, 即满足 $c_1 \leq_c c_2$.

本体定义元模型 (Ontology Definition Metamodel 简称 ODM)^[4] 是国际组织 OMG (Object Management Group) 正在制订的本体工程 (Ontology Engineering) 标准. 其目的是给出本体语言的 MDA (Model Driven Architecture)^[5] 元模型定义, 以便使用 MDA 模型技术和工具来建模和管理本体. ODM 的最直接的表现就是可以用类似 UML (Unified Modeling Language)^[6] 的图形式方式表示本体. 使用 ODM 所创建的本体模型, 图形元素丰富, 简单直观, 易于理解, 便于本体工程师和领域工程师之间的交流. 为了方便计算机的自动处理, 以 ODM 为基础, 我们提出了一种简化的本体图模型, 其定义如下:

定义 2 本体图模型 一个本体 O 的图模型是一种有向的类型图 $G = \langle V, E \rangle$ 其中结点集 V 的元素对应 O 中的概念或关系, 边集 E 的元素对应 O 中的函数、概念层次或关系层次, 并且 $v \in V, v = \langle t, A \rangle$, 其中结点类型 t 对应本体概念或关系的 ODM 类型 (如 OWLClass), A 是结点的属性集合, $a \in A, a = \langle name, value \rangle$, 即属性是名称和值的二元组; $e \in E, e = \langle t, from, to \rangle$, e 的类型对应 ODM 中的表示关系的类型 (如 OWLObjectProperty), $from, to \in V$, 分别表示边的始点和终点, 边的方向表示结点间的语义方向, V 称为本体元素, E 称为本体关系 (不同于本体定义中“关系”的含义).

基于本体和本体图模型定义可以将本体表示成图的形式, 这种图形方式的图模型清晰地体现出本体的结构性, 举例如下:



例 1 一个本体用基于 ODM 的模型表示如图 1 所示:

该本体的图模型则由 10 个结点和 10 条边组成, 结点表示本体元素:

$$v_1 = \langle \text{OWLClass} \{ \text{localName} = \text{Vehicle} \} \rangle$$

- $v_2 = \langle \text{OWLClass} \{ \text{localName} = \text{Car} \} \rangle$
- $v_3 = \langle \text{OWLClass} \{ \text{localName} = \text{Color} \} \rangle$
- $v_4 = \langle \text{RDFSdatatype} \{ \text{localName} = \text{xsd:string} \} \rangle$
- $v_5 = \langle \text{OWLdatatypeProperty} \{ \text{localName} = \text{colorName} \} \rangle$
- $v_6 = \langle \text{RDFSdatatype} \{ \text{localName} = \text{xsd:NonNegativeInteger} \} \rangle$
- $v_7 = \langle \text{RDFXMLliteral} \{ \text{lexicalForm} = "1" \} \rangle$
- $v_8 = \langle \text{OWLrestriction} \Phi \rangle$
- $v_9 = \langle \text{OWLfunctionalProperty} \{ \text{localName} = \text{colorOfVehicle} \} \rangle$
- $v_{10} = \langle \text{OWLrestriction} \Phi \rangle$

边表示本体关系, 边的方向表示终点对始点的依赖, 边上标签的含义一般与依赖关系相反:

- $e_1 = \langle \text{RDFSsubClassOf} \{ v_1, v_2 \} \rangle$
- $e_2 = \langle \text{RDFSdomain} \{ v_3, v_5 \} \rangle$
- $e_3 = \langle \text{RDFSrange} \{ v_4, v_5 \} \rangle$
- $e_4 = \langle \text{RDFtype} \{ v_6, v_7 \} \rangle$
- $e_5 = \langle \text{OWLobjectProperty} \{ v_5, v_8 \} \rangle$
- $e_6 = \langle \text{OWLcardinality} \{ v_7, v_8 \} \rangle$
- $e_7 = \langle \text{RDFSdomain} \{ v_1, v_9 \} \rangle$
- $e_8 = \langle \text{RDFSrange} \{ v_3, v_9 \} \rangle$
- $e_9 = \langle \text{OWLobjectProperty} \{ v_9, v_{10} \} \rangle$
- $e_{10} = \langle \text{OWLcardinality} \{ v_7, v_{10} \} \rangle$

