MD4:一种综合的跨本体实体语义相似度计算方法*

黄宏斌,张维明,邓 苏,董发花
(国防科学技术大学 信息系统与管理学院,长沙 410073)

摘 要:面向广域分布环境下信息资源共享与服务的需要,设计了基于本体的元数据模型,并在MD3模型的基础上给出了一种基于该元数据模型的跨本体的语义相似度计算方法——MD4模型。MD4充分利用本体对实体的描述信息,重点讨论了实体名称、实体属性、实体语义环境以及实体实例等相似度的计算,把MD3模型扩展到MD4模型,使得信息资源实体间语义相似度的计算更全面、精确。

关键词:元数据模型;本体;语义相似度;MD4模型
中图分类号:TP311.12 **文献标志码**:A **文章编号**:1001-3695(2008)08-2335-04

MD4:integrated approach of determining semantic similarity among entity between different ontologies

HUANG Hong-bin, ZHANG Wei-ming, DENG Su, DONG Fa-hua
(*College of Information System & Management, National University of Defense & Technology, Changsha* 410073, *China*)

Abstract: Oriented the demand of sharing information and service in the distribute network, this paper designed the metadata model based ontology. The MD3 model systematically evaluated semantic similarity across different ontologies dispense with integrating different ontologies into a shared ontology. Based on the MD3 model, focused on the not-hierarchical relations evaluating of concepts from different ontologies, extended the MD3 model to MD4 model to make the semantic similarity of concepts from different ontologies more comprehensive and precise.

Key words: metadata model; ontology; semantic similarity; MD4 model

随着网络的发展,在一些大型企事业团体和虚拟组织环境中存在着大量的业务信息系统。组织内的各单位依据业务或专业知识进行分布,并且管理着部分信息资源,不同单位有着不同的工作重心和目标,因此在不同工作领域下对底层的信息资源采用不同的概念实施建模。这些广域网络环境下的信息资源具有动态、分布、多元和无序等特点,各部门或组织之间的信息相对封闭,形成了一个个信息孤岛,造成信息资源的浪费,同时信息资源的异构性为信息之间的共享与互操作带来了困难。对用户而言,访问并有效利用这些信息资源、实现各节点分布信息资源的共享和交换以及分布信息系统之间的互操作成为一个亟需解决的问题。

本体在人工智能、信息检索、Web服务发现等领域中扮演着越来越重要的角色,利用本体对信息资源进行建模能够很好地反映信息资源的语义。基于本体的元数据通过本体来表示元素模型,将有效地促进领域知识的共享和语义表达,能够为用户和应用提供语义查询和信息汇集能力。但是,现有基于本体的信息共享与服务系统中几乎都假设共享一个集中的全局本体,用来访问信息资源,这样的维护开销比较大。但在广域网络环境下,这种假设是不存在的。在网络环境中每一个节点仅维护本地信息资源的局部领域视图,通过网络中节点的协作来发现、理解和使用领域范围内分布、异构、不断变化的信息资源,异构问题越来越严重,如何解决信息资源的互操作成为一个比较棘手的问题。本体映射和本体集成是解决信息资源语义异构很好的方法,其中一个关键步骤就是语义相似度的计算。如何提高语义相似度计算精度也是提高语义信息检索质量的关键之一。语义相似度一般是指计算本体概念间的相似度,多数方法所考虑的概念是基于一个本体的,跨本体概念间的方法比较少,MD3(triple matching-distance model)模型是一种典型的计算跨本体概念间相似度的方法[1]。

1 常用语义相似度计算方法

1.1 基于特征匹配的模型

Tversky模型是典型的特征匹配模型,提出了基于集合理论,用特征匹配过程来计算相似度的方法[2]。它指出相似度不仅由两个概念的相同属性决定,而且由它们的不同属性决定。这种模型没有考虑属性与概念间的密切程度。由于该模型符合信息领域对相似度的认知,在信息领域有众多学者在Tversky模型的基础上建立语义相似度模型,如文献[3~5]等。

