

第 5 章



程序、软件与系统

5.1 程序、语言与软件

5.1.1 程序

程序(包括计算机程序和软件程序)是指一组指示计算机执行动作或做出判断的指令,通常用某种程序设计语言编写,运行于某种目标计算机或智能仪器上。

1. 程序的基本结构

早在 1966 年,Bohm 和 Jacopin 就证明了程序设计语言中只要有 3 种基本控制结构,就可以表示出各种复杂的程序结构。这 3 种基本控制结构是顺序、选择和循环。对于具体的程序语句来说,每种基本控制结构都包含若干语句。

(1) 顺序结构。顺序结构表示程序中的各操作是按照它们出现的先后顺序执行的,如图 5-1(1)所示,先执行 A 模块,再执行 B 模块。

(2) 选择结构。选择结构表示程序的处理步骤出现了分支,它需要根据某一特

定的条件选择执行其中一个分支。选择结构有单选择、双选择和多选择 3 种形式。如图 5-2(2)所示,当条件 P 的值为真时执行 A 模块,否则执行 B 模块。

(3) 循环结构。循环结构表示程序反复执行某个或某些操作,直到某个条件为假(或为真)时才可终止循环。在循环结构中最重要的是什么情况下执行循环以及哪些操作需要执行循环。如图 5-1(3)所示为当型循环结构:当条件 P 的值为真时,就执行 A 模块,然后再次判断条件 P 的值是否为真,直到条件 P 的值为假时才向下执行。如图 5-1(4)所示为直到型循环结构:先执行 A 模块,然后判断条件 P 的值是否为真,若 P 为假,再次执行 A 模块,直到条件 P 的值为真时才向下执行。

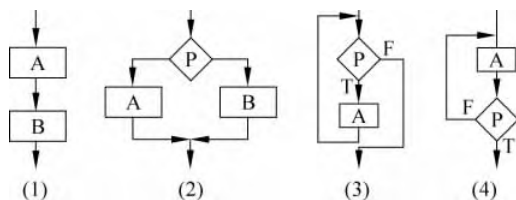


图 5-1 程序的 3 种基本结构

2. 程序的执行方式

程序一般是用高级语言编写的,如 C/C++ 和面向对象的 Visual 系列。高级语言程序在计算机上是不能直接执行的,因为计算机只能执行二进制程序。因此,要将编写的程序翻译成二进制程序。在计算机上执行程序的方式有两种:一是解释执行方式,二是编译执行方式。

(1) 解释执行方式。即翻译一句,执行一句,也就是边解释边执行。这种方式每次运行程序时都要重新翻译整个程序,效率较低,执行速度慢,如 BASIC 语言程序。解释执行方式按照源程序中语句的动态顺序逐句进行分析解释,并立即执行,直至源程序结束。

(2) 编译执行方式。即在程序第一次执行前,先将编写的程序翻译成二进制程序,然后直接执行这个翻译好的二进制程序。程序的翻译过程叫编译。现在的大多数语言都是采用这种方式。编译执行方式把源程序的执行过程严格地分成编译和运行两大步,即先把源程序全部翻译成目标代码,再运行此目标代码,最后获得执行结果。

5.1.2 计算机语言

计算机语言的发展是一个不断演化的过程,其推动力是对抽象机制更高的要求及对程序设计思想更好的支持。计算机语言从最开始的机器语言,到汇编语言,再到各种结构化高级语言,最后演化到支持面向对象技术的面向对象语言。

1. 机器语言

二进制是计算机语言的基础。计算机发明之初,人们编写一串串由 0 和 1 组成的指令序列交由计算机执行,这种语言就是机器语言。使用机器语言是十分痛苦的,特别是在程序有错需要修改时比较麻烦。由于每台计算机的指令系统不同,所以,在一台计算机上执行的程序,要想在另一台计算机上执行,必须重新编程,重复工作量大。但是,由于使用的是针对特定型号计算机的语言,故而运算效率是所有语言中最高的。机器语言是第一代计算机语言。

2. 汇编语言

为了降低使用机器语言编程的难度,人们对机器语言进行了改进:用一些简洁的英文字母、符号串来替代特定的二进制指令串,例如用 ADD 代表加法,用 MOV 代表数据传递等。这样,人们便可以很容易读懂并理解程序要干什么,纠错及维护也变得方便很多,这种程序设计语言就是汇编语言,即第二代计算机语言。然而计算机不认识这些符号,这就需要一个专门的程序负责将这些符号翻译成二进制的机器语言,这种翻译程序称为汇编程序。汇编语言同样十分依赖于机器硬件,移植性不好,但效率很高。针对计算机特定硬件而编制的汇编语言程序,能准确发挥计算机硬件的功能和特长,程序精炼而质量高,所以汇编语言至今仍是一种常用而强有力的软件开发工具。

3. 高级语言

从最初与计算机交流的“痛苦经历”中,人们意识到应该设计一种这样的语言:它接近于数学语言或人类的自然语言,同时又不依赖于计算机硬件,编出的程序能在所有机器上通用,这种就是高级语言。经过努力,1954 年第一个完全脱离机器硬件

