多构件库统一刻面检索机制

马 亮,谢 冰,杨芙清
(北京大学计算机科学技术系,北京 100871)

摘 要: 不同构件库之间实现互通可以有效提高复用者获取构件的效率.现有相关工作如 UDM/BIDM、DCH、Uranus、UDDI 等存在对多构件库检索技术研究的不足.分类是检索的基础,刻面分类是构件库常用分类方法之一,本文在刻面分类的基础上研究了多构件库检索的基本原理,并由此提出一种统一的刻面检索机制,为不同构件库之间互通的实现奠定了基础.

关键词: 软件复用; 构件库; 刻面分类; 检索
中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2002) 12A-2149-04

The Unified Facet-Based Method to Retrieve Component in Multi-Library

MA Liang, XIE Bing, YANG Fu-qing
(*Dept. of Computer Science & Technology, Peking University, Beijing* 100871, *China*)

Abstract: To implement sharing resources in different component libraries can improve the efficiency of finding components. Relative researches include UDM/BIDM, DCH, Uranus and UDDI. But they don't support retrieving components in existing different libraries. Classification is the base of retrieval. Faceted classification is an important scheme to classify component. This paper discusses the principle of component retrieval based on faceted classification and gives an unified facet-based method to retrieve components in multi-library, laying a foundation for sharing resources in different component libraries.

Key words: software reuse; component library; faceted classification; retrieval

1 引言

十几年来构件库技术的研究和实践取得了相当成果,但还存在一些不足之处,其中主要问题是现有构件库之间难以实现互通,复用者为了获得所需构件不得不逐一访问每个库,检索效率较低.为了实现构件库之间的互通,RIG 组织提出了库间互操作标准 UDM/BIDM[1,2].近年来,一些研究人员进行了分布式构件库方面的研究,如 Ci J.X.等人研制了分布式构件中心(DCH)[3]、Sun Y.C.等人研制了 Uranus 系统[4],同时工业界也出现了可用于实现分布式构件库的 UDDI 技术[5].但目前工作都存在对多构件库检索技术研究的不足,UDM/BIDM 仅仅是一种构件库数据的中间表示,DCH 和 Uranus 侧重于解决局部构件库之间的拓扑结构问题,UDDI 主要支持 Web 服务的发布和查询,这些技术都缺乏对常用构件分类方法的支持.

我们认为统一的构件检索机制是实现多构件库之间互通的重要前提,其目标是辅助复用者构造统一的检索条件,由检索工具自动在各构件库中进行检索.分类是检索的基础,常用的构件分类方法包括自由文本、枚举、属性/值和刻面分类四种[6],其中刻面分类在构件库中得到了广泛的应用.本文以刻面分类为基础对统一的构件检索机制进行研究.

2 多构件库刻面检索原理

刻面分类方法由一组描述构件本质特征的刻面组成[7],每个刻面从不同角度对构件进行分类,刻面由一组术语构成,术语之间具有一般/特殊关系,形成结构化的术语空间.

2.1 相关概念的定义

分类模式(Classification Mode)是构件库中构件所拥有的一组共同分类特征的集合.不同构件库适应不同领域特性,在使用刻面分类方法时可能产生不同的刻面分类模式,例如 REBOOT 系统定义了 Abstraction、Operations、Operates on、Dependencies 四个刻面[8],则这四个刻面及相应的术语空间构成一个刻面分类模式.

定义 1 刻面是一棵有向树,根结点的值是该刻面的名称,根结点所有子树构成的森林称为术语空间,记为 $\{T_i = \langle V_i, E_i\rangle\}$,其中 $V_i = \{v \mid v$ 是一个结点,v 的值是一个术语$\}$,$E_i = \{\langle v_1, v_2\rangle \mid v_1, v_2 \in V_i, v_1$ 和 v_2 代表的术语具有一般/特殊关系$\}$,术语空间中任意两个术语都不相同.

术语之间的一般/特殊关系具有如下性质:设某个术语空间 TS 中术语 v_1、v_2 是结点 k_1、k_2 的值,则 TS 中存在一条 k_1 到 k_2 的路径(v_1 和 v_2 存在一般/特殊关系).

图 1 是一个简单的刻面.