1.2 基于语义距离的模型

基于语义距离的模型根据概念在层次节构中的位置来计算语义相似度。层次节构中节点表示概念,边表示语义关系。在概念层次树中,任何两个节点之间有且只有一条最短路径,把这条路径的长度作为两个概念的语义距离的度量。该模型适合很多专业领域,但它过分依赖事先定义好的语义网结构,只考虑了概念间的聚合(is~a)关系,计算出来的有相同父类的子类之间的相似度值很粗糙。

收稿日期:2006-07-20;**修回日期**:2007-11-26 **基金项目**:国家自然科学基金资助项目(70771110)

作者简介:黄宏斌(1975-),男,江苏如皋人,博士研究生,主要研究方向为信息综合处理、辅助决策(hongbinhuang2000@yahoo.com.cn);张维明(1962-),男,教授,博导,博士,主要研究方向为信息系统、智能决策;邓苏(1963-),男,教授,博导,博士,主要研究方向为数据仓库、辅助决策.

1.3 基于信息内容的方法

基于信息内容的方法[4]不依赖于层次结构的词典，只要有概率模型存在，这种方法就可以应用。两个概念的相似性决定于它们共享信息的程度，共享信息含量越大，两个概念越相似。两个概念的共享信息含量用两个概念在概念树上的所有共同“超类”所具有的最大信息含量来表示。

1.4 混合模型

MD(matching-distance)模型是一种典型的混合模型[6]，该模型以 Tversky 的比率模型[2]为基础。其中，特征权重是根据实体之间的语义关系来确定的。MD 模型综合了基于特征匹配和基于语义距离两种方法，该方法在语言学上有很深的根基，并定义了非对称的语义相似度计算函数。

2 MD3 模型

概念间相似度的计算方法根据要比较的概念是否来自同一个本体分为单本体和跨本体概念相似度计算方法。基于语义距离和基于信息内容的方法分别利用了本体的结构信息(概念术语的位置信息)和信息内容，不同本体的结构和信息内容不能直接进行比较，这两种方法适合单本体概念相似度的计算。跨本体的相似度方法通常使用混合的或基于特征的方法。传统的跨本体概念间相似度的计算通过手工或半自动方式把不同本体集成为一个共享本体，然后把本地本体中的概念映射到这个共享本体，再利用单本体概念间相似度的计算方法来完成跨本体概念间的比较。在某些情况下，这种集成共享本体的方法是不现实的，或者形成这个共享本体的代价很大，必须考虑直接针对两个不同本体来比较其概念。其中，MD3 模型是一种典型的计算跨本体概念间语义相似度的方法，是 MD 模型在跨本体应用中的扩展。MD3 模型[1]用一个通用的虚拟的根把两个本体联系起来，分别计算概念名称、特征属性以及语义邻居之间的相似度。

MD3 模型是一种跨本体概念间相似度计算框架。计算实体类 a 和 b 之间的相似度通过计算同义词集、特征属性和语义邻居之间的加权和，公式如下：

$$\mathrm{sim}(a,b) = wS_{\mathrm{synelts}}(a,b) + uS_{\mathrm{features}}(a,b) + vS_{\mathrm{neighborhoods}}(a,b) \quad (1)$$

其中：w、u、v 表示了各组成部分的重要性。特征属性细化为组成部分、功能以及其他属性。概念 a 和 b 的语义邻居及其特征属性(即概念的部分、功能及其他属性)也通过同义词集合描述，每一个相似度的计算都通过 Tversky[4] 公式：

$$S(a,b) = |A\cap B|/[\,|A\cap B| + \alpha(a,b)|A-B| + (1-\alpha(a,b))|B-A|\,] \quad (2)$$

其中：A、B 分别表示概念 a 和 b 的描述集合；$A-B$ 表示属于 A 但不属于 B 的术语集($B-A$ 相反)。参数 $\alpha(a,b)$ 由概念 a 和 b 在各自层次结构中的深度确定，公式如下：

$$\alpha(a,b) = \begin{cases} \mathrm{depth}(a)/(\mathrm{depth}(a)+\mathrm{depth}(b)); & \mathrm{depth}(a) \leqslant \mathrm{depth}(b) \\ 1-\mathrm{depth}(a)/(\mathrm{depth}(a)+\mathrm{depth}(b)); & \mathrm{depth}(a) > \mathrm{dsepth}(b) \end{cases} \quad (3)$$

