

软件工程及其开发环境

Software Engineering and Its Developmental Environments

杨芙清 方裕 郑刚

(北京大学计算机科学与技术系)

提 要 本文研究了软件工程中的各种技术和方法,探讨了计算机辅助软件工程(CASE)和软件工程环境的研究状况,并且介绍了一个基于面向对象技术的集成化软件工程开发环境“青鸟”系统。

软件工程的提出源于60年代末期“软件危机”并在较短的时间内发展成为一个完整的学科方向。软件工程旨在为软件的开发提供工程化的手段,从而提高软件系统的可靠性、可理解性和易维护性,提高软件生产率,降低开发成本,使软件生产摆脱“手工作坊”式的落后生产方式,成为真正的工业化大生产。在软件工程学科方向形成以来的20多年间,就理论研究和工程实践两个方向进行了大量的工作。各种方法、技术的研究取得了长足的进展,其中有些已经比较成熟,并广泛应用于软件的开发,而也有各种新方法和新技术还处在不断的探索过程中。特别是近年来软件工程环境的出现,为软件工程开发注入了新的活力,使软件生产初步具备了工业化生产的能力。

一、软件开发方法和软件工程模型

为了使软件的开发达到工程化生产的要求,人们研究并提出了许多种软件开发方法,相应的研究领域称为软件方法学。

在现有的软件开发方法中,最成熟、应用最为广泛的是结构化开发方法(Structured Developing Method)。它包括结构化分析、结构化设计,以及结构化程序设计,为软件开发提供了一套规范的方法。结构化开发方法的基本思想是自顶向下逐步求精,分析大型软件系统的开发复杂性。由于其快速、自然和方便的特点,以及较长的开发历史,而得到不断完善和广泛应用。据调查,在欧、美、日本和我国的软件行业中,90%以上的项目至今仍用结构化方法进行开发。

与结构化方法相应的瀑布模型(Waterfall Model)将软件开发的整个周期划分为需求分析、系统设计、编码、测试与软件维护等几个阶段,提出了软件生命周期(Software Life—Cycle)的概念,成为开发软件产品的一个行之有效的工程化模型。瀑布模型经过多年的发展和完善,业已为人们广泛接受,成为一种传统的和标准的开发模式。

从80年代开始,人们逐渐认识到瀑布模型的不足。它的主要缺点在于:生命周期内各开发阶段的划分试图把一个充满回溯的过程加以线性化,并硬性加以割裂,并不能从本质上反映软件开发过程本身的规律;过分强调复审,并不能完全避免较为频繁的变动。例如,任务需求的变化会导致整个系统修改的困难,重新开发和维护也是难处理的问题。因而人们提出了其他的开发模型,最著名的为速成原型方法。(Rapid Prototyping)它并不严格遵循瀑布模型所划分的开发阶段,而强调软件开发人员与用户的不断交往,通过原型的演进不断适应用户任务改变的需求,将维护和修改阶段的工作尽早进行,使用户验证提前,从而希望软件产品更加适用。此外近年来人们还提出增量式(Incremental)开发模型和螺旋式(Spiral)开发模型等,它们都从不同的侧重点试图避免瀑布模型的缺陷,描述适合软件开发规律的模式。

二、面向对象的开发方法

80年代初,Xerox PARC推出了新颖的程序设计环境Smalltalk,从而提出了一种全新的软件开发方法——面向对象的方法(Object-oriented

Technology)。面向对象的方法的基本思想是:对问题领域进行自然的分割,以更接近人类通常思维的方式建立问题领域的模型,以便于对客观的信息实体进行结构模拟和行为模拟,从而使设计出的软件尽可能直接地表现问题求解的过程。在这里,对象作为描述信息实体的统一概念,通过封装、类属、继承和实例化等机制支持软件系统的构造,并为软件重用提供了支持。这一领域的研究近年来非常活跃,出现了各种面向对象的语言,如Smalltalk, C++, objective-C, Eiffel等;采用面向对象方法进行需求分析和设计的工作也取得了一定的成果。