收稿日期:2002-06-06;修回日期:2002-09-13
基金项目:国家 863 项目(No.2001AA113070)

实际应用中即使名称不同的刻面也可能代表同一个分类视角，例如刻面“功能”和“function”、刻面“应用环境”和“使用环境”. 若两个刻面 F_1 和 F_2 都是从分类视角 A 对构件进行分类，则称 F_1 和 F_2 是基于分类视角 A 的同视刻面，记为：$[A]F_1 \approx F_2$. 若 F_1 和 F_2 的分类视角完全相同或者一个刻面的分类视角包含另一个，则可以简记为：$F_1 \approx F_2$.

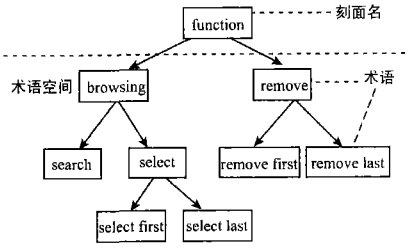

图 1 一个简单的刻面

定义 2 刻面分类模式 FM 是一组刻面的集合，记为 $FM = \{F \mid F$ 是一个刻面$\}$，$\forall F_1, F_2(F_1 \in FM \wedge F_2(FM \Rightarrow \neg F_1 \approx F_2))$.

定义 3 刻面分类信息 S 是一个三元组，记为 $S = \langle m, n, v\rangle$，其中 m 是刻面分类模式的名称，n 为 m 中一个刻面的名称，v 是该刻面的一个术语.

构件一般由分类信息、构件属性、构件关系、构件规约和构件实体组成，除刻面分类信息外，其它成分与本文研究内容关系不大，为了表达形式的简明，暂时将它们忽略，给出构件的定义如下：

定义 4 构件 C 是一个二元组，记为 $C = \langle ID, S\rangle$，其中 ID 是构件标识，S 是一个刻面分类信息的集合.

同样，针对本文的研究，我们仅考虑构件库中的构件和刻面分类模式，给出构件库的定义如下：

定义 5 构件库 CL 是一个二元组，$CL = \langle C, M\rangle$，其中 C 是构件的集合，M 是一个刻面分类模式，C 和 M 满足以下条件：$\forall c, s$(设构件 $c = \langle ID, S\rangle$，分类信息 $s = \langle m, n, v\rangle$，则 $s(S \wedge c(C \Rightarrow m =$ 刻面 M 的名称)，即库中构件只能使用本库的刻面分类模式分类.

检索条件通常根据分类模式构造. 基于刻面分类进行检索时，检索条件由多个刻面名/术语对组成，例如，某刻面检索条件为$\{\langle$“功能”，“sort”$\rangle$，$\langle$“使用环境”，“Linux”$\rangle$，$\langle$“层次”，“编码”$\rangle\}$，希望找到功能是 sort，在 Linux 环境下使用的代码件. 为此我们给出如下定义：

定义 6 刻面检索条件是一个二元组，记为$\langle m, FT\rangle$，其中 m 是一个刻面分类模式的名称，$FT = \{\langle n, v\rangle\}$是刻面名/术语对集合，$\forall \langle n, v\rangle(\langle n, v\rangle \in FT \Rightarrow n$ 是 m 中的一个刻面名$) \wedge \forall \langle n_1, v_1\rangle\langle n_2, v_2\rangle(\langle n_1, v_1\rangle(FT \wedge \langle n_2, v_2\rangle \in FT \Rightarrow n_1 \neq n_2)$.

2.2 刻面检索的基本原理

刻面检索是给定一个刻面检索条件，从库中寻找所有满足该条件的构件的过程. 刻面检索条件可以分解为若干基本检索条件. 设 $p = \langle m, FT\rangle$是一个刻面检索条件，$FT = \{\langle n_1, v_1\rangle, \cdots, \langle n_k, v_k\rangle\}$，则称$\langle m, n_i, v_i\rangle (i = 1, \cdots, k)$为 p 包含的基本检索条件.

设 $p = \langle m_p, n_p, v_p\rangle$是一个基本检索条件，$c = \langle i_d, S\rangle$是一个构件，若存在 $s_i(S, s_i = \langle m, n_i, v_i\rangle$，且 p 和 s_i 满足以下条件，称构件 c 满足基本条件 p，记为 $Fit(c, p)$：

(1)刻面 $n_p \approx$ 刻面 n_i；

(2)$v_p = v_i \vee m$ 中存在一结点 d_p 到 d_i 的路径、d_p、d_i 的值分别等于 v_p、v_i.

