应用模糊方法的设计模式挖掘策略研究

王宇新,田 佳,郭 禾,吴树朋,杨元生
WANG Yu-xin,TIAN Jia,GUO He,WU Shu-peng,YANG Yuan-sheng

大连理工大学 电子与信息工程学院,辽宁 大连 116024
Department of Electronic and Information Engineering,Dalian University of Technology,Dalian,Liaoning 116024,China
E-mail:wyx@dlut.edu.cn

WANG Yu-xin,TIAN Jia,GUO He,et al.Research on design pattern mining strategy based on fuzzy method.Computer Engineering and Applications,2010,46(2):150-153.

Abstract: Mining design patterns from source code is a very important technology for improving the intelligibility and maintainability of software.In this paper,a new matching method based on fuzzy is presented to mine design patterns.A matrix model is used to describe design patterns and source code as the basis of matching.Clustering method is adopted to optimize the source code model in order to improve the efficiency of matching.Combining fuzzy theory to patterns matching,this method introduces the static and dynamic information to enhance the accuracy of matching.Experimental results demonstrate the improvement of matching in accuracy and integrality,avoiding the invalidation for some special design patterns.

Key words: design pattern;fuzzy matching;pattern mining;matrix model;clustering

摘 要:从系统源码中挖掘设计模式对软件的可理解性和可维护性具有重要意义。基于模糊理论,提出一种模式匹配方法,实现设计模式挖掘。其中,使用基于类关系的素数矩阵模型对设计模式结构及源码信息进行描述,并作为匹配的模型基础;采用聚类方法对源码模型进行优化,提高匹配效率;将模糊方法与设计模式匹配策略相结合,引入静态和动态信息,提高匹配的正确性。实验结果证明此方法在精确性和完整性方面得到了很大的提高,并且避免了对特殊模式的失效性。

关键词:设计模式;模糊匹配;模式挖掘;矩阵模型;聚类
DOI:10.3778/j.issn.1002-8331.2010.02.045 **文章编号**:1002-8331(2010)02-0150-04 **文献标识码**:A **中图分类号**:TP302.1

1 引言

设计模式提供了模式结构中每个类的角色信息及各组成元素之间的关系,是面向对象设计的一个高级抽象。因此,对于规模和复杂度不断增加的软件系统,从其源码中挖掘设计模式对提高系统的可维护性和可理解性具有重要意义。对于缺少分析和设计文档的软件系统来说,研究设计模式挖掘技术也有利于软件系统的文档化。

设计模式挖掘的本质是一个模式匹配过程,匹配方法的选择直接影响到模式挖掘的质量。由于设计模式本身的描述特征,使得其匹配具有一定的模糊性和主观性,因此可以引入人类的模糊决策能力,将模糊方法与设计模式匹配策略相结合,以提高匹配的精确性和完整性。

2 相关工作

随着设计模式的不断发展与成熟,对设计模式的挖掘研究工作越来越多[1-6],在国际上已取得了普遍关注。

Jing Dong 等人提出一种基于矩阵和权重的匹配方法,实现设计模式挖掘[1]。用矩阵表示源码和设计模式的结构;用权重表示源码和设计模式的类关系,并作为矩阵中的值。该方法能够很好地挖掘结构型设计模式,但对于结构相同而动机不同的设计模式,只根据类名进行推断,无法正确反映模式的本质,因而不能得到精确的结果。

面向 Java 语言,冯铁等人在文献[2]中提出一种约束满足的匹配方法。该方法的缺点仍然是没有考虑动态信息,虽能较好地挖掘出行为型模式,但却不够精确。

N.Tsantalis 等人基于图顶点的相似值(similarity scoring)算法,将设计模式和源码用多个矩阵表示,并计算两矩阵的相似矩阵 $\boldsymbol{S}$,根据 S_{ij} 的值,挖掘设计模式实例[3]。S_{ij} 值越大表示设计模式类 i 与源码类 j 相似度越高。该方法的缺点是只能挖掘类与类的相似度,而无法从整体上实现挖掘目的。

