

软件的集合定义和共性初探

张 银 明

(电子工程系)

摘要 本文对某些系统软件和应用软件的集合定义和共性作了初步的探讨. 软件共性的研究, 无论从理论上或从实践上讲皆是极有价值的, 因而是一个值得重视的研讨方向.

关键词 系统软件, 应用软件, 共性

0 前言

软件是一个集合. 按列举法可描述为

$$S = \{\text{DOS1.0}, \text{TOS}, \text{SIPROS}, \text{ROS}, \text{NOS}, \text{UNICOMP}, \text{UCO}, \text{FORTRANCOMP}, \text{JC}, \text{LOGMIS}, \text{MACMIS}, \text{CAD}, \text{CAI}, \dots\}.$$

它的元素可划分成两个子集, 一为系统软件 SS(Software System), 一为应用软件 AS(Application System). 可表示为

$$SS = \{\text{DOS1.0}, \text{TOS}, \text{SIPROS}, \text{ROS}, \text{NOS}, \dots\},$$

$$AS = \{\text{LOGMIS}, \text{MACMIS}, \text{CAD}, \text{CAI}, \dots\}.$$

按照软件工程学的定义, 软件是程序以及开发、使用和维护程序所需的所有文档. 这里的文档包括: (1) 面向研制人员的文档. 这是确保软件按质量要求而有效地进行开发的各种文件资料, 如系统说明书、模块说明书等; (2) 面向用户的文档. 诸如用户手册、操作手册以及说明如何使用、维护和修改程序等文件资料. 这是保证软件系统有效和正常运行所必需的. 因而, 若按集合定义的叙述法, 软件集合 S 和子集 SS, AS 可描述为

$$S = \{x | x \text{ 是程序及有关文档}\},$$

$$SS = \{x | x \text{ 是面向硬件的程序系统及其有关文档}\},$$

$$AS = \{x | x \text{ 是面向用户的程序系统及其有关文档}\}.$$

由于集合是由那些具有共同性质的元素汇集而成的一个整体, 所以, 集合中的元素是具有共性的. 居于此, 本文对其中某些集合的共性作一些归纳和探讨.

* 本文1991-02-25收到.

1 软件的共性

软件和硬件一样是组成计算机系统的不可缺少的部分. 随着计算机技术的迅猛发展和广泛应用, 软件集合的元素急骤增多, 极为丰富, 但它们皆以自身的特性来表明其相应的归属. 我们可从以下几个方面来讨论它们的共性.

1.1 软件的集合属性

软件既是集合, 当然具有一般集合的属性. 但为进一步解释软件集合的特性, 故对其其中的一些属性稍加说明.

(1) 软件集合 S 是个全集. 在计算机软件的范围中, 任何软件元素组成的集合皆是 S 的子集. 如上述的 SS 和 AS 是 S 的子集. 假设 $SN_{ij} (i=1, 2, \dots, n; j=1, 2, \dots, m)$ 是软件元素, 则 $SN_{ij} \in S$. 因而由 SN_{ij} 组成的集合

$$S_i = \{SN_{i1}, SN_{i2}, \dots, SN_{im}\}$$

显然是 S 的子集, 亦即 $S_i \subseteq S$. 根据定义, S 是全集.

(2) 集合 $S' = \{SS, AS\}$ 是 S 的一个覆盖. 显然 $AS \subseteq S$ 及 $SS \subseteq S$, 且 $SS \neq \emptyset, AS \neq \emptyset$, 又有 $SS \cup AS = S$, 故 S' 是 S 的一个覆盖.

倘若在划分 SS 和 AS 所含元素时, 约定任一软件元素仅属两者之一, 那么 S' 也是 S 的划分.

(3) 集合 S 是有限集, 因而是至多可数集. 虽然目前各种软件元素极为繁多, 但终究是有限的, 故其组成的集合是有限集.

(4) 作为集合, S 同其他集合一样具有自反性和传递性等属性. 这里就不再进行更多的讨论.

1.2 软件生命期

软件共性的一个重要方面就是任何软件都有一个生命期. 其主要阶段为: (1) 需求分析: 解对软件的要求; (2) 软件规划: 确定工作域、预测耗费资源及成本估计等; (3) 需求分析定义: 解决做什么的问题, 则对工作域进行求精和细化, 分析各种可行解, 构造系统的数据流、数据结构、配置软件元素; (4) 软件设计: 解决如何做的问题, 根据需求分析定义, 选择设计方法, 确定软件结构, 把结构元素变换成过程描述; (5) 编码阶段: 选取合适的语言, 对软件结构的描述进行编码; (6) 验证和测试: 检验所研制的软件是否合乎要求, 验证文档资料是否完善, 从发

