形式概念分析在软件工程中的应用

蒋 平,任胜兵,林 鹃
(中南大学 信息科学与工程学院,湖南 长沙 410083)

摘 要:随着现代软件工程的不断发展,软件开发的效率显得尤为重要。如何高效率地开发出满足各种不同用户需求的软件已成为当今软件工程开发中的热点。传统的软件开发由于过分地依赖于文档,其开发效率及其灵活性受到了很大的影响。文中描述了形式概念分析方法这种软件工程中新型的设计方法,用以发现一群类所表现出的共同或者重复的特征。这些共同特征将通过一种表现所有特征局部关系的格的形式,发现跟其相关的那些关联,进而更合理地安排软件开发的各项进度,提高软件开发的效率。文中也描述了形式概念分析方法在软件工程一些具体阶段的应用。

关键词:形式概念分析;概念格;软件工程
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1673-629X(2008)04-0127-03

Using Formal Concept Analysis for Software Engineering

JIANG Ping, REN Sheng-bing, LIN Juan
(Academy of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract: With the quick and continuous development of the modern software engineering, the efficiency of software seems to be very important. How to develop high-efficient software, which can satisfy various different customers, has become a hot point within the software engineering development nowadays. Because of being too depended on documents, the basic mode of the tradditonal software engineering is influenced by its efficiency and vivids. This text has described a new design method which is called the formal concept analysis, it is being used to describe a kind of design method of software engineering. The characters in common will use the form of lattice, thus let people discover its related connections among them, then planning the time appropriately, which could increase the efficiency of software development. In this text, also introduce its application in some parts of the software engineering.

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

0 引 言

随着计算机的发展,现在的软件工程开发项目变得十分复杂和庞大。这样的项目已经不是一个人可以独立开发的,因为它需要使用大型数据库、网络传输,需要组织项目组,需要合理的开发技术和方案等。因此,软件开发的合理技术就显得很重要。以下将介绍一种运用概念分析方法在软件工程中的应用,从而提高软件开发的效率。

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),即本体,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工程的开发中呢?下面将详细介绍。

收稿日期:2007-07-12
作者简介:蒋 平(1983-)男,湖北安陆人,硕士研究生,主要研究遗留系统改造及软件建模;任胜兵,硕士生导师,副教授,主要研究软件理论及其演化重构。

1 形式概念分析

形式概念分析[1]是应用数学和格论的一个分支,它建立在概念和概念层次的数学化基础之上。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1:一个形式化的语境(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 gIm,或者$(g,m)\in I$ 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

对象 / 属性	……	m	……	……
……	……	……	……	……
g	……	X	……	……
……	……	……	……	X

定义 2:对一个对象集 A,定义 $A' = \{m \in M \mid gIm$,对所有的 $g \in A\}$(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 $B' = \{g \in G \mid gIm$,对所有的 $m \in B\}$(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 $A \subseteq G, B \subseteq M$,并且 $A' = B, B' = A$。A,B 分别称作概念 (A,B) 的外延(Extent)和内涵(Intent)。$\beta(G,M,I)$ 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 $A1 \subseteq A2$,那么 $(A1,B1)$ 被称作 $(A2,B2)$ 的子概念(Subconcept),$(A2,B2)$ 则是 $(A1,B1)$ 的超概念(Superconcept),记为 $(A1,B1) \leqslant (A2,B2)$。"$\leqslant$"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 $\beta(G,M,I)$,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工程中的应用。如图 1 所示。

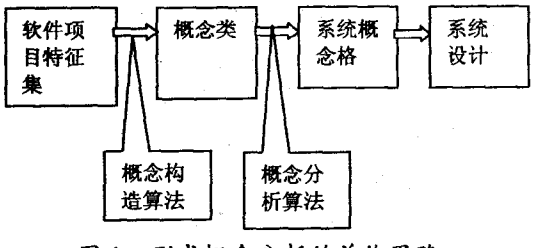

图 1 形式概念分析的总体思路

根据图 1,软件项目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[4](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 形式概念分析的应用

形式概念分析在各个行业领域都得到应用[5],以下按照图 1 所示思路,叙述其在具体的软件工程的一些开发环节中是如何应用的。

2.1 需求分析中的应用

需求分析[6](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工程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

