面向领域特征聚类的构件组装优化方法

马 华
MA Hua

湖南涉外经济学院 计算机学部,长沙 410205
Department of Computer,Hunan College of International Economics,Changsha 410205,China
E-mail:cnmahua@gmail.com

MA Hua.Optimization method on component composition for domain feature clustering.Computer Engineering and Applications,2009,45(21):197-200.

Abstract: The component composition of Internetware became difficult in the open,dynamic and uncertain platforms such as the Internet.An optimization method on component composition is proposed for domain feature clustering.In it,Ontology classification and similarity measure are introduced,and a clustering algorithm based on partition is designed which can achieve accurate component clustering on the basis of domain feature.Attribute values of QoS about component and link are normalized,and the algorithm based on dynamic programming is proposed to solve global optimization problem of component composition.The algorithmic analysis and experiment prove that this method is effective and feasible.

Key words: component composition;domain feature;clustering;dynamic programming;Peer-to-Peer(P2P)

摘 要: Internet 环境的开放、动态和难控等特点,使网构软件的构件组装问题变得十分复杂。提出了一种面向领域特征聚类的构件组装优化方法。通过引入本体分类和相似度比较方法,设计了一种基于划分的聚类算法,以实现基于领域特征的精确的构件聚类。通过对构件和链路的多维 QoS 指标的换算,给出了应用动态规划方法求解面向领域特征簇的构件组装全局最优解的算法实现。算法分析和实验仿真表明了该方法的有效性和可行性。

关键词: 构件组装;领域特征;聚类;动态规划;P2P
DOI: 10.3778/j.issn.1002-8331.2009.21.057 **文章编号:** 1002-8331(2009)21-0197-04 **文献标识码:** A **中图分类号:** TP311

1 引言

网构软件(Internetware)是实现 Internet 中各软件实体互连、互通、协作和联盟的类似 WWW 的软件 Web(Software Web)[1]。网构软件的开发过程是一个基于丰富基础构件资源平台的构件组装过程,即根据用户需求对基础构件资源进行选择、组合,从而生产出新的应用系统。

文献[2]提出了一种网构软件环境下基于非确定性推理的构件服务质量动态评估方法,可以为构件的选择提供决策支持。文献[3]基于 Ontology 和一阶谓词逻辑,设计了一个信任驱动的服务选取算法。文献[4]提出了一种支持自适应构件组装的网构软件构件库模型。以上研究均限于网构软件中单一构件选择的局部优化,而开放、动态、难控的 Internet 环境下网构软件的全局最优的构件组装问题,则相对要更加复杂[4]。

领域特征体现了特定系统具有的能力或特点,它是功能性的需求或对系统质量属性的要求[5]。基于语义本体描述构件的领域特征,可极大地提高构件组装选取目标构件时的查准率。因此,提出一种 P2P 计算环境下基于领域特征聚类实现构件组装的方法。该方法中,通过本体分类和相似度比较方法实现构件资源的领域特征聚类,并应用动态规划算法实现构件组装的全局最优。最后,通过算法分析和实验仿真,表明了该方法求解网构软件构件组装问题的有效性和可行性。

2 网构软件中的构件组装问题描述

网构软件的开发过程实质上是一个基于领域特征分析的构件组装过程,如图 1 所示。

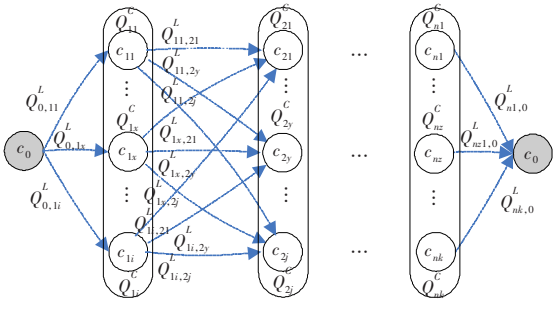

图 1 网构软件中的构件组装问题

基金项目: 湖南省教育厅资助科研项目(the Scientific Research Fund of Huran Provincial Education Department under Grant No.07C425);湖南省教育科学十一五规划课题(the Hunan Province Eleventh Five-Year Programs of Educational Science under Grant No.XJK08CXJ001)。
作者简介: 马华(1979-),男,高级工程师,主要研究领域为构件组装,服务计算和工作流。
收稿日期: 2009-05-05 **修回日期:** 2009-06-22