第一个条件要求构件中必须存在和基本检索条件分类视角相同的分类信息. 第二个条件要求基本检索条件和分类信息中的术语相等或具有一般/特殊关系. 例如，构件 $c = \langle id,$ $\{\langle m,$“编程语言”，“$VC++$”$\rangle\}\rangle$满足基本检索条件 $p = \langle m'$，“程序设计语言”，“$C++$”$\rangle$.

设 $CL = \langle C, M\rangle$是一个构件库，则 $R = \{c \mid \forall c(c \in C \wedge Fit(c, p))\}$是基本检索条件 p 在 CL 中的检索结果.

设 p 是一个刻面检索条件，$p_1, \cdots, p_k$ 是 p 包含的所有基本检索条件，$R_1, \cdots, R_k$ 分别是这些基本检索条件在 CL 中的检索结果，则 $R = R_1 \cap \cdots \cap R_k$ 是 p 在 CL 中的检索结果.

设 $CL_1, CL_2, \cdots, CL_n$ 是 n 个构件库，p 是一个检索条件，$R_1, R_2, \cdots, R_n$ 是 p 分别在这 n 个库中的检索结果，则 $R = R_1 \cup R_2 \cup \cdots \cup R_n$ 是 p 在 $CL_1, CL_2, \cdots, CL_n$ 中的检索结果.

2.3 检索条件的判定

在对多个构件库进行检索时，基本检索条件可能和库中所有刻面的分类视角都不相同，即使分类视角相同，基本检索条件中的术语也可能不在库的刻面分类模式中，这两种情况都没有必要对该库进行检索. 因此，对检索条件进行判定可以提高多构件库检索的效率.

设构件库 $CL = \langle C = \{C_i, i = 1, 2, \cdots, p\}, M = \{F_i, i = 1, 2, \cdots, q\}\rangle, p = \langle m, n, v\rangle$是一个基本检索条件、以下算法判断 p 是否需要在 CL 中检索：

```
NeedRetrieve(CL, p)
{ Result = FALSE;
  for (i = 1; i <= q && ! Result; i++)   // 依次判断每个刻面
    if(F_i ≈ 刻面 n)
      for(F_i 的每个术语 t)
        if (t == v) {Result = TRUE; break;}
  return Result;
}
```

3 统一刻面分类模式

在检索单个构件库时，复用者根据库的分类模式构造检索条件，完成检索操作. 当对多个构件库进行检索时，复用者通常需要逐一了解不同库的分类模式，多次为同一需求构造不同的检索条件，检索效率较低. 因此，我们希望在各库分类模式的基础上建立一种统一分类模式，使得复用者可以据此构造各库普适的检索条件. 统一刻面分类模式的基本思想是通过合并运算将不同构件库刻面分类模式中的相同术语合并，形成一个唯一的刻面分类模式.

3.1 刻面合并算法

设 $F_1 = \langle V_1, E_1\rangle$、$F_2 = \langle V_2, E_2\rangle$是两个刻面，若 F_1 和 F_2 满足如下条件，则称 F_1 和 F_2 是可合并的：

(1) $F_1 \approx F_2$;

(2)一个刻面的名称是另一个刻面中的一个术语.

我们将刻面 F_1 和 F_2 的合并运算记为 $F = F_1 \cup F_2$.合并原则是:合并后 F_1 和 F_2 中的术语以及它们之间的一般/特殊关系保持不变.下面算法将两个可合并的刻面 F_1 和 F_2 合并为刻面 F,设 F_1 的名称是 F_2 中一个术语,$F = \langle V, E\rangle$.

Merge(F_1, F_2)

E = {所有边 $e(E_2, e = \langle E_2$ 的根, $v\rangle$}; V = {E_2 的根和根的儿子};

$E' = E_1 + E_2 - E$;

$E_{new} = E$;

while (E' ! = *NULL*) {

for (所有$\langle u, v\rangle(E_{new})$ {

$E_{new} = E_{new} - \langle u, v\rangle$;

for (所有$\langle s, w\rangle(E'$, s 和 v 的值相等){

$E' = E' - \langle v, w\rangle$;

if(w 代表的术语不在 F 中) {

$V = V + \{w\}$; $E = E + \{\langle v, w\rangle$; $E_{new} = E_{new} + \{\langle v, w\rangle$;}

}

else // 设$\langle v', w'\rangle(E$, w 和 w'的值相等

if(s 和 v'的值不相等) // 设 E 中 x 和 s 的值相等,需要调整结点的层次

{$E = E - \{\langle v', w'\rangle\}$; $E = E + \{\langle x, w'\rangle\}$;}

}

}

}

3.2 统一刻面分类模式的建立

统一刻面分类模式的建立思想是将所有刻面划分为若干相互独立的同视刻面集合,然后为每个同视刻面集合生成一个唯一刻面,最后用这些新刻面建立统一刻面分类模式.划分后的每个同视刻面集合应当确保可以合并为一个刻面,下面先给出一个同视刻面集合的划分方法,需要在人工辅助下完成.

