

软件技术漫谈

徐家福 (210008南京大学计算机软件研究所)

摘 要

作者根据多年来从事计算机软件研究与开发的体会, 撰成此文, 就软件技术的现状与前景进行了讨论, 内容包括: 传统技术、新技术、难点、以及结束语四部分。

六十年代迄今, 软件技术不断发展, 然而, 相对硬件而言, 其发展速度颇为缓慢, 令人兴叹“望尘莫及”。原因众多, 难以尽述, 不揣浅陋, 略陈己见, 就正通人。

1. 传统技术

1.1 基础

软件发展初期, 紧密依托硬件, 以冯·诺依曼结构计算机系统为其基础。换言之, 冯氏单机为其物质基础。

1.2 特征

1.2.1 本质串行 冯氏机器的本质串行性导致传统软件技术主要针对顺序计算。由于基础是单机, 并发性表现为多任务的夹插执行, 多机系统出现后, 才可能考虑真正的并行。

1.2.2 实现级语言 传统程序设计过程仅当问题明确, 解法选定, 算法确定后, 才选用合适的语言来描述算法, 并能借助各类处理程序进行处理。其所用语言乃用以描述算法的语言, 习惯上称作实现级语言。古典高级语言从 FORTRAN, ALGOL, COBOL, PASCAL, 直到 ADA 均属此类。它们相对低级语言而言, 对程序人员的负担有所减轻, 但对计算机用户而言, 要求仍苛。

1.2.3 开发方法自顶向下 结构程序设计方法出现后, 风靡一时, 符合人们思考

问题、解决问题的过程, 先总体, 后局部, 由粗到精, 逐步精化。结构性表现为层次性, 便于稳扎稳打, 步步为营。软件开发的瀑布模型与结构程序设计方法本质一致。

1.2.4 软件本身缺乏智能 软件一经设计定型, 其功能即已确定。由于软件本身缺乏智能, 无法在工作过程中视需要, 通过学习, 扩大功能, 以适应新的需求。

1.2.5 理论基础薄弱 理论基础主要为自动机理论, 形式语言理论, 并且着重语法研究, 语义研究薄弱, 语用研究未见开展。

1.3 代表性工作

1.3.1 实现级语言 基本上以 N. Chomsky 语法理论为基础的各类古典高级语言, 从 FORTRAN, ALGOL, 直到 ADA。其主要成份是顺序成份, 所含并发成份只不过是顺序成份的基础上添加少许并发设施而已, 这类语言, 如前所述, 均为实现级语言, 对用户要求较苛, 使用欠便。

1.3.2 单机操作系统 各类单机操作系统, 用以控制、管理多项任务的执行, 以及多种资源的分配与使用。

1.3.3 设计、实现级工具 在瀑布模型下, 以结构程序设计方法为手段, 发展了各类软件工具。这主要是设计级、实现级的工具, 用以支撑各类软件的开发与维护。

1.3.4 程序正确性验证程序 六十年代中、后期,出现了 P. Naur 与 R. W. Floyd 关于程序正确性验证的快照法与断言法。这些方法确有助于探讨程序的正确性,以及在一定意义下验证程序的正确性。然而,由于它们存在如下缺陷: 1. 验证程序的书写者须对被验证的程序的结构透彻了解; 2. 验证程序往往较被验证的程序更为复杂; 3. 验证程序本身也存在正确性问题。因而,二十多年来,虽然文章发表不少,具体系统仍然处于实验阶段,恐难投入实用。

2. 新技术

2.1 软件领域问题

六十年代中期,曾经出现“软件危机”,致使软件发展失控,软件工程出现后,一度危机缓解,很快又出现新的危机。归结起来,问题如下:

2.1.1 功能不强 这主要是从总体上看,迄今大多数软件缺乏自适应、自学习、自组织的功能。从而导致不能一件多用,对人力、财力、物力、时间等各方面均显著不利。

2.1.2 质量欠佳 衡量软件的质量有众多因素。从用户角度看,主要是: 1. 正确性; 2. 坚固性; 3. 扩充性; 4. 复用性; 5. 兼容性。这些方面虽然一直有人研究,然而,第一,在软件工程实践中实际起作用的系统不多; 第二,迄今未见其科学的理论基础,换言之,这些问题的研究尚未上升到理论阶段。如前所述,重要的正确性问题理论上和实际上均未解决。