图 1 中,c_0、c_{11}~c_{nk}代表构件,其中,c_{11}~c_{1i}是满足领域特征 1 的构件集合,c_{21}~c_{2j}是满足领域特征 2 的构件集合,依次类推,c_{n1}~c_{nk}是满足领域特征 n 的构件集合。位于同一个构件集合中的构件在功能上没有区别,但提供的服务质量(QoS)存在差异。网构软件环境下,各个业务领域内基础构件的数量非常庞大,因此,如何依据领域特征进行精确的构件聚类,以创建图 1 中的构件集合,成为实现构件组装的必要条件。

图 1 中的 c_0 被称为初始装配点,即构件组装的起始点。如果 c_0 选择组装了 c_{11},则 c_{11} 成为新的装配点,继续进行下一步装配。$Q^L_{i,j}$ 表示 c_i、c_j 间链路(i,j)的 QoS 值,需要考虑端到端时延和可靠性等指标。Q^C_i 表示 c_i 的 QoS 值。显然,对由多个构件集合中不同构件的选择,组装后的网构软件将获得不同的 QoS 值。由此,网构软件构件组装问题的目标是保证由 n 个构件组装而成的网构软件的 QoS 值最优。

3　面向领域特征聚类的构件组装模型

3.1　总体结构

P2P(Peer-to-Peer)计算[6]以信息和服务的共享与管理、业务的整合与协作为研究内容,以构建完全自主、分布计算的互联系统为目标,这与网构软件的自主性、协同性、反应性、演化性和多目标性要求相一致,因而它可以为网构软件中的构件组装提供有力支持。基于网构软件的领域特征分析,为满足网构软件的构件组装需求,提出了一种 P2P 计算环境下基于领域特征聚类的构件组装模型,如图 2 所示。

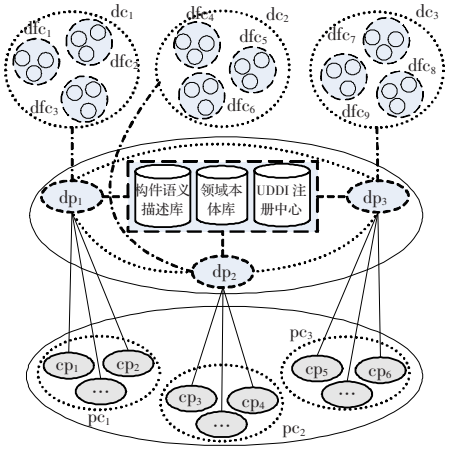

图 2　基于领域特征聚类的构件组装模型

图 2 中的 P2P 网络采用了由领域对等体(Domain Peer,DP)和构件对等体(Component Peer,CP)构成的两层混合式拓扑结构。DP 由可靠性高和性能好的节点组成,它们构成一个结构化的领域网络。每个 DP 管理一个业务领域,为下层提供构件的注册、发布和检索等服务。下层是由 CP 组成的 P2P 网络。CP 根据其向外提供服务的构件的业务领域,通过邻近的上层对等体,注册到相应的 DP 上。上、下层之间在逻辑上通过领域簇(Domain Cluster,DC)聚集在一起。所有的构件资源均以 Web 服务形式封装和发布,并被划分成不同的领域簇(DC)。在领域簇内,依据领域特征的不同,使用基于划分的聚类方法创建多个领域特征簇(Domain Feature Cluster,DFC)。

使用 DAML-S 作为本体建模语言,实现服务构件的语义化描述。Profile 文件建立了服务构件的相关参数与特定本体中的概念间关联的语义信息;Process 文件建立了服务构件中各个操作的基本信息与特定本体中的概念间关联的语义信息,Grounding 文件则建立了 WSDL 和 Process 描述的映射。这样,基于这些服务构件的语义描述文件和领域本体库(如 WordNet),建立构件本体。通过对这些构件本体进行分类和相似度比较可以实现构件资源的领域特征聚类,进而构件组装的实现成为可能。