的高级语言——FORTRAN 问世。60 多年来,已出现了几百种高级语言,其中,有几十种语言影响较大,应用普遍性好,如 FORTRAN、ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C、PROLOG、Ada 及 C++、VC、VB、Delphi、JAVA 等。高级语言的发展经历了从早期语言到结构化程序设计语言,从面向过程到非过程化程序语言的过程,相应地,软件的开发也由最初的个体手工作坊式的封闭式生产方式发展为产业化、流水线式的工业化生产方式。

4. 第四代语言

第四代语言(Fourth-Generation Language, 4GL)具有简单易学、用户界面良好、非过程化程度高、面向问题、只须告知计算机“做什么”而不必告知计算机“怎么做”等特点。用 4GL 编程的代码量较传统语言明显减少,并可提高软件生产率。这类语言有 ADA、MODULA-2、SMALLTALK-80 等。第四代语言的出现属于商业驱动,4GL 一词最早出现在 20 世纪 80 年代初的软件厂商广告和产品介绍中。1985 年美国召开全国性的 4GL 研讨会,在这之后,许多计算机专家对 4GL 展开了全面研究。20 世纪 90 年代,基于数据库管理系统的 4GL 商品化软件在软件开发中获得应用,使得 4GL 成为面向数据库应用的主流工具。

5. 计算机语言的未来发展趋势

面向对象程序设计和数据抽象在现代程序设计思想中占有很重要的地位,未来语言的发展将不再是一种单纯的语言标准,而是采用完全面向对象的形式,更易于表达现实世界,更易于为人编写。语言使用人员将不再只是专业的编程人员,人们完全可以用订制真实生活中一项工作流程的简单方式来完成编程。计算机语言发展的特性:①简单性:提供最基本的方法来完成指定的任务,用户只须理解一些基本的概念,就可以用它编写出适合于各种情况的应用程序;②面向对象:提供简单的类机制及动态的接口模型,对象中封装状态变量及相应的方法,实现模块化和信息隐藏;提供一类对象的原型,并且通过继承机制,子类可以使用父类所提供的方法,实现代码的复用;③安全性:用于网络、分布环境下时有安全机制保证;④平台无关性:与平台无关的特性使程序可以方便地被移植到网络上的不同机器和不同平台。

6. R 语言

R 语言是由数据操作、计算和图形展示功能整合而成的套件,包括统计分析、绘图语言和操作环境。R 是属于 GNU 系统的一个自由、免费、源代码开放的软件,其功能包括数据存储和处理、数组计算操作、数据分析工具、数据显示的图形功能及可以编程(包括条件、循环、自定义函数、输入/输出功能)。R 语言诞生于 1980 年,属于 AT&T 贝尔实验室开发的一种用来进行数据探索、统计分析和作图的解释型 S 语言系列,后由 MathSoft 公司完善。随后,新西兰奥克兰大学的 Robert Gentleman 和 Ross Ihaka 及其他志愿人员开发了 R 系统。

7. Python 语言

Python 是一种面向对象的解释型计算机程序设计语言,由荷兰人 Guido van Rossum 于 1989 年开发,1991 年公开。Python 是自由软件,源代码和解释器 Python 遵循 GPL(GNU General Public License)协议。Python 有丰富、强大的库,被喻为胶水语言,因为它能把用其他语言制作的各種模块很轻松地连接在一起,常见的一种应用是使用 Python 快速生成程序的原型(有时甚至是程序的最终界面),然后对其中有特别要求的部分用更合适的语言改写。2017 年编程语言排行榜中 Python 高居首位。

5.1.3 软件

1. 软件

软件(Software)是一系列按照特定顺序组织的计算机数据和指令的集合。一般来讲,软件被划分为编程语言、系统软件、应用软件以及介于系统软件和应用软件之间的中间件。软件并不只是包括可以在计算机(这里的计算机是指广义的计算机)上运行的程序,与这些程序相关的文档一般也被认为是软件的一部分。简单来说,软件就是程序加文档的集合体。

软件是用户与硬件之间的接口界面,用户主要是通过软件与计算机进行交流。软件是计算机系统设计的重要依据,为了方便用户,为了使计算机系统具有较高的总体效用,在设计计算机系统时,必须全局考虑软件与硬件的结合以及用户的要求和软件的要求。

2. 软件分类

一般来讲,软件划分为系统软件和应用软件,其中系统软件包括操作系统和支撑软件(包括微软发布的嵌入式系统,即硬件级的软件,可使计算机及其他设备的运算速度更快,更节能),如图 5-2 所示。

