在软件维护中形式概念分析的应用

章婕

（大连海事大学 计算机科学与技术 辽宁省大连市 中国 116026）

摘 要 形式概念分析是一种有效的数据挖掘工具，它是一种层次化的形式对象分析方法，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。近十几年来，形式概念分析技术已在软件工程领域，特别是软件维护的各项活动中得到了广泛的应用，并取得成功。本文从软件维护的角度，如软件理解、修改影响分析、重构、调试与测试等方面总结了从 2000—2013 年形式概念分析在这些领域的研究进展。这些研究成果的分类方法是基于一种软件维护活动框架进行论述，最后文章给出了形式概念分析在软件维护领域的研究趋势与展望。

关键词 形式概念分析；软件维护

中图法分类号 TP311.90 DOI 号 10.3969/j.issn.1001-3695.2015.04.033

The application of formal concept analysis in the software maintenance

Zhang Jie

(Computer science and technology，Dalian maritime university, LiaoningDalian,116026,China)

Abstract Formal concept analysis(FCA)is an effective tool for data mining.It is all approach to identify conceptual structures among data sets．It is a very useful clustering technique providing a formal framework to explore the binary relationship among the entities and recognize groups of entities that exhibit common properties.This paper tries to survey the studies of the FCA applications in software maintenance field from 2000 to 2013． These research results are categorized according to a simple framework of software maintenance activities，such as software comprehension,change impact analysis，refactoring，debugging and testing．Finally，the potential research directions in the related field are discussed．

Key words formal concept analysis(FCA)；software maintenance；

1 引言

形式概念分析(Formal Concept Analysis，FCA)是建立在数学的基础之上的，它是一种基于形式背景下进行数据分析和规则提取的有效工具。Ganter 等人将其作为一个较好的数据分析方法，深化、完善该理论基础，并将它们扩展到各种现实应用中[1]。形式概念分析提供了一种较好的层次化(形式)对象的分析方法，它能够识别那些具有共同(形式)属性的一组(形式)对象的组合[1]。一方面，形式概念分析已具备较完善的理论基础，在应用到软件工程，特别是软件维护活动中时，具有形式化方面的表达能力，从而具有较强的理论和技术说服力；另一方面，形式对象和形式属性这种二元关系经常出现在软件世界中，对这种二元关系处理的方法也推动形式概念分析技术在软件领域应用的不断发展．形式概念分析技术能够有效地对已有代码和文档进行建模和分析，生成一种图形化的格，提高软件代码和文档的抽象层次，从而辅助软件维护和测试人员对已有代码和文档的理解。因此，目前形式概念分析在软件工程中的各种成功的应用主要是在软件维护的诸多活动中，包括逆向工程、重构、程序理解等[3,4]。本文主要对这些研究工作进行介绍。

章婕，女，1995 年生，学士，主要研究领域数据库方向

2 基本概念

形式概念分析提供一种聚类具有共同形式属性的形式对象的方法，其输入是形式背景，输出是具有偏序(层次)关系的概念格。本文为区分形式概念中的对象、属性与软件中对象以及属性，将形式概念分析中的对象称为形式对象，而属性称为形式属性。一些相关定义表示如下[1]：

定义 1 形式背景 通常定义为一个三元组(0，A，R)，其中 0 是形式对集合，A 是形式属性集合，R 是 0 和 A 之间的二元关系，满足 R⊆0×A。形式概念分析中另外一个重要的概念是形式概念。给定某个形式背景后，只需满足一定的条件就可得到该形式背景上的形式概念，定义如下：

定义 2 形式概念 形式概念表示具有共同形式属性的形式对象的最大集合，定义为二元组(X,Y)，并且满足这样的关系：

$\tau(A)=X \wedge \sigma(O)=Y$，其中：

$\tau(A)=\{o \in O \mid \forall a \in A:(o,a) \in R\}$

$\sigma(O)=\{a \in A \mid \forall o \in X:(o,a) \in R\}$

X 称为形式概念的外延(Extent)；而 Y 称为形式概念的内涵(Intent)。形式概念之间通常存在着一种层次(偏序)关系，我们通常将这种层次关系定义为概念与子概念的关系。子概念的定义如下：

定义 3 子概念

子概念的关系为构造概念格提供了一种自然的偏序关系。Birkhoff 早在 1940 年就证明这样的概念格是一种完全格[2]，定义如下：

定义 4 概念格(Concept Lattice)

$$L(C)=\{(O,A) \in 2^{O} \times 2^{A} \mid O=X \wedge A=Y\}$$

根据这些基本定义，可以得到形式概念分析的一些特征，可根据这些特征进行实际应用，这些特征包括：①形式概念决定了具有共同形式属性的一组最大的形式对象的集合；②概念格显示了一种形式对象以形式属性的层次分类；③概念格中的某个格与子格关系可以说明一些属性是某抽象属性的不同实例.

