

# 分布式环境下领域本体的自动提取<sup>\*</sup>)

高英 郭荷清 邹智敏

(华南理工大学计算机科学与工程学院 广州 510641)

**摘要** 分布式环境下领域本体模型必须具备明确而且清晰的语义,以便于被共享和利用,本文基于该目的,提出了一种从分布式环境下自动提取本体的方法,并对如何提取本体中概念、关系、关系冲突消解和自动提取算法做了详细的阐述。

**关键词** 领域本体,概念,关系,关系冲突消解,自动提取算法

## The Extraction Algorithm of Domain Ontology Automatically in Distributed Environment

GAO Ying GUO He-Qing ZOU Zhi-Min

(Department of Computer Science and Engineering, South China University of Technology, Guangzhou 510641)

**Abstract** Domain ontology model must hold explicit and clear semantic for share and use. Based on this purpose, the paper provides a method of extracting domain ontology automatically in distributed environment, and describes how to extract concepts, relations, solving relation conflict and automatic extraction algorithm.

**Keywords** Domain ontology, Concept, Relation, Relation conflict solution, Automatic extraction algorithm

分布式环境中的领域知识建模与普通环境下的领域知识建模存在以下差异<sup>[1]</sup>:①分布式环境中的领域知识模型需要被大量的用户共享,即领域本体所表达的领域知识不是个人私有的,而是被一个群体所接受的知识。②在分布式环境下的领域知识模型所涉及的知识应该更加全面,不仅需要考虑领域中概念之间纵向继承的知识体系,而且需要表示概念之间横向联系的知识。③在分布式环境下,领域知识模型的语义必须明确而清晰,明确而清晰的语义有利于分布环境中计算机对知识的共享和利用。

本体建立的过程主要以手工为主,而不是工程行为。具体领域本体的手工建立和描述是复杂和费时的过程<sup>[2-6]</sup>。本文提出了一种从分布式环境下自动提取本体的方法。第2节提出了提取领域本体中概念、关系、及解决关系冲突的规则;第3节对领域本体的自动提取算法做了详细的阐述;第4节用一个实例进行了验证。

### 1 领域本体

**定义 1-1** 领域本体定义为五元组。其中

Domain-Ontology ::= {OntoConcepts, Relations, Rules, Axioms, Individuals};

OntoConcepts ::= {concept};

Relations ::= {R(C<sub>1</sub>, C<sub>2</sub>), C<sub>1</sub>, C<sub>2</sub> ∈ OntoConcepts};

Rules ::= {rule};

Axioms ::= {axiom};

Individuals ::= {Individual | σ(individual) ∈ OntoConcepts}。

OntoConcepts 为领域内以术语形式出现的概念集合,概念具有描述事物性质(Properties)的属性(Attributes)。世界是由具有性质的事物组成的,事物都有性质,具体事物的性质

被称为实质性质,概念上事物的性质被称为形式化性质、属性或断言。性质分为内在性质和相互性质,内在性质依赖于某个具体事物,相互性质依赖于两个或多个具体事物。性质的存在不依赖于人的感知,属性则是根据人的感知指派给事物的特征。Relations 为领域内概念之间的二元关系集合。通常,领域中本体之间存在的关系有泛化关系(Generalize),整体-部分关系(PartOf),实例关系(InstanceOf),关联关系(Association),属性关系(AttAssociation)。Rules 为规则的集合。任何关于概念或关系的可能取值或它们之间可能关系的组合关系的约束被称为规则。规则定义为 RULE: X<sub>1</sub> × ... × X<sub>m</sub> × T → S, 其中, X<sub>1</sub>, ..., X<sub>m</sub> ∈ OntoConcepts ∪ Relations, T 表示时间实例的集合, S 表示关于 X<sub>1</sub>, ..., X<sub>m</sub> 的陈述。RULES 表示在某些观察点, S 对 X<sub>1</sub>, ..., X<sub>m</sub> 可能取值或组合关系的陈述成立。AXIOMS 为公理的集合,用于定义规则或关系之间的关系和规则。公理定义为 AXIOMS: X<sub>1</sub> × ... × X<sub>m</sub> → S, 其中, X<sub>1</sub>, ..., X<sub>m</sub> ∈ OntoConcepts ∪ Relations ∪ Rules, Axioms 表示在任何条件下, S 对 X<sub>1</sub>, ..., X<sub>m</sub> 可能取值或组合关系的陈述成立。INDIVIDUALS 为领域本体内概念实例的集合。其中, σ: INDIVIDUALS → ONCEPTS 是概念实例到所属概念的映射。概念实例具有概念定义的属性。

### 2 领域本体的提取

#### 2.1 领域本体的提取规则

本文的前提是分布式环境下语义相似的元素已经形成聚类,本节的目的是从这些聚类中提取有代表性的概念,形成领域本体中的概念,即 2<sup>E</sup> → C。设 Cl<sub>k</sub> 为某个聚类,是具有一定级别语义相似程度的元素集合, C<sub>k</sub> 为从 Cl<sub>k</sub> 中提取出来的本体概念。本体概念具有如下性质:

**性质 2-1** 不相交性:不同聚类被提取为本体中不同的

<sup>\*</sup>)国家 973 高科技研究发展计划基金资助项目(G20000263)、广州市科技公关项目(B2-109-550)。高英 在职博士,讲师,主要研究领域为信息系统的集成与安全;郭荷清 教授,博士生导师,主要研究领域为信息系统的集成与安全。