其图模型如图 2 所示 (省略了各结点和边的类型及结点属性):

本体表示为图模型, 则可以使用图论方法处理本体. 图 3 给出了图 2 中本体图模型对应的邻接矩阵. 图中表格单元中的图形符号表示语义关系, 其方向性由行和列分别表示, 列为关系的第一个元素, 行为关系的第二个元素. 每个 \times 表示结点与自身的关系, 不同的本体语言下这种关系可能有不同的语义, 如 RDFS 中每个类都是自己的子类^[7], 本文不关心这种自身的关系. 图中 \blacklozenge 表示行列中两个结点间存在着直接的语义关系, \odot 表示行列中两个结点间不存在或存在间接的语义关系. \blacklozenge 和 \odot 表示不同环境和应用中语义关系形式的抽象. 例如, 当语义表示两个结点间可达路径上结点的个数 (包括两端) 时, 列 v_1 和行 v_{10} 交叉处变为数值 3, 即 $v_1 \rightarrow v_9 \rightarrow v_{10}$, 而列 v_{10} 与行 v_1 交叉处变为数值 0, 即不可达.

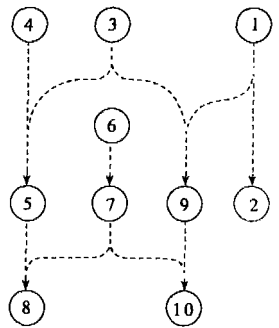


图 2 本体图模型示例

图 3 给出了图 2 中本体图模型对应的邻接矩阵. 图中表格单元中的图形符号表示语义关系, 其方向性由行和列分别表示, 列为关系的第一个元素, 行为关系的第二个元素. 每个 \times 表示结点与自身的关系, 不同的本体语言下这种关系可能有不同的语义, 如 RDFS 中每个类都是自己的子类^[7], 本文不关心这种自身的关系. 图中 \blacklozenge 表示行列中两个结点间存在着直接的语义关系, \odot 表示行列中两个结点间不存在或存在间接的语义关系. \blacklozenge 和 \odot 表示不同环境和应用中语义关系形式的抽象. 例如, 当语义表示两个结点间可达路径上结点的个数 (包括两端) 时, 列 v_1 和行 v_{10} 交叉处变为数值 3, 即 $v_1 \rightarrow v_9 \rightarrow v_{10}$, 而列 v_{10} 与行 v_1 交叉处变为数值 0, 即不可达.

3 本体演化波及效应分析与量化

3.1 本体演化介绍

为了使用本体图模型来分析本体演化波及效应, 我们

| | | | | | | | | | | |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| | v_1 | v_2 | v_3 | v_4 | v_5 | v_6 | v_7 | v_8 | v_9 | v_{10} |
| v_1 | × | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |
| v_2 | ◆ | × | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |
| v_3 | ⊙ | ⊙ | × | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |
| v_4 | ⊙ | ⊙ | ⊙ | × | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |
| v_5 | ⊙ | ⊙ | ◆ | ◆ | × | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |
| v_6 | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | × | ⊙ | ⊙ | ⊙ | ⊙ |
| v_7 | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ◆ | × | ⊙ | ⊙ | ⊙ |
| v_8 | ⊙ | ⊙ | ⊙ | ⊙ | ◆ | ◆ | × | ⊙ | ⊙ | ⊙ |
| v_9 | ◆ | ⊙ | ◆ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | × | ⊙ |
| v_{10} | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ◆ | ⊙ | ◆ | × | ⊙ |

