

KBE — Pascal: 基于知识的程序编辑

郑国梁 唐 清

(南京大学计算机科学系) (江苏省人民银行)

KBE — PASCAL: KNOWLEDGE BASED PROGRAM EDITING

Zheng Guoliang Tang Qing

(Nanjing University) (The People's Bank of China Jiangsu Branch)

ABSTRACT

KBE-Pascal is an interactive programming assistant system. The intention is that the programmer will do the hard parts of design and implementation while the KBE-Pascal will assist him wherever possible.

The major underpinning of KBE-Pascal is plan, a kind of representation for programs, which abstracts away from the inessential features of a program and represents the basic logical properties of the algorithm explicitly.

The main design goal of KBE-Pascal is rapidly and accurately building up a program by referring to the fragments in the plan library. Its object language is Pascal. The current system is composed of four parts: a coder that can create program text corresponding to a plan; a library of plans for common algorithm fragments; a plan editor which makes it possible for a programmer to modify a program by modifying its plan; and a text editor which makes it possible for a programmer to edit on program text directly.

摘 要

KBE-Pascal 是一种交互式的程序设计助手系统, 其意图是程序员完成设计和实现的困难部分, 由 KBE-Pascal 尽可能地帮助他。

KBE-Pascal 的主要基础是格局, 这是一种表示程序的方法, 它抽去了程序的非基本特性, 且明显地表示出算法的基本逻辑特性。

1990 年 1 月 22 日收到, 1990 年 5 月 7 日定稿。

KBE-Pascal 的设计目标是通过引用格局库中的断片, 快速正确地构造程序。它的目标语言是Pascal。目前该系统由四部分组成: 编码程序, 用以产生对应于格局的程序正文; 一个通用算法片断格局库; 格局编辑程序, 使得程序能通过修改格局而达到修改程序的目的; 正文编辑程序, 使得程序员能直接对程序正文进行编辑。

§ 1. 引言

计算机软件发展至今, 一个主要方面就是开发软件工具, 提供新型软件技术, 以简化软件开发各个生成期的任务。

软件工具可追溯到五十年代出现的汇编程序, 它使人们摆脱了复杂枯燥的机器代码。随后, 操作系统、解释或编译系统、正文编辑程序、数据库管理系统等各类工具应运而生。

七十年代开始, 软件工具的改进和软件技术的发展, 大致分成两个方向。一种是传统的软件开发模式, 其特性表现为非形式化和未能摆脱人工方式。如: 结构化程序设计、主程序员负责制、HIPO 文档编制技术、生成周期模式等等。在此基础上, 人们寻求以计算机系统对软件开发与维护的支持。先后推出了诸如SREM 软件需求分析工具、cornell 程序综合系统、MAP 软件维护工具等等。另一种是基于自动化的软件开发新模式, 其特性表现为自动化、智能化和形式化。它是多年来人工智能和软件工程两个领域相互交叉和渗透的结晶。其最终目标是实现自动程序设计, 但目前仍处于实验性阶段。目前流行的有以下几个方面^{[1][2]}: 程序验证、演绎综合、变换综合、例示程序设计、智能化助手、自然语言规格说明、甚高级语言。

本文根据C.Rich 和R.C.Water 等提出的程序员助手^{[3][4][5]}的基本思想实现一种交互式的程序设计助手系统KBE-Pascal。又称为基于知识的程序编辑, 该系统属于智能化助手的领域。目的是帮助而不是代替程序员。其中心思想是把程序可看作由标准程序片断或格局复合而成。这样对于程序员完成设计和实现的困难部分, 系统能尽可能地随时帮助程序员去解决。如跟踪细节、编档、验证、调试和修改程序等。

§ 2. 系统特点

KBE-Pascal 系统所采用的方法是把程序视为由标准程序片断或格局复合而成, 其系统具有以下一些特点。

在功能上, 它相当于用户的助手, 帮助完成编程的一部分工作, 与其它自动程序系统相比, 其实现方法显得灵活多变。可按照设计目标, 加强或弱化某一部分的功能。

在本系统中, 一个程序无需一定要在抽象级上进行描述, 对库中知识没有自动选择功能, 加到程序中时, 也不要求它是正确无误的。从抽象级到代码间各个层次上, 系统均可以进行程序的构筑和修改。因而使用方便、灵活。