概念。即

$$(C_k \neq C_q \Rightarrow Cl_k \cap Cl_q = \phi)$$

**性质 2-2** 属性的合并性:本体概念  $C_k$  的属性为其对应的聚类中元素属性的并集。即  $P(C_k) = \cup P(e_i) \forall e_i \in C_k$ 。

合并时,如果不同元素的属性具有一定的语义相似性,那么这些属性将合并为概念的一个属性。

**性质 2-3** 属性的相交性:本体概念  $C_k$  的特征属性( $C_k$ )为其对应的聚类中元素属性的交集。即

$$FP(C_k) = \cap P(e_i) \forall e_i \in C_k$$

概念是从具有一定语义相似程度的候选聚类中提取出来的。候选聚类被定义为:

**定义 2-4** 候选聚类,记作  $\overline{Cl}_i$ ,为聚类分析中元素间的相似性大于等于阈值  $\lambda$  的聚类,即  $\overline{Cl}_i = Cl_i^S, S \geq \lambda$ 。

阈值  $\lambda$  决定聚类中的元素在语义上的相似程度。如果  $\lambda$  非常高,那么本体概念将从高同类的元素中提出,提取出来的本体概念将进一步通过做泛化分析进行重组;如果  $\lambda$  较低,那么本体概念将从更多的异类的元素中提出。因此,候选聚类的选择取决于阈值  $\lambda$ , 阈值  $\lambda$  不能自动决定,需要领域专家的参与。本文从大到小逐步减小  $\lambda$ , 按层次形成聚类树,根据形成的聚类树,由领域专家判断合适的  $\lambda$ 。

候选聚类是提取本体中概念的起点。概念和概念属性的提取取决于一组提取规则:

**CR1:** 设候选聚类中有  $n$  个元素  $e_1, \dots, e_n, n(e_1), \dots, n(e_n)$  为元素  $e_1, \dots, e_n$  对应的元素名,  $(mc, MC)_{e_1}, \dots, (mc, MC)_{e_n}$  概念的提取规则为:

① 从候选聚类包含的所有元素中推导出领域本体的一个概念,该概念能代表聚类中的所有元素。

② 概念必须具有一个名称,且具有唯一性。

③ 概念名为聚类中某个元素名或者是字典中这些元素名的同义词或上位词。

④ 概念的基数  $(mc, MC)_C$  与  $(mc, MC)_{e_1}, \dots, (mc, MC)_{e_n}$  中具有最少限制的基数保持一致,  $mc = \min\{mc_{e_1}, \dots, mc_{e_n}\}, MC = \max\{MC_{e_1}, \dots, MC_{e_n}\}$ 。

**CR2:** 设候选聚类中有个属性  $p_1, \dots, p_m, d_{p_1}, \dots, d_{p_m}$  为属性  $p_1, \dots, p_m$  对应的域,  $n_{p_1}, \dots, n_{p_m}$  为  $p_1, \dots, p_m$  对应属性名,  $(mc, MC)_{p_1}, \dots, (mc, MC)_{p_m}$  为属性的基数,概念属性  $p = \langle n_p, d_p, (mc, MC)_p \rangle$  的提取规则为:

① 如果  $p_1, \dots, p_m$  具有同义词关系或者上下位关系,则属性  $p$  的名称  $n_p$  为  $n_{p_1}, \dots, n_{p_m}$  中某个属性名或者是字典中这些属性名的同义词或上位词。

②  $p$  的域  $d_p$  与  $p_1, \dots, p_m$  中有最少限制的域一致;

③  $p$  的基数  $(mc, MC)_p$  与  $(mc, MC)_{p_1}, \dots, (mc, MC)_{p_m}$  中具有最少限制的基数保持一致,  $mc = \min\{mc_{p_1}, \dots, mc_{p_m}\}, MC = \max\{MC_{p_1}, \dots, MC_{p_m}\}$ ;

④ 如果  $n_{p_1}, \dots, n_{p_m}$  具有同义词关系或者上下位关系,但是域不相互兼容,则由领域专家决定是否提取  $p$ 。

### 2.2 领域本体中概念关系的提取规则

本体中概念关系的提取是通过分析概念的属性和聚类中元素与其它元素的关系进行的。设概念  $C_i, C_k$ , 对应的聚类分别为  $Cl_i, Cl_k, e_{ip} \in Cl_i, e_{kq} \in Cl_k, p=1, \dots, n, q=1, \dots, m$  为分别为聚类中含有的元素,  $P(C_i), P(C_k)$  为概念的属性集,  $KP(C_i), KP(C_k)$  为概念的特征属性集,关系的提取规则如下。

**RR1(Generalize):** 泛化关系的提取规则。

① 如果  $a: e_{ip} \text{ kindof } e_{kq}$ , 或  $b: e_{ip}$  和  $e_{kq}$  对应的  $SDS-G_{ER}$  上, 连接  $e_{ip}$  和  $e_{kq}$  的弧的类型  $\langle e_{ip}, e_{kq} \rangle \in A_5 | A_6$ , 则  $C_i \text{ kindof } C_k$ 。

② 在  $Cl_i, Cl_k$  中, 存在多个元素对具有 Generalize 关系, 则将这些关系合一化, 即  $C_i \text{ kindof } C_k$ 。

③ 如果  $P(C_i) \cap P(C_k) \neq \phi$ , 则生成一个新的概念  $C_k, P(C_k) = P(C_i) \cap P(C_k), C_i \text{ kindof } C_k, C_k \text{ kindof } C_k$ 。

**RR2(Aggregation):** 整体-部分关系的提取规则。这种关系显示一个概念是另一个概念的组成部分, 通常在两个具有不相交性的概念间发生这种关系。

① 在  $SDS-G_{XML}$  上, 连接  $e_{ip}$  和  $e_{kq}$  的弧的类型  $\langle e_{ip}, e_{kq} \rangle \in A_1$ , 即  $e_{kq}$  是  $e_{ip}$  的子元素, 则  $C_k \text{ partof } C_i$ 。

