

基于本体的信息系统语义异构性分析^①

王洪伟, 霍佳震, 刘仲英

(同济大学经济与管理学院, 上海 200092)

摘要:每一种信息系统都有适合自身环境与工作效率的领域模型,因而系统间存在各种异构性.如何克服系统间的异构性,尤其是语义异构,是实现信息共享的关键问题.本体是对领域概念化的一个显式的规格说明,可以用来描述信息系统的语义.利用描述逻辑建立了本体的模型,并从本体的角度分析信息系统异构性形成的机理,将信息系统之间的语义异构性归结为概念化冲突异构和术语指派冲突异构两大类,随后提出通过本体映射消除冲突的解决办法,最后建立基于本体映射的异构系统语义集成框架,并对该框架下的数据请求、数据答复、以及数据搜索策略进行分析.

关键词:本体模型;异构性;信息系统;描述逻辑

中图分类号: N94; TP18 **文献标识码:** A **文章编号:** 1007-9807(2008)05-0053-11

0 引言

调查显示,超过85%的信息系统并没有按照能与其他系统合作而设计,数据模型也是为了最大程度地满足自身目标而设计的.正是由于缺少科学的规划,多数系统呈现出信息孤立的状态.随着虚拟企业等组织形式的出现,企业间的信息交换变得更加迫切.可是由于系统设计没有参照统一的标准,开发者对领域的认识也不一致,造成了信息类型、描述方式和存储方式的多样性,这些差异造成了系统间的异构性^[1,2].

信息系统的异构性主要表现为不同的硬件、操作系统、数据模式、数据语义和数据质量^[3].目前,异构性问题已在不同程度上得到解决.中间件(如CORBA、DCOM)、Web Service技术已经较好地解决了不同平台、软件系统间的互操作问题.XML及其相关技术(如DTD、RDF)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.但不同系统之间如果没有事先约定的DTD,还是无法支持在语义层上的集成.此外,还存在一些其它的信息交换标准(如EDI),却由于过于面向特定应用而无法得到普遍的应用^[4].

可以说,对语义层上异构的系统集成目前仍未得到较好解决.

信息系统内部的数据通常都遵循某种默认的假设,一旦脱离系统的实际背景,系统中的数据将变得难以理解.通常,系统运行在特定的环境,因而没有必要将默认的假设显式地表达出来.可一旦被置于开放式的环境中,这些假设必须要加以显式地描述.本体(ontology)是指特定领域概念化的一个显式的规格说明(explicit specification of conceptualization)^[5].实际上,任何一个系统的背后都隐含地依赖于某个本体.领域本体描述了信息系统所用术语之间的关系,并将其中隐含的语义显式地表达出来.当多个系统信息交换时,各自的本体必须要显式化,通过建立本体间的映射,有助于实现异构系统的信息共享^[6,7].

1 本体模型

1.1 本体的基本概念

知识表示源于对领域的概念化(conceptuali-

① 收稿日期:2006-09-19;修订日期:2008-07-06.

基金项目:国家自然科学基金资助项目(70501024);教育部人文社会科学资助项目(05JC870013);上海市重点学科建设项目资助(B310).

作者简介:王洪伟(1973—),男,大连人,博士,副教授. Email:hwwang@mail.tongji.edu.cn

zation). 概念化是人们出于某种目的对希望表示的世界的一个抽象、简化的视图,主要包括:研究领域的对象、概念和其它假定存在的实体,以及维持它们的关系. 每一个信息系统都隐含着或明确地建立在某种概念化的基础上.

本体是一个源于哲学的概念,原意是指关于存在及其本质和规律的学说,近年来被知识工程领域引入,特指对概念化的一个显式的规格说明. 总体而言,本体研究还处于起步阶段,主要集中在:1)本体创建的工程化方法,类似于软件工程对软件开发的作用. 2)本体的形式化表示、转换与重用^[8-10]. 3)本体的应用:如智能搜索^[11,12]、信息系统集成^[13]、知识管理系统^[14,15]、语义 Web 的设计^[16]等. 目前已有部分领域本体被开发,如 Cyc^[17]、WordNet^[18]、TOVE^[19]、Enterprise Project^[20],这些本体模型在目的、规模以及形式化程度上存在很大的差异. 本体使用的形式化描述语言有 KIF^[21]、Ontolingua (www.ksl.stanford.edu/software/ontolingua/),LOOM^[22]、Flogic^[23]. 目前,对本体的形式化描述及高效的本体推理与检验算法仍是研究的热点^[24].

描述逻辑是一种用来描述概念和概念层次关系的知识表示语言,可以看成谓词逻辑的子语言^[25,26]. 尽管没有一阶逻辑的表达能力强,但其提供的推理服务是可判定的,更适用于本体的检验. 描述逻辑的语法还可以转换成 RDF 形式,从而支持 Web 环境下本体部署. 为此,本文将采用描述逻辑建立本体模型,并利用该模型研究信息系统的异构性.

1.2 本体模型的基本结构

定义 1 本体模型是一个 7 元组,记为 $O = \langle T, X, T_D, X_D, A_A, T_C, T_R \rangle$. 其中, T 是术语集; T_D 是术语定义集; X 为实例集; X_D 为实例声明集; A_A 为属性分配集; T_C 为术语注释集; T_R 为术语约束集.

定义 2 给定 $O = \langle T, X, T_D, X_D, A_A, T_C, T_R \rangle$, 语义解释为一个 2 元组,记为 $I = \langle \Delta', (\cdot)'\rangle$. 其中, $\Delta' \neq \emptyset$ 为 O 的论域, $(\cdot)'$ 是解释函数,它将 T 中的每个原子类 C 都映射为 Δ' 的一个子集 $C' \subseteq \Delta'$, 将 T 中的每个原子属性 P 都映射为

一个二元关系 $P' \subseteq \Delta' \times \Delta'$, 将 X 中的每一个体 a 映射为 Δ' 中的一个元素 $a' \in \Delta'$. 下面给出了一个本体实例.

