第26卷 第11期 计 算 机 工 程 2000年11月
Vol.26 № 11 Computer Engineering November 2000
• 青鸟工程及其CASE工具专题 • 文章编号：1000—3428(2000)11—0087—04 文献标识码：A 中图分类号：TP311.56

构件库功能集模型

张海飞，袁 磊，夏宽理
（复旦大学计算机系，上海 200433）

摘 要：如何快速有效地在构件库中查询到满意的构件，人们大都采用经验化的方法。该文基于NATO标准的构件库，构造了一个可以帮助构件复用者更好的查询构件库的数据模型，并分析了该模型的建立和动态更新方法。

关键词：构件库；决策支撑树；功能集

A Model of Software Component Librarys Function Set

ZHANG Haifei，YUAN Lei，XIA Kuanli
(Computer Department of Fudan University, Shanghai 200433)

【Abstract】Software component reusers only have experiential ways to find the wanted component in a given component library so far. This essay will build a data model based on NATO standands-based component librarys to help reusers find the aim components more fastly and easily. The dynamic adjust of the data model is also introduced with some algorithms.

【Key words】Component library; Decision support tree; Function set

在青鸟工程的实施中，构件库的研究成为一个重要的课题。青鸟构件库采用与NATO标准基本一致的标准进行定义。本文就是在开发青鸟工程的大环境下作为独立课题完成的。本文不局限于青鸟构件库，较适合于基于特定领域的构件库。

1 功能集模型

从查询的角度来看，整个构件库可以有多个维。在下面的讨论中，我们只讨论功能维。

1.1 用集合表示构件库的所有功能

定义1：从集合的角度定义构件库的功能维。

设C为整个构件库，Pi为构件，则C={ Pi，0<=i<k }。当k=0时，构件库为空集，由于我们的构件库功能集合由此提取，因此也为空。

我们将单个构件表示如下：

Pi={ Fi,1 ,Fi,2 ,Y↑,Fi,ni }，Fi,j表示功能。

定理1：一个没有重复构件的构件库中不存在两个完全相同的功能表示。

证明：我们定义功能表示采用的主键是：构件表示主键+功能名称。由于没有重复的构件表示，所以不同构件的功能表示不可能相同。根据构件设计的一个基本原则：同一构件不存在两项相同功能，得证。

推论1：当i<>j时,Fi,x <> Fj,y

由定义1和定理1得。

由推论1得集合C={ F1,1,f1,2,.....,f1,n1，F2,1 ,f2,2 ,.....,F2,n2，Fk,1 ,fk,2 ,.....,Fk,nk （*）

到目前为止，我们只说明F代表了Fine Grain的功能，到底细到哪一个程度，与构件库的规模、构件库中构件的性质有关。

定义2：我们将（*）重新写成：

C'={ |F'x,1 ,F'x,2 ,.....,F'x,nk|，x∈N。C的每个子集被命名为Fx。Fx在C中唯一。

显然C=C'。

在我们的模型中，同一Fx中的元素都有某种相似性。当将这种相似性定义为某种功能相似性时，我们就得到了建立模型的基础。

1.2 功能集模型：热带树模型TTM（Tropical Tree Model）

按照Fx来建立查询的结果是一个相似功能集合，该集合中的多项功能指向了多个构件。该结果在查询者进行决策后即可以得到某个确定的构件。因此，TTM应该在一定程度上支持决策分析。一个可行的策略是将Fx进行抽象和分类，组织成树状结构。由于该数据结构的每个节点是一个功能子集，与常规树结构有所不同，因为一个节点上派生出许多“分支”，我们称其为热带树。

在TTM中，节点分为两种类型：叶子节点Fx，用◯表示；抽象功能描述节点Abs_ Fx，用□表示。

一个Abs_ Fx节点有最大子树数量的限制，记为DC，我们称其为决策量（DecisionCount）。DC的大小会影响TTM的高度，这个高度可以被看作最大决策步数MDSC（Max Decision Step Count）。