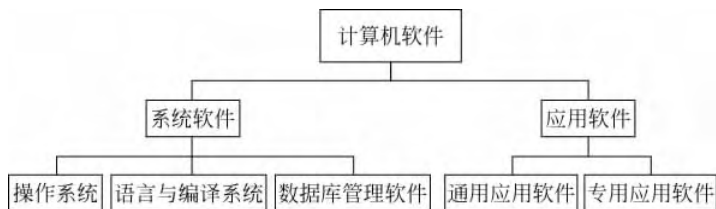


图 5-2 软件分类

1) 系统软件

系统软件可为计算机提供最基本功能,分为操作系统、数据库管理系统、系统实用程序等。

(1) 操作系统是一种管理计算机硬件与软件资源的程序,同时也是计算机系统的内核与基石,负责内存管理与配置、系统资源供需的优先次序、输入与输出设备、操作网络与管理文件系统等事务。操作系统分为 Windows、Unix、Linux、Mac OS 及 OS/2 等。

(2) 数据库管理系统是对数据库进行有效管理和操作的系统,是用户与数据库之间的接口,它提供了用户管理数据库的一套命令,可完成数据库的建立、修改、检索、统计和排序等功能。关系型数据库管理系统应用广泛,常见的有 FoxPro、SQL Server、Oracle、Sybase、DB2 和 Informix 等。

(3) 系统实用程序是一些工具性的服务程序,便于用户使用和维护计算机。主要的系统实用程序有语言处理程序、编辑程序、连接装配程序、打印管理程序、测试程序和诊断程序等。

2) 应用软件

(1) 通用应用软件,是某些具有通用信息处理功能的商品化软件。它的特点是具有通用性,可以被许多有类似应用需求的用户使用,它所提供的功能往往可以由用户通过选择、设置和调配来满足需求。比较典型的通用软件有文字处理软件、表格处理软件等。

(2) 专用应用软件,是满足用户特定要求的应用软件。某些情况下,用户对数据处理的功能需求存在很大的差异性,通用软件不能满足要求时,需要由专业人士采取单独开发的方法,开发满足特定要求的专门应用软件,例如数值统计分析软件、财务核算软件等。

3. Matlab 软件

Matlab 是 MathWorks 公司出品的商业数学软件,是用于算法开发、数据可视化、数据分析及数值计算的高级技术计算语言和交互式环境,包括 Matlab 和 Simulink 两大部分。Matlab 可以运算矩阵、绘制函数和数据、实现算法、创建用户界面、连接其他编程语言的程序等,主要应用于工程计算、控制设计、信号处理与通信、图像处理、信号检测以及金融建模设计与分析等领域。

5.2 操作系统

5.2.1 操作系统概述

1. 定义

操作系统(Operating System, OS)是一种管理计算机硬件与软件资源的程序。负责管理计算机系统的硬件资源、软件资源和数据资源,控制程序运行,改善人机界面,为其他应用软件提供支持等,使计算机系统的所有资源最大限度地发挥作用,为用户提供方便、有效、友善的服务。

操作系统是最靠近硬件的系统软件,它把硬件裸机改造成为一台功能完善的虚拟机,使得计算机系统的使用和管理更加方便,计算机资源的利用效率更高,上层的

应用程序可以获得比硬件提供的功能更多的支持。

2. 功能

操作系统是一个庞大的管理控制程序,主要包括处理器管理、设备管理、文件管理、存储管理、作业管理 5 个方面的管理功能。操作系统主要对硬件资源和软件资源进行分配和调度。

- (1) 处理器管理是根据一定的策略将处理器交替地分配给系统内等待运行的程序。
- (2) 设备管理负责分配和回收外部设备,控制外部设备按用户程序的要求进行操作。
- (3) 文件管理向用户提供创建文件、撤销文件、读写文件、打开和关闭文件等功能。
- (4) 存储管理的功能是管理内存资源,主要实现内存的分配与回收、存储保护及内存扩充。
- (5) 作业管理的功能是为用户提供一个使用系统的良好环境,使用户能有效地组织自己的工作流,并使整个系统高效运行。

3. 操作系统的发展历史

(1) 20 世纪 80 年代前。第一部计算机并没有操作系统,这是由于早期计算机的建立方式(如同建造机械算盘)与效能不足以执行如此程序。1947 年,晶体管被发明,莫里斯·威尔克斯(Maurice V. Wilkes)发明了微程序方法。20 世纪 60 年代早期,商用计算机制造商制造了批次处理系统,此系统可将工作的建置、调度及执行序列化。厂商为每一台不同型号的计算机设计了不同的操作系统,因此为某计算机写的程序无法移植到其他计算机上执行,即使同型号的计算机之间也不行。1964 年,IBM System/360 推出了一系列大型计算机,OS/360 是适用于整个系列产品的操作系统。1963 年,奇异公司与贝尔实验室合作,以 PL/I 语言建立 Multics,为 Unix 系统奠定了良好的基础。

(2) 20 世纪 80 年代。早期最著名的磁盘启动型操作系统是 CP/M。1980 年,微软公司与 IBM 签约,并且收购了一家公司出产的操作系统,修改后改名为 MS-DOS。在解决了兼容性问题后,MS-DOS 变成了 IBM PC 上最常用的操作系统。同年,另一个操作系统 Mac OS 出现,即苹果计算机的 Mac OS,其采用图形用户界面,用户可以用鼠标下拉式菜单、桌面图标、拖曳式操作与双单击等实现操作。