$O = \langle \{ \text{Person, Woman, Man, Mother, Father, Parent, Child, Son, Daughter, Wife, Husband, Son-in-law, Daughter-in-law, Sex, Grandmother, MotherWithoutSon, hasSex, hasChild, hasMother, hasFather, hasParent, hasSon, hasDaughter, hasHusband, hasWife, hasOffspring, age, name} \}, \{ \text{female, male} \}, \{ \text{Woman} \equiv \text{Person} \wedge \exists \text{hasSex.} \{ \text{female} \}, \text{Man} \equiv \text{Person} \wedge \exists \text{hasSex.} \{ \text{male} \}, \text{Mother} \equiv \text{Woman} \wedge \exists \text{hasChild. Person, Father} \equiv \text{Man} \wedge \exists \text{hasChild. Person, Parent} \equiv \text{Father} \vee \text{Mother, Grandmother} \equiv \text{Mother} \wedge \exists \text{hasChild. Mother, Son} \equiv \text{Man} \wedge \text{Child, Daughter} \equiv \text{Woman} \wedge \text{Child, Wife} \equiv \text{Woman} \wedge \exists \text{hasHusband. Man, Husband} \equiv \text{Man} \wedge \exists \text{hasWife. Woman, Son-in-law} \equiv \text{Man} \wedge \exists (\text{hasWife} \times \text{hasParent}). \text{Person, Daughter-in-law} \equiv \text{Woman} \wedge \exists \text{hasHusband} \times \text{hasParent}). \text{Person, MotherWithoutSon} \equiv \text{Mother} \wedge \forall \text{hasSon.} \perp, \text{hasChild} \equiv \text{hasSon} \vee \text{hasDaughter, hasParent} \equiv \text{hasFather} \vee \text{hasMother, hasHusband} \equiv \text{hasWife}^-, \text{hasParent} \equiv \text{hasChild}^-, \text{hasOffspring} \equiv \text{hasChild}^+ \}, \{ \text{Sex(female), Sex(male)} \}, \{ \text{Person} \sqsubseteq \exists \text{age. xsd : Integer} \wedge = 1, \text{age} \wedge \exists \text{name. xsd : String} \}, \emptyset, \{ \exists \text{hasSex.} \{ \text{male} \} \sqcap \exists \text{hasSex.} \{ \text{female} \}, \text{Child} \sqsubseteq \text{Person} \} \rangle$

术语集 T 由原子术语构成. 原子术语分为原子类术语与原子属性术语. 如上述代码中, Person、Sex 为原子类术语, hasChild、hasWife 为原子属性术语. 原子类术语有 2 个特殊的类型:根类“T”与空类“ \perp ”, 相应的语义解释为 $T' = \Delta', \perp' = \emptyset$.

术语公式是由原子术语与术语构造符相互作用形成的表达式,可分为类术语公式和属性术语公式. 如代码中 $\text{Man} \wedge \exists \text{hasChild. Person}$ 是类术语公式, $\text{hasWife} \times \text{hasParent}$ 是属性术语公式. 表 1 构建出 11 种术语构造符,其中, C 与 D 类术语, P 与 R 是属性术语.

表1 本体模型的术语构造符及其解释

Table 1 Term constructors and interpretations in ontology model

构造符的名称及语法	构造符的解释
类术语否: $\neg C$	$(\neg C)^I = \Delta^I - C^I$
属性术语否: $\neg P$	$(\neg P)^I = \Delta^I \times \Delta^I \setminus P^I$
术语合取: $C \wedge D, P \wedge R$	$(C \wedge D)^I = C^I \cap D^I, (P \wedge R)^I = P^I \cap R^I$
术语析取: $C \vee D, P \vee R$	$(C \vee D)^I = C^I \cup D^I, (P \vee R)^I = P^I \cup R^I$
属性的积: $P \times R$	$(P \times R)^I = \{(a, c) \in \Delta^I \times \Delta^I \mid \exists b. (a, b) \in P^I \wedge (b, c) \in R^I\}$
属性的逆: P^-	$(P^-)^I = \{(b, a) \in \Delta^I \times \Delta^I \mid (a, b) \in P^I\}$
属性的传递闭包: P^+	$(P^+)^I = \bigcup_{i \geq 1} (P^i)^I$
全称量词: $\forall P. C$	$(\forall P. C)^I = \{a \in \Delta^I \mid \forall b. (a, b) \in P^I \rightarrow b \in C^I\}$
全称量词: $\forall P. \{z\}$	$(\forall P. C)^I = \{a \in \Delta^I \mid \forall b. (a, b) \in P^I \rightarrow b = z^I\}$
存在量词: $\exists P. C$	$(\exists P. C)^I = \{a \in \Delta^I \mid \exists b. (a, b) \in P^I \wedge b \in C^I\}$
存在量词: $\exists P. \{z\}$	$(\exists P. C)^I = \{a \in \Delta^I \mid \exists b. (a, b) \in P^I \wedge b = z^I\}$
属性数目下限约束: $\geq n, P$	$(\geq n, P)^I = \{a \in \Delta^I \mid \{b \mid (a, b) \in P^I\} \geq n\}$
属性数目上限约束: $\leq n, P$	$(\leq n, P)^I = \{a \in \Delta^I \mid \{b \mid (a, b) \in P^I\} \leq n\}$

定义3 术语关系: 给定 $O = \langle T, X, T_D, X_D, A_A, T_C, T_R \rangle$, D, E 为2个术语, I 为任意一个解释,

- 1) 如果 $D^I \subseteq E^I$, 则称 E 包含 D , 记为 $D \sqsubseteq E$.
- 2) 如果 $D \sqsubseteq E$ 和 $E \sqsubseteq D$, 则称 D 与 E 等价, 记为 $D \equiv E$.
- 3) 如果 $D \sqsubseteq \neg E$ 和 $E \sqsubseteq \neg D$, 则称 D 与 E 非交, 记为 $D \nabla E$.

定义4 术语定义项: 它是一个等价关系 $C \equiv D$, 表示 C 用 D 来定义, C 称作前项, D 为后项, 其中, C 是原子术语, D 是术语公式. 如代码中 $Mother \equiv Woman \wedge \exists \text{hasChild. Person}$.

术语定义集由一组满足以下条件的术语定义项组成, 表示为 $T_D = \{C_1 \equiv D_1, C_2 \equiv D_2, \dots, C_n \equiv D_n\}$, 其中 $C_i \in T, D_i$ 为术语公式, D_i 中的术语均来自 T .

- 1) 对任何 $i, j (i \neq j, 1 \leq i \leq n, 1 \leq j \leq n)$, 有 $C_i \neq C_j$.
- 2) 如果存在 $C'_1 \equiv D'_1, C'_2 \equiv D'_2, \dots, C'_m \equiv D'_m$, 且 C'_i 出现在 D'_{i-1} 中 ($1 < i \leq m, m \leq n$), 则 C'_i 必不出现在 D'_m 中.

实例集 X 是个体的集合, 实例分为特指实例与泛指实例. 而实例声明集 X_D 则由类的实例声明与属性的实例声明组成:

- 1) 类的实例声明 $C(a)$, 表示个体 a 属于类

C , 如 $\text{Sex}(\text{male})$.

- 2) 属性的实例声明 $P(a, b)$, 表示个体 a, b 之间存在关系 P .

属性分配集将属性术语分配给类术语, 记为 $A_A = \{C_1 \sqsubseteq D_{11} \wedge D_{12} \wedge \dots \wedge D_{1m_1}, C_2 \sqsubseteq D_{21} \wedge D_{22} \wedge \dots \wedge D_{2m_2}, \dots, C_n \sqsubseteq D_{n1} \wedge D_{n2} \wedge \dots \wedge D_{nm_n}\}$. 其中, $C_i \in T$ 为原子类术语, D_i 可以是以下形式:

- 1) $\exists P. D$ 或 $\exists P. \{z\}$: 表示类术语具有属性 P , 并且属性 P 某些值的类型为类 D 或者为个体 z ;
- 2) $\forall P. D$ 或 $\forall P. \{z\}$: 表示类术语具有属性 P , 并且属性 P 所有值的类型均为类 D 或者为个体 z ;
- 3) $\geq n, P$ 或 $\leq n, P$: 表示类术语至少或最多拥有 n 个属性 P .

