一种基于决策树的构件查询方法*

李 戈+, 张 路, 谢 冰, 邵维忠

(北京大学 信息科学与技术学院 软件研究所,北京 100871)

A Decision Tree Based Component Retrieving Method

LI Ge+, ZHANG Lu, XIE Bing, SHAO Wei-Zhong

(Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

+ Corresponding author: Phn: +86-10-62751794, E-mail: lig@sei.pku.edu.cn

Received 2004-01-16; Accepted 2004-03-29

Li G, Zhang L, Xie B, Shao WZ. A decision tree based component retrieving method. *Journal of Software*, 2004,15(Suppl.):45~52.

Abstract: Reusing the components from the component repository is a way for software reuse. However, finding appropriate software components is often challenging. In order to help the users to find their components, the decision tree method is introduced to component repository to help inexperienced users' retrieving. Based on this standpoint, the process of retrieving component from component library is analyzed at first; secondly, the solutions about how to build the decision tree, how to retrieve components from the decision tree and how to save the decision rules are discussed; and then the method of decision tree based component retrieving is presented. Further more, a prototype decision tree based component retrieval system is introduced, and a preliminary experiment is presented to illustrate the method in the end of this paper.

Key words: CBSD; software reuse; component library; decision support; component retrieving

摘 要: 在基于构件的软件开发中,构件查询是影响软件开发效率的关键性问题.为了降低复用者的查询难度,提高构件查询效率,提出将决策树方法引入构件查询过程中,从软件开发项目的构件查询历史信息中获取决策信息,辅助构件查询.在对传统构件查询过程进行分析的基础上,首先对决策树方法与构件库查询过程结合的需求进行了探讨,然后针对如何利用构件查询历史信息建立决策树,如何利用决策树进行构件查询,以及如何对决策树规则集进行存储等关键问题给出了解决方案,提出了一种基于决策树的构件查询方法.还对基于该方法搭建的构件查询系统的基本功能进行了介绍,并给出了该系统的一个应用实例.

关键词: CBSD;软件复用;构件库;决策树;构件查询方法

* Supported by the National Natural Science Foundation of China under Grant Nos.60073015 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2001AA113070 (国家高技术研究发展计划(863)); the National Grand Fundamental Research 973 Program of China under Grant No.2002CB31200003 (国家重点基础研究发展规划(973))

作者简介: 李戈(1977－)男,山东淄博人,博士生,主要研究领域为软件复用,软件工程;张路(1973－),男,博士,主要研究领域为软件工程,程序理解,软件度量;谢冰(1970－),男,博士,主要研究领域为软件工程,形式化方法,分布式系统;邵维忠(1946－),男,教授,博士生导师,主要研究领域为软件工程及环境,面向对象方法,操作系统.

软件复用是实现软件工业化生产,提高软件生产率和生产质量的一条现实可行的途径[1].基于构件的软件开发(CBSD)是软件复用的主要方式之一.基于构件的软件开发过程包括构件生产、构件管理和构件组装三个基本活动[2].构件管理是连接构件生产和构件组装的重要环节.构件库(component library,简称CL)是进行构件管理的基础设施,它的主要目标是解决大量软件构件的存储与查询问题[3].随着构件化软件生产的迅速发展,构件库中存储的可复用构件的数量不断增加,构件库的分类模式和查询方式也变得越来越复杂,从构件库中查询构件的难度也不断增大[4].这是因为构件的查询环境与多种因素相关,例如:构件库对构件的描述方式、分类模式;构件复用者对构件模型和分类模式的理解;复用者对构件查询条件的描述方式等等.构件数量的增加使这些环节的复杂性不断提高,给复用者的构件查询带来了困难.

本文提出了一种基于决策树的构件查询方法,从构件库的查询历史信息中提取决策规则,辅助复用者进行构件查询.本文首先对传统构件查询过程进行分析,然后对决策树方法与构件库查询系统的结合进行了讨论,针对如何利用构件查询历史信息建立决策树,如何利用决策树进行构件查询,以及如何对决策树规则集进行存储等关键问题提出了解决方案.

1 构件查询过程分析

在软件构件库中,对构件某个方面显著特性的抽象称为特征(feature)[5];对构件的某个特征所进行的描述称为特征值(feature value);在构件查询过程中,对所要查找的目标构件进行描述的一系列特征值的集合称为查询需求特征集(component requirement features);复用者通过一个查询过程最终确定的构件集合称为构件查询结果集(final target components).一般构件的查询过程,是一个从查询需求特征集到构件查询结果集的映射过程.这个过程可以分解为两个不断循环的步骤,其中每一步的进行都要求有来自构件库的相关知识的辅助,我们将这些知识称为查询辅助知识(component knowledge),如图1所示.