3 一个简单的软件维护活动框架

软件维护在软件演化过程中必不可少。基本的软件维护过程应包括如下几个活动：与修改请求相关的软件理解，修改影响分析，修改实施(包括重构以及修改传播分析)，修改后系统的调试与测. 图 1 给出软件维护活动的一个简单框架模型，它包含了基本的软件维护活动。在该框架中，首先是用户提出修改请求；然后进行特征定位，即把自然语言的修改请求规约成维护人员可理解的修改请求，并进一步理解与这次修改请求相关的软件系统以及系统内部各组件间的依赖关系；下一步就是进行修改影响分析，通过影响分析，管理人员可评估出这次修改所带来的影响范围，从而决定是否接受这次更改请求，若接受，则进入修改实施阶段，这个过程包括重构以及修改传播分析；当完成这次修改之后，需重新测试修改后的系统，确保修改的正确性与一致性，以及修改没有给系统其它部分带来副作用。本文总结的一些基于形式概念分析的应用主要从图 1 中的软件维护活动框架出发，讨论近十几年来形式概念分析在软件维护这些应用中的一些相关研究工作。

本文针对图1软件维护原系统活动框架中的四种软件维护活动分别进行介绍：软件理解、修改影响分析、重构、调试与测试。

4.1 形式概念分析应用于软件理解

软件系统在实施修改前，维护人员首先要对维护的软件进行软件理解以决定如何对系统进行修改。软件理解是软件维护与演化过程中的首要活动，也是最难的活动之一，它包括了对软件文档和程序中元素及各种依赖关系的理解。

形式概念分析在软件理解中应用较多地集中在特征定位方面。特征定位用于将自然语言形式的修改请求映射到代码层次的修改，识别哪些代码实现需求中的哪些功能(或者非功能)特性。Xue 等人则通过形式概

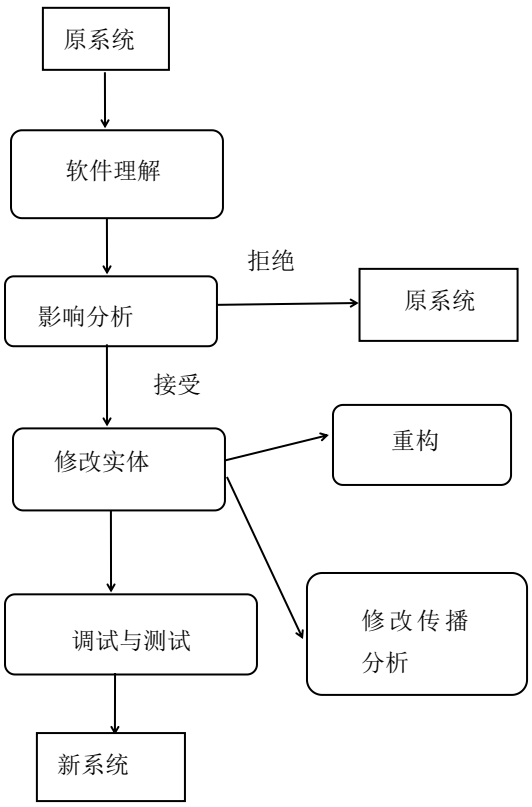

图 1　软件维护活动框架

念分析从软件代码演化过程中不同产品的变体中挖掘出共同的特征，这样有利于后期相似特征的代码重用[5]。

形式概念分析在软件理解中的应用不仅包括对软件各元素依赖关系的理解，还包括软件演化过程中演化性的分析。Xu 等人提出一种基于模糊形式概念分析的程序聚类方法。还有一些工作从代码演化过程中提取开发过程和演化过程中的一些模式去理解源程序。另外，协同修改模式也可通过概念格提取出来，这为后续软件演化以及维护过程中所需的维护活动提供一个有效的度量方法。

4.2 形式概念分析应用于修改影响分析

当对软件进行修改时，肯定会对软件的其他部分造成一些潜在影响，从而导致软件的不一致性。软件修改影响分析是用来识别软件修改对软件其他部分所带来的潜在影响[6]。ToneHa 将概念分析与分解切片结合起来，提出一种分解切片概念格的概念[35]，该方法主要适用于过程内针对某语句或者变量修改的影响分析。

主要是将形式概念分析技术应用于面向对象语言程序的影响分析，其所分析的是类和成员方法之间的关系，得到一种新的面向对象程序的中间表示，类与方法依赖关系格。在该格上进行影响分析比较容易实现，且结果也是一种层次化的结果，可更好地指导维护人员准确定位修改所带来的影响。

