形式概念分析在软件开发中的应用

翟正国

（大连海事大学，信息科学与技术学院，1120161212）

摘要：近十多年，在软件开发领域，形式概念分析技术已得到了广泛的认可，尤其在软件开发的各项活动中得到了广泛的应用。形式概念分析技术是能够从二元关系中挖掘出具有共同形式属性的一组形式对象的聚集,是一种层次化的形式对象分析方法。相对而言，传统意义上的软件开发因为过分地依赖于文档,其开发效率及灵活性受到了很大程度的制约。随着现代软件开发的不断发展,软件开发已越发呈现出效率优先的趋势。因此，如何高效地开发出满足用户多样化需求的软件，是当今软件开发领域的热点问题。文中描述了基于形式概念分析方法在软件开发领域中的应用,并在其中寻找到共同特征，这些共同特征将找寻与其相关的对象，并以一种表现所有特征局部关系的概念格的形式呈现出来,进而更合理地设计和规划软件开发的各项进度,大大提高软件开发的效率。

关键词：形式概念分析；概念格；软件开发

Using Formal Concept Analysis for Software Development

Zhai Zhengguo

(Dalian Maritime University, College of Information Science and Engineering,1120161212)

Abstract:In the field of software development, formal concept analysis technology has been widely recognized, especially in the software development activities have been widely used. Formal concept analysis technology is a kind of hierarchical formal object analysis method, which can extract a set of formal objects with common form attributes from binary relations. Relatively speaking, the traditional sense of software development because of over-reliance on the document, its development efficiency and flexibility have been largely restricted. With the continuous development of modern software development, software development has become more and more the trend of giving priority to efficiency. Therefore, how to efficiently develop software to meet the diverse needs of users, is a hot issue in the field of software development. This paper describes the application of formal conceptual analysis in the field of software development, in which common features are sought, which will find objects related to them and present them in the form of a concept lattice that expresses all the local features of the feature , And then more rationally design and planning the progress of software development, greatly improving the efficiency of software development.

Key words: formal concept analysis;concept lattice;software development

0 引　言

随着时代的发展，软件开发项目也变得越发复杂，需要成立专项开发小组，以及合理的开发技术和开发方案才能完成。因此 ，寻找到科学合理的软件开发方法能够提高开发效率，加快项目进度。以下将介绍形式概念分析方法在软件开发实践过程中的应用,以及提高软件开发效率的最佳方法。

形式概念分析(Formal Concept Analysis,FCA)是建立在数学基础之上 ,对组成软件本体的概念、属性以及关系等用形式化的语境表达出来,然后根据语境,构造出概念格(Concept Lattice),从而清楚地表达出本体的结构。那么什么是形式概念分析方法 ?它是如何应用在软件开发中的呢?下面将做详细的介绍。

1 形式概念分析

形式概念分析[1]建立在概念和概念层次的数学化基础之上，是应用数学和格论的一个分支。一个概念就是最大限度地收集对集

合中共同特点有帮助的元素,并且运用形式概念分析的方法,可以发现、构造和展示由属性(Attributes)和对象(Objects)构成的概念(Concept)及其之间的关系。因而,形式概念分析的方法已经运用在软件开发等众多环节之中。

1.1 相关概念及定义

定义 1 [2]:一个形式化的语境(Context)k =(G , M ,I),包括两个稽核(G 和 M)和一个二元关系 。在这个语境中, G 中的元素称为对象, M 中的元素称为属性。一般用 g I m,或者(g,m)∈ I 来表达对象 g 和属性 m 的关系,读作"对象 g 具有属性 m"。根据定义 1,可以通过矩阵来表示语境。每行的开头是对象名,每列的开头是属性名。行 g 和列 m 的交叉表示对象 g 具有属性 m,见表 1。

表 1 语境的矩阵表示

属性 / 对象	m	m	m	m
g	I			
g		I		
g			I	

定义 2:对一个对象集 A,定义 A'={m∈M|g I m,对所有的 g∈A}(即 A 中所有的对象共有的属性集合)。相应地,对一个属性集 B,定义 B'={g∈G|g I m,对所有的 m∈B}(即包含 B 中所有的属性的集合)。

定义 3:语境(G,M,I)中的形式概念(Formal Concept)是个集合对(A,B),其中 AG,BM,并且 A'=B,B'=A。A,B 分别称作概念(A,B)的外延(Extent)和内涵(Intent)。β(G,M,I)表示语境(G,M,I)中的所有概念集。