三、软件工程的研究方向

目前关于软件工程技术的研究主要沿着两个方向发展。一个方向是基于形式化技术和人工智能技术,来构造自动化、智能化的软件工程环境。形式化方面的工作是以严格的数学和逻辑系统为基础,希望达到从软件的规格说明到软件的代码的自动转换和过渡。典型的工作如著名的VDM,慕尼黑大学主持的CIP项目,中国科学院的XYZ系统等。这些系统的优点在于统一的形式框架,严格的数学逻辑基础,正确性证明和自动化方法;但其弱点在于适用性不够,表现在对于大型复杂系统难于完全形式化地描述,另外对于开发人员的数学素质往往有较高的要求,因而应用面尚显狭窄。智能化开发环境的研究希望借助于人工智能的技术,使计算机系统更多地具有人类思维的优点。这方面的工作如“七五”国家科技攻关项目天马系统。但是,大型软件系统的开发是一项复杂的系统工程,涉及技术、组织、文化等多个方面,其决策知识庞杂,难以简炼。在很大程度上影响其实用性。值得注意的是借助人工智能和认知科学的成果,有可能解决某些难于构造形式化模型或称不良结构(ill-structured)等问题领域的需求。这方面的研究近年来也越来越得到人们的关注。软件工程的另一个主要方向是在传统的软件开发过程基础上进行软件工具和开发环境的构造,从而辅助开发人员提高生产效率,改善软件的质量。这一方面的重要成果是计算机辅助软件工程(CASE)及软件工程开发环境的研究。

四、软件工程开发环境

随着计算机在各个领域的广泛应用(例如MIS, CAD/CAM, CAI等),人们希望将这一先进的工具应用于计算机软件本身的开发,提高软件工程原则、方法和技术的可操作性,为软件工程提

供方便的辅助手段,因而将这种辅助称为计算机辅助软件工程(CASE)。CASE是一个较新的领域,在国际上形成和发展不过十余年时间。CASE经历了两个发展阶段。第一代CASE是对软件开发过程中的各种活动孤立地进行辅助,包括各种软件工具和工具包,如需求分析工具、设计工具、编码和调试工具、测试工具和逆向工程工具等。大量的软件工具对软件工程实践起了很大的推动作用,它们基本上覆盖了软件生命周期的各个阶段,可以部分支持软件开发和维护的各项活动,并逐渐开始为广大软件技术人员接受和采用。

由于大量的工具是独立研制的,彼此之间联系松散,使用方法差异很大,甚至互不兼容,因而如何构造能够支持软件生命全周期,适合不同层次软件技术人员使用,支持包括技术活动和管理活动等软件生产各方面活动的集成化软件工程支撑环境,成为第二代CASE的主要方向。这一代CASE大都采用数据库作为基础来组织环境。从环境的结构、功能和组成看,产生了提高个体开发能力的单体型CASE环境,支持大型软件开发的协同型CASE环境,支持分布式开发的分散型CASE环境等。典型的包括ExceleratorCASE(Index公司)、AD/Cycle(IBM公司)、Softbench(HP公司)等。利用面向对象技术开发的环境目前是国际上较为新颖和活跃的课题。这种环境一般以对象库(Object Base)为基础来组织环境,支持封装性、实例化、类属和继承,达到了较高的集成度和开放性的统一。这方面已有一些成果,如布劳恩大学的Garden, Atherton公司的Software Back Plane, IDE的Software through Picture,以及我国“七五”期间的国家科技攻关成果青鸟系统等。

实践表明,采用CASE工具和环境进行软件开发的机构在不同程度上提高了开发效率。据调查,在软件生命周期的前期,大约可使需求分析和设计工作的效率提高30~40%。另外,由于使用了CASE工具和环境,使开发过程中更容易发现错误和不一致性,使需求规格说明更好地反映用户的要求,并使软件开发的管理工作更加有效,从而使软件质量能够得到提高。

五、青鸟系统

青鸟系统是一个集成化的软件工程支撑环境,是我国“七·五”重点科技攻关项目SEP的重要成果之一,其研制成功对我国软件在工程、技术和方法上的发展具有重要意义。

青鸟系统的研制目的是建立软件工业化生产系统的核心,促进软件工程技术的应用和发展,提高软件生产率,改善软件产品的质量,降低软件生

产和维护的费用,为系统软件和应用软件的开发提供良好的支撑环境。

青鸟系统的研制涉及到各种概念、方法、技术和工具的选择,采用了较成熟的技术对软件生命周期的各个阶段予以支持。其主要特色在于集成化、开放式和面向对象的方法。

由于面向对象的方法接近人类的自然思考方式,符合软件工程开发环境的需要,提高了可重用性。因此,青鸟系统采用面向对象的概念和技术作为环境的统一的概念模型和理论基础,由环境信息库BETA—87、各种工具和可剪裁的统一用户界面所组成。环境信息库和统一的用户界面是青鸟系统的基本集成机制,各种工具和软件开发项目的信息都作为被集成的对象存放在环境信息库中,由统一的用户界面来操纵,用户通过界面使用工具从事开发工作。整个环境的设计以面向对象的技术为基础,实现了概念、方法、技术、工具四个层次的逻辑集成,同时体现了统一的用户观点、统一的数据存放方式和统一的操作模式三方面的物理集成。