图 3 图 2 的本体邻接矩阵

首先介绍一下本体演化的相关内容。

定义 3 本体演化 本体演化是指本体对出现的变化的适时修改及这些变化到附属产物的一致性传播^[8]。

一种被广泛接受的本体演化过程模型包括如下六个阶段^[9]:

- (1)变化的捕获: 从显示的需求说明或变化发现方法的结果中捕获本体的变化;
- (2)变化的表示: 形式化和显式地表示本体的变化;
- (3)变化的语义: 计算为了保证本体一致性而产生的额外变化;
- (4)变化的传播: 更新依赖于变化本体的数据、其它本体和应用程序, 以保持一致性;
- (5)变化的实现: 以一种事务处理的方式实现需求的和导出的本体变化;
- (6)变化的确认: 评估结果, 如果需要的话, 开始新一轮的本体演化。

从上述定义和说明中可以看出, 本体变化所产生的波及效应的界定和度量是本体演化中一个重要的问题。本文首先从依赖关系已知的本体演化波及效应分析入手, 基于本体图模型给出其分析和度量方法, 然后考虑依赖关系未知情况下本体演化波及效应的推 拉模型。

3.2 依赖已知情况下的本体演化波及效应分析与量化

3.2.1 波及效应的界定

定义 4 本体语义关系矩阵 设本体元素集合 $X = \{E_1, E_2, \dots, E_n\}$, 则本体语义关系矩阵 $M_R = [R_{ij}]$, 其中 R_{ij} 表示元素 E_j 与 E_i 间的语义关系, $i, j = 1, 2, \dots, n$, 并且

$$R_{ij} = \begin{cases} 1, & \text{当 } E_j \text{ 与 } E_i \text{ 之间存在直接语义关系时} \\ 0 & \text{当 } E_j \text{ 与 } E_i \text{ 之间不存在直接语义关系时} \end{cases}$$

从定义可以看出, 可以通过将本体邻接矩阵图中十字交叉处填充上一定语义的数值后得到本体语义关系矩阵。

定义 5 本体可达矩阵 设本体元素集合 $X = \{E_1, E_2, \dots, E_n\}$, 本体语义关系矩阵 M_R 对应的关系为 $R \subseteq X^2$, R 的传递闭包 $R^+ = R \cup R^2 \cup \dots \cup R^n$, R^+ 对应的矩阵 $M_{R^+} =$

$$M_R \vee M_R^2 \vee \dots \vee M_R^n = \bigvee_{k=1}^n M_R^k, M_{R^+} = M_{R^+}^{i-1} \vee M_R, i = 1, 2, \dots,$$

n , 称为本体可达矩阵, 其中 R_{ij}^+ 表示元素 E_j 与 E_i 间的可达关系,

$$R_{ij}^+ = \begin{cases} 1, & \text{当 } E_j \text{ 可达 } E_i \text{ 时} \\ 0 & \text{当 } E_j \text{ 不可达 } E_i \text{ 时} \end{cases}$$

例如, 图 2 对应的本体语义关系矩阵和本体可达矩阵 (不考虑结点对自身的可达性) 分别为:

$$M_R = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$M_{R^+} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

可以看出, 该本体图模型中结点 v_3 的修改将影响到 v_5, v_8, v_9 和 v_{10} , 因此, 通过可达矩阵可以界定某一本体元素的变化所影响或波及到的其它元素的范围。

3.2.2 相关的量化计算

定义 6 本体元素对本体结构的贡献 设本体元素集合 $X = \{E_1, E_2, \dots, E_n\}$, 本体可达矩阵 $M_{R^+} = [R_{ij}^+]$, 则

$Contribute(E_j) = \sum_{i=1}^n R_{ij}^+$ 表示本体元素 E_j 的修改所影响的其它元素的个数, 称为该元素对本体结构的贡献 (或贡献大小)。