2.1.3 生产率低 软件工程已出现二十余年,但软件开发与维护基本上仍处于手工阶段,致使软件生产率奇低。据统计,一名中等水平的程序人员年生产率不过4000行左右程序代码。

2.2 类别

2.2.1 智能化技术 智能化技术指的是如何使软件具备智能的技术。如前所述,

由于传统软件本身缺乏智能,致使系统一经设计定型,功能即完全确定,无法视需要在其工作过程中扩大功能。因此,为使软件能多快好省地服务于各个领域,必须研究与开发各类具备智能的软件。智能化技术是扩大软件功能的关键途径,必须予以高度重视。

2.2.2 自动化技术 自动化技术指的是如何使软件开发过程自动化的技术。软件自动化的广义理解是,尽可能在软件开发过程中利用计算机系统,以减轻人工劳动,显著提高生产率。其狭义理解是,从用户需求定义,经功能规格说明,设计规格说明,直到可执行的代码的全部过程尽可能借助计算机系统自动进行。这是提高软件生产率的根本途径。目前设计级以下各阶段发展相对成熟,特定领域的软件自动化的实验性工作较多,而需求级、功能级的自动化技术,以及通用软件自动化系统大都处于实验探索阶段。

2.2.3 集成化技术 集成化技术指的是如何使软件具有集成性的技术。集成性一词有两层含义: 一为内部集成性。为使一系统能成为真正的开放式系统、易扩充系统,其各个组成部分之间的接口必须一致,否则,就很难将各个组成部分汇集为一个整体,也很难往原有系统中添加新的组成部分,这就是内部集成性的真谛。二为外部集成性,为使一系统能嵌入另一系统中作为其组成部分,前者的外部接口必须和后者的内部接口一致,否则,就难于将前者嵌入后者,这就是外部集成性,总之,接口一致性构成集成性的实质。发展集成化技术有助于在原有软件的基础上开发新软件,既有助于提高生产率,也有助于提高软件质量。

2.2.4 并行化技术 并行化技术指的是如何使计算机系统对各个对象的处理适当并行,借以提高系统处理实效。并行性是现代计算机系统区别于传统计算机系统的重要标志,其实现一般需软、硬结合。从软件方面看,有所谓并行粒度问题。粗粒度者如程

序与程序(任务与任务)的并行,中粒度者如模块与模块的并行,细粒度者如运算与运算的并行。有的语言(如FP语言)本身具有潜在的并行性,为使这些潜在的并行性转为现实并行性,必须研制相应的软件。并行化技术涉及的问题有:1. 并行算法,2. 并行语言,3. 并行处理程序,4. 自动问题分解,5. 并行度的权衡与选定,等等。总之,并行化技术是提高系统实效的关键技术,也是难度很大的技术。

2.2.5 自然化技术 自然化技术指的是如何使人与计算机系统的接口界面尽可能采用自然语言的技术。如所熟知,社会信息化的主要特征是计算机及其服务业面向人人。当然不能也不要求“人人”均了解人工语言。从这个意义上说,只有实现了软件自然化,用户与系统的接口界面尽可能采用自然语言,计算机系统才可能真正家喻户晓,才可能实现社会信息化。这里首先要研究自然语言,对炎黄子孙而言,首先要研究汉语,要研究汉语的语法与语义的形式化,用合适的形式体系去描述汉语的语法、语义。其次,才是汉语处理程序。不从根本上进行基础性研究,结果只能是“欲速不达,事倍功半。”

3. 难点

3.1 对软件开发过程本质的认识

3.1.1 变动性 “变是不变的真理”在软件领域表现得尤为突出。软件在使用过程中必须不断维护,其中包括1. 纠正性维护。使用中一旦发现错误,便即时纠正;2. 适应性维护。随着环境的改变,使原有软件适应新的环境;3. 完善性维护。原有软件在设计时难免有不完善之处,在使用中视需要逐步完善。变动性是软件的重要特征。