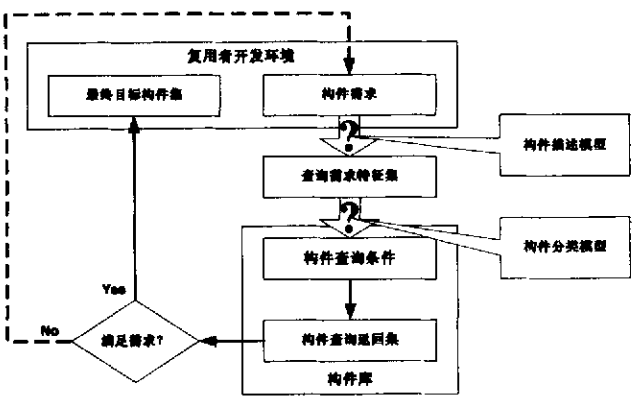

图1 构件查询过程分析

第1步,描述构件特征:复用者将构件需求描述为查询需求特征集.这个过程要求复用者用一系列的特征值对所需要的构件的特征进行描述,这些特征值的集合形成了本次查询的查询需求特征集.在这个步骤中,复用者所需要的查询辅助知识是构件库所采用的构件描述模型.构件描述模型(component description model)是构件库对构件进行存储所采用的描述模型.构件描述模型知识能够帮助复用者更准确的向构件库传达所需要构件的特征描述,能够帮助复用者将构件查询需求转换成相对应的构件查询条件.

第2步,组织查询条件:复用者将查询需求特征集组织成构件查询条件.所谓构件查询条件(component query condition)是指复用者按照构件库查询规范对查询需求的表达.在一般构件库中查询规范是依赖于构件分类模型(taxonomy model)制定的,构件分类模型是指构件库对构件进行分类存储时所采取的分类方式和规范[6-8].在这个步骤中,构件分类模型知识能够辅助复用者将一系列的特征值组织成符合查询规范的查询条件.

构件库根据复用者所提交的构件查询条件对构件库进行检索,并将查询结果返回给复用者,我们把由构件库直接返回给复用者的构件查询结果称为构件查询返回集(query result set),而把复用者最终确定的构件集合称为构件查询结果集.若复用者在第2步获得的构件查询返回集过大,则说明复用者所提供的查询约束,不能将查询返回集约束到足够小的范围内,则需要复用者回到构件查询的第1步,在前一次查询的基础上再次进行上述两步查询,直到得到满足复用者需求的构件查询结果集.若该构件查询结果集能够满足复用者的构件需求,则该查询过程为成功查询.

从以上分析可以看出,为了顺利完成构件查询,复用者不但要具备构件描述模型和构件库分类模式的知识,而且要能够利用这些知识对目标构件和查询条件进行合理的描述.然而对无经验的复用者而言,了解这些纷繁复杂的描述模型和分类模式知识是一项令人厌倦的工作,因为这些信息与复用者所进行的实际软件开发需求没有直接的联系[4].所以随着构件库中构件数量的增加,分类模式多样化,构件描述模型的复杂化,要找到合适的构件就越困难.然而,在相同或相近的软件开发项目中,由于软件开发环境和软件需求的相似性,复用者会产生相同或相似的构件查询需求,这时构件查询结果集表现出较大的相似性.所以在这些开发项目中可以利用构件查询历史帮助没有经验的复用者进行构件查询.因此,我们把决策树方法引入到查询系统中,把复用者在构件库中选择构件的过程看作一个决策过程,把要查找的构件看作决策的目标,利用构件查询过程的相似性,通过决策树方法挖掘出构件查询历史数据集中已有构件的决策规律,在决策树工具的辅助下,帮助复用者利用这些决策规律进行构件选择,以降低构件查询的难度,提高构件查询效率.

2 构件查询相关定义

定义 1. 设构件库中全体构件的集合为 C,构件数目为 n_{comp},设构件库中任意构件为 c_i,则构件库可以表示为 $C=\{c_i|i\in\{1,2,3,...,n_{comp}\}\}$.

定义 2. 设构件库中构件描述模型所使用的全部特征的集合为 F,其中所含特征的数目为 $n_{feature}$,设任意用来描述构件的特征为 f_j,则全部构件特征集合可以表示为: $F=\{f_j|j\in\{1,2,3,...,n_{feature}\}\}$.

定义 3. 设任意构件特征 f_j 可以有 m_j 种特征值,f_j 的任意特征值为 $v(f_i)$且 $v(f_i)=v_{jk}$,则任意特征 f_j 的特征值集合为 $Value(f_j)=\{v_{jk}|k\in\{1,2,3,...,m_j\}\}$.

定义 4. 在一次成功的构件查询过程中,设复用者向构件库提交的查询需求特征集为 $T_{query}=\{v(f_j)|\forall f_j\in F_{query}\}$,其中 $F_{query}=\{f_j|j\in\{1,2,3,...,n_{query}\}\}$,$F_{query}$ 称为构件查询特征集,是 T_{query} 中所描述的构件特征 f_j 的集合,n_{query} 为 F_{query} 中所包含的构件特征的数目;设 $C_{result}=\{c_i|i\in\{1,2,3,...,n_{result}\}\}$ 为该次查询的构件查询结果集,其中 n_{result} 为构件查询结果集中包含的构件数;则一次成功的查询过程可以表示成 $Q=(f(T_{query})\rightarrow C_{result})$,其中 $f(\ )$ 是从查询需求特征集到构件查询结果集的映射.

