不同本体中概念语义距离的一种度量方法*)

张德海[1,2] 朱 耀[2,3]
(云南大学软件学院 昆明 650091)[1]
(中国科学院计算技术研究所智能信息处理重点实验室 北京 100080)[2]
(中国科学院研究生院 北京 100080)[3]

摘 要 本体合并与映射中的一个重要步骤是检测不同本体中的相似概念,以提供合并或者映射的作用点。通常,具有相似名称的概念常被作为候选。然而,完全有可能出现具有文字相似名称的概念在语义上却不相似,甚至不相容。本文提出了一个从四方面比较概念语义距离的方法,综合考虑了一个概念在本体中的位置以及概念的属性这些对于确定概念涵义非常重要的信息。

关键词 本体,本体合并,本体映射,语义距离,概念比较

Measuring Semantic Distance between Concepts in Ontologies

ZHANG De-hai[1,2] ZHU Yao[2,3]
(School of Software, Yunnan University, Kunming 650091, China)[1]
(Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China)[2]
(Graduate University of the Chinese Academy of Sciences, Beijing 100080, China)[3]

Abstract An important task for ontology merging and mapping is to detect similar concepts in different ontologies as likely merging or mapping points. Usually, concepts with similar names are considered to be promising candidates. However, it is quite probable that concepts with lexically similar names are not semantically similar or compatible. In this paper, we presented a method to compare the semantic distance between concepts from four perspectives, capturing the meaning of a concept from its position in the containing ontology as well as its defined attributes.

Keywords Ontology, Ontology merging, Ontology mapping, Semantic distance, Concept comparison

1 引言

本体是一种有效的知识表示、信息共享和系统建模的工具,广泛地使用在基于知识的系统、语义 Web、自然语言处理等应用中。

本体在信息系统中的广泛使用导致很多具体的本体被信息技术的实践者们构建出来。有时我们需要连接或者集成不同来源的本体。比如我们可能要将一个领域本体和一个顶层本体(如 SUMO)映射(Ontology Mapping),或是需要合并不同设计者实现的应用本体。根据 Noy 和 Musen 的定义[6],在本体映射中,两个源本体仍然存在,只是在本体的概念间建立映射;在本体合并(Ontology Merging)中,将创建作为源本体融合版本的单个本体。

本体合并、本体映射中的一个重要步骤是检测两个本体中的相似概念,以便为合并或者映射提供可能的作用点。这个问题的一个直接的解决方法是寻找具有相似文字名称的概念。直观想法是具有相似名称的概念语义上也是相似的。然而,即便在两个本体中具有相同名称的概念也可能是语义迥异的。比如,同是名为"联合国秘书长"的两个概念,一个出现在 2005 年构建的本体中,另一个出现在 2007 年构建的本体中。尽管两个概念有相同的名称,却是指称不同的个体。在这样的情况下,仅仅考虑概念的名称就变得不可靠了。为了证实具有相似名称的概念在语义上也是相似、相容的,我们需要深入比较概念的语义。

过去,数据库与信息系统的研究者已经做了一些实质上是概念比较的工作,尽管他们可能不使用概念这个术语,而使用类型、类或者对象模式。在类型映射方面,Lehmann 和 Cohn 要求类型有对应的典型实例,并且他们同时使用类型之间、典型实例之间的交集来决定类型映射的可靠性[4]。Kashyap 和 Sheth 采用了基于上下文的方法来估计数据库对象的语义和模式相似性[3]。Weinstein and Birmingham 使用他们称作滤子测度、匹配测度、概率测度的 3 种测度来计算不同本体中的描述相容性[9]。

本文中我们不直接计算概念的语义相似性,而是计算概念的语义距离,由语义距离间接得出概念相似性。我们提出从 4 个方面计算概念语义距离:父概念集、子概念集、概念内涵、概念外延性。与以前的工作不同之处在于,我们认为一个特殊本体中的一个概念的意义同时取决于概念在本体中的位置和概念所具有的属性。前者反映了概念与宿主本体中其它概念间的关系,尤其是上下位关系,因此考虑了宿主本体的结构约束;后者反映了概念的内涵。关于内涵比较,我们将详细讨论针对几种常见的属性值类型,诸如布尔型、枚举型、实数、