针对文献[3]的缺点,Jing Dong 在文献[4]中提出一种基于模板匹配的改进方法 CCn(normalized cross correlation),从整体结构上实现对设计模式的挖掘。该方法用不同于文献[3]的矩阵描述设计模式和源码,再将两个待匹配的矩阵表示成向量,利用 CCn 方法计算向量的相似值,将相似值为 1 的矩阵所代表的源码作为设计模式实例。存在的缺陷是待匹配的矩阵维数必须

作者简介:王宇新(1973-),男,博士生,讲师,CCF 会员,主要研究领域为软件工程、计算机系统结构;田佳(1983-),女,硕士生,主要研究领域为软件工程、设计模式;郭禾(1955-),男,教授,博导,主要研究领域为计算机系统结构;吴树朋(1984-),男,硕士生,主要研究领域为软件工程;杨元生(1946-),男,教授,博导,主要研究领域为算法设计与分析。

收稿日期:2009-08-26 **修回日期**:2009-11-16

相同,矩阵维数的处理导致匹配算法复杂度比较高。另外,对于一些行为型设计模式,仍然无法正确识别。

以上方法均没有考虑动态信息,文献[5]则考虑将动态信息和静态信息相结合,实现设计模式挖掘。但没有对实验结果做详细分析,无法对该方法进行质量的评价。

综合考虑上述研究工作的优缺点,提出了一种基于模糊的匹配方法,实现设计模式挖掘。结合模糊数学的相关理论,将静态信息和动态信息作为因素,提高挖掘行为型设计模式的准确性。其中,利用基于素数表示的矩阵模型,实现对设计模式和源码信息的描述;采用聚类方法,对源码模型进行优化,提高匹配效率。通过对应用大量设计模式的开源代码 JUnit、JHotDraw 和 JRefactory 进行实验分析,证明该方法能更好地识别结构相同动机不同的设计模式,有效地提高了挖掘的精确性和完整性。

3　基于模糊的设计模式匹配方法

模式匹配是设计模式挖掘的核心所在。由于设计模式本身的描述特征,使得模式匹配具有一定的模糊性和主观性,因此可以引入人类的模糊决策能力,将模糊理论应用到模式匹配上。基于模糊理论[7],提出一种设计模式匹配方法,提高模式挖掘的完整性和精确性。

源码因素集 $\boldsymbol{X}=\{x_1,x_2,\cdots,x_t\}$,表示一组可以描述源码结构信息的因素。隶属度函数 $GMD(\boldsymbol{A})=\mu_A(x_i)=\max(\mu_A(x_1),\mu_A(x_2),\cdots,\mu_A(x_n))$,表示 $\boldsymbol{A}$ 相对隶属于 x_i。模糊匹配模型描述如下:

(1)确定设计模式和源码模型。

(2)对源码模型给出一个客观的因素集 $\boldsymbol{X}=\{x_1,x_2,\cdots,x_t\}$。

(3)依据所要挖掘的设计模式,对每个源码模型计算该源码因素的信任度向量 $\boldsymbol{T}_i=(t_{1i},t_{2i},\cdots,t_{ti})$。

(4)根据各因素的重要性,为各因素分配权重向量 $\boldsymbol{W}_i$。

(5)如果 $\mathrm{GMD}(\boldsymbol{T}_i\wedge\boldsymbol{W}_i)\geqslant$设计模式阈值,则源码模型 i 是该设计模式的一个实例。

3.1　设计模式模型的确定

目前设计模式描述方法均是半形式化方式。为了从源码中挖掘设计模式,需将其进行形式化描述。在 UML 与自然语言描述的基础上,使用基于素数的矩阵模型,实现对设计模式的形式化描述。

