基于标签推荐的Mashup服务聚类

黄 媛[1,3] 李 兵[1,2,3] 何 鹏[1,3] 熊 伟[1,3]

(武汉大学软件工程国家重点实验室 武汉 430072)[1] (武汉大学复杂网络研究中心 武汉 430072)[2]

(武汉大学计算机学院 武汉 430072)[3]

摘 要 聚类Web服务能大大提高Web服务搜索引擎检索相关服务的能力。ProgrammableWeb.com是一个很流行的在线社会Mashup网站。作为基于Web的应用程序,Mashup本质上是开发者提供的Web服务。结合Mashup服务的描述文档和相应标签提出一种新颖的Mashup服务聚类的方法,此外还提出一种标签推荐的方法来改进服务聚类的性能。实验结果表明,基于标签推荐的服务聚类方法的聚类精度比其他两种实验方法要高,说明提出的标签推荐策略有效扩充了标签数较少的Mashup服务,从而带来更多相关标签信息,因而聚类效果更好。

关键词 Mashup,标签,推荐,聚类

中图法分类号 TP39 **文献标识码** A

Mashup Services Clustering Based on Tag Recommendation

HUANG Yuan[1,3] LI Bing[1,2,3] HE Peng[1,3] XIONG Wei[1,3]

(State Key Laboratory of Software Engneering, Wuhan University, Wuhan 430072, China)[1]

(Complex Network Research Center, Wuhan University, Wuhan 430072, China)[2]

(School of Computer, Wuhan University, Wuhan 430072, China)[3]

Abstract Clustering Web services would greatly boost the ability of Web service search engine to retrieve relevant ones. The ProgrammableWeb.com is a popular online social Mashup site. Mashup as a Web-based application is Web services that developers provide. We proposed a novel approach, in which both description documents and tags are utilized for Web service clustering. Furthermore, we presented a tag recommendation strategy to improve the performance of this approach. The experimental results show that the accuracy of tag recommendation-based services clustering is higher than other two methods, which indicates that the tag recommended strategy effectively expands the number of tags of Mashup services with few tags, so introducing more tag information, thus the clustering effect is better.

Keywords Mashup, Tag, Recommendation, Clustering

1 引言

随着Web2.0技术的发展,Mashup如雨后春笋一般快速增长。Mashup网站是一个Web页面或应用程序,从两个或以上的外部在线资源连接数据。外部资源是其他的Web站点,Mashup开发员使用不同的方法获得站点上的数据,这些方法包括但又不仅仅局限于APIs、XML数据源和屏幕抓取[1]。每天都有大量Mashup涌现,因此我们需要一个平台来浏览这些Mashup。一些在线的社会平台,如Yahoo Pipes, Microsoft Popfly, ProgrammableWeb允许用户发布各种API,共享信息和第三方平台Mashup,迄今为止已注册了6730个Mashup和6783个开放的API。

最近,Mashup成为社会标注的Web资源。在这种情况下,Mashup演变成为一种轻量级的以用户为中心的方法,用以集成Web上的应用和数据。通过在线社区,非技术人员能够以新颖的方式集成已有的应用,这促进了Mashup的使用。由于用户友好界面Mashup工具的快速增加,例如YahooPipes, Intel Mash Maker和IBM Mashup中心,开发者将Mashup视作一种新的方法应用到更广阔的领域。

ProgrammableWeb是一个流行的在线社区,用户发布Mashup,并且对Web API和Mashup进行标注、排序。图1所示的是ProgrammableWeb上名称为Productism的Mashup应用的一些注册信息,包括Mashup的名称、描述信息、使用的API和对应标签等。这方面吸引了大量研究者的关注,促进了许多关于在线用户行为和Mashup创建的研究。在本文的工作中,标注的标签利用ProgrammableWeb发现API、标签与Mashup之间的关系。为了得到更精确的Mashup服务聚类,用这种关系改进Mashup服务聚类的性能。通过从ProgrammableWeb上爬下来的真实数据集上的试验评价了本文提出的方法。

