一种基于 Internet 的 JAR 包使用信息收集方法

邹艳珍 刘昌盛 李 萌 谢 冰
(北京大学信息科学技术学院软件研究所 北京 100871)
(高可信软件技术教育部重点实验室 北京 100871)

摘 要 随着软件复用技术的发展,Internet 上出现了大量可以被利用的软件资源,如 Web Services,JAR 包等。但是,这些软件资源常常缺乏必要的描述信息和使用说明。为此,提出了一种基于 Internet 的 JAR 包使用信息收集方法,以帮助用户检索并整理 Internet 上已经存在的 JAR 包描述信息和用户使用评论,辅助软件复用的成功进行。基于该方法,设计并实现了北京大学软件资源库 JAR 包使用信息收集子系统。该系统目前已经为 6000 余个 JAR 包资源收集、整理了相关的描述和评论信息。

关键词 软件资源,软件资源库,JAR 包,使用信息
中图法分类号 TP311 **文献标识码** A

Internet Based JAR Usage Information Collection Method

ZOU Yan-zhen LIU Chang-sheng LI Meng XIE Bing
(Software Institute, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)
(Key Laboratory of High Confidence Software Technologies, Ministry of Education, Beijing 100871, China)

Abstract With the development of software reuse technologies, more and more reusable software assets emerge on the Internet, such as Web Services, JAR packages and so on. However, necessary description and using help information are often missed in these software assets. Therefore, we proposed an Internet based JAR package usage information collection method, which finds, reorganizes further, the description and the user commented corresponding to a JAR package. Based on this work, we implemented the usage information collection tool for JAR package in PKU Software Asset Repository. It has collected and organized description and user comment information for more than 6000 JAR packages.

Keywords Software asset, Software asset repository, JAR package, Usage information

1 引言

随着软件复用技术的发展,Internet 上出现了大量可以被利用的软件资源,如 Web Services,JAR 包等。但是,一些软件资源本身并没有提供必要的描述信息或使用说明,更没有提供足够的软件质量信息[1-3]。因此,为了帮助复用者准确地把握和理解这些软件资源,需要提供一种软件资源使用信息的搜索方法。一方面,这些使用信息能够帮助复用者有效了解该资源所能完成的功能[4];另一方面,某个软件资源可能已经被复用多次,存在着很多使用评论信息。这些使用评论信息有助于用户进一步了解该资源在可靠性、安全性等多方面的质量,从而为复用者进行复用决策提供重要依据。

目前,在软件资源使用信息收集方面已经存在很多研究工作。譬如,卡内基梅隆大学的 Seacord 等[5]设计了 Agora 系统,它针对 JavaBean, ActiveX 等类型的软件资源分别设计 Agent 来获取和整理信息;维也纳技术大学的 Platzer 等[6]提出了搜索引擎 VSSE4WS,它从 UDDI 站点中自动地获取关于 Web Service 的相关信息,以此来弥补 UDDI 中查询机制的不足;北京大学的李琰等[2]提出了从 Internet 上收集 Web Service 描述信息的方法;韩国项浦大学的 Jinhan 等[7,8]提出了一种将示例程序代码加入 Java 文档的方法,从而使得用户更加方便地阅读使用信息。与此同时,传统搜索引擎也提供了相应的技术支持,如 google code search[9]等。

上述工作为收集基于 Internet 的软件资源使用信息奠定了基础,但仍存在以下不足:首先,通用搜索引擎的返回结果往往是一系列包含相关内容的网页,用户需要从网页中查找所有相关的信息,这样的查询模式会增加用户查询的成本。而且,通用搜索引擎并没有对结果进行文件格式等方面的检查,因此返回的结果有一部分可能并不是用户所期望的信息;其次,现有研究工作缺乏对用户使用报告等这类信息的处理,没有区分使用信息的类型(描述、评论等),不方便用户进行信息浏览、查询和使用。

基于上述目标,本文提出了一种 JAR 包使用信息收集方法。该方法能够检索 Internet 上有效的 JAR 包使用信息,如

到稿日期:2010-07-16 返修日期:2010-09-30 本文受国家高技术研究发展计划(2007AA010301-01),国家重点基础研究计划(2005CB321805),国家创新研究群体科学基金(60821003)资助。