(3) 20 世纪 90 年代。这个时期出现了许多影响个人计算机市场的操作系统。

由于图形化用户界面日趋繁复,操作系统的能力也越来越复杂、巨大,因此强韧且具有弹性的操作系统就成了迫切的需求。1990年,开源操作系统 Linux 问世。Linux 内核是一个标准 POSIX 内核,其血缘可算是 Unix 家族的一支。Linux 与 BSD 家族都搭配 GNU 计划所发展的应用程序,由于使用许可证及历史因素, Linux 取得了相当可观的开源操作系统市占率。苹果公司于 1997 年推出的新操作系统 Mac OS X 取得了巨大的成功。

(4) 21 世纪初。大型主机与嵌入式系统可使用的操作系统日趋多样化。有许多大型主机近期开始支持 Java 及 Linux,以便共享其他平台的资源。嵌入式系统近期百家争鸣,从给 Sensor Networks 用的 Berkeley Tiny OS 到可以操作 Microsoft Office 的 Windows CE。

5.2.2 不同的操作系统

1. 操作系统分类

操作系统的分类如下。

(1) 批处理操作系统(Batch Processing Operating System)将一批作业提交给操作系统后就不再干预,由操作系统控制它们自动运行。批处理操作系统分为单道批处理操作系统和多道批处理操作系统。

(2) 分时操作系统(Time Sharing Operating System)是使一台计算机采用时间片轮转的方式同时为几个、几十个甚至几百个用户服务的一种操作系统。

(3) 实时操作系统(Real Time Operating System)是指使计算机能及时响应外部事件的请求,并在规定的时间内严格完成对该事件的处理,并控制所有实时设备和实时任务协调一致地工作的操作系统。

(4) 网络操作系统(Network Operating System)是基于计算机网络的,在各种计算机操作系统上按网络体系结构协议标准开发的软件,包括网络管理、通信、安全、资源共享和各种网络应用,其目标是相互通信及资源共享。

(5) 分布式操作系统(Distributed Software Systems)是支持分布式处理的软件系统,是在由通信网络互联的多处理机体系结构上执行任务的系统,包括分布式操作系统、分布式程序设计语言及其编译(解释)系统、分布式文件系统和分布式数据库系统等。

2. 典型的操作系统

(1) Windows。它是一款微软公司开发的窗口化操作系统,采用了 GUI 图形化操作模式,比从前的指令操作系统(如 DOS)更为人性化。Windows 操作系统是目前世界上使用最广泛的操作系统。

(2) UNIX。它是一款强大的多用户、多任务操作系统,支持多种处理器架构,属于分时操作系统,1969 年由 AT&T 贝尔实验室开发。

(3) Linux。它是 UNIX 操作系统的一种克隆系统,1991 年 10 月 5 日正式向外公布。Linux 是一款免费的操作系统,用户可以通过网络或其他途径免费获得,并可以任意修改其源代码,这是其他操作系统所做不到的。正是由于这一点,程序员可以根据自己的兴趣和灵感对其进行改变,来自全世界的无数程序员参与了 Linux 的修改、编写工作,这让 Linux 吸收了无数程序员的精华而不断壮大。

(4) Mac OS。它是苹果公司为 Mac 系列产品开发的专属操作系统,1985 年由史蒂夫·乔布斯(Steve Jobs)组织开发,是一款图形界面的操作系统。

(5) Android。它是一种基于 Linux 的自由及开放源代码的操作系统,主要使用于移动设备,如智能手机和平板电脑等,由 Google 公司和开放手机联盟领导及开发,中文名称为“安卓”。Android 操作系统最初由 Andy Rubin 开发,主要支持手机,2005 年 8 月由 Google 收购注资。

(6) iOS。它是由苹果公司开发的移动操作系统,2007 年 1 月 9 日发布,最初是设计给 iPhone 使用的,后来陆续套用到 iPod touch、iPad 及 Apple TV 等产品上。iOS 与苹果的 Mac OS X 操作系统一样,属于类 Unix 的商业操作系统。

(7) 银河麒麟(Kylin)。它是国防科技大学研制的开源服务器操作系统,是 863 计划重大攻关科研项目,打破了国外操作系统的垄断,是一套中国自主知识产权的服务器操作系统。银河麒麟 2.0 包括实时版、安全版和服务版。

(8) YunOS。它是阿里巴巴集团旗下的智能操作系统,融合了阿里巴巴在云数据存储、云计算服务及智能设备操作系统等多领域的技术成果,可搭载于智能手机、智能穿戴、互联网汽车、智能家居等多种智能终端设备。根据统计,截至 2016 年 7 月,搭载 YunOS 的物联网终端数量已经突破 1 亿。

5.3 软件工程

5.3.1 软件工程概述

1. 基本概念

软件工程(Software Engineering)是一门研究用工程方法构建和维护有效的、实用的和高质量的软件的学科。它涉及程序设计语言、数据库、软件开发工具、系统平台、标准及设计模式等方面。软件工程是应用计算机科学、数学、工程科学及管理科学等原理开发软件的工程,借鉴传统工程的原则和方法提高质量,降低成本。其中,计算机科学、数学用于构建模型与算法,工程科学用于制定规范,设计范型,评估成本及确定权衡,管理科学用于计划、资源、质量、成本等管理。