(1)基于矩阵的设计模式描述模型

GoF 按照设计模式的目的分为创建型、结构型和行为型三种。虽然目的不同,但基本信息都可以用类之间的关系约束来表示,这种关系很适合用矩阵表示[1,3-4]。由于类之间的约束关系复杂,代表类关系的矩阵值必须具有可识别性。按照文献[1]的思想,根据素数的乘积唯一性,采用素数表示不同的类关系,并将出现频率较高的元素用较低的素数表示。表 1 显示了不用类关系的素数值。

表 1　类关系的素数表示

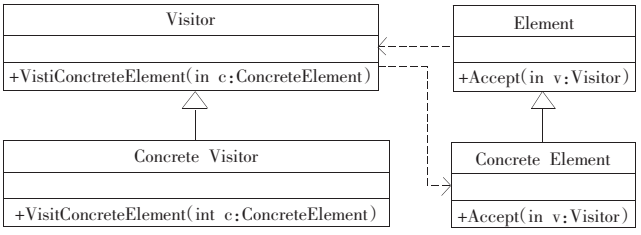
类关系	素数值
Generalization	2
Association	3
Dependency	5
Realization	7
Aggregation	11
Composite	13

在此基础上,矩阵中的各值可以用如下公式计算:

$$M(i,j)=\begin{cases}\prod_{n=1}^{N}V_n(i,j) & N\geqslant 1\\ 0 & N=0\end{cases}$$

其中,N 表示类 i 与类 j 之间的关系数目;$V(i,j)$表示类 i 与类 j 的关系值。

以图 1(a)的 Visitor 模式为例,其矩阵模型表示为图 1(b)。



(a)访问者模式

$$\begin{bmatrix}0&0&0&5\\5&0&0&0\\2&0&0&0\\0&2&0&0\end{bmatrix}$$

(b)访问者模式矩阵图

图 1　设计模式及其矩阵图表示

根据上述定义,把 Gamma 等人[8]提出的 23 个通用设计模式形式化成矩阵,并存储到可维护的设计模式模型库中。

3.2　源码模型的确定

设计模式挖掘并不是直接对源码进行挖掘,而是将源码转换成一种中间表示形式,而这种中间表示方式会影响到匹配方法的选择。

(1)源码模型的建立

为了更好地与设计模式矩阵相匹配,同样采用 3.1.1 节中的矩阵形式描述源码信息,并给出一种从源码到矩阵的转换方法,其流程图如图 2 所示。首先利用反向工具如 EA(Enterprise Architect)将源码生成 UML 图;然后表示成 XMI(XML Metadata Interchange)形式,再通过 XSLT 提取与设计模式挖掘相关的信息,生成 XML;最后生成 3.1.1 节中的矩阵模型。

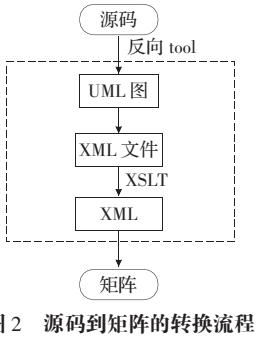


图 2　源码到矩阵的转换流程

按照上述流程,生成的矩阵是一个 $n\times n$ 的复杂矩阵(n 表示源码中类的个数)。显然挖掘某个设计模式时,源码的初始矩阵中会存在大量的冗余信息,严重影响到匹配算法的效率。因此,在挖掘设计模式之前,需要对初始矩阵做进一步的优化处理。

(2)模型优化

文献[4]中虽然对最初的矩阵进行优化,但效果并不是很理想,一定程度上影响匹配算法的速度。则采用基于模型的聚类方法进行优化,将设计模式矩阵作为参考模型,从源码矩阵中