本系统作为一种编辑程序, 它能很好地处理语义, 可视为面向语法编辑程序^[6]的后继。本系统不但能理解语言的基本语法结构, 而且还能理解有关程序设计的技术和术语。

与甚高级语言不同, 本系统不是面向规格说明的, 而注重于程序合成和修改的过程。

§ 3. KBE-Pascal 系统的组成

KBE-Pascal 系统是基于知识的程序编程, 其目标语言为Pascal 语言, 它使得程序员能直接对程序的算法结构进行操作, 而不一定对程序正文结构或语法结构进行操作, 本系统的主要设计目标是使程序员通过使用格局库中的算法片断能迅速而正确地构筑出所需程序。

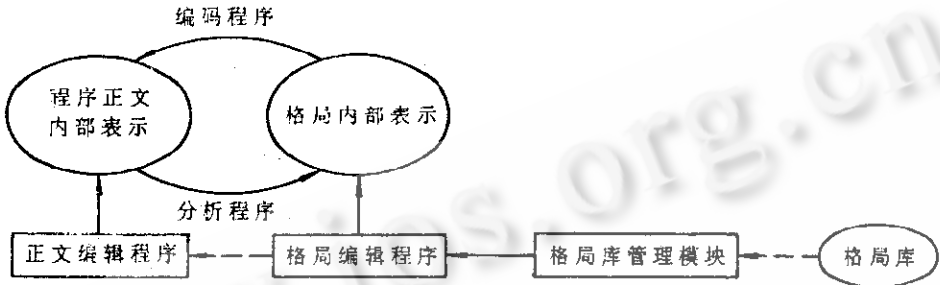


图1 KBE-Pascal 系统组成

如图1 所示, 本系统对正在处理的程序有两种表示方法: 程序正文和格局。格局是对程序的一种抽象表示方法, 它抽去了程序的非基本属性, 只显式地描述出算法的基本逻辑特性, 是整个系统的理论基础。

本系统还为程序生成一设计记录簿, 在其中存放有关该程序迄今已知的各种知识。

对应于两种表示方法, 系统提供了两种模式的编辑程序, 格局编辑程序提供一组基于知识的命令, 使程序员直接对格局进行各种操作, 命令形式用英语动词短语表示。正文编辑程序使程序员直接在源代码上进行编辑操作。上述两种操作方式可以互换, 因为编码程序和分析程序保证了程序这两种表示方法之间的一致性。

格局库管理模块作为格局库的接口, 它具备简单的推理、查找和选择功能, 并用以将格局的外部表示转换成机器内部表示。在格局库中, 使用了一种称为格局演算的形式描述方法, 刻划了一组常用的算法片断。

整个系统除编码程序的部分工作外, 其余工作基本上独立于特定的程序设计语言。

§ 4. 格局和格局演算

格局(plan, 又称计划)

一个格局类似于一分层的流程图, 并在其中显式地给出控制流和数据流表示。格局的基本单位是段, 一个段对应于一个计算单位, 它含有若干输入输出端口, 这些端口确定它接收的输入值和产生的输出值。段还有一个名字, 用来表示它所完成的运算和操作。格局中可含有子格局, 这使格局在任意层次描述程序的逻辑结构。

格局有两种表示方法, 即内部表示和外部表示。前者是为构造某一程序而具体应用到某一格局时, 在机器内存中相应地有一种表示格局的方法, 我们使用的是树型结构。

后者是指在库内对格局的表示方式,具体形式是线性存放的正文结构。在本系统中,对用户而言,这两种表示形式都是不可见的,用户所见到的只是已转换的程序正文。

格局表示的一个重要特性就是抽去了程序的一些非基本的特性,即删除用特定程序设计语言表述时程序所需具备的一些特性,仅保留实际算法所必需的特性。如控制流和数据流,在具体某种语言中如何实现和表示它们就从格局表示中略去。

格局表示的另一个重要特性是尽可能地将所表示的对象和运算局部化。格局的层次结构保证了这一点,使控制流之间毫无影响。

格局演算(plan calculus)