3.2　概念定义

模型中的 6 个关键概念定义如下:

定义 1　构件对等体(Component Peer,CP)由 P2P 网络中的任意服务构件组成。单个构件对等体表示为 $cp=\{id,n,u,d,co,S,O,Q\}$。其中,id 为唯一标识,n 为描述名,u 为调用地址,d 为所属领域,co 为描述了 cp 中领域特征的构件本体,S 为输入输出参数集合,O 为操作集合,Q 代表综合考虑了构件多项因素的 QoS 加权值。

定义 2　领域对等体(Domain Peer,DP)每个领域对等体负责管理一个业务领域相关的所有构件资源。单个领域对等体表示为 $dp=\{id,d,pc,dc,cps\}$。id 是唯一标识,d 是 dp 管理的业务领域名,pc 表示所属自然簇的 id,dc 表示所属领域簇的 id,cps 是在 dp 注册的 cp 的集合。

定义 3　自然簇(Physical Cluster,PC)是物理上相邻的对等体组成的集群。每个自然簇由一个 dp 和多个 cp 组成。单个自然簇表示为 $pc=\{id,dp,cps\}$。id 是唯一标识,dp 是 pc 内主题对等体的 id,cps 是 pc 内 cp 的集合。

定义 4　领域簇(Domain Cluster,DC)是由隶属于同一领域的构件对等体组成的集群。每个领域簇由一个 dp 和多个 cp 组成。单个领域簇表示为 $dc=\{id,dp,cps,cos\}$。id 是唯一标识,dp 是 dc 内领域对等体的 id,cps 是 dc 内 cp 的集合,cos 是 dc 内 cps 对应的构件本体的集合。

定义 5　构件本体(Component Ontology,CO)指使用 DAML-S,并引用领域本体库描述构件领域特征的本体。单个构件本体描述为 $co=\{id,n,c,i,r,f,a,s\}$。其中,id 为唯一标识,n 为名字,c 为概念集,i 为概念实例集,r 为概念集合上的关系集合,f 为概念集合上的函数集合,a 为公理集合,s 是 co 与其所在的领域特征簇中簇代表的相似度值。

定义 6　领域特征簇(Domain Feature Cluster,DFC)是由 dc 中具有较高领域特征相似度的 cp 组成的集群,是构件本体的聚类。单个领域特征簇表示为 $dfc=\{id,c^*,s,cps,num\}$。其中,id 为唯一标识,c^* 为簇代表,s 为 dfc 内所有除 c^* 外的其他 cp 以 c^* 为参照对象时计算的构件本体的相似度值总和,cps 为隶属于 dfc 中的 cp 集合,num 为 dfc 中的 cp 的总数。这样,dfc 中本体相似度的均值为:$s'=dfc_i.s/(dfc_i.num-1)$。

4　基于领域特征的构件聚类算法

通过引入构件本体,采用基于划分的聚类方法实现面向领域特征的构件聚类,并借鉴了 A.Maedche 提出的本体相似度比较方法[7],设计 Similarity()函数用于计算构件本体的相似度。

4.1　算法描述

基于领域特征的构件聚类算法描述如下:

(1)从 DC 中选取一个领域簇 dc_m,其包含的构件本体数为

$n=|dc_m.cos|$，并从中选取 k 个本体作为初始的簇代表 $c_1^*\sim c_k^*$，它们构成了一个簇代表集 C^*，并由此构造了 k 个构件本体簇 $dfc_1\sim dfc_k$；