术语注释集可表示成 $T_c = \{t_c(t_1), \dots, t_c(t_n)\}$, 其中 $t_c(t_i)$ 是对原子术语 $t_i \in T$ 的语义进行自然语言描述. 术语注释可借助语义字典来构造, 如 WordNet.

术语约束集 T_R 由若干个术语关系 (包含关系、等价关系、非交关系) 组成, 用来限定术语之间的关系. 如代码中, $\exists \text{hasSex. \{male\}} \nabla \exists \text{hasSex. \{female\}}$.

作者在文献[27 ~ 29] 对上述本体模型的组成、性质以及一致性检验方法进行了详细描述, 此不赘述.

2 基于本体的信息系统异构性分析

2.1 信息系统语义冲突的两阶段

信息系统之间的语义异构源于所依赖的本体模型之间的语义异构^[30]. 由于系统运行环境、开发者认知能力及建模习惯的不同, 系统间的语义异构在本体模型创建阶段就产生. 因此, 本文认为信息系统的异构性分析应当从本体模型的创建过程入手. 作者在文献[31,32]对本体模型的创建过程进行了研究, 概括起来, 这个过程可分为两个阶段:

1) 第一阶段: 领域概念化阶段. 首先确定特

定领域所涉及的各种概念, 如类、属性、关系、实例和约束, 并通过自然语言将它们含义明确地表达出来, 并尽可能避免歧义. 概念化的结果是产生一系列的概念描述, 包括类描述、实例描述、属性描述、关系描述、约束描述等.

2) 第二阶段: 领域概念化的术语指派阶段. 这一阶段可以看成是对第一阶段生成的各种概念描述的编码阶段, 它的任务就是对第一阶段所识别出的各种概念描述进行术语指派.

与本体模型创建的两阶段相对应, 本文将信息系统的语义异构性也划分为两种类型, 即概念化冲突异构和术语指派冲突异构, 分别简称概念化冲突和术语指派冲突, 如图1所示.

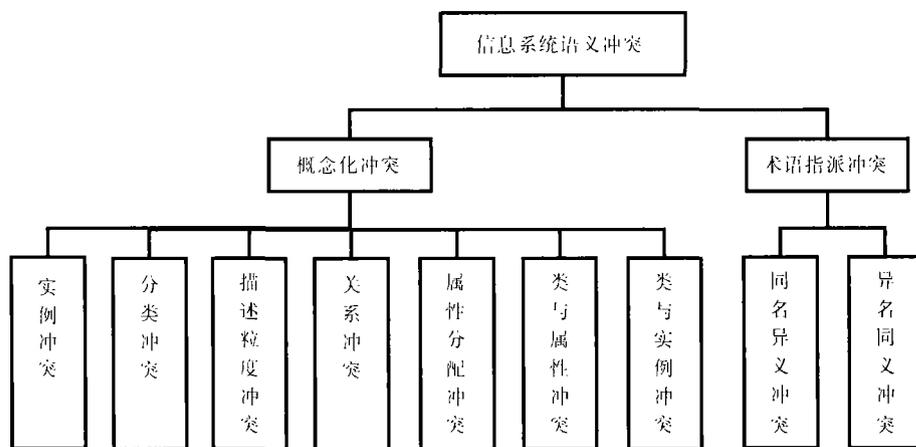


图1 基于本体的信息系统语义异构性的分类图

Fig. 1 Types of semantic heterogeneity between information systems based on ontology

2.2 信息系统语义冲突的类型及解决办法

(1) 符号说明

1) 采用引号标注的中文表示概念化阶段人脑所识别出的概念, 采用英文表示人们在术语指派阶段为概念描述所指派的术语, 即具体的编码. 例如“客户”表示概念描述, Customer表示为“客户”这个概念描述指派的术语.

2) $O_i = \langle T_i, X_i, T_{Di}, X_{Di}, A_{Ai}, T_{Ci}, T_{Ri} \rangle$ 为信息系统 I_i 所依赖的本体模型.

3) R_{i-j} 表示本体 O_i 到 O_j 的转换关系集, 包括“ \equiv ”和“ \sqsubseteq ”操作符.

4) xsd 表示数据类型的名称集合, um 表示度量单位的名称集合.

(2) 概念化冲突

概念化冲突是由于 O_1 与 O_2 对特定领域中的

同一概念的认识差异所造成, 具体而言, 可细分为以下几种类型:

1) 分类冲突: 指 O_1 与 O_2 在概念化阶段均识别出某一个类概念, 但对这个类概念的分类方法不同.

示例1 假设 O_1 与 O_2 均识别出“客户”这个概念, O_1 将“客户”分为“国内客户”和“国外客户”两个概念, 而 O_2 却将“客户”分成“亚洲客户”、“美洲客户”、“欧洲客户”、“非洲客户”四个概念.

形式化表示: 在 O_1 中 $\{Customer, DomesticCustomer, ForeignCustomer\} \subseteq T_1$, $\{Customer \equiv DomesticCustomer \vee ForeignCustomer\} \subseteq T_{D1}$, 而在 O_2 中 $\{Customer, AisanCustomer, EuropeanCustomer, AmericanCustomer, AfricanCustomer\} \subseteq T_2$,

$\{Customer \equiv AisanCustomer \vee EuropeanCustomer \vee AmericanCustomer \vee AfricanCustomer\} \subseteq T_{D2}$.

转换方法:建立 O_1 与 O_2 的转换关系:

$R_{1-2} = \{O_1 : Customer \equiv O_2 : Customer, O_1 : DomesticCustomer \sqsubseteq O_2 : Customer, O_1 : ForeignCustomer \sqsubseteq O_2 : Customer\}$. $R_{2-1} = \{O_2 : Customer \equiv O_1 : Customer, O_2 : AisanCustomer \sqsubseteq O_1 : Customer, O_2 : AfricanCustomer \sqsubseteq O_1 : Customer, O_2 : EuropeanCustomer \sqsubseteq O_1 : Customer, O_2 : AmericanCustomer \sqsubseteq O_1 : Customer\}$.

2) 实例冲突:与分类冲突类似,是指 O_1 与 O_2 在概念化阶段均识别出某一个类概念,但对这个类概念却存在不同的实例化.

示例 2 假设 O_1 与 O_2 均识别出“VIP 客户”这个概念, O_1 将“VIP 客户”细化为“联通”和“移动”两个实例,而 O_2 却将“VIP 客户”分成“电信”、“网通”、“移动”三个实例.

形式化表示:在 O_1 中 $\{VIPCustomer\} \subseteq T_1$, $\{ChinaUnicom, ChinaMobile\} \subseteq X_1$, $\{VIPCustomer(ChinaUnicom), VIPCustomer(ChinaMobile)\} \subseteq X_{D1}$,而在 O_2 中 $\{VIPCustomer\} \subseteq T_1$, $\{ChinaTelecom, ChinaNetcom, ChinaMobile\} \subseteq X_2$, $\{VIPCustomer(ChinaTelecom), VIPCustomer(ChinaNetcom), VIPCustomer(ChinaMobile)\} \subseteq X_{D2}$.

