

# C++ 程序理解辅助系统\*

吴穹 金成柏 金淳兆

(吉林大学计算机科学系 长春 130023)

**摘要** 本文介绍了一个针对 C++ 语言的程序理解辅助系统 HOOPUS (hypertext-based object-oriented program understanding support system). 该系统基于程序结构分析技术, 以易于理解的形式显示程序中的信息, 辅助理解 C++ 程序的功能与结构. 另外, 它还利用 hypertext、多维表现、导航与定位等技术对理解过程提供支持.

**关键词** 程序理解, 面向对象, 超级文本, 软件维护.

所谓“程序理解”就是通过一定的手段来弄清一个程序是“做什么”和“怎么做”的, 它是软件维护、软件重用和逆向工程的基础和关键步骤. 程序理解是一种需要高度智力活动的过程, 如何对这一过程提供自动化支持, 已经成为软件工程领域的研究热点之一. 程序理解问题的研究途径大体可分为 2 种: 基于知识的程序功能识别技术和基于结构分析的程序信息提供技术. 前者利用人工智能技术, 把人类专家理解程序的活动加以自动化, 来弄清程序的功能. 理解程序的主体应是为为此而开发的程序理解系统. 这种技术目前仍处于理论探索阶段. 后者通过分析程序结构, 抽取并提供理解程序所需的各种信息, 以此来辅助人的理解活动. 理解程序的主体是人. 本文讨论的是基于结构分析的程序理解辅助技术.

目前, 以 C++, Smalltalk 为代表的面向对象程序设计语言(OOPL)被广泛使用. 面向对象程序的理解工作已经被提到日程上来了. 近几年来, 国内外已开发了许多针对过程式语言的程序理解系统. 但由于对象式风范不同于过程式风范, 面向对象程序所涉及的实体及实体之间的关系更为复杂, 所以仍需发展针对面向对象语言的程序理解技术.

本文讨论了一个针对 C++ 语言的程序理解辅助系统 HOOPUS (hypertext-based object-oriented program understanding support system) 的设计与实现.

## 1 HOOPUS 系统的设计原则

为了对程序理解过程提供有效的支持, 需要研究人在手工状态下进行程序理解的过程, 并找出这一过程中的难点所在.

人理解程序的过程, 就是根据程序中的各种信息, 在头脑中建立表示该程序的功能与结

\* 本文研究得到国家 863 高科技项目资助. 作者吴穹, 1973 年生, 硕士生, 主要研究领域为软件工程, 面向对象技术. 金成柏, 1962 年生, 讲师, 主要研究领域为软件工程, 面向对象技术. 金淳兆, 1937 年生, 教授, 主要研究领域为软件工程.

本文通讯联系人: 吴穹, 长春 130023, 吉林大学计算机科学系

本文 1995-10-16 收到修改稿

构模型的过程. 这个过程包括 3 个反复进行的活动<sup>[1]</sup>: 1) 信息获取: 获知程序中各种实体(模块、数据项等)之间的关系; 2) 信息分析: 根据实体之间的关系, 建立或进一步修改已建立的模型; 3) 信息输出: 记录或交流理解的结果.

在信息获取活动中, 人往往采用分而治之的策略. 也就是说, 人并不是直接对整个程序进行信息获取活动, 而是依据某种原则将程序划分成若干相互关联的程序片段, 逐个获取这些程序片段中的信息. 例如, 人在获取过程式程序中的信息时, 可以以模块为单位进行划分, 对于面向对象程序, 则可以以类为单位. 人在同一时刻只能观察一个程序片段, 我们称该片段为视窗. 在信息获取活动中, 有些信息的获取依赖于另一个程序片段的理解结果. 如果该程序片段仍未被理解, 则有可能需要从这个程序片段转移到与它相关的另一个程序片段, 这种转移的过程称为视窗切换. 如果该程序片段在另一个理解人员的理解范围之内, 则需要进行成员之间的交互. 由此可见, 信息获取过程是一个夹杂着视窗切换和成员交互的嵌套过程.

关于信息分析活动的过程, 目前还没有统一的认识, 目前描述该过程的理论主要有 3 种<sup>[1]</sup>: 1) 自底而上的理解; 2) 自顶向下的理解; 3) 随机的理解. 在信息输出活动中, 人将已经理解的程序片段以抽象的形式描述, 并允许其他人共享这些抽象描述.

