基于语义相似度的Web数据库不精确查询方法

孟祥福[1] 张霄雁[1] 马宗民[2] 张志艳[1]
(辽宁工程技术大学电子与信息工程学院 葫芦岛 125105)[1]
(东北大学信息科学与工程学院 辽宁 110819)[2]

摘 要 为了解决普通用户对于Web数据库的不精确查询问题,提出了一种基于语义相似度的Web数据库不精确查询方法。对于一个给定查询,该方法首先在查询历史中找出一个(或若干)与其相似度高于给定放松阈值的查询,然后从数据库中找出与这些查询相匹配的元组作为当前查询的不精确查询的结果,最后将这些查询结果按其对初始查询的满足程度进行排序。实验结果表明,提出的不同查询之间的语义相似度评估方法性能稳定、评估结果合理,不精确查询方法具有较高的查全率和排序准确性。

关键词 Web数据库,不精确查询,关联规则,语义相似度
中图法分类号 TP391 **文献标识码** A

Semantic Similarity-based Approach for Answering Imprecise Queries over Web Databases

MENG Xiang-fu[1] ZHANG Xiao-yan[1] MA Zong-min[2] ZHANG Zhi-yan[1]
(College of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, China)[1]
(College of Information Science and Engineering, Northeastern University, Shenyang 110819, China)[2]

Abstract To deal with the problem of answering the Web database imprecise queries, this paper proposed a semantic similarity-based Web database imprecise query approach. For a given query, one or several similar queries in the query history will be found firstly, and the similarity of each similar query to the original query is greater than the given relaxation threshold. Then, the tuples matched to these queries are treated as the imprecise query results to the current query. Finally, the result tuples are ranked according to their satisfaction to the original query. Results of experiments demonstrate that the query similarity measuring method proposed is stable and reasonable, and the imprecise query method proposed has higher recall and the ranking accuracy as well.

Keywords Web database, Imprecise query, Association rule, Semantic similarity

1 引言

随着World Wide Web的不断发展,网络上的在线数据库也越来越多,这些可访问的在线数据库称为Web数据库[1]。随着Internet的普遍应用和Web数据库中所蕴含信息量的快速增长,访问Web数据库已成为人们获取信息的重要手段[2,3]。然而,当前的Web数据库查询处理模式仅支持严格的查询匹配,返回的查询结果也完全满足用户提出的查询要求,而实际上普通用户的查询意图往往是模糊或不精确的,也就是说用户提交的查询并不一定能完全体现其查询意图,一些与查询相关的信息也可能是用户需要的。例如,通过Amazon中文图书网站的查询接口,用户可能会提出如下查询:

Q:-(译著者="严蔚敏",商品名="数据结构")

对于上述查询,当前的Web数据库查询处理模式只能返回严格满足严蔚敏所著的数据结构图书,而实际上用户也可能希望看到严蔚敏所著的其他图书,或者是其他译著者所著的数据结构图书,并且用户还希望这些图书按照对初始查询的相关程度进行排序。由此可见,对于Web数据库不精确查询方法的研究有着十分重要的现实意义。

目前,已有一些工作研究了数据库的模糊查询方法[4-6]。这些方法允许用户提出直接含模糊词或模糊关系的模糊查询条件,然后基于模糊集理论[7],通过隶属函数和知识库将模糊查询条件转换成传统数据库所支持的精确查询条件,最后再到数据库中获取相关结果元组,从而实现对数据库的不精确查询。然而,这类方法存在的最大不足就是隶属函数的设定需要领域知识的支持,查询转换过程不是全自动的。本文从评估不同查询之间语义相似度的角度,研究了Web数据库的不精确查询方法,该方法能够独立于领域和用户而自动运行。

本文第2节介绍不精确查询定义和解决方案;第3节描述不精确查询的实现方法;第4节是效果与性能实验评价;最

到稿日期:2011-05-19 返修日期:2011-07-05 本文受国家青年科学基金项目(61003162)资助。
孟祥福(1981-),男,博士,讲师,主要研究方向为Web数据库和XML柔性查询技术,E-mail:marxi@126.com;张霄雁(1983-),女,硕士,助教,主要研究方向为Web数据库与XML查询优化技术;马宗民(1965-),男,博士,教授,主要研究方向为智能数据与知识工程;张志艳(1978-),女,硕士,讲师,主要研究方向为XML不精确查询技术。