格局演算是C.Rich设计用来表示程序,对程序进行推理并用于描述标准的数据抽象和控制抽象库的形式体系。本系统采用它来表示格局,并作为系统的知识表示方法,来描述存放于格局库中的格局外部表示。与其它形式体系相比,格局演算具备四个特性:与语言无关、可添加性、可验证性和多层次观点。

格局在格局演算中是一种合成规格说明,由定义部分(称作角色roles)和限制部分所构成,根据角色类型,格局分为数据格局和时态格局(简称格局)两类,分别体现了数据抽象和控制抽象。本系统对运算的描述分段和格局两类,段中有简单段和分岔段之分,分别对应于操作和测试。在本系统中,段视为终结段,即可用具体的目标语言来实现。格局则视为非终结段。以下简略给出相应的描述方法。

(1) 简单段

Segment: 段名: Simple; (* Simple 表明简单段*)
input: 对象名(属性)表; (* 对象名亦即输入输出端口名*)
output: 对象名(属性)表; (* 属性表示可能的数据类型*)
body: 运算体 (* 用类Pascal 语言刻划*)

(2) 分岔段

Segment: 段名: Split;
input: 对象名(属性)表;
test: 测试条件

(3) 格局

plan: 名
roles: 角色名(属性)表; (* 角色是计划的组成部分*)
dflow: 数据流表;
cflow: 控制流表; (* 控制流表示为(名, 名)*)

§5. 系统的实现

格局的内部表示

格局的内部表示是系统理解程序的基础,考虑到本系统最基本的两种操作:一是通过记录在设计记录簿中的知识对程序进行推理,以确定如何实现数据流和控制流;二是将不同的格局片断连接在一些,以实现预期所要完成的目标。所以在本系统中,用于表

示程序的格局就是一个由数据流和控制流连结起来的操作网络。格局可以分段成由子段构成的层次结构, 用两个指针 Link, Subplan: planptr 来表示, 其中 planptr 是指向格局说明的指针类型, Link 用于链接处于同一层次的格局结点, Subplan 则指向该格局的第一个子段。

一张格局图中, 各结点之间除了存在上述两种连接关系外, 处于同一层次属于某一结点的那些子结点之间还存在两种限制: 控制流和数据流。由于控制流反映的是程序的控制结构较复杂, 许多结构仅用上述两种指针无法描述, 为此设立了新指针 controlfrom, controlfrom 1: planptr 来描述。这里为简化问题, 规定格局说明中每个结点只允许最多有两个指针指向它。

对数据流对的处理, 将前者的内容填入到后者中, 在最后编程时, 用新内容代替旧端口名就可实现该数据流。

在格局图中, 各结点可以为段、数据格局和格局, 在说明中就用变体记录的形式加以综合说明。

程序正文的内部表示

程序正文用以显示又是格局操作的最终结果。为方便处理, 程序正文采用双向链表结构表示, 并设立一指针域用以指向该程序正文是由哪一个格局结点生成的。

格局库以及格局库管理模块(PLMM)

本系统使用 Modula-2 系统中流式文件表示格局库, 每一个功能上独立的运算或说明单位都作为单独的逻辑块存放在格局库中。

格局间的关系目前有四种: 一个格局是另一个格局的组成部分; 一个格局是另一个格局的一种特例; 一个格局是另一个格局的一种推广; 还有一种体现多层次概念的覆盖关系。PLMM 的推理部分主要是得到格局相互间的上述关系。

格局库管理模块是格局库的接口, 下一步应演变为知识库管理系统(KBMS)。它将具备演绎、推理、自动选择、自动匹配、自动查找等功能。目前 PLMM 只具备简单的推理、查找和选择功能。PLMM 通过查找索引, 可将系统需要的格局找出。它还可将格局的外部描述转换成格局的内部表示, 并初步处理数据流和控制流。

编码程序

编码程序将分段的内部表示转换成可执行的 Pascal 源代码, 生成的目标程序结构类似于分段格局的结构。

在实现编码程序时, 主要解决三方面的问题: 处理循环和分岔结构; 数据流的实现方法; 当程序太零碎时, 即不能生成可执行的程序代码时, 如何表示和处理相应的程序正文。

