程序理解中基于类图的特征挖掘

胡圣明，李青山，褚 华，陈 平
(西安电子科技大学 软件工程研究所，陕西 西安 710071)

摘要：针对从大型系统源代码逆向恢复出的类图十分复杂，不利于系统理解和特征挖掘的问题，从类图中抽象出类型依赖图(TDG)，并分为无权值及带权值类型依赖图，利用图上的集合划分算法对 TDG 进行分层抽象的特征挖掘，挖掘算法将图中的节点划分到不同的集合中，每个集合展现系统关键设计的一个侧面. 采用 TDG 上的分层算法能够有效地降低类图的复杂度并挖掘出系统设计特征.

关键词：程序理解；类图；特征挖掘；类型依赖图
中图分类号：TP311 **文献标识码**：A **文章编号**：1001-2400(2006)04-0550-07

Aspect mining from the class diagram in program comprehension

HU Sheng-ming, *LI Qing-shan*, *CHU Hua*, *CHEN Ping*
(Research Inst. of Software Engineering, Xidian Univ., Xi′an 710071, China)

Abstract: The class diagram which is recovered from the large system's source code is too complex to comprehend. This paper presents two kinds of TDG (Type Dependency Graph) abstracted from the class diagram, TDG without weight and TDG with weight. Then a mining algorithm is applied to the TDG to achieve the hierarchy abstraction and the system's design aspects. The mining algorithms allote the TDG vertices into different subsets with each subset presenting one design aspect of the system. The complexity of the class diagram is reduced remarkably with design aspects obtained.

Key Words: program comprehension; class diagram; aspects mining; type dependency graph

理解面向对象系统的关键就是理解其类图的结构[1]，并从类图中挖掘出系统的关键设计特征[2]. 在大型复杂的面向对象的系统中，类的设计、类与类之间的关系十分复杂，理解一个类的设计目的及其对象的运行状态通常会涉及其他相关类[3]，这导致系统理解非常困难，很难找到一个入口点进行分析理解工作，抽象出高层系统设计特征需要付出更多的努力.

许多正向工程工具能从已有系统中抽取类图，例如，Rational Rose, Paradigm Plus, OEW, Domain Objects 和 COOL:Jex 等. 但是，要支持完全的循环迭代再工程，仅仅抽取类图是不够的，逆向产生的静态模型应该具有多个抽象层次并体现领域特征，从而为程序理解提供更全面的支持.

现有的大多逆向工程工具和环境都支持动态模型的提取、呈现甚至是进一步的抽象，有些方法和工具还利用目标系统不同层次和侧面的静态模型来过滤、切片同一系统的动态模型，提高动态模型的抽象层次和可理解性. 如 Ovation 工具使用执行模式视图在不同抽象层次上管理和可视化程序的执行，Scene 工具产生事件踪迹并把它们可视化为剧情图，SCED 在逆向工程中被用于动态剧情的呈现和抽象，其剧情图扩展了 OMT 剧情图，在语义上与 UML 序列图相似，Shimba 是一个逆向工程环境，它用 Rigi 呈现和分析软件的静态结构，用 SCED 呈现和抽象动态剧情，ISVis 是一个支持浏览和分析执行剧情的可视化工具，它属于致力于促进遗产软件系统进化的 MORALE 工具集. 但这些逆向工程工具逆向抽取和抽象结果的呈现方式大多

收稿日期：2005-08-07
基金项目：国家教育部博士点基金(20030701009)；研究生创新基金(05009)；国家自然科学基金项目(60473063)
作者简介：胡圣明(1979-)，男，西安电子科技大学博士研究生.

是自己定义的,不标准,很难用于循环迭代式的再工程活动.

笔者在类图的基础上,提出一种基于抽象类图,即类型依赖图的挖掘算法,对类型依赖图进行分层抽象,将挖掘的结果可视化地呈现给系统分析理解人员,通过算法的分析结果找到系统分析与理解的入口点以及相关类之间的设计特征,结果采用标准 UML 统一建模语言呈现在 Rose 工具中.