将各个元素对本体结构的贡献大小排序, 可以清楚地看出元素在本体中的重要性, 也可以用于指导用户制订本体修改策略。如图 2 对应的本体可达矩阵的各列之和分别为 3 0 4 2 1 3 2 0 1 和 0 所以各元素 (用本体图模型中的结点表示) 对本体结构的贡献大小排列为 $v_3 > v_1 = v_6 > v_4 = v_7 > v_5 = v_9 > v_2 = v_8 = v_{10}$ 。由该排序可以看出, v_3 的变化对本体结构的影响要大于 v_7 的变化, 而 v_2, v_8 和 v_{10} 的修改对本体结构的其它部分没有影响。在需求允许的情况下, 要尽可能修改对本体结构贡献小的元素, 以免产生大的波及影响。

定义 7 本体内聚度 设本体元素集合 $X = \{E_1, E_2, \dots, E_n\}$, 本体可达矩阵 $M_{R^+} = [R_{ij}^+]$, 引入一个集合来记录本体元素 E_i 所依赖的其它元素的集合: $Dep(E_i) = \{E_j$

$|R_{ij} = 1, i \neq j\}$, 则本体内聚度为:

$$Cohesion(O) = 1/n \sum_{i=1}^n \frac{|Dep(E_i)|}{n-1}$$

根据定义可知 $|Dep(E_i)| = \sum_{j=1}^n R_{ij}^+$, 可以根据对本体可达矩阵的各行求和得到. 本体内聚度越大表示本体图模型中各元素联系越密切, 对本体的修改将更困难. 本体内聚度也可以用于本体质量的评估.

定义 8 本体元素受影响程度 假设本体图模型 $G = (V, E)$ 中有向边的距离都为 1, 结点 v_i 到 v_j 的最短距离为 $D(v_i, v_j)$, 则 v_j 对应的本体元素受 v_i 对应本体元素的变化影响程度为:

$$Effect(v_i, v_j) = 1/D(v_i, v_j)$$

不同本体元素的变化会对某一本体元素产生不同程度的影响, 表现为本体图模型中到达同一结点的不同的路径长度. 本体元素对应结点距离变化元素(变化源)对应结点越远, 则该元素所受的影响程度值越小. 例如, 删除图 2 中的结点 v_3 (即删除 Color 类), 则受影响的元素有 v_5, v_9, v_8 和 v_{10} , 各元素受影响程度分别为 1, 1, 0.5 和 0.5. 如果本体元素对应结点与变化元素对应结点间没有路径, 则变化对该元素没有影响, 如删除 v_3 对 v_1 就没有影响. 当多个本体元素发生变化时, 它们对同一本体元素的影响(即合成效应)等于各个变化元素对该本体元素的影响程度之和.

3.2.3 本体演化的基本操作的波及影响分析

将本体表示为图模型, 则本体演化的基本操作对应为对图的操作, 具体分析如下:

(1) 删除一个元素: 如果可达矩阵中第 i 列全为 0, 则可以直接删除元素 E_i , 而不会对其它元素产生影响, 只会影响本体的结构. 例如在图 2 中删除 v_2, v_8 和 v_{10} . 如果可达矩阵中第 i 列至少存在一个 1, 则删除元素 E_i 将会对本体其它元素产生影响, 其影响大小和各元素受影响程度可以按上面的定义计算. 例如在图 2 中删除 v_3 要比删除 v_9 影响大, 这两种删除对 v_{10} 的影响程度分别为 0.5 和 1, 而对 v_1 则没有影响.

(2) 增加一个元素: 在本体中增加一个元素, 首先要提供与之有直接语义关系的元素, 进而形成新的可达矩阵, 最后在整个本体范围内判定被影响的本体元素, 计算方法同上.

(3) 修改一个元素: 没有改变语义连接的修改, 对本体结构没有影响, 而改变语义连接的修改可以看作是删除元素和增加元素的组合操作, 可以按照上面两种情况分析.