到稿日期:2012-04-23 返修日期:2012-07-20 本文受国家自然科学基金(61273216,61272111,61202032,61202048)和教育部博士点基金(20090141120022)资助。

黄 媛(1983-),女,博士生,主要研究方向为复杂网络、服务计算、软件工程,E-mail:ttkldaisy@163.com(通信作者);李 兵(1969-),男,教授,博士生导师,主要研究方向为网络化软件、服务计算、复杂网络、软件工程、人工智能、云计算;何 鹏(1988-),男,博士生,主要研究方向为软件网络、软件工程;熊 伟(1973-),男,博士生,主要研究方向为服务计算、复杂网络。

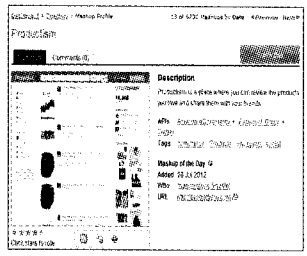


图 1　Mashup 应用的信息

本文的主要贡献如下：

(1)利用 Mashup 描述文本和标签信息提出一种新颖的 Mashup 服务聚类方法。

(2)提出一种标签推荐策略来改进 Mashup 服务的聚类性能。

(3)使用自己开发的网爬工具将 ProgrammableWeb 上从 2005 年(建站时)到 2011 年 1 月 12 日(本文工作开展时)所有 Mashup 应用的名称、描述信息、API 和标签信息爬了下来，存储在本地数据库中。这些数据是本文工作的基础。

2　相关工作

文献[2]介绍了一种个性化的交互标签推荐算法，依赖用户标注的标签的共现情况对个性化标签数据提供了一种特定的处理方法。P-TAG 算法[3]给 Web 页自动产生个性化标签，产生的标签与目标 Web 页的内容相关。Begelman 等人提到一种基于聚类的方法[4]，其将语义相关的用户标签进行聚类。标签以图的形式表示，每一个节点代表一个标签，节点之间的边代表它们之间的关系。基于相似性，同一个聚类中的标签会被推荐给用户。Xu 等人[5]提出一种基于 Web 页和用户之间的相似性的主题排序的方法，通过一种基于图的排序算法标签被排序，这种方法考虑了文档和用户偏好之间的相关性。

Xin Dong 等人从 WSDL 文档中提取参数名，然后将它们聚类成概念的形式[6]，从 Web 服务结构(包括服务名、文本、操作描述、输入/输出描述等)的角度计算了服务之间的相似性，并提出一种名为 Woogle 的搜索引擎，它支持 Web 服务相似性搜索。Nayak[7]提出基于搜索会话的服务聚类问题，而不是个人查询。Songlin Hu 等人[8]利用基于内容的模型来处理服务发现问题。Fangfang 等人[9]通过充分利用 WSDL 文档中的词语来找 Web 服务潜在的语义，还有其他的方法也用来计算两个服务之间的语义距离。文献[10]中作者使用关联规则算法识别频率标签共现模式，这些模式被认为是用户兴趣的主题。根据每个标签和对应主题之间的关系，不同主题之下的标签和链接被聚类。Ramage 等人[11]将社会性标签引入到基于 LDA 的 K-means 和启发式聚类方法当中，聚类结果用 Web 目录 ODP 进行评估。

3　基于扩充和精化的标签推荐

从图 2 可以看到标签频率的分布，Mashup 的标签频率最大值是 3，因为许多 Mashup 并没有使用 API，所以利用标签比用 API 更有效。许多关于计算标签和被标注实体之间相关性的研究都是从自身角度出发，然后衡量其相关性。在 ProgrammableWeb 中标注 Mashup 的标签数最多为 6，最少为 1，标签频率最大值为 3，标签数目分布不均。由于本文用到标签来计算两个服务的相似性，而标签数目过少将降低标签相似性的值，因此本文用标签扩充的方法来扩充每个 Mashup 服务的标签。这里针对标签数很少的 Mashup 服务推荐相关标签的标签推荐方法就很好地解决了上述问题。