1　类图的逆向抽取

存在多种手段获取系统的完整类图,最常见的是通过文档(设计模型)获取类图的设计,但文档和真实的系统之间存在着不一致性,导致这种不一致性的原因是多方面的,最根本的原因是从设计到代码的映射是通过手工编程实现,这就会产生设计到代码的偏移. 常用的 CASE 工具能够做到代码框架的生成,但还不能实现从设计到代码的完全自动生成,而且,系统的设计需要经过反复的修改,是一种迭代反复的过程,在这个迭代的开发过程中,很容易造成设计与代码间的偏移距离增大,使得设计文档的可信度降低.

描述系统设计最准确的"文档"是源代码,它与实际运行系统完全一致. 因此,从源代码获取到的信息的可信度最高. 目前已有很多工具支持从源代码中直接生成类图,但其逆向恢复功能存在较大的局限性. 为从任何 C++实现的面向对象系统源代码中获取准确的信息,这里采用一个自主研发的、嵌入 Rational Rose 的逆向工程工具 XDRE,其体系结构如图 1 所示.

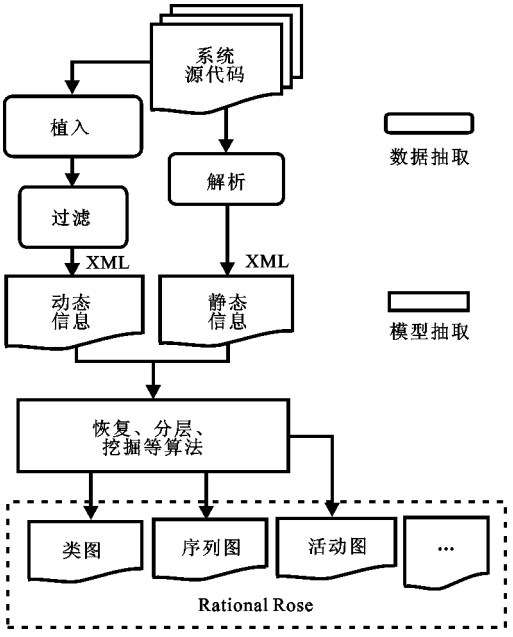

图 1　XDRE 的体系结构图

XDRE 工具首先对源代码进行解析,将解析获得的系统静态信息存储在 XML 格式的文件中,采用 XML 文件能够实现 XML 模式到面向对象环境中类层次的影射模型,方便面向对象系统对数据的存取[4]. 解析过程中,同时对源代码植入软件触发器,这里触发器的工作是为了获取系统运行时的动态信息,并保证植入软件触发器后的系统在功能上与植入前等价. 当系统运行时,触发器收集动态信息并将其存储到另一 XML 文件中,至此,系统的动态信息和静态信息都获取到,再将各种恢复、分层和挖掘算法应用到已收集的数据上,从而最终获得系统的高层体系结构图[3]. XDRE 工具借助 OpenC++实现静态分析、软件触发器植入以及运行时动态信息收集[5~7]. OpenC++首先对源代码进行分析,将源代码转换成代码片段并将代码片段的分析树返回给开发者,开发者可在分析树上添加自己的计算节点从而实现软件触发器的植入,OpenC++再将修改后的代码片段进行组合而得到最终运行的系统[8,9]. 图 2 是 XDRE 恢复的一个类图实例,结果在 Rational Rose 建模工具中呈现.

图 2 为一采用 C++实现的呼叫中心系统的类图,体系结构为客户/服务器体系结构,系统仍在实际的运行中. 系统包含了 129 个 C+源代码文件,93 个类,共计 47 296行源代码. 系统实现与 ANSI C++标准兼容,可运行在 UNIX/Solaris/Linux 和 Windows(95,98,2000,NT,XP)平台上. 恢复所得到的类图十分复杂,系统的分析理解人员是很难入手去分析如此复杂的一个系统,获得系统的高层设计特征更加困难. 因此,对其进行分层抽象的挖掘是十分必要的.

2　类图的特征挖掘

针对图 2 所示的复杂类图,必须按照一定的规则对其进行重新划分和组织,以达到特征挖掘的目的.