程序理解辅助系统主要对信息获取和信息输出 2 种活动提供自动化支持, 而信息分析活动则由人来独立完成. 在信息获取和信息输出活动中, 人遇到的主要困难体现在以下几个方面:

1. 在视窗切换的过程中, 查找和定位花费了大量的时间.
2. 在经过一系列的视窗切换之后, 有时无法回到进行切换前的位置或者忘记了当前局部理解工作的目的. 因为在手工状态下, 保存和记忆视窗的切换过程比较繁杂、效率极低.
3. 难于了解系统的整体结构以及程序单位在系统范围内的使用情况.
4. 成员之间的交互和局部理解成果共享比较困难或者需要花费大量的时间.

C++程序与过程程序相比更难于理解, 这主要体现在以下 2 个方面: ① C++程序中实体之间的关系更为复杂, 主要增加了类与类之间的继承关系、友元关系, 类与成员函数之间的从属关系, 成员函数之间的重载和多态关系; ② 动态约束增加了程序执行过程的不确定性.

以下我们将利用程序理解辅助系统进行程序理解工作的人称为用户, 将用户需要理解的程序称为目标程序. 为了帮助用户克服上述困难, 程序理解系统至少应在以下 3 方面提供支持: 1) 提供有关目标程序的尽可能全面的、多层次的信息; 2) 对用户的理解过程提供支持, 主要包括加速查找和定位过程以及保存视窗切换过程 2 个方面; 3) 用户之间的交互及局部理解成果的共享.

为了给用户有效的支持, 一个程序理解辅助系统至少需要包括 4 个部分: 1) 信息抽取器: 负责从目标程序中抽取信息, 信息应尽可能全面, 以适应理解工作不同的背景和目的; 2) 程序信息库: 负责储存从目标程序中抽取的信息; 3) 信息库服务员: 接收对程序信息库的访问请求, 由它操作程序信息库, 将有关信息加以适当组织后提供给申请者, 它还负责支持程序信息库的共享; 4) 用户界面: 将信息以易于理解的形式显示给用户, 对用户的理解过程提供支持, 并支持用户之间的交互及局部理解成果的共享. 信息库服务员的引入符合数据抽

象的原则,它使得系统的其他部分不必关心程序信息库的具体结构和操作方法,便于维护程序信息库的完整性和一致性,也便于支持对程序信息库的共享。

### 2 系统概述

HOOPUS 系统是基于 hypertext 技术的、针对 C++ 语言的程序理解辅助系统,其系统结构如下所示:

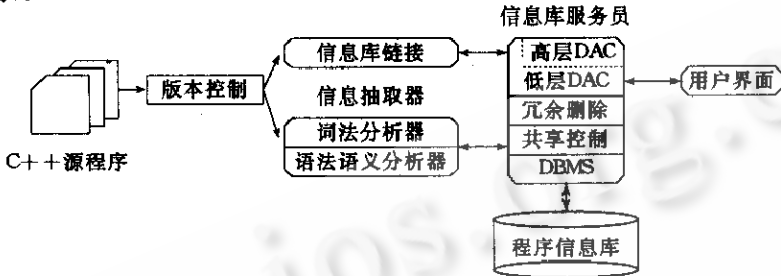


图1 HOOPUS系统结构图

HOOPUS 系统的工作过程分为 2 个阶段:(1)程序信息库建立;(2)目标程序理解. 程序信息库建立阶段使用了增量分析技术. 它的工作过程是:用户将一个理解任务提交给版本控制部分,版本控制部分判定哪些源文件被更新了,并启动信息抽取器只对那些被更新的源文件执行信息抽取动作,创建中间信息库,最后启动信息库链接部分将分立的中间信息库链接成程序信息库. 目标程序理解阶段的工作过程是:用户界面部分根据用户的请求从程序信息库中获得信息,以易于理解的形式显示给用户,它利用多维表现、导航与定位、point-and-click, hypertext 等技术,对用户的理解过程提供全面的支持. 在这个阶段,系统支持对程序信息库的共享以及局部理解成果的交互.