如果将每个Fx节点看作一个数据库表，那么一个Abs_ Fx则是一个基于某些表或其它视图的视图。

在Fx节点中，每个功能的主要数据项包括：功能名称，一个指向所属构件的指针。此外还应该有一些支持决策的信息，与具体实现有关。

TTM可以从既有构件库模型派生出来，也可以伴随着

作者简介：张海飞(1976～)，男，研究生，主攻方向为软件工程；袁 磊，研究生；夏宽理，教授
收稿日期：2000-04-07

构件库的建立而逐步生长。其提供的按“功能”查询能力依赖于以下几个因素：1)TTM的成熟度。指该TTM已经表示了构件库中多少比例的构件；2)适当的参数选择。包括DC，MDSC以及Fx节点的信息项设置等；3)Abs_ Fx节点的生成算法。在本文后面将描述一个简单的算法。这些算法与具体实现有关；4)TTM树的查询算法；5)友善的用户界面。提供超文本链接方法查询构件，并建议提供构件功能比较以支持决策。

1.3 TTM功能的细化

建立TTM的一项重要工作是确定构件库的“功能”元素。确定工作需要经过调研。

· 调研的对象：构件复用者以及希望按TTM查询构件库的软件开发者。特别要指出的是，当复用者对构件库中的每一个构件都十分了解时，将不适合成为调研对象。

· 调研的目的：了解复用者准备开发某项功能时是如何到构件库中寻找合适的构件的。这里必须指出，被调查人员往往凭借历史经验查询联机帮助或文档，其方法或过程是模糊的。调研人员必须将其明朗化，规范化。

· 调研手段：可采用功能对照表格，以及一系列的关于选择依据的问题。

· 调研结果：功能分解图和TTM树的一些参数。由于构件库是不断发展的，这些初始参数在一段时间后可能会变更。

在第2节的例子中，我们将看到一张功能分解图。

1.4 建立TTM的Fx节点和Abs_ Fx节点的算法

假设C中有n个构件，经划分细化，C'中有m项功能，可以得到如下一个n × 2^m矩阵PF:

	00…0	00…1	……	11…1
P0				
P1				
…				
Pn-1				

在这个矩阵中，每一列的标识都是m为二进制数，并且大小从0到2^m-1.

这2^m列正好对应于所有m项功能的组合情况。

PF的元素值为：

$$PF[i,j]=\begin{cases}0, & \text{表示构件Pi不具有功能组合j，以空表示}\\ 1, & \text{表示构件Pi具有功能组合j}\end{cases}$$

这里特别要指出，列标识中只有k位是1的列表示了k个功能的组合。

1.4.1 Fx节点的生成方法

```
For Col=00...0 to 11...1 do
Begin
  Fcol=∮
  For Row=0 to n-1 do
  Begin
    If PF[Row,Col]=1 Then
       Fcol= Fcol+{ 列Col对应的功能子集 }/*这里是集合加法*/
  End
End
```

按照上述算法，当m比较大时，一个可怕的后果产生了：Fx节点一共有2^m个。但是值得庆幸的是，这些节点中有许多是空集，而且我们还可以采取一定的措施将算法中的m值的大小控制在可以接受的范围内，后面将再次讨论。

1.4.2 Abs_ Fx节点的生成算法

由于Abs_ Fx节点是一种视图，其生成方式受到查询需求的约束，因而不同的数据库应该根据具体情况采取不同的算法。

本文采取的算法基于二叉树，该算法的一些启发性规则可以列举如下：

① 如果某两个列的标识仅相差1位，那么考虑在这两个列上建立视图，如果符合②，建立的视图的标识为将不同为置为*号的原列标识。例如110⊕100=1*0。

② 如果某两个列的内容比较接近，考虑将其合并；否则不合并。

③ *号通表0与1。*号与0或1运算的结果仍为*。

④ Abs_ Fx节点= Abs_ Fx节点⊕Abs_ Fx节点| Abs_ Fx节点 Fx节点 | Fx节点 ⊕ Fx节点

⊕运算符将在例子中给出解释。

⑤ 为了缩小合并列的跨度，采用二分法。

在第2节的例子中，我们将运用这些规则生成一棵TTM。

1.5 更新TTM

1.4 中提出的算法依据已有构件库生成一棵TTM。当构件库的构件增加或某个构件版本更新时，应该采取有效的方法来更新和繁荣已经长成的TTM。

(1)增加新构件　当一个新构件Pn={ fn,1,fn,2,...fn,nn}被加入到构件库C中时，对于TTM来说，可能产生两种情况：1)该构件没有增加新的功能，仅是TTM中已有功能的某种组合；2)该构件增加了新的功能。