转换方法:建立 O_1 与 O_2 的转换关系: $R_{1-2} = \{O_1 : VIPCustomer \equiv O_2 : VIPCustomer, O_1 : ChinaMobile \equiv O_2 : ChinaMobile, O_1 : ChinaUnicom \sqsubseteq O_2 : VIPCustomer, O_1 : ChinaMobile \sqsubseteq O_2 : VIPCustomer\}$. $R_{2-1} = \{O_2 : VIPCustomer \equiv O_1 : VIPCustomer, O_2 : ChinaMobile \equiv O_1 : ChinaMobile, O_2 : ChinaTelecom \sqsubseteq O_1 : VIPCustomer, O_2 : ChinaNetcom \sqsubseteq O_1 : VIPCustomer, O_2 : ChinaMobile \sqsubseteq O_1 : VIPCustomer\}$.

3) 描述粒度冲突:指 O_1 与 O_2 在概念化阶段虽然都识别出某一个类概念,但对这个类概念的描述粒度不同.

示例 3 O_1 与 O_2 均识别出“客户”这个概念, O_1 直接用一个类概念“客户”来表示,而 O_2 却用“消费者”与“供应商”这两个类概念来表示.

形式化表示: O_1 中 $\{Customer\} \subseteq T_1$,而 O_2 中 $\{Consumer, Supplier\} \subseteq T_2$.

转换方法:建立 O_1 与 O_2 的转换关系: $R_{1-2} =$

$\{O_1 : Customer \equiv O_2 : Consumer \vee O_2 : Supplier\}$, $R_{2-1} = \{O_2 : Consumer \sqsubseteq O_1 : Customer, O_2 : Supplier \sqsubseteq O_1 : Customer\}$.

4) 描述范畴冲突:指 O_1 与 O_2 在概念化阶段均识别出某一个类概念,但对这个类概念的范畴认识却不同.

示例 4 O_1 与 O_2 均识别出“海外客户”这个类概念, O_1 认为“海外客户”包括“亚洲客户”和“美洲客户”、“欧洲客户”三个子类概念,而 O_2 却认为“海外客户”包括“亚洲客户”、“美洲客户”、“欧洲客户”、“非洲客户”四个子类概念.

形式化表示:在 O_1 中 $\{ForeignCustomer, EuropeanCustomer, AmericanCustomer, AisanCustomer\} \subseteq T_1$, $\{ForeignCustomer \equiv AisanCustomer \vee AmericanCustomer \vee EuropeanCustomer\} \subseteq T_{D1}$,而在 O_2 中 $\{ForeignCustomer, AmericanCustomer, EuropeanCustomer, AisanCustomer, AfricanCustomer\} \subseteq T_2$, $\{ForeignCustomer \equiv AmericanCustomer \vee AisanCustomer \vee EuropeanCustomer \vee AfricanCustomer\} \subseteq T_{D2}$.

转换方法:建立 O_1 与 O_2 的转换关系: $R_{1-2} = \{O_1 : ForeignCustomer \sqsubseteq O_2 : ForeignCustomer, O_1 : AisanCustomer \equiv O_2 : AisanCustomer, O_1 : AmericanCustomer \equiv O_2 : AmericanCustomer, O_1 : EuropeanCustomer \equiv O_2 : EuropeanCustomer\}$, $R_{2-1} = \{O_2 : ForeignCustomer \equiv O_1 : ForeignCustomer, O_2 : AisanCustomer \equiv O_1 : AisanCustomer, O_2 : AmericanCustomer \equiv O_1 : AmericanCustomer, O_2 : EuropeanCustomer \equiv O_1 : EuropeanCustomer, O_2 : AfricanCustomer \sqsubseteq O_1 : ForeignCustomer\}$.

5) 关系冲突:指 O_1 与 O_2 在概念化阶段虽然都识别出了两个类概念,但对概念间关系的认识却不相同.

示例 5 O_1 与 O_2 均识别出“制造商”与“产品”这 2 个类,但 O_1 认为“制造商”与“产品”之间是“生产”关系,而 O_2 认为“制造商”与“产品”之间是“出售”关系.

形式化表示: O_1 中 $\{Manufacturer, Product, producing\} \subseteq T_1$, $\{Manufacturer \sqsubseteq \exists producing. Product\} \subseteq A_{A1}$, O_2 中 $\{Manufacturer, Product, selling\} \subseteq T_2$, $\{Manufacturer \sqsubseteq \exists selling. Product\} \subseteq A_{A2}$.

转换方法:关系冲突的解决取决于关系术语的类型,有些关系术语可以通过转换来解决,如关系术语 isMadeFrom 与 hasComponent 之间可以建立关系术语转换,即 $O_1 : \text{isMadeFrom} \sqsubseteq O_2 : \text{hasComponent}$. 而有些关系术语则不可以建立转换,如示例4中 $O_1 : \text{producing}$ 与 $O_2 : \text{selling}$ 在语义上没有任何关系,因此不能建立转换,但这不影响 O_1 与 O_2 之间类术语的转换, O_1 与 O_2 的转换关系: $R_{1-2} = \{O_1 : \text{Manufacturer} \equiv O_2 : \text{Manufacturer}, O_1 : \text{Product} \equiv O_2 : \text{Product}\}, R_{2-1} = \{O_2 : \text{Manufacturer} \equiv O_1 : \text{Manufacturer}, O_2 : \text{Product} \equiv O_1 : \text{Product}\}$.

6) 属性分配冲突:指 O_1 与 O_2 在概念化阶段都识别出相同的属性概念和类概念,却将属性概念分配给不同层次的类概念(父类 - 子类).

示例6 O_1 与 O_2 均识别出类概念“客户”及其子类概念“流失客户”,但 O_1 将属性“流失时间”分配给“客户”, O_2 将该属性分配给“流失客户”.

形式化表示: O_1 中 $\{\text{Customer}, \text{LostCustomer}, \text{leavingTime}\} \subseteq T_1, \{\text{Customer} \sqsubseteq \exists \text{leavingTime. xsd : Time}\} \subseteq A_{A1}, \{\text{LostCustomer} \sqsubseteq \text{Customer}\} \subseteq T_{R1}$, 而 O_2 中 $\{\text{Customer}, \text{LostCustomer}, \text{leaving-Time}\} \subseteq T_2, \{\text{LostCustomer} \sqsubseteq \exists \text{leavingTime. xsd : Time}\} \subseteq A_{A2}, \{\text{LostCustomer} \sqsubseteq \text{Customer}\} \subseteq T_{R2}$.

