

# 基于刻面分类和本体的类库检索\*

余金山, 刘云城

(华侨大学 计算机学院, 福建 泉州 362011)

**摘要:** 介绍一种结合刻面分类描述和本体语义的类库管理和检索方法, 通过刻面分类描述机制和本体的语义扩展与推理能力, 有效地对类进行存储和管理, 方便用户快捷准确地从类库中寻找适合自己需要的类, 从而更好地达到软件复用的目的。

**关键词:** 类库; 软件复用; 刻面分类; 本体

中图分类号: TP311

文献标识码: A

文章编号: 1674-7720(2010)23-0014-05

## Combining faceted classification and ontology for class library retrieval

YU Jin Shan, LIU Yun Cheng

(College of Computer, Huaqiao University, Quanzhou 362011, China)

**Abstract:** This paper presents a system which provides a technique combining faceted classification and ontology for the management and retrieval of class library. By faceted classification and semantic and reasoning retrieval based on ontology, the system establishes an effective mechanism for the storage and management of class. users can quickly and accurately from the class library to find a class suited to their needs, hence better achieve the purpose of software reuse.

**Key words:** class library; software reuse; faceted classification; ontology

软件复用是提高软件生产力和质量的一种重要技术, 已被视为是解决软件危机、提高软件生产率和质量的现实可行途径。目前研究最多的是基于构件的复用, 或称之为基于构件的软件工程 (CBSE)。而要实现 CBSE, 其最根本的问题是要有可用的软构件。完整的构件库是实现 CBSE 的关键, 但各个开发者或开发机构要创建各自的、绝对完整的构件库是不可能的, 或在经济上与时间上是不可行的。

面向对象方法不仅为软件复用的实现提供了内在的结构支持, 也通过类库的类而直接支持代码复用。当前, 许多面向对象开发工具都建立了自己的类库, 类库所起的作用越来越重要。类库已成为一种现成的、宝贵的可复用的软件资产。

但是, 类和软构件是有区别的, 类库也不是一般的软构件库。目前大部分的类库管理系统都存在一些问题, 如只能通过浏览导航的方式查找类, 这限制了程序开发人员快速、准确、高效地检索出所需的类。

本文介绍一个结合刻面分类描述和本体语义的类

库管理和检索方法, 通过刻面分类描述机制和本体的语义扩展及推理能力, 有效地对类进行存储和管理, 方便用户快捷准确地从类库中寻找适合自己需要的类, 从而更好地达到软件复用的目的。主要内容有: 给出一个合适的刻面模型对类进行不同维度的描述; XML 在刻面分类描述中的应用; 在刻面分类的基础上建立本体库, 用来表示类的整个分类体系, 以及基于刻面分类和本体的检索方法。

目前国内外对类库的改进研究有: 吕枫华等人提出的基于类相似性比较的类库检索方法, 着重讨论了比较类相似程度的近似度量方法, 及其在类库检索中的应用, 并且给出了一个行之有效的基于规则的类库检索工具<sup>[1]</sup>。叶青青等人提出的基于语义与句法的 Java 类库检索方法, 使得开发者可以使用自然语言描述待开发软件的功能语义, 提高可复用类库查询的精度<sup>[2]</sup>。徐正权等人提出的新的 Softstore 软件库系统模型<sup>[3]</sup>, 其主要思想是利用相对比较成熟的 Internet 以及数据库管理技术, 对软件资源进行有效管理。

### 1 类库的刻面分类与描述

类的整体说明信息包括类的语义信息和类的关系

\* 基金项目: 福建省自然科学基金项目 (A080013)

信息。类的语义信息包括类名、属性、服务、所属类型、所属开发阶段等信息。类的关系信息包括类与其他类的关系信息。

在类库中,每个元素都是一个类,这些类由属性和作用于属性之上的操作构成。属性可以分为静态属性和动态属性。静态属性属于所有对象的全体,意味着不存在该类的对象,静态属性也是存在的;动态属性随着对象的存在而存在,随对象的改变而改变。操作也可以分为静态操作和动态操作。两者不同之处在于静态操作可以不存在操作的主体,而动态操作必须要有操作的主体,即发起者。