图 2 XDRE 恢复的一个类图实例

2.1 无权值类型依赖图(TDG)

面向对象系统典型的特征是允许用户自定义类型——类，系统部分特征就体现在类的定义以及类间关系上，图 3 描述了用户自定义类型及其所依赖类型间的关系：用户自定义类型依赖于程序设计语言和第三方库提供的类型.

由图 3 可看出，类型间具有相互的依赖关系，如果一种类型依赖于其他类型，那么理解其他类型就成为理解此类型的基础，并且，根据类型间的不同关系，进行抽象与分层后，可得到系统类图的关系设计特征.

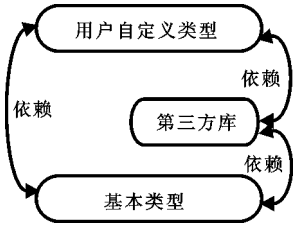

图 3 类型间的依赖关系

将所有用户自定义的类型抽象为节点，而类间关系抽象为有向边，则类图转化为一种有向图，这里称之为类型依赖图，若不考虑各种类型间的具体关系，统一采用依赖关系来表示，那么称此有向图为无权值的类型依赖图. 对无权值的类型依赖图进行集合划分，便可找到理解系统应遵循的路径. 采用形式化的方法描述，TDG 可表示为 $<V,E>$ 二元组，其中 V 表示顶点的集合，而 E 表示有向边 $e=(v_1,v_2)$ 的集合. 在无权值 TDG 中，一个顶点表示一种类型，而一条有向边表示边的起点类型依赖于边的终点类类型.

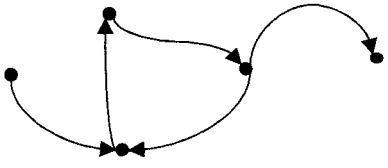

图 4 无权值 TDG

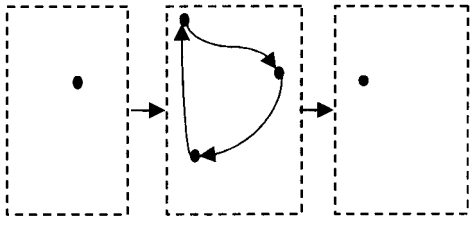

图 5 无权值 TDG 的划分

图 4 为无权值 TDG 的一个实例. 对图 4 所示的 TDG 进行集合划分，可得到图 5 所示结果，图 5 中间部分就是分析人员需要关注的类图设计特征，挖掘的结果甚至可帮助验证类图设计的合理性.

2.2 无权值 TDG 挖掘

无权值 TDG 挖掘的主要目的是根据类间的依赖关系，将类划分到具有层次性的不同集合中，每个集合中的类间关系表现了系统的关键设计特征. 具体的挖掘算法如算法 1 所示.

算法 1 无权值 TDG 挖掘算法

Input: TDG$<V,E>$

Output: $AV[1\cdots N]$

Begin

lowlevel=0; highlevel=N;

```
LOOP (V 不为空)
  FOR (V 中每个节点 Vi)
    IF (Vi 出度为零)
    ELSE-IF (Vi 入度为零)
      将 Vi 加入集合 AG[highlevel]中
    END-IF
  END-FOR
  IF (AG[lowlevel]不为空)
    从 TDG 上删除 AG[lowlevel]中节点
    lowlevel++
  IF (AG[highlevel]不为空)
    从 TDG 上删除 AG[highlevel]中节点
    highlevel--
  IF (AG[highlevel]==AG[lowlevel]为空)
    合并 TDG 中一个循环链路中所有节点为一个节点
END-LOOP
End
```

采用无权值 TDG 的挖掘算法可将复杂的类图划分成粒度非常小的集合,每个集合中类与类间耦合度非常小.此算法适于寻找系统分析的切入点,凡是在较低层次集合中出现的节点所表示的类都应当首先被分析理解,因为它们是其他节点(类型)的基础.由于无权值 TDG 并不区分类间具体关系,而是采用统一的依赖关系描述,因此会导致将一棵类层次树上的类划分到不同的集合中,而对于一棵类层次树(如继承关系产生的层次树),其上所有的类在同一个集合中时更易于理解.