对于这两种情况，本文采取了不同的方法来调整和繁荣TTM。在第2节的例子中，我们能够看到算法的具体实现。

(2) 构件版本更新　版本更新产生两种情况：已有功能完善和增加新功能。对于前者，TTM的结构不作调整，仅在节点内部的参考信息中改变一些信息。对于后者，可以采取(1)中增加新功能的算法来完成。

2 功能集的生成和动态更新的一个实例

现在，假设我们拥有一个小的构件库：编辑构件库。下面让我们建立TTM，并且利用前面提到的各种算法的思想来调整和繁荣它。

2.1 初始构件库定义

在我们的初始构件库中，拥有以下5个构件：Tedit，Tfmtedit，Ttextedit，Trichtextedit，Toleword，分别用P0...P4表示。

经过调研得知，该构件库得复用者关心以下4个功能：f1——能够输入常规字符串，f2——能够控制字符串的最大长度，f3——能够设置多种字体，f4——可以进行打印格式控制。

构件与功能的对照表如表1所示。

我们看到，表1的前半部分为空，这是常见的情况。

2.2 生成Fx节点、Abs_Fx节点，初始TTM

由表1我们很容易得到Fx节点的定义如下：

F1000={ P0.f1,P1.f1,P2.f1,P3.f1,P4.f1 }

F1001={ P4.f1,P4.f4 }

F1010={ P3.f1,P3.f3,P4.f1,P4.f3 }

F1011={ P4.f1,P4.f3,P4.f4 }

F1100={ P0.f1,P0.f2,P1.f1,P1.f2 }

F1001={ P1.f1,P1.f2,P1.f4 }

由这些节点，我们采用⊕运算来得到Abs_Fx节点。

Abs_F110* = F1100 ⊕ F1001 ={ f1, f2, ¬f3}

Abs_F101* = F1010 ⊕ F1011 ={ f1, f3, ¬f2}

Abs_F1*0* = Abs_F110* ⊕ F1000 ={ f1,¬f3}

Abs_F10** = Abs_F101* ⊕ F1001 ={ f1,¬f2}

Abs_F1*** = Abs_F1*0* ⊕ Abs_F10** ={ f1}

由此构造出的TTM如图1所示。

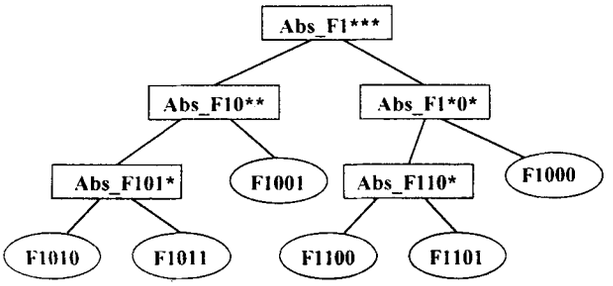

图1 TTM

2.3 增加新构件与TTM动态调整

设P5为Trichedit，此时构件功能对照表如表2所示。

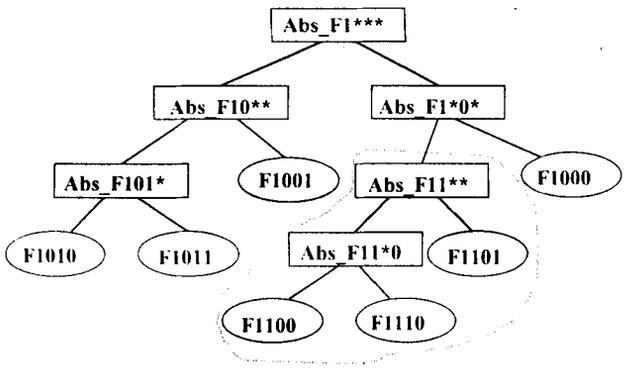

图2 重构后的TTM

我们不需要从头开始重新生成TTM，而是采取调整已有TTM的方法。以下是调整方法：

· 在F1000中添加P5.f1，F1000={ P0.f1,P1.f1,P2.f1,P3.f1,P4.f1,P5.f1}

在F1010中添加P5.f1，P5.f3，

F1010={ P3.f1,P3.f3,P4.f1,P4.f3,P5.f1,P5.f3}

在F1100中添加P5.f1，F1100={ P0.f1,P0.f2,P1.f1,P1.f2,P5.f1 }

· 我们发现原TTM中没有F1110，此时的策略是在TTM中寻找Fx，使x与1110只相差1位，然后构造新Abs_Fx节点。

重新构造后的TTM如图2所示。

被调整的子树用虚线包围。我们注意到,除了增加了一个Abs_F1x 节点外,还将Abs_F110* 修改为Abs_F11**。

在上述算法的第二种情况中，如果在TTM中找不到满足条件的Fx，则采取搜索Abs_Fx 节点的方法。由于*号的通配作用，一定能够找到一个满足条件的Abs_Fx 节点。此时，再根据调整算法进行调整。此时，由被选中的Abs_Fx节点和Fx节点组成新的Abs_Fx 节点，挂在被选中的Abs_Fx节点的父节点的下面。

例如，对于图2表示的TTM，增加新构件P6=Tdbedit，该构件为功能集合新增加了一个功能F5——可以对数据库字段值编辑。限于篇幅在此不一一详述，感兴趣的读者可直接与我们联系。

3 控制功能矩阵规模