3 基于本体的元数据模型

本体(ontology)是领域知识的概念化说明，它将特定领域有关的对象、概念及其关系以形式化的说明来严格规定。利用本体建立面向语义的元数据模型，可以将元数据中实体类的含义、类间及对象间的关系更加明确地表达出来，从而支持广域分布环境下的概念建模、信息搜索与交换、信息资源共享与服务系统建设等研究。

假设 T 为领域语义词典(domain semantic dictionary, DSD)，是领域中术语词汇的集合，主要包括类术语、属性术语、关系术语等。D_{basic} 是预定义的基本数据类型集合；D_{enum} 是预定义的枚举类型。领域语义词典用来规范元数据中元素的命名，定义了词汇间的语义关系，DSD 是一个受限的词汇知识库，面向的是领域的常用词汇和专用词汇。其目标是：a)规范不同信息领域中描述信息资源的词汇，从而可以有效地限制任意词汇描述引发的不一致性；b)它是中词汇语义相似度计算的基础。DSD 采用《知网》的语义描述结构进行组织，主要对实词进行描述，是一种开放结构，可扩充。

基于信息资源都是信息单元组成这一理解，采用信息单元实体类和实体类之间的关系以及实体对象、相关约束和规范描述信息资源。具体定义如下：

定义 1 元数据模型是一个八元组 $MD:=\langle E,A,L,H^C,R,I,F,P\rangle$。其中：

a) E 是信息单元实体类集合，$\forall c\in E, c=(\mathrm{name},A^c)$，$\mathrm{name}(c)$ 表示 c 的命名词汇，$\mathrm{name}(c)\in T$。其中，$A^c=\{x\mid x(x\in A)\cap(\mathrm{att}(x)=c)\}$。att 是函数集中的属性映射函数。

b) A 是信息单元实体类的属性集合，属性又分为基础属性和复合属性。$\forall a\in A, a=(\mathrm{name},dt)$，$\mathrm{name}(a)$ 表示 a 的命名词汇，$\mathrm{name}(a)\in T, dt\in D_{\mathrm{basic}}\cup D_{\mathrm{enum}}$。

c) L 取值域集，$L=D_{\mathrm{basic}}\cup D_{\mathrm{enum}}$。

d) H^C 表示类间的二元层次关系，层次关系是有向传递的偏序关系，包括了类间的继承 is-a 关系和聚合(组合) part-of 关系。

e) R 是实体类之间的二元关系集。任何所连接的实体类超过两个的关系都能够转换为一组二元多对关系集合，因此，该模型中的关系设计为二元关系。

f) I 为实例集，是信息单元实体类对象集合。

g) F 表示函数集，主要包括如下函数：

(a) $\mathrm{att}: A\rightarrow E\cup R$，属性函数，将属性分配给某一实体类或关系；

(b) $\mathrm{val}: V\rightarrow(E,A)$，属性取值函数；

(c) $\mathrm{inst}: E\rightarrow 2^I$，信息单元实例化函数，可以写为 $\mathrm{inst}(E)=I$ 或 $\mathrm{E(I)}$[8]；

(d) $instr: R\rightarrow 2^{I\times I}$，关系实例化函数，可以写为 $\mathrm{instr}(R)=\{I_1,I_2\}$ 或 $R(I_1,I_2)$[8]。

(e) $\mathrm{dom}: R\rightarrow E$，由 $\mathrm{dom}(R):=\Pi_1(rel(R))$ 给出其定义域[1]；

(f) $\mathrm{range}: R\rightarrow E$，由 $\mathrm{range}(R):=\Pi_2(rel(R))$ 给出其值域[1]；

h) P 为约束规则集，所有信息单元实体类和关系要满足的约束的集合。

定义 2 给定 $MD:=\langle E,A,H^C,R,L,I,F\rangle$；语义解释为一个三元组，记为 $I=\langle\Delta^I,\Delta^L,\cdot^I\rangle$。其中，$\Delta^I$ 是一个非空集合，包含领域中的所有个体；Δ^L 是一个非空集合，包含领域中的所有数据值；$\cdot^I$ 是解释函数，它将 E 中的每个实体类 C 都映射为 Δ^I 的一个子集 $C^I(C^I\subseteq\Delta^I)$。每个具体域 L 解释成一个集合

