

# 逻辑程序设计对软件开发的影响

余金山

[计算机科学(电脑)系]

**摘要** 本文研究逻辑程序设计,广泛地讨论了它在软件开发实践中的作用及其对软件工程的影响。

**关键词** 逻辑程序设计,软件工程,软件开发

## 0 引言

逻辑程序设计(LP)具有许多突出的,独特的优点,如严密、牢固的理论基础;简明易懂的语义,允许显式地表示知识,高度的非过程性,面向问题求解的说明性编程,程序简洁,易学易懂等等。因而,它被誉为是软件新技术的最具典型的代表,是人工智能、软件工程、数据库、计算机体系结构的核心和桥梁,可以统一起这些领域中的全部结果<sup>[1]</sup>;代表了新一代计算程序设计的主流<sup>[2]</sup>。特别地,逻辑程序设计还由于被日本选中作为第五代计算机(FGCS)的核心而引起了全世界的关注。FGCS的主要负责人 Kazuhiro Fuchi 等人在文[3]中写道:我们想强调 FGCS 研究的关键特征不是知识信息科学,也不是 Non Von Neumann 体系结构,而是逻辑程序设计。

高效率、低成本地开发高质量的软件是软件工程研究的主题,是计算机界甚至整个信息界努力的最重要的目标之一,也是最迫切、最困难的问题之一。那么,逻辑程序设计作为典型的软件新技术,新一代计算程序设计的代表,或退一步而言,逻辑程序设计语言作为软件开发的最基本的工具——语言,能在软件开发实践中起些什么作用?可能对软件工程产生什么影响呢?本文试图对这些问题作一定的解答。

## 1 逻辑程序设计在软件开发中的作用

逻辑程序设计在人工智能领域内已受到了充分的重视。在专家系统、自然语言处理、定理证明、抽象问题求解、智能数据库等人工智能领域乃至新一代计算的一般性应用领域——知识信息处理<sup>[2]</sup>都得到了广泛的应用,并取得了良好的效果。但是,逻辑程序设计(语言)作为一

• 本文 1991-04-18 收到。

种软件新技术和新一代计算的程序设计语言,它(们)除了能处理新一代计算的新的应用问题之外,还应该能够支持且应更好地支持传统的、老的应用。

从应用的观点出发,逻辑程序设计的理想目标是使得该技术成为一种能够广泛地支持各种应用(甚至包括实时的工业控制)的基础技术,而逻辑程序设计语言应成为一种有效的通用性语言。因而应用逻辑程序设计进行软件开发,可以取得理想效果。(1)技术、工具(直至整个支持环境)、数据库和程序形式上的高度统一——基于逻辑。(2)程序的高度简洁性,逻辑程序设计的说明性编程特点允许用户只写出关于某一领域问题的有关知识作为公理以及待解决的问题作为定理,而不必关心问题求解过程的任何细节。(3)缩短软件开发周期。在形式上,一个逻辑程序与一个基于逻辑程序设计的规格说明没有什么区别。从这一点看,逻辑程序设计使得软件开发免去了设计和编码阶段的工作(最多需要一定的程序变换以提高效率),测试与验证工作也变得更加容易和可靠。以上几点再加上逻辑程序设计的良好数学基础,可以在很大程度上提高软件生产率和软件质量,而且十分有利于软件生产的自动化,从而在根本上解决“软件危机”问题。

尽管由于存在着一些有待解决的困难(将在后面加以讨论),逻辑程序设计在目前还未能实现以上所讲的理想目标,它至少可以(或已经)在软件开发的下列几个方面作出不同程度的贡献。

### 1.1 形式化方法

逻辑程序设计是基于—阶谓词逻辑的,具有坚实、严密的理论基础;精确、简明的语义。模型论的说明性语义,不动点的指称语义和证明论的操作语义。逻辑是已知的一种功能最强和通用的形式描述语言。基于逻辑的形式方法与基于编译技术的形式方法和基于代数映射的形式方法并称为目前流行的三大类形式化方法<sup>[4]</sup>。而形式化与工程化被认为是软件工程的两个必不可少的方面<sup>[5]</sup>。文[4]和[6]对形式化软件方法的重要性进行了详细的论述。