2. 软件工程过程

软件工程过程可概括为基本过程类、支持过程类和组织过程类 3 类。

(1) 基本过程类,包括获取过程、供应过程、开发过程、运作过程、维护过程和管理过程。

(2) 支持过程类,包括文档过程、配置管理过程、质量保证过程、验证过程、确认过程、联合评审过程、审计过程及问题解决过程。

(3) 组织过程类,包括基础设施过程、改进过程及培训过程。

3. 软件生命周期

同其他任何事物一样,一个软件产品或软件系统也要经历孕育、诞生、成长、成熟、衰亡等阶段,一般称为软件生命周期(Software Development Life Cycle, SDLC),它是软件产生直到报废的生命周期,周期内有问题定义、可行性分析、需求分析、系统设计、编码、调试、测试、验收、运行、维护升级到废弃等阶段,这种按时间分层的思想方法是软件工程中的一种思想原则,即按部就班,逐步推进,每个阶段都要有定义、工作、审查和形成文档以供交流和备查,从而提高软件的质量。

5.3.2 软件开发方法

1. 结构化方法

结构化方法(Structured Method)是一种传统的软件开发方法,它由结构化分析、结构化设计和结构化程序设计 3 部分有机组合而成。其思想是把一个复杂问题的求解过程分阶段进行,而且这种分解是自顶向下,逐层分解,每个阶段处理的问题都控制在人们容易理解和处理的范围内。结构化方法的基本要点是自顶向下,逐步求精和模块化设计。结构化方法按软件生命周期划分,有结构化分析(SA)、结构化设计(SD)、结构化实现(SP)3 个阶段。

2. 面向对象方法

面向对象方法(Object-Oriented Method)是一种把面向对象的思想应用于软件开发过程中指导开发活动的系统方法,简称 OO 方法,是建立在“对象”概念基础上的方法学。所谓面向对象,就是基于对象概念,以对象为中心,以类和继承为构造机制,来认识、理解、描述客观世界和设计、构建相应的软件系统。对象是由数据和允许的操作组成的封装体,与客观实体有直接对应关系,一个对象类定义具有相似性质的一组对象。继承性是对具有层次关系的类的属性和操作进行共享的一种方式。

3. 软件复用与构件

软件复用(Software Reuse)就是将已有的软件成分用于构造新的软件系统,以缩减软件开发和维护的花费。无论对可复用构件原封不动地使用,还是进行适当的修改后再使用,只要是用来构造新软件,都可称作复用。

构件(Component)是面向软件体系架构的可复用软件模块,它可以是被封装的对象类、类树、功能模块、软件框架(Framework)、软件构架(或体系结构,Architectural)、文档、分析件及设计模式等。开发人员可以通过组装已有的构件来开发新的应用系统,从而达到软件复用的目的。

4. 软件产品线

软件产品线是一组具有共同体系构架和可复用组件的软件系统,它们共同构建

支持特定领域产品开发的软件平台。一个软件产品线由一个产品线体系结构、一个可复用构件集合和一个源自共享资源的产品集合组成,是组织开发一组相关软件产品的方式。软件产品线的产品是根据用户的基本需求对产品线架构进行定制,将可复用部分和系统独特部分集成而得到的,方法集中体现一种大规模、大粒度软件复用实践,是软件工程领域中软件体系结构和软件重用技术发展的结果。

软件产品线的思路是将软件的生产过程分到3类不同的生产车间进行,即应用体系结构提取车间、构件生产车间和基于构件、体系结构复用的应用集成(组装)车间,从而形成软件产业内部的合理分工,实现软件的产业化生产。软件产品线结构如图5-3所示。

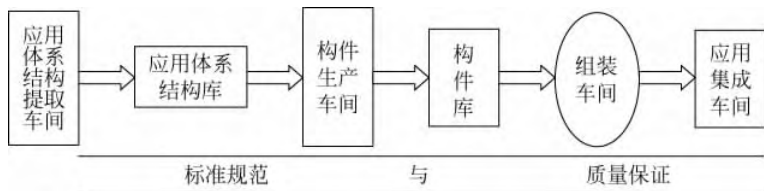


图 5-3 软件产品线

5.3.3 软件开发工具

1. 软件开发工具的概念

软件开发工具是用于辅助软件生命周期开发的计算机程序。它是在高级程序设计语言之后,软件技术进一步发展的产物。其目的是在开发软件过程中给予人们各种不同方面、不同程度的支持或帮助,支持软件开发的全过程,而不是仅限于编码或其他特定的工作阶段。

2. 软件开发工具的分类

软件开发工具有以下几种分类。

- (1) 软件需求工具,包括需求建模工具和需求追踪工具。
- (2) 软件设计工具,用于创建和检查软件设计,因为软件设计方法具有多样性,这类工具的种类很多。
- (3) 软件构造工具,包括程序编辑器、编译器和代码生成器、解释器和调试器等。