② 在  $e_{ip} \in \overline{Cl}_i$  和  $e_{kq} \in \overline{Cl}_k$ , 且  $e_{kq} \text{ partof } e_{ip}$ , 则  $C_k \text{ partof } C_i$ 。

③ 在  $\overline{Cl}_i, \overline{Cl}_k$  中, 存在多个元素对具有 Aggregation 关系, 则将这些关系合一化, 即  $C_k \text{ partof } C_i$ 。

**RR3(Association):** 关联关系的提取规则。这种关系显示两个概念参与了某种关系, 通常是通过外键或 IDREFR (IDREFS) 提取的。

① 在  $SDS-G_{ER}$  上, 连接  $e_{ip}$  和  $e_{kq}$  的弧的类型  $\langle e_{ip}, e_{kq} \rangle \in A_3 | A_4$ , 则  $C_i \text{ Association } C_k$ 。

② 在  $SDS-G_{XML}$  上, 连接  $e_{ip}$  和  $e_{kq}$  的弧的类型  $\langle e_{ip}, e_{kq} \rangle \in A_3$ , 则  $C_i \text{ Association } C_k$ 。

**RR4(AttAssociation):** 属性关系的提取规则。这种关系显示概念和概念属性之间的关系。

**RR5(InstanceOf):** 实例关系的提取规则。这种关系显示一个对象是某个概念的实例。提取规则如下:

设  $\overline{Cl}_i = \{e_{i1}, \dots, e_{in}\}, EXT(e_{ij}) j=1, \dots, n$  为  $e_{ij}$  的外延, 则  $\bigcup_{j=1}^n e_{ij} \text{ Instance Of } C_i$ 。

### 2.3 概念间关系冲突的解决规则

领域本体中, 任意两个概念间应该具有确定类型的关系。如果在关系的提取过程中, 两个概念间出现多种类型的关系, 则需要确定哪种类型的关系最终成为领域本体中概念间的关系。冲突产生的原因可能是选择了较小的阈值, 聚类中包含了较多的异构元素, 采取的冲突解决的规则是:

**CSR1:** 选择较大的阈值, 使聚类中包含较多的同类元素。

**CSR2:** 如果两个概念间存在多种类型的关系, 选择概念间最弱的关系作为最终的关系类型, 如关联关系。然后由领域专家验证。

**CSR3:**  $\exists C_1, C_2, C_3, C_1 \text{ KindOf } C_2, C_1 \text{ KindOf } C_3, C_2 \text{ KindOf } C_3$ , 如果  $\forall e_{i1} \in \overline{Cl}_1, \exists e_{2j} \in \overline{Cl}_2, e_{i1} \text{ KindOf } e_{2j}$ , 则删除  $C_1, C_3$  之间的关系, 保留关系  $C_1 \text{ KindOf } C_2, C_2 \text{ KindOf } C_3$ 。否则由领域专家解决冲突。

### 3 领域本体的提取算法

基于上述规则, 本节提出了从聚类中提取领域本体的算法。

$OntoConcepts = \{C_1, \dots, C_n\}, C_i = (cName, AttributeList, FAttributeList, LinkSet)$  描述了一个完整的领域本体。  $cName$  是概念名, 在领域本体中具有唯一性。  $AttributeList, FAttributeList$  分别为概念的属性和特征属性集, 两者的区别在于  $AttributeList$  中的属性为聚类中元素属性的并集,  $FAt-$

tributeList 中的属性为聚类中元素属性的交集。LinkSet 是概念与领域本体中其它概念的关系集。

AttributeList, FAttributeList =  $\{(attName, modifier, dType) | modifier \in \{*, +, ?\} \wedge dType \in DOM\}$

属性集、特征属性集是概念的属性集,通过三元组描述概念的每个属性,attName 表示属性名;modifier 表示属性是否为空值、单值或多值。dType 表示属性具有的数据类型。

LinkSet =  $\{(cName, lType, sCard, tCard, lRole) | lType \in \{Generalize, Aggregation, Association\} \wedge sCard, tCard \in Cardinalities \wedge lRole \in attName \cup kindOf \cup partOf \cup AssName\}$

cName 是参与关系的目标概念;lType 描述领域本体中任意两个概念间的关系类型;sCard, tCard 分别为源概念和目标概念的基数;lRole 是关系角色,当 lType  $\in \{Generalize, Aggregation\}$ ,关系的含义很清楚,因此 lRole = kindOf 或 lRole = partOf;如果目标概念为源概念的属性,则 lRole = attName;如果 lType  $\in \{Association\}$ ,则 lRole = assName,由于外键、IDREF、IDREFS 导致关联关系的产生,而且外键、IDREF、IDREFS 等属性的名字也表达了关联关系的含义,因此在自动提取概念间关系时,assName 来源于外键、IDREF、IDREFS 等属性的名字。

设  $SoM = \langle \langle o, o, OWN, \tau \rangle | \tau \in \{C', A', CA', AA'\} \rangle$  为映射集。如果  $\tau = 'C'$ ,表示提取过程中,聚类中元素到概念的映射,o 表示元素,OWN 表示概念,表示元素所在的模式。如果  $\tau = 'A'$ ,表示提取过程中,元素属性到概念属性的映射,o<sub>i</sub> 为元素属性,o<sub>i</sub> 为概念属性,OWN 为属性对应的元素, $\tau = 'CA'$  和  $\tau = 'AA'$  的含义将在后面介绍。

设  $C_i, PreRelationSet = \{(cName, lType, sCard, tCard, lRole)\}$  为首次使用概念间关系提取规则得到的概念  $C_i$  的初始关系集。五元组中每一元的含义与 LinkSet 中每一元的含义相同。PreRelationSets =  $\{C_i, PreRelationSet\}$  是所有概念的初始关系集。