$L^I(L^I \subseteq \Delta^L)$，每个关系 R 解释成一个二元关系 $R^I(R^I \subseteq \Delta^I \times \Delta^I)$，每个属性 A 解释成一个二元关系 $A^I(A^I \subseteq \Delta^I \times \Delta^L)$，每个个体 c 解释成一个元素 $c^I(c^I \in \Delta^I)$，每个值 l 解释成一个元素 $l^I(l^I \in \Delta^L)$。

对于定义 1 中的实体由三部分组成：实体名、实体属性以及实体间的语义关系，如图 1 所示。实体名由知网描述；实体的属性细化为功能、部分以及其他属性；实体的语义关系包含两部分，即层次语义关系和非层次语义关系。其中最常见的层次语义关系是上下位关系和部分整体关系。这类关系是有向传递的非对称关系，而定义的非层次关系不具有传递性。因此计算实体间相似度需对这两类关系分别讨论。

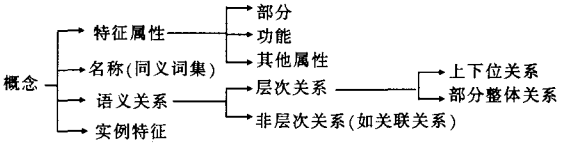

图 1　基于本体元数据模型的描述组成

定义 3　相似度。两个实体元素具有某些共同特征时，则定义它们是相似的。相似程度用相似度来表示，形式上相似度计算需满足：

a) $\mathrm{sim}(x,y) \in [0 \cdots 1]$，表示相似度的计算值为 [0,1] 中的一个实数。

b) 当且仅当 $x=y$，$\mathrm{sim}(x,y)=1$。

c) 如果两个对象没有任何公共特征，则 $\mathrm{sim}(x,y)=0$。

d) $\mathrm{sim}(x,y)=\mathrm{sim}(y,x)$，相似关系是对称的。

由语义相似度定义和常识知识有以下假设[7]：

假设 1　对于实体 C_1 和 C_2，如 $\mathrm{id}(C_1)=\mathrm{id}(C_2)$，则 C_1 和 C_2 为同一实体。

假设 2　如果两个实体具有相同的超实体，则这两个实体可能是相似的。

假设 3　如果两个实体具有相同的子实体，则这两个实体可能是相似的。

假设 4　如果两个实体具有相同的兄弟，则这两个实体可能是相似的。

假设 5　如果两个实体具有相同的实例，则这两个实体可能是相同的。

假设 6　如果两个实体具有相同的属性，则这两个实体可能是相同的。

同时基于 OWL 实体类等价关系描述，有如下假设：

假设 7　如果两个实体显式等价，则两个实体相似度为 1。

以上假设是进行语义相似度计算的基础。

4　MD4 模型

MD3 的不足在于其语义邻居只考虑了层次语义关系没有考虑语义关系中非层次关系的影响，同时也未考虑对象实例对于相似度的影响。本文在 MD3 模型的基础上，参考了其实体名称相似度，并扩展了实体间非层次关系和实例对实体语义相似度的影响，把 MD3 模型扩展到 MD4（fourfold matching-distance model）模型。

4.1　实体名称相似度

本文元数据模型中所有实体名称均是在领域语义词典中定义的，因此实体名称相似度计算可以采用基于知网的概念距离。知网[9]中概念的语义用义原来描述。义原是描述概念语义的最小单位，一共采用了 1 500 多个义原，这些义原的作用并不是等同的，而存在复杂的关系。为了简单起见，这里仅考虑义原的上下位关系，根据这种关系，所有义原构成了一个树状的层次体系。假如两个义原在该层次体系中的路径为 d，可得到两个义原之间的语义相似度，公式如下所示：

$$\mathrm{sim}(p_1,p_2)=a/(d+a) \tag{4}$$

式中，a 是一个可调节的参数。由于概念的语义是由多个义原来表示的，可以计算每个义原的相似度来考虑其重要性，就可以得到实体之间的名称相似度，计算方法如下：

$$S_{\mathrm{name}}(c_1,c_2)=\sum_{i=1}^{m} w_i \max_{1\leqslant j\leqslant n} \mathrm{sim}(p_i,p_j) \tag{5}$$