定义 4:如果(A1,B1),(A2,B2)都是语境中的概念,并且 A1A2,那么(A1,B1)被称作(A2,B2)的子概念(Sub concept),(A2,B2)则是(A1,B1)的超概念(Super concept),记为(A1,B1)≤(A2,B2)。"≤"反映了概念间的层次关系。由层次关系搭构的所有(G,M,I)的概念记作 β (G,M,I),被叫作概念格[2](Concept Lattice)。

线路图[3](Line Diagram)是概念格的图示化表示。线路图包含语境中对象、属性之间的关系,是语境的一个等价表示形式。在特定的语境中,包含类的继承。通过查看线路图,能够容易地发现属性之间的依赖和关联。

1.2 形式概念分析方法的思路解析

形式概念分析方法只是一种数学分析方法,运用到具体的软件分析过程中,可以通过以下环节来实现在软件开发中的应用。如图 1 所示。

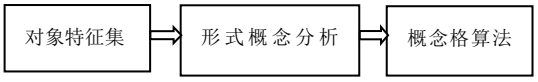

图 1 形式概念分析的总体思路

根据图 1,软件对象特征集的构造即是通过对软件需求分析后所要实现的各项功能的一个系统分析总结,从而抽象表达出各种具体特征集合。同时,将各特征集合运用形式概念分析方法,构造出一些具体的概念,并将这些概念抽象地表达为具体的类的形式。最后,通过分析概念类之间的各项联系,引入概念格[2](Concept Lattice,CL)的形式,由具体的构造方法构造出系统的分析概念格图。最终,根据这些分析,运用到系统设计的各个阶段,提高软件项目的开发效率。

2 形式概念分析在软件开发中的应用

形式概念分析在各个行业领域都得到应用,以下按照图 1 所示思路,描述其在具体的软件开发的一些开发环节中是如何应用的。

2.1 在需求分析中的应用

需求分析[1](Requirements Analysis,RA)主要是针对软件的各项需求进行具体的分析,而形式概念分析应用在软件开发需求分析阶段主要是实现图 1 中的第一步——构造软件项目特征集。而这个过程也就是上述对 FCA 讲述中的语境及属性集的构造。如表 2 所示。

表 2 对象属性集

属性 / 对象	需求分析	结构设计	详细设计	编码	测试	软件维护
调试					X	X
用例			X			X
帮助	X		X			X
环境			X		X	X
控制	X					

表 2 中:“×”表示连接对象和属性的符号,比如说(调试,测试)可以组成一个对象-属性集,表示了其在软件开发中操作的优先级别和对象属性。同时,对于其他部分的开发工作,也可以通过对其具体的特征集合进行收集通过对软件开发中各个传统模式下所对应的部分工作集和其属性的对应关系,运用形式分析的方法可以得到表 2 所示的项目特征集合,从而也为软件的需求分析奠定了基础。

2.2 在结构设计中的应用

软件的结构设计主要是针对软件的数据结构模式,在需求分析的基础之上,通过合理化组织,加以分析设计,从而得出设计的方法。在形式概念分析方法中表现为通过表 3 中的对象-属性集构造出具体的概念[4]。概念分析算法[3]如下:

```
K=1;//当前的概念的个数
For(i=1;i<=5;i++)
{
    For(j=1;j<=14;j++)
     {
        If ((Gi,Mj)∈I)
        Fk=Fk ∪ Gi;//如果对象 Gi 具有属性 Mj,则把 Gi 添加到 Fk 中
     }
}
PK=(Fk,Mj);//形成一个概念
K++;
}
```

现在通过一个假设的系统来说明形式概念分析方法的概念构造原理[7]。这个系统包括 4 个属性(分别假设为 A、B、C、D)以及一个包含 5 个对象的记录。利用相关的分析器可以分析出每个项目在每一个过程中的使用情况(“×”表示使用),如表 3 所示

表 3 对应关系表

属性 / 对象	A	B	C	D
a	X			
b		X	X	
c	X			
d				
e				X

表 4 表 3 数据所对应的集合

概念	项目集合	特征集合
P_1	{a,c}	{A}
P_2	{b}	{B}
P_3	{b}	{C}
P_4	{e}	{D}
P_5	空集	{A,B}
P_6	空集	{A,C}
P_7	空集	{A,D}
P_8	{b}	{B,C}
P_9	空集	{B,D}
P_{10}	空集	{C,D}
P_{11}	空集	{A,B,C}
P_{12}	空集	{A,C,D}
P_{13}	空集	{B,C,D}
P_{14}	空集	{A,B,C,D}
P_{15}	{a,b,c,d,e}	空集