TTM的规模主要来自矩阵的列数。矩阵的列数与TTM中的功能数量是指数关系。所以我们不应该在一棵TTM中涉及过多的功能。比如说我们可以取20。

一个大型的构件库的功能成千上百，此时将其用一棵TTM来表示显然不可能。考虑这样一个事实：构件的功能成集团形分布。比如说，上面提到的编辑功能就可以看作一个功能分布集团。再比如说，报表制作功能也可以看作一个功能集团。这些大大小小的功能集团之间内容可以重叠。

我们的策略就是对功能数适中的功能集团建立TTM。对于较大的功能集团，可以将其按照一定的策略划分。对于功能数过少的集团，我们考虑合并，但合并不是必需的。

这样，对于一个大约有1000个功能的构件库，在TTM功能数最大值取20的情况下，我们可以得到约50-100棵TTM。

建立TTM的目的是查询。我们不妨将一系列TTM的树根作为一个索引表。这样，当查询者想要具有某种编辑功能的构件时，用“编辑构件”作索引，很快能够在“编辑功能TTM”的帮助下找到满意的构件。

由于选取功能集团的缘故，上述模型特别适合于基于特定领域的构件库，因为这样的构件库的功能集合拥有以下几个特点：

表1 构件与功能对照

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
P0									1				1			
P1									1				1	1		
P2									1							
P3									1		1					
P4									1	1	1	1				

说明：在例子中，数据仅用来演示算法，并无严格的实际意义。

表2 P5为Trichedit时的构件功能对照表

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
P0									1				1			
P1									1				1	1		
P2									1							
P3									1		1					
P4									1	1	1	1				
P5									1		1		1		1	

· 功能数量在一定范围内;

· 功能集中度，集功能相似程度比较高;

· 查询针对性强.

4 结束语

本文建立的TTM是一棵二叉树，也就是说该TTM支持二元决策. 如何将其扩展为更一般的多元决策树，还有待于进一步研究. 本文提出的启发性算法忽略了许多优化因素，优化能够提高TTM的质量. 这些优化包括构件功能之间复杂的继承关系、功能的重叠性等. 如何将继承关系结合到本文提出的模型之中是一个值得进一步研究的问题.

参考文献

1 Software Reuse-Edited by Even-Andre Karlsson. John Wiley & Sons Ltd,Baffins Lane,Chinchester West Sussex PO19 1UD, England,1995

2 Berard E V.Essays on Object-oriented Software Engineering, Wolume I,Prentice-Hall,1993

3 J B Project Group.青鸟构件库系统的设计与实现.Technical Report , Peking Univ., 1997

※※

（上接第70页）

$$\sum_{i=1}^{n} W_i = 1$$

总体相似度的计算公式为：$\sum_{i=1}^{n} W_i * S_i$

这里W_i分别表示S_i的重要性，每个案例的总体相似度是通过各特征属性的权值计算实现的，获得总体相似度后，系统对候选案例进行排序，可以根据情况淘汰不合格的候选案例，从而使特征查询的检索效率更高. 淘汰策略可以采用设置阈值的方法，在检索时使用者给出阈值，系统将显示出大于阈值的所有历史案例. 阈值是一个可调节的量，如选出的候选案例较多，设计分析人员可以调小阈值; 如选出的候选案例较少，则可以增大阈值. 从而能够选出一个或几个与当前案例更加匹配的最佳案例，提供给维修性设计分析人员.