*)教育部博士点基金资助项目(20050673001),云南省自然科学基金资助项目(2003F0005Q)。张德海 博士研究生,主要研究方向为人工智能、知识工程;朱 耀 硕士研究生,研究方向为人工智能、机器学习、理论计算机科学。

向量、字符串、区间和概率分布函数，来分别计算属性值差异。

在第2节，我们定义本体模型并给出一个具体的本体例子。第3给出概念比较方法的详细描述。最后是结论。

2 本体模型

Gruber定义本体为“用于帮助程序和人类共享知识的概念化规约”[2]。概念化是指用实体来陈述世界知识。

一个本体有4个组成部分：概念、槽、侧面和实例。

- 概念表示在一个领域里一族或一类具有相似属性的实体。比如，鸟类就是生物领域里的一个概念。概念被组织成子概念-父概念层次体系，并且按照subset-of关系实现单继承。每个概念都有一组槽。父概念的槽可以被子概念继承。
- 槽用来描述一个概念的属性或关系。槽值可以是单值，也可以是多值。每一个槽可以被槽框架中的一组侧面描述，比如值类型、定义域、单位等等。
- 侧面用于描述槽的属性或关系，比如值类型是约束一个槽的取值类型的侧面。
- 实例是概念的个体成员。*instance -of* 关系将一个实例和其所属的概念连接起来。比如汤姆是人类这个概念的一个实例。并且我们使用汤姆 *instance -of* 人类来表示这样一个事实。

本体的形式化定义如下：

定义1 一个本体 O 是一族概念框架、槽框架和实例框架，表示为五元组(C,S,E,R,H)。C 是概念的有限集，包括根概念ROOT，是本体中所有其它概念的父概念；S 是槽的有限集；E 是实例的有限集；R 是关系的有限集，但不包括上下位关系；H 是一族直接上下位关系，称为分类层次，并且 H 包括 *subset -of* 和 *instance -of* 关系。

下面是一个本体的例子。图1显示了一个名为 O_1 的本体的概念层次。

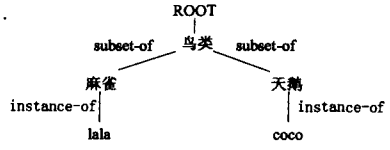

图1 本体 O_1 的概念层次

概念 ::鸟类 *subset -of* ROOT
{
[会飞 :＝ 是],
[翅膀数 :＝ 2],
[体温类别 :＝ 恒温],
[喜食 :＝ 昆虫],
[体长 :＝ [15,228] cm],
[体重:＝ [4,6000] g]
}
概念 :: 麻雀 subset-of 鸟类
{
[生殖方式 :＝ 卵生],
[孕期:＝ [11,12] d],
[体长 :＝ [10,15] cm],
[体重:＝ [20,100] g]
}
概念 :: 天鹅 subset-of 鸟类
{
[生殖类型 :＝ 卵生],
[孵化期:＝ [28,30] d],
[体长 :＝ [30,170] cm],
[体重:＝ [3000,23000]g]
}
实例 :: lala instance-of 麻雀
{
[喜食 :＝ 种子],
[体长 :＝ [14,14] cm],
[体重:＝ [50,50]g]
}
实例 :: coco instance-of 天鹅
{
[喜食 :＝ 蓓蕾],
[体长 :＝ [63,63]cm],
[体重:＝ [2900,2900] g]
}
槽 :: 会飞
{
[值类型:＝ 布尔型],
[定义域 :＝ 鸟类]
}
槽 :: 孕期
{
[值类型:＝ 自然数区间],
[定义域 :＝ 鸟类],
[单位 :＝ d]
}
槽 :: 喜食
{
[值类型:＝ 字符串],
[定义域 :＝ 鸟类],
[最大取值数目 :＝ 4]
}

在这个说明性的本体中，我们定义了三个概念鸟类、麻雀和天鹅；两个实例lala(一个麻雀的实例)和coco(一个天鹅的实例)；还有一些槽，比如会飞、孕期、喜食。通过subset-of和instance-of关系，这些概念和实例被组织成树结构。