(4) 元素的合并: 在本体元素的合并过程中, 可达矩阵可以作为指导合并的依据, 也可以用于求取合并后新的可达矩阵. 如果矩阵两列对应的内容相同, 则表示这两个元素在本体结构中语义连接相同, 可以直接合并; 如果内容相差很大, 则不适于直接合并. 对于可以合并的元素, 合并后新的可达矩阵可以用分块矩阵的计算方法从原有可达

矩阵中计算出来.

(5) 元素的拆分: 元素的拆分是元素合并的逆过程, 需要用新的可达矩阵来计算, 在此不再赘述.

(6) 本体图中边的修改: 本体图模型中边的修改直接改变语义连接, 因此这种修改必须提供新的语义连接, 进而形成新的可达矩阵, 最后在整个本体范围内判定被影响的本体元素, 计算方法同上.

3.3 依赖未知情况下的本体演化波及效应分析与量化

目前, 本体的一个重要应用领域是语义网. 本体发布后, 使用该本体的数据、本体和应用程序都是不可知的, 本体的修改对整个语义网的影响将是很难计算的. 本文借用 Maedche 等人^[8]提出的基于推 拉机制的本体同步思想, 实现了基于推 拉机制的依赖未知情况下的本体演化波及效应分析.

3.3.1 基于推 拉机制的本体同步技术介绍

在文献 [8] 中, 作者根据本体数目和本体所在物理结点的情况, 将本体演化分为单一本体演化、依赖本体演化和复制本体演化三种情况. 对于仅有一个本体的单一本体演化, 作者通过定义本体演化策略, 提供给用户选择保持本体一致性的方法; 对于依赖本体演化(多个位于同一物理结点上的本体), 作者给出了一种基于推机制的本体变化传播方式, 将本体的变化立即传播到位于同一结点上的依赖于该本体的其它本体; 对于复制本体演化(多个位于不同物理结点上的本体), 由于存在远程本体的本地拷贝, 作者给出了一种基于拉机制的本体变化传播方式, 仅在依赖本体需要的时候才将本体变化传播过去.

3.3.2 基于推 拉机制的本体演化波及效应分析

上述基于推 拉机制的本体同步技术可以很好地移植到本体演化波及效应分析中.

定义 9 依赖本体 包含其它本体的本体称为依赖本体.

定义 10 本体变化描述 设变化源的本体元素集合 $X = \{E_1, E_2, \dots, E_n\}$, $M_R = [R_{ij}^+]$ 是本体可达矩阵, 则本体变化描述 $C_D = \langle X_C, M_C \rangle$, 其中 $X_C \subseteq X$ 表示发起变化的本体元素集合, 称 $M_C = [R_{ab}^+]$ 为变化矩阵, 其中 $\{R_{ab}^+\} \subseteq \{R_{ij}^+\}$, 并且 $E_a, E_b \in X_C$ 或受变化所影响.

本体变化描述包含了发起变化的本体元素集合和受影响的本体元素对应的变化矩阵. 当本体变化描述传递到依赖本体时, 可以根据 M_C 和依赖本体的语义关系矩阵 M_R

$$\text{共同构造新的本体语义关系矩阵 } M_{new} = \begin{bmatrix} M_C & 0 \\ M & M_R \end{bmatrix}, \text{ 其中}$$

M 中的元素表示依赖本体与 X_C 中元素间的语义连接. 利用该矩阵可以计算新的可达矩阵, 进而求出 X_C 中元素的变化对依赖本体的影响.

