B语言与方法

——实用形式化开发指南

[英] K.Lano 著　鲍涌　刘建元　邹德林　译

高等教育出版社

B语言与方法

——实现形式化开发指南

B语言与方法

——实用形式化开发指南

[英] K. Lano 著
鲍涌 刘建元 邹德林 译

高等教育出版社

（京）112号
图字：01-97-0405号

图书在版编目(CIP)数据

B语言与方法：实用形式化开发指南/(英)拉诺(Lano,K.)著；鲍涌等译．—北京：高等教育出版社，1998.6
书名原文：The B Language and Method：A Guide to Practical Formal Development
ISBN 7-04-006411-1

Ⅰ.B… Ⅱ.①拉… ②鲍… Ⅲ.B英言-程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字(98)第15375号

*

高等教育出版社出版
北京沙滩后街55号
邮政编码：100009 传真：64014048 电话：64054588
新华书店总店北京发行所发行
化学工业出版社印刷厂印装

*

开本 850×1168 1/32 印张 9.875 字数 250 000
1998年7月第1版 1998年7月第1次印刷
印数 0 001—2 096

定价 10.70 元

凡购买高等教育出版社的图书，如有缺页、倒页、脱页等质量问题者，请与当地图书销售部门联系调换

内容简介

本书介绍了B抽象机符号语言以及用于支持形式化规格说明和高集成系统开发的方法。首先，介绍了B语言的发展历史及与其他语言的关系，然后，从集合、序列等基本数学符号表示到B语言的结构化机制，再到它所支持的大型程序设计方法，全面地介绍了B符号语言的描述方法。本书可作为计算机专业本科生或研究生的形式化方法课程教材，也可作为指导实际开发的参考书。

译者的话

大型软件系统的开发，需要采用工程的开发方法。在传统的软件工程开发中，人们通常采用自然语言加图形符号来描述软件系统的功能和结构。用自然语言书写系统规格说明，其优点是便于人们阅读和理解其含义，但它也存在着明显的缺点，这就是自然语言的描述往往是不精确的，有时是有歧义的。

形式化方法的主要思想，是以一种精确的方法严格地描述软件系统的定义与需求，并且这种精确的表述方法是建立在严格的数学基础上的。它通过集合、谓词逻辑、函数映射等数学方法，保证软件系统从规格说明到代码生成每一阶段的每一次形式变换都有严格的数学推演和正确性证明，从而保证其语义不变。

B语言与方法就是这样一种支持从规格说明到代码生成整个开发过程的形式化方法。它使用相对简单且为人们所熟悉的集合符号表示法来描述抽象机的状态变换，从抽象机开始，逐步细化，直到最后的实现程序。鉴于B语言的这些特点，我们翻译了K. Lano所著的《B语言与方法——实用形式化开发指南》一书。本书可作为计算机专业本科生或研究生软件工程课程的辅助教材或教学参考书，也可供从事形式化方法研究、应用的研究人员和开发人员阅读参考。

本书前言及第一、二章由鲍涌翻译，第三、四章由刘建元翻译，第五、六章及附录由邹德林翻译。付梓前，高等教育出版社姚玉洁编审仔细审读了译稿，提出了许多宝贵意见，在此谨表谢意。

限于译者水平，难免有错误和不妥之处，请广大读者批评指正。

译　者

1998年5月

序

本书综合介绍B抽象机符号语言，以及用它来支持形式化规格说明和高集成系统开发的方法。本书目的有二：一是能够作为大学本科生或研究生的教材；二是给产业界用户提供实用的入门书，内容涉及B语言结构化机制的使用、证明的使用以及与代码生成有关的特殊问题。关于B语言语义的更深层问题将在附录中给出。

本书首先讨论B语言的历史及其与其他语言的关系，然后将从集合、序列等基本的数学符号表示，到B语言的结构化机制，再到它所支持的大型程序设计，逐步构造出B符号表示的描述方法。我们从现存软件开发方法，特别是面向对象分析与设计的角度，着重介绍B语言的使用。

第一章介绍B AMN概况、历史以及相对于其他形式化方法而言其所处的地位。

第二章简要描述B语言开发过程中的每一阶段。

第三章深入描述从图形符号表示的分析模型到一个完全形式化规格说明的演变过程概况，展示了从OMT符号表示到B AMN规格说明的系统化转换，同时讨论了内部一致性证明和动画的有效性验证技术。