领域本体中概念生成算法如下:

```
GenOntoConcept Algorithm;
GenOntoConcept( ) {
Input: Cl = {C1, ..., Cn}, Rules = {CR}; // Cl: 聚类集, Rules: 一组概念提取规则
Output: OntoConcepts, SoM; // 算法的输出为领域本体的概念集, 映射集
for Cl_i in Cl do // 每个聚类形成一个概念
    Generate New Concept C_i in OntoConcepts;
    C_i.cName = n(e1) ΔCR1 ... ΔCRn n(e_m), e1, ..., e_m in Cl_i; // 运用概念提取规则提取概念名
    for e_j in Cl do
        SoM = SoM ∪ {⟨e_j, C_i, S_i, 'C'⟩ | e_j in S_i}; // 记录聚类中元素到概念的映射
    Compute equivalent class of attribute P^E = {p_i}; // 计算聚类中所有属性的等价类
    for each [p_i] in P^E do {
        attName_i = n(p1) ΔCR1 ... ΔCR2 n(p_m), p1, ..., p_m in [p_i]; // 运用属性提取规则提取属性名
        (mc, MC)_p = (mc, MC)_p1 ΔCR1 ... ΔCR2 (mc, MC)_p_m; // 运用属性提取规则提取属性基数
        if (mc=0 and MC=1) then modifier_i = "?";
        if (mc=0 and MC>1) then modifier_i = "*";
        if (mc≥1 and MC>1) then modifier_i = "+";
        dType_i = d_p1 ΔCR1 ... ΔCR2 d_p_m; // 运用属性提取规则提取属性域
        C_i.AttributeList = C_i.AttributeList ∪ {attName_i, modifier_i, dType_i}; // 将提取的属性加入概念属性集
        if (|Cl_i| = |[p_i]|) and (∀ p1, p2 ∈ [p_i], p^-(p1) ≠ p^-(p2)) then // 判断是否为特征属性, P^-(p1) 为属性所属的元素
            C_i.FAttributeList = C_i.FAttributeList ∪ {attName_i, modifier_i, dType_i}; // 将提取的特征属性加入概念特征属性集
        C_i.LinkSet = C_i.LinkSet ∪ {attName_i, AttAssociation, l, modifier_i, attName_i}; // 建立概念与概念属性的关联关系
        for p_i in [p_i] do // 记录元素属性到概念属性的映射
            SoM = SoM ∪ {⟨p_i, attName_i, e_i, 'A'⟩ | p_i ∈ P(e_i)}; }
}
```

```
OntoConcepts = OntoConcepts ∪ // 将当前提取的概念加入概念集
{C_i.cName, C_i.AttributeList, C_i.FAttributeList, C_i.LinkSet};
}}
```

领域本体中概念关系的生成算法:

```
GenOntoConRelationship Algorithm;
GenOntoConRelationship( ) {
Input: OntoConcepts = {C1, ..., Cn} | Cl = {C1, ..., Cn}, Rules = {RR, CSR};
PreRelationSets = NULL; // OntoConcepts: 概念集, Cl: 聚类集, Rules: 一组概念间关系的提取规则, 关系冲突解决规则, PreRelationSets: 概念的初始关系集
Output: {C1, LinkSet, ..., Cn, LinkSet}; // 算法的输出为概念间的关系集
for each C_i in OntoConcepts do {
    compute C_i.PreRelationSet by RR;
    PreRelationSets = PreRelationSets ∪ {C_i, PreRelationSet}
} // 运用概念间关系提取规则提取概念的初始关系集
for each (Cbsi, Cj) do {
    if (only exist one kind of relationship in C_i and C_j) then // 如果概念 C_i, C_j 间仅存在一种类型的关系
        fetch tuple from C_i.PreRelationSet or C_j.PreRelationSet; // 将关系从 C_i, C_j 的初始关系集中取出
        switch(lType) // 根据关系的类型, 设置 C_i, C_j 的最终关系集
        case "Generalize": C_i.LinkSet = C_i.LinkSet ∪ (C_j.cName, Generalize, *, ?, KindOf-); break;
        case "Aggregation": C_i.LinkSet = C_i.LinkSet ∪ (C_j.cName, Aggregation, l, *, PartOf-); break;
        case "Association": C_i.LinkSet = C_i.LinkSet ∪ (C_j.cName, Association, C_i.modifier, C_j.modifier, AssName-); break;
        C_j.LinkSet = C_j.LinkSet ∪ (C_i.cName, Association, C_j.modifier, C_i.modifier, AssName-); break;
        delete tuple from C_i.PreRelationSet and C_j.PreRelationSet; // 将 C_i, C_j 间的关系从各自的关系初始集中删除
    if (exit more than one kind of relationship in C_i and C_j) then // 如果关系初始集中显示概念间 C_i, C_j 存在多种类型的关系
        compute (C_i, C_j, lType, sCard, tCard, lRole) by CSR; // 根据关系冲突解决规则求出概念间的最终关系
        C_i.LinkSet = C_i.LinkSet ∪ (C_i, C_j, lType, sCard, tCard, lRole);
        C_j.LinkSet = C_j.LinkSet ∪ (C_i, C_j, lType, sCard, tCard, lRole);
        delete all related tuples from C_i.PreRelationSet and C_j.PreRelationSet; // 将 C_i, C_j 间的所有关系从各自的关系初始集中删除
}
```

为了进一步减少领域本体中的概念数,使其更具有抽象性和通用性,需要对上述算法得到的领域本体做进一步的提炼。在试验中,观察到某些概念,即不存在向其它概念发出的关联关系,也不存在其它概念向它发出的关联关系,仅作为子概念与某个概念存在关系或作为“部分”概念与它的“整体”概念存在关系,且基数为“?”,则可以将其从领域本体中删除。