3 本体概念比较

虽然直观上具有相似文字名称的概念倾向于语义相似，但实际中即便在两个本体里具有相同名称的概念也可以有不相容的语义，这通常是由于名称的模糊性和随意性造成的。为便于说明问题，我们给出在另一个本体 O_2 中的概念鸟类的定义如下，本体 O_2 的概念层次如图2所示。

概念 :: 鸟类 subset-of 动物
{
[有羽毛 :＝ 是],
[翅膀数 :＝ 2],
[体温类别 :＝ 恒温],
[喜食 :＝ 种子],
[体长 :＝ [5,280]cm],
[体重:＝ [2,200000] g]

}

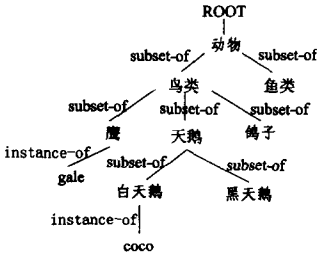

图 2　本体 O_2 的概念层次

这里，我们会发现 O_1 中的概念鸟类(用鸟类$_1$ 表示)和 O_2 中的概念鸟类(用鸟类$_2$ 表示)在 4 个方面有区别：

- 父概念集
- 子概念集
- 内涵
- 实例集

O_1 中概念鸟类$_1$ 是 ROOT 的直接子概念，并且有两个子概念麻雀、天鹅和两个实例 lala，coco。O_2 中概念鸟类$_2$ 是动物的直接子概念，并且有三个直接子概念鹰、天鹅、鸽子，两个间接子概念白天鹅、黑天鹅和两个实例 gale，coco。

一个概念的内涵是其属性集合。概念鸟类在本体中 O_1 有属性会飞，但是在本体中 O_2 没有。反之，它在本体中 O_2 有属性有羽毛，但是在本体中 O_1 没有。

在我们的基于本体的知识表示模型中，一个概念被定义为一个有一族槽的框架，这些槽描述了该概念的属性。相应地，实例框架也定义了描述实例属性的一族槽。

我们可以通过概念间的 subset-of 关系得到一个概念的所有父概念和子概念，通过概念实例间的 instance-of 关系得到所有的实例。

下面我们给出一个度量两个本体中的两个概念的语义距离的方法。将从 4 个方面考虑：父概念集、子概念集、内涵和可用的实例集。下面用 c 和 c'代表所考虑的分别在本体 O_1 和 O_2 中的两个概念。我们将 c'称为 c 的对应物。

3.1　概念的父概念集比较

我们认为一个概念的父概念比其子概念重要，因为概念从父概念继承属性。基于这个原因，在比较一个概念和它的对应物时，我们分别比较父概念集和子概念集。

定义 2　令 $Sup(c)$ 是概念 c 的父概念集，并且定义 $Sup(c)$ 如下：

$$Sup(c) = \{a \mid a \in C, c \ subset\text{-}of \ a\}$$

这里 C 是 c 所属的本体的概念集合。

用 $D_{Super}(c,c')$代表 c 和 c'的父概念集差异，定义 $D_{Super}(c,c')$如下：

$$D_{Super}(c,c') = 1-\frac{|Sup(c)\cap Sup(c')|}{|Sup(c)\cup Sup(c')|} \quad (1)$$

这里 c 在 O_1 中，c' 是 c 在 O_2 中的对应物。

例 1　概念鸟类在例子本体 O_1 和 O_2 中的父概念集差异是

$$D_{Super}(\text{鸟类}_1,\text{鸟类}_2) = 1-\frac{|\{ROOT\}|}{|\{ROOT,\text{动物}\}|}=\frac{1}{2}$$

3.2　概念的子概念集比较

类似于概念的父概念集比较，我们可以定义子概念集比较。

定义 3　令 $Sub(c)$是概念 c 的子概念集，并且定义 $Sub(c)$如下：

$$Sub(c) = \{a \mid a \in C, a \ subset\text{-}of \ c\}$$

用 $D_{Sub}(c,c')$代表 c 和 c'的父概念集差异，定义 $D_{Sub}(c,c')$如下：

$$D_{Sub}(c,c') = 1-\frac{|Sub(c)\cap Sub(c')|}{|Sub(c)\cup Sub(c')|} \quad (2)$$