邹艳珍 女,博士,讲师,主要研究领域为软件工程、软件复用技术等,E-mail:zouyz@sei.pku.edu.cn;刘昌盛 男,硕士生,主要研究领域为软件工程;李 萌 男,博士生,主要研究领域为软件工程、软件构件技术等;谢 冰 男,教授,主要研究领域为软件工程、形式化方法等。

JAR包的描述信息和用户评论等，并对这些信息进行了整理和分类，以帮助用户高效地理解、使用一个JAR包类型的软件资源。基于本方法，设计并实现了北京大学软件资源库JAR包使用信息收集子系统，为软件资源库中的6000余个JAR包资源收集、整理了相关的描述和评论信息。

本文第2节介绍了该方法的概览；第3节、第4节和第5节依次给出了该方法的具体设计；第6节对方法中相关算法进行了试验与分析；最后进行了总结和展望。

2 JAR包使用信息收集方法概览

针对基于Internet的JAR包使用信息收集，需要解决3个核心问题，即如何构造查询条件来表征一个JAR包；如何在Internet上搜索相关使用信息；如何将JAR包的相关评论信息和描述信息分离出来。针对这3个问题，本文将基于Internet的JAR包信息收集方法分为3个阶段：特征信息获取阶段、信息收集阶段和数据分析阶段。这3部分之间的关系如图1所示。

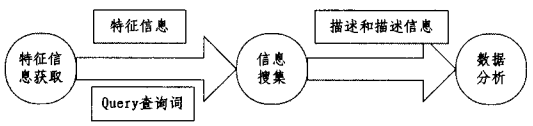

图1 JAR包使用信息收集和整理的3个阶段

特征信息获取阶段的主要工作是抽取JAR包特征信息以及Query构造，并将基于提取的特征信息构造的查询词Query作为下一阶段的输入。一般来说，仅以JAR包资源实体的名字作为搜索引擎的查询词不能保证返回结果均与该JAR包相关。因此，为了从Internet上获得与JAR包相关的描述和评论信息，需要解析JAR包，以得到能够唯一标识该资源的特征信息。

信息搜集阶段的主要工作是从Internet上收集JAR包相关的使用信息，并将收集到的信息作为下一阶段的输入。目前JAR包使用信息收集大体采用两种方式：a) 通用全网搜索引擎。网页搜索引擎是帮助人们从Internet上的海量数据中获取所需信息的一种方便、快捷的途径。借助于网页搜索引擎，需要对得到的相关网页的URL信息进行冗余处理、信息抽取等；b)特定站点检索。在研究过程中，我们了解到一些JAR包发布和维护站点通常提供了某些JAR包的描述信息或用户评论，如Maven[10]，SourceForge[11]，ComponentSource[12]等。这些网站往往有专门的人员进行管理和维护，对网页中的数据进行较为严格的审核，因此信息质量较高。为此，本文从收集的效果以及收集信息所花费的代价等多方面综合考虑，采取了全网搜索引擎与特定站点搜索相结合的JAR包使用信息获取策略，具有较好的互补性。

数据分析阶段的主要工作则是将收集到的JAR包描述和使用评论信息进行分离。实际使用过JAR包编程的程序员都不难发现，官方发布的JAR包信息都偏向于描述其优势方面，而用户很难了解到实际使用过程中的效果，因此用户往往非常需要那些实际使用后的用户反馈、评论信息，比如容错性、算法效率、适用范围等。而上述方法获得JAR包描述和评论信息是杂糅在一起的，不方便用户分类阅读。为此，本文提出了一种基于机器学习的描述和评论信息分离的算法，以帮助复用者更加清晰地理解一个JAR包资源。

在下面的章节中，本文将从这3个阶段出发，分别介绍JAR包使用信息收集方法的详细过程。

3 JAR包特征信息抽取以及Query构造

一般而言，单纯依靠一个JAR包的名字很难获得足够精确的JAR包使用信息。为此，需要抽取更多的JAR包特征来构造查询条件，以提高JAR包使用信息检索的效率。

本文通过解读JAR包中特定文件来构造JAR包特征向量。首先，从META-INF/ MANEFEST.MF的文件(该文件是在JAR文件生成时自动创建的)中得到JAR包的资源发布者、发布版本、发布平台以及入口类等信息；其次，根据Main-Class属性获得JAR文件入口类名，并进一步抽取该类包含的主要接口名称等信息。综合这些信息，本文可将一个JAR包特征向量描述为一个四元组(name, version, mainclass, author)，其中name表示JAR包的名字，version表示版本，mainclass表示入口函数，author表示提供JAR包的个人或组织。