复用者在构件库中选择构件的过程是一个决策过程,对各个决策结果进行选择所需要考虑的因素称为决策因素,T_{query} 是复用者所考虑的决策因素的集合,C_{result} 中的构件是决策的结果.在相同或相近的软件开发项目中,同一构件可能出现在对应不同 T_{query} 的 C_{result} 中,即同一构件可以与不同的决策因素相关,不同的决策因素能够产生相同的构件选择.决策树能够通过对构件查询历史信息的学习,在 T_{query} 与 C_{result} 的对应关系中获得 $C_{result}^{\cdot}$ 中构件的决策规则[9],并能够支持复用者使用这些规则进行构件查询.

3 基于决策树的构件查询

决策树是一种支持数据分类的决策支持工具,它是一种用于数据分类的树型数据结构.决策树的建立通过两个步骤完成:首先建立一个描述已知决策因素与决策结果对应关系的数据集模型.该模型中的每一条记录由多个决策因素和一个决策结果组成,表示了它们之间的对应关系,该模型称为训练数据集.然后,使用该模型对决策树进行训练,使训练集中的各个决策因素分布到树型结构的各个中间节点中,各个决策结果分布到叶结点中,形成由根结点到叶节点的决策规则的集合.上述过程与构件查询的结合讨论如下:

3.1 构件查询训练数据集的组织

定义 5. 构件复用者所进行的任意构件查询 Q 的历史记录记为 $h(Q)=\{h_q|q\in\{1,2,3,...,s_{query}\}\}$,其中 $h_q=\{T_{query}\rightarrow c_i|c_i\in C_{result}\}$,称为构件查询记录,$s_{query}$ 为记录 Q 所使用的构件查询记录 h_q 的数目.

定义 6. 设具有相近的软件开发环境和软件需求的开发项目的集合为 $P=\{p_i|i\in\{1,2,3,...,n_{project}\}\}$,称为项目集合,其中 $n_{project}$ 为 P 中所含软件项目的个数,则由项目集合 P 中所有成功查询记录 h_q 所组成的数据集合称

为面向项目的构件查询历史数据集(project oriented history),记为 $H_{POH}=\{h_l|l\in\{1,2,3,...,s\}\}$,其中 s 为 H_{POH} 所含成功查询记录的数目.

如上所述,对于软件项目集合 P 的构件查询历史数据集 H_{POH},复用者向构件库提交的 T_{query} 中包含了大量对相同构件特征的描述,即数据集 H_{POH} 中 h_l 所包含的 F_{query} 具有较大的相似性,而对于某些构件特征采用了相同的特征值进行描述.基于这种相似性,我们从数据集 H_{POH} 所包含的 F_{query} 中选取出部分属性,针对项目集合 P,建立一个用于构件需求描述的构件特征集合,称为构件需求特征模型.

定义 7. 设历史数据集 H_{POH} 中,h_l 所包含的所有构件查询特征集 F_{query} 的合集为 F_{POH},则 $\forall F_{query}\in F_{POH}$;从 F_{POH} 中选取部分构件特征组成 F_{RFM},则 $F_{RFM}=\{f_j|j\in\{1,2,3,...,n_{rfm}\}\}$,且有 $F_{RFM}\in F_{POH}$.根据软件开发环境和软件需求的近似性,为 F_{RFM} 中的相应属性指定默认查询特征值,特征值不能确定的构件特征标定为'*ND*'(not ensured).

参照构件需求特征模型,将每个构件查询记录 h_l 中的 T_{query} 所包含的构件特征值添加到构件需求特征模型中,覆盖相应的默认值,得到符合构件需求特征模型的查询特征值集 T'_{query},则与构件查询历史数据集 H_{POH} 对应的面向项目集合 P 的决策树工具训练数据集(project oriented training data)可以构造为 $D_{POT}=\{d_p|p\in\{1,2,3,...,s_{POT}\}\}$,其中 s_{POT} 为 D_{POT} 中所含样本的数目,$d=\{T'_{query}\rightarrow c_i|\forall c_i\in C\}$.