例 2　O_1 中概念鸟类的子概念集是{麻雀，天鹅}，O_2 中概念鸟类的子概念集是{鹰，天鹅，鸽子，白天鹅，黑天鹅}，所以概念鸟类在 O_1 and O_2中的子概念集差异是

$$D_{Sub}(\text{鸟类}_1,\text{鸟类}_2) = 1-\frac{1}{6}=\frac{5}{6}$$

3.3　概念的内涵比较

一个概念的内涵是其属性集合。在我们的基于本体的知识表示模型中，一个属性被表示成槽值约束断言。这样概念的内涵差异就可以表述成它们的槽值约束断言间的差异。

我们如下定义概念的内涵。

定义 4　概念 c 的内涵是 c 的框架中的一族槽值约束断言，用 $I(c)$代表，形式上，$I(c) = \{s_i := v_i \mid s_i \in S^c\}$，其中 s_i 是一个槽，S^c 是描述概念 c 的槽集。

比较鸟类$_1$ 和鸟类$_2$ 的框架表示，我们可以列出两点不同：

(1)有些槽在鸟类$_1$ 和鸟类$_2$ 中取值不同，比如喜食、体长、体重。

(2)有些槽仅在鸟类$_1$ 中出现，而另一些则仅在鸟类$_2$ 中出现，比如会飞、有羽毛。

基于以上分析，我们定义内涵差异 $D_{Intention}$ 如下：

$$D_{Intention}(c,c')=\alpha\left(1-\frac{|S_1\cap S_2|}{|S_1\cup S_2|}\right)+\beta\frac{\sum_{s_i\in S_1\cap S_2} d_s(v_i,v_i')}{|S_1\cup S_2|} \quad (3)$$

其中 S_1 是 c 的框架的槽集合，S_2 是 c'的框架的槽集合。$d_s(v_i,v_i')$是槽值 v_i和 v_i'的差，v_i和 v_i'是槽 s_i 在两个框架中的取值，并且 $s_i \in S_1 \cap S_2$。

公式(3)的前半部分是 c 和 c'的不同槽的数目比，体现了 c 和 c'的槽集合的差异。公式的后半部分体现了 c 和 c'的槽值的平均差异。参数 α 和 β 分别是两部分的权重。这里我们没有考虑槽的权重，在实际中可以给每个槽赋予一个权重，因为不同的槽具有不同的概念辨识能力，比如有羽毛是一个将鸟类区别于其它概念的重要属性。

可以看出 $D_{Intention}$ 具有以下几个性质。首先，$D_{Intention}$ 是对称的。其次，当 S_1 和 S_2 不交时 $D_{Intention}$ 等于 1。再次，S_1 和 S_2 重叠越多，$D_{Intention}$ 越取决于槽值的差异。

由于槽值类型的多变，怎样计算 $D_{Intention}$ 中的 $d_s(v_i,v'_i)$就成为一个重要的问题。槽值类型可以是布尔型、枚举型、实数、向量、字符串、区间和概率分布函数等等。一种可行的解决方法是对于不同的槽值类型，采用不同的计算方法。有时，需要领域知识来针对某种特别的槽值类型决定有意义的计算方法。