信息库服务员是访问程序信息库的唯一接口. 它控制所有对程序信息库的操作,并为程序信息库的共享提供支持. 信息库服务员由高层 DAC 例程、低层 DAC 例程、冗余删除和共享控制 4 部分组成. DAC(data access command)命令是专门为存取程序信息库中的信息而设计的,它分成高层 DAC 命令和低层 DAC 命令 2 部分. 低层 DAC 命令提供了一组对程序信息库的完备的操作,而高层 DAC 命令则是一组常用的低层 DAC 命令序列的封装. 高层 DAC 命令的引入有利于减少用于信息传递的开销,提高整个系统的效率. 冗余删除部分负责标识目标程序中的冗余信息,以进一步减轻用户的理解负担.

### 3 系统功能

HOOPUS 系统对用户理解程序提供以下几个方面的支持:1)提供有关目标程序的尽可能全面的、多层次的信息;2)加速查找和定位的过程;3)提供多维表现能力;4)保存视窗切换过程;5)支持用户间的交互及局部理解成果的共享.

C++ 程序是由实体及实体之间的关系组成的,而实体就是具有类型的标识符. 所以,信息获取活动实际上就是从目标程序中发现实体之间关系的过程. HOOPUS 系统能够向用户提供有关实体间关系的信息,用户可以直接利用这些信息进行信息分析活动,这实际上简化了信息获取活动,从而简化了程序理解的过程. 另外,本系统还提供诸如类继承关系图、

调用关系图等高层结构图来帮助用户了解目标程序的整体结构. HOOPUS 系统可以向用户提供的实体间的关系有以下几种:

- 类与类之间的关系:
  - 类 C1 继承类 C2
  - 类 C1 引用类 C2
  - 类 C1 是类 C2 的友元类
- 类与数据项之间的关系:
  - 数据项 D 是类 C 派生的实例
  - 数据项 D 是类 C 的成员变量
- 类与函数之间的关系:
  - 函数 F 是类 C 的成员
  - 函数 F 是类 C 的友元
- 函数与函数之间的关系:
  - 函数 F1 调用或激活函数 F2
  - 函数 F1 重载函数 F2
  - 函数 F1 是函数 F2 的多态函数
- 函数与数据项之间的关系:
  - 数据项 D 是函数 F 的局部变量
  - 数据项 D 被函数 F 引用
- 数据项与语句之间的关系:
  - 语句 S 引用数据项 D

HOOPUS 系统利用 hypertext 技术来支持用户的查找和定位过程. 超级文本(Hypertext)是由一组节点以及节点间的链构成的. 该系统利用 hypertext 技术将目标程序及相关信息转化成超级文本, 将程序中的实体转化成节点, 实体间的联系转化成链, 允许用户通过链在目标程序中漫游. hypertext 技术在本系统中主要体现在 2 个方面: 1) 用户界面中每个窗口的内容都是语法敏感的(Syntax Sensitive), 即当光标移动到某个实体的名字上时, 系统自动识别该实体的类型, 并根据类型的不同提供不同的菜单, 允许用户观察与该实体相关联的实体; 2) 系统提供导航图(Pilot Graph)机制来帮助用户迅速地了解他目前理解的部分在目标程序中的位置, 以及这部分和其他部分之间的关系. 前者可以帮助用户迅速从一个实体切换到与其相关联的实体上, 而对于间接相关的实体则可以利用后者来完成这项任务.

HOOPUS 系统利用多窗口和 point-and-click 技术来实现多维表现能力. 多维表现能力是指一个系统允许用户可以同时从多个视角观察目标程序, 并能够灵活地在视角之间进行切换. 多维表现能力允许用户对相似的实体进行比较或从多个侧面观察一个部分, 便于快速、准确地理解目标程序.

HOOPUS 系统利用历史树以树形结构来记录视窗的派生过程, 它的引入将原来线性的理解过程变为树形的, 这使用户在理解过程中始终保持明确的目的, 并减轻了用户在不同视窗间切换所带来的负担.

为了共享用户之间的局部理解成果, HOOPUS 系统为以下几种实体提供附注(Annotation): 类、数据项、成员函数和标号, 允许用户将局部理解的成果写入附注, 这将有利于提高一组用户间信息交互和成果共享的效率.