设概念  $C_i$  将被删除, $C_i$  的父概念或整体概念为  $C_j$ 。

1) 如果  $C_i.AttributeList = NULL$ ,则直接将其加入到父概念或整体概念的属性集中,并将转换过程加入到  $SoM$ ,令  $\tau = 'CA'$ ,表示概念到属性的映射,

$o_i = C_i, o_i = C_j.cName, OWN = C_i$ ;

2) 如果  $C_i.AttributeList \neq NULL$ ,则将  $C_i$  的属性加入到  $C_j$  的属性集中,并将转换过程加入到  $SoM$ ,令  $\tau = 'AA'$ ,表示是概念属性到概念属性的映射,

$o_i = C_i.cName + '.' + C_i.AttributeList.attName_i, o_i = o_i, OWN = C_i$ 。

领域本体的修正算法:

```
Revise() ontoConcept Algorithm;
Revise() ontoConcept( ) {
Input: OntoConcepts = {C1, ..., Cn}; // OntoConcepts: 概念集
Output: OntoConcepts = {C1, ..., Cn}; // 算法的输出为修正后的概念集
for each C_i in OntoConcepts do {
    if (|C_i.LinkSet| = 1) // 判断 C_i 是否与其它概念之间仅存在一种关系
        if (C_i.LinkSet.lType = 'Generalize' and C_i.LinkSet.sCard = '?' ) or
            if (C_i.LinkSet.lType = 'Aggregation' and C_i.LinkSet.sCard = ?)
}
```

```

'?' then // 如果 Ci 与其它概念仅存在 Generalize 或 Aggregation 且 Ci 的概念基数为 1
{ Ci = Ci. LinkSet, cName; // 得到 Ci 的父概念或整体概念
if (Ci. AttributeList = NULL) then // 如果是原子概念, 将概念 Ci 转换为 Ci 的属性, 并在 SoM 中记录转换过程
attName = Ci. cName; modifier = "?";
Ci. AttributeList = Ci. AttributeList ∪ {attName, modifier};
SoM = SoM ∪ {<Ci, Ci. cName, Ci, 'CA'>};
if (Ci. AttributeList ≠ NULL) then // 如果是非原子概念, 将概念 Ci 的属性转换为 Ci 的属性, 并在 SoM 中记录转换过程
for each Cj. AttributeList[j] ∈ Ci. AttributeList do {
Ci. AttributeList = Ci. AttributeList ∪ {Cj. AttributeList[j]. attName,
Cj. AttributeList[j]. modifier, Ci. AttributeList[j]. dType}
SoM = SoM ∪ {<Ci, AttributeList[j]. attName, Ci. AttributeList[j]. attName, Ci, 'AA'>};
OntoConcepts = OntoConcepts - {Ci}; // 将概念 Ci 从概念集中删除
}
}

```

#### 4 本体提取实例

例 4-1 模式 A、B 如图 1、2 所示。

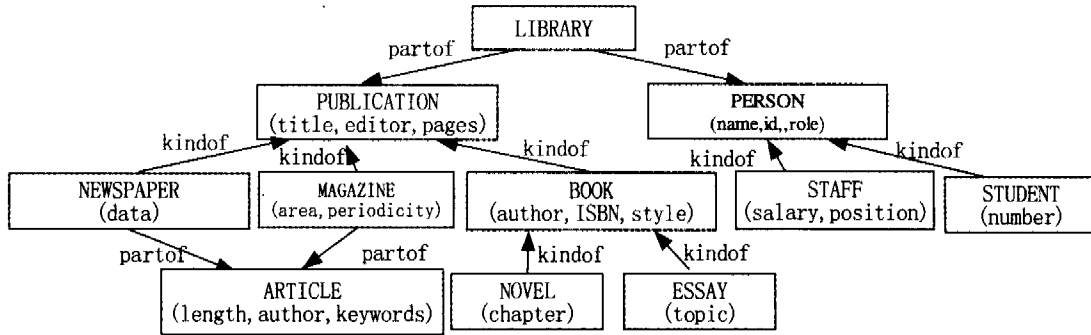


图 1 模式 A

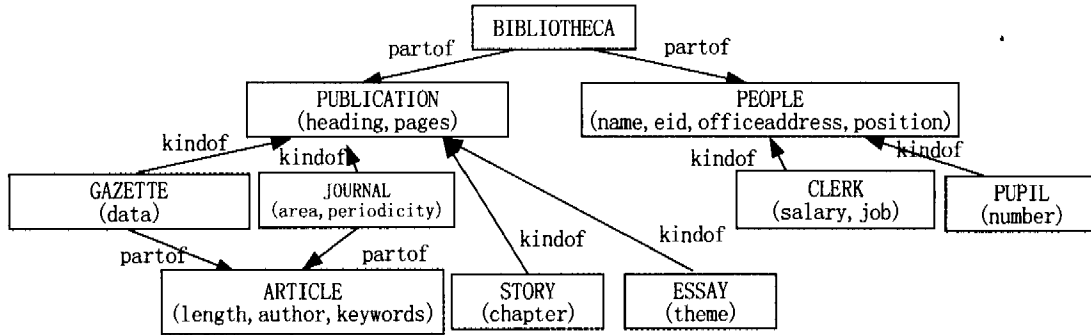


图 2 模式 B

1) 按照前面的相似性分析和聚类分析, 设阈值  $\lambda = 0.93$ , 计算得到下面的 11 个聚类:

- $\bar{C}_1 = \{LABRARY_A, BIBLIOTHECA_B\}$ ,
- $\bar{C}_2 = \{PUBLICATION_A, PUBLICATION_B\}$
- $\bar{C}_3 = \{PERSON_A, PEOPLE_B\}$ ,
- $\bar{C}_4 = \{NEWSPARER_A, GAZETTE_B\}$
- $\bar{C}_5 = \{MAGAZINE_A, JOURNAL_B\}$ ,
- $\bar{C}_6 = \{BOK_A\}$ ,
- $\bar{C}_7 = \{STAFF_A, CLERK_B\}$
- $\bar{C}_8 = \{STUDENT_A, PUPIL_B\}$ ,
- $\bar{C}_9 = \{ARTICLE_A, ARTICLE_B\}$
- $\bar{C}_{10} = \{NOVEL_A, STORY_B\}$ ,
- $\bar{C}_{11} = \{ESSAY_A, ESSAY_B\}$

2) 运用规则 CR1, CR2, 从中提取概念  $C_2$ :

$n(C_2) = n(e_{21}) \Delta_{CR1} n(e_{22}) = PUBLICATION_A \Delta_{CR1} PUBLICATION_B$ ;

属性的等价类:

PUBLICATION

$n(p_1) = title_A \Delta_{CR2} heading_B = title, n(p_2) = editor,$

$n(p_3) = pages_A \Delta_{CR2} pages_B = pages$

$C_2 = \{PUBLICATION, \{title, editor, pages\}, \{\}$

按同样的方法提取其它 10 个概念:

- $C_1 = \{LIBRARY, \{\}, \{\}$ ,
- $C_3 = \{PEOPLE, \{name, id, address, role\}, \{\}$
- $C_4 = \{NEWSPAPER, \{date\}, \{\}$ ,
- $C_5 = \{MAGAZINE, \{area, periodicity\}, \{\}$
- $C_6 = \{BOOK, \{author, ISBN, STYLE\}, \{\}$ ,
- $C_7 = \{EMPLOYEE, \{salary, job\}, \{\}$
- $C_8 = \{STUDENT, \{number\}, \{\}$ ,
- $C_9 = \{ARTICLE, \{length, author, keywords\}, \{\}$
- $C_{10} = \{NOVEL, \{chapter\}, \{\}$ ,
- $C_{11} = \{ESSAY, \{topic\}, \{\}$

$SoM = \{ \langle LABRARY, LABARY, A, C \rangle, \langle BIBLIOTHECA, LABRARY, B, C \rangle,$

$\langle PUBLICATION, PUBLICATION, A, C \rangle, \langle PUBLICATION, PUBLICATION, B, C \rangle,$

$\langle PERSON, PEOPLE, A, C \rangle, \langle PEOPLE, PEOPLE, B, C \rangle, \langle GAZETTE, NEWSPAPER, B, C \rangle$

$\langle NEWSPAPER, NEWSPAPER, A, C \rangle, \langle MAGAZINE, MAGAZINE, A, C \rangle,$

$\langle JOURNAL, MAGAZINE, B, C \rangle, \langle BOOK, BOOK, A, C \rangle, \langle STAFF, EMPLOYEE, A, C \rangle$

$\langle CLERK, EMPLOYEE, B, C \rangle, \langle STUDENT,$

STUDENT, A, C), <PUPIL, STUDENT, B, C>  
 <ARTICLE, ARTICLE, A, C>, <ARTICLE, ARTICLE, B, C>, <NOVEL, NOVEL, A, C>  
 <STORY, NOVEL, B, C>, <ESSAY, ESSAY, A, C>, <ESSAY, ESSAY, B, C> } U  
 <title, title, PUBLICATION<sub>A</sub>, A>, <heading, title, PUBLICATION<sub>B</sub>, A>, <editor, editor, PUBLICATION<sub>A</sub>, A>, <pages, pages, PUBLICATION<sub>A</sub>, A>, <pages, pages, PUBLICATION<sub>B</sub>, A>, <name, name, PERSON<sub>A</sub>, A>, <name, name, PEOPLE<sub>B</sub>, A>, <id, id, PERSON<sub>A</sub>, A>, <eid, id, PEOPLE<sub>B</sub>, A>, <officeaddress, address, PEOPLE<sub>B</sub>, A>, <role, role, PERSON<sub>A</sub>, A>, <position, role, PEOPLE<sub>B</sub>, A>, <data, data, NEWSPAPER<sub>A</sub>, A>, <data, data, GAZETTE<sub>B</sub>, A>, <area, area, MAGAZINE<sub>A</sub>, A>, <area, area, JOURNAL<sub>B</sub>, A>, <periodicity, periodicity, MAGAZINE<sub>A</sub>, A>, <periodicity, periodicity, JOURNAL<sub>B</sub>, A>, <salary, salary, STAFF<sub>A</sub>, A>, <salary, salary, CLERK<sub>B</sub>, A>, <position, job, STAFF<sub>A</sub>, A>, <job, job, CLERK<sub>B</sub>, A>, <number, number, STUDENT<sub>A</sub>, A>, <number, number, PUPIL<sub>B</sub>, A>, <author, author, BOOK<sub>A</sub>, A>, <ISBN, ISBN, BOOK<sub>A</sub>, A>, <style, style, BOOK<sub>A</sub>, A>, <length, length, ARTICLE<sub>A</sub>, A>, <length, length, ARTICLE<sub>B</sub>, A>, <author, author, ARTICLE<sub>B</sub>, A>, <author, author, ARTICLE<sub>A</sub>, A>, <keywords, keywords, ARTICLE<sub>A</sub>, A>, <keywords, keywords, ARTICLE<sub>B</sub>, A>, <theme, topic, ESSAY<sub>B</sub>, A>.