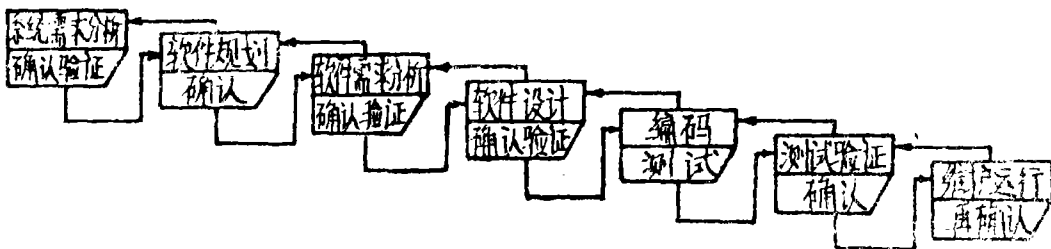


图1 软件生命周期管理模式

测试程序是否有错及能否满足预定功能;(7)运行维护:投入实际使用,并在运行期间继续进行查错纠错和必要的优化完善等维护工作;(8)停止使用或废旧更新.至此,该软件已完成其历史使命.

生命周期是软件的重要概念,它揭示了软件研制怎样归纳求解空间且在其间求得较为理想的可行解.从而能更有效、更科学地组织和管理软件生产,以便获得最经济最可靠的软件产品.这也使软件的研制和生产过程规范化. B. W. Boehm 为保证软件可靠性而提出了连续验证生命周期的模式,如图1所示.它也是一种有效的软件生命周期的管理模式.

1.3 软件质量的规范和标准

在软件研制中,为保证软件产品具有好的功效和技术经济指标,都有一组约束条件作为检验的依据,也就是衡量软件的质量指标,主要是可靠性、可维护性、正确性、可移植性、运行效率、程序风格、健壮性以及文档资料的完整性、正确性、易理解性等.

开发过程应按生命周期的管理模式进行管理,使之达到:(1)开发过程的作业标准化;(2)每个作业的表达形式规范化;(3)文档资料的格式标准化;(4)系统使用的符号规范化;(5)规范化的开发原则和制度.

1.4 软件开发方法

软件开发过程虽采用各种方法,但都统一于软件工程学的开发方法,它们是系统分析方法、设计方法、编程方法、调试方法、定义说明方法、结构化的构造方法及文档资料的描述方法.从软件开发生的管理方面,还有软件质量的定量标准与质量管理方法、组织方法、研制时间和进度的计算方法等.这些方法的目标在于科学地使用各种资源、有效地组织力量,以确保开发的高质量、低消耗.

总之,软件有其明显的共同特征,正是这些共性使丰富多采的各种软件归属于同一范畴.综合起来,软件是人们同计算机取得联系和合作的桥梁.人们的任何需求都是通过相应的软件来发挥计算机的功效.实质上,运行中的软件可看成是一个函数,亦即定义为

$$O = S(I)$$

其中的自变量 I 是人们的输入信息,而因变量便是输出信息,函数 S 则是软件.所以,软件是问题求解过程的描述和实现.

我们知道,可计算性的定义是:凡可用某种程序语言来描述的东西就是可计算的.因而,软件所描述的对象都是可计算的.明确这一点,也许有助于对软件特性的认识.

2 系统软件的共性

系统软件是软件的一个子集,因而它具有软件所具有的特性.此外,又有它本身的一些特性,这是它区别于应用软件的标志.

首先,系统软件是计算机系统不可缺少的重要组成部分.没有它计算机便不能发挥作用.它把计算机、人和应用软件三者有机地联系成为一个整体,起着协调、指挥的桥梁作用.一方面它把用户或应用软件所表述的愿望转换成机器所能理解的语言和相应的动作;另一方面又把执行的结果或用户所期望的信息转告给用户.其次,系统软件是计算机应用开发的支撑环境.它既面向硬件,使硬件资源得到最充分的发挥;又面向用户,让用户能方便地掌握和使用.显

然,它是沟通人机通讯的重要工具.

这里对系统软件的共性不准备作更多的讨论,只对两个最重要的子集——操作系统 OS 和编译系统 CS(Compiler system)的共性作扼要地概括.

2.1 OS 的功能共性和集合定义

操作系统的最主要功能是进行资源管理,其共同的功能有:(1)进行处理机管理,包括调度管理和进程调度管理;(2)进行存贮管理.主要含有:①为进入运行状态的作业动态地分配内存空间;②存贮保护;③扩充存贮空间;(3)进行 I/O 管理;(4)进行文件管理.操作系统把程序及数据文件皆视为文件,并当作重要资料进行必要的调度和编辑管理.