3 应用过程示例

实现CBR的问题求解可分为以下步骤: 1）案例的整理与表示; 2）案例的匹配与选择; 3）案例的修正与新案例的增添. 设计案例的提取和利用是其中的关键问题，设计对象属性很多，必须提取设计对象内在和外在的本质特征，充分反映设计对象的特征属性.

下面以电源系统中测试性设计为例，说明CBR应用.

案例的表示方法如下:

案例中的特征元素:

名称: 故障类型（例如: 开路）

功能块: 故障发生所在的功能块

失效模式: 对整个电源系统的影响

测试点:

测试参数: 需要的测试内容（例如: DC 电压）

故障信号特征: 指示的信号特征;

测试设备: 用到的测试设备仪表

通过建立的案例库，运用CBR的方法，帮助用户确定测试点及测试方法，把故障隔离到某一功能块. 该案例库中的案例可以从设计师或维修性、测试性专家那里获得. 这样，当有新的故障信息出现时，基于案例的推理提供基于部件或功能块的测试信息，实现故障隔离. 下面选取案例库中的几个案例说明案例的表示及求解过程.

	CASE 1	CASE 2
功能块:	整流器	整流器
故障:	二极管开路	二极管开路
故障模式:	负载下波动增强	频率减小
测试点:	整流器输出	整流器输入
测试参数:	峰间电压	频谱分析
测试信号特征:	峰间电压增强	频率是工作电流的1/2

用户通过输入有关的故障模式，与案例库中设计特征相比较，进行匹配与选择，最后得到测试点、测试方式，如果找不到匹配的案例，则把问题作为新的案例进行处理，并把它存入案例库.

在复杂的案例推理过程中，都涉及到相似度的计算问题，针对不同的问题域需采用不同的相似度算法.

4 结束语

CBR具有知识获取的容易性、与人类专家思维的贴近性、系统结构开放性、解释合理性并导致系统可接受性等优点，为人工智能领域中许多问题的解决带来了新的思路，是人工智能领域中较为看好的一种方法. 但将它应用于维修领域从目前掌握的资料来看，尚处于起步阶段，存在一些有待进一步研究的问题，主要包括:

1）维修知识组织和表示方法的研究 知识的表示方法目前已有多种形式，不同的知识表示方式各有其特点，同时也有其不足的一面. 目前存在的知识表示方法往往难以准确表示维修性设计分析中的维修知识，尽快找到一种既适合于维修知识表达又满足案例推理的知识表示方法，是维修性领域中应用基于案例推理所必须解决的一个问题.

2）相似度计算方法的研究 复杂的案例推理过程中，都涉及到相似度的计算问题，不同的问题需采用不同的相似度算法. 详细分析目前存在的相似度计算方法，以针对不同的问题采用不同的相似度计算方法，提高系统解的准确性，也是目前基于案例推理在维修性设计分析中应用需要认真研究的问题.

3）虚拟集成案例研究 如何集成被相似性度量法则所激活的多个案例，以生成一个同靶案例匹配得更好的虚拟集成案例（virtual integrated case）作为优先解，从而提高系统的适应能力，是另一个需进一步研究的问题.

参考文献

1 Hinkle D, Toomey C. Appling Case-based Reasoning to Manufacturing. AI Magzine,1995

2 史忠植. 高级人工智能. 北京: 科学出版社, 1998

3 何家峰, 叶虎年. 面向对象的专家系统的研究. 华中理工大学学报, 1999, 27（8）: 16-20

4 张治洪, 童 溶. 基于范例推理的结核病专家系统. 天津理工学院学报, 1997, 13（3）: 74-79

5 陈敏华. 圆柱齿轮传动设计专家系统. 南京理工大学学报, 1998, 22（5）: 402-406