借助于JAR包的特征向量，可以构造出种类多样的查询词，即Query。查询词的构造方法很多，本文的方法是将JAR包的名字搭配其他关键字一起构成查询词，譬如“name+mainclass”构成的依据名字和入口函数的查询、“name+version”构成的依据名字和版本的查询等。

4 基于Internet的JAR包使用信息搜集

如上所述，本文采用全网搜索引擎和针对特定站点抓取相结合的策略来获取JAR包相关的描述和使用评论信息。其中，针对特定站点网页的抓取，本文为每个特定站点设计并实现了爬虫程序，将主机名与查询词一起构成相关网页的URL地址。针对全网范围内的相关网页抓取，本文利用Google搜索引擎提供的API进行网页抓取，并对得到的网页信息进行预处理、网页信息抽取等相关工作，以获得可用的JAR包使用信息。

4.1 相关网页的预处理

首先对Google API返回的结果网页进行预处理，其目的是去除返回结果中没有价值的信息，并且缩小待处理目标集合，提高方案的效率。预处理的主要工作包括无效URL处理、冗余信息处理。

1)无效URL处理：通过采用不同的策略构造的查询词得到的相关网页的URL中，有相当一部分不是用户期望得到的结果。这些网页不包含任何实际的内容，仅仅提供了JAR包实体的下载链接或者镜像文件。对于这样的网页，需要从返回结果的URL中移除。

2)利用Web搜索引擎进行信息检索时，往往会返回大量的近似镜像网页，导致信息冗余。在本文的实现中采用MD5算法[13]作为冗余信息检测的方法。MD5算法成熟、易于实现，程序的可移植性强，消除冗余信息效果比较好，与现有算法相比具有较小的时间、空间复杂度和较大的实用价值。

4.2 网页信息的抽取

针对获得的大量与JAR包相关的网页资源，还需要对这些网页进行信息抽取，以得到JAR包相关的使用信息。

一般来说，通过普通的深度优先遍历就可以得到网页资源各个标签节点，这些节点包含的文字将成为网页内容提取

的候选文本。注意到已经获得了 JAR 的特征向量，因此 JAR 包相关使用信息提取可转化为网页中提取的候选文本与 JAR 自身特征向量的文本相似度问题。

针对文本相似度问题，目前研究人员提出了基于 VSM 的文本相似度计算方法[14]、基于词条共现的文本相似度计算方法[15]等。VSM 模型是一种基于统计的文本表示模型，没有考虑文本上下文之间的语义关系，分类精度不高；词条共现模型是一种以统计为基础的自然语言处理模型，若干个词共同出现在文本的同一窗口单元中（如一句话、一个自然段或者是已经定义的一个窗口等）则认为这若干个词在意义上是相互关联的。共现的概率越高，其相互关联越紧密。基于词条共现的文本相似度计算方法既考虑到 VSM 模型的简单、易用以及高效的特点，又结合了词条共现对 VSM 模型准确度稍低的弥补作用。为此，本文使用了基于词条共现的文本相似度计算方法，具体方法是：计算每一段候选文本的文本特征向量，然后采用基于词条共现的文本相似度计算方法，通过计算它与 JAR 包自身特征信息文本的相似性，挑选出相似度最大的一段候选文本作为结果进行保存。

为了去除候选文本中的一些无效信息，降低这些无效信息对相似度的影响，本文还采取了以下两个处理：

1）移除停用词：停用词（Stop Words），即搜索引擎在索引页面或处理搜索请求时自动忽略的某些词，通常意义上是指出现频率很高但是又没有太大实际意义的词，主要包括了语气助词、副词、介词等，它们降低搜索的效率，因此在搜索引擎的具体实现中会忽略这些特定的常用词。在本系统中我们构建了一个包含 42 个词条的停用词表。

2）大小写折叠与词根化（Stemming）：大小写折叠就是将文本中的所有字母均转化为统一的小写形式或者统一的大写形式。词根化就是将每一个单词还原为该词的原型，例如“depressed”，“depressive”词根化的结果为“depression”。词根化的过程包含了去掉一个或者几个后缀而将其压缩为词根形式，将其转化为中性术语。