图 2 给出了 HOOPUS 系统的典型界面, 其中包括 2 个窗口: 类关系图和调用关系图, 它们即可以作为用户了解程序结构的手段, 也可以作为导航图使用. 当鼠标点到 Animal 字样时, 系统自动识别出它是一个类并提供类菜单, 在该菜单中选择实例对象表, 将显示如图 3 所示的窗口.

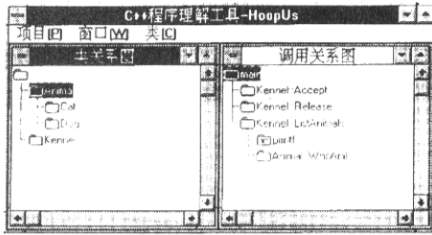


图2 类关系图和调用关系图

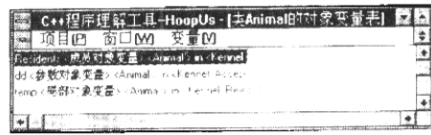


图3 类的实例对象表

### 4 相关工作

程序理解系统的研究工作是从70年代后期开始的. 在10几年的发展过程中, 研究工作主要针对过程式语言的程序理解问题. 目前国内外有关面向对象程序理解问题的研究工作才刚刚起步. 下面我们介绍一些具有代表性的程序理解系统, 并总结和比较它们的优缺点.

PUNS<sup>[2]</sup>是IBM公司开发的用于支持IBM/370汇编语言的程序理解系统. 该系统由3部分组成: (1)信息库及信息库服务员; (2)源程序分析器; (3)用户界面. 该系统是分布式的, 系统的前2部分以IBM/370 30XX为宿主机, 而用户界面部分是在工作站(PC或PS/2)上实现的. 该系统支持对程序信息库的共享.

类似的系统还有IBM公司开发的Integrated Hypertext and Program Understanding Tools<sup>[3]</sup>, 该系统是IBM公司开发的针对多种环境(MVS和VM等操作系统)、多种语言(包括IBM内部汇编语言PL/AS及C语言等)的大型程序理解系统. 该系统中引入了一种称为分阶段分析的技术, 该技术允许递增地从源代码中抽取信息, 并允许用户中断不感兴趣的信息抽取工作, 以此来适应程序理解工作的不同背景.

CIA<sup>[4]</sup>是加利福尼亚大学伯克利分校开发的针对C语言的程序理解系统. CIA的用户界面可以分成低级和高级2个层次: 低级部分利用命令行方式, 输入对有关程序信息的请求, 并得到简单的回答; 高级部分只能给出函数调用关系图等有关程序的总体结构信息. 类似的系统还有支持C++语言的CIA++<sup>[5]</sup>, 支持FORTRAN语言的FAST<sup>[6]</sup>, 以及OMEGA<sup>[7]</sup>, ENCORE<sup>[8]</sup>, Harvard Programming Development System<sup>[9]</sup>等程序理解系统.

XREF/XREFDB<sup>[10]</sup>系统是由Brown大学开发的针对C++, C和Pascal 3种语言的程序理解系统. 该系统的用户界面是集成在GNU Emacs文本编辑器之上的, 通过一个附加的窗口提供有关程序的信息. 类似的利用集成编辑器作为用户界面的系统还有支持Trellis/Owl语言的Trellis<sup>[11]</sup>, 支持InterLisp语言的MasterScope<sup>[12]</sup>, 支持Fortran语言的R<sup>n</sup> environment<sup>[13]</sup>, 支持C语言的CScope.<sup>[14]</sup>

TMC++<sup>[15]</sup>是澳大利亚Linz大学开发的针对C++语言的程序理解系统. 该系统与前几个系统不同, 它没有利用数据库来存储从源程序抽取的信息, 而是将源程序分割成若干小的碎片, CScope<sup>[14]</sup>系统中也使用了同样的处理方法. 这种方法在大型程序的理解工作中, 将导致系统性能的极度恶化. 该系统的用户界面部分具有同步表现能力, 它由2个列表框和1个编辑框组成: 第1个列表框可以选择目标程序中的类, 第2个列表框可以选择该类的方法、子类或超类, 这时编辑框中会显示出以上所选择部分的源代码.

下表给出了 HOOPUS 系统与以上各个系统在功能方面的比较.