3.2 基于决策树的构件查询

3.2.1 构件查询决策树的建立

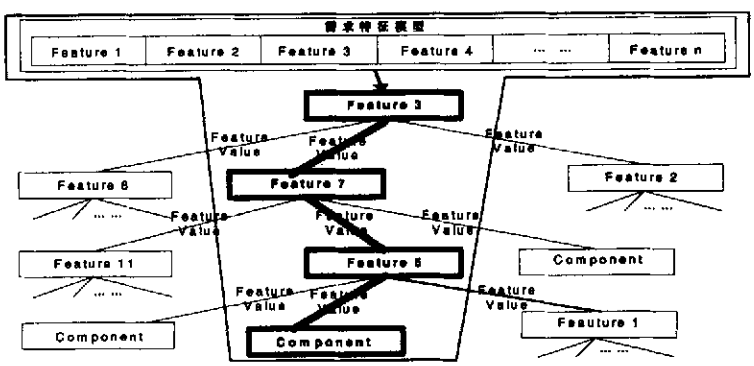

图 2 决策树训练结果示意图

利用训练数据集对决策树工具进行训练,所得到的训练结果是一棵包含着训练数据集中各个构件的分类信息的决策树 $DT(D_{POT})$.(决策树的训练过程不作为本文讨论的重点,相关文献可以参考文献[9,10]) 决策树 $DT(D_{POT})$ 的每个中间节点由特征需求模型 F_{RFM} 中的一个特征 f_i 进行标示,中间节点的各个分支分别标示了这个特征的各个取值,$DT(D_{POT})$ 的叶节点则标示着属于该叶节点的构件.训练后决策树 $DT(D_{POT})$ 的一棵子树如图 2 所示.该决策树中每条从根节点到叶节点的路径,由相应中间节点所标示的构件特征 f_j 及特征值 v_{jk},组成了叶节点所标示的构件 c_i 的决策规则 $r(c_i)$.所谓决策规则是指构件特征 f_j 的取值 v_{jk} 与构件 c_i 之间的判别关系.$r(c_i)$ 表示如下:

$$r(c_i)=\left\{rule\rightarrow c_i\middle|rule=\left\{v_{jk}\middle|j\in\{1,2,3,...,t\},k\in\{1,2,3,...,m_j\}\right\},v_{jk}\in Value(f_j),f_j\in F_{RFM},c_i\in C\right\}$$

由于训练数据集 D_{POT} 是由面向软件项目的构件查询历史数据集构造而来,$r(c_i)$ 是查询结果集中包含构件 c_i 的成功构件查询记录的归纳[9],因此 $r(c_i)$ 表示了训练数据集 D_{POT} 中所包含的构件 c_i 的决策规律.

3.2.2 基于决策树的构件查询算法

如图 2 所示,决策树训练完成时,需求特征模型中的各个特征被分布到决策树的各个中间节点中.各构件特征在决策树中的位置,是由该特征在训练数据集中所提供的分类信息的信息增益[9]所决定的.信息增益是决策树对各个决策因素进行树型排列的依据,越接近根节点的决策因素所提供的信息增益越大,即该决策因素能够对决策结果的确定产生最大的影响.对于由训练数据集 D_{POT} 获得的决策树 $DT(D_{POT})$ 而言,在复用者所提供的构件需求特征模型 F_{RFM} 中,较接近根节点的构件特征 f_j 的特征值选择,能够为构件查询结果 C_{result} 的选择提供最大的信息增益.所以,当复用者需要从构件库中获取可复用构件时,可以按如下算法进行:

设决策树 $DT(D_{POT})$ 的根节点为 *root*;设任意中间节点为 $node_i$,由 $node_i$ 所标示的构件特征为 $feature(node_i)$,

则 $feature(node_i)\in F$,且 $v(feature(node_i))\in T_{query}$;设任意叶结点为 $leaf_i$,由 $leaf_i$ 所标示的构件为 $comp(leaf_i)$,则 $comp(leaf_i)\in C$;设全部叶节点的集合为 L,全部叶结点的数目为 n_{leaf},则 $L(DT(D_{POT}))=\{leaf_i|i\in\{1,2,3,...,n_{leaf}\}\}$;设任意节点 $node_i$ 的全部子节点的集合为 $Sub(node_i)=\{submode_{ij}|j\in\{1,2,3,...,n_{node_i}\}\}$,其中,$subnode_{ij}$ 为节点 $node_i$ 的子节点,n_{nodei} 为 $node_j$ 的子节点的数目;设由连接 $node_i$ 与 $subnode_{ij}$ 的边所标示的特征值为 $v(node_i,subnode_{ij})$,则 $v(node_i,subnode_{ij})\in Value(feature(node_i))$.基于的决策树的构件查询算法如下:

算法. 基于的决策树的构件查询算法

输入:复用者根据自己的需求,为构件特征模型 F_{RFM} 中的相应特征 f_i 选择特征值 $v(f_i)$;保留 F_{RFM} 中符合本次查询需求的默认特征值;对特征模型中不能确定特征值的特征赋值为'ND',得到本次查询的构件需求特征集 T_{query}.

输出:构件查询结果集 C_{result}

算法:

Step 1. 对构件需求特征集 T_{query} 中所有特征设置匹配标志,以注明该特征是否在查询过程中曾被使用,设任意特征 f_i 的匹配标志为 $f_i.matched$,默认值为 $false$.