后总结全文。

2 不精确查询定义和解决方案

本节首先给出不精确查询的定义，然后描述不精确查询解决方案。

2.1 不精确查询定义

假设 Web 数据库 D 中包含 n 条元组$\{t_1,\cdots,t_n\}$，其模式由 m 个分类型和数值型相混合的属性 $A=\{A_1,A_2,\cdots,A_m\}$ 构成，Q 为 D 上的一个合取查询，形式为 $Q=\sigma\wedge_{i\in\{1,\cdots,k\}}C_i$，其中 $2\leqslant k\leqslant m$，C_i 的形式为 $A_i\theta a_i$，$\theta\in\{>,<,=,\geqslant,\leqslant,\text{between}\}$，如果 θ 是操作符 between，则 a_i 用区间$[a_{i1},a_{i2}]$表示，且 $A_i\theta a_i$ 的形式为“A_i between a_{i1} and a_{i2}”。基本查询条件“$A_i\theta a_i$”中的每个属性 A_i 都是属性集 A 中的属性，a_i 包含于属性 A_i 的值域当中。通过放松初始查询 Q，使得查询结果中的任一元组对初始查询 Q 的满足程度都高于指定的放松阈值 $T_{\text{sim}}\in(0,1]$，表示为

$$\tilde{Q}(D)=\{t\mid t\in D,\text{Satisfaction}(t,Q)>T_{\text{sim}}\}\quad(1)$$

式中，$\tilde{Q}(D)$是 D 中对初始查询 Q 的满足程度高于给定放松阈值 T_{sim}的查询结果集合，T_{sim}由领域专家或用户给定。

2.2 解决方案

本文提出的不精确查询方法的基本思想是，首先对于用户提交的初始查询，从查询历史(以往使用过该系统的用户提交的查询记录的集合)中找出与其相似的历史查询记录，这些历史查询记录与给定查询的相似度高于给定的放松阈值；然后，在数据集上分别执行这些查询，得到查询结果集，并且去掉重复元组；最后，对查询结果中的元组按其对初始查询的满足程度进行排序。图 1 给出了本文提出的不精确查询解决方案的整体框架。

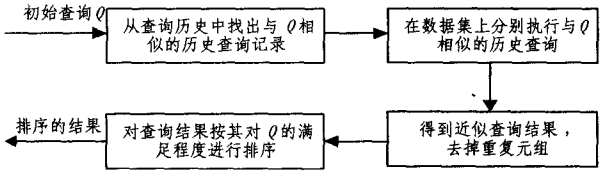

图 1 不精确查询解决方案的整体框架

由图 1 可见，Web 数据库不精确查询主要分为 3 个步骤：

(1) 查询之间的语义相似度评估：对于一个给定的查询，评估它与查询历史中不同查询记录之间的语义相似度，进而获取那些相似度高于给定放松阈值的历史查询。

(2) 查询执行：在数据库上执行所有相似的查询，获得查询结果集，去掉重复元组。

(3) 查询结果排序：对于查询结果中的元组，按其对初始查询的满足程度进行排序，最后把有序的查询结果返回给用户。

3 不精确查询实现方法

本节首先描述不同查询之间的语义相似度评估方法，然后给出查询执行和结果排序方法。

3.1 查询之间的语义相似度评估

如前所述，对于一个给定查询，首先需要从查询历史中找出与其相似的查询，然后通过在 Web 数据库上执行这些相似查询获取相关结果。因此，查询之间的语义相似度评估在本文所提的不精确查询方法中具有重要作用。