为说明问题，我们给出几种具有代表性的槽值类型的 $d_s(v_i,v_i')$的计算方法。

(1)槽值类型是布尔型：

$$d_s(x,y)=\begin{cases}1, & x\neq y\\ 0, & x=y\end{cases} \quad (4)$$

(2)槽值类型是枚举型：

$$d_s(x,y)=\begin{cases}1, & x\neq y\\ 0, & x=y\end{cases} \quad (5)$$

可见布尔型实际是一种特殊的枚举型。

(3)槽值类型是实数：

$$d_s(x,y)=\frac{|x-y|}{\mathrm{Max}(|x|,|y|)} \tag{6}$$

例如，$d_s(2,8)=\frac{|2-8|}{8}=\frac{3}{4}$

(4)槽值类型是向量：

假设两个 n-维的向量 $x=(x_1,\cdots,x_n)$ 与 $y=(y_1,\cdots y_n)$。下面是4种常用的向量距离计算方法[8]：

欧氏距离：

$$d_s(x,y)=\sqrt{\sum_{i=1}^{n}w_i(x_i-y_i)^2} \tag{7}$$

曼哈顿距离：

$$d_s(x,y)=\sum_{i=1}^{n}w_i|x_i-y_i| \tag{8}$$

曼格萨里恩距离：

$$d_s(x,y)=\max_{1\leqslant i\leqslant n}w_i|x_i-y_i| \tag{9}$$

闵可斯基距离：

$$d_s(x,y)=\sqrt[p]{\sum_{i=1}^{n}w_i(x_i-y_i)^p} \tag{10}$$

上面公式中的 w_i 是权重参数。

因为实数可以看成1-维向量，所以上面的距离也适用于实数类型。

(5)槽值类型是字符串：

Levenshtein 建立了一种比较字符串差异的方法，称作编辑距离[5]。两个字符串之间的编辑距离定义为将一个字符串变换为另一个字符串所需的最少的字符的插入、删除、替换操作。Levenshtein 同时给出了计算编辑距离的动态规划算法。例如，两个串“Tele_com”和“Telecom”之间的编辑距离为 ed(“Telecom”,“Tele_com”) = 1。利用编辑距离我们可以定义槽值类型为字符串时的 $d_s(x,y)$ 为

$$d_s(x,y)=\frac{ed(x,y)}{\mathrm{Max}(strlen(x),strlen(y))} \tag{11}$$

其中 $strlen()$ 是求字符串长度的函数。

这样，d_s(“Telecom”,“Tele_com”) $=\frac{1}{8}$。

(6)槽值类型为区间：

槽的取值可能是一个区间，比如体重：= [2,200000]g，所以有必要比较两个区间之间的差异。假设有两个区间 $g_1=[x_1,y_1]$，$g_2=[x_2,y_2]$，则 $d_s(g_1,g_2)$ 可以定义为

$$d_s(g_1,g_2)=1-\frac{\|g_1\cap g_2\|}{\|g_1\cup g_2\|} \tag{12}$$

其中 $\|h\|$ 是区间的取模函数，定义为：

$$\|h\|=\begin{cases}0, & \text{if } h=\phi,\\ \delta, & \text{if } h=[x,x],\\ |y-x|, & \text{if } h=[x,y],x\neq y\end{cases} \tag{13}$$

其中 δ 是[0,1]中的一个较小的实数，这个公式是刘惟一和田雯所著《数据模型》一书中所定义的一个变形[7]。

例3 若 $g_1=[4,6000]$，$g_2=[2,200000]$，则

$$d_s(g_1,g_2)=1-\frac{\|[4,6000]\|}{\|[2,200000]\|}=1-\frac{5996}{199998}\approx 0.7$$

例4 计算 $D_{Intention}$(鸟类$_1$，鸟类$_2$)，其中 $\alpha=\beta=0.5$。

$$D_{Intention}(\text{鸟类}_1,\text{鸟类}_2)=0.5(1-\frac{5}{7})+0.5(\frac{0+0+1+0.23+0.7}{7})\approx 0.28$$

(7) 槽值类型是概率分布函数：

Kullback-Leibler 距离常用来度量两个概率分布之间的距离[1]。

对于两个分布列 $p(x)$ 和 $q(x)$，K-L 距离定义为

$$d_s(p,q)=\sum_x p(x)\log\frac{p(x)}{q(x)} \tag{14}$$

对于两个概率密度函数，K-L 距离定义为

$$d_s(f,g)=\int f(x)\log\frac{f(x)}{g(x)}dx \tag{15}$$

3.4 概念的外延性比较

一个概念的外延是所有的实例形成的集合。但是由于本体不一定是完备的，因此可能不能枚举出本体中一个概念所有的实例，我们只能得到那些存在于概念的宿主本体中的实例。因此，我们这里进行的是概念的外延性比较，而不是概念的外延比较。

令 $E(c)$ 是概念 c 的在同一个本体中的实例集，形式上，$E(c)=\{e\mid e\ instance\text{-}of\ c\}$。这里 $E(c)$ 包括 c 的直接实例和所有 c 的子概念的实例。例如，在 O_1 中，概念鸟类$_1$ 有两个实例 lala 和 coco，而它的对应物鸟类$_2$ 在 O_2 中有两个实例 gale 和 coco。