转换方法:建立 O_1 与 O_2 的转换关系: $R_{1-2} = \{O_1 : \text{Customer} \equiv O_2 : \text{Customer}, O_1 : \text{LostCustomer} \equiv O_2 : \text{LostCustomer}, O_1 : \text{leavingTime} \equiv O_2 : \text{leavingTime}\}, R_{2-1} = \{O_2 : \text{Customer} \equiv O_1 : \text{Customer}, O_2 : \text{LostCustomer} \equiv O_1 : \text{LostCustomer}, O_2 : \text{leavingTime} \equiv O_1 : \text{leavingTime}\}$

7) 属性类型冲突:指 O_1 与 O_2 在概念化阶段都识别出同一个类概念,并为其指定了相同的属性集,但属性的值类型不同.

示例7 O_1 与 O_2 均识别出“葡萄酒”这个类,并为其指定“颜色”这个属性,但 O_1 将“颜色”分成红、白和玫瑰红, O_2 将“颜色”值限定为红和白.

形式化表示: 在 O_1 中, $\{\text{Wine}, \text{hasColor}, \text{Color}\} \subseteq T_1, \{\text{red}, \text{white}, \text{rose}\} \subseteq X_1, \{\text{Color}(\text{red}), \text{Color}(\text{white}), \text{Color}(\text{rose})\} \subseteq X_{D1}, \{\text{Wine} \sqsubseteq \exists \text{hasColor. Color}\} \subseteq A_{A1}$, 在 O_2 中, $\{\text{Wine}, \text{hasColor}, \text{Color}\} \subseteq T_1, \{\text{red}, \text{white}\} \subseteq X_2, \{\text{Color}(\text{red}), \text{Color}(\text{white})\} \subseteq X_{D2}, \{\text{Wine} \sqsubseteq$

$\exists \text{hasColor. Color}\} \subseteq A_{A2}$.

转换方法:建立 O_1 与 O_2 的转换关系: $R_{1-2} = \{O_1 : \text{Wine} \equiv O_2 : \text{Wine}, O_1 : \text{Color} \equiv O_2 : \text{Color}, O_1 : \text{red} \equiv O_2 : \text{red}, O_1 : \text{white} \equiv O_2 : \text{white}\}, R_{2-1} = \{O_2 : \text{Wine} \equiv O_1 : \text{Wine}, O_2 : \text{Color} \sqsubseteq O_1 : \text{Color}, O_2 : \text{red} \equiv O_1 : \text{red}, O_2 : \text{white} \equiv O_1 : \text{white}\}$.

示例8 O_1 与 O_2 均识别出“长度”这个属性,但 O_1 中,长度的单位为米,而 O_2 中,长度的单位为厘米.

形式化表示: 在 O_1 中, $\{\text{Car}, \text{Length}, \text{hasLength}, \text{hasUnit}\} \subseteq T_1, \{\text{Car} \sqsubseteq \exists \text{hasLength. Length}, \text{Length} \sqsubseteq \exists \text{dataType. xsd : Decimal} \wedge \exists \text{hasUnit. um : Meter}\} \subseteq A_{A1}$, 在 O_2 中, $\{\text{Car}, \text{Length}, \text{hasLength}, \text{hasUnit}\} \subseteq T_2, \{\text{Car} \sqsubseteq \exists \text{hasLength. Length}, \text{Length} \sqsubseteq \exists \text{dataType. xsd : Decimal} \wedge \exists \text{hasUnit. um : Centimeter}\} \subseteq A_{A1}$.

转换方法:建立 O_1 与 O_2 的转换关系: $R_{1-2} = \{O_1 : \text{Car} \equiv O_2 : \text{Car}, O_1 : \text{Length} \equiv O_2 : \text{Length}\}, R_{2-1} = \{O_2 : \text{Car} \equiv O_1 : \text{Car}, O_2 : \text{Length} \equiv O_1 : \text{Length}\}$.

8) 类与属性冲突:指 O_1 与 O_2 在概念化阶段都识别出同一个类概念, O_1 通过一个新建的类来描述,而 O_2 通过在已有类的基础上引入一个属性来描述.

示例9 O_1 与 O_2 均识别出“男客户”与“女客户”这两个类概念, O_1 为此分别建了两个类,而 O_2 则在已有的“客户”这个类中加入“性别”这一属性.

形式化表示: 在 O_1 中, $\{\text{Customer}, \text{FemaleCustomer}, \text{MaleCustomer}\} \subseteq T_1, \{\text{MaleCustomer} \sqsubseteq \text{Customer}, \text{FemaleCustomer} \sqsubseteq \text{Customer}\} \subseteq T_{R1}$, 在 O_2 中, $\{\text{Customer}, \text{hasSex}, \text{Sex}\} \subseteq T_1, \{\text{male}, \text{female}\} \subseteq X_2, \{\text{Sex}(\text{male}), \text{Sex}(\text{female})\} \subseteq X_{D2}, \{\text{Customer} \sqsubseteq \exists \text{hasSex. Sex}\} \subseteq A_{A2}$.

转换方法: 建立 O_1 与 O_2 的转换关系: $R_{1-2} = \{O_1 : \text{Customer} \equiv O_2 : \text{Customer}, O_1 : \text{FemaleCustomer} \equiv O_2 : \text{Customer} \wedge \exists O_2 : \text{hasSex. } \{O_2 : \text{female}\}, O_1 : \text{MaleCustomer} \equiv O_2 : \text{Customer} \wedge \exists O_2 : \text{hasSex. } \{O_2 : \text{male}\}\}, R_{2-1} = \{O_2 : \text{Customer} \equiv O_1 : \text{Customer}\}$

9) 类与实例冲突:指 O_1 与 O_2 在概念化阶段都识别出同一个类概念, O_1 通过创建一个新类来

描述,而 O_2 通过创建已有类的实例来描述.

示例 10 O_1 与 O_2 均识别出“研究生”这个概念,以及“硕士生”和“博士生”两个子概念.在 O_1 中将“硕士生”、“博士生”视为“研究生”的实例,而在 O_2 中将它们视为“研究生”的子类.

形式化表示:在 O_1 中, $\{Postgraduate\} \subseteq T_1$, $\{masterCandidate, doctorCandidate\} \subseteq X_1$, $\{Postgraduate (masterCandidate), Postgraduate (doctorCandidate)\} \subseteq X_{D1}$, 而 O_2 中, $\{Postgraduate, MasterCandidate, DoctorCandidate\} \subseteq T_2$, $\{MasterCandidate \sqsubseteq Postgraduate, DoctorCandidate \sqsubseteq Postgraduate\} \subseteq T_{R2}$.

转换方法:建立 O_1 与 O_2 的转换关系: $R_{1-2} = \{O_1 : Postgraduate \equiv O_2 : Postgraduate, O_1 : masterCandidate \equiv O_2 : MasterCandidate, O_1 : doctorCandidate \equiv O_2 : DoctorCandidate\}$, $R_{2-1} = \{O_2 : Postgraduate \equiv O_1 : Postgraduate, O_2 : MasterCandidate \equiv O_1 : masterCandidate, O_2 : DoctorCandidate \equiv O_1 : doctorCandidate\}$.

(3) 术语指派冲突

术语指派冲突是指,在概念化阶段识别出了概念后,随后在为这些概念指定术语(即编码)时产生的冲突.