寻找给定模型的最佳拟合。具体方法是按照设计模式中的类关系值,将源码矩阵中与该值相等或成倍数的值进行聚类。此方法有效地提高了匹配的效率。

以图 3 为例,要从中挖掘 Visitor 模式,需按照 Generalization 和 Dependency 类关系值聚类,聚类后的矩阵为 $\begin{bmatrix}0&0&0&5&5\\5&0&0&0&0\\2&0&0&0&0\\0&2&0&0&0\\0&2&0&0&0\end{bmatrix}$ 和 $\begin{bmatrix}0&0&0\\2&0&0\\2&0&0\end{bmatrix}$。但仍存在一些重复信息,如 $\begin{bmatrix}0&0&0&5&5\\5&0&0&0&0\\2&0&0&0&0\\0&2&0&0&0\\0&2&0&0&0\end{bmatrix}$ 中最后两行两列所表示的信息相同,即类 CarElement 和类 BusElement 作用相同。将其归一化处理,得到 $\begin{bmatrix}0&0&0&5\\5&0&0&0\\2&0&0&0\\0&2&0&0\end{bmatrix}$ 和 $\begin{bmatrix}0&0\\2&0\end{bmatrix}$。最终的矩阵确定为 $\begin{bmatrix}0&0&0&5\\5&0&0&0\\2&0&0&0\\0&2&0&0\end{bmatrix}$,这是由于 $\begin{bmatrix}0&0\\2&0\end{bmatrix}$ 的维数小于 Visitor 模式矩阵维数,所以它一定不是 Visitor 模式的候选实例而将其过滤。可以看出,聚类并优化后的矩阵几乎不包含冗余信息,而且匹配的维数处理次数由 C_8^4 缩减到 1,大大提高了匹配的效率。

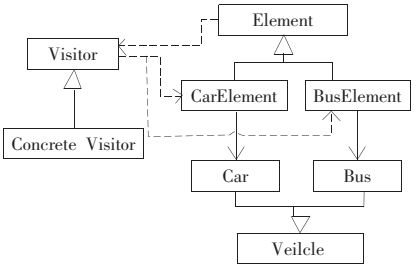


图 3 部分源码信息的 UML 图

3.3 源码信任度的计算

设计模式的结构信息主要包括类本身的信息、类之间的静态信息和类之间的动态信息。所以,定义源码的因素集 X={类信息,静态关系信息,行为信息}。则信任度向量中各因素的计算方法如下:

(1)类信息

将类本身的信息提取出来,主要提取类的类型(主要包括接口类、抽象类和一般类)信息,按照如下公式将其表示成 0-1 之间的值。

$$X_{\text{类信息}}=\begin{cases}\dfrac{x}{n} & x<n\\ 1 & x\geqslant n\end{cases}$$

其中,x 表示源码中抽象类的数目;n 表示设计模式中抽象类的数目。

(2)静态关系信息

文献[4]中采用 CCn 方法能够较好地挖掘结构型设计模式,在对源码矩阵进行优化的基础上,采用该方法可以更有效地匹配源码矩阵和设计模式矩阵的静态信息,将计算结果作为静态关系信息的信任度值。

(3)行为信息

对于结构相同而动机不同的设计模式,如 State 模式和 Strategy 模式,按照文献[4]的方法,由于它们的矩阵模型相同,导致匹配结果也相同,对两种模式无法正确区分。为了解决上述问题,将动态信息作为一个因素考虑进去。

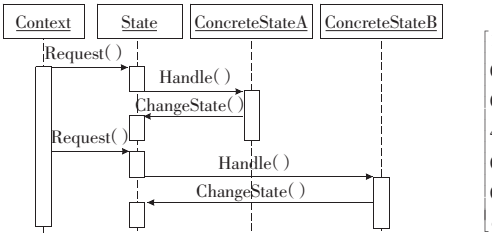
行为信息信任度值的计算方法如下:

(1)通过 UML 序列图获得设计模式和源码的动态信息;

(2)将序列图转换成矩阵:将序列图用 XMI 表示,再通过 XSLT 提取重要信息,形成 XML,最后转化成矩阵。矩阵的定义为:列表示参与类的个数;行表示任务数,其中第 i 行表示第 i 个任务。矩阵值由正数、负数、0 和 ∞ 表示,正数表示任务的调用者,负数表示任务的被调用者,0 表示没有调用与被调用关系,∞ 表示一个生命期结束;