### 1.2 规格说明

规格说明必须是精确的,相容和完备的。但要达到这一目标十分困难。基于LP的规格说明,有如下五方面的优点:(1)有利于进行数学验证。文[7]对数学验证的重要性、要求进行了深入的论述,并给出了令人鼓舞的实际结果。(2)基于LP的规格说明是可执行的。(3)逻辑程序的“双向性”(或称为可逆性)、“多用途”和演绎功能对规格说明的测试和排错十分有益,它将有助于发现规格说明中存在的某些较难发现的错误。(4)从形式上讲,一个逻辑程序与一个规格说明没有什么差别,因而开发逻辑程序的方法与工具也可用于支持规格说明的产生。(5)逻辑程序设计语言是一种合适的规格说明语言。它不但具备形式规格语言应有的四个特征<sup>[8]</sup>,即:(a)严密的数学定义;(b)足够强的表达能力;(c)能支持原型开发;(d)便于从非形式的需求定义转换为形式化的规格说明。而且比基于代数的规格说明来得浅显易懂,比较容易为用户和开发人员所接受。例如,欧洲共同体 ESPRIT 计划开发的系统 ASPIS(Application Software Prototype Implementation System——一种基于知识的 CASE 环境)中所采用的规格说明语言 RSL(Reasoning Support Logic)及其实现都是基于逻辑程序设计的。另外在一个作者称之为具有革命性意义的系统 CAEDE<sup>[9]</sup>中则直接使用 PROLOG 语言作为规格说明语言以及其它设计过程的描述语言。

### 1.3 快速原型方法

逻辑程序设计语言,尽管存在某些局限性,但仍不失为一种良好的通用原型语言.特别是对于所谓的“探索型”和实验型”或“抛弃型”的原型方法,基于 LP 的快速原型方法有其独特的优点,得到了广泛的应用.如前面提到的 ASPIS 系统原型支持模块的实现就是基于 LP 的.基于逻辑程序设计的原型方法的若干突出特点:(1)可以简洁地表示出系统的主要功能.(2)由于 PROLOG 语言的普及与改进,原型的实现既迅速、方便简单、经济.实际经验表明 PROLOG 程序的构造与修改比 LISP 还要容易得多,而 LISP 的编程效率是 C、Ada 等传统语言的两倍.(3)可以灵活地选用多种原型建造方法.如,可执行规格说明、程序变换方法、增量式设计等等.(4)有利于应用人工智能的研究成果.

#### 1.4 开发智能性的软件工具和软件系统中的智能部件或有关模块

用 PROLOG 开发智能性的或一般的软件工具已得到了较多的探讨和应用,并被认为是一种支持工具开发的有前途的方法<sup>[9]</sup>.用 PROLOG 开发嵌入到用传统语言开发的软件系统中的某些模块(智能的或非智能的),或反之,从 PROLOG 程序调用传统语言程序模块;或更一般地,开发 PROLOG 和传统语言相结合的软件系统;其重要性和益处近年来已得到了一定的认识.在逻辑程序设计语言存在的缺陷得到解决之前,这是一种实用的策略,具有重要的现实意义.一些较新的语言产品已提供了 PROLOG 和传统语言两者之间的接口功能,为这样的应用提供了基础.例如, Turbo PROLOG 不但提供了与 C、PASCAL 汇编等语言的接口,通过其工具集 Toolbox,还可以访问 Reflex, dBASE ■, Lotus-1-2-3 等系统的数据文件;英国 Alvey 计划使用的主要工具 POPLOG 允许开发人员同时使用 PROLOG, LISP, POP-11, Fortran, C, PASCAL, BASIC 等语言,并把它们融合成一个完整的软件系统.

#### 1.5 与面向对象的设计方法相结合

面向对象的计算是一种新的、有前途的方法.并发逻辑程序设计语言,如 Concurrent PROLOG,对于实现面向对象的程序设计不但具有充分的表达能力,而且还为面向对象的方法提供了一定的形式化基础.一些研究指出,把面向对象的数据库与 LP 相结合也是有意义的.

#### 1.6 开发人机界面

随着计算机使用的日益普及,人机界面的作用也越来越重要,近年来受到了充分的重视.开发具有智能形为的和基于自然语言的人机界面是重要方向之一.而逻辑程序设计语言则是实现这种人机界面的强有力的工具.逻辑程序设计语言具有良好的演绎推理功能,且研究已经证明,编译技术中的扩充属性文法以及用来分析自然语言的各种广义短语结构文法,都可以看做是 Horn 子句逻辑记法的变种.事实上,逻辑程序设计语言的主要代表 PROLOG,最初就是作为自然语言理解的工具而开发出来的.