1) 异名同义冲突:指 O_1 与 O_2 在概念化阶段都识别出同一个类概念,而 O_1 和 O_2 分别采用不同的术语来描述.

示例 11 O_1 与 O_2 在概念形成阶段均识别出“黄金客户”这个概念,但分别采用术语“VIPCustomer”与“GoldenCustomer”来表示.

转换方法:建立 O_1 与 O_2 的转换关系: $R_{1-2} = \{O_1 : VIPCustomer \equiv O_2 : GoldenCustomer\}$, $R_{2-1} = \{O_2 : GoldenCustomer \equiv O_1 : VIPCustomer\}$.

2) 同名异义冲突:指 O_1 与 O_2 在概念化阶段分别识别出两个不同的类概念,而 O_1 和 O_2 却分别采用同一个术语来描述.

示例 12 O_1 与 O_2 在概念形成阶段分别识别出“便携式电脑”、“笔记簿”这两个概念,但都采用了术语“Notebook”来表示,

转换方法:在系统集成时,同名异义冲突解决办法通常是重新命名,或建立 O_1 与 O_2 的转换关系: $R_{1-2} = \{O_1 : Notebook \frown O_2 : Notebook\}$, $R_{2-1} = \{O_2 : Notebook \frown O_1 : Notebook\}$.

现实中的异构系统往往是多种类型的冲突交织在一起(形成路径如图 2 所示),最终形成更为复杂的系统间异构性.但不论多么复杂,最终都可以归结为上述基本类型的组合.

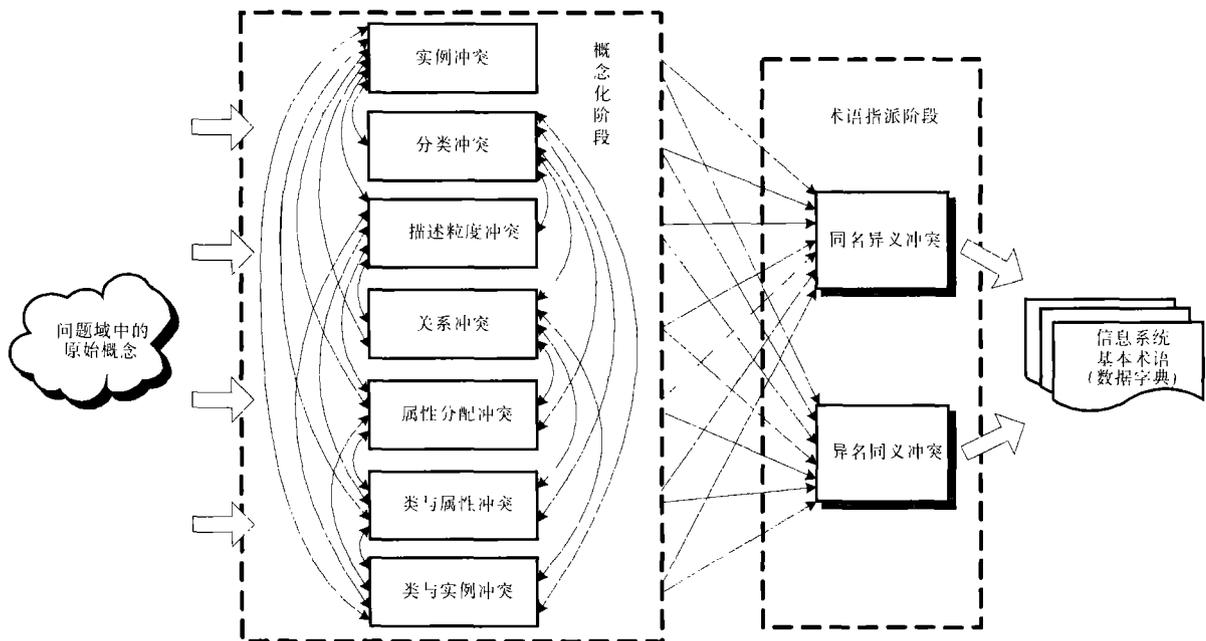


图 2 信息系统语义异构的形成路径示意图

Fig. 2 Paths of causing semantic heterogeneity of information systems

3 基于本体的信息系统集成模式

3.1 基于本体映射的系统集成框架

任何信息系统都对应于某个问题域(problem domain),域中所有术语及相互关系就是该系统所依赖的领域本体.同种类型的信息系统由于开发者不同,所依赖的本体模型通常也不同.为了实现异构系统的语义集成,就应当建立本体模型之间的映射.有两种映射方式:一是在任意两个系统的本体模型之间转换;一是以某个系统的本体模型为中介,将其他系统的本体模型映射到该中介上.前者过于复杂,因为如果存在 n 个本体模型,就要建立 C_n^2 个转换,而且某个本体模型一旦变动,就要相应地调整 $n - 1$ 个转换,而后者可以克

服前者的缺陷.

中介本体是指公认的、具有权威性的本体模型.中介机构可以是行业协会、标准化组织、业内巨头,也可以由企业通过协商产生.例如,动态联盟中各盟员都具有自己的本体模型,这些本体都是为了提高各自的效率而量身定制的,它们之间可能存在异构性冲突,从而影响联盟内的信息共享.这种情况下,盟主可将自己的本体模型作为中介,各盟员在加入时,需要将自己的本体模型映射到中介转换平台上,从而实现整个联盟的信息共享.

图3给出了基于本体映射的系统集成框架,它包括数据请求处理器、具有中介功能的本体模型和若干个本体转换集,转换集描述了不同信息源的本体模型与中介的转换关系,转换集的格式已在第3节中说明.

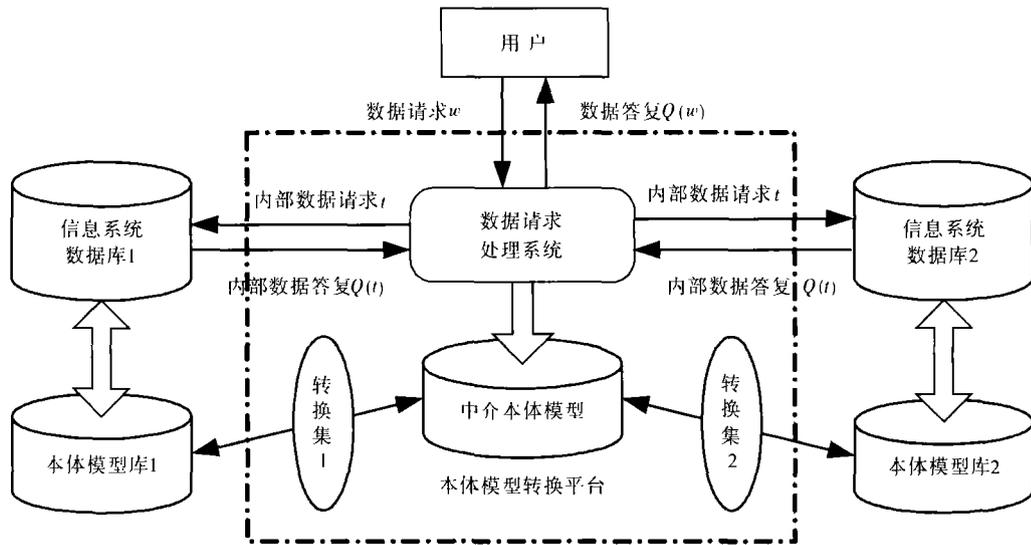


图3 基于本体映射的信息系统语义集成框架

Fig. 3 Integration framework for information systems in semantic based on ontology mapping

3.2 系统集成框架下的数据请求与数据答复

1) 数据请求模式

信息共享,实际上是一个当用户向系统发出数据请求,系统从语义层面上理解用户的需求,并返回合适的数据给用户的过程.为此,本文将信息共享抽象为数据请求和数据答复过程.