3) 运用规则 RR1, RR2, RR4, 提取概念间的初始关系集:

C<sub>1</sub>. PreRelationSet = { ( PUBLICATION < AGGREGATION, -, -, PartOf ), ( PEOPLE, AGGREGATION, -, -, PartOf ) };

C<sub>2</sub>. PreRelationSet = { ( LIBRARY, AGGREGATION, -, -, KindOf ), ( NEWSPAPER, GENERALIZE, -, -, KindOf ), ( MAGAZINE, GENERALIZE, -, -, KindOf ), ( BOOK, GENERALIZE, -, -, KindOf ), ( NOVEL, GENERALIZE, -, -, KindOf ), ( ESSAY, GENERALIZE, -, -, KindOf ) };

C<sub>3</sub>. PreRelationSet = { ( LIBRARY, AGGREGATION, -, -, PartOf ), ( EMPLOYEE, GENERALIZE, -, -, KindOf ), ( STUDENT, GENERALIZE, -, -, KindOf ) };

C<sub>4</sub>. PreRelationSet = { ( PUBLICATION, GENERALIZE, -, -, KindOf ), ( ARTICLE, AGGREGATION, -, -, PartOf ) };

C<sub>5</sub>. PreRelationSet = { ( PUBLICATION, GENERALIZE, -, -, KindOf ), ( ARTICLE, AGGREGATION, -, -, PartOf ) };

C<sub>6</sub>. PreRelationSet = { ( PUBLICATION, GENERALIZE, -, -, KindOf ), ( NOVEL, GENERALIZE, -, -, indOf ), ( ESSAY, GENERALIZE, -, -, KindOf ) };

C<sub>7</sub>. PreRelationSet = { ( PEOPLE, GENERALIZE, -, -, KindOf ) };

C<sub>8</sub>. PreRelationSet = { ( PEOPLE, AGGREGATE, -, -, KindOf ) };

C<sub>9</sub>. PreRelationSet = { ( NEWSPAPER, AGGREGATION, -, -, PartOf ), ( MAGAZINE, AGGREGATION, -, -, PartOf ) };

C<sub>10</sub>. PreRelationSet = { ( BOOK, GENERALIZE, -, -, KindOf ), ( PUBLICATION, GENERALIZE, -, -, KindOf ) };

C<sub>11</sub>. PreRelationSet = { ( BOOK, GENERALIZE, -, -, KindOf ), ( PUBLICATION, GENERALIZE, -, -, KindOf ) }.

4) 由概念的初始关系集建立概念间的关系, 如图 3 所示。

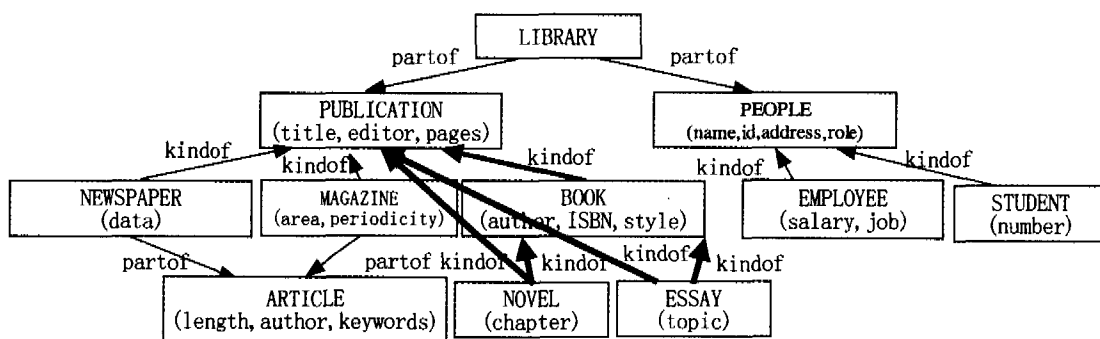


图 3 被提取的领域本体

由图中黑色粗线显示概念间出现的关系冲突, NOVEL, EASSAY PartOf PUBLICATION, BOOK, BOOK PartOf PUBLICATION。由于  $STORY_B \leq NOVEL_A$ ,  $ESSAY_B \leq ESSAY_A$ , 因此根据关系冲突解决规则, 得到解决冲突后的领域本体, 如图 4 所示。

小结 本文的贡献在于: 提出了提取领域本体中概念、关系、及解决关系冲突的规则; 提出了领域本体的自动提取算法。

### 参考文献

- 1 徐振宁. 基于语义 WEB 的信息化表达于语义化过程的研究: [博士学位论文]. 2002
- 2 Maedche A, Staab S. Discovering Conceptual Relations from Text, In: Proceedings of the 14th European Conference on Artificial Intelligence, August 2000. 21~25

(下转第 186 页)

机学报,2003,26(3):310~322

4 樊兴华,仲昕,张勤,等. 因果图推理的一种新方法[J]. 计算机科学,2001,28(11):48~52

5 樊兴华,张勤,黄席樾. 可能性传播图模型的专家知识获取方法[J]. 计算机科学,2001,28(1):53~56

6 张勤,樊兴华,黄席樾,等. 因果图用于复杂系统故障诊断研究[J]. 计算机工程与应用,2002,38(4):43~47

7 Fan Xing-Hua, Sun Mao-Song. A reasoning algorithm of applying causality diagram to fault diagnosis of complex hybrid systems [A]. In: Proceedings of the World Congress on Intelligent Control and Automation [C]. Hangzhou, China, 2004, 2(5): 1741~1745

8 Shi Qingxi, Wang Hongchun, Zhang Qin. Intelligent fault diagnosis technique based on causality diagram [A]. Proceedings of

the World Congress on Intelligent Control and Automation [C]. Hangzhou, China, 2004, 2(5): 1751~1755