2.3 在系统设计中的应用

形式概念分析应用在软件开发系统设计阶段所要完成的任务主要是构造系统的概念格。这里使用一种自底向上的计算方式计算出概念格。

概念格构造算法[3](CLCA)如下, 这里使用一种自底向上的计算方式计算出概念格。

变量说明:

N:语境中单个对象具备属性的最大数目,即是格中的继承层次数目,但不包括上下界所占的层次。

A:集合,其中元素为语境中单个属性或多个属性的集合。

Node:本体节点集。Node(i, j)表示第 i 层上

的第 j 个节点。

算法:

(1)计算出 N;

(2)for(int i=N;i≥1;i--)

{

计算出 A;//A 中元素是属性集,这些元素中包含

的属性个数为 i;

```
for(j=1;j<size of(A);j++)
  {
    a=A′ (j);
  b=A(j);
  if(a'==b&&a==b')
{
Node(i,j).attribute list=b;
Node(i,j).object list=a;
```

判断该 attribute list(即 b)是否包含于下层节点的 attribute list 中,确定节点间父子关系;

}

}

}

通过各个集合之间的关系可以得出相应的概念之间的关系,进而得到系统概念格。将表 1 的数据,通过如表 2 所示的方法,得到如表 3 的数据作为概念分析算法的输入数据,可以得到相对应的概念格。如图 2 所示。

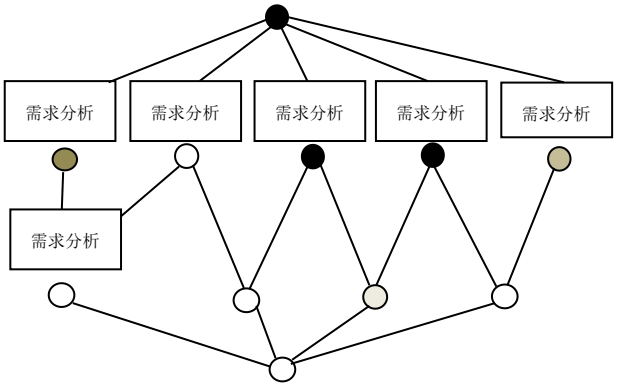

图 2 系统概念格

从图中对象的数量和节点的颜色可以看出,系统对于每个对象的事件分配是不同的。颜色最暗的表示最需要紧急处理的事件或者过程,稍微暗些的表示可以通过其它环节进行预先或者推迟处理的过程,最后则是按照系统正常开发次序进行完成的工作。

3 结语

在介绍了形式概念分析方法之后,在遵循概念分析方法的总体思路的基础上,在软件开发的各个工作阶段的相关应用上进行了详细的说明。可以看出,软件开发中 FCA 的运用可以很好地改善开发效率,它清楚地表达了软件和概念之间的关系。

参 考 文 献

[1] 蒋平.形式概念分析在软件工程中的应用[J].计算机技术与发展,2008.

[2] 胡可云，陆玉昌，石纯一. 概念格及其应用扩展[J]. 清华大学学报：自然科学版，2000.

[3] 谢润.概念格建格算法研究[D].西南交通大学，2006.

[4] 刘树鹏，李冠宇.基于形式概念分析的本体合并方法[J].计算机工程与设计，2011(04):09-12.

[5] 乌弘毅，黄应辉.模糊改您格构建的 BORDAT 方法[J].计算机技术与发展，2010(10):15-21.

[6] 蒋平.概念格构建系统的设计与时间[J].计算机技术与发展，2010(10):01-05.

[7] Stumme G, Maedche A.FCA-merge:bottom-up merging of ontologies[C]//17th Artificial Intelligence(IJCAI'01).Germany:Springer, 2001 :225-230.

[8] Uschold M.Ontologies:Principles, Methods and Applications[J] .The Knowledge Engineering Review , 1996, 11(2):93-120.

[9] Tilley T, Cole R, Becker P, etal.A Survey of Formal Concept Analysis Support for Software Engineering Activities[C]In:Stumme G.Proceedings of the First International Conference on Formal Concept Analysis - ICFCA'03.[s.l.] :[s. n.] , 2003:119 -124 .