Step 2. 将复用者所提交的构件需求特征集 T_{query} 输入决策树 $DT(D_{POT})$根节点 $root$;若对 T_{query} 中任意特征 f_i 特征值 $v(f_i)$='ND'成立,则转 Step7;

Step 3. 若在根节点 $root$ 的子节点集合 $Sub(root)$中,子节点 $subnode_j$ 与 $root$ 的连接边所标示的特征值满足 $v(root,subnode_j)=v(feature(root))$,则标记 $feature(root).matched=true$ 被匹配,并将 T_{query} 从 $root$ 传递给 $subnode_j$;

Step 4. 若构件需求特征集 T_{query} 中任意特征值不等于'ND'的特征 f_i 的匹配标志值 $f_i.matched$ 均为 $true$,转 $Step6$;

Step 5. 设 T_{query} 所在的当前节点为 $node_i$,若在节点 $node_i$ 的子节点集合 $Sub(node_i)$中,子节点 $subnode_{ij}$ 与当前节点 $node_i$ 的连接边所标示的特征值满足 $v(node_i,subnode_j)=v(feature(node_i))$则标记 $v(feature(node_i))$被匹配,并将 T_{query} 从 $node_i$ 传递给 $subnode_{ij}$,转 Step3;

Step 6. 设 T_{query} 所在的当前节点为 $node_i$,若 $node_i\square L(DT)$, 则将当前叶节点所标示的构件 $Comp(node_i)$加入构件查询结果集 C_{result};

Step 7. 设 T_{query} 所在的当前节点为 $node_i$,对以节点 $node_i$ 为根节点的子树进行广度优先遍历,在遍历过程中遇到任意叶节点时,将 $leaf_i$ 所标示的构件 $comp(leaf_i)$加入构件查询结果集 C_{result}.

按照决策树中构件的返回次序对构件查询结果集进行排序,并将排序后的构件集合作为构件查询结果集返回给复用者.在上述过程中,每个从决策树中返回的构件 c_i 都满足相应的决策规则 $r(c_i)$,由上文可知,$r(c_i)$是查询结果集中包含构件 c_i 的成功构件查询记录的归纳,所以若构件查询历史中对 c_i 的查询记录能够满足复用者需求,则本次查询结果也能够满足复用者需求.

3.3 决策规则集的存储

由上文分析可知,对于构件特征集为 F 的构件库 C,存在针对软件项目 P 的构件查询历史数据集 H_{POH};以项目 P 的构件需求特征模型 F_{RFM} 为基础,可以由数据集 H_{POH} 构造训练数据集 D_{POT}.设由训练数据集 D_{POT} 所得的决策树为 DT;每条由根节点到叶节点的路径,所标示的决策规则为 r;所有根节点到叶节点的路径的集合组成的决策规则集为 R;对于不同的软件开发项目,可以建立包含不同决策规则集合的决策树工具;对于同一个构件库可以对不同软件项目提供支持.上文定义的构件库 C、软件项目 P 和决策树 DT 间的关系可以通过如图 3 所示的数据模式进行存储.

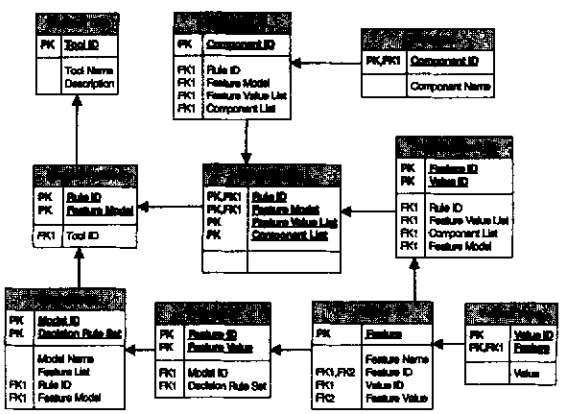

图 3 决策规则数据存储模式

4 基于决策树的构件查询系统

以上文所述基于决策树的构件查询方法为基础,我们把决策树工具引入到传统构件库查询系统中.根据基于决策树的构件查询方法的要求,对传统构件库查询系统结构进行了调整,构建了基于决策树的构件查询系统.该系统以青鸟构件库管理系统[11]为基础,旨在提供面向具体软件开发项目的构件查询服务.本节主要对基于决策树的构件查询系统的组成及各模块的主要功能进行描述.