第四章给出从设计到实现的细化过程，包括细化的证明、形成组件开发与复用基础的分层开发范例以及C语言的代码生成。

第五章列举采用此开发过程的两个扩展性实例研究。

每章配有不同难度的B AMN规格说明与开发过程的习题，并附习题答案。全部习题均通过了分类测试。

附录A列出了习题答案。附录B提供了B AMN语义方面更

进一步的信息。附录C介绍了一些适用于B语言的证明技术。

本书附有术语和符号索引以及参考文献。

致谢：

我要感谢(伦敦)帝国学院的同事们对本书内容的形成与发展所给予的支持。这些内容已用作理科硕士生(MSc)的"规格说明理论与验证"课程的一部分。我特别要感谢Tom Maibaum，是他鼓励我使用B语言来讲授形式化方法。Howard Haughton 和 Krysia Broda 对本书的技术内容作了大量有益工作。Steve Schuman，Jeremy Dick，B-Core（英国）有限公司的 Ib Sørensen 和 Dave Neilson 对本书文字表述给予了很大帮助和指导。

K. Lano

目　录

第一章 引 言

B抽象机符号规格说明语言目前正为产业界和学术界越来越多的人所关注，它最初的研究工作是在20世纪80年代的初期和中期由J. R. Abrial以及BP研究中心的MATRA和GEC Alsthom研究小组进行的。它是少数几种具有较强商品化工具支持的形式化方法之一，支持从规格说明到代码生成全部开发周期。同时它还继承了它的前身——Z语言的优点，即基于人们所熟悉且便于理解的数学基础。

在本章中，我们将概述用B语言进行软件开发的过程，列出迄今仍在使用的一些语言的应用。我们还将通过与其他一些使用面较广的语言和方法相比较的方式来试图说明B语言是一种比其他形式化技术更具有商品化和工业化的适用语言。

1.1 形式化方法

“形式化方法”一词在软件工程与计算学科领域中引起了强烈的反响，它既包括把形式化技术看成是软件工程中新主流的积极性反响，也包括认为它过于侧重形式化方法研究工作的怀疑性反响。

用所谓的B AMN形式化方法，我们可以简单地借助于数学符号表示法，如谓词逻辑、表示集合的符号、序列、函数以及其他抽象数据类型，以一种精确的方法来描述软件系统的需求与设计。另外，它不仅是一种符号表示法，正像B符号表示法本身的含义一样，它支持对B AMN规格说明特性的数学推理，包括规格说明的内部一致性验证(即包括一个能由可实现系统满足的公理系统)和细化过程的验证(设计步骤)。

尽管这种符号表示法可以支持证明和整个形式化开发过程，但我们也可以采用一种循序渐进的方式，结合现有的其他开发方法来使用它。在第三章中，我们将给出用图形符号表示法（实体-关系-属性图以及状态转换图）和这种形式化符号表示法之间的联系，这种联系可以使开发者使用数学形式对无法用图形方法表达的所需系统的复杂特性进行精确描述。在这种情况下，形式化方法可作为一种可选择的方法来使用，它并不是去替换现存的技巧与方法，而是对其进行补充，如在 Fusion 和 Syntropy 面向对象方法中便成功地采用了这种方法，见参考文献[14,16]。

在本书 1.1.2 节中，我们将对 B AMN 开发过程的每一阶段进行详细介绍，但作为一种可选择的方法，它只可以用在我们所需要的那些阶段上，为我们提供帮助指导，获得一个指定层次上的软件完整性和软件质量。

1.1.1　规格说明的作用

为什么一个软件系统的规格说明是非常有用的？其理由很多，目标系统越大、越复杂，这些理由显得越具有实际意义。一个重要的好处就是，形式化规格说明（一种精确且抽象的描述）可以在需求（它是抽象而不精确的）和可执行代码（具体而精确的）之间提供一种中间过程（见图 1.1）。

规格说明是一种软件所需功能和行为的抽象而精确的描述。所以，它可以通过技术手段来证实客户需求的有效性，如动画技术或证明特定预期特性的尝试。描述 D 对于描述 C 具有有效性是指对“D 满足 C 中所指明的特性”这一事实的检查，其中 C 是一种非形式化或半形式化描述。在这种情况下，它是一种检查，即检查“是否为用户想要的指定系统”，而 C 是客户期望的非形式化集合。