其中：m、n 为实体；c_1、c_2 为义原数；w_i 为第 i 个义原所占权重。

4.2　实体属性的相似度

文献[1]中，特征属性被细化为组成部分、功能以及其他属性三类。这样做的优点是可以依据语言环境的不同，根据属性对实体语义相似度影响的大小不同分配不同的权值。把三部分属性相似度的加权和作为实体特征属性的相似度，公式为

$$S_{\mathrm{attr}}(a^p,b^q)=w_1S_1(a^p,b^q)+w_2S_2(a^p,b^q)+w_3S_3(a^p,b^q) \tag{6}$$

其中：S_1、S_2、S_3 分别是组成部分、功能和其他属性的相似度，其相应的权值为 w_i，并且满足 $w_i \geqslant 0$，$\sum_{i=1}^{3} w_i=1$。

一般来说，元数据定义时属性有属性名称、属性描述及属性数据类型。为了计算属性相似度，两个属性的所有信息都应该进行比较。计算属性名称相似度与计算实体名称相似度方法一致。属性的数据类型或者是一个简单的数据类型或者是一个实体类型（对象数据类型），如果两个数据的数据类型都是简单的数据类型，简单数据类型之间的相似度[10]如表 1 所示。对象数据类型和简单数据类型是不相容的，数据类型之间的相似度为 0。若比较两个对象数据类型，则调用实体名称相似度进行计算。

表 1　简单数据类型的相似度计算

数据类型	integer	long	float	decimal	string
integer	1	0.9	0.8	0.7	0.3
long	0.9	1	0.8	0.7	0.3
float	0.8	0.8	1	0.7	0.3
decimal	0.7	0.7	0.7	1	0.3
string	0.3	0.3	0.3	0.3	1

4.3　语义关系的相似度

语义关系包括层次语义关系和非层次语义关系。层次关系为有向传递，非层次语义关系不具有传递性（如关联关系）。

1）层次语义关系的计算

计算层次语义关系借鉴文献[1]中语义邻居的定义，以实体为中心向周围辐射，设定一个语义半径，半径取值的大小反映与实体之间的亲疏关系。划定语义邻居的范围集合进行匹配，取集合中的最大值作为语义邻居之间的相似度。语义邻居计算公式如下：

$$N(a^o,r)=|c_i^o| \qquad \forall i, d(a^o,c_i^o) \leqslant r \tag{7}$$

则层次语义关系相似度计算：

$$S_h(a,b)=|A \cap B| / |A \cup B| \tag{8}$$

其中：A、B 分别代表实体 a、b 的语义邻居集合；$|\cdot|$ 表示集合

的势。

2)非层次语义关系的计算

定义实体的上位词为实体所有父类的集合,公式如下:

$$UC(C_i,H)=\{C_j\in C|H(C_i,C_j)\} \tag{9}$$

基于实体上位词的定义,定义实体的匹配公式如下:

$$CM(C_1,O_1;C_2,O_2)=|UC(C_1,H_1)\cap UC(C_2,H_2)|/|UC(C_1,H_1)\cup UC(C_2,H_2)| \tag{10}$$

如果关系的定义域或值域是实体 c,则称这些关系为与实体 c 相关的非层次关系,公式如下:

$$R_c(P)=\{\mathrm{dom}(R_x)=c\cup\mathrm{range}(R_x)=c|R_x\in P,c\in C\} \tag{11}$$