5 JAR 包描述和评论信息的分离

上述方法获取的 JAR 包使用信息包含了 JAR 包相关描述信息和用户评论信息。其中描述信息大多用于介绍 JAR 包的使用方法或功能，用户评论信息则主要涉及 JAR 包的可用性等质量因素。在软件复用过程中，用户会基于不同的目标在不同时刻关注不同的使用信息。因此，需要提供 JAR 包使用信息的分类机制，即分离描述和使用评论信息的分离器，以方便用户有针对性地了解 JAR 包资源。

本文在研究中发现，JAR 包的描述信息通常是一段“客观”文本，而 JAR 包的用户使用评论信息则相对是一段相当“主观”文本。主要体现在：a）一些特定网站上的用户评论往往是量化的（几颗星等）；b）相当多的用户评论包含带有明显感情色彩的词汇，如“good”，“bug”等。为此，本文将 JAR 包使用信息分类看作是一个将主观文本与客观文本分离的问题。

主观文本与客观文本分离是自然语言处理领域中的一个研究课题，不少研究者提出了行之有效的方法[16-18]。目前，机器学习是文本分类的主流技术，主要包含如下几个部分：分类器的选择、训练样本的准备以及特征提取。在分类器方面，支持向量机（SVM）方法针对小样本数据学习具有很强的优势，其最终决策仅由少数的支持向量确定，计算的复杂性取决于支持向量的数目，而不是整个样本空间，有效地避免了“维数灾难”。因此，本文采用了支持向量机来作为系统的分类器，尝试了用一种基于机器学习的方法将 JAR 包相关的描述和使用评论信息进行分离。

如图 2 所示，描述和评论信息分离机制包含两个核心部分：训练模块和分类模块。训练模块的最终目的在于产生一个 SVM 分类器，作为分类模块中的分类依据。本文选择常用的 LibSVM 分类器，它简单易用，提供了很多默认的参数设置。利用这些默认的参数就可以解决常见的分类问题。

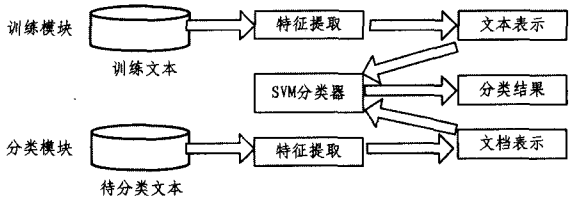

图 2 JAR 包描述和评论信息分离示意图

特征的选取是分类准确性的关键，在针对搜集到的描述信息和评论信息进行研究之后，提出了有助于区分描述和评论信息的 3 个特征：

1）信息的长度。通过观察，我们搜集到的评论信息长度一般较短，即使用者对资源的反馈一般简明扼要，而描述信息一般比较全面，内容较长。

2）评论信息中存在许多第一人称代词，因此整理了相应的代词表，例如“I”，“My”等，将信息中命中这个代词表中的词汇的个数作为第 2 个特征。

3）评论中含有许多主观情感词汇，例如“awesome”，“good”，“easy”等，也经常包含一些表达特定情感的符号，例如“^-^”，“:-)”，“:)”等。因此我们建立了主观词情感列表，将命中这个列表中词汇的个数作为第 3 个特征。

这 3 个特征对于分类效果的影响，本文将在实验阶段进行验证。在今后进一步工作中，还将进一步研究并提取其他的更有代表性的表征文本类别的特征信息，从而进一步提高分类的准确率。

6 验证与分析

基于上述方法，本文在北京大学软件资源库[19]中设计并实现了基于 Internet 的 JAR 包使用信息收集工具，为 JAR 包收集并整理相关的描述和使用评论信息。

6.1 JAR 使用信息收集结果

目前，北京大学软件资源库中约有 6000 多个 JAR 包资源实体。通过运行 JAR 包使用信息收集工具，本文共为资源库中近 4000 个 JAR 包资源搜集到了相关使用信息，约占资源库 JAR 包总数的 2/3。平均每个 JAR 包资源收集到了 7 条使用信息，其中每条使用信息包括其内容摘要、来源（URL 地址）、收集时间等，用户可以根据 URL 地址等获得详细的 JAR 包资源描述或者使用评论信息。