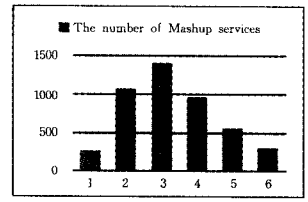


图 2　Mashup 服务和标签频率之间的关系

整个标签推荐的过程如图 3 所示。从图中可以看到整个标签推荐过程分为 2 部分，首先计算用户定义的标签和其他标签的共现系数，然后选择每个用户定义的标签的前 k 个共现标签作为候选标签。图 3 中 k 值设为 4，Mashup 服务 Amazigg 的 top-4 标签为 Map，Youtube，Events，search。有许多计算共现的方法，本文使用 Jaccard 系数的方法，具体计算公式如下：

$$Co(t_i,t_j)=\frac{|t_i \cap t_j|}{|t_i \cup t_j|} \tag{1}$$

式中，$|t_i \cap t_j|$ 代表同时拥有标签 t_i 和 t_j 的服务数目，$|t_i \cup t_j|$ 代表拥有标签 t_i 或者 t_j 的服务数目。经过标签推荐第一步之后，对每个用户定义的标签 $u \in U$(U 是用户定义的标签集)，我们能得到候选标签列表 C_u。

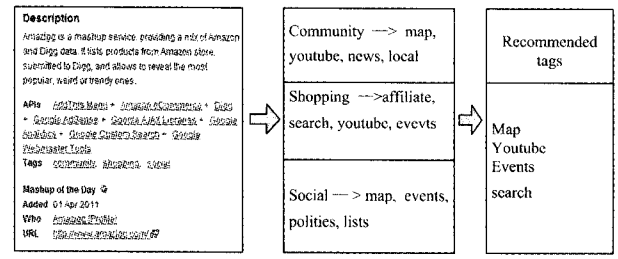


图 3　标签推荐过程

当每个用户定义的标签已知时，第二个步骤标签聚合就用来将扩充的标签合并成为一个单独的标签序列。这一部分基于投票求和策略定义的标签聚合方法，促进了候选标签的重新排列。

这里定义了 3 种不同类型的标签。

用户定义的标签 U：用户赋予 Mashup 服务的标签集合。

候选标签集 C_u：对一个用户定义的标签 $u \in U$，C 是 u 的所有候选标签的集合，C_u 是 u 的 top-k 个共现标签。

推荐的标签集 R：由标签推荐过程产生的 n 个最相关标签的列表。

对一个给定的候选标签集合 C，标签聚合步骤用来产生最终的标签推荐列表 R，在第二步中，对候选标签和选择的 top-k 个标签进行排序，并将其作为最后的推荐标签。这里使用的聚合策略：投票和求和。

投票：计算候选列表中所有标签的共现值，通过共现的分数来排列标签，然后选择最终的推荐结果。

$$vote(t,u)=\begin{cases}1, & \text{if } t \in T_u \\ 0, & \text{otherwise}\end{cases} \tag{2}$$

$$score(t)=\sum_{u \in U} vote(t,u) \tag{3}$$

求和：在求和策略中，通过对 c 和用户定义的标签 u 之间共现值的求和，我们计算了候选标签 c 的分数，方程如下：

$$score(c)=\sum_{u \in U} Co(u,c) \tag{4}$$

式中，$Co(u,c)$的值利用式(1)计算。

4 Web服务聚类

4.1 预处理文档

这一步骤处理Mashup描述文档，包括移除停用词、使用词干分析器找到源词等。

(1)提取词语。将句子划分成词，然后提取名词作为词语特征词。

(2)移除停用词。使用停用词列表，其中包含要排除的词。这个列表用来移除普遍使用的不能区分主题的词。

(3)处理词干。使用已有的词干算法比如Porter算法来将一个词转化为它的词干或词根的形式。

(4)选择关键词。应用tf-idf阈值方法来选择文档集的关键词。

4.2 皮尔逊相关系数

皮尔逊相关系数是一种度量两个向量相关性的方法。皮尔逊相关系数的公式有很多种形式，这里给定词集 $T=\{t_1,\cdots,t_m\}$，一种普遍采用的公式形式如下：