2.3　带权值 TDG 及其挖掘

无权值 TDG 挖掘算法未考虑到类间耦合度,例如具有继承关系的类之间的耦合度是非常高的,它们应被划分在同一个集合中,而关联关系的耦合度相对较低,相应的类可被划分到不同集合中,因此,在 TDG 上可加上权值以表示类间耦合度,权值较大的边表示边所连接的类具有较紧密的联系.扩展 TDG 为 $<V,E,N>$ 三元组,其中 N 表示 TDG 中具有的不同权值个数,权值高的边表示其两端的类应该被最后划分或者不划分.

图 6 为带权值 TDG 实例,其中 $N=3$,代表结点间具有 3 种耦合度不同的关系,需要至少进行 N 次分层抽象.相应的带权值 TDG 分层挖掘算法如算法 2 所示.

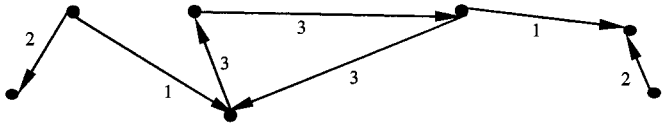

图 6　带权 TDG

算法 2　带权值 TDG 挖掘算法

输入:TDG$<V,E,N>$

输出:特征集合

开始　令 $K=N$

第一步:将一权值为 K 的边及其连接的两个节点划分到一个新集合中;

第二步:若集合中任意一个节点和集合外的节点存在一条有向边,且权值为 K,则将集合外节点加入到此集合中;

第三步:重复第二步直到集合内与集合外不存在权值为 K 的有向边;

第四步:若 TDG 中存在权值为 K 的边,转到第一步;

第五步:令 $K=N-1$,若 $K\geqslant 1$ 转到第一步,否则结束.

结束

将带权值 TDG 挖掘算法应用到如图 6 所示的 TDG 上,可得到图 7 所示的结果.图 6 表明在类图中共有 7

个类节点,类间的关系分为 3 种,分别具有不同的耦合度. 耦合度较高的节点(类)间关系会被首先划分到一个集合中,图 7 表明权值为 3 的有向边的所连接的节点会被首先合并,其次为权值为 2 和 1 的有向边.

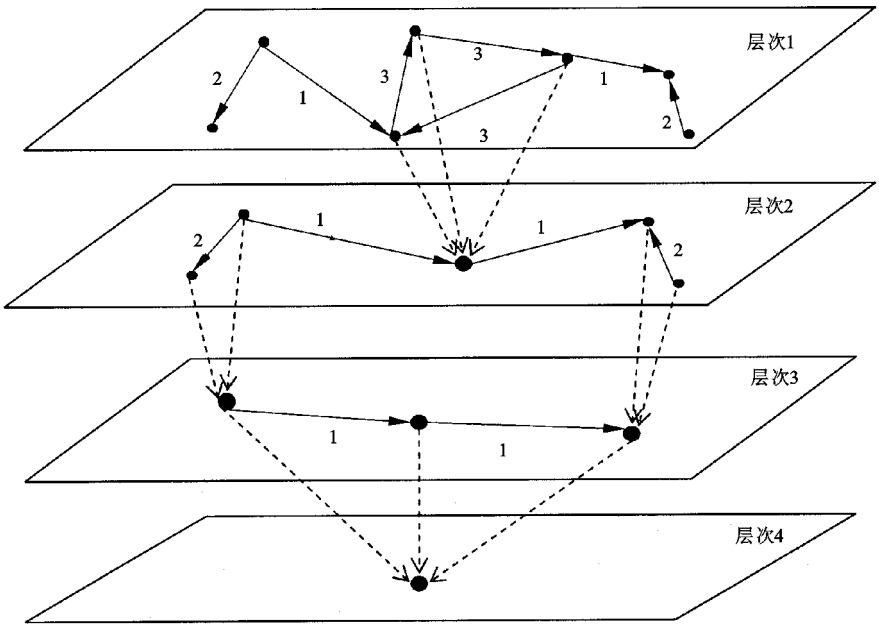

图 7 多次分层挖掘

3 实验研究

为理解图 2 所示类图,将类图分别转化为无权值和带权值 TDG,并应用挖掘算法,结果呈现在 Rational Rose 中. 图 8 为无权值 TDG 的分层挖掘结果. 每一个包表示系统中类型的集合,且高层包中的类依赖于底层包中的类,例如,Aspect01 中的类依赖于 Aspect00 中的类.