图 3 描述了该工具从 SourceForge 网站上搜集到的关于 JAR 包 TightVNC 的使用信息。其中第 1 条描述信息详细介绍了该 JAR 包资源由来、所实现的主要功能以及典型的应用环境等信息；后面 4 条用户评论信息介绍了用户使用该 JAR

包之后的感受，包含 2 条正面评论和 2 条负面评论。

内容摘要	来源 URL	搜集时间
TightVNC is an improved version of VNC, great free remote-desktop tool. The improvements include bandwidth-friendly "Tight" encoding, file transfers in the Windows version, enhanced GUI, many bugfixes, and more. (more)	http: // sourceforge. net/projects/vnc-tight/	2010-10-12 12:10:36
Version 2.0 is easy to install, stable, supports fast user switching in Windows and works great as a service in Windows Vista (more)	http: // sourceforge. net/projects/vnc-tight/reviews/	2010-10-9 12:10:36
Doesnt support cryptografy.	http: // sourceforge. net/projects/vnc-tight/reviews/	2010-10-9 12:10:36
new version server opens 2 prosesses. what you are Releasing is unusable buggy shit (more)	http: // sourceforge. net/projects/vnc-tight/reviews/	2010-10-9 12:10:36
Quick and easy on both Win 2000 and XP. Used the DfMirage driver for performance, so next to no impact. (more)	http: // sourceforge. net/projects/vnc-tight/reviews/	2010-10-9 12:10:36
...	...	...

图 3　JAR 包 TightVNC 的描述和用户评论信息示例

注意在上述结果中，有约 1/3 的 JAR 包没有找到其对应的描述或者使用评论信息。通过研究分析这些 JAR 包，我们发现未能找到相关信息可以归结为以下几个原因：首先，有些 JAR 的应用领域相当狭窄，复用该 JAR 包的用户数比较少，导致 Internet 上与该资源相关的讨论较少；另外，虽然有些 JAR 包被一定数量的用户复用，但是这些用户在复用过程中或复用结束之后并未发表自身的使用体验，这也导致了不能收集到资源相关的描述和使用评论信息。

6.2　描述和评论信息分离的效果

为了方便用户进行阅读和理解，本文对收集到的 JAR 包描述信息和用户评论信息进行了分类。为了验证分离算法的效果，本文从系统搜集到的使用信息中随机提取 150 条评论信息和 150 条描述信息作为试验数据集，并使用 3-交叉验证(Cross Validation)得到稳定、可靠的分类模型。

在实验过程中，将数据集分成均等的 3 份，轮流交替地将其中的两份作为训练数据，剩下的一份作为测试数据，3 次结果的均值作为对算法精度的估计。交叉检验能够摆脱数据稀疏性带来的误差，力求最终结果的准确性和可靠性。实验采用准确率、召回率以及 F-Score 作为评价分类性能的主要指标，其计算公式如式(1)－式(3)所示。其中对于给定的某个类别，变量 A 表示被正确分到该类的实例个数，变量 B 表示被误分到该类的实例个数，变量 C 表示属于该类但被误分到其它类别的实例的个数，参数 β 用来为准确率(Precision)和召回率(Recall)赋予不同的权重。当 β 取值 1 时，准确率和召回率被赋予相同的权重。

$$Recall=A/(A+C) \tag{1}$$

$$Precision=A/(A+B) \tag{2}$$

$$F\text{-}Score=((\beta^2+1)*P*R)/(\beta^2*P+R) \tag{3}$$

不同的核函数可能对分类模型产生不同的影响。为了选取较优的核函数，本文还通过对不同的核函数作用下的分类结果进行了比较。如表 1 所列，针对上述实验数据，使用线性核函数的分类器能够获得 94%的准确率和召回率，从而得到较优的分类模型。而同样条件下，使用 RBF 核函数的分类器只获得了 86.2%的准确率和 83.3%的召回率，使用 Sigmoid 核函数的分类器获得的准确率、召回率和 F-Score 都更低。注意，由于准确率和召回率都是系统关注的指标，因此本文中取 $\beta=1$。