	PUNS	CIA	XREF	TMC++	IBM*	HOOPUS
利用数据库	是	是	是		是	是
信息库服务员	单向	单向	双向		双向	双向
支持信息共享	是				是	是
增量分析		支持				支持
分阶段分析					支持	
hypertext 技术	支持		支持	支持	支持	支持
同步表现				支持		
多维表现	支持				支持	支持
导航和定位	支持				支持	支持
支持的 OOPL			C++	C++		C++

\* IBM 代表 IBM 公司的 Integrated Hypertext and Program Understanding Tools<sup>[5]</sup>

## 5 结束语

HOOPUS 系统是在 PC 机上,在 Microsoft Windows 环境下用 C++ 语言开发的,总代码量是 20 000 行,其中信息库管理部分是用 CodeBase 实现的.该系统主要有以下特点:1) 提供了有关目标程序的比较全面的信息;2) 利用 hypertext 技术来加速查找和切换的过程;3) 利用历史树来记录视窗切换的过程;4) 利用附注来支持局部理解成果的共享.

该系统还可以在以下几个方面做进一步工作:1) 研究能否将程序结构分析技术与基于知识的(自动)程序理解技术相结合,进一步减轻用户的理解负担;2) 研究新的针对面向对象程序的图形表示技术;3) 在用户界面部分中增加理解工作的进度储存和信息输出功能.

## 参考文献

- 1 Corbi T A. Program understanding: challenge for the 1990s. IBM System J., 1989, 28(2): 294~306.
- 2 Cleveland L. A program understanding support environment. IBM System J., 1989, 28(2): 324~344.
- 3 Brown P. Integrated hypertext and program understanding tools. IBM System J., 1991, 30(3): 363~391.
- 4 Chen Y F, Nishimoto M Y, Ramamoorthy C V. The C information abstraction system. IEEE Trans. Software Eng., 1990, 16(5): 325~334.
- 5 Grass J E, Chen Y F. The C++ information abstractor. In: USENIX C++ Conf. Proc., 1990. 265~277.
- 6 Browne J C, Johnson D B. FAST, a second generation program analysis system. In: Proc. second Int. Conf. Software Engineering, 1977. 142~148.
- 7 Linton M A. Implementing relational views of programs. In: Proc. ACM SIGSOFT/SIGPLAN Software Engineering Symp. Practical Software Development Environment, May 1984.
- 8 Zdonik S B, Wegner P. A database approach to languages, libraries, and environments. Tech. Rep. CS-85-10, Brown University Department of Computer Science, 1985.
- 9 Cheatham T E. An overview of the harvard program development system. In: Software Engineering Environments, New York, North-Holland, 1981.
- 10 Lejter M, Meyers S, Reiss S P. Support for maintaining object-oriented programs. IEEE Trans. Software Eng., 1992, 18(12): 1045~1052.
- 11 O'Brien P D, Hallbert D C, Kilian M F. The trellis programming environment. In: Proc. 1987 Conf. Object-Oriented Programming Systems, Languages and Applications (OOPSLA '87), 1987. 91~102.
- 12 Teitelman W, Mesinter L. The interlisp programming environment. IEEE Computer, 1981, 14(4): 25~34.

- 13 Hood R T, Kennedy K. A programming environment for Fortran. Tech. Rep. TR84-1, Rice University, June 1984.
- 14 Steffen J L. Interactive examination of a C program with CScope. In: Proc. USENIX Association Winter Conf., 1985. 170~175.
- 15 Sametinger J. A tool for maintenance of C++ programs. In: Proc. IEEE Conf. Software Maintenance, 1990. 54~59.

## C++ PROGRAM UNDERSTANDING SYSTEM BASED ON STRUCTURE ANALYSIS TECHNIQUE

Wu Qiong Jin Chengbai Jin Chunzhao

(Department of Computer Science Jilin University Changchun 130023)

**Abstract** This paper describes a program understanding support system for C++ language: HOOPUS(hypertext-based object-oriented program understanding support system). This system is based on program structure analysis technique, it shows the program information in a clear way to facilitate understanding the function and structure of C++ program. It also uses hypertext, multi-dimension view and navigation technique to facilitate program understanding process.

**Key words** Program understanding, object-oriented, hypertext, software maintenance.