基于知识的格局编辑程序

基于知识的格局编辑程序用于支撑一组命令, 使得程序员能直接按照算法结构编辑程序。命令语言用简单的英语动词短语方式表示。由动词关键词后接一个或若干个名词短语构成, 命令格式:

Keyword reference with Instance

引用(reference) 是由格局生成程序时对角色的引用或指称。这里的角色可以是格局名、段名、端口名或目标代码中未实现的部分。具体引用哪一个角色, 它是根据当前的格局位置, 亦即由程序正文的当前指针来决定的。

命令语言的另一个基本成份是引用格局库中的逻辑功能块,以此来构造程序,该成分称之为实例(instance)。实例分成两类: Pascal 源语言代码和对格局库的引用。

目前KBE-Pascal系统所支撑的命令有Define, fill, remove, replace, insert, coyp等等。

正文编辑程序

本系统提供全屏幕正文编辑方式,程序员在该状态下可直接对程序正文进行操作,使得程序员能方便地实现尚未完成的细碎部分,最终得到可执行的程序代码。

§6. 结 语

整个系统使用Modula-2语言编写,在PC/AT微机上实现。由于系统颇大和时间所限, KBE-Pascal系统的功能尚待完整,其中分析程序基本上尚未实现,但考虑到软件工程环境的发展方向,着手于这样一种新型的程序设计模式的探讨与研究是很有价值和前途的。

参考文献

- [1] D.Bartow, A Perspective on Automatic Programming, Proceedings of 9th International Conference on Software Engineering, 1987.
- [2] C.Rich & R.C.Waters, Automatic Programming: Myths and Prospects, Computer Vol.21, No.8, 1988.
- [3] R.C.Waters, The Programmer's Apprentice: Knowledge Based Program Editing, IEEE Trans. on Software Engineering Jan. 1982.
- [4] R.C.Waters, The Programmer's Apprentice: a Session with KBE macs, IEEE Trans. on Software Engineering Nov. 1985.
- [5] C.Rich & R.C.Waters, Artificial Intelligence and Software Engineering, Morgan Kaufmann, Los Altos, California, 1986.
- [6] 郑国梁, 李辉, 面向多种语言的语法制导编辑系统SEG的设计与实现, 计算机学报, Vol.10, No.2, 1987.

关于召开全国机器翻译学术研讨会的征文通知

为了促进国内机器翻译技术的发展,增进同行间的交流与合作,国家高技术智能计算机系统专家组与中国计算机学会决定联合举办“全国机器翻译理论与技术”学术讨论会。征文范围包括与机器翻译有关的基础理论和应用技术,大致包括(但不局限于)以下几个方面:一. 基础理论 1. 各种文法体系 2. 词典语义学 3. 知识表示 4. 机器学习 5. 自动文摘 6. 篇章语言学 7. 语料库语言学 8. 分析算法 9. 自然语言接口 10. 电子词典 11. 对话现象处理 12. 自然语言知识库系统 13. 语用理论 14. 话语生成 15. 专用软件环境 16. 专用机硬件设计等等。二. 机译系统开发 1. 国内外机译系统介绍 2. 机译发展前景预测 3. 自动翻译与机助翻译 4. 基于语法的机译系统设计 5. 基于语义的机译系统设计 6. 基于知识的机译系统设计 7. 基于智能的机译系统设计 8. 基于语料库的机译系统设计 9. 其它机译系统设计技术 10. 机译评估技术 11. 机器词典开发 12. 机译文法设计 13. 生成技术 14. 机译软件设计 15. 译前编辑与译后编辑 16. 源文自动输入技术等等。

会议日期: 1992年4-5月 地点: 长沙(暂定)

稿件要求: 会议将通过出版社出版文集,来稿应为国内外未公开发表过的文章,在理论或应用研究方面确有创见,叙述清楚,行文流畅。来稿不超过8000字,附中英文摘要,分别不超过200字,一式四份,书写整洁(最好软盘加打印),不退稿,请自留底稿。

截稿日期: 1991年12月31日(以邮戳为准) 通知接受日期: 1992年2月1日

来稿请寄: 北京2704-2信箱 刘群 韩向阳 赵丽云 电话: 2565533-525或526