表 1　不同核函数下的信息分离结果

核函数	准确率	召回率	F-Score
线性核函数	94%	94%	94%
RBF 核函数	86.2%	83.3%	84.7%
Sigmoid 核函数	34.4%	40%	34.1%

描述和评论信息分离的效果还受到文本特征提取的影响。如前所述，本文在对搜集到的描述信息和评论信息进行研究之后，提出了有助于区分描述和评论信息的 3 个特征：信息的长度、代词表和主观词汇表。为了观察这些特征的影响，还进行了如下实验：针对上述同样的评论信息和描述信息，在分类过程中选取特征时轮流去掉所述 3 个特征中的 1 个，将剩下的 2 个特征作为分类模块的特征输入，从而得到了不同的分类效果。如图 4 所示，去掉“信息的长度”这个特征后，分类算法的准确率降低到了 75.1%，召回率和 F-Score 也分别降低到了 74.7%和 74.6%。同样条件下，尽管去掉情感特征词的影响比去掉人称代词的影响要小(去掉情感词特征的准确率为 87.1%，而去掉人称代词特征的准确率为 81.8%)，但是每个特征对准确率和召回率都是至关重要的。

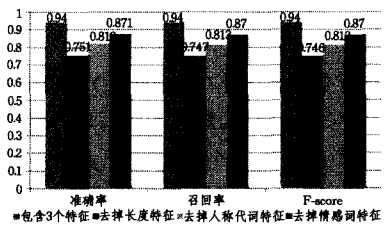

图 4　不同特征对分类效果的影响

从上述实验结果不难发现，去掉长度特征时分类的效果显著降低。这是因为在研究中发现，评论信息的长度一般较短，而描述信息通常全面而广泛、内容较长，所以长度成为区分 JAR 包描述和评论信息的一个重要特征；其次，人称代词对分类效果的影响也比较大。这是因为在客观的描述信息中很少出现“I”，“My”这类词汇；相对而言，情感词对分类效果的影响最低。这里面可能有两个原因：其一是很多带有情感特征的词汇，例如“excellent”也可以出现在描述信息中，故区分度不高；其二是我们的实验数据的数量还相对较少，从中学习到的情感词汇也比较少，从而降低了该特征对分类算法的影响。

结束语　收集软件资源使用信息是用户在软件复用过程中非常关注的一个问题。本文提出了基于 Internet 的 JAR 描述和使用信息收集方法，并实现了 JAR 包描述和使用评论信息的分离。在下一步工作中，将进一步探讨如何改进 JAR 包相关使用信息的文本特征抽取方法，以提高使用信息收集的准确率；其次，将尝试从 JAR 包的评论信息中挖掘出更多的规律，以进一步提高资源描述和使用评论信息分离方法的效率。

(下转第 179 页)

需要一个或多个输入和输出。从输入到输出，服务具有处理请求并返回答复的机制。如网络服务包含了进程、序列、流和链接。业务逻辑的语义可以用词组结构表达，同样地 FOIL 的语句也采用词组结构作谓语，这样保证了业务逻辑语义的真值。然而 FOIL 无法表达 WS-BPEL 具有的并发、遍历和外部链接等结构。为了解决这个问题，本文引入 FOL 的命题(Proposition)符号来支持服务的并发、遍历和外部链接语义，如下所示：

for all x in T, $\langle x, P(Q(x))\rangle = f \rightarrow$ iff P 与 $Q(x)$ 链接(外部链接 link)且在 $Q(x)$ 完成后顺序完成(sequence)，f 在 T 业务世界内为真。

for all x, y in T, $\langle x, y, P(x) \wedge Q(y)\rangle = f \rightarrow$ iff $P(x)$ 和 $Q(x)$ 并发(并发 flow)，f 在 T 内为真。

for all x in T, $\langle x, (P(\forall x))\rangle = f \rightarrow$ iff 对于变量 x，$P(x)$ 都具有 f(遍历 foreach)，f 在 T 内为真。

根据上述命题，FOIL 是能够满足网络服务的要求的。

4 自动化从 FOIL 语句到服务编排

由于 FOIL 语句的材料充分和形式正确，再加入了命题符号辅助表达并发、遍历和外部链接，一段短的伪码就能实现从 FOIL 语句到基于 XML 结构编排网络服务的 WS-BPEL 结构，这样实现了动态绑定，也就实现了服务编排的自动化。

```
Read a formula;
If (P(x) ∧Q(y))    //并发和顺序
Then ⟨Sequence⟩⟨flow P⟩⟨flow Q⟩⟨/Sequence⟩
If (P(Q(x)))    //链接和顺序
Then ⟨PartnerLink P⟩⟨PartnerLink Q⟩
If (P(for all x))    //遍历
Then ⟨foreach (p(x))⟩
```