对于查询之间的语义相似度评估，传统方法利用两个查询的结果集之间的重叠程度(即相同元组数)来进行衡量。在关系数据模型下，两个元组相同是指它们在所有属性上都具有相同的属性值。然而，该方法存在一个严重不足：如果两个查询在同一属性上指定了两个不同的分类型值，那么对应的两个查询结果集之间将不会有重叠的元组。例如，基于 Amazon 中文图书数据库，它包含一个关系表 BookDB(商品名，译著者，出版社，定价，ISBN，出版时间，类别，折扣)，令“Q_1:-BookDB(译著者＝李春葆∧定价 between 10 and 20)”和“Q_2:-BookDB(译著者＝严蔚敏∧定价 between 10 and 20)”是 BookDB 上的两个查询。在关系数据模型下，这两个查询虽然都指定了相同的定价区间，但它们之间的相似度仍然为 0，因为它们在“译著者”属性上指定了两个完全不同的值，从而导致对应这两个查询的结果集之间交集为空。实际上，即便是两个查询结果集之间不存在完全相同的元组(即元组在所有属性上都匹配)而仅是元组在部分属性上相匹配，它们之间也可被认为是相似的。重新考虑上述两个查询，一方面，它们都指定了相同(或相似)的定价区间；另一方面，它们虽然在“译著者”属性上指定的两个值“李春葆”和“严蔚敏”之间没有显示出任何的相似性，但这两个译著者可能在相同的出版社、相同的时间以相似的价格出版过相同题材的图书。本文目的在于使用这种隐含的关系鉴别出两个查询之间的语义相似性。如果保持严格的元组结构匹配，那么两个查询之间相似，仅是因为它们的结果集之间包含了相同的元组。所以，仅当“李春葆”和“严蔚敏”是合作者的情况下，这两个查询之间才是相似的。然而，实际上即便他们不是合作者，“李春葆”和“严蔚敏”之间也是密切相关的，因为从 Amazon 中文图书数据库上查找到由他们撰写的图书来看，他们在相同的时间(2002，2004 和 2006 年)以 10～20 元的价格通过清华大学出版社发行过主题为“数据结构”方面的图书。基于上述直觉，本文提出把关系数据模型转换到向量空间模型来有效地获取两个不同查询之间的这种隐含关系。

对于两个不同查询之间的语义相似度评估，首先将对应每个查询的结果集转换成一个语义表，该语义表是一个两栏结构，由属性(Attributes)和包(Bags)两列构成，表 1 和表 2 分别给出了 BookDB 上对于查询“译著者＝严蔚敏∧定价 between 10 and 20”和查询“译著者＝李春葆∧定价 between 10 and 20”的语义表，该语义表可看成是查询结果集的一个向量表示，其中每个属性都对应向量中的一个分量。语义表中包含了对应于关系中每个属性的一组关键字，关键字从查询结果集中抽取，对于分类型属性，每一个不同的属性值就是一个关键字；对于数值型属性，通常将其值域划分成若干数值区间，其中每个数值区间都作为一个关键字，为使每个数值区间包含的元组数大致相同，本文采用等深直方图划分方法。为使关键字具有语义功能，本文用关键字和它在查询结果集中对应的元组数(即〈关键字：对应的元组数〉)来表达关键字的语义，这里将对应于同一属性的所有〈关键字：对应的元组数〉称为包(Bags)。例如，在表 1 中，属性“定价”上的包为“10～15：4；15～20：5”，包中的关键字分别是“10～15”和“15～20”，其中关键字“10～15”对应的值为 4，这表明该查询结果集中

有 4 本图书的定价在 10～15 元之间。

表 1　对应查询"译著者＝严蔚敏∧定价 between 10 and 20"的语义表

Attributes	Bags
商品名	数据结构:9
译著者	严蔚敏:9
出版社	清华大学:9
定价	10～15:4;15～20:5
ISBN	B3020:4;B9787:5
出版时间	2007:3;2004:4;2002:2
类别	教材:6;计算机:3
折扣	3～5:0;5～7:0;7＋:9

表 2　对应查询"译著者＝李春葆∧定价 between 10 and 20"的语义表

Attributes	Bags
商品名	数据结构:10
译著者	李春葆:10
出版社	清华大学:8;高等教育:2
定价	10～15:5;15～20:5
ISBN	B3141:6;B0535:4
出版时间	2007:4;2004:4;2002:2
类别	教材:6;计算机:4