3.3.3 本体演化服务模型

由于文献 [8] 中并未考虑依赖性未知情况下的本体演化问题, 为此我们提出了一种基于注册机制的本体演化服务模型. 该模型的结构如图 4 所示: 图中每个本体及其所

在物理节点的信息被注册保存在一个结点数据库 (Node Database) 中, 当某个本体 (如 Ontology A) 发生变化时, 虽然本体对依赖关系未知, 但它将变化描述发送到本体演化服务器上, 服务器查询结点数据库, 得到和变化本体同一结点的本体 (如 Ontology B) 的信息, 以推的方式发送同步通知, 由该本体根据自身的依赖关系决定是否需要波及效应分析和本体更新等操作, 服务器还将本体的变化描述连同变化时间等信息保存到变化数据库 (Changes Database) 中, 位于其它结点 (如 Node N2) 的本体 (如 Ontology C) 在需要时, 使用拉的方式从本体服务器取得本体变化信息, 以更新需要的本体副本 (如 Copy of Ontology A). 为了提高可伸缩性, 本体演化服务器也可以是服务器的群集 (如图 4 所示).

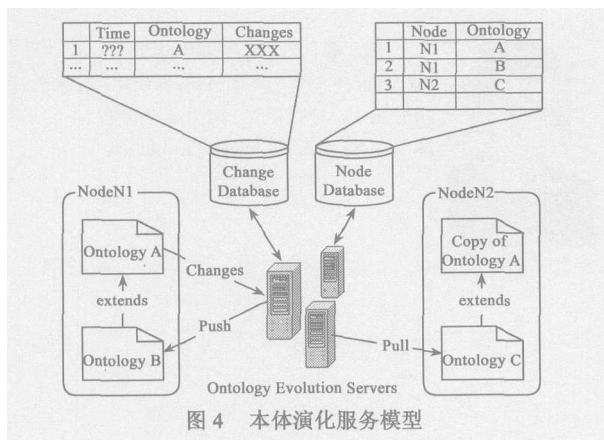


图 4 本体演化服务模型

4 相关工作

本体演化问题是当前本体工程和语义网研究的热点问题, 研究工作正在广泛和深入地进行中. 本文使用的本体演化过程模型, 最早是由文献 [9] 的作者提出的, 并得到了广泛的认可. 文献 [8] 的作者给出了本体演化的三种情况: 单一本体演化、依赖本体演化和复制本体演化, 并对不同情况给出了演化策略、基于推和基于拉机制的本体同步方法. 本文以这些内容为基础提出了本体演化服务框架. W3C (World Wide Web Consortium) 在制订本体语言标准 OWL (Web Ontology Language)^[10] 时也考虑了本体演化问题, 如 OWL 语言中的 owl:backwardCompatibleWith 和 owl:incompatibleWith 等. 文献 [11] 给出了支持本体扩展和版本化的本体语言的模型理论, 并将其应用到 RDF 和 OWL 上. 在文献 [12] 中, 作者从结构、逻辑和用户定义三个角度给出了 OWL 语言的一致性描述. 文献 [13] 中, 作者将 MDA 模型转换技术应用到本体演化中, 给出了解决本体演化问题的另一种思路. KAON^[14] 和 Proégé^[15] 等本体建模和管理工具也已经增加了本体演化和版本化的支持.

与之前研究工作的侧重点不同, 本文不是解决本体变化及一致性管理问题, 而是从定量分析角度给出了本体演化波及效应的分析和计算方法.

5 结束语

本文从本体定义元模型出发, 提出了一种本体图模型, 给出了图模型的邻接矩阵及可达矩阵, 以此为基础, 给出了依赖已知情况下本体演化的波及效应分析和量化界定方法, 并借鉴已有的基于推/拉机制的本体同步方法, 提出了一种支持本体演化波及效应分析的本体演化服务模式, 为本体演化的计算机自动处理奠定了基础.

进一步的研究工作包括针对具体本体语言的本体演化波及效应分析方法的细化, 矩阵算法的优化, 本体演化服务系统的设计和实现, 以及基于波及效应的演化过程评估等.