(2)for（int i=1;i<=n;i++ ）//遍历整个领域簇

{

//将 co_i 指派到与其有最大相似度的簇代表所在的簇 dfc_j

$co_i.s=\text{Max}\{Similarity(co_i,co_j^*)\},co_i\in O^*,co_j^*\in O^*$

$dfc_j.s+=co_i.s$

}

(3)for（int i=1;i<=k;i++）//遍历各个领域特征簇

{

从 dfc_i 中随机选择一个非簇代表 co_r 作为临时簇代表

$$s''=\sum_{k=1}^{n}\{Similarity(co_k,co_r)\},n=dfc_i.num \ \&\& \ k\neq r$$

if（$s''>dfc_i.s$）

$dfc_i.c^*=co_r$

}

(4)if（$dfc_1\sim dfc_k$ 的簇代表不再变化）

聚类过程结束，dc_m 中的领域特征簇已经生成

else

重复(2)，继续聚类

(5)继续其他领域簇的聚类操作。

4.2　算法分析

在以上的聚类过程中，dfc 内的每个构件本体记录了它与簇代表 c^* 的相似度 s，因此可通过给定不同的相似度阈值调节 dfc 的大小。同时，也可通过修改 k 的大小来改变簇的划分，从而影响相似度的计算结果。各个簇的平均相似度值 $s'=dfc_i.s/(dfc_i.num-1)$可作为调整 k 值的依据。在 s'过低时，可以调高 k 值，再重新执行以上聚类算法。当每个本体自成一个簇时，$s'=0$。

通常情况下，将根据业务专家的领域分析预定义若干领域特征本体，并将其作为各 dfc 的簇代表，从而可在确保查准率的前提下，加快基于领域特征的聚类过程。本算法通过领域特征聚类方法生成精确的领域特征簇，从而为网构软件构件组装提供了决策的基础。

5　面向领域特征簇的构件组装算法

将图 1 中的各个构件视为构件对等体(CP)，则可根据图 1 定义一个有向图 $G(V,E)$，它具有以下特点：

(1)G 是一个 k 分图，$2\leqslant k$，cp_0（即图 1 中的 c_0）即是源点，也是终点。图 1 中，$k=n+2$，即领域特征簇、起点和终点各构成一个分图。

(2)G 是一个带权的有向图。边<i,j>的权值不仅由构件对等体 cp_i（即图 1 中的 c_i）、cp_j（即图 1 中的 c_j）间链路(i,j)的 QoS 值决定，还应该考虑 cp_j 的 QoS 值。QoS 值越大，则权值越大。

基于图 G 的构件组装问题实际上是一个求从 cp_0 出发，返回 cp_0 的最长路径问题。图 1 中的每个构件集合就是一个领域特征簇 dfc，每个 dfc 包含这条路径的一个节点。第 i 次决策就是确定 dfc_i 中选入路径的节点编号。显然，这个问题适合使用动态规划方法来解决。

5.1　边权值的计算

如图 1 所示，当位于装配点 cp_i（即图 1 中的 c_i）时，接下去是否选择下一个 dfc 中的 cp_j 作为新的装配点，将受到链路(i,j)的 QoS 值 $Q_{i,j}^L$ 和构件 c_j 的 QoS 值 Q_i^C 的影响。因此通过对 $Q_{i,j}^L$ 和 Q_i^C 进行换算实现边<i,j>的权值的度量。

构件的 QoS 定义为 $Q^C=\{Q^{ct},Q^{cc},Q^{ca}\}$，其中，$Q^{ct}$、$Q^{cc}$、$Q^{ca}$ 分别代表构件的执行时间、服务代价和可用性。链路的 QoS 定义为 $Q^L=\{Q^{ld},Q^{la}\}$，其中，Q^{ld}、Q^{la} 分别代表通信链路的端到端时延和可靠性。为便于计算和分析，将组装特定构件时的网络链路和该构件合称为“链路–构件对”，其 QoS 定义为 $Q^{LC}=\{Q^s,Q^c,Q^a\}$，其中，Q^s 代表链路–构件对的响应时间，即 $Q^s=Q^{ct}+Q^{ld}$；Q^c 代表链路–构件对的成本，$Q^c=Q^{cc}$；Q^a 代表链路–构件对的可用性，$Q^a=Q^{ca}*Q^{la}$。

构件组装时，用户对满足特定领域特征要求的构件的 QoS 需求偏好，可用三维行矩阵表示为：$W_i^r=\{w_1,w_2,w_3\}$。w_i 是对第 i 个待组装构件及调用链路 QoS 指标的权重偏好，$0\leqslant w_i\leqslant 1$，并且，$w_1+w_2+w_3=1$。$W_i^r$ 可采用熵值法等确定。通过将“链路–构件对”的 QoS 值进行归一化操作，并根据 W_i^r 加权求和，从而计算出任意两个构件对等体间的边的权值。令 $L_{i,j}$ 表示链路(i,j)间边的权值。