因为语义表中包含了对应于每个属性的包，并且每个包中又包含了一组〈关键字:对应的元组数〉，所以可用 Jaccard 系数来计算两个不同语义表中对应包之间的语义相似度。在此基础上，通过合并两个语义表中所有对应包之间的语义相似度，可获得两个语义表之间的语义相似度(也就是这两个语义表所对应的两个查询之间的语义相似度)。需要指出的是，在评估两个语义表之间的相似度过程中，语义表中的每个属性对于用户的重要程度是不同的。例如，对于两本不同的图书，在决定二者之间的相似度时，它们的商品名和定价可能比出版时间显得更重要。因此，两个不同查询之间的相似度，应该是相应的两个语义表中带权重的包之间相似度之和，即

$$VSim(Q_1,Q_2)=\sum_{i=1}^{m}Sim(S_1.Bag_i,S_2.Bag_i)\times W(A_i) \quad (2)$$

式中，Q_1 和 Q_2 代表两个不同的查询；S_1 和 S_2 分别是对应查询 Q_1 和 Q_2 结果集的两个语义表(每个语义表都包含 m 个属性)；Bag_i 是语义表中第 i 个属性对应的包；$W(A_i)$是属性 A_i 的权重($\sum_{i=1}^{m}W(A_i)=1$)。

注意，为了减少在线查询阶段的计算工作量，在离线阶段利用本文方法预计算出每对不同历史查询之间的语义相似度，然后把相似的历史查询聚合到一个集合中，并且从中选出一个代表性查询。在线查询阶段，当新的查询到来时，只需计算该查询与每个查询集合中的代表性查询之间的相似度，即可快速找出与其最为相似(即相似度高于给定阈值)的那些历史查询记录。

3.2　查询执行和结果排序

在线查询阶段，对于一个给定的查询，利用上述查询之间的语义相似度评估方法，在查询历史中找出相似度阈值高于给定放松阈值的那些历史查询记录，然后通过在数据集上依次执行这些查询得到查询结果。不精确查询实现算法描述如下(算法 1)。

算法 1　不精确查询实现算法

输入：查询集合 QC＝(Q_1,Q_2,…,Q_n)，Web 数据库 D

输出：近似查询结果集 T

1. 令 U 为一个大小为 n 的数组，用于存储查询集合 QC；
2. for (i＝1;i<＝n;i＋＋)
3. 　Q_i(D)←在 D 上执行查询 Q_i，获取查询结果；
4. 　T←T＋Q_i(D)
5. end for
6. 去掉 T 中重复元组，并按元组返回顺序排序；
7. return T

算法 1 的输入是查询 Q 以及与其相似的历史查询所构成的集合 $QC=(Q_1,Q_2,\cdots,Q_n)$。QC 中的历史查询按照对 Q 的相似度进行排序，也就是说，QC 中的 Q_1 是用户给定的查询，Q_n 是查询历史中与 Q_1 相似度最低的历史查询(但相似度必须高于给定放松阈值)。然后，依次执行 QC 中的查询，获取相应的查询结果(步骤 2－5)。最后，去掉 T 中所有重复元组，并按元组返回的先后顺序对元组进行排序(步骤 6－7)，这样就能确保与当前用户查询最为相关的元组排在前面，因为执行查询的先后顺序是以历史查询与当前查询的相似度为依据的，相似度较高的查询先执行，相似度较低的查询后执行。相应地，返回的查询结果也就是按其对初始查询的满足程度降序排列的。

4　性能实验评价

4.1　实验设置

数据集：从 Amazon 中文图书网站[8]随机抽取出 1720000 条图书信息记录，合成数据集 BookDB(商品名，译著者，出版社，定价，出版时间，ISBN，类别，折扣)，其中商品名、译著者、出版社、ISBN 和类别是分类型属性，定价、出版时间和折扣是数值型属性。

查询历史：邀请 20 个来自不同专业的研究生作为测试者，他们作为不同类型的用户向 BookDB 提交查询，利用这种方式为 BookDB 收集了 1000 条查询作为查询历史。

实验环境：所有的实验都在 P4 3.2GHz 处理器、1GRAM 和内置 160G 硬盘驱动器、运行 Windows XP 和 Microsoft SQL Server 2005 的环境下进行，用 JAVA 和 SQL 语言实现所有的算法，以 JDBC 方式连接 RDBMS，Web 服务器使用 Tomcat5.5。