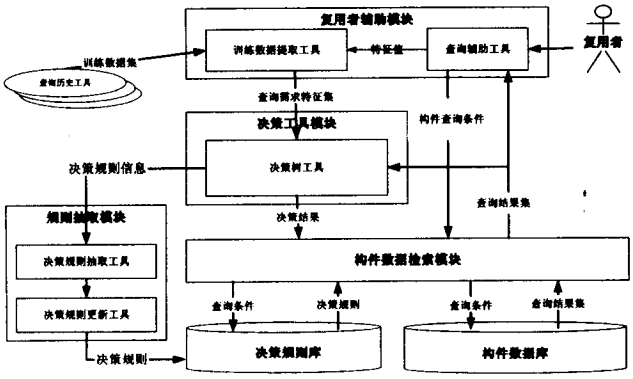

图4 基于决策树的构件查询系统

基于决策树的构件查询系统由4个模块和两个数据库组成:复用者辅助模块,决策工具模块,规则抽取模块,构件数据检索模块,决策规则库和构件数据库,如图4所示.其中构件数据库和构件数据检索模块的功能和结构与现有构件库系统相同[11,12],本文不作详细叙述.其他模块的主要功能如下:

复用者辅助模块

该模块包括两个工具:查询辅助工具和训练数据提取工具.查询辅助工具作用是利用构件需求特征模型,帮助复用者对所要查找的构件进行描述;并将复用者的描述信息转换为决策树工具可接受的数据格式,提供给决策树工具模块.训练数据提取工具的主要功能是,从历史查询记录所包含的成功查询记录中,采集每个查询条件对需求特征模型中相关特征的描述信息,并将这些描述组织成决策树工具的训练数据集提交给决策树工具模块.

决策树工具模块

决策树工具模块的作用是首先根据训练数据集形成决策规则,用户进行查询时根据需求特征集利用决策规则选择构件.它可以由满足以下条件的决策树工具组成:

① 决策树工具必须具备依据相应决策规则对输入数据进行分类选择的能力;② 由于查询辅助工具所提供的构件需求特征集包含离散数据,因此该决策树工具必须支持对离散数据的处理;③ 决策树工具必须具有将自身存储的决策规则进行抽取和形式化表示的能力;④决策树工具能够支持根据历史数据的不断变化,对决策规则进行更新.

决策规则抽取模块

该模块介于决策树工具模块与决策规则库之间,主要作用是将决策树工具训练形成的决策规则集抽取到决策规则库中;同时支持决策规则库的更新维护.该模块由决策规则抽取工具和决策规则更新工具组成.决策规则抽取工具的作用是从决策树工具中抽取决策规则,并根据决策规则与决策树工具、软件开发项目间的数据关系,存入决策规则库中.决策规则更新工具具有对决策规则库的数据管理与数据操作进行分离的作用,在确保决策规则集合数据一致性的基础上,对决策规则库中保存的决策规则进行更新.

决策规则库

由上文可知,对于不同的软件开发项目,可以建立包含不同决策规则集合的决策树工具;对于同一构件库可以对不同软件项目提供支持,不同软件项目由相应的构件需求特征模型支持.决策规则库的主要功能是用于存储决策规则以及决策规则集与相应软件项目、相应构件需求特征模型,相应训练数据集之间的对应关系.决策规则库的存储模式如图3所示.

在系统使用过程中,首先对构件库管理系统使用过程所保存的查询查询历史数据进行分析,根据具体软件开发项目的不同需求,建立相应的构件需求特征模型,并根据该模型从查询历史数据中提取建立决策树工具所

需的训练数据集.利用该数据集对决策树工具进行训练得到相应的决策规则,经规则提取模块保存于决策规则库中.需要进行构件查询时,根据决策规则库中相应的决策规则集重构决策树工具,在决策树工具的辅助下进行构件查询.

5 应用实例

选择C4.5决策树[9,10]作为决策工具,选择3个具有相同开发环境和相近软件需求的管理信息系统开发项目的构件查询历史数据作为测试项目集,基于决策树的构件查询系统的一个基于的应用实例描述如下:

对所选项目的构件查询历史数据中,所有查询条件所包含的构件特征进行分析,确定需求特征模型$F_{RFM}(MIS)$.如图5所示,该需求特征模型表现出软件项目构件需求注重人机交互的特点.

Feature	Comp_Leve	Module	Language	Input_Typ	Output_Typ	Net_Protoco	Operation_Type	Data_Proces	Version	Related_Id	Visual_Effecti	Vision Type	Presentatio
含义	构件层次	构架模块	实现语言	输入类型	输出类型	网络协议	操作类别	数据加工类型	版本信息	相关外设	视觉效果	of视类型	表现形式
默认值	Coding	Presentatio	Java	Object	Object	TCP/IP	ND	ND	Original	ND	3D	MutiDimensio	SourceCode

图5 需求特征模型

参照构件需求特征模型,在训练数据提取工具的支持下,将各个构件查询历史记录中所包含的构件特征值添加到构件需求特征模型中,覆盖相应的默认值,得到符合构件需求特征模型$F_{RFM}(MIS)$的决策树工具的训练数据集,如图6所示.