(4) 软件测试工具,包括测试生成器、测试执行框架、测试评价工具、测试管理工具和性能分析工具。

(5) 软件维护工具,包括理解工具(如可视化工具)和再造工具(如重构工具)。

(6) 软件配置管理工具,包括追踪工具、版本管理工具和发布工具。

(7) 软件工程管理工具,包括项目计划、追踪工具、风险管理工具和度量工具。

(8) 软件工程过程工具,包括建模工具、管理工具和软件开发环境。

(9) 软件质量工具,包括检查工具和分析工具。

3. 软件开发工具的功能要求

软件开发工具应提供以下 5 个方面的功能。

(1) 认识与描述客观系统。这只用于软件工程的需求分析阶段。由于需求分析在软件开发中的地位越来越重要,人们迫切需要在明确需求和形成软件功能说明书方面得到工具的支持。与具体的编程相比,这方面工作的不确定性程度更高,更需要经验,更难形成规范化。

(2) 存储及管理开发过程中的信息。在软件开发的各阶段都要产生及使用许多信息。当项目规模比较大时,信息量就会大大增加,当项目持续时间较长的时候,信息的一致性就成为一个十分重要、十分困难的问题。如果再涉及软件的长期发展和版本更新,则有关的信息保存与管理问题就显得更为突出了。

(3) 代码的编写或生成。在整个软件开发工作过程中,程序编写工作量占了相当比例,提高代码的编制速度与效率显然是改进软件工作的一个重要方面。所以,代码自动生成和软件模块重用是必须考虑的两个方面。

(4) 文档的编制或生成。文档编写也是软件开发中十分繁重的一项工作,不仅费时费力,还很难保持一致。在这方面,计算机辅助的作用可以得到充分发挥。在各种文字处理软件的基础上,已有不少专用的软件开发工具提供了这方面的支持与帮助,有利于保持与程序的一致性,而且最后归结于信息管理方面的要求。

(5) 软件项目管理。这一功能是为项目管理人员提供支持的。对于软件项目来说,一方面,由于软件的质量比较难以测定,所以不仅需要根据设计任务书提出测试方案,还需要提供相应的测试环境与测试数据,人们希望软件开发工具能够提供这些方面的帮助;另一方面,当软件规模比较大的时候,版本更新后,各模块之间及模块与使用说明之间的一致性以及向外提供的版本控制等,都可能存在一系列十分复杂

的管理问题,如果软件开发工具能够提供这方面的支持与帮助,将有利于软件开发工作的进行。

5.4 知识工程与数据工程

5.4.1 知识工程与数据工程概述

1. 知识工程

知识工程(Knowledge Engineering)是一门新兴的工程技术学科,最早由美国人工智能专家 E. A. 费根鲍姆提出。它是社会科学与自然科学相互交叉和科学技术与工程技术相互渗透的产物。知识工程是运用现代科学技术手段高效率、大容量地获得知识、信息的技术,目的是为了最大限度地提高人的才智和创造力,掌握知识和技能,提高人们借助现代化工具利用信息的能力,为智力开发服务。作为一种工程技术,知识工程主要是研究如何组成由电子计算机和现代通信技术结合而成的新的通信、教育、控制系统。在建立专家系统时,主要处理专家或书本知识,内容包括知识的获取、知识的表示、知识的运用和处理 3 个方面。人工智能与计算机技术的结合产生了所谓“知识处理”的新课题,即要用计算机来模拟人脑的部分功能,解决各种问题,回答各种询问,从已有的知识推出新知识等。

知识工程过程包括如下 5 个活动。

(1) 知识获取。包括从人类专家、书籍、文件、传感器或计算机文件获取知识,知识可能是特定领域或这个活动特定问题的解决程序,也可能是一般知识或者是元知识解决问题的过程。

(2) 知识验证。即验证知识,直到它的质量是可以接受的。测试用例的结果通常被专家用来验证知识的准确性。

(3) 知识表示。获得的知识被组织在一起的活动叫作知识表示。这个活动需要准备知识地图及在知识库中进行知识编码。

(4) 推论。这个活动包括软件的设计,使计算机做出基于知识和细节问题的推论,然后该系统可以根据推论结果给非专业用户提供建议。

(5) 解释和理由。这个活动包括设计和编程的解释功能。

知识工程的过程中,知识获取被许多研究人员和实践者认为是一个瓶颈,限制了专家系统和其他人工智能系统的发展。

2. 数据工程

数据工程的概念早在 20 世纪 80 年代就被提出,并同时建立了相应的学术组织。1984 年,国际数据工程大会由美国电气及电子工程师学会 IEEE 发起并组织召开,该会定期在世界各大城市轮流举行相关议题的学术研讨会,它是一个信息管理系统的学术协会。该组织给出的数据工程定义为“关于数据生产和数据使用的信息系统工程”,主要内容包括数据建模、数据标准化、数据管理、数据应用和数据安全等。由此可见,数据工程实质上是指数据库建设与管理的工程,虽然它所涉及的主要内容直接与大数据有关,如结构化数据表示、数据管理等,但它重点强调的是有关数据库方面的信息理论、方法和技术,在相当程度上弱化了大数据分析的价值发现功能。