设已有刻面为 F_1、F_2、…、F_n,对这 n 个刻面依此执行以下操作:

(1)若当前没有一个同视刻面集合和待划分刻面 Fi 的分类视角相同,则增加一个新同视刻面集合,用 Fi 的名称作为新同视刻面集合的名称.继续处理下一个刻面;

(2)若 F_i 可划分到同视刻面集合 f_s 中,则 $f_s = f_s + \{F_i\}$;

(3)若 f_s 中没有一个刻面和 F_i 是可合并的,则在 f_s 中选择一个分类视角与 F_i 最接近的刻面 F,若 F 的分类视角大于 F_i,则在 F 中选择一个适当结点增加一个值与 F_i 名称相等的儿子,否则在 F_i 中选择一个适当结点增加一个值与 F 名称相等的儿子;

(4)若 f_s 的名称是 F_i 中的一个术语,则将 f_s 的名称改为 F_i 的名称,并且依次检查其它同视刻面集合 f_s',若 f_s'的名称是 F_i 中的一个术语,则 $f_s = f_s + f_s'$,删除 f_s';

(5)继续处理下一个刻面.

上述方法中第三步需要人工辅助完成,例如已有同视刻面集合 f_s = {"表示方法"、"编程语言"},待划分刻面 F_i 名为"汇编语言",若 f_s 中的刻面和 F_i 都不是可合并的,则需要人工选择刻面"编程语言",在其中增加一个名为"汇编语言"的术语.

设同视刻面集合 $f_n = \{F_1, F_2, \cdots, F_m\}$,下面的方法为 f_n 生成一个唯一刻面 F.令 $f_n' = f_n$,执行以下步骤:

(1)选择 f_n'中的任一刻面 F_i,并从 f_n'中删除 F_i;

(2)对所有 $F_j(f_n'$,设 r_i、r_j 分别为 F_i 和 F_j 根,若 $\exists v(v(F_i \wedge v$ 和 r_j 的值相等)$\vee \exists v(v \in F_j \wedge v$ 和 r_i 的值相等),则令 $F_i = F_i \cup F_j$,从 f_n'中删除 F_j;

(3)若 $f_n' \neq \emptyset$,转(2),否则算法结束.

算法结束后,F_i 即为合并后的唯一刻面.

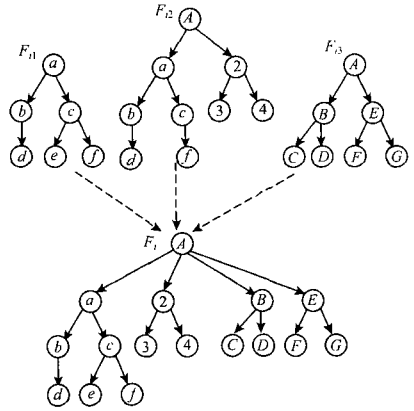

图 2　统一刻面分类模式中刻面的构造

图 2 是上述算法的一个简单例子,设{F_{i1}, F_{i2}, F_{i3}}是一个同视刻面集合,F_i 是算法生成的一个唯一刻面.

设全部刻面可划分为 n 个独立的同视刻面集合,分别为每个同视刻面集合生成新刻面 F_i, $i = 1, \cdots, n$,则统一刻面分类模式 UFCM = {$F_1, F_2, \cdots, F_n$}.

根据上述算法,我们很容易证明:若统一刻面分类模式根据构件库 CL_1、CL_2、…、CL_n 的刻面分类模式建立,则这 n 个构件库中的任何一个构件 c,通过统一刻面分类模式都可以构造一个基本检索条件 p,使得 Fit(c, p).