4.2　语义相似度合理性测试

该实验目的是以用户调查方式来验证第 3.1 节提出的不同查询之间语义相似度评估方法的合理性。邀请了 20 位研究生评价由该方法计算得到的不同查询之间的语义相似度是否合理。要求每位研究生首先从 1000 条查询历史中任选 30 条感兴趣的查询，然后根据本文方法计算得到的查询之间的语义相似度，为其提供与每个查询 Q 相似的 Top-10 个查询(按照相似度降序排列)。注意，这里为 BookDB 中每个属性设定的权重分别为：商品名:0.2;译著者:0.2;出版社:0.1;定价:0.15;ISBN:0.05;出版时间:0.1;类别:0.15;折扣:0.05。测试者可通过执行查询观察相应的查询结果集，在此基础上他们的任务是，从对应于每个查询 Q 的 Top-10 个相似查询列表中选取他们认为与 Q 最相关的查询，并且从以下两个方面说明查询 Q' 与 Q 的相似性：

(1) 查询 Q' 与 Q 中具有相同(或相关)的查询值，即查询 Q' 指定在某个(些)属性上的查询值与查询 Q 在那个(些)属性上指定的查询值相关。例如，商品名"C 语言程序设计"与

"C++语言算法与程序设计"相关，所以在同一属性上指定了相同或相关查询值的查询之间认为是彼此相似的。

(2) 查询Q'与Q的结果相关：虽然查询Q'与Q指定的查询值之间不具相关性，但查询Q'的结果与Q的结果相关。例如，查询"商品名='三国演义'"与"译著者='罗贯中'"的查询结果之间是相关的，但这两个查询本身并不相关。

本文用错误率(Error)量化用户调查结果，错误率是用Top-10个查询中用户认为不相关的查询个数除以10，即

$$\mathrm{Error}(\mathrm{Related}(Q))=\frac{|\mathrm{NotRelevant}(\mathrm{Related}(Q))|}{10}=\frac{10-\mathrm{Relevant}(\mathrm{Related}(Q))}{10} \quad (3)$$

对于用户调查中的所有查询，本文方法的错误率平均低于20%。此外，调查结果还显示，在相关的查询集合中，测试者发现平均有60%的相关查询与相比较的查询之间具有相关的查询值；对于剩余的40%来说，测试者需要执行查询并且观察查询结果才能确定其相关性。表3给出了查询历史中与给定的3个查询之间语义上相似的查询样本集合。我们注意到，虽然在2M的数据集上获得的查询之间的语义相似度低于从4M的数据集上获得的语义相似度，但是两者相差并不大。因此，该算法是稳定的。

表3 相似度评估的鲁棒性

Query Q	Similar queries	Similarity (2M)	Similarity (4M)
商品名=C语言程序设计	商品名=C++语言算法与程序设计	0.82	0.83
	商品名=全国计算机等级考试：二级C语言	0.73	0.76
	商品名=数据结构(C语言版)	0.57	0.6
商品名=三国演义	译著者=罗贯中	0.93	0.97
	商品名=水浒传	0.49	0.53
	商品名=三国志	0.21	0.23
商品名=数据结构∧译著者=严蔚敏	商品名=数据结构∧译著者=吴伟民	1	1
	商品名=数据结构∧译著者=李春葆	0.68	0.71
	商品名=数据结构∧译著者=金远平	0.23	0.24

4.3 不精确查询效果测试

该实验目的是以用户调查方式测试本文提出的不精确查询和结果排序方法(Imprecise Query & Results Ranking, IQRR)的效果。邀请了15位测试者(包括大学教师、公务员、企业管理者、公司职员和学生)，每位测试者根据自己的需求和偏好，对BookDB提出1条测试查询。由于要求测试者从整个数据集中查找所有与查询相关的元组是不现实的，因此本文采用了如下策略：对于BookDB上的每个测试查询Q_i，生成一个相应的数据集H_i，H_i中包含了100条与该查询相关的和不相关的适当元组组合。然后，将所有测试查询和对应的数据集提供给每个测试者，他们的任务是从数据集H_i中标出与测试查询Q_i相关的所有元组和前15个最为相关的元组。在此基础上测试不精确查询和结果排序方法的效果。