9 王洪春,张勤. 基于因果图的一种近似推理算法[J]. 重庆大学学报(自然科学版),2004,27(8):96~99

10 沈文武,汪成亮,程克非,等. 因果图转换为信度网的算法[J]. 重庆大学学报(自然科学版),2004,27(10):33~36

11 Mitauo Y. The median for a L-R fuzzy number [J]. Microelectron Relia, 1995, 35 (2) : 269~271

12 杨伦标,高英仪. 模糊数学原理及应用[M]. 广州:华南理工大学出版社,1993

13 Zhang Qin, An Xuegao, Gu Jin, et al. Application of FBOLES-a prototype expert system for fault diagnosis in nuclear power plants [J]. Reliability Engineering and System Safety, 1991, 34 (2) : 225~235

(上接第 158 页)

以上两个试验表明:在多 Agent 强化学习模型基础上使用学习算法,五个 Agent 都能够学习得到关于长期得益的知识,相互之间的预测能够逐渐准确,二者的联合动作更为连贯,且趋向最优联合动作,从而成功概率逐渐收敛为较高的稳定值,算法有效性得到了验证。另外,这种学习系统还可以推广应用到其它方面,如著名的 Taxi 问题、任务调度和机器人足球中。

**结论与展望** 多 Agent 学习技术使得系统能够适应不确定的环境,提高系统的问题求解能力。多 Agent 强化学习不必具备明确的环境模型,因此该方法在学习者对环境了解甚少的问题域中非常适用。本文的研究目标是针对多 Agent 协作团队这种协作系统中多 Agent 协作求解过程的特点,研究适合的学习方法,以提高协作求解的效率和系统的整体性能。提出了一个新的多 Agent 协作强化学习模型,这个学习模型能加快学习速率,并且降低状态空间和动作空间。基于这个模型,根据 Agent 动作的长期得益的估计,提出了一个新的动作选择策略,并实现了多 Agent 协作学习的学习算法。文中以一个猎人-猎物追逐问题为例,实现并应用了这一学习算法,试验结果表明多个 Agent 通过采用该学习算法最终找到最优联合动作策略。尽管该例讨论的是五个 Agent 的情况,这一结论可以推广用于多个 Agent 的情况。

本文的研究还可以根据具体应用领域进一步细化,同时

进一步的研究将从两个方面展开:一是在相互依赖的竞争性目标下学习协作动作和多 Agent 协作团队在开放环境中进行协作求解时的学习方法;二是在 Agent 不完全知道其他 Agent 行为策略集的情况下,Agent 如何学习以获得最优策略。

参 考 文 献

1 Sutton R, Barto AG. Reinforcement Learning: An Introduction. MIT Press, 1998

2 Tan Ming. Multi-agent reinforcement learning: independent vs. cooperative Agents. In: Proceedings of the 10th International Conference on Machine Learning (ICML-93), 1993. 330~337

3 Watkins C J C H, Dayan P. Q-learning. Machine learning. 1992, 8:272~292

4 蔡庆生,张波. 一种基于 Agent 团队的强化学习模型与应用研究. 计算机研究与发展,2000,37(9)

5 Irwig K, Wobcke W. Multi-Agent Reinforcement Learning with Vicarious Rewards. Electronic Transactions on Artificial Intelligence, 1999, 3(B): 23~45

6 Mataric M J. Interaction and intelligent behavior. [Ph D Thesis]. Department of Electrical Engineering and Computer Science, MIT, USA, 1994

7 Bowling M. Convergence problems of general-sum multi-agent reinforcement learning [A]. In: Langley P, ed. Proceedings of the Seventeenth International Conference on Machine Learning [C]. San Francisco: Morgan Kaufmann Publishers, 2000. 89~94

8 Benda M, Jagannathan V, Dodhiawalla R. On optimal cooperation of knowledge sources. [Technical Report]. BCS-G2010-28. Boeing AI Center, Boeing Computer Services, Bellevue, WA, August 1985

(上接第 168 页)

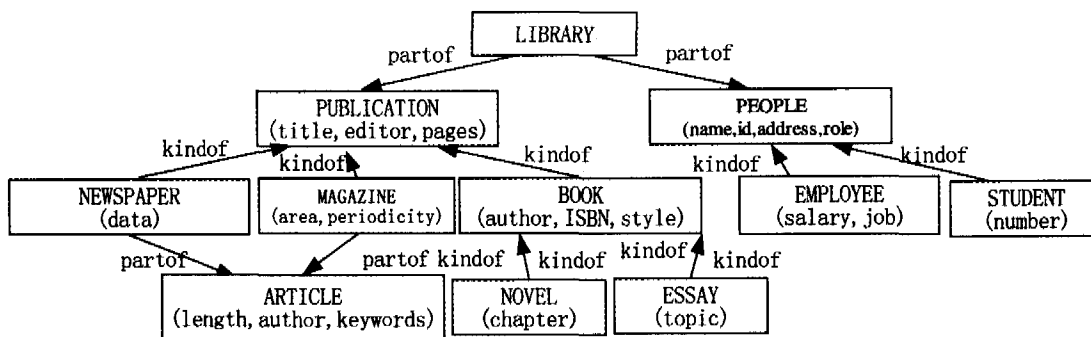


图 4 关系冲突解决后的领域本体

3 Kietz J, Maedche A, Volz R. A Method for Semi-automatic Ontology Acquisition from a Corporate Intranet. Workshop "Ontologies and text", co-located with EKAU, 2000

4 Suryanto H, Compton P. Discovery of Ontologies from Knowledge Bases. In: Proceedings of the First International Conference on Knowledge Capture, 2001. 171~178

5 Deitel A, Faron C, Dieng R. Learning ontologies from RDF annotations. Proceedings of the IJCAI Workshop in Ontology Learning, Seattle, 2001

6 Papatheodorou C, Vassiliou A, Simon B. Discovery of Ontologies for Learning Resources Using Word-based Clustering. ED-MEDIA, 2002