4.3 形式概念分析应用于重构

目前相关的形式概念分析应用于软件修改实施主要集中在重构领域。重构可在不改变软件原有功能的前提下，通过调整程序代码内部结构改善软件质量和性能，使其程序的设计模式和架构更趋合理，提高软件的可扩展性和可维护性。形式概念分析很自然地为人们提供一个层次聚集的视图，这就为识别程序中的层次与聚集打下好的基础，目前将形式概念分析应用重构主要包括类层次结构重构，模块结构，“坏味道”设计和代码重构，由低层次语言规范向高层次的语言规范进行重构等。

重构活动最早开始关注的是对“坏味道”设计和代码进行重构。Moha 等人通过形式概念分析识别的方法去识别程序中的“坏味道”设计。而 Arévalo 等人则通过形式概念分析技术将类层次区分为好的层次设计、坏的层次设计以及不正常的层次设计，这样可直接指导维护人员去理解相关的类层次以及修改类层次中存在的“坏味道”。Joshi 则提出一种内聚格的表示方法，该格可用于指导维护人员对哪些类进行提取，转移哪些方法，以及确定需要的属性和删除没有使用到的属性等等[7]。

模块化是软件设计中非常重要的一个概念，还有一些研究利用形式概念分析对程序进行模块重构。TKim 等人则提出一种面向对象的概念分析方法进行模块重构[8]，面向对象的概念分析方法除保持传统的形式概念分析方法的一些优点外，它可通过粗粒度的方式直接识别模块，因而更适用于大规模

软件的模块重构。

4.4 形式概念分析应用于调试与测试

当实施修改后，需对修改后的软件进行测试以确保修改后的软件具有一致性。当某个测试用例未通过时，需进行错误定位，确定程序的哪部分导致这个错误。

软件调试需要执行一系列测试来定位错误，Ammons 等人给出一种将形式概念分析应用于调试形式化时态规约的方法[9]，他们的方法一般只需检查原有轨迹数量的 1／3 就可分类检查出错误与正确的规约。孙小兵等人提出了一种测试覆盖概念格，根据该概念格利用三种策略进行错误定位[10]。

另外，部分研究人员将形式概念分析技术应用于软件测试，主要用于测试用例集的生成以及约减。

5 研究趋势与展望

形式概念分析技术在程序理解和重构方面已取得大量的研究成果，而形式概念分析在修改影响分析、调试以及测试方面的研究工作目前还处于起步阶段，甚至在修改后的系统验证方面还没有找到相关研究文献，因而形式概念分析技术在这些方面的研究还有待深入。

尽管形式概念分析技术已有一些工具支撑，但这些工具主要还是用于概念格的构造，当这些工具应用到某个具体的领域中还有一些不适应，需进一步修改和扩展，而这方面的工具目前还相当缺乏，所以开发适合某个具体领域形式概念分析的工具很有必要。

6. 结语

形式概念分析技术本身经过几十年的发展和完善，已成为具有较强形式化理论支撑的软件分析技术，该技术近年来被广泛应用到软件维护各种活动中，包括软件理解，修改影响分析，修改实施，调试与测试。尽管形式概念分析在软件维护中有着广泛的成功应用，但其在应用领域深化、工具支撑等方面还需进行深入的研究。

参考文献

[1]Ganter B,Wille R.Formal Concept Analysis： Mathematical Foundations[M]．Berlin：Springer-Verlag,1996．

[2]Birkhoff G．Lattice Theory[M]．USA：American Mathemafical Society,1940．

[3]Tilley T,Cole R， Becker P,et a1.A survey of formal concept analysis support for software engineering actifities[A].LNCS3626:FormalConceptAnalysis[C].Berlin:Springer,2005.250—271．

[4]Tonella P.Formal concept analysis in software engineering[A] ． Proceedings of Intemational Conference on Software Engineenng[C].Edinburgh,Scotland:IEEE,2004.743—744．

[5]Xue YX，Xing ZC,Jarzabek S．Feature location in a collection of product variants[A].Proceedings of the Working Conference on Reverse Engineering[C].Kingston,ON,Canada:IEEE,2012．145—154．

[6]Tonella P．Using a concept lattice of decomposition slices for program understanding and impact analysis[J].IEEE Transactions on Software Engineering,2003,29(6)：495—509．

[7]Joshi P,Joshi RK.Concept analysis for class cohesion [A]. Proceedings of the 2009 European Conference On Software Maintcnance and Reengineering[C]．Bombay Powar,Mumbai：IEEE,2009.237—240．

[8]Kim HH,Bae DH.Object-oriented concept analysis :ware modularization[J]．IET Software,2008,2(2)：134—148．

[9]]Ammons G,Mandelin D,Bodik R,et a1.Oebuggiag temporal specifications with concept analysis[A].Proceedings of the Conference On Programming Language Design and Implementation[C].San Diego,CA：ACM,2003.182—195．

[10]Sun XB,Li BX,Wen WZ．CLPS-MFL：Using concept lattice of program spectrum for effective multi-fault localization[A]. Proceedings of the 13th Intemational Conference on Quality Software[C].USA:IEEE,2013.204—207．