结束语 FOIL 语句作为业务逻辑的元语言，证明了是材料充分和形式正确的。引入命题符号后，FOIL 就能够表达 WS-BPEL 的结构，满足服务编排的需求。给出的伪码显示了从业务逻辑到服务编排自动化的可能性，避免了书写 WS-BPEL 的繁琐，还容易维护。动态绑定服务组件是服务编排自动化的实现。

参 考 文 献

[1] Fitting M. First Order Intensional Language[J]. Annals of Pure and Applied Logic, 2004, 127(1-3): 171-193

[2] Web Services Business Process Execution LanguageVersion 2.0, OASIS Standard[EB/OL]. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html, 2007-04-11

[3] Hodges W. Tarski's truth definitions. In the Stanford Encyclopedia of Philosophy[EB/OL]. http://plato.stanford.edu/entries/tarski-truth/Fenstermacher, 2001

[4] Josuttis N M. SOA in Practice: The Art of Distributed System Design (Theory in Practice)(1 edition)[M]. O'Reilly Media, August 24, 2007

[5] Magnus P D. Forall x: an introduction to formal logic[M]. version 1.24 [080109], University at Albany, State University of New York (tutorial)

(上接第 164 页)

参 考 文 献

[1] Hummel O, Atkinson C. Using the Web as a Reuse Repository [C]// Proceedings of the International Conference on Software Reuse (ICSR-9). 2006: 298-311

[2] Li Yan, Liu Yao, Zhang Liang-jie, et al. An Explortary Study of Web Services on the Internet[C]// Proceedings of the International Conference on Web Services (ICWS). 2007: 380-387

[3] Fan J, Kambhampati S. A Snapshot of Public Web Services[J]. ACM SIGMOD Record, 2005, 34(1): 24-32

[4] Dekel U, Herbsleb J D. Improving API documentation usability with knowledge pushing [C]// Proceedings of IEEE 31st International Conference on Software Engineering. 2009: 320-330

[5] Seacord R C, Hissam S A, Wallnau K C. AGORA: a Search Engine for Software Components [J]. IEEE Internet Computing, 1998, 2(6): 62-70

[6] Platzer C, Dusdar S. A Vector Space Search Engine for Web Service[C]// Proceedings of the Third European Conference on Web Services (ECOWS). 2005: 62-71

[7] Kim Jinhan, Lee Sanghoon, Hwang Seung-won, et al. Towards an Intelligent Code Search Engine [C]// Proceedings of Twenty-Fourth Conference on Artificial Intelligence (AAAI-10). Atlanta, Georgia, USA, July 2010: 1358-1363

[8] Kim Jinhan, Lee Sanghoon, Hwang Seung-won, et al. Adding Examples into Java Documents[C]// Proceedings of the 24th IEEE/ACM International Conference on Automated Software Engineering (ASE 2009). Nov. 2009: 540-544

[9] http://www.google.com/codesearch

[10] http://maven.apache.org/

[11] http://www.sourceforge.net/

[12] http://www.componentsource.com/

[13] RFC 1321, R. L. Rivest. The MD5 Message Digest Algorithm [S]

[14] 郭庆琳，李艳梅，唐琦. 基于 VSM 的文本相似度计算的研究. 计算机应用研究[J]. 2008, 25(11): 3256-3258

[15] 曹恬，周立，张国煊. 一种基于词共现的文本相似度计算[J]. 计算机工程与科学, 2007, 29(3): 52-73

[16] Pang Bo, Lee Lillian. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimun Cuts [C]// Proceedings of the ACL. 2004: 271-278

[17] Hatzivassiloglou V, McKeown K R. Predicting the semantic orientation of adjectives [C]// Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics. 1997: 174-181

[18] Wiebe J, Bruce R, Bell M. Learning subjective language[J]. Computational Linguistics, 2004, 30(3): 277-308

[19] TSR[EB/OL]. http://tsr.trustie.net