(1)不精确查询方法的查全率测试

查全率是指查询结果中的相关元组数与数据集中的相关元组总数之比，表示为

$$\mathrm{Recall}=\frac{\#\ \text{relevant tuples in query results}}{\#\ \text{relevant tuples in dataset}} \quad (5)$$

图2给出了IQRR在不同放松阈值下的查全率。

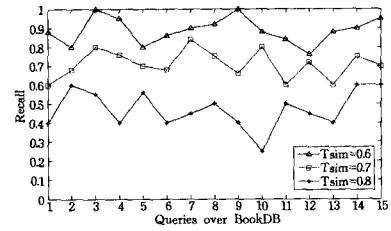

图2 IQRR在不同放松阈值下的查全率

从图2可以看出，一方面放松阈值越低，IQRR的查全率越高；另一方面，当放松阈值接近0.6时，IQRR方法的整体查全率趋于稳定，并且平均查全率达到0.89以上，从而为放松阈值的设定提供了一个参考值。

(2) 排序方法的准确性测试

该实验的目的是为了测试不精确查询下结果排序方法的准确性。沿用上述实验设置，在此基础上考察由本文排序方法返回的前15个元组与测试者标注的15个元组之间的重叠程度，用准确性(Accuracy)评价这种重叠程度，准确性定义为

$$\mathrm{Accuracy}=\frac{|\mathrm{top}-15\ \mathrm{tuples}\cap\mathrm{relevant\ tuples}|}{15} \quad (6)$$

式中，|top−15 tuples∩relevant tuples|代表排序结果的前15个元组与测试者标注的15个元组之间相同的元组数。Accuracy的值介于0到1之间，值越大表明两个集合之间相同的元组数越多，排序方法的准确性就越高，也就是说，用户认为最为相关的结果被排到了前面。本文方法在BookDB数据集上的排序准确性如表4所列。

表4 BookDB上的排序准确性

QueryID	Precision
Q1	0.8
Q2	0.73
Q3	0.67
Q4	0.87
Q5	0.56
Q6	0.67
Q7	0.8
Q8	0.73
Q9	0.87
Q10	1
Q11	0.87
Q12	0.93
Q13	0.67
Q14	0.53
Q15	0.8
Averaged	0.77

实验结果表明，该排序方法在BookDB数据集上的平均排序准确性为0.77，能够较好地满足用户偏好。

结束语 本文提出了一种基于语义相似度的Web数据库不精确查询方法。其主要贡献包括以下方面：(1) 提出了一种基于向量空间模型的查询之间语义相似度评估方法；(2) 提出了不精确查询下的查询结果排序方法。实验结果表明，提出的不同查询之间的语义相似度评估方法性能稳定、评估结果合理，不精确查询方法具有较高的查全率和排序准确性。下一步将开展XML数据的不精确查询方面的研究。

参考文献

[1] Nambiar U, Kambhampati S. Answering imprecise queries over web databases [C]// Proceedings of the 22nd International Conference on Data Engineering. 2006:45-54

[2] Su W, Wang J, Huang Q, et al. Query result ranking over e-commerce web databases [C]// Proceedings of the 15th ACM Conference on Information and Knowledge Management. 2006:575-584

[3] Chen Z Y, Li T. Addressing diverse user preferences in SQL-Query-Result navigation [C]// Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data. 2007:641-652

[4] Bosc P, Hadjali A, Pivert O. Empty versus overabundant answers to flexible relational queries [J]. Fuzzy Sets and Systems, 2008,159(12):1450-1467

[5] Ma Z M, Yan L. Generalization of strategies for fuzzy query translation in classical relational databases [J]. Information and Software Technology, 2007,49(2):172-180

[6] Ma Z M, Meng X F. A knowledge-based approach for answering fuzzy queries over relational databases [C]// Proceedings of the 12th International Conference on Knowledge-based and Intelligent Information and Engineering Systems. 2008,5178:623-630

[7] Zadeh L A. Fuzzy sets. Information and Control [J]. 1965, 8(3):338-353

[8] http://www.amazon.cn

(上接第 144 页)