定义1 数据请求:给定信息系统 $I_s < D, O >$,其中 D 是数据, O 是系统所依赖的本体, D 在 I_s 中是按照 O 的术语分类组织的.用户向 I_s 提交的数据请求为 ω ,采用递归形式可以抽象表示为:

$$\omega ::= t \mid \omega \wedge \omega' \mid \omega \vee \omega' \mid \omega \wedge \neg \omega' \mid (\omega) \mid \varepsilon \tag{1}$$

其中, t 为用户提交的关键词(包括类术语或个体), ε 为空请求.由式1可知,数据请求 ω 实际上就是若干个关键词 t 经过逻辑运算,形成的概念表达式,它反映了用户对数据请求的意愿.

定义2 信息答复:给定信息源 $I_s = < D, O >$,用户提交的数据请求为 ω ,答复模式 Q 就是 I_s 根据 ω 的要求对数据所采用的查询策略,返回

的查询结果为 $Q(\omega)$.

性质1 令 Q 是一个答复模式, ω 为一个数据请求, 有以下式子成立:

- 1) $Q(\omega \wedge \omega') = Q(\omega) \cap Q(\omega')$;
- 2) $Q(\omega \vee \omega') = Q(\omega) \cup Q(\omega')$;
- 3) $Q(\omega \wedge \neg \omega) = Q(\omega) \setminus Q(\omega')$;
- 4) $Q(\varepsilon) = \emptyset$.

性质1的证明省略. 由式1可知, 数据请求 ω 就是若干个关键词 t 经过逻辑运算形成的概念表达式. 只要得到 $Q(t)$, 再根据性质1进行集合运算, 就能够得到用户所需要的信息 $Q(\omega)$. 对于 $Q(t)$, 可按照下面查询策略获得.

2) 系统集成框架下的数据搜索策略

在查询各系统的数据时, 用户参照中介本体模型上有关术语, 提交数据请求. 数据请求处理系统根据不同的本体模型转换集, 将提交的数据请求转换成对各个信息系统的数据请求, 然后分别发送给不同的系统. 信息系统在接受到各自的数据请求后, 按照一定的规则搜索数据, 然后将搜索结果返回给数据请求处理系统. 数据请求处理系统将各系统返回的数据汇总后, 一并发送给用户. 本文建立了以下搜索规则:

1) 当用户提交关键词 t , 并且转换集 R_i 中存在关系 $t \equiv t_1, t \equiv t_2, \dots, t \equiv t_n$, 则客户资源数据库 i 返回的结果为 $Q(t) = r(t_1) \cup r(t_2) \cup \dots \cup r(t_n)$, 其中 $r(t)$ 为资源数据库中以概念 t 为主题进行索引的资源集.

2) 当用户提交关键词 t , 并且转换集 R_i 中存在关系 $t \equiv t_1 \vee t_2 \vee \dots \vee t_n$, 则客户资源数据库 i 返回的结果为 $Q(t) = r(t_1) \cup r(t_2) \cup \dots \cup r(t_n)$.

3) 当用户提交关键词 t , 并且转换集 R_i 中存在关系 $t \equiv t_1 \wedge t_2 \wedge \dots \wedge t_n$, 则客户资源数据库

i 返回的结果为 $Q(t) = r(t_1) \cap r(t_2) \cap \dots \cap r(t_n)$.

4) 当用户提交关键词 t , 并且转换集 R_i 中存在关系 $t \sqsubseteq t_1, t \sqsubseteq t_2, \dots, t \sqsubseteq t_n$, 则客户资源数据库 i 返回的结果为 $Q(t) = r(t_1) \cup r(t_2) \cup \dots \cup r(t_n)$.

5) 当用户提交关键词 t , 并且转换集 R_i 中存在关系 $t \sqsupseteq t_1, t \sqsupseteq t_2, \dots, t \sqsupseteq t_n$, 则客户资源数据库 i 返回的结果为 $Q(t) = r(t_1) \cap r(t_2) \cap \dots \cap r(t_n)$.

规则1)、2)、3)、4)的查准率较高, 而规则5)可以提高系统的查全率.

4 结束语

语义层面上的系统集成一直是信息化建设的难题. 本文借鉴本体的思想, 利用描述逻辑建立本体形式化模型, 从本体创建过程着手分析异构系统语义冲突的类型, 并提出异构系统间术语的映射方法. 最后建立异构系统集成框架, 将数据共享抽象为数据请求与数据答复过程, 并对数据答复过程所采用的搜索策略进行分析.

相对于传统的基于语法的集成方案(如数据字典或元数据映射等方法), 基于语义的系统集成方案, 能够借助逻辑推理实现术语的自动映射, 从而克服基于语法的方案必须通过穷举来实现映射的不利之处. 本文构建的本体模型是采用描述逻辑来描述的, 因篇幅所限, 基于描述逻辑的本体推理请参考[27~29]. 此外, 还有一些方面有待深入研究: (1) 复杂异构问题的快速简化方法; (2) 领域本体模型的工程化建立方法; (3) 基于描述逻辑的高效本体推理算法.

参考文献:

- [1] Ceri S, Widom J. Managing semantic heterogeneity with production rules and persistent queues[A]. Proc. of the 19th VLDB Conference[C]. Dublin, Ireland, 1993. 108—119.
- [2] Hajime Kitakami, Yasuma Mori, Masatoshi Arikawa. An intelligent system for integrating autonomous nomenclature databases in semantic heterogeneity[A]. Database and Expert System Applications, DEXA '96[C]. Zürich, Switzerland, 1996. 187—196.
- [3] Kim W, Seo J. Classifying schematic and data heterogeneity in multi-database systems[J]. IEEE Computer, 1994,