Comp_Name	Comp_Level	Module	Language	Input_Type	Output_Type	Net_Protocol	Operation_Type	Data_Process	Version	Related_ID	Visual_Effection	Vision_Type	Presentation
Comp1707	Coding	DataConnection	Pascal	ND	Object	TCP/IP	Computing	Others	ND	Mouse	Moving	2DGraphics	Binary
Comp2188	Coding	DataBase	Java	Object	DataTable	Others	Computing	Others	Standard	ND	Moving	SystemElement	ND
Comp5188	ND	BusinessLogic	ND	ControlData	Object	ND	Computing	Others	Original	Displayer	Table	Statistics	Binary
Comp1208	Coding	SystemConnection	Pascal	Object	Object	TCP/IP	Communication	Transforming	Standard	KeyBoard	3D	Table	ND
Comp6578	Testing	ND	Java	Object	DataTable	ND	I/O Operation	Transformtion	Original	Mouse	Color	Text	Binary
Comp3261	ND	SystemConnection	Basic	ControlData	ND	HTTP	Communication	ND	Original	MouseKey	Color	Statistics	SourceFile
Comp3269	Document	BusinessLogic	Java	Object	DataTable	Others	ND	Compressing	Standard	MouseKey	Color	Signal	SourceFile
Comp5689	Coding	Presentation	C/C++	Object	ND	Others	Computing	Encryption	ND	Printer	ND	2DGraphics	Binary
Comp5787	Coding	BusinessLogic	Basic	MediaStream	Object	TCP/IP	I/O Operation	Compressing	Original	ND	Color	Table	ND
Comp2863	Designing	BusinessLogic	Perl	ControlData	DataTable	TCP/IP	Computing	Recognizing	Original	KeyBoard	3D	2DGraphics	ND
Comp3089	Coding	BusinessLogic	Java	Object	Object	Others	ND	ND	ND	Mouse	Table	Statistics	ND
Comp1157	Designing	DataConnection	C/C++	DataSource	DataStream	HTTP	Computing	Encrypting	Standard	Mouse	ND	Text	ND
Comp2760	Designing	BusinessLogic	Java	Object	Object	Others	ND	Compressing	Original	Displayer	ND	SystemElement	SourceFile
Comp1677	Coding	DataBase	C/C++	Object	DataTable	TCP/IP	Communication	ND	Standard	Printer	ND	Statistics	SourceFile
...	...	...	...	...	...	...	...	...	...	...	...	...	...

图6 决策树训练数据集

利用这个训练数据集对决策树工具进行训练,得到用于辅助构件查询的决策树.当该*MIS*项目的复用者需要从构件库中获取可复用构件时,首先从决策规则库中提取与软件开发项目相对应的决策规则集合和构件需求特征模型,对决策树工具进行初始化;然后在查询辅助工具的帮助下利用构件需求模型对所需要的构件进行描述,再由查询辅助工具将查询需求输入到决策树中.例如:在构件库中查找一个能够对指定数据表进行排序,并且能够对排序结果进行二维图形展示的控件,查询条件及返回结果如图7所示.

Input Type = DataSource;Data Process = Ranking;Visual Effection = Table;

Retrieve Result Based On Decision Tree

Comp_ID	Comp_Level	Module	Language	Input_Type	Output_Type	Net_Protocol	Operation_Type	Data_Process	Version	Related_ID	Visual Effection	Vision_Type	Presentation
Comp13499	Coding	DataBase	Pascal	DataSource	Data	Others	Computing	Ranking	Original	Mouse	Table	MutiDimension	Text
Comp5635	Coding	DataBase	Others	DataSource	Data	FTP	Others	Ranking	Original	N/A	Table	MutiDimension	Text
Comp2506	Coding	Presentation	Java	DataSource	Object	N/A	N/A	Ranking	Original	MouseKey	Table	SystemElement	Binary
Comp8422	Designing	Others	Pascal	DataSource	Others	Others	Communication	Ranking	Others	Others	Table	Text	Text
Comp18482	Coding	Others	Pascal	DataSource	DataTable	HTTP	Communication	Ranking	Original	Mouse	Table	MutiDimension	Text
Traditional Retrieve Result:													
Comp2838	Coding	Presentation	Others	DataSource	Data	HTTP	I/O Operation	Ranking	N/A	Others	Table	MutiDimension	SourceFile
Comp2953	Designing	Others	Java	DataSource	Data	TCP/IP	I/O Operation	Ranking	Original	KeyBoard	Table	Text	N/A
Comp3606	Coding	Presentation	Java	DataSource	Object	N/A	N/A	Ranking	Original	MouseKey	Table	SystemElement	Binary
Comp3238	Designing	DataConnection	Pascal	DataSource	DataStream	N/A	Communication	Ranking	Original	Printer	Table	Statistics	SourceCode
Comp5691	Coding	BusinessLogic	Basic	DataSource	Object	Others	N/A	Ranking	Standard	MouseKey	Table	Text	Text
Comp5595	Coding	DataBase	Others	DataSource	Data	FTP	Others	Ranking	Original	N/A	Table	MutiDimension	Text
Comp5438	Document	N/A	C/C++	DataSource	Data	HTTP	Others	Ranking	Others	KeyBoard	Table	Others	Graphics
Comp6622	Designing	Others	Pascal	DataSource	Others	Others	Communication	Ranking	Others	Others	Table	Text	Text
Comp2701	Designing	DataBase	Basic	DataSource	Object	N/A	Computing	Ranking	Original	N/A	Table	MutiDimension	Graphics
...	...	...	...	...	...	...	...	...	...	...	...	...	...