这样,我们可以把 OS 这个集合描述成

$$OS = \{x | x \text{ 是软件且能对 CPU、存贮、I/O 和文件进行管理} \}$$

也就是说,操作系统 OS 是由那些能对 CPU、存贮、I/O 和文件进行管理的软件元素组成的集合.

2.2 编译系统的功能共性和集合定义

编译系统的任务是把源程序翻译成目标程序 或者说把源语言程序改造成另一种使计算机能够直接理解和运行的目标语言程序,而两者在逻辑上是等价的.当然,从源程序到目标程序的整个变换过程是极其复杂的.但归结起来,主要具有五个方面的功能:(1)能对源程序进行词法分析,从而识别出单词;(2)具有语法分析能力,把单词符号串分解成各类语法单位;(3)生成中间代码,则对各类不同语法范畴按语言的语义进行初步翻译工作;(4)对中间代码进行优化.诸如公共子表达式的提取、循环优化、算符归约等,以便能在最后阶段产生出更为高效的目标代码;(5)生成目标代码,亦即将优化后的中间代码转换成指定机器上的绝对指令代码或可重新定位的指令代码或汇编指令代码.

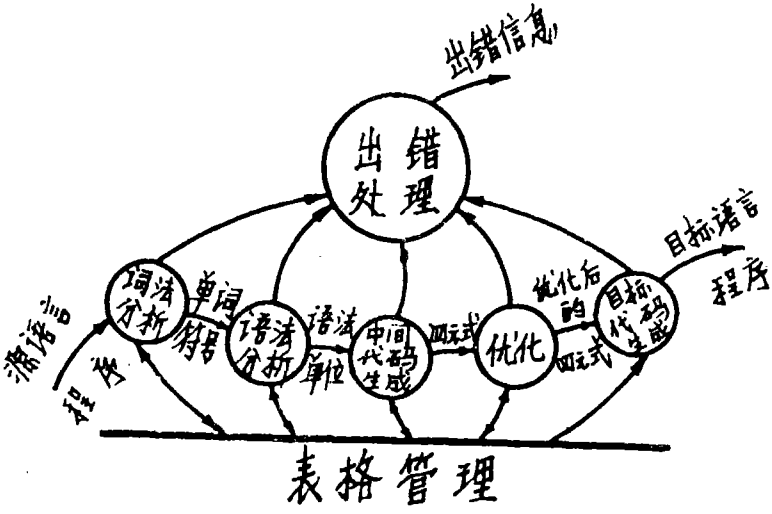


图2 编译系统的功能模式

虽然并非所有编译程序都具有这些能力,但作为一个较完善的编译则应具备这些主要的功能.据此,我们可以把编译系统 CS 定义成如下的集合

$CS = \{x | x \text{ 是程序且能把源语言程序转换成目标语言程序}\}$

更为详细的描述可以是

$CS = \{x | x \text{ 是程序且能对源语言程序进行词法分析、语法分析、产生中间代码、优化及生成目标代码}\}$.

由于通常的编译程序还含有出错处理与表格管理的功能,因而其功能模式可表示成图2的形式.这样,编译系统的集合定义可更详细地叙述成为

$CS = \{x | x \text{ 是程序且具有对源语言程序进行词法分析、语法分析、产生中间代码、优化及生成目标代码、出错处理及表格管理的能力}\}$.

3 高级程序设计语言

程序设计语言既非硬件又不是程序,因而较特殊,但由于它是非硬件的、专门为程序设计服务的工具,人们不加探究地将它归属于软件的集合,多数人确认为它是系统软件.

高级语言由于接近于数学和工程语言、直观、自然和易于理解,故而所编制的程序易读、易写,易于交流、存档和出版,且便于编辑和验证其正确性.它具有交流算法和计算实现的双重目的,从而具有易于表达算法实现及便于计算机实现的特征.一般而言,高级程序设计语言应具有如下六方面主要内容.

(1)字母表= $\{\text{有限字符集}\}$;

(2)规定单词符号形成的词法规则.例如

单词= $\{\text{数据,标识符,基本字,算符,界符}\}$

数据= $\{\text{数值数据,逻辑数据,字符数据,指示器}\}$

= $\{\text{〔整数,实数,复数,双精度数,算术运算〕,〔逻辑真,逻辑假,逻辑运算〕,〔字符型数据,字符运算〕,指示器数据}\}$

标识符= $\{\text{由字母或数据组成的且以字母为首字符的字符串}\}$.

(3)规定由单词符号到语法单位形成的语言规则.如

语法单位= $\{\text{表达式,语句,分程序,函数,过程,程序}\}$.

表达式= $\{\text{运算量,算符}\}$

语句= $\{\text{赋值语句,控制语句,说明语句}\}$.