- 24(12): 12—18.
- [4] 曾国荪, 陈洪中. 探索信息系统的异构性问题[J]. 计算机工程与应用, 2003, 19: 1—4, 54.
Zeng Guosun, Chen Hongzhong. Probing the heterogeneity problems for information systems[J]. Computer Engineering and Applications, 2003, 19: 1—4, 54. (in Chinese)
- [5] Nicola Guarino. Formal ontology and information system[A]. In Nicola Guarino eds. Formal Ontology in Information System [C]. IOS Press, 1998.
- [6] Cui Z, V A M Tamma, F Bellifemine. Ontology management in enterprise[J]. Journal of BT Technology, 1999, 17(4): 98—107.
- [7] Gruber T R. Toward principles for the design of ontologies used for knowledge sharing[J]. International Journal of Human and Computer Studies, 1995, 43(5/6): 907—928.
- [8] Guarino, Nicola. Formal ontology, conceptual analysis and knowledge representation[J]. International Journal of Human and Computer Studies, 1995, 43(5/6): 625—640.
- [9] Levashova T V, Pashkin M P, Smirnov A V. Ontology management[J]. Journal of Computer and Systems Sciences International, 2003, 42(4): 621—633.
- [10] Levashova T V, Pashkin M P, Shilov N G. Ontology management II[J]. Journal of Computer and Systems Sciences International, 2003, 42(5): 744—756.
- [11] Aitken Stuart, Reid Sandy. Evaluation of an ontology-based information retrieval tool[A]. Proc. of the Workshop on Applications of Ontologies and Problem-Solving Methods of 14th European Conference on Artificial Intelligence [C]. Berlin: AMIA Inc., 2000. 497—501.
- [12] Saias José, Quaresma Paulo. A methodology to create ontology-based information retrieval systems[J]. Lecture Notes in Artificial Intelligence. Springer-Verlag, 2003, 2902: 424—434.
- [13] Wache H, Vogele T, Visser U, *et al.* Ontology-based integration of information: A survey of existing approaches[A]. Proc. of the IJCAI-01 Workshop on Ontologies and Information Sharing [C]. Seattle, USA, August 4—5, 2001.
- [14] 程少川, 程向前, 张朋柱. 基于本体系统的知识管理人工智能研究[J]. 管理科学学报, 2003, 6(2): 17—22.
Cheng Shaochuan, Cheng Xiangqian, Zhang Pengzhu. Study on artificial intelligence of knowledge management based on ontology[J]. Journal of Management Sciences in China, 2003, 6(2): 17—22. (in Chinese)
- [15] Gramajo J, Riano D. A knowledge management platform to extract and process information from the web[A]. 5th Joint Conference on Knowledge Based Software Engineering [C]. Maribor, Slovenia, 2002.
- [16] Heflin J D. Towards the Semantic Web: Knowledge Presentation in a Dynamic, Distributed Environment[D]. Maryland: University of Maryland, USA, 2001.
- [17] Lenat D B, Guha R V. Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project[M]. MA: Addison-Wesley, 1990.
- [18] Christiane Fellbaum. WordNet: An Electronic Lexical Database[M]. Cambridge: MIT Press, 1998.
- [19] Fox M S. The Tove project: Towards a common sense model of the enterprise[A]. In: C Petrie eds. Enterprise Integration [C]. AAAI Press, 1992. 291—316.
- [20] Uschold M, King M, Moralee S, *et al.* The enterprise ontology[J]. The Knowledge Engineering Review, 1998, 13(1): 31—89.
- [21] Genesereth M R, Fikes R E. KIF Version 3.0. Reference Manual[R]. Technical Report Logic-92-1, Stanford: Stanford University, 1992.
- [22] MacGregor. Inside the LOOM classifier[J]. SIGART Bulletin, 1991, 2(3): 70—76.
- [23] 朱立, 范启通, 胡运发等. 基于 F-logic 的概念语义网[J]. 计算机工程, 1997, 23(6): 48—53.
Zhu Li, Fan Qitong, Hu Yunfa, *et al.* Semantic Network Based on F-logic[J]. Computer Engineering, 1997, 23(6): 48—53. (in Chinese)
- [24] 金芝. 知识工程中的本体论研究[A]. 陆汝钫主编: 世纪之交的知识工程与知识科学[C]. 北京: 清华大学出版社, 2001: 447—465.
Jin Zhi. Study on ontology in knowledge engineering[A]. In: Nicola Guarino eds. Knowledge Engineering and Knowledge

- Science Crossing the Century[C]. Beijing: Tsinghua University Press, 2001: 447—465. (in Chinese)
- [25] Baader F, McGuinness D, Nardi D, *et al.* The Description Logic Handbook: Theory, Implementation and Applications [M]. Cambridge: Cambridge University Press, 2002.
- [26] Borgida A. On the relative expressiveness of description logics and predicate logics[J]. *Artificial Intelligence*, 1996, 82(1—2): 353—367.
- [27] Yu Changrui, Wang Hongwei, Luo Yan. Extended ontology model and ontology checking based on description logics[A]. The 3rd International Conference on Fuzzy Systems and Knowledge Discovery[C]. Lecture Notes in Artificial Intelligence, Springer, LNAI 4223: 607—610.
- [28] Wang Hongwei, Jiang Fu, Wu Jiachun. Study on formal ontology model: Constructing customer ontology in CRM context [A]. Proc. of 9th Americas Conference on Information System[C]. Tampa, Florida, August 4—6, 2003.
- [29] 王洪伟, 侯立文, 蒋 馥. 基于本体的元数据扩展模型的检验[J]. *系统工程理论与实践*, 2006, 26(10): 57—66.
Wang Hongwei, Hou Liwen, Jiang Fu. Checking problems of ontology-based metadata extended models[J]. *Systems Engineering-theory & Practice*, 2006, 26(10): 57—66. (in Chinese)
- [30] P R S Visser, D M Jones, T J M Bench-Capon, *et al.* Assessing Heterogeneity by Classifying Ontology Mismatches. Proc. of International Conference on Formal Ontology in Information Systems[C]. FOIS'98, IOS Press, 1998.
- [31] 王洪伟. 基于本体的元数据模型建立研究[D]. 上海: 上海交通大学, 2004.
Wang Hongwei. Research on the Construction of Metadata Models Based on Ontology[D]. Shanghai: Shanghai Jiaotong University, 2004. (in Chinese)
- [32] 王洪伟, 伊 磊, 王洪滨. 面向关系模式的领域本体获取[J]. *计算机工程*, 2007, 33(3): 1—3, 23.
Wang Hongwei, Yi Lei, Wang Hongbin. Acquisition of domain ontology from relational schemes[J]. *Computer Engineering*, 2007, 33(3): 1—3, 23. (in Chinese)

Study on semantic heterogeneity of information systems based on ontology

WANG Hong-wei, HUO Jia-zhen, LIU Zhong-ying

School of Economics and Management, Tongji University, Shanghai 200092, China

Abstract: Every information system has its own domain model for its environment and efficient task operations, which results in diverse heterogeneities, especially the semantic heterogeneity. So a shared domain model is required in order to make heterogeneous systems communicate effectively and enable information sharing. Ontology is an agreement about shared conceptualizations, which can make the meaning of the different vocabularies used explicit, and so the semantics of diverse information systems can be captured by ontology-definitions of terms. This paper constructs an ontology model using description logic, divides the types of heterogeneity of information system into two categories from the ontological view: conceptualization conflict and coding conflict, and analyses approaches to solve conflict by ontology mapping. Finally, this paper constructs a semantic-oriented integration architecture for heterogeneous systems based on ontology mapping, and investigates the model for data request and data response under the architecture, and proposes five rules for data search.

Key words: ontology; heterogeneity; information system; description logic