作为CASE环境,青鸟系统以软件生命周期的瀑布模型和结构化的软件开发方法为基础提供成套的软件工具,连续支持软件开发维护全过程,同时兼顾对速成原型等新技术的支持。集成的主要工具包括:

需求分析阶段——提供了需求描述及分析工具RDATA等,可以支持数据流程图及数据字典的描述与分析,为书写和分析软件需求规格说明书提供辅助手段。

软件设计阶段——提供了软件设计工具SDT等,可以支持模块分解以确定软件系统的总体结构,支持详细设计的伪码说明和模块结构图的编辑、转换工作。

编码阶段——提供了C语言编码与排错工具CCD等,可以提供对C语言程序语法制导的编辑和调试一体化的支持。

软件调试阶段——提供了C程序测试工具CPTT等,可支持进行C语言的静态分析和动态测试。

运行维护阶段——提供了C语言程序理解工具CPCT等,可支持软件维护和逆向工程工作。

贯穿整个软件生命周期,青鸟系统提供了统一的信息管理机构——环境信息库和统一的用户界面,软件开发与维护过程中产生的所有信息实体,包括项目、工具、中间信息等,均在环境信息库系统的监控下,通过对用户界面的操作进行存取、加工和传播使用。用户界面提供了描述和定义开发对象的手段,具有统一性、友好性和可剪裁性。

此外,青鸟系统提供了一套完整的软件工程

管理措施和相应的支持机制,支持软件工程项目中各类人员的协同工作,从而增强管理工作的可见性和自动化。

青鸟系统的开放性表现在两方面:提供了固定类型“UNIX对象”,具备与UNIX的接口能力;提供了一套面向对象的工具开发设施,支持对象及其加工方法的构造继承与重用,从而使得环境具有积累能力。

青鸟系统的集成化,开放性和面向对象的特点,使其在软件的工程、技术和方法的研究与开发上占有突出地位。

六、未来的发展与困难

软件工程经过二十多年的发展,取得了许多经验和成果,但也有很多教训。从近年来的研究情况来看,软件工程的未来必须考虑以下几个方面的因素:

重视应用——软件工程目前已成为“计算机化的应用工程”,软件工程师也变为系统工程师。为了选择合适的系统软件与硬件,他们除了了解计算机知识,也必须了解应用领域。

重视相关领域——软件工程师不能狭隘地就软件工程论软件工程,为了使软件工程真正走向实用,必须重视硬件的支持、应用的需要以及人员教育等问题。

重视形式化方法——从目前的情况来看,形式化方法已逐步从理论研究走向实际应用。为了把软件工程建立在严格的科学基础上,必须重视形式化方法在软件工程中的地位。

注重对实时系统的支持——这是一个很重要的应用领域,应根据实时系统的要求,支持其高可靠性和高安全性。

认真对待新技术——目前,软件开发的各种新技术层出不穷,应该认真研究各种新技术与软件工程技术的关系,借助于它们使软件工程本身不断完善和实用。

软件工程环境标志着软件工程的发展水平,这方面的研究是软件工程界的热点之一。从发展动向来看将致力于以下几个方面的突破:

集成化与开放性的统一——如何将这一对既相互矛盾又相互影响的因素进行有机地结合和折衷,将是影响环境开放和应用的一个重要课题。

控制集成和过程集成——对象的概念为环境的数据集成提供了有力的手段。此外,环境应当提供一种组织和控制的手段,支持软件开发人员组织软件开发活动以及协调工具之间的关系,实现工具之间的平滑过渡。使开发者能够方便地按照

(下转第40页)

其研究题目之一即是全面鉴定法的标准和实施。可以说,这一方法在美还是刚刚起步,深入的研讨方兴未艾。

多文化多语言教育

近年来,由于不断的移民和各民族出生率之间的差异,美国人口的民族比例发生较大的变化,少数民族的人口增长速度加大。据美国审计署统计,从1980年到1989年间,美国人口平均增长率为9.2‰,然而白人增长率只有7‰,黑人达14.5‰。拉美西班牙血统人1989年已占全国人口的8.2%。1989年在外国出生的美国公民占人口总数的5.8%。伴随人口的变化是政治上的变化。少数民族参政日趋踊跃。这些社会变化反映在教育上,则是近年来的多文化多语言的教材和教程的发展。