对象 / 属性	需求分析	结构设计	详细设计	编码	测试	综合	资格测试	安装	认同支持	软件维护
调试					×					×
概念					×					×
用例			×							×
识别	×	×								
再工程			×							×
控制	×									
帮助	×									
环境			×							×

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,表示了其在软件工程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。

通过对软件工程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,运用概念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 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 1 中的对象 - 属性集构造出具体的概念[1]。

构造概念的算法[2]如下:

```
K=1; //当前的概念的个数
For(i=1;i<=N;i++)
{for(j=1;j<=N;j++)
If(Mj,Fi)∈T) mk = mkUMj; //如果过程 Fi 使用了变量
```

M_j,则把 M_j 添加到 m_k 中

}

$C_k=(m_k,F_i)$; //形成一个概念

K++;

For(n=I;n>=1;n--) //检验概念是否重复或者可以合并

{if(C_k<= C_n){delete C_k;k--;break;}

Else if(m_k= m_n)

{C_n= C_n V C_k;delete C_k;k--;break;}

}

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 5 个过程(分别假设为 P_1、P_2、P_3、P_4、P_5、P_6)以及一个包含 9 个字段的记录。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　9 个项目与 6 个特征的对应关系

	P_1	P_2	P_3	P_4	P_5	P_6
NAME	×					×
TITLE	×					
INTTIAL	×				×	
PREFIX	×				×	
NUMBER				×		×
NUMBER-EXT				×		×
ZIPCD				×		×
SIREET			×	×		
CITY		×		×		

通过概念分析方法得出表 3 对应的概念,如表 4 所示。

表 4　表 3 数据所对应的概念

概念	项目集合	特征集合
顶端	{NA,T,I,P,NU,NE,Z,S,C}	空集
C_1	{NA,T,I,P}	{P_1}
C_2	{NU,NE,Z,S,C}	{P_4}
C_3	{NA,NU,NE,Z}	{P_6}
C_4	{I,P}	{P_1,P_5}
C_5	{NA}	{P_1,P_6}
C_6	{C}	{P_2,P_4}
C_7	{S}	{P_3,P_4}
C_8	{NU,NE,Z}	{P_4,P_6}
尾端	空集	{P_1,P_2,P_3,P_4}

接下来的工作是分析这些变量的相关性,分析各个项目特征以形成概念,进而形成系统的概念格。

2.3　系统设计中的应用

形式概念分析应用在软件工程系统设计阶段所要完成的任务主要是构造系统的概念格。比较著名的概念格构造算法有 Ganter's Next-Concept Algorithm[3], Christian Linding 的 Fast Concept Analysis[8]等,这些算法计算出了格的所有概念以及层次关系。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。

A:集合,其中元素为语境中单个属性或多个属性的集合。

Node:节点集。Node(i,j)表示第 i 层上的第 j 个节点。

算法:

计算出 N:

For(int i = N;i >= 1;i--)

{

计算出 A:

For(j = 1;j < sizeof(A);j++)

{a = A'(j);

B = A(j);

If(a' = b&&a = b')

{Node(i,j).attributelist = b;

Node(i,j).objectlist = a;

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

}

}

}

分析各个项目集合之间的关系是概念分析算法的核心,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。

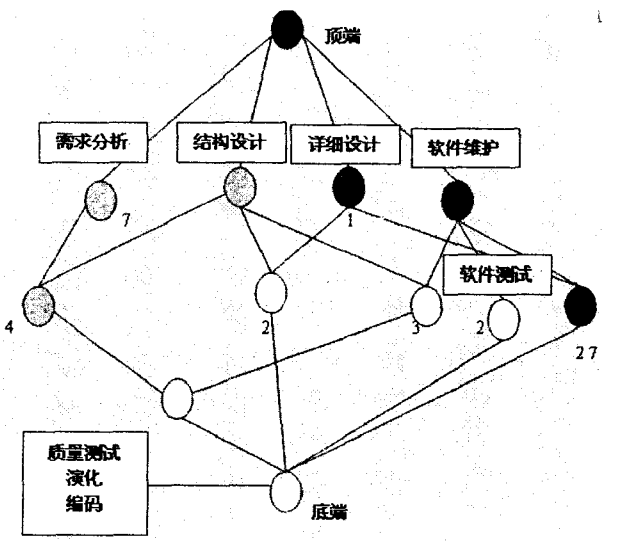

图 2　系统概念格的构造

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。

(下转第 213 页)