还可进一步把非层次关系细化为实体的 in 关系和 out 关系(可以认为非层次关系的方向是从定义域到值域,以此来定义 in 和 out 关系)。in 关系是指实体 c 是非层次关系的值域,定义如下:

$$R_{c\_I}=\{\mathrm{range}(R_x)=c|R_x\in P,c\in C\} \tag{12}$$

而 out 关系是指实体 c 是非层次关系的定义域,定义如下:

$$R_{c\_O}=\{\mathrm{dom}(R_x)=c|R_x\in P,c\in C\} \tag{13}$$

比较实体的非层次关系,首先应找出两个本体中与这两个实体相关的同类非层次关系(无须考虑不同类的非层次关系),进而比较这些同类非层次关系的另外一项之间的相似度(如果要比较的实体是非层次关系的定义域,分别找出这个关系的值域,通过实体匹配公式对其进行比较,反之亦然)。

下面以 in 关系为例描述比较的过程(out 关系类似):

$$R_I=R^p_{a\_I}\cap R^q_{b\_I}$$

其中:$R^p_{a_I}$表示本体 p 中与实体 a 相关的 in 关系;$R^q_{b_I}$表示本体 q 中与实体 b 相关的 in 关系,所以其交集 R_I 表示本体 p、q 中与实体 a、b 相关的公共 in 关系集合。如果实体 a、b 没有公共的 in 关系,则 R_I 为空,无须下面的计算。对于公共 in 关系集合,公式如下:

$$S_{\text{non-h-I}}(a,b)=\sum_{i=1}^{|R_I|}CM(\mathrm{dom}(R_{Ii}),p;\mathrm{dom}(R_{I_i}),q)/|R_I| \tag{14}$$

对 in 关系和 out 关系进行加权综合,得到非层次关系相似度的公式如下:

$$S_{\text{non-h}}(a,b)=iS_{\text{non-h-I}}(a,b)+oS_{\text{non-h-o}}(a,b) \tag{15}$$

其中:i、o 为权值,反映了不同类型关系对实体相似度的影响。对层次关系以及非层次关系计算结果进行综合,得到实体语义环境的相似度。定义如下:

$$S_{\text{neighborhoods}}(a,b)=w_1S_h(a,b)+w_2S_{\text{non-h}}(a,b) \tag{16}$$

其中:w_1、w_2 分别是层次关系和非层次关系的权重。因为层次关系是本体中最重要的关系,所以其比重理应较大,即 $w_1>0.5>w_2$,且 $w_1+w_2=1$。

4.4 实例特征相似度

实体 a、b 的实例特征相似度是通过资源元数据描述中具体实例和实例特征来评估。实体 a、b 实例特征相似度计算函数为

$$S_{\text{inst}}(a,b)=P(a\cap b)/P(a\cup b)=P(a,b)/[P(a,b)+P(a,\bar{b})+P(\bar{a},b)] \tag{17}$$

基于实例特征计算实体相似度牵扯到三个概率:$P(a,b)$、$P(a,\bar{b})$、$P(\bar{a},b)$。其中,$P(a,b)$是从一个本体的实例空间中随机选取一个实例属于实体 a,并且同时属于实体 b 的实例在实例空间中所占的比重。

4.5 综合公式

基于上面的分析,最后综合四部分的相似度值[10]得到跨本体实体间语义相似度的综合公式:

$$\mathrm{sim}(a,b)=wS_{\text{name}}(a,b)+uS_{\text{attr}}(a,b)+vS_{\text{neighborhoods}}(a,b)+rS_{\text{inst}}(a,b) \tag{18}$$

其中:w、u、v、r 分别表示各部分所占的权重,且 $w+u+v+r=1$,权重没有统一的规定,根据实际应用问题的不同而不同。最简单的方法是 $w=u=v=r=0.25$。当某一部分相似度计算为 0 时,可以扩大其他权重值,确定合适的权重也有很多种方法。