$$SIM_p(\vec{t_a},\vec{t_b})=\frac{m\sum_{t=1}^{m}w_{t,a}\times w_{t,b}-TF_a\times TF_b}{\sqrt{[m\sum_{t=1}^{m}w_{t,a}^2-TF_a^2][m\sum_{t=1}^{m}w_{t,b}^2-TF_b^2]}} \quad (5)$$

式中，$TF_a=\sum_{t=1}^{m}w_{t,a}$且$TF_b=\sum_{t=1}^{m}w_{t,b}$。

两个词集的关联系数SIM是一个介于-1到1之间的数，它表明两个集合之间关联的重要性，若$SIM<0$，表明两个变量是负相关，即一个变量的值越大，另一个变量的值反而会越小。SIM的绝对值越大表明相关性越强，要注意的是这里并不存在因果关系。若$SIM=0$，表明两个变量间不是线性相关，但有可能是其他方式的相关(比如曲线方式)。当$\vec{t_a}=\vec{t_b}$时$SIM=1$。在后面的试验中用皮尔逊相关系数计算两个Mashup描述文本之间的相似性。

4.3 API、tag相似性

服务的标注数据描述了服务的功能，提供了额外的文本和语义信息。本文提出一种利用标签信息改进Mashup服务聚类的性能的方法。通过Jaccard系数[12]方法，利用以下公式计算TAG和API的相似性。

$$\mathrm{sim}_{Tag}(s_i,s_j)=\frac{|T_i\cap T_j|}{|T_i\cup T_j|} \quad (6)$$

$$\mathrm{sim}_{API}(s_i,s_j)=\frac{|A_i\cap A_j|}{|A_i\cup A_j|} \quad (7)$$

式中，$|T_i\cap T_j|$意为同时标注Mashup服务s_i和s_j的标签数，$|T_i\cup T_j|$是指这两个服务的标签集T_i和T_j的并集。$|T_i\cup T_j|=|T_i|+|T_j|-|T_i\cap T_j|$，那么$A_i$，$A_j$意为同时标注Mashup服务$s_i$和$s_j$的API服务的数目。

4.4 相似性集成

通过以上的相似性计算公式，得出两个Mashup服务的组合相似性$\mathrm{sim}(s_i,s_j)$的计算公式如下：

$$\mathrm{sim}(s_i,s_j)=\alpha*Description(s_i,s_j)+\beta*APICooccurencysim(s_i,s_j)+(1-\alpha-\beta)*TagCooccurencysim(s_i,s_j) \quad (8)$$

式中，α是描述文本层相似性的权值，$Description(s_i,s_j)$是Mashup服务描述文本层相似性，$\mathrm{sim}(s_i,s_j)$由两个服务间的3部分相似性构成，α,β是用户自定义的描述层、API层的权值，且$\alpha+\beta=1$。

4.5 基于Mashup服务相似性的K-Means算法

数据聚类是指根据内在性质将数据分成一些聚合类，每一聚合类中的元素尽可能具有相同的特性，不同聚合类之间的特性差别尽可能大。长期以来，人们提出许多数据聚类算法，在众多聚类算法中K-Means算法的应用领域非常广泛，包括图像及语音数据压缩，使用径向基函数网络进行系统建模的数据预处理，以及异构神经网络结构中的任务分解[13]。

传统的K-Means算法是一种基于分割的聚类方法，在数据挖掘领域中得到了最广泛应用。其思想是以K为参数，把n个对象分割为K个簇，以使簇内具有较高的相似度，而簇间的相似度较低，并使得在每个聚类中所有值与该聚类中心距离的总和最小。每个聚类中心是每个聚类的均值。相似度的计算根据一个簇中对象的平均值来进行，算法选择的相似性度量通常是欧几里德距离的倒数，即两者的距离越小表示两者的相似性越大，反之则相似性越小。