类与其他类之间具有几种不同的关系<sup>[4]</sup>。

(1)继承关系。在继承关系中子类自动继承父类的属性和方法,子类依赖父类而存在,如果单独复用子类,而忽略了其父类,必然会导致信息的不完整,而出现复用错误。所以,作为依赖方的子类入库时,需要同时保存父类的信息以及它们之间的继承关系。

(2)聚合关系。当整体和部分的关系紧密时,整体不能缺少部分,所以作为依赖方的整体入库时,需要同时保存部分类的信息和它们之间的聚合关系。

(3)实例连接关系。一般通过对象的属性来表示一个对象与另一个对象之间的依赖关系。一般来说,该关系并不影响对象作为独立存在的类入库。

(4)消息连接。即所谓的类与类之间的合作关系。消息连接具有很强的依赖性。一方要通过调用另一方的服务来实现本身所提供的服务,如果缺少对方的支持,则本身无法完成相应的服务。因此作为依赖方的调用类入库时,需要同时保存被调用类的复用信息以及它们之间的合作关系。

通过以上几种关系的分析可知,依赖性在各个关系中普遍存在。如果一个类不能单独存在,必须依赖其他类存在并得以保持其完整性,则需要将被依赖类和依赖类以及它们之间的关系同时入库保存。

本系统用剖面分类表示类本身的信息,类与其他类之间的关系则通过本体描述。

根据剖面分类描述的精简、一致性、全面性、剖面正交性、易于理解的原则,选择以下剖面 and 子剖面:

类的基本信息:类的 ID、名称、版本、类的入库时间等。

类的功能:操作、操作对象、操作描述等。

类的应用环境:开发语言、硬件环境、操作平台、数据库平台等。

类的应用领域:类的具体应用范畴。

类的剖面分类与术语示例如表 1 所示。

用类的剖面信息建立类库的剖面模型,使用 XML 语言进行描述,同时在类入库的时候,根据用户所提交的信息,自动生成 XML 文档。例如 Dbconnect 类的剖面

表 1 类的剖面分类与术语示例

剖面信息	术语举例
基本信息	Dbconnect, String, IO 等
功能	数据库,业务逻辑,文件操作
操作对象	数据库,文件,树,XML 等
应用领域	通用,商业,银行,办公等
开发语言	Java, VC++ 等
操作系统	Windows, Linux 等
数据库平台	Mysql, db2, sqlserver 等

信息:

```
< ?xml?version? ="1.0"?encoding? ="GB2312"?standalone =
"yes"??>
```

```
<?XML-stylessheet?type="text/xsl"?href="yxfqust.xml"??>
```

```
<! -- class faceted information -->
```

```
<?xml version="1.0"?>
```

```
<Class>
```

```
<! -- basic information -->
```

```
<BasicInfo>
```

```
<ID>001</ID>
```

```
<Name>DBConnect</Name>
```

```
<Version>1.0</Version>
```

```
<Provider>sun</Provider>
```

```
<EnrollTime>2000-01-01</EnrollTime>
```

```
</BasicInfo>
```

```
<! -- application platform -->
```

```
<ApplicationPlatform>
```

```
<DL>Java</DL>
```

```
<Database>mysql,sqlserver</Database>
```

```
<system>windows,linux</system>
```

```
</ApplicationPlatform>
```

```
<! -- application domain -->
```

```
<ApplicationDomain>information system</ApplicationDomain>
```

```
<! -- operation information -->
```

```
<OperationInfo>
```

```
<Operation>connect to database </Operation>
```

```
<OperationOn>database</OperationOn>
```

```
<OperationDesc >connect to database in different ways </
```

```
OperationDesc>
```

```
</OperationInfo>
```

```
</Class>
```

从具体问题的需求出发,XML 具有如下的突出优点<sup>[5]</sup>:

(1)可读性。用 XML 描述对于任何用户或是计算机程序来说,都容易阅读和处理。

(2)XML 的自描述性使得客户端在收到数据的同时也能理解数据的逻辑结构和含义,从而能直接对这些自描述的 XML 文件中的数据进行操作,增强了检索的语义和作用。

(3)表示的独立性和内容的独立性带来的灵活性。内

《微型机与应用》2010 年第 29 卷第 23 期

容与表示的分离,支持用户界面个性化。表示和操作的分离,支持不同数据源的无缝连接。

(4)强大的表达能力。无论是何种数据源,新的、老的,是否异构,XML都可以用统一的模式进行描述和访问。

(5)可扩展性。XML可以在不破坏现有结构和系统的情况下增加新的数据字段,不影响新老客户端的使用。

XML的这些特点可以为使用者在需要访问不同的源类库时提供标准的统一接口。

检索采用XML树的索引技术和模型匹配检索算法<sup>[6-8]</sup>。

一个刻面描述可以展开成为一棵刻面描述树,而XML文档的数据结构就是树状结构。对一个刻面描述方案,可以将其中的刻面和子刻面分别映射为树中对应的父节点和子节点,对采用某个刻面描述方案描述的类,可以将其对应的刻面描述术语映射为对应的叶子节点。

对于类的查询也可以相应地表示为一棵查询树,即将查询中出现的刻面名和子刻面名转化为对应的父节点和子节点,将查询刻面的术语值转化为叶子节点,并且用一个虚拟的根节点将它们组合成为一棵查询树。因此,对于类的检索就转化为查询树与库中类的刻面描述树之间的匹配。

用户在客户端输入查询条件后,由服务层中的元数据服务组件负责根据用户查询条件生成X-Query查询语句,并进行同义词与近义词处理。然后将生成的X-Query查询语句通过数据访问接口的查询解释器分解查询请求,按照数据映射提供的存储规则与对象合成器交互,使用XML数据的无序树匹配的松弛匹配技术进行检索。

## 2 建立本体表示类库中类的依赖性关系

前面的分析指出类库中的类普遍存在依赖性关系,这些依赖性关系具有很强的语义信息,例如继承、某些类不能单独存在必须依赖其他类而存在才得以保持其完整性等。这里引入本体的主要目的为:(1)利用本体描述类之间的关系;(2)通过本体扩展类库的刻面分类描述体系,利用本体具有的良好概念层次结构和对逻辑推理的支持,实现基于语义上的匹配检索,从而提高查全率和查准率。

本体不仅可以描述类与类之间的各种关系及刻面术语之间的同义词和上下文关系,还可以描述刻面术语之间所具有的其他各种关系,从而尽量从语义上来建立类信息的全面描述。

(1)概念类。概念类表示特定领域中的一组或一类实体,每个概念的不同特点可以由不同的属性描述。如工作描述、功能、行为。语义上讲,它表示的是对象的集合。

(2)公理是表示某一个特定领域内的一些永真式,用

来描述和解释元素及其元素之间的关联和约束问题。

(3)关系代表概念类之间的关系,形式上定义为 $n$ 维的笛卡尔乘积的子集。基本关系是类的层次关系,即所谓的继承关系。除此之外,还有连接关系,连接关系表示除了上下位关系之外的其他关系。具体关系如表2所示。

表2 本体基本关系表

关系名	关系描述
Part-of	表达概念之间整体与部分的关系
Kind-of	表达概念之间的继承关系,如父类与子类
Instance-of	表达概念类与实例之间的关系
Attribute-of	表达某个概念是另一个概念的属性

(4)实例代表元素。从语义上可以理解为对象。

本体的表示采用基于Web的本体描述语言OWL。OWL能够用于清晰地表达词汇表中词汇以及词汇之间的本体关系。用户需要借助本体查询语言来对本体进行利用。本体查询语言作为应用程序使用本体的一个接口。

本体的开发使用由HP实验室研究开发的基于Java的开放源代码Jena。选择Jena的理由是:它为解析RDF、RDFS、OWL和SPARQL提供了一个编程环境和一个基于规则的推理引擎;Jena允许将数据保存到硬盘或OWL文件或关系数据库中;Jena的推理机制支持在创建模型的时候将推理机与模型相关联,能实现基于规则的推理。