5.2　算法描述

令待组装构件的总数，即领域特征簇的总数为 n，则此时图 G 是一个 n+2 分图。设图 G 中节点总数为 $num=|V|$，由于图 1 中起点和终点相同，故实际节点总数为 num+1。设计邻接矩阵 $\boldsymbol{C}[i,j]$记录边<i,j>的权值，一维数组 $vd[i]$记录节点 i 所属的领域特征簇的 id，二维数组 $dfc[i,j]$记录领域特征簇 dfc_i 中的节点 j 的 id。

此外，算法执行还需要一些数据存储，如定义了一维数组 $cost[i]$记录节点 i 到达终点的最长路径值，一维数组 $path[i]$记录走过的路径上的节点 id，一维数组 $fd[i]$记录节点 i 的后继节点 id，经过该节点的路径最长。

使用前向递推算法求解构件组装的步骤如下：

(1)初始化：从图 G 中读入边的权值信息存入 $C[,]$、$vd[\ ]$和 $dfc[,]$，将 $cost[\ ]$和 $path[\ ]$的各个元素均置为 0，d=0。

(2)从终点出发，向前进行递推求解。

```
for（int j=num;j>=1;j--）
{
    t=0;maxvalue=0;
    foreach int r in dfc(vd(j)+1) //在后续 dfc 的节点中遍历
        //查找到达终点路径更长的后继节点
        if ((c[j,r]+cost[r])>maxvalue)
        {min=C[j,r]+cost[r];t=r;}
    cost[j]=C[j,t]+cost[t];//求出 j 到达终点的最短路径值
    fd(j)=t;//记录节点 j 的后继节点
}
```

(3)从起点开始,找出到达终点的最长路径。

$path[1]=path[n+2]=1$;//起点和终点均为节点 1,即 cp_o

for ($j=2;j<=n+1;j++$)

$path[j]=fd[path[j-1]]$;

(4)输出构件组装问题的最优解。

5.3 算法分析

根据 5.2 节的描述可以看出,算法的时间复杂度主要由两部分构成,包括算法第(2)节的时间复杂度和第(3)节的时间复杂度。算法第(2)节的时间复杂度是 O($num*z$),其中,$z=dfc_i.num$,$1\leqslant i\leqslant n$。显然,z 是一个远小于 num 的数。算法第(3)节的时间复杂度是 O(n),n 作为领域特征簇的总数,就实际情况而言,一个网构软件组装的构件个数不会太大。

算法的空间复杂度主要是图 G 的存储,包括邻接矩阵 $\boldsymbol{C}$,需要 num^2 个存储单元。其他数据结构的占用空间均远小于 $\boldsymbol{C}$。

6 仿真实验

在对算法进行时间复杂度和空间复杂度分析的基础上,进一步对第 5 章的构件组装算法进行了仿真实验。实验采用伪随机方式生成构件及链路的多维 QoS 属性。选取领域特征簇总数 n 为 2~30 之间,在各个簇中构件数均为 100 的情况下,测试在足够复杂度情况下算法的执行开销,得到 CPU 运行时间的增长曲线。由于构件-链路对 QoS 计算代价相对稳定,为便于直观比较,该实验忽略其时间消耗。实验结果如图 3 所示。由图可知,随着领域特征簇的不断增加,算法的执行开销并没有急剧增加。当领域特征簇总数为 30,构件总数达到 3 000 时,算法得到最终解的执行时间仍在 6 s 以内。显然,这样的执行开销完全可以接受,该文方法是有效和可行的。

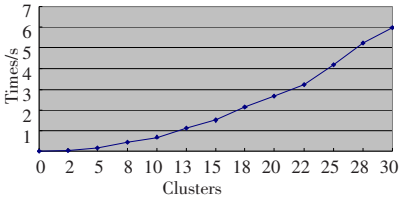

图 3 算法的执行开销

7 结语

提出一种 P2P 计算环境下面向领域特征聚类实现构件组装的方法,通过引入本体分类和相似度比较方法,设计了一种基于划分的聚类方法实现领域特征簇的计算,该方法可以实现精确的构件聚类,从而为构件组装过程提供决策基础。通过"链路-构件对"的 QoS 计算,可以换算得到边的权值,从而将网构软件的构件组装问题转换为一个多段图问题。给出了应用动态规划方法求解面向领域特征簇的构件组装全局最优解的算法实现。算法分析和足够复杂度下的实验仿真表明了该方法的有效性和可行性。