这里提出一种基于Mashup服务相似性的K-Means算法，算法中尽量让不同簇中的点作为初始中心点，且这些点与簇中其他点关系紧密。

首先构建一个Mashup服务间的相似度矩阵M，其定义如下：

$$M=\begin{bmatrix}s_{11} & \cdots & s_{1n}\\ \vdots & \ddots & \vdots\\ s_{n1} & \cdots & s_{nn}\end{bmatrix} \quad (9)$$

式中，s_{ij}表示Mashup服务i和服务j之间的相似度，其中n为数据集中所有Mashup服务的总数，$s_{ij}=s_{ji}$，$s_{ii}=0$。

$$P=\{a_1,a_2,\cdots a_n\} \quad (10)$$

$$a_i=\frac{s_{i1}+s_{i2}+\cdots s_{in}}{n} \quad (11)$$

式中，P为Mashup服务的平均相似度集合，a_i表示Mashup服务的平均相似度。

基于Mashup服务相似性的K-Means算法如下：

输入：Mashup服务的向量空间模型，聚类个数k，参数γ，i=0(表示已选择的聚类中心的个数)

输出：k个聚类

步骤1 计算Mashup服务的相似度，构造相似度矩阵M；

步骤2 根据式(11)构造集合P，并且对集合P进行升序排序；

步骤3 初始中心点集I初始化为空集，即I=∅，删除集：Delete=∅；

步骤4 从集合P中选择值最大的Mashup服务S_j作为中心点，将S_j加入初始中心点集$I=I\cup\{S_j\}$，已选择中心点个数加1，即i=i+1；

步骤5 根据M寻找和S_j所有相关的Mashup服务，然后从集合P中删除，即如果$\mathrm{sim}(s_i,s_j)>\gamma$，$P=P-\{a_1\}$且$Delete=Delete\cup\{a_1\}$；

步骤6 如果P=∅且i<k，将删除集合中的Mashup服务重新加入集合P，即P=Delete；

步骤7 重复步骤(3)—(6)，直到i=k结束输出初始中心点集I；

步骤8 根据式(8)计算每个Mashup服务和各聚类中心的相似度，将Mashup服务分配到与其相似度最大的聚类中心点所代表的簇中；

步骤9 通过公式$\mathrm{centorid}_i=\frac{\sum_{s_k\in C_i}s_k}{n}$重新计算各聚类中心点；

步骤10 重复步骤(8)、(9)，直到算法收敛或达到指定的迭代次数；

步骤11 输出聚类结果。

5 实验

5.1 数据集和评价指标

为了评价 Mashup 服务聚类的性能，从 ProgrammableWeb 上爬下来 4505 个 Mashup 服务作为实验的数据集。对于每一个 Mashup，得到其服务名称、描述文本、标签等信息。

这里选择的聚类度量指标为精度，它广泛应用于信息检索领域。

$$Precision_{c_i}=\frac{succ(c_i)}{succ(c_i)+mispl(c_i)} \tag{12}$$

式中，$succ(c_i)$ 是成功聚类到类 c_i 中的 Mashup 服务的数目，$mispl(c_i)$ 是错误划分到类 c_i 中的 Mashup 服务的数目。

每一个 Mashup 服务对应的每一个标签和其相关度可以分为 3 个等级：不相关、相关、很相关。这里使用 top-k Discounted Cumulative Gain(DCG) 作为标签推荐评价指标：