(3)利用 CCn 方法计算两矩阵的相似度,作为行为信息信任度值。

以图 4(a)所示的 State 模式序列图为例,按照上面的定义,其矩阵可以表示为图 4(b)。



(a)State 模式序列图

$$\begin{bmatrix}1&-1&0&0\\0&2&-2&0\\0&-3&3&0\\4&-4&0&0\\0&5&0&-5\\0&-6&0&6\\\infty&\infty&\infty&\infty\end{bmatrix}$$

(b)序列图的矩阵表示

图 4 State 模式序列图及其矩阵表示

4 实验结果分析

开源项目 JUnit、JHotDraw 和 JRefactory 中应用了很多著名的设计模式,很多设计模式挖掘的文章都以它们作为评价的标准。同样对这 3 个开源项目进行设计模式挖掘来验证方法的有效性。表 2 列出部分行为型设计模式的挖掘结果,并将其与文献[3-4]进行了比较。表中可以看出,该方法能很好地区分 State

表 2 实验结果

设计模式	JUnit			JHotDraw			JRefactory		
	该文	文献[3]	文献[4]	该文	文献[3]	文献[4]	该文	文献[3]	文献[4]
Version	v4.7	v3.7	v3.8	v6.0	v5.1	v6.0	v2.9.18	v2.6.24	v2.6.24
State	0	3	3	12	22	29	7	11	12
Strategy	3			17			15		
Template Method	1	1	×	5	5	×	19	17	×
Observer	4	4	×	3	5	×	1	0	×
Visitor	0	0	×	1	1	×	2	2	×

和 Strategy 模式,同时对行为型设计模式的挖掘在精确性和完整性方面也有很大的提高。

以图 5 所示的 JHotDraw 部分源码图为例,分析该文方法的有效性。它表示的是一个 State 模式实例。按照文献[4]方法,对其分别进行 State 模式和 Strategy 模式挖掘。由于两个模式的静态结构相同,即矩阵表示相同,匹配计算得到的结果也均为 1,所以挖掘的结果为该部分源码既是 State 模式又是 Strategy 模式。

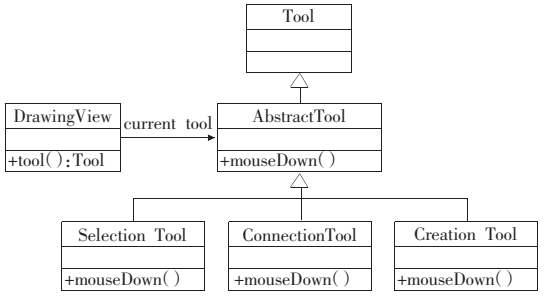


图 5　JHotDraw 的源码部分实例

按照该文的方法,根据 3.1.1 节和 3.1.2 节,可以得到 State 模式矩阵和 Strategy 模式矩阵均为 $\begin{bmatrix}0&3&0\\0&0&0\\0&2&0\end{bmatrix}$,优化后的源码矩阵为 $\begin{bmatrix}0&3&0&0\\0&0&0&0\\0&7&0&0\\0&0&2&0\end{bmatrix}$。按照 3.1.3 节的计算方法,得到 $\boldsymbol{T}_{1State}=1.0$,$\boldsymbol{T}_{2State}=0.97$,$\boldsymbol{T}_{3State}=0.81$。其中,只考虑一个周期的行为信息,因此对源码序列图矩阵无需考虑生命期,矩阵表示为 $\begin{bmatrix}1&-1&0&0&0\\0&2&-2&0&0\\0&-3&3&0&0\\4&-4&0&0&0\\0&5&0&0&-5\\0&-6&0&0&6\\7&-7&0&0&0\\0&8&0&-8&0\\0&-9&0&9&0\end{bmatrix}$。从而得到 State 模式下源码的信任度向量为 $\boldsymbol{T}_{State}=(1.0,0.97,0.81)$。同理,可以得到 Strategy 模式下源码的信任度向量 $\boldsymbol{T}_{Strategy}=(1.0,0.97,0.23)$。