参考文献:

[1] 杨芙清,吕建,梅宏.网构软件技术体系:一种以体系结构为中心的途径[J].中国科学 E 辑:信息科学,2008,38(6):818-828.

[2] 吴国全,魏峻,黄涛.基于非确定性推理的网构软件服务质量动态评估方法[J].软件学报,2008,19(5):1173-1185.

[3] 王远,吕建,徐锋,等.一种面向网构软件体系结构的信任驱动服务选取机制[J].软件学报,2008,19(6):1350-1362.

[4] 赵丽娜,张引,叶修梓.基于 P2P 网络的网构软件自适应性研究[J].浙江大学学报:工学版,2008,42(8):1130-1136.

[5] 张伟,梅宏.一种面向特征的领域模型及其建模过程[J].软件学报,2003,14(8):1345-1356.

[6] Sanjay G,Shashishekar S T.Service-based P2P overlay network for collaborative problem solving[J].Decision Support Systems,2007,43(2):547-568.

[7] Maedche A,Staab S.Measuring similarity between ontologies[C]//Proc of the European Conference on Knowledge Acquisition and Management,Madrid,Spain,October 1-4,2002.

(上接 178 页)

Until *INIT* 标为 *SOLVED*(成功),或 *INIT* 的 f 值小于 μ(失败);

End

显然,根据上述算法结构,对于初始结点,可找到带标记的且满足搜索阈值要求的置信度最高的目标结点。5.1 的示例中,如搜索阈值 μ 定为 0.65,则可得出 $f(AGE(\mathrm{Li},50))=f(\neg ACCIDENT(X))=0.75$,即到结点 $\neg ACCIDENT(X)$ 的路径为最可信路径,从而推出李先生不会出交通事故。

6 总结

基于中介逻辑思想,合理修改与或图,将模糊知识的推理问题转化为状态空间中的搜索问题,并给出了一种否定信息的处理方法。此方法不仅便于计算机处理,而且可以节省计算机对于原子模糊概念的存储空间,更为模糊推理的计算机实现提供了一种思路。对于现实生活中的预测推理,如经济预测等问题上具有现实意义。

参考文献:

[1] Wagner G.Partial logics with two kinds of negation as a foundation for knowledge-based reasoning[C]//Gabbay D,Wansing H.What Is Negation.Oxford:Oxford University Press,1999:1-35.

[2] Kaneiwa K.Negations in description logic-contraries,contradictories,and subcontraries[C]//Dau F,Mugnier M L,Stumme G.Contributions to ICCS2005.Kassel:Kassel University Press,2005:66-79.

[3] Zhu Wu-jia,Xiao Xi-an.Propositional calculus system of medium logic(I)[J].Nature Magazine,1985(8):315-316.

[4] 朱梧槚,肖奚安.数学基础概论[M].南京:南京大学出版社,1996.

[5] Pan Zheng-hua,Zhu Wu-jia.An interpretation of infinite valued for medium proposition logic[C]//Proc of IEEE-Third International Conference on Machine Learning and Cybernetics,2004,4(7):2495-2499.

[6] 洪龙,肖奚安,朱梧槚.中介真值程度的度量及应用(I)[J].计算机学报,2006,29(12):2186-2193.

[7] 陈世福,陈兆乾.人工智能与知识工程[M].南京:南京大学出版社,1997.

[8] 吴望名.模糊推理的原理和方法[M].贵阳:贵州科技出版社,1994.

[9] 王岑,潘正华.基于中介逻辑的模糊知识表示及应用[J].计算机工程与科学,2008,30(11):80-82.

[10] 潘正华.知识中不同否定关系的一种逻辑描述[J].自然科学进展,2008,18(11):66-74.