(4)定义单词符号和语法单位的语义规则;

(5)定义和调用过程,以及参数传递方式;

(6)存贮空间的分配和管理方式.

这样,高级程序语言从所含的内容来定义相应集合的话,便可描述为:

$HPL = \{\text{字母表,词法规则,语法规则,语义规则,辅助定义,存贮管理}\}$.

当然这些规则和定义可构成有意义的单词、短语、语句,且用于编制程序.

如果从功能上分析,通常用于科技计算和数据处理的语言可归结出以下的功能:(1)定义功能:标识符、变量、函数、数组、子程序、过程;(2)赋值和运算功能:赋值、算术运算、逻辑运算、字符串运算以及更进一步的向量运算和矩阵运算等;(3)控制功能:条件判别、转移、循环控制、递归控制、嵌套控制、暂停、停止和结束控制;(4)文件处理功能:顺序文件、随机或排序文件、索

引文件的建立,数据结构或记录格式的定义,记录及字段的存取、更新处理;(5)I/O 处理功能:信息的接受、显示和打印的功能;(6)内部函数:常用的数学函数、字符函数、图形函数及辅助函数(如日期函数);(7)参数传递、存贮分配和管理功能;(8)说明和注释功能:优先说明、等同说明、运算说明等,以及可进行必要的注释。

因而,若从功能方面来定义它的集合,则可描述如下

$HPL = \{x | x \text{ 是语言且具有定义、运算、控制、文件和 I/O 处理、提供函数、参数似递、存贮管理、说明和注释等功能}\}.$

4 MIS 的共性及其集合定义

应用软件丰富多彩、种类繁多,其中 MIS 是最为普及、应用范围广、数量多的应用软件,因而具有代表性.根据本人近几年来负责研制的各种 MIS 和参阅的资料,按照软件工程的功能划分法,可把共同的功能简单归结为以下五点.

(1)输入功能:接受信息记录的输入和编辑,并把确认后的信息记录存入数据库文件.

(2)维护功能:对已存入数据库文件的信息记录进行记录查找、显示、修改、插入和删除,并在维护后进行必要的处理.

(3)查询功能:有时亦称检索功能,亦即能接受查询条件的输入,并按该条件进行必要的优化,尔后进行搜索,给出相应提示或结果.

(4)统计处理功能:对数据信息的汇总、统计、计算、必要的分析和处理.

(5)输出功能:显示或打印各种报表、图形及综合性的数据信息.

据此,MIS 的集合可定义为

$MIS = \{x | x \text{ 是应用软件且具有信息输入、维护、查询、统计处理和输出等功能}\}.$

研究应用软件的共性在于分析同一类应用软件普遍存在于它们之中的共同规律性,透过各自表现出来的特殊性——个性,而抓住其最本质的内在联系——共性,进而归纳出实现这些共性的通用模式.本对这种模式进行算法设计,以指导研制一套实现它的自动生成系统,为该共性复盖下的一类应用软件的研制提供较为完善的工具.因而,这种研究既是对应用软件研制的经验总结和理论上的探讨,又为应用软件的自动生成提供指导和奠定基础.

5 关于共性研究的建议

本文仅对少数软件的共性及其集合的定义作了初浅的探讨.由于共性的讨论,意在揭示一类软件的本质联系,找出共同规律,归结通用模式,进行算法设计,以指导生成工具的研制.因而,既有理论上又有实践价值的值得重视的研究方向.

要很好地进行共性研究,笔者认为:首先应对软件进行合乎科学的归类,以明确划分各软件集合所含的元素.其次,各集合元素的共性宜从功能上进行归纳.这是因为功能既是一个系统的最主要指标和特征,又是用户要求的最关键指标,而且符合软件工程学的系统设计原则.所以,从功能方面研究归纳软件的共性无论从科学性或从实践性上分析都是较为合适的.

参 考 文 献

- [1] Lan Campbell, *Information and Software technolog*, May/Sept. ,30, (1988).
[2] 孙振飞、应振树, 软件工程概论, 湖南科学技术出版社, (1987).
[3] 张银明, 输入模块的设计和实现, 化工电子计算, 3(1989), 1—7.
[4] 张银明, 维护模块的通用模式、设计和实现, 化工电子计算, 3(1990).

Preliminary Approach to the Set Definition and the
General Character of Software

Zhang Yinming

(*Department of Electronic Engineering*)

Abstract The author approaches preliminarily to the general characater of some system software and application software as well as their definition based on the set. The general character of software as a topic of study is extremely valuable in the eye of theory and practice. It is thus an orientation of deliberation merits attention.

Key words system software, application software, general character