$$DCG_k=rel_1+\sum_{i=2}^{k}\frac{rel_i}{\log_2 i} \tag{13}$$

式中，rel_i 是 Mashup 服务对应的第 i 个标签的相关等级。

5.2 实验方法

图 4 是本文服务聚类的框架。由于 Mashup 数据包括描述文本、API 和手动标注的标签，我们遵循以下的步骤进行聚类：

步骤 1　对 Mashup 服务的描述文本进行预处理，比如移除停用词，使用词干分析器找到源词，识别命名实体等；

步骤 2　使用投票求和的方法对标签进行扩充和精化之后的结果为推荐的标签；

步骤 3　将经过上两步之后得到的经过预处理之后的 Mashup 描述文本和 API 服务、推荐的标签及原来手动标注的标签，合在一起进行聚类。

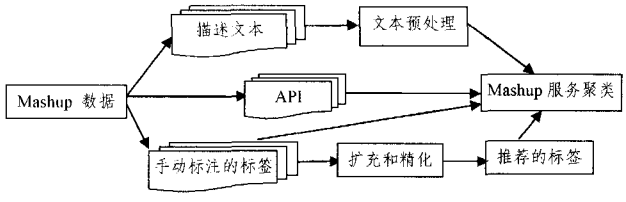


图 4　服务聚类的框架

5.3 实验结果分析

为了评价本文中的标签推荐方法，从 4505 个 Mashup 服务中选择 200 个标签数为 2 的 Mashup 服务进行标签推荐。我们邀请了 3 名有 Web 服务开发经验的程序员对标签推荐的结果进行评分，表 1 所列为标签推荐中选择 top-k 个标签的 k 值在 1 到 5 之间变化时推荐标签的平均 DCG 值，可见 k 等于 3 时标签推荐得到的平均 DCG 值最大。

表 1　标签推荐的平均 DCG

	k=2	k=3	k=4	k=5	k=6
平均 DCG	2.84	3.67	3.456	2.55	2.97

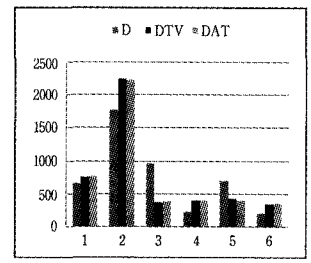
这一节介绍了 3 种 Mashup 服务相似性计算方法：

(1) D：在这种方法中，通过计算 Mashup 服务的描述文本相似性来对 Mashup 服务(利用式(5)计算)进行聚类。

(2) DAT：在这种方法中，通过计算 Mashup 服务的描述文本、API 和原有标签的组合相似性(利用公式(8)计算)来对 Mashup 服务进行聚类。

(3) DTV：在此方法中，首先利用投票求和策略实现标签推荐的过程，然后利用描述文本和推荐之后的标签的组合相似性计算 Mashup 服务的相似性，从而对 Mashup 服务进行聚类。

这里采用本文中的基于 Mashup 服务相似性的 K-Means 算法对以上 3 种相似性计算方法的聚类结果进行比较，结果如图 5、图 6 所示。



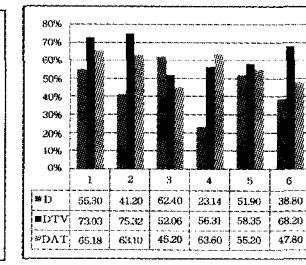


图 5　Mashup 服务聚类结果统计　图 6　Mashup 服务聚类精度比较

使用了本文的基于 Mashup 服务相似性的 K-Means 算法之后的聚类结果如图 5 所示，4505 个 Mashup 服务被分为 6 类。为了验证分类结果是否有意义，手工对 6 个分类中共 4505 个 Mashup 应用进行分析，以此作为聚类结果的基线。第 1 类的 Mashup 服务是与网站相关的内容，比如 ebay 电子商务平台、del. ici. ous 书签网站、twitter 社交网站等。第 2 类是一些与地图相关的 mashup 应用，且大多数应用都使用了“Google Maps”这个 API 服务。第 3 类是关于 Google 小工具的内容。第 4 类主要是与艺术密切相关的 Mashup 服务，例如照片、音乐等。第 5 类主要是与通信内容相关的 Mashup 服务。