3.1.2 是否领域有关 软件开发风范与模型均和领域有关。领域不同,合适的风范与模型可能也不同。但软件开发过程的本质是否也和领域有关,目前尚难断定,共性

为何,个性为何,这些都是有待研究的问题。

3.1.3 是否思维规律有关 软件主要是由人开发的。各人思考问题的习惯、方式不尽相同,思维规律也不尽相同,但软件开发过程的本质是否也和思维规律有关,亦亟待研究。

3.2 功能智能化

3.2.1 对智能的理解 “智能”含义,众说纷纭。然而,多数人认为,智能指学习能力,亦有人认为,智能指问题求解能力,可能前一种说法更为合适。

3.2.2 软件能否智能化 如果软件智能化指的是使软件本身具备智能(参见2.2.1),则事实表明,软件是能智能化的。如果软件智能化指的是使软件具备人类智能,则当然是有条件的。

3.2.3 智能化途径 目前流行的归纳学习、分析学习、联接学习(人工神经网络)以及遗传学习等途径各有利弊,应视具体情况,酌情采用,或另创新途径。

3.3 程序正确性

3.3.1 相对性 程序正确性总是相对某一基准而言的。一般说来,程序(实现级)是正确的,指的是它能全部体现相应功能规格说明中的功能,而功能级程序(即功能规格说明)是正确的,则指它能体现需求定义中的全部需求。

3.3.2 级别 过去考虑较多的是实现级程序的正确性,今后宜更多考虑设计级与功能级程序的正确性。

3.3.3 理论与实际 理论上的正确性与实际的正确性从理论上说应该是一致的,但实际上又往往不能完全一致。问题在哪里,理论原因还是实际原因,有待研究。

3.3.4 开发过程的正确性 开发过程不正确,则任何级别的程序正确性均无法保证。

3.4 用户友善性

3.4.1 重要性 软件质量优劣的最终

Fortran新标准的主要特征及 面向过程语言的生命力问题

郭浩志 (国防科技大学计算机系, 长沙410073)

摘 要

In this paper, main characteristics of the new standard Fortran language are outlined and the life-force problem of procedure-oriented language is further discussed.

从FORTRAN77问世起(1978年),至八十年代末,为产生一个FORTRAN的新标准,曾做了大量工作。1987年秋新标准的初稿被发往世界各地广泛征求意见。在认真研究了500多份答复之后,对初稿进行了大范围的修改。在再次收集各方意见的基础上,1990年

评判者是用户。对用户不友善的软件,用户当然不会乐于使用。

3.4.2 自然语言 为使软件对用户友善,人与系统的接口界面应尽可能采用自然语言,这一点颇为重要。

3.4.3 接口界面结构 采用自然语言只是接口界面的表示问题,更重要的还是接口界面的结构。结构宜简明、力戒烦琐。

4. 结束语

软件技术过去困难,现在困难,将来仍然困难。当然,困难的性质与程度会有所不同。只有迎着困难上,战略上藐视,战术上重视,理论联系实际,双管齐下,科学态度,所面临的困难定将逐一克服,软件的本质、软件开发过程的本质定将逐步摸清。届时软件领域将出现前所未有的欣欣向荣的局面,计算机科学技术的发展将别开生面,柳暗花明又一村。

1月在美国德克萨斯州召开了FORTRAN工作会议,统一了新标准的技术工作,并非正式地取名为Fortran90。同年3月该标准的最终结构得到国际标准化组织Fortran工作组ISO-IEC JTC1/SC22/WG5和美国标准协会Fortran分委员会ANSI X3J3的双重同意。

参 考 文 献

- [1] Freeman, P., Strategic Directions in Software Engineering, Past, Present and Future, Information Processing 89
- [2] 徐家福, 软件笔谈, 计算机科学, 1990.3
- [3] Liskov Barbara, Structure of Distributed Programs, 12th International Conference on Software Engineering, March 1990
- [4] Xu Jiafu et al., Some Research on Inductive Program Synthesis, University Computing (U. K.), 1990 No.3
- [5] 徐家福、戴敏、吕建, 从软件功能规格说明到设计规格说明的转换, 计算机学报, 第14卷第2期, 1991.2
- [6] Xu Jiafu, Dai Min, Lu Jian, From What to How, Advances in Chinese Computer Science Vol.3, 1991