#### 1.7 与演绎数据配合

演绎数据库是当今数据库技术发展最重要的方向之一<sup>[10]</sup>.对于以演绎数据库为中心的应用系统的应用程序的开发,采用逻辑程序设计语言作为主语言或查询语言具有许多优点<sup>[10]</sup>.逻辑程序设计语言也适合于开发传统关系数据库(甚至包括层次和网状的)的应用程序.实际上,传统的关系数据库可被看成是演绎数据库的一种特例.

#### 1.8 其它

除了以上介绍的作用之外,逻辑程序设计还对软件开发的下列五个方面有积极的影响.

(1)软件测试.对于动态测试的关键问题——测试实例生成,基于 LP 的方法是一种可行的且

十分有效的方法. 对于静态测试的符号执行、测试自动化以及近来发展起来的基于知识的测试技术, 逻辑程序设计的思想和语言也是十分有用的. (2) 用于加强软件开发的某些侧面. 例如, 加强配置管理和判定表(或判定树)的定义与实现. (3) 支持软件开发新模式. 例如 PLEASE<sup>(11)</sup> (一种结合逻辑程序设计和强制式语言特点的语言)对增量式软件开发的支持. 文[9]还论述并证实了 PROLOG 语言在支持可视范例软件开发模式中的重要作用. (4) 逻辑程序的简洁性和说明性对维护工作十分有利. (5) 社会影响. 其一是如文[2]所说的, 逻辑程序设计的简单性对软件开发人员的心理具有积极的影响; 其二是逻辑程序的广泛应用将有助于人们改变旧的思维方式和传统的编程观念, 增加对新一代计算的认识.

## 2 问题与前景

以上我们讨论了逻辑程序设计在软件开发中可能产生的作用. 从理论上讲, 逻辑程序设计确实具有许多非常突出和吸引人的优点, 有些方面在八十年代初就已进行了一定的探讨. 但是至今跟普遍的实际应用仍有很大差距. 这虽然也符合软件工程中实践落后于理论的普遍规律(一般认为, 一种软件新技术或新概念的出现, 约需经过 15 至 20 年的时间方能为工业界所普遍接受), 但 LP 要在软件开发中真正发挥应有的巨大作用还有待于如下四个问题的解决. (1) 提高逻辑程序设计语言系统的实现效率. 目前, 作为 LP 语言主要代表的 PROLOG 语言仍然存在着时空效率差的重大问题. 一般认为, 这是限制 LP 语言广泛应用的主要原因<sup>(12)</sup>. (2) 逻辑程序设计语言本身的改进和完善. 这方面包括语言的语法结构和语义结构的改进. 现有的 PROLOG 语言, 从逻辑程序设计的角度看, 尚与理想的逻辑程序设计语言存在一定的距离; 从软件工程的观点看, 则对数据抽象、信息隐藏和模块化等原则缺乏有力支持手段. (3) 逻辑程序开发方法学和开发工具的建立. 由于逻辑程序设计的独特风格和新颖性, 传统的开发方法和开发工具已不能适应或不能完全适应逻辑程序的开发, 从软件工程的观点看, LP 具有许多优点<sup>(13)</sup>, 但也对软件工程提出了新的要求. (4) 人们观念的改变. 逻辑程序设计代表了一种完全不同于传统程序设计的观念和思维方式. 正如文<sup>(14)</sup>所提出的, 这种技术的推行在很大程度上依赖于人们习惯的改变.

前两个问题一直是人们积极研究探讨的方向, 并已有较大的进展. 在提高 LP 语言的实现效率方面, 对编译技术、智能回溯、索引结构、并行算法以及其它有效的优化措施都进行了深入的研究. PROLOG 系统的执行速度在不断提高, 可望达到传统语言的水平<sup>(11)</sup>. 在语言本身的改进方面, 对于控制自动化、逻辑非的实现以及基于多种推理规则的非子句逻辑程序设计系统的研究, 也取得了不少有用的成果. 例如 NU-PROLOG 不但较好地实现了控制的自动化和逻辑非问题, 还提供了强有力的数据库设施, 从而向纯说明性的逻辑程序设计迈进了一大步, 并改善了语言的可应用性. 此外, 有不少 PROLOG 语言提供了模块化和元程序设计功能, 实现了与其它语言的接口, 这也大大地提高了语言的可应用性.

对于方法学和工具的研究, 近年来也得到了充分的重视. 一些有用的工具已得到开发或正在开发. 但对方法学的研究则没有什么大的进展.