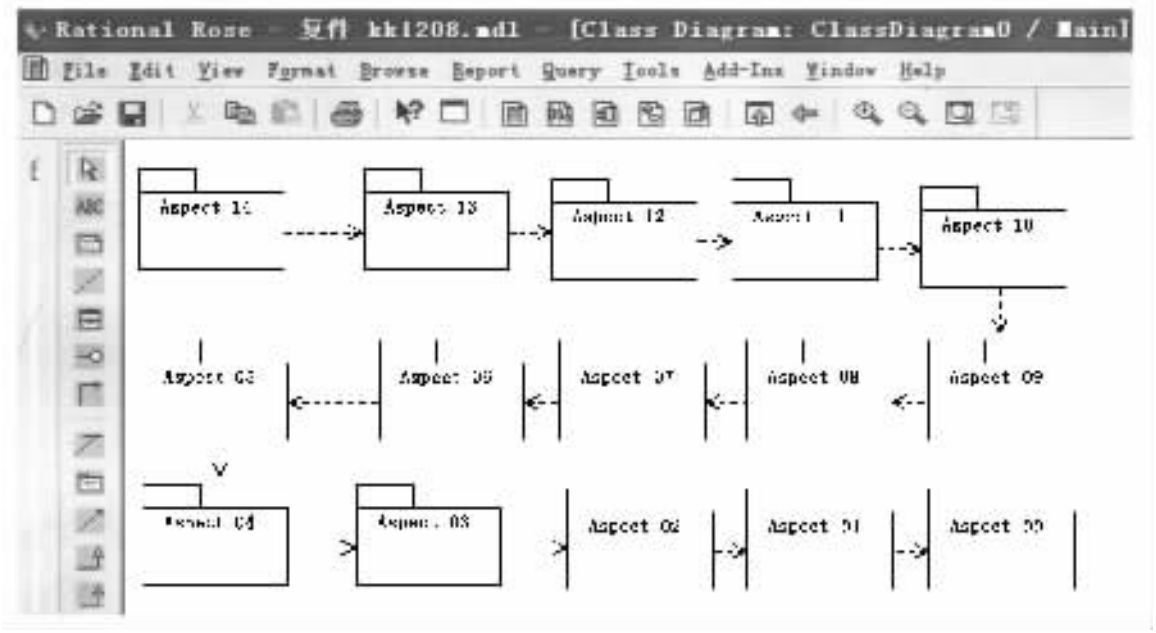

图 8 无权值 TDG 挖掘结果

在从类图转化为带权值 TDG 的过程中,将类间泛化关系的权值设为最大,然后依次是依赖关系和关联

关系,则分层挖掘的第 1 层和第 2 层结果分别如图 9 和图 10 所示. 第 1 层已将类间的泛化设计特征挖掘出来,而第 2 层则表现了类间的依赖关系特征. 由结果可看出,类图的复杂性已显著降低,并且各种关系的设计特征被直观地呈现出来.