美国普通教育历来是以白人文化为中心,英语为媒介。为了适应多元社会的发展和变化,多文化的改革集中体现在教材上,其中以文科教材为最。无论历史还是文学,在教材编写中一方面大大增加了以少数民族(印地安人、黑人、拉美西班牙血统、亚裔等)为题的史实和选用少数民族作家(尤其是女作家)的作品;另一方面也增加了以少数民族本土文化(世界各国)为题的课程内容,加强教育的世界性和全球性。

与多文化教育并驾齐驱的是双语或多语教育。为了让新移民和少数民族子女能顺利地接受基础教育,不少学校纷纷设立双语或多语课程。学生可以用本语接受教育,同时学习英语,当他们的英语水平提高后,再转入普通课程。非英语学生稀少的乡村城镇学校,则可从教育部申请特殊经费对非英语学生进行双语个别辅导教学,保证他们的学习进度。

多文化多语教育的实施减少了少数民族和新移民子女在语言文化上的障碍,为他们接受和完成基础教育提供了条件。由于少数民族和新移民子女的辍学率在全国为首,多文化多语言教育亦为基础教育提供了保障。当然,从多文化多语言教育改革中受益的不只是少数民族和新移民子女。这一改革大大丰富了教材和教程,使学生在过程中增长文化和社会知识,开阔眼界,消除成见,有利于民族团结和国家昌盛。

然而,多文化多语言的教程教材改革也有很大局限。首先在裁减教育经费时,这些项目往往首当其冲。同时,尽管少数民族学生数目急剧上升,在学校内任教的少数民族教师数目却由于种种原因不断下降。在双语式多语教育中许多教师是临时任教,没有受过系统的专业培训,多文化、多语

言教育质量受到很大影响。总之,这项改革在多大程度上能摆脱现实生活的阴影,消除种族歧视和隔阂,使学生真正受到平等教育仍是个公开疑问。

上述几点远不能包括美国普通教育的全面发展,只是以管窥豹,略见一斑。但可以看出美国普通教育的特点首先表现在从传统的一元化到现代的多元化转变。这一转变既反映在教程、教材和科目上,也反映在教学方式方法、教育测定和教学手段上。转变过程中扩展了横向、纵向的网式联络。综合教程体现了科目间的勾通,集体讲课表现了教师间的合作,电子化教学充分利用了现代科技的发展,全面鉴定法把考与学融为一体,多文化多语言教育力图消除民族隔阂和语言障碍……这些变化使割裂的教育变为息息相关的整体,教育与现实更紧密地结合,生动多彩,更好地调动师生的积极主动性。

(责任编辑 蔡德诚)

(上接第36页)

各种开发过程进行工作。

界面的自动生成——环境应向用户提供风格一致的操作界面,各种工具的使用界面由环境提供的机制统一生成。这将有利于环境的应用。

软件工程开发环境的标准化——如同操作系统和用户界面系统一样,软件工程环境的标准化日益得到人们的重视,国际上许多组织都在积极从事这方面的工作。欧洲共同体在八十年代中期推出了可移植公共工具环境标准PCTE;美国国防部主持制定了面向ADA的开发环境标准CAIS;美国商业部的NIST机构也提出集成化软件工程环境的参考模型,试图描述环境的各个组成部分之间的接口标准。标准化工作的开展必将推动和促进软件工程的发展和应用。

软件工程的发展为软件开发带来了一系列观念上的革新。除了技术本身的因素之外,非技术因素在软件工程发展中的作用和影响也不容忽视。如同50年代、60年代从汇编语言编程到高级语言编程的转变一样,现在软件人员也面临一个新的观念上的转变。在这里,社会的、心理的、人际的诸多因素对于软件工程的应用都有重要影响。由于众多的潜在用户都形成了自己的工作习惯,不同的软件生产机构都有各自的开发规范和项目组织方法,在软件工程技术的普及应用之中,如何改变开发人员的工作习惯和进行技术上的再教育将是一个非常困难的任务。这些非技术因素有可能变成阻碍软件工程发展和推广的严重障碍,因而必须引起足够的重视并认真采取措施加以解决。

(责任编辑 宋义荣)