我们使用 $E(c)$ 和 $E(c')$ 的对称差来定义外延性差异，用 D_{Exten} 来表示。

$$D_{Exten}(c,c')=\frac{|E(c)\cup E(c')-E(c)\cap E(c')|}{|E(c)\cup E(c')|}=1-\frac{|E(c)\cap E(c')|}{|E(c)\cup E(c')|} \tag{16}$$

例5 计算鸟类在 O_1 和 O_2 中的 D_{Exten}。

$$D_{Exten}(\text{鸟类}_1,\text{鸟类}_2)=1-\frac{|\{coco\}|}{|\{lala,gale,coco\}|}=1-\frac{1}{3}=\frac{2}{3}$$

3.5 概念语义距离

我们已经讨论了概念差异的4个方面。现在可以将这4个差异方面集成为一个语义距离，用 $Dist(c,c')$ 来表示概念 c 和 c' 之间的语义距离。

$$Dist(c,c')=\theta_1 D_{Super}+\theta_2 D_{Sub}+\theta_3 D_{Intention}+\theta_4 D_{Exten}$$

其中 $\theta_1,\theta_2,\theta_3,\theta_4$ 是权重参数。

例6 计算鸟类$_1$ 和鸟类$_2$ 的语义距离，其中 $\theta_1=0.3$，$\theta_2=0.1$，$\theta_3=0.5$，$\theta_4=0.1$。

$$Dist(\text{鸟类}_1,\text{鸟类}_2)=0.3\times\frac{1}{2}+0.1\times\frac{5}{6}+0.5\times 0.28+0.1\times\frac{2}{3}\approx 0.44$$

结束语 本文提出了一种本体中概念的比较方法。这种方法能够用来度量不同本体中具有相似名称的概念的语义距离，以决定这些概念是否应该在本体合并或本体映射的过程中被合并或被映射。这里，在比较两个概念在它们的宿主本体中的位置差异时，我们仅考虑了 subset-of，instance-of 这样的上下位关系；在比较内涵时我们仅考虑了具有相同值类型的同名槽。进一步的工作可以考虑概念间的其它关系和具有不同值类型的同名槽。另外，在我们的方法中有很多权重参数，这些参数可以通过由人工标注的本体对和相似度训练数据来估计出。通过进一步的实验我们可以检验这种方法在度量概念相似性方面的有效程度。

参 考 文 献

[1] Cover T M, Thomas J A. Elements of Information Theory. John

Wiley & Sons,1991

[2] Gruber T R. Towards Principles for the Design of Ontologies Used for Knowledge Sharing // Poli R, Guarino N, eds. International Workshop on Formal Ontology. Padova, Italy, 1993

[3] Kashyap V, Sheth A P. Semantic and schematic similarities between database objects: a context-based approach. VLDB Journal, 1996: 176-304

[4] Lehmann F, Cohn A G. The EGG/YOLK Reliability Hierarchy: Semantic Data Integration Using Sorts with Prototypes // CIKM'94. 1994: 272-279

[5] Levenshtein I V. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Cybernetics and Control Theory, 1996, 10(8): 707-710

[6] Noy N F, Musen M A. An Algorithm for Merging and Aligning Ontologies: Automation and Tool Support // 16th National Conference on Artificial Intelligence (AAAI-99), Workshop on Ontology Management. Orlando, FL, 1999

[7] 刘惟一, 田雯. 数据模型. 科学出版社, 2001

[8] 史忠植. 高级人工智能(第二版). 科学出版社, 2006

[9] Weinstein P, Birmingham W. Comparing concepts in differentiated ontologies // Proc. of AW-99. 1999

(上接第 115 页)

```
    Backup(Rec. adv_out);}
}
```

算法 3 接收 BACK 消息的处理程序

```
//R 从 I_rup 接收 BACK 消息
OnReceiving(BACK, I_rup)
{//向 I_up 发送 Rec. adv_out
  SendMsg(Advout, Adv_out, I_up);
}
```