通过对一步步细化过程的证明，规格说明也可以用来作为已开发可执行代码的验证基础，描述 D 对于描述 C 具有验证性是指对“D 满足 C 中所指明的特性”这一事实的检查，其中 C 是形式化

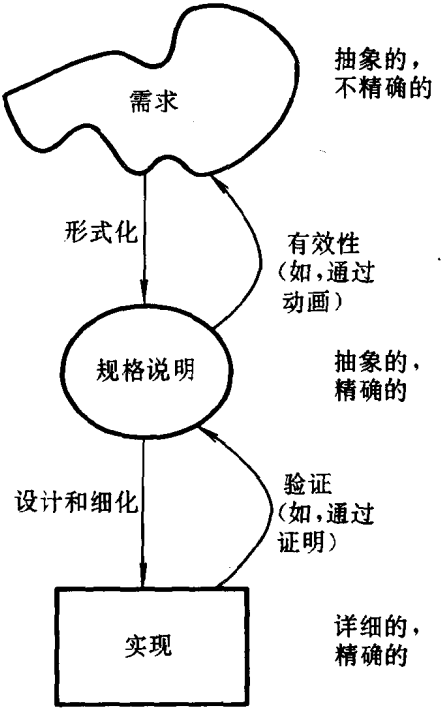

图 1.1 开发过程中规格说明的作用

描述。从原理上讲，这种检查是相对于规格说明的代码功能正确性的根本保证。

因此，规格说明有助于将一个从需求到编码非常困难的开发阶段分解成两个较困难的阶段。规格说明的有效性检查使开发人员在理解需求时所产生的错误（或需求本身的不一致性和不完整性）能在早期阶段，即在付诸编码之前，便被探查出来。因此，它有助于降低编码纠错阶段的成本费用。编码验证可以使开发人员对验证后的结果系统更充满信心。

测试仍可用在正常的开发过程中，以检查编码对于客户需求是否合法、有效。的确，在缺少规格说明的情况下，这只是一种检查形式，通过进行这种检查可标识所开发系统的正确程度。但是，对

于实际系统，测试总是一种不完全的检查，即它只能标识出符合(规格说明)的失误情况，而不能证明在所有情况下编码均完全符合规格说明。

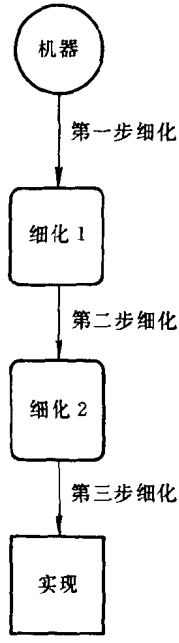

图 1.2　B 语言中规格说明分层次序

形式化方法技术的当前状态表明，无论是当前的安全系统，安全报警系统，还是具有非常大市场价值而其开发成本分散在单位成本中，仅有较少的增长的系统，这些论证只是在有限的情况下对审定通过规格说明具有充分的推动作用。其问题是：

(1)大型系统需要大型的规格说明。因此，需要有较好的技术对规格说明进行分解和模块化，以便对规格说明的各个部分能够

进行相对独立的分析。

(2)验证工作即使对中等复杂程度的模块也是不可行的,或者是高代价的。

B 语言方法试图通过定义一系列方法,将一个大的系统描述分解成一些互相联系的子系统描述,用这样的方法来解决上述第一个问题。这种分解并不意味着可执行系统的设计必须遵循相同的分解方法;它只是一种将一个大型规格说明分成一些较小的、较容易控制的部分的方法。

B 语言方法试图通过使用结构化机制来解决上述第二个问题,结构化机制使得对一个组件(模块)特性的证明能够复用在其他组件特性的证明中,即 B 语言结构化机制的作用实际上是对规格说明的内部一致性有关的证明法则数量的限制。它也允许构造一些中间层的规格说明,以便把一个可能非常困难的验证步骤分解成一些较小的步骤(见图 1.2)。