[6] Curbera F, Duftler M, Khalaf R, et al. Bite: Workflow composition for the Web[A]// Fifth International Conference on Service-oriented Computing-ICSOC 2007[C]. LNCS, 4749. Vienna, Austria: Springer Berlin / Heidelberg, 2007:94-106

[7] Brambilla M, Ceri S, Facca F M, et al. Model-driven design and development of semantic Web service applications [J]. ACM Transactions on Internet Technology, 2007,8(1):3.1-3.31

[8] Bertoli P, Pistore M, Traverso P. Automated composition of Web services via planning in asynchronous domains[J]. Artificial Intelligence, 2010,174(3/4):316-361

[9] Xiong P, Fan Y, Zhou M. A Petri net approach to analysis and composition of Web services [J]. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2010,40(2):376-387

[10] Ozorhan E K, Kuban E K, Cicekli N K. Automated composition of Web services with the abductive event calculus[J]. Information Sciences, 2010,180(19):3589-3613

[11] Murata T. Petri nets: Properties, analysis and applications[J]. Proceedings of the IEEE, 1989,77(4):541-580

[12] Hamadi R, Benatallah B. A Petri net-based model for Web service composition[A]// Proceedings of the 14th Australasian database conference-Volume 17[C]. Adelaide, Australia: Australian Computer Society, Inc., 2003:191-200

[13] 钱柱中,陆桑璐,谢立. 基于 petri 网的 Web 服务自动组合研究[J]. 计算机学报,2006,29(7):1057-1066

[14] Tan W, Fan Y, Zhou M. A Petri net-based method for compatibility analysis and composition of Web services in business process execution language[J]. IEEE Transactions on Automation Science and Engineering, 2009,6(1):94-106

[15] Xia Y, Chen J, Zhou M, et al. A Petri-net-based approach to QoS estimation of Web service choreographies [A]// International Workshops on Advances in Web and Network Technologies, and Information Management [C]. LNCS, 5731. Suzhou, China: Springer Berlin / Heidelberg, 2009:113-124

[16] Wang Y, Lin C, Ungsunan P D, et al. Modeling and survivability analysis of service composition using stochastic Petri nets[J]. The Journal of Supercomputing, 2011,56(1):79-105

[17] Chiola G, Marsan M A, Balbo G, et al. Generalized stochastic Petri nets: A definition at the net level and its implications[J]. IEEE Transactions on Software Engineering, 1993,19(2):89-107

[18] Jensen K. Coloured Petri nets: Basic concepts, analysis methods and practical use[M]. Volume 1, basic concepts. Berlin: Springer-Verlag, 1997:234

[19] 郭玉彬,杜玉越,奚建清. Web 服务组合的有色网模型及运算性质[J]. 计算机学报,2006,29(7):1067-1075

[20] Ratzer A V, Wells L, Lassen H M, et al. CPN tools for editing, simulating, and analysing coloured Petri nets[A]// 24th International Conference on Applications and Theory of Petri Nets (ICATPN 2003) [C]. LNCS, 2679. Eindhoven, The Netherlands: Springer Berlin/Heidelberg, 2003:450-462

[21] Hirel C, Tuffin B, Trivedi K S. SPNP: Stochastic Petri nets. Version 6.0[A]// 11th International Conference on Computer Performance Evaluation Modelling Techniques and Tools [C]// LNCS, 1786. Schaumburg, IL, USA: Springer Berlin/Heidelberg, 2000:354-357

[22] Sherief N H, Abdel-Hamid A A, Mahar K M. Threat-driven modeling framework for secure software using aspect-oriented stochastic Petri nets[A]// The 7th International Conference on Informatics and Systems (INFOS 2010) [C]. Cairo, Egypt: IEEE Computer Society, 2010:1-8

[23] Perez-Palacin D, Merseguer J. Performance evaluation of self-reconfigurable service-oriented software with stochastic Petri nets[J]. Electronic Notes in Theoretical Computer Science, 2010,261:181-201

[24] 林闯,王元卓,杨扬,等. 基于随机 Petri 网的网络可信赖性分析方法研究[J]. 电子学报,2006,34(2):322-332