观念的改变是需要一定的时间过程的. 因此值得研究的是如何采取一些有效的措施来缩短这个过程. 教育已被证明是促进人们观念改变、加速技术转移的最重要、最有效的手段之一.

随着逻辑程序设计课程的普遍开设和逻辑程序设计思想的广泛传播,人们对逻辑程序设计的认识必将日益加深.

此外,硬件技术的惊人发展也给予 LP 有力的支持.例如,直接以逻辑程序设计为核心的 FGCS 的进展以及其它新一代计算机的研制;在传统的计算机方面,目前,386 微机系统的内存已达 16MB(并备有 64MB 的虚存);16MB 的动态 RAM 芯片在两年前就已研制成功;性能接近巨型机 Cray-1 的微处理器 i860 已进入市场.据最新预测,到 2000 年,VLSI 技术将可在 1 平方英寸的晶片上集成 5 千万只晶体管,因此 2000 年的微机将可配置多个 CPU,其运算速度将超过 2000MIPS(29 亿条指令/S),综合性能将超过今天的巨型机.逻辑程序设计的创始人 R. Kowalski 认为,目前限制 PROLOG 语言的更广泛应用的主要原因并不在于 PROLOG 本身的缺陷,而在于实现资源的不足<sup>[12]</sup>.按照这个观点,硬件技术的发展前景无疑给 LP 的广泛应用带来了极大的希望.

### 3 结论

本文讨论了逻辑程序设计在实际软件开发中的作用.LP 作为一种具有代表性的软件新技术,不但能支持以知识信息处理为中心的新应用,在支持以数据处理为主的传统应用方面也具有许多突出的优点.逻辑程序设计的良好思想和技术对软件工程具有重要且深远的影响.但是,LP 要在实践中得到广泛的应用还面临着若干有待解决的问题.

探讨逻辑程序设计对软件开发的影响属于 AI 与 SE 交叉研究的一个子领域,其普遍、深远的意义是明确的<sup>[6]</sup>.但就目前的情况看,作为短期目标,我们认为加强下面五个方面的研究工作将具有更加重要的现实意义.(1)逻辑程序设计语言与传统语言和其它 AI 语言的接口,或结合(即所谓的多范例语言).(2)多语言集成环境的研制和推广使用.如,英国 Alvey 计划的主要实现工具 POPLOG 和我国研制的 XYZ 系统.(3)从实践(SE)的观点出发,对 LP 进行研究.(4)加强实例分析,让更多的人分享有关的实践经验和教训.(5)加强教育和培训工作,促进 LP 思想的传播和技术转移.

大力发展软件工程,创建中国的软件产业是我国计算机界当前及今后相当长一段时间内的重要目标,因而正如文[5]所指出的,对新兴技术的探索、研究和应用尤为重要.

### 参 考 文 献

- [1] Kowalski, R. B., 新一代计算中的软件工程和人工智能, 计算机科学, 5(1986).
- [2] 鲁汉榕, 新一代计算的若干问题, 计算机科学, 2(1989).
- [3] Kazuhiro Fuchi, et al, 逻辑程序设计在 FGCS 计划的作用, 计算机科学, 4(1988).
- [4] 唐稚松, XYZ 系统的设计思想, 软件学报, 1(1990).
- [5] 扬美清、徐家福、朱三元等, 软件工程的发展与我们的设想, 计算机技术, 3(1989).
- [6] 余金山, 软件工程中的 AI 思想及其研究, 计算机科学, 6(1989).
- [7] Mills, H. D., et al, *IEEE Software*, 4, 8(1987), 19—25.
- [8] 陈火旺等, 基于 Ada 的软件开发新模式, 计算机科学, 4(1989).
- [9] Buhr, R. 等, 软件 CAD: 一种革命性方法, 计算机科学, 6(1989).

- [10] 余金山, 演绎数据库的研究与发展, 计算机科学, 3(1990). ,
- [11] Twerwilliger, R. B. and Campbell, R. H. , *J. Systems and Software*, 10, 2(1989), 151—165.
- [12] Kowalski, R. , *Logic Programming*, in Proc. IFIP 83, R. E. A. Mason (Ed. ), Amsterdam, North-Holland, (1983).

## The Effects of Logic Programming on Software Development

Yu Jinshan

(*Department of Computer Science*)

**Abstract** This paper deals with logic programming. The role it plays in the practice of software development and its effects on software engineering are extensively discussed.

**Key words** logic programming, software engineering, software development