对于规格说明,B 语言方法的核心元素是分层开发示例,这将在下一节中讨论。

1.1.2 B 语言开发过程

B 语言支持规格说明,并且支持继规格说明之后所有的细化和设计步骤。这一点与第一代形式化方法(尤指 Z 语言)有所不同,第一代形式化方法注重于需求的形式化,但不注重规格说明的可正确执行的实现程序的构造。B 语言称为一种广谱语言或方法,因为它包含了可执行的描述和高度抽象的数学描述。

图 1.3 给出了一个完整的 B 语言开发过程概貌。

这个生存周期的各个阶段包括以下一些内容:

1. 需求分析

问题域和系统需求的非形式化或结构化模型的创建,其结果是一个分析模型的集合(见 3.1 节)。

2. 规格说明开发

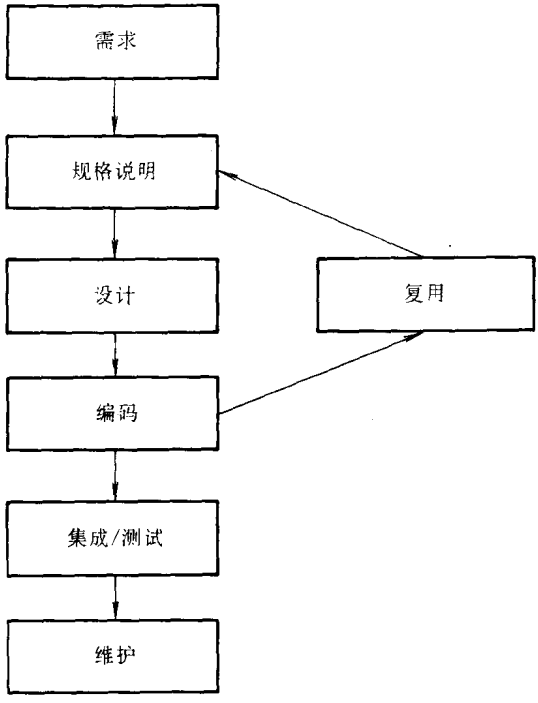

图 1.3　B 语言开发生存周期

(1)在抽象机中对分析模型的诸元素进行形式化，使用这些分析模型，将规格说明分成在概念上有意义的组件(见 3.2 节)。

(2)检查与所选择的需求相对应的规格说明和测试方案的动画(见 3.3 节)。

(3)内部一致性法则的生成，以及内部一致性法则的证明(见 3.4 节)。

此阶段产生一个形式化规格说明。

3. 设计

(1) 系统实现程序分解组件的标识，包括从现存的开发程序

中或规格说明库中取出的可复用组件(见2.4节，4.1节)。

(2)形式化规格说明中所选择的组件的细化步骤的建立(见2.5节,4.2节)。

(3)细化证明法则的产生和证明(见4.3节)。

此阶段产生一个形式化设计。

4. 编码/集成/测试

(1)在最低层的设计中,代码生成器的应用(见4.5节)。

(2)使用基于需求的测试实例对所生成的代码进行测试。

此阶段产生一个可执行的实现系统。

在一个指定开发阶段的描述之后,将注明本书中与该阶段有关的各章节。

如果需要一个快速、严格的开发过程,在此过程中有语法分析、类型检查以及支持在早期的开发阶段发现错误的规格说明动画，则上述阶段的1,2(1),2(2),3(1),3(2),4(1)和4(2)是必需的。实际上,也可以省略3(1)阶段而直接进入到B AMN描述层,该层已非常接近于编码。

如果缺少证明的支持,这种开发形式近似于使用像Z或VDM这样的第一代形式化方法进行开发。但是,正像IBM公司在CICS开发中特别演示的一样[15],这样的开发在改善软件质量方面是非常有效的。

证明可以用于规格说明层中,以增加发现错误的可能性(无论是由于规格说明编写者对需求的误解或对符号表示法的误解所产生的错误,还是由于人的错误或需求本身的错误)。自动证明支持手段在履行绝大多数内部一致性法则中是非常有效的。所以,如果一条法则没有自动证明,那么通过检查来确定它是否真的成立是很值得的。从证明一条法则的失败,通常可以确定错误的根源,正像我们将在3.4节中讨论的那样。

最后,如果需要一个完整的形式化开发,它满足绝大多数像MOD DS 00－55这样的标准需求,参见[52],那么还可以尝试和