学者李腊生认为,数据工程应该是突出大数据分析价值发现目标的工程技术,是建立在大数据背景下的数据工程。从这个意义上说,数据工程是指将工程思维引入大数据领域,综合采用各种工程技术方法设计、开发和实施新型的数据产品,并利用相关数据分析技术创造性地揭示与发现隐藏于数据中的特殊关系,为价值创造与发现提供系统解决方案的一门学科,是大数据、信息技术与工程方法的综合体,如图 5-4 所示。

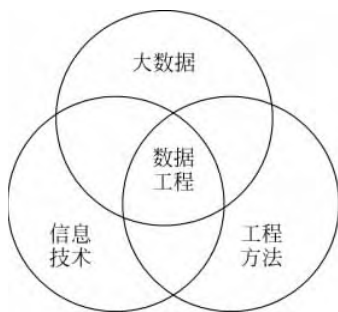


图 5-4 数据工程定义

5.4.2 知识管理与数据管理

1. 知识管理

21 世纪,企业的成功越来越依赖于企业所拥有知识的多少和质量,利用拥有的知识创造竞争优势和持续竞争优势对企业来说始终是一个挑战。

知识管理是知识经济时代涌现出来的一种新的管理思想与方法,它融合了现代信息技术、知识经济理论、企业管理思想和现代管理理念。知识管理,即在组织中构建一个量化与质化的知识系统,让组织中的资讯与知识,通过获得、创造、分享、整合、记录、存取、更新及创新等过程,不断回馈到知识系统内,形成永不间断的知识累积和智慧循环,在企业组织中成为管理与应用的智慧资本,帮助企业做出正确的决策,以适应市场的变迁。简言之,知识管理就是对知识、知识创造过程和知识的应用进行规划和管理的活动。

知识管理要遵循 3 条原则:①积累原则,知识积累是实施知识的管理基础;②共享原则,指一个组织内部的信息和知识要尽可能公开,使每一个员工都能接触和使用公司的知识和信息;③交流原则,即在公司内部建立一个有利于交流的组织结构和文化气氛,使员工之间的交流毫无障碍。知识积累是实施知识管理的基础;知识共享是使组织的每个成员都能接触和使用公司的知识和信息;知识交流则是使知识体现其价值的关键环节,它在知识管理的 3 个原则中处于最高层次。

根据知识能否清晰地表述和有效转移,可以把知识分为显性知识(Explicit Knowledge)和隐性知识(Tacit Knowledge)。显性知识也称编码知识,人们可以通过口头传授、教科书、参考资料、期刊杂志、专利文献、视听媒体、软件和数据库等方式获取,也可以通过语言、书籍、文字、数据库等编码方式传播,容易被人们学习。隐性知识是迈克尔·波兰尼(Michael Polanyi)在 1958 年从哲学领域提出的概念,指那种人们知道但难以言述的知识。



图 5-5 隐性知识与显性知识的转化关系

显性知识和隐性知识相互转化的过程:①群化(Socialization),即通过共享经验产生新的意会性知识的过程;②外化(Externalization),即把隐性知识表达成为显性知识的过程;③融合(Combination),即显性知识组合形成更复杂、更系统的显性知识体系的过程;④内化(Internalization),即把显性知识转变为隐性知识,成为

企业的个人与团体的实际能力的过程。如图 5-5 所示为隐性知识与显性知识的转化关系。

2. 数据管理

数据管理是利用计算机硬件和软件技术对数据进行有效的收集、存储、处理和应

用的过程,目的在于充分有效地发挥数据的作用。实现数据有效管理的关键是数据组织。

数据管理经历了人工管理、文件系统、数据库系统 3 个发展阶段。在数据库系统中所建立的数据结构,更充分地描述了数据间的内在联系,便于数据修改、更新与扩充,同时保证了数据的独立性、可靠性、安全性与完整性,减少了数据冗余,提高了数据共享程度及数据管理效率。

5.4.3 知件

1. 知件

知件是独立的、计算机可操作的、商品化的、符合某种工业标准的、有完备文档的、可被某一类软件或硬件访问的知识模块。

专家系统和知识库在某些方面类似于知件,但它们都不是知件。专家系统是传统意义上的软件,因为它包括以推理程序为核心的一系列应用程序模块。知识库也不是知件。首先是因为它至少包含一个知识库管理程序,并不满足知件的基本条件(即只含知识);其次是因为这些知识库的知识表示和界面一般不是标准化的,难以用即插即用的方式和任意的软件模块组合使用,而且一般的知识库还没有商品化。知件应该成为一种标准的部件,更换知件就像更换计算机上的插件一样方便。