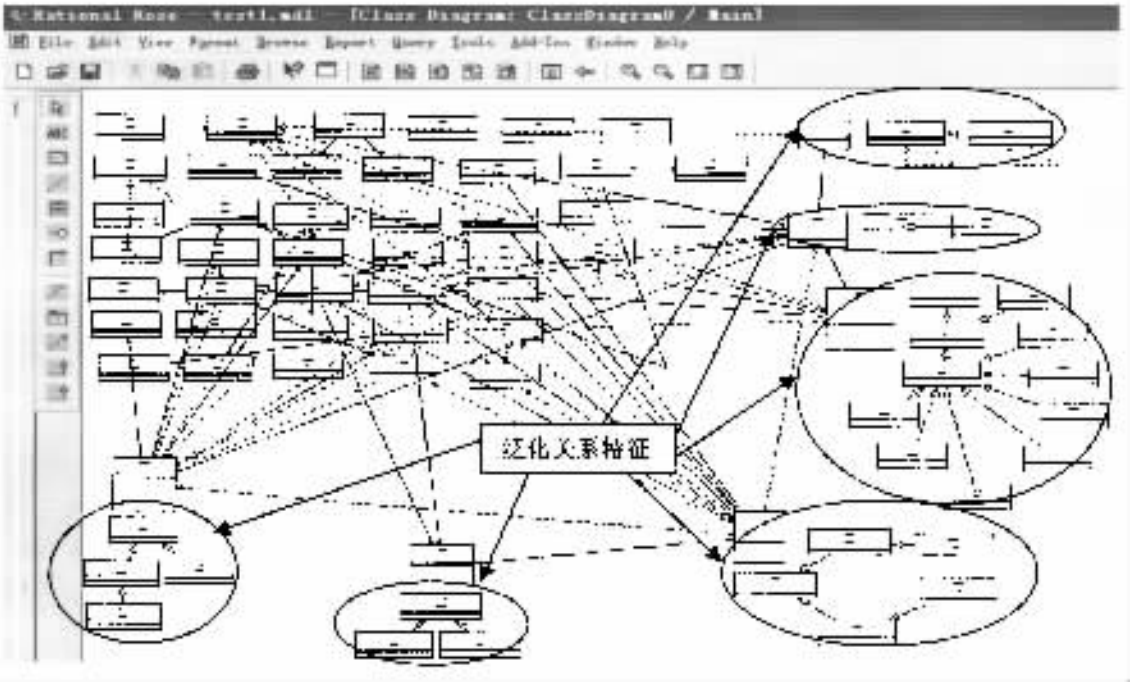

图 9　带权值 TDG 第 1 层挖掘结果

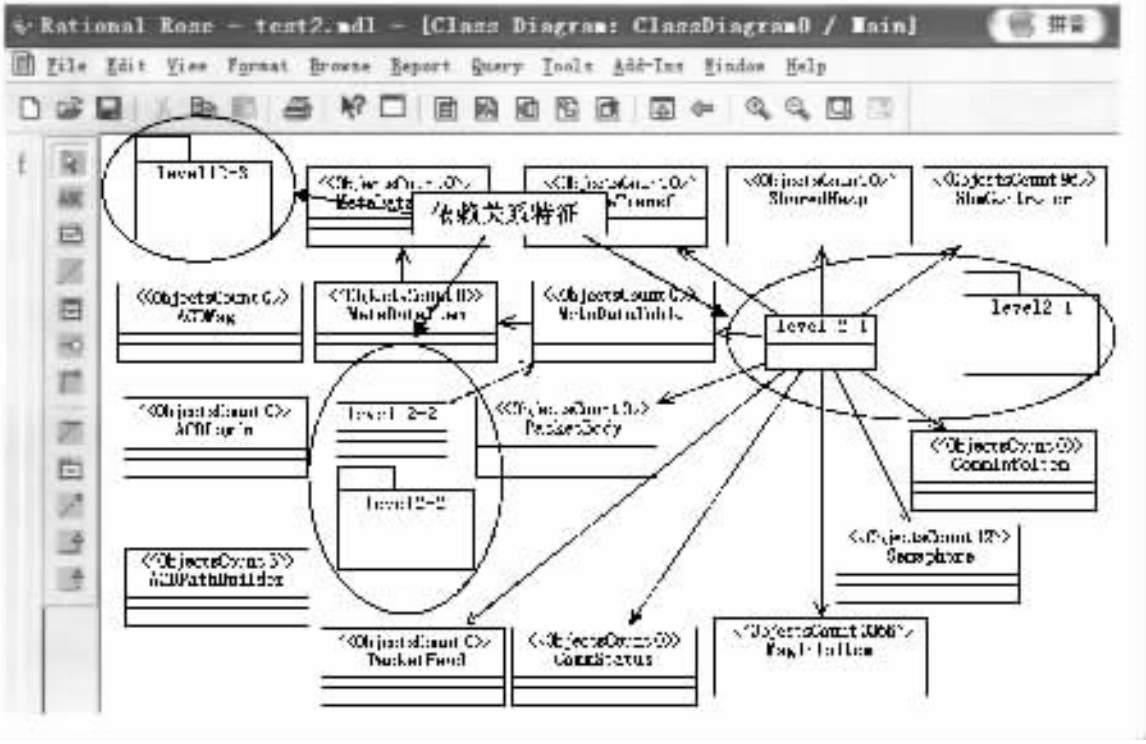

图 10　带权值 TDG 第 2 层挖掘结果

4 结 束 语

将类图转化为 TDG 并应用挖掘算法能够有效降低系统类图的复杂度并挖掘出系统的高层设计特征,有助于系统的理解、维护和进化. 无权值 TDG 及其算法给出系统理解的先后路径;带权 TDG 能够准确地表示类间耦合度,使挖掘算法发现的设计特征更合理. 但现有的算法仅仅考虑到任意两个类间仅具有一种关系,当两个或多个类之间具有两种或两种以上关系时,挖掘结果的粒度较粗,即集合中类及类间的关系仍然较为复杂. 因此,下一步的工作重点是改进带权值 TDG 挖掘算法,使其达到更细的挖掘粒度.

参考文献:

[1] Mancoridis S, Mitchell B S, Rorres C, et al. Using Automatic Clustering to Produce High-Level System Organizations of Source Code[A]. Procedings of the 1998 International Workshop on Program Understanding[C]. Ischia: IEEE Computer Society, 1998. 45-52.

[2] Chikofsky E J, Cross J H. Reverse Engineering and Design Recovery: a Taxonomy[J]. IEEE Software, 1999, 7(1): 13-17.

[3] 李青山. 面向对象软件的动态模型设计恢复与体系结构抽象[D]. 西安:西安电子科技大学,2003.

[4] Li Qingshan, Chen Ping. Study of the XML Data Binding Model at Object Level[J]. Journal of Xidian University, 2001, 28(6): 768-771.