5 初步试验验证

5.1 实验评价准则

为方便与 MD3 算法进行对照,参考文献[1],本文采用信息检索领域查全率和查准率作为评价相似度计算的主要准则,并重新定义如下:

$$\text{recall}=|A\cap B|/|A| \tag{19}$$

$$\text{precision}=|A\cap B|/|B| \tag{20}$$

其中:A 表示实际相似的实体;B 表示通过模型计算所得的相似实体;|·|表示集合的势。

5.2 实验结果及分析

文献[1]中把 WordNet 本体和 SDTS 本体进行综合得到 WS 本体,并以这三个本体为例进行实验。笔者对 WS 本体利用定义 1 进行重新扩展定义,加入非层次关系以及实例的描述,使其满足本文中基于本体元数据模型的描述。本文以 WS 本体和 EWS 本体为例,分别利用 MD3 模型和 MD4 模型进行计算。

针对 WS 本体和 EWS 本体集,各部分采用相同的权重,计算结果如表 2 所示。

表 2 计算结果

模型	名称/%	属性/%	关系/%	实例/%	recall/%	precision/%
MD3	34	33	33	0	44	97
MD4	25	25	25	25	68	98

从上面的结果可以知道,MD4 模型对实体的各个维度进行了综合的考虑,无论是在查全率还是在查准率上,MD4 模型比 MD3 模型都有所提高。必须指出的是,虽然 MD4 模型在计算相似度时有一定的优势,但是,对实体全面的描述只是一个理想,只能在一定程度上的小范围内才能实现。针对具体的应用,MD4 模型需要进行相应的改动。

6 结束语

语义相似度计算方法的评估通常是把计算结果与人的主观判断结果进行比较,两者越接近,说明语义相似度方法就越好。由于知识背景和认知经验不同,判断结果主观性很强,即使取一群人判断结果的平均值,也只能相对客观地反映这一群人的认知特性,不能客观反映实体之间的语义相似性。因此如何客观地评估语义相似度计算方法还有待于深入研究。

本文提出的 MD4 模型继承了 MD3 模型的优点,对 MD3 模型进行了完善。在描述丰富的本体中,通过选择合适的权值,在 MD4 模型理论上可以确保语义相似度的计算更全面、准确。当然,在语义相似度计算过程中存在大量权值的设定,对性能存在一定的影响。如何方便快捷地选择合适的权值是未来工作的重点。

(下转第 2383 页)

```
Entity of Action
end action-name1
……
action action-name#n TYPE type-name
Entity of Action
end action-name#n
end IF
end active-link-name
```

循环事务活动链接的描述如下:

```
begin active-link-name
while condition-expression; control-expression
do
action action-name1 TYPE type-name
Entity of Action
end action-name1
……
action action-name#n TYPE type-name
Entity of Action
end action-name#n
end while
end active-link-name
```

为了操作的可变性,中间脚本中含有变量。系统运行时,利用客户程序传递过来的数据完成对 XML 脚本解释,实现对不同业务逻辑的操作。

5　结束语

针对传统软件工程设计思想的弱点,通过引入基于 XML 的中间脚本,提出了一种新的设计事务处理软件的方法,解决了程序员编写事务处理软件程序的烦恼,减少了传统软件工程中的编码和测试,并且对于用户需求只需要考虑功能需求,而对于用户的性能、可靠性和安全等需求统一在函数中做成最优。在软件设计阶段,无须对程序结构等进行分析,而是一个根据用户需求定义基本单元与相关约束的过程;在维护阶段,当用户的需求变化后,可以利用此模型快速升级,真正做到了设计事务处理软件时可见即可得,使用事务处理软件时所得即所需,解决了传统软件工程无法解决的问题。最后通过建立 ICETIP 开发平台,说明了此方法的可行性和有效性。因此,利用该模型设计事务处理软件只需两个步骤:需求分析和系统部署。

参考文献:

[1] WOOLDRIDGE, JENNINGS R. Pitfalls of agent-oriented development[C]//Proc of the 2th International Conference on Autonomous Agents. New York: ACM Press, 1998:385-391.

[2] PERRY D E, WOLF A L. Foundations for the study of software architecture[J]. ACM SIGSOFT Software Engineering Notes, 1992, 17(4):40-52.

[3] GARLAN, PERRY D E. Introduction to the special issue on software architecture[J]. IEEE Trans on Software Engineering, 1995, 21(4): 269-274.

[4] SHAW M, GARLAN D. Software architecture: perspectives on emerging discipline[M]. Englewood Cliffs: Prentice Hall, 1996.

