

程序设计与知识抽象

张正轴

广西区科委

摘 要

人们正努力从传统的程序设计观念中逐步解放出来, 寻求一种更远离计算机而更接近问题本质的程序设计思想与方法。知识抽象是程序设计过程中萌发并在实践中不断完善的、有益的思想。本文提出了知识抽象对程序设计、对计算机软件产业的重要作用的观点, 以期引起软件工作者对知识抽象的重视。

对于计算机科学, 人们通常都简单地将它理解成是研究计算机和研究如何编写解决问题的程序的科学, 或者说, 计算机科学要解决的问题是, 什么是计算机以及如何编写你需要完成的任务的程序。这一观点的结果将程序看成与计算机关系非常密切, 认为程序的运行就是让计算机以特定的方式进行工作。然而, 在软件开发的历史上, 一个重要的思想已逐渐明确并正在不断完善, 即知识抽象。它使程序逐渐远离计算机并更靠近问题。

应用机器语言或汇编语言编写程序, 被认为是“直接”使计算机按照程序员的意愿办事的方式。由于程序被看成是计算机行为的工具, 因而构造这样的工具成了程序员的基本工作。他们去编写程序, 使计算机完成所期望的问题级的活动。于是, 产生这样的结果, 首先是, 在一般情况下, 程序中所期望的问题信息是不可见的。当问题级信息已被翻译成计算机级的指令时, 有关问题的全部信息都被“隐藏”起来了。问题缺乏可见性, 使程序员及其他人处于实际上的不可理解的状况之中。由此而产生的另一个结论是软件“不软”: 如果不依赖专家, 就不能在原系统基础上重新设计程序, 系统的工作也就不能改变。由于缺乏支持软件重使用和重设计的软件环境, 用户不可能利用已有的软件成果作进一步的开发, 造成软件资源的浪费。

人们为使程序设计节省劳动力, 为了程序的简洁, 创造了子程序、宏汇编。在实践中, 人们很快认识到子程序等技术实质上是一种抽象程序功能的方法。通过使用(调用)子程序, 有助于理解和进一步开发系统软件。子程序、标准函数块的显著特点是定义准确、结构清晰、功能单一。这些基本模块具有明确简便的接口。所有这些, 使得用户能较好地利用原有软件资源开发出新的软件。在此认识的基础上, 人们又努力开发面向领域的语言。通过实践, 尽管未能得到满意的结果, 却产生了三个重要的观念: 1. 计算模式被从原来的依赖于硬件结构的情况下解放出来; 2. 将以领域术语表达程序的思想作为一个有效的目标。构造面向领域的语言, 就可以去设想人们最少忘记基本计算模型的存在, 而以领域术语编写程序; 3. 程序变成了研究的合理客体, 它既独立于计算机本体, 又独立于它所解决的某一具体的问题。

这些思想进一步发展,促使了对程序设计语言的研究。当程序设计语言Lisp把函数求值过程抽象成计算机模式,并把数据结构抽象成表示函数的代表程序的工具时,就使得程序变成为研究的合理客体。Lisp是一种计算机的表处理语言,是迄今在人工智能科学领域中应用较多的一种程序设计语言。人们认为这种语言曾武装了一代人工智能科学家。P.H. Winston认为Lisp语言是人工智能的工具,不仅对人工智能的机器实现有重要意义,而且也是人工智能理论研究的重要工具。Lisp把复杂的计算过程从实际计算机中抽象出来,使之在程序设计中易于实现。在某种意义上讲,它使程序变得更独立于计算机。Lisp在程序表达式中提供了程序控制,于是就允许程序员能够动态地生成、修改并执行程序,也使人们获得编制程序过程的工具。

在六、七十年代中,作为对在程序里缺乏问题可见性的反应,出现了软件工程革命。这导致了抽象数据类型和过程抽象的形成。利用这些研究结果,综合起来将促使面向对象程序设计的实现。可以说,模拟语言Simula67就是面向对象程序设计思想的一个尝试。

为了实现真正的面向对象的程序设计,知识抽象的研究再一次提到了人们的面前。人们必须对问题级的描述以及所期望的结果的说明作逐步提炼和程序验证。人们试图把描述提炼为愈来愈小的子问题,直至达到所应用的计算能力就可以立即解决子问题的地步。在程序验证方面,人们说明所期望的问题级的活动,然后试图去确保所产生的程序和说明一致。然而,有关逐步提炼的工作仅开始于程序设计风格和技术的指导,而不是直接把问题级的信息并进程序中。程序验证仅导致了自动程序设计研究工作,产生了“可执行性说明”等少数可应用的结果。

通过逻辑程序设计的研究和专家系统的建立,人们越来越认识到,在计算机程序中,真正重要的是“知识”。代表程序与问题领域的唯一真实连接是由程序实现的领域的知识。这是R.J.Abbott提出的知识抽象的原则。我们用“知识程序设计”来指那些能明确“见到”领域级知识的程序设计。一般说,在软件中的抽象已经给程序提供了语言工具,借助这些工具,可以直接论及抽象元素并把它们应用于程序设计。过程抽象使程序员能够命名和定义子程序,句法抽象使程序员能够定义指定领域的语法和面向领域的语言的语义,数据类型抽象使程序员能在程序中命名、定义和论述数据类型。通过知识抽象,程序员可以直接论述该程序所表达的知识,至少是部分知识。知识程序设计对程序员的影响和注意力,是在连接程序与需解决的问题的那些元素上。因此,利用知识抽象,程序员可以对问题级知识进行程序级控制,使程序向直接涉及领域问题迈进了一大步。

可以肯定,知识抽象的研究,必将对程序设计带来极其深刻的变革。

参 考 资 料

1. R.J.Abbott, 知识抽象, ACM通讯, 第30卷, 第8期, 1987年8月
2. Gerhard Fischer, Cognitive View of Reuse and Redesign, IEEE SOFTWARE, 1987年7月
3. P.H.Winston, B.K.P.Hora, Lisp程序设计, 清华大学出版社, 1983年

PROGRAM DESIGN AND KNOWLEDGE ABSTRACTION

Zhang Zhengyou

(Science and Technology Commission of Guangxi Zhuang Autonomous Region)

ABSTRACT

People are doing their best to release themselves gradually from the traditional concept of program design and making a search for a new kind of thought and method about program design which is more independent of computer but more dependent on essence of problem. Knowledge abstraction is a useful thought which germinates in the process of program design and perfects itself gradually in practice. In order to attract the software researchers to pay attention to knowledge abstraction, this article has pointed out an idea that knowledge abstraction plays an important role in program design and software industry of computer.