对OWL处理而言,语义逻辑的处理才是推理机制的实现。Jena提供的OWL支持包括:方便地访问标准OWL的类和属性;支持多种版本的OWL规范;在基本的查询中通过subClassOf这样的关系来实现类的层级访问和使用。

对于类库检索,本体库中关系的重要意义在于支持语义推理和扩展查询,以提高查全率和查准率。要进行基于本体的类库检索,首先应该建立一个本体库,针对类库进行分析和抽取概念:类库、类、类的基本信息、类的刻面信息、类的接口信息、刻面、术语等。同时分析概念与概念之间的关系。

用OWL语言来描述类库本体可在描述逻辑的基础上建立概念分类层次,然后定义概念类的属性和创建类的实例。

首先确定本体所包含的概念和概念的层次,如类库中出现的概念包括以下几个实体:Library、Class、Facet、Terms等。

可以用OWL语言定义如下:

(1)OWL的概念通过owl:Class定义。

```
<owl: Class rdf:ID="Library">
```

```
<rdfs: label>类库</rdfs: label>
```

```
</owl: Class>
```

(2)概念之间的继承关系可以用<rdfs: subClassOf>表

示。如功能刻面是刻面的一个子概念:

```
<owl: Class rdf: ID="OperationFacet">
<rdfs: label>功能刻面</rdfs: label>
<rdfs: subClassOf rdf resource="#Facet">
</owl: Class>
```

(3)定义概念的属性。属性是一个二元的关系。OWL定义了两种属性,即 owl:Datatype Property 和 owl:Object Property。前者是概念的成员和数据类型,后者是不同概念之间的关系。如定义一个 DatatypeProperty 属性的 className:

```
<owl: DatatypeProperty rdf: ID="className">
<rdfs: domain rdf: resource="Class">
<rdfs: range rdf: resource="xsd:string">
</owl: DatatypeProperty>
```

下面定义一个 ObjectProperty 属性的 containClass:

```
<owl: ObjectProperty rdf: ID="containClass">
<rdfs: domain rdf resource="#Library"/>
<rdfs: range rdf: resource="#Class">
</owl: ObjectProperty>
```

(4)传递性 TransitiveProperty

对称性 SymmetricProperty、自反性 Inverseof、等价传递性 InverseFunctionalProperty。

(5)定义类与类之间的等价关系

```
<owl: InverseFunctionalProperty rdf: ID="equals">
<owl: inverseOf rdf: resource="#equals"/>
<rdfs: domain rdf: resource="#class1">
<rdfs: range rdf: resource="#class2">
<rdf type rdf: resource="http://www.w3.org/2002/07/
owl#TransitiveProperty"/>
```

```
<rdf type rdf: resource="http://www.w3.org/2002/07/
owl#ObjectProperty"/>
```

```
<rdf type rdf: resource="http://www.w3.org/2002/07/
owl#SymmetricProperty"/>
```

```
<rdf type rdf: resource="http://www.w3.org/2002/07/
owl#FunctionalProperty"/>
```

Owl 可以用来定义和表示类与类、类与属性之间丰富的语义关系。在面向对象中类与类之间的关系是确定的,如继承、依赖、聚合等。

本体库的建立是相当麻烦的,如果能够在软件设计过程中抽取本体,则以后本体的建立会节省不少的时间。目前,将面向对象的思想与本体建模的思想相结合得到了一定的研究,这些研究成果对领域本体的构建有一定的推进作用。例如参考文献[9]给出的一种从 UML 类图到 OWL 映射方法。

### 3 加入本体后的类库检索

在进行检索时,首先利用本体知识对检索条件进行预处理,对检索条件进行加强与扩展,然后检索引擎用处理后的条件进行检索,将得到更符合用户需求的返回

结果。利用本体含有的语义信息检索出语义上匹配的一类。

基于本体描述的类库,可以分作三层:最底层为数据服务层,存放刻面描述完毕的类实体,实现对类的存储,包括本体库和 XML 数据库等;中间层为功能应用层,包括本体的推理检索模块,数据访问等功能,主要采用组件开发技术来完成功能;最外层是表示层,负责与用户的交互及为用户显示界面和提供一个可视化的浏览接口。如图 1 所示。