算法 4 接收故障恢复消息的处理程序

```
OnReceiving(FROM_FAILURE)
{//修改主存路由表
 Restore(Main_routing_table);
 //修改主存 adv_out 表
 Restore (Main_ adv_out);
 //向 I_up 发送 Rec. adv_out
 SendMsg(Advout, Adv_out, I_up);
 //R 向所有 I_rdowni 发 BACK 消息
 For(i=1,n,i++) SendMsg(BACK, I_rdowni);
}
```

定理 1 基于崩溃/恢复模式的路由协议，满足可靠性条件 Ω_{adv_as}。证明从略。

崩溃/恢复模式依靠这几个机制来处理暂时故障：恢复数据库是稳定的，当路由器发生故障后，它能恢复到故障前的状态；使用 TCP 协议确保文档和订阅能可靠、有序地传输；Rec. adv_{out} 维护一组路由器及其上游路由器间的确认信息，在上游路由器发生故障期间，路由器上保存改变量，当上游路由器恢复后，该路由器能前滚到和当前订阅一致的状态；序列号被嵌入在所有的消息中。

结束语 本文系统地、形式化地研究了订阅/发布系统中的可靠性问题。基于轨迹序列和线性时态逻辑定义了订阅/发布系统和带通知的订阅/发布系统的可靠性条件。设计了崩溃/恢复模式的路由协议，该路由协议满足带通知的订阅/发布系统的可靠性条件要求，能有效地处理具有局部性、临时性的路由器故障和链路故障。本文对订阅/发布系统中可靠性条件的研究成果可用于 Internet 上大规模发布/订阅系统路由算法的可靠性分析。

参 考 文 献

[1] Costa P, Migliavacca M, Picco GP, et al. Epidemic algorithms for reliable content-based publish-subscribe: An evaluation // Proc. of the ICDCS 2004. Tokyo: IEEE Computer Society, 2004: 552-561

[2] Bhola S. Topology changes in a reliable publish/subscribe system. Technical Report, RC23354. Yorktown Heights: IBM Thomas J. Watson Research Center, 2004

[3] Chand R, Felber PA. A scalable protocol for content-based routing in overlay networks // Proc. of the 2nd IEEE Int'l Symp. on Network Computing and Applications. Cambridge: IEEE CS Press, 2003: 123-130

[4] Bhola S, Strom R E, Bagchi S, et al. Auerbach. Exactly-Once delivery in a content-based publish-subscribe system // Lala J, ed. Proc. of the Int'l Conf. on Dependable Systems and Networks (DSN 2002). Washington: IEEE Computer Society Press, 2002: 7-16

[5] Opyrchal L, Prakash A. Secure distribution of events in content-based publish subscribe systems // 10th USENIX Security Symposium. August 2001

[6] Carzaniga A, Wolf A L. Forwarding in a content-based network // Proceedings of ACM SIGCOMM 2003. Karlsruhe, Germany, 2003: 163-174

[7] Carzaniga A, Rosenblum D S, Wolf A L. Content-based addressing and routing: A general model and its application. Technical Report CU-CS-902-00. Department of Computer Science, University of Colorado, January 2000

[8] Cugola G, Nitto E D, Fugetta A. The JEDI event-based infrastructure and its application to the development of the opss wfms. IEEE Transactions on Software Engineering, 2001: 27 (9): 827-850

[9] Bricconi G, Tracanella E, Nitto E D, et al. Analyzing the behavior of event dispatching systems through simulation // HiPC '00: Proceedings of the 7th International Conference on High Performance Computing. London, UK, Springer Verlag, 2000: 131-140

[10] Mühl G. Large-Scale Content-Based Publish/ Subscribe Systems. PhD thesis. Darmstadt University of Technology, 2002

[11] Diao Y, Rizvi S, Franklin M J. Towards an Internet-Scale XML Dissemination Service // Proceedings of VLDB 2004. Toronto, Canada, August 2004

[12] Diao Y, Franklin M J. High-Performance XML Filtering: An Overview of YFilter. IEEE Data Engineering Bulletin, March 2003

[13] Pnueli A. The temporal semantics of concurrent programs. Theoretical Computer Science, 1981(13): 45-60