图7 构件查询示例

在上述过程中,复用者在查询需求特征模型的基础上构造查询需求特征集,在决策树工具的辅助下直接根据查询需求特征集得到查询结果.系统利用有经验复用者的构件查询记录对决策树工具进行训练,把复用者的经验知识转换为决策树工具的决策规则,保存于决策规则库中;当无经验复用者进行构件查询时,可以直接利用这些决策规则,由构件需求特征集直接得到相应的构件结果集.基于决策树的构件查询方法减轻了构件查询系统对复用者所具有的构件分类、描述知识的依赖,降低了构件查询难度,提高了构件查询效率.

6 结束语

构件查询作为构件库理论研究的一个重要方面一直保持着较高的研究热度.以降低构件查询难度,提高构件查询准确性为中心也有较多的工作,但是大多数工作都围绕着构件查询方法和构件查询规范[13~15]展开.在本文中我们在对复用者的构件查询过程进行分析的基础上,提出利用构件查询的历史信息辅助无经验复用者进行构件查询的方法,并将决策树方法引入构件查询过程中,对决策树工具辅助下的构件查询方法进行了讨论,提出了一个基于决策树的构件查询方法.在接续的研究工作中,我们将更加深入地对基于决策树的构件复用方法进行探索,并对软件开发环境和软件需求的相似性的判定,查询需求特征模型的选择进行进一步的研究,以提高构件查询方法的自适应能力.

References:

[1] Heineman GT, Councill BT. Component-Based software engineering: putting the pieces together. Reading: Addison-Wesley, , 2001.

[2] Yang FQ, Mei H, Li KQ. Software reuse and software component technology. Acta Electronica Sinca, 1999,27(2),68~75 (in Chinese with English abstract).

[3] Mili H, Mili F, Mili A. Reusing software: Issues and research directions. IEEE Trans. on Software Engineering, 1995,21(6): 528~562.

[4] Ye YG. Fischer: Promoting reuse with active reuse repository systems. In: Frakes WB, ed. Proc. of the 6th Int'l Conf. on Software Reuse. Springer, 2000. 302~317.

[5] Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS. Feature-Oriented domain analysis (FODA) feasibility study. Technical Report, CMU/SEI-90-TR-21, 1990.

[6] Maarek YS, Berry DM, Kaiser GE. An information retrieval approach for automatically constructing software libraries. IEEE Trans. on Software Engineering, 1991,17(8):800~813.

[7] Penix J, Baraona P, Alexander P. Classification and retrieval of reusable components using semantic features. In: Proc. of the 10th Knowledge-Based Software Engineering Conf. Boston: IEEE Computer Society Press, 1995. 131~138.

[8] Wang YF, Zhang Y, Zhu SY, Qian LQ. Retrieving components based on faceted classification. Journal of Software, 2002,13(8):1546~1551 (in Chinese with English abstract).

[9] Rivest R. Learning decision trees. Machine Learning, 1987,2:229~246.

[10] Quinlan JR. C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann, 1994.

[11] Li KQ, Guo LF, Mei H, Yang FQ. An overview of JB (JadeBird) component library system JBCL. In: Proc. of the 24th Int'l Conference on TOOLS ASIA'97. Beijing: IEEE Computer Society Press, 1997. 206~213.

[12] Chang JC, Li KQ, Guo LF, Mei H, Yang FQ. Representing and Retrieving Reusable Software Components in JB (Jadebird) System. Acta Electronica Sinca, 2000,28(8):20~23.

[13] Fischer B. Specification-Based browsing of software component libraries. In: Proc. of the 13th IEEE Int'l Conf. on Automated Software Engineering. Honolulu: IEEE Computer Society, 1998.

[14] Lindig C. Concept-based component retrieval. In: Ohler JK, Giunchiglia F, Green C, Walther C, eds. Working Notes of the IJCAI-95 Workshop: Formal Approaches to the Reuse of Plans, Proofs, and Programs. Montreal, 1995. 21~25.

[15] Mili A, Mili R, Mittermeir R. Storing and retrieving software components: A refinement-based system. IEEE Trans. on Software Engineering, 1997,SE-23(7):445~460.

附中文参考文献:

[2] 杨芙清,梅宏,李克勤.软件复用与软件构件技术.电子学报,1999,27(2):68~75.

[8] 王渊峰,张涌,朱三元,钱乐秋.基于刻面描述的构件检索.软件学报,2002,13(8):1546~1551.

[12] 常继传,李克勤,郭立峰,梅宏,杨芙清.青鸟系统中可复用软件构件的表示与查询.电子学报,2000,28(8):20~23.