参考文献:

- [1] T Bemers-Lee, J Hendler, O Lassila. The semantic web [J]. Scientific American, 2001, 284(5): 34-43
- [2] W N Borst. Construction of Engineering Ontologies for Knowledge Sharing and Reuse [D]. Enschede University of Twente, 1997.
- [3] E Bozsak, M Ehrig, S Handschuh, et al. KAON - towards a large scale semantic web [A]. Proc of the Third International Conference on E-Commerce and Web Technologies (EC-Web 02) [C]. London, UK: Springer-Verlag, 2002, 304-313.
- [4] Object Management Group, Inc. Ontology Definition Metamodel Revised Submission to OMG / RFP ad/2003-03-40 [DB/OL]. <http://www.omg.org/docs/ad/05-01-01.pdf>, 2005-01-01/2005-09-01.
- [5] Object Management Group, Inc. MDA Guide Version 1.0.1 [DB/OL]. <http://www.omg.org/cgi-bin/apps/doc?omg/03-06-01.pdf>, 2003-06-01/2005-09-01.
- [6] Object Management Group, Inc. UML 2.0 Infrastructure Specification [DB/OL]. <http://www.omg.org/cgi-bin/apps/doc?ptc/03-09-15.pdf>, 2003-09-15/2005-09-01.
- [7] P Hayes. RDF Semantics - W3C Recommendation 10 February 2004 [DB/OL]. <http://www.w3.org/TR/2004/REC-rdfmt-20040210/>, 2004-02-10/2005-09-01.
- [8] A Maedche, B Motik, L Stojanovic. Managing multiple and distributed ontologies in the semantic web [J]. The International Journal on Very Large Data Bases, 2003, 12(4): 286-302.
- [9] L Stojanovic, A Maedche, B Motik, N Stojanovic. User-driven ontology evolution management [A]. Proc of the 13th European Conference on Knowledge Engineering and Knowledge Management (EKAW 2002) [C]. Siguenza, Spain: Springer-Verlag, 2002, 285-300.
- [10] M Dean, G Schreiber. OWL Web Ontology Language Reference - W3C Recommendation 10 February 2004 [DB/OL]. <http://www.w3.org/TR/owl-ref/>, 2004-02-10/2005-09-01.

OL]. <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>, 2004-02-10/2005-09-01

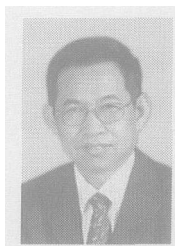
- [11] J Heflin, Z Pan. A model theoretic semantics for ontology versioning[A]. Proc of the Third International Semantic Web Conference (ISWC 2004) [C]. Hiroshima, Japan: Springer-Verlag, 2004. 62-76
- [12] P Haase, L Stojanovic. Consistent evolution of OWL ontologies[A]. Proc of the Second European Semantic Web Conference (ESWC 2005) [C]. Heraklion, Crete, Greece: Springer-Verlag, 2005. 182-197
- [13] J Longfei, L Lei, Y Dong. A model transformation based conceptual framework for ontology evolution[A]. Proc of the 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2005) [C]. Melbourne, Australia: Springer-Verlag, 2005. 325-331.
- [14] T Gabel, Y Sure, J Voelker. KAON-ontology management infrastructure[R]. Karlsruhe, Germany: Institute AIFB, University of Karlsruhe, 2004.
- [15] N F Noy, S Kunnatur, M Klein, M A Musen. Tracking

changes during ontology evolution[A]. Proc of the Third International Semantic Web Conference (ISWC 2004) [C]. Hiroshima, Japan: Springer-Verlag, 2004. 259-273

作者简介:



金龙飞 男, 1977年2月出生于吉林省白山市, 吉林大学计算机科学与技术学院博士研究生, 主要研究方向为本体工程和语义网。
E-mail: jinlongfei@vip.163.com



刘 磊 男, 1960年8月出生于吉林省白城市, 吉林大学计算机科学与技术学院教授, 博士生导师, 主要研究方向为程序分析、本体工程和语义网。
E-mail: liulei@jlu.edu.cn