是由哪些基本度量经过运算得出来的;

c) 将基本度量映射到企业信息模型中的数据项。

通过对企业指标的分析,获得与指标相关的度量定义,将这些度量定义映射到企业信息模型中的信息项,如果现有企业信息模型不包括这些信息的描述,对该信息模型进行扩展,加入对这些基本信息项的描述。

4 结　语

企业过程度量是对企业过程进行改进和优化的基础。分析了企业过程模型与信息模型的特点,提出一个企业度量信息模型,实现了企业过程模型与信息模型之间的映射,从而实现从现有信息库中获取信息来实现企业过程的度量。阐述了度量信息模型在衔接过程模型与信息模型的作用和意义,简单描述了度量信息模型实现方法,为企业过程模型和信息模型的集成提供了一种机制。

参考文献:

[1] 陈雪松.企业软件过程度量实践[J].清华大学学报:自然科学版,2003,23(9):37-43.

[2] Basili V R,Caldiera G,Rombach H D. The goal question metric paradigm[J]. Encyclopedia of Software Engineering,1994,2(1):528-532.

[3] 李亚红,郝克刚,葛　玮.基于GQM模型的软件项目进度的度量过程[J].计算机应用,2005,25(6):1448-1450.

[4] 陈祖荫,刘建丽.GQM软件度量模式的某些决策问题[J].北京工业大学学报,2000,26(2):45-48.

[5] 杨基平,余忠华.GQM方法及其应用研究[J].制作业自动化,2003,25(6):20-23.

[6] 立　春,李建奇.度量软件过程——改进软件过程[J].软件世界,2002,20(4):132-137.

[7] 李育泽.我国企业信息化现状与发展趋势浅析[J].今日湖北理论版,2007,1(1):9-10.

[8] 葛世伦.企业信息模型研究[J].华东船舶工业学院学报:自然科学版,2001,15(3):73-78.

[9] John M,David C,Chery J,et al. Practical Software Measurement: Objective Information for Decision Makers[M]. [s. l.]:Addison Wesley/Pearson,2002.

[10] Briand L C,Differding C M,Rombach H D. Practical Guidelines for Measurement-Based Process Improvement[J]. Software Process Improvement and Practic,1996,2(4):253-280.

(上接第129页)

3 结　语

在介绍了形式概念分析方法之后,在以概念分析方法的总体思路的基础上,在具体的软件工程各个工作阶段的相关应用进行了详细的说明。可以看出,软件工程中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。然而,在软件开发的其它阶段,如何更合理更优化地运用形式概念分析是今后本课题主要的研究发展方向。

参考文献:

[1] Wolff K E. A First Course in Formal Concept Analysis-How to understand line diagram[J]. Statistical Software ,1993,14(4):429-438.

[2] Carpineto C,Romano G. A lattice conceptual clustering system and its application to browsing retrieval[J]. Machine Learning,1996,24(2):95-122.

[3] GanterB, Wille R. Formal Concept Analysis: Mathematical Foundationgs[M]. Berlin:[s.n.],1999.

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

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

[6] Tilley T,Cole R,Becker P,et al. 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.

[7] Cui Zhan,Jones D M,O'Brien P. Issues in Ontology-based Applications[J]. SIGMOD Record,2002,31(1):43-48.

[8] Lindig C. Fast Concept Analysis[EB/OL]. 2002. http://www.st.cs.uni-lindig.pdf.

(上接第209页)

时传输方法[J]. 计算机应用,2007(3):703-705.

[4] 郎　锐. VC++实现对远程计算机屏幕的监视[EB/OL]. 2006. http://www.easytu.com/software/program/VC++/2006-10-27/14216.html.

[5] 张友生.远程控制编程技术[M].北京:电子工业出版社,2002.

[6] 罗　红,慕德俊.桌面图形图像序列压缩与传输研究[J].计算机应用,2005(6):1299-1304.