[5] 余雪丽. 软件体系结构及实例分析[M]. 北京:科学出版社, 2004.

[6] 孙志勇. 多 agent 系统体系结构及建模方法研究[D]. 合肥:合肥工业大学,2004.

[7] FOSTER I, KESSELMAN C. The grid: blueprint for a future computing infrastructure[M]. Beijing: China Machine Press, 2005.

[8] 李程旭. 基于网构软件理论的交通综合平台研究[D]. 大连:大连理工大学,2005.

[9] HU Chun-ming, HUAI Jin-peng, SUN Hai-long. Web service-based grid architecture and its supporting environment[J]. Journal of Software, 2004,15 (7): 1064-1073.

[10] 黄双喜, 范玉顺. 一类通用的适应性软件体系结构风格研究[J]. 软件学报, 2006, 17 (6): 1338-1348.

[11] MEDVIDOVIC N, GRUNBACHER P, EGYED A, *et al*. Bridging models across the software lifecycle[J]. Journal of Systems and Software, 2003, 68(3): 199-215.

[12] PARNAS D L. Education for computing professionals[J]. IEEE Computer, 1990, 23(1): 17-22.

[13] BROOKS F P. No silver bullet-essence and accidents of software engineering[C]// Proc of Information Processing. Amsterdam: Elsevier Science Publishers, 1986: 1069-1076.

[14] COLE R, SCHLICHTING R. Editorial: configurable distributed systems[J]. IEE Proceedings Software, 1998, 145(5):129.

[15] PEYMAN O, NENAD M. Architecture-based runtime software evolution[C]// Proc of International Conference on Software Engineering. Kyoto: IEEE Computer Society, 1998:19-25.

[16] OREIZY P, TAYLOR R N. On the role of software architectures in runtime system reconfiguration[C]// Proc of the International Conference on Configurable Distributed Systems. Washington DC: IEEE Computer Society, 1998.

[17] 田边. 基于 agent 的软件体系结构与应用[D]. 西安:西北工业大学,2000.

[18] 田边, 戴航, 戴冠中. PDUIMS——基于持久存储的用户界面管理系统的设计与应用[J]. 计算机学报, 2000, 23(6):660-666.

[19] 贺岚, 狄玉来. 基于构件的软件设计模型[J]. 计算机研究与发展,1998,35(5):451-454.

(上接第 2338 页)

参考文献:

[1] RODRIGUEZ M A, EGENHOFER M J. Determining semantic similarity among entity classes from different ontologies[J]. IEEE Trans on Knowledge and Data Engineering, 2003,15(2):442-456.

[2] TVERSKY A. Features of similarity[J]. Psychological Review, 1977, 84(4):327-352.

[3] LEE J, KIM M, LEE Y. Information retrieval based on conceptual distance in is-a hierarchies[J]. Journal of Documentation, 1993, 49(2):188-207.

[4] RESNIK P. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity and natural language[J]. Journal of Artificial Intelligence Research, 1999,11: 95-130.

[5] RADA R, MILI H, BICKNELL E, *et al*. Development and application of a metric on semantic nets[J]. IEEE Trans on System, Man, and Cybernetics, 1989,19(1): 17-30.

[6] RODRIGUEZ M A. Assessing semantic similarity among spatial entity classes[D]. Orone: University of Maine,2000.

[7] 程勇,黄河,邱莉榕,等. 一个基于相似度计算的动态多维概念映射算法[J]. 小型微型计算机系统, 2006, 27(6):975-979.

[8] MAEDCHE A, EACHARIAS V. Clustering ontology-based metadata in the semantic Web[C]// Proc of the 6th European Conference on Principles of Data Mining and Knowledge Discovery. London: Springer-Verlag, 2002: 348-360.

[9] 刘群,李素建. 基于《知网》的词汇语义相似度计算[J]. 中文计算语言学, 2002,7(2):59-76.

[10] NGAN L D, HANG T M, GOH A E S. Semantic similarity between concepts from different OWL ontologies[C]// Proc of IEEE International Conference on Industrial Informatic. 2006.