令权重 $\boldsymbol{W}_{State}=(0.3,0.8,0.7)$,$\boldsymbol{W}_{Strategy}=(0.3,0.7,0.6)$,则 $GMD(\boldsymbol{T}_{State}\wedge\boldsymbol{W}_{State})=0.81$,$GMD(\boldsymbol{T}_{Strategy}\wedge\boldsymbol{W}_{Strategy})=0.7$。根据专家经验,State 模式的阈值为 0.75,Strategy 模式的阈值为 0.79。因此,$GMD(\boldsymbol{T}_{State}\wedge\boldsymbol{W}_{State})=0.81>0.75$ 且 $GMD(\boldsymbol{T}_{Strategy}\wedge\boldsymbol{W}_{Strategy})=0.7<0.79$,即可以得出该源码是 State 模式实例。

5　结论

在文献[4]的基础上,提出了一种基于模糊的模式匹配方法实现设计模式挖掘。该方法引入模糊的概念,将设计模式的静态结构和动态信息作为因素,实现设计模式与源码的匹配。另外,使用了一种基于素数的矩阵模型,实现对源码和设计模式的形式化描述;利用聚类方法,对源码矩阵模型进行优化,很大程度上提高了匹配效率。实验结果证明此方法能够更好地识别特殊的行为型模式,在精确性和完整性方面得到了很大的提高。

参考文献:

[1] Dong Jing,Lad D S,Zhao Ya-jing.DP-Miner:Design pattern discovery using matrix[C]//Leaney J.Proceedings of 14th Annual IEEE International Conference and Workshops on Engineering of Computer-Based Systems(ECBS).Tucson USA,Washington:IEEE Computer Society,March 26-29,2007:371-380.

[2] 冯铁,李文锦,张家晨,等.面向 Java 语言的设计模式抽取方法的研究[J].计算机工程与应用,2005,41(25):28-33.

[3] Tsantalis N,Chatzigeorgiou A,Stephanides G,et al.Design pattern detection using similarity scoring[C]//IEEE Transactions on Software Engineering.Piscataway:IEEE Computer Society,November 2006,32(11):896-909.

[4] Dong Jing,Sun Yong-tao,Zhao Ya-jing.Design pattern detection by template matching[C]//Proceedings of the 23rd Annual ACM Symposium on Applied Computing(SAC 2008).Fortaleza Brazil,New York:ACM,March 16-20 2008:765-769.

[5] Li Fan,Li Qing-shan,Su Yang,et al.Detection of design patterns by combining static and dynamic analyses[J].Journal of Shanghai University,2007,11(2):156-162.

[6] Kaczor O,Guéhéneuc Y G,Hamel S.Efficient identification of design patterns with bit-vector algorithm[C]//Proceedings of the 10th European Conference on Software Maintenance and Reengineering.Bari Italy,Washington:IEEE Computer Society,March 22-24,2006:175-184.

[7] 胡宝清.模糊理论基础[M].武汉:武汉大学出版社,2004.

[8] Gamma E,Helm R,Johnson R,et al.Design patterns:Elements of reusable object-oriented software[M].[S.l.]:Addison Wesley Publishing Company,1995.

(上接 120 页)

[2] Hooper J W,Chester R O.Software reuse and methods[M].New York:Plenum Press,1991.

[3] de Lucena V F.Facet-based classification scheme for industrial automatiom software components[C]//Sixth International Workshop on Component-Oriented Programming at ECOOP 2001,Budapest,Hungary,2001.

[4] 王渊峰,薛云皎,张涌,等.刻面分类构件的匹配模型[J].软件学报,2003,14(3).

[5] 王渊峰,张涌,任洪敏,等.基于刻面描述的构件检索[J].软件学报,2002,13(8).

[6] Shasha.AtreeGrep-approximate searching in unordered trees[C]//Proceedings of SSDBM 2002,Edinburgh,July 2002:89-98.

[7] Zhuge Hai.An inexact model matching approach and its applications[J].Journal of Systems and Software,2003,67:201-212.