通过知件的形式,可以把软件中的知识含量分离出来,使软件和知件成为两种不同的研究对象和两种不同的商品,使硬件、软件和知件在 IT 产业中三足鼎立。对软件开发过程施以科学化和工程化的管理,就形成了软件工程。类似地,对知件开发的过程施以科学化和工程化的管理,就形成了知件工程,两者有某些共同之处,但也有很多不同。计算机发现知识,或计算机与人合作发现知识已经成为一种产业,即知识产业。而如果计算机生成的是规范化的、包装好的、商品化的知识,即知件,那么这个生成过程(包括维护、使用)涉及的全部技术的总和可以称为知件工程。它与软件工程既有共同之点,也有许多不同的地方,从某种意义上可以说,知件工程是商品化和大规模生产形式的知识工程。

2. 知件工程

根据知识获取和建模的 3 种不同方式,知件工程有下述 3 种开发模型。

(1) 熔炉模型。它适用于存在着可以批量获取知识的知识来源的情形。采用类自然语言理解技术,让计算机把整本教科书或整批技术资料自动地转换为一个知识库,也可以把一个专家的谈话记录自动地转换为知识库。这个知识库就称为熔炉。由于成批资料中所含的知识必须分解成知识元后在知识库中重新组织,特别是当这些知识来自多个来源(多本教科书、多批技术资料、多位专家及它们的组合)时,更需要把获取的知识综合起来,这种重新组织的过程就是知识熔炼的过程。熔炉中的知识称为知识浆。熔炉模型的基本结构如图 5-6 所示。

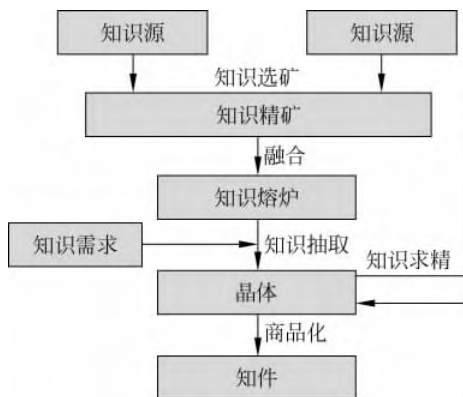


图 5-6 熔炉模型

(2) 结晶模型。它适用于从分散的知识资源中提取和凝聚知识。结晶模型的基本构思:在知件的整个生命周期中,新的、有用的知识是不断积累的,它需要一个获取、提炼、分析、融合、重组的过程。从这个观点看,人们周围的环境更像是一种稀释了的知识溶液,提取知识的过程就像是一个结晶过程。由于其规模之大,所以称它为知识海。而熔炉模型中的知识浆则是浓缩了的知识溶液。对知识的需求就像一个结晶中心,围绕这个中心,海里的知识不断析出并向它聚集,使结晶越来越大。知识晶体的结构就是知识表示和知识组织的规范,需要两个控制机制来控制知识晶体的形成和更新过程。第 1 个机制称为知识泵,它的任务是从分散的知识源中提取并凝聚知识。前文提到的类自然语言就是这样一种知识泵。它不仅可以控制知识析取的内容,还可以控制知识析取的粒度。已经获得的知识晶体可以作为新的知识颗粒进入

结晶模型如图 5-7 所示。

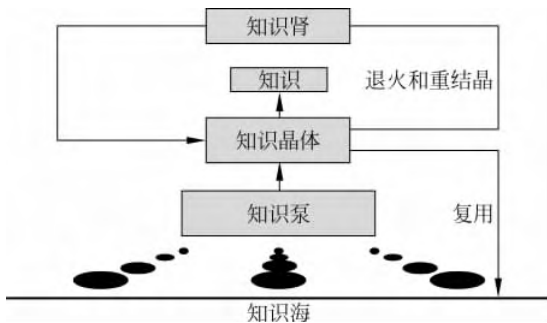


图 5-7 结晶模型

享隐知识)4 个阶段,如图 5-8 所示。

还只是一个半成品,需要经过进一步的加工才能成为知件。

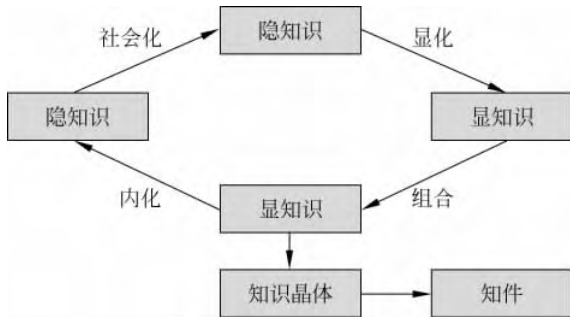


图 5-8 螺旋模型

思考题

1. 什么是计算机语言？什么是软件？什么是程序？
2. 软件怎么分类？
3. 程序有哪几种基本结构？程序有几种执行方式？
4. 简述计算机语言发展历史。
5. 什么是第四代语言？
6. 什么是操作系统？
7. 简述操作系统的历史、功能和分类。
8. 介绍几种主要的操作系统。
9. 什么是软件工程？什么是软件生命周期？
10. 简述结构化方法。
11. 简述面向对象方法。
12. 简述软件复用和构件技术。
13. 什么是知识工程？什么是数据工程？
14. 什么是知识管理？什么是数据管理？
15. 什么是知件？