[5] 李青山,陈 平,王 伟,等. 逆向工程中反射植入的研究[J]. 计算机学报,2004, 27(4): 535-542.

[6] 李青山,陈 平,王 伟. 一种基于反射和开发编译的 C++植入机制[J]. 系统工程与电子技术,2003 25(7): 851-855.

[7] 褚 华,李青山,陈 平,等. 一种基于 UML 序列图的状态图合成方法[J]. 系统工程与电子技术,2005,27(3): 524-528.

[8] Chiba S. OpenC++ 2.5 Reference Manual[DB/OL]. http://www.csg.is.titech.ac.jp/~chiba/opencxx/reference.pdf, 2003-05-13.

[9] Chiba S. A Study of Compile-time Metaobject Protocol[DB/OL]. http://citeseer.ist.psu.edu/chiba96study.html, 2005-06-08.

(编辑: 齐淑娟)

我校成功举办空间、航空与导航电子学国际会议

2006 年 4 月 10~12 日,由我校承办的 2006 年空间、航空与导航电子学国际会议在我校逸夫图书馆举行. 会议由日本电子情报通信学会空间、航空与导航电子学分会主办,中国空间技术研究院、韩国航天研究院、日本宇宙航空研究开发机构等单位协办. 来自海内外的 60 多名专家代表出席了本次会议.

会议分两个会场,共计 60 多场次学术报告. 与会人员就航空、导航和通信等领域的最新研究成果及发展前景进行了广泛的交流和深入的研讨. 会后,来自日本的 30 多位代表参观了学校和西安高新技术开发区. 他们对我校电子通信领域的研究水平和高新区的快速发展留下深刻的印象.

空间、航空与导航电子学国际会议由日本发起,目前已成为一个系列性国际学术会议,曾在日本、韩国等不同国家轮流召开. 此次会议的成功举办,不仅促进了国际空间科学技术的合作与发展,而且对扩大学校影响,提高我校新开设的空间信息科学专业的学术水平具有积极的意义.

(转自《西电科大报》2006 年第 4 期)