从图 6 可以看到，用 D 方法计算 Mashup 服务相似度进行聚类的平均聚类精度为 45.46%，用 DAT 方法得到的平均聚类精度为 56.68%，DTV 方法计算相似度进行聚类的平均聚类精度为 63.88%。可见，用描述文本与本文的标签推荐方法相结合来计算 Mashup 服务相似性的方法进行服务聚类精度最高，DAT 方法次之，仅用描述文本计算相似性进行聚类精度相对最低。对凝聚层次聚类方法也做了比较，但是凝聚层次聚类方法时间复杂度明显太高，这里不多分析。实验结果说明对于标签数较少的 Mashup 服务来说，通过标签推荐的方法扩充其标签，增加了许多相关的标签，这样一定程度地提高了 Mashup 服务聚类的精度。

结束语　文中提出一种标签推荐的方法来改进 Mashup 服务聚类的性能，在 DTV 方法中我们利用 Mashup 描述文本的相似性和其对应推荐后的标签的相似性的组合作为 Mashup服务的相似性。为了验证服务聚类的性能，从 ProgrammableWeb 上爬下来 4505 个 Mashup 服务作为实验的数据集，然后使用本文中基于 Mashup 服务相似性的 K-Means 算法对 Mashup 服务进行聚类，同时对仅用描述文本(D)、描述文本与对应的 API 和相应的标签(DAT)组合这两种计算相似性的方法进行比较，结果表明 DTV 比其余两种方法聚类效果要好，说明本文提出的标签推荐策略有效扩充了标签数较少的 Mashup 服务，带来了更多相关标签信息，使其聚类效果更好。

参 考 文 献

[1]　Yu Jin, Benatallah B, Casati F, et al. Understanding Mashup

development[J]. IEEE Internet Computing, 2008, 12(5): 44-52

[2] Garg N, Weber I. Personalized, interactive tag recommendation for flickr[C]// Proceedings of the 2008 ACM Conference on Recommender Systems. 2008: 67-74

[3] Chirita P A, Costache S, Nejdl W, et al. P-tag: large scale automatic generation of personalized annotation tags for the Web[C]// Proceedings of the 16th International Conference on World Wide Web. 2007: 845-854

[4] Begelman G, Keller P, Smadja F. Automated tag clustering: Improving search and exploration in the tag space[C]// Proc. Collaborative Web Tagging Workshop WWW. Edinburgh, U. K., 2006: 274-288

[5] Xu S, Bao S, Fei B, et al. Exploring folksonomy for personalized search[C]// Proc. 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Singapore, 2008: 155-162

[6] Dong X, Halevy A, Madhavan J, et al. Similarity search for Web services[C]// International Conference on Very Large Data Bases. 2004: 372-383

[7] Nayak R. Data mining in web service discovery and monitoring [J]. International Journal of Web Services Research, 2008, 5 (1): 62-80

[8] Hu S, Muthusamy V, Li G, et al. Distributed automatic service composition in large-scale systems[C]// Proc. of Distributed Event-Based Systems Conference. 2008: 233-244

[9] Liu F, Shi Y, Yu J, et al. Measuring similarity of web services based on wsdl[C]// International Conference on Web Services. 2010: 155-162

[10] Li X, Guo L, Zhao Y. Tag-based social interest discovery[C]// Proc. 17th Int. Conf. World Wide Web. Beijing, China, 2008: 675-684

[11] Ramage D, Heymann P, Manning C D, et al. Clustering the tagged Web[C]// Proc. 2nd ACM Int. Conf. Web Search Data Mining. Barcelona, Spain, 2009: 54-63

[12] 潘伟丰，李兵，邵波，等. 基于软件网络的服务自动分类和推荐方法研究[J]. 计算机学报，2011，34(12)：2355-2369

[13] 王伟强，高文. Intemet 上的文本数据挖掘[J]. 计算机科学，2000，27(4)：32-37

(上接第 166 页)