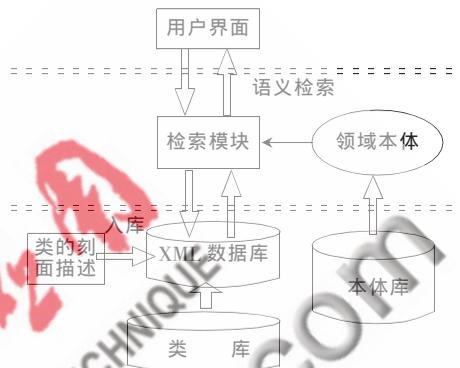


图 1 基于本体的类库检索系统结构

实现采用 MVC 架构的 Struts。

本体的查询模块通过 Jena 提供的推理机接口,利用同义、上下位和语义关系,或者第三方推理机推理出隐含的语义,从而使得查询得以扩展。利用 Owl 本体概念的描述逻辑进行相关和相近概念的推理运算。

(1)用户通过查询界面输入查询条件。

(2)生成初始查询,如分词、关键词匹配。

(3)查询扩展。对初始查询进行语义扩展,然后把扩展后的术语集通过查询界面显示给用户。本体扩展包括以下几种类型:同义扩展、层次扩展、属性扩展、公理扩展和规则扩展。返回的术语集包括术语之间的关系以及相应的解释。

(4)细化:用户根据返回的术语集进一步明确自己的需求。

(5)执行查询和返回查询结果,同时考虑将检索结果按照查询匹配度进行排序。

### 4 实验评测

环境和工具:Tomcat 服务器,Eclipse 开发平台,Protégé,sql server2000 数据库,Jena 推理机,Java 类库中的资源。

类库常规关键词检索方法与本系统检索方法的效率对比,如表 3 所示。

表 3 检索结果对比表

检索方法	检索项目	检索结果				
		库中 相关	实际 检出	检出 相关数	查准 率/%	查全 率/%
类库常规检索方法	文件处理	2 200	2 700	1 600	59.2	72.7
本系统检索方法	文件处理	2 200	2 300	2 200	95.6	100

类作为面向对象设计中的重要元素,为软件复用的实现提供了良好的条件。为了改进现有的类库检索不方便、效率低等问题,本系统在刻画描述的基础上,结合本体语义检索的思想,较好地解决了传统的类库检索方法无法真正理解用户的检索意图,以及查全率和查准率相对差的缺点,用户可方便快捷地从类库中寻找适合自己的类,从而更好地达到软件复用的目的。下一步工作将是实现类库中的类的刻画自动分类和本体信息提取。

#### 参考文献

- [1] 吕枫华,王和珍,费翔林.类相似性的比较及其在类库检索中的应用[J].软件学报,1997,8(4):278-282.
- [2] 叶青青,江水.基于语义与句法的Java类库检索方法与系统[J].计算机工程,2004,30(23).
- [3] 徐正权.类库系统研究及其与软件库的融合[J].计算机与数字工程,2007,38(6):67-69.
- [4] 张文娟.面向对象软件开发工具集中类库和模型库的研究与实现[J].计算机工程,2002,28(3).

- [5] 余金山.基于XML,Tamino和CORBA的软构件管理与检索技术[J].华侨大学学报,2008,29(4):518-521.
- [6] MICHELS P. MIHAILA G A. SIMEON J. Put a tree pattern in your algebra[C]. IEEE 23rd International Conference on Data Engineering, 2007. ICDE 2007. IEEE 2007: 246-255.
- [7] 徐如志,钱乐秋,程建平,等.基于XML的软件构件查询匹配算法研究[J].软件学报,2003,14(7).
- [8] 张海龙,彭鑫.基于刻画与本体的资源描述与检索系统的设计与实现[J].计算机应用与软件,2007,24(9).
- [9] 刘振中,刘勇.基于UML类图的OWL本体映射方法[J].计算机工程,2009,35(13):40-45.

(收稿日期:2010-07-15)

#### 作者简介:

余金山,男,1952年生,教授,主要研究方向:软件工程、网络计算、人工智能应用等。

刘云城,男,1985年生,硕士研究生,主要研究方向:软件工程。