上述算法中,两个刻面合并的过程需要对其中每个结点进行一次读取和判断操作,算法时间复杂度和结点个数是线形关系.每个同视刻面集合中的 N 个刻面需要进行 $N-1$ 次合并运算.因此总的时间复杂度是[结点个数 * 刻面个数].

4 分类模式注册中心

为了建立统一分类模式,各构件库分类模式需要用一种公共设施统一进行管理,我们将这种公共设施称为分类模式注册中心(CMR).多构件库统一刻面检索机制将以 CMR 为核心实现,如图 3 所示.各构件库将自己的刻面分类模式在 CMR 中注册,CMR 建立统一刻面分类模式并发送给检索工具,复用者据此构造统一检索条件,由检索工具自动在各库中进行检索.

我们设计了一个 CMR 原型,使用 XML 表示刻面分类模

式.XML 不仅是目前 Internet 上数据交换的标准,而且 XML 文档的树形结构能够很自然地对刻面分类模式进行表示.此外,文档对象模型(DOM)提供了强大的操纵 XML 元素的能力,便于统一刻面分类模式建立算法的实现.

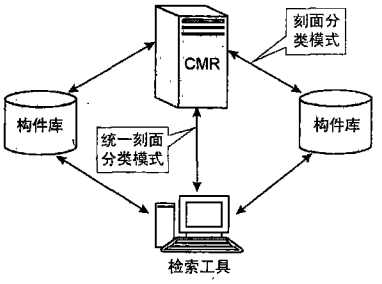

图 3 多构件库统一刻面检索机制

我们选用一个通用构件库和一个操作系统构件库的刻面分类模式进行了统一刻面分类模式建立的实验.前者包含"功能"、"应用领域"、"层次"、"使用环境"和"表示方法"五个刻面,共 79 个术语,后者包含"功能"、"领域"、"环境"、"语言"四个刻面,共 47 个术语.两个分类模式注册后,管理员将这些刻面划分为五个同视刻面集合.CMR 所建立的统一刻面分类模式包含五个刻面,共 114 个术语.从结果来看,统一刻面分类模式的刻面和术语层次划分都比较合理.考虑到实际应用中刻面合并的情况比较复杂,因此 CMR 应当为管理员提供对统一刻面分类模式进行校正的功能,如调整层次划分明显不合理的结点的层次,以提高统一刻面分类模式的实用性.

5 结束语

刻面检索是构件库的主要检索方法之一,本文讨论了多构件库的统一刻面检索机制,为实现多构件库之间互通奠定了基础.本文的研究没有考虑不同分类模式中术语出现同义词的情况,实际应用中应当对此进行处理.可以在 CMR 中建立同义词词典,从每组同义词中选择一个作为规范词汇.在建立统一刻面分类模式前需要将各构件库刻面分类模式中的术语用规范词汇替换.

实际应用中实现构件库之间互通还需要其它相关工作的支持,如为构件库制定公共查询服务的标准访问接口,为检索结果建立标准数据模型等等,这些是我们进一步的工作.

参考文献:

[1] RPS-0002. RIG Uniform Data Model for Reuse Libraries(UDM)[S].

[2] RPS-0001. RIG Basic Interoperability Data Model (BIDM)[S].

[3] Ci J X, Tsai W T. Distributed component hub for reusable software components management [A]. Computer Software and Applications Conference[C]. USA, 2000. 429 - 435.

[4] Sun Y C, Kao M L, Lei C L. A fully distributed approach to repositories of reusable software components[J]. Journal of Information Science and Engineering, 2001, 17(1): 147 - 158.

[5] UDDI 技术白皮书[DB/OL]. http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf.

[6] Frakes W B, Pole T P. An empirical study of representation methods for reusable software. Software Engineering[J]. IEEE Transactions on, 1994, 20(8): 617 - 630.

[7] Ruben Prieto-Diaz, Peter Freeman. Classifying software for reusability [J]. IEEE Software, 1987, 1: 6 - 16.

[8] Guttorm Sindre, Reidar Conradi, Even-Andr Karlsson. The REBOOT approach to software reuse[J]. J System Software, 1995, 30: 201 - 212.

作者简介:

马 亮 男,1971 年生于湖南,北京大学计算机科学技术系博士研究生,主要研究领域为软件工程、Web 环境下的信息集成与共享.

谢冰 男,1970 年生于湖南,1998 年毕业于国防科技大学计算机系,获博士学位,现为北京大学计算机科学技术系副研究员,主要研究领域为软件工程、形式化方法、分布式系统等.