服务组合领域的可信性评估。相对于嵌入式系统等其他软件实体，通用的层次型软件评估模型拥有良好的适用性，而本文提出的 Web 服务组合可信评估方法具有一定的局限性。

结束语 Web 服务可信旨在分析、评估服务本身的质量，在 Web 服务应用日益广泛的今天，用户的需求已经不再是单一服务可以满足的，组合服务的质量和可信度便越来越受到关注。由此，本文结合软件可信评估规范，提出一种通用的原子服务可信评估模型；再从 WS-BPEL 中抽取服务组合的控制结构，配置原子服务在组合服务中的权重，通过深度优先搜索遍历服务组合的执行流程，从而得出组合服务的可信评估结果；最后针对一个网络购物的实例进行了分析。

与其他 Web 服务可信评估工作相比，本文工作具有以下特点：

(1)服务可信评估模型能覆盖 Web 服务的产品、过程、资源等各方面，而不仅仅针对单一属性。

(2)原子服务可信评估模型中的评估准则和服务组合可信评估算法适用于以比率型标度进行计算，使得可信评估结果更为精确。

(3)设计服务组合可信评估求解算法，充分考虑原子服务在服务组合执行过程中的权重。

在本文的服务组合可信评估过程中，单一地考虑了服务组合的控制结构，对于原子服务之间的消息响应机制、数据传递方式以及隐私保护等因素未予以考虑。在未来的工作中，将对原子服务之间的数据传递方式和隐私保护等做进一步的研究。

参 考 文 献

[1] Zhang J. Trustworthy Web Services: Actions for Now [J]. IT Professional, 2005, 7(1): 32-36

[2] Zaki M, Athman B. Reputation Bootstrapping for Trust Establishment among Web Services[J]. IEEE Internet Computing, 2009, 13(1): 40-47

[3] 蔡斯博，邹艳珍，邵凌霜，等. 一种支持软件资源可信评估的框架[J]. 软件学报，2010，21(2)：359-372

[4] 郎波，刘旭东，王怀民，等. 一种软件可信分级模型[J]. 计算机科学与探索，2010，4(3)：231-239

[5] 洪宏，黄志球，沈国华，等. 支持软件可信评估的框架及其应用研究[J]. 计算机科学与探索，2011，5(2)：170-178

[6] 沈国华，黄志球，钱巨，等. 软件可信评估模型及其工具实现[J]. 计算机科学与探索，2011，5(6)：553-561

[7] Zhao Wei-nan, Sun Hai-long, Huang Zi-cheng, et al. A User-Oriented Approach to Assessing Web Service Trustworthiness [C]// ATC'10 Proceedings of the 7th International Conference on Autonomic and Trusted Computing. 2010: 195-207

[8] Kalepu S, Krishnaswamy S, Loke SW. Verity A QoS Metric for Selecting Web Services and Providers[C]// Proceedings of the 4th International Conference on Web Information Systems Engineering. ROME: Fourth International Conference on Web Information Systems Engineering Workshops. 2003: 131-139

[9] Kim Y, Doh K G. A Trust Type Based Model for Managing QoS in Web Services Composition[C]// Proceedings of the 2007 International Conference on Convergence Information Technology. Washington: IEEE Computer Society, 2007: 438-443

[10] 刘国奇，朱志良，王浩，等. 一种 Web 服务 QoS 可信性评价模型[J]. 小型微型计算机系统，2009，30(11)：2216-2221

[11] 肖文，张自力，李伟华. 基于 QoS 的可信 Web 服务组合研究[J]. 计算机科学，2011，38(6)：173-176

[12] 孟琳琳，赵伟男，孙海龙，等. Web 服务可信证据收集与评估机制研究[J]. 计算机科学与探索，2011，5(7)：642-651

[13] 曾晋，孙海龙，刘旭东，等. 基于服务组合的可信软件动态演化机制[J]. 软件学报，2010，21(2)：261-276

[14] OASIS. Business Process Execution Language (BPEL4WS/BPEL) [OL]. http://docs. oasis-open. org/wsbpel/2. 0/OS/wsbpel-v2. 0-OS. pdf, 2007