

换格式构造出内部表示的能力。

- Agent 系统中可以包含人的角色。人可以通过一个代理者 Agent 和设计系统相互作用,同时代理者可以和其它 Agent 相互作用,收集信息,帮助人完成工作。

- 某些 Agent 用来存放知识或记录设计决定,这样设计知识和元知识在系统中可以长期保存。原则上,这是非常容易实现的,因为所有的信息可以通过 Agent 间的通讯语言进行交流。

- 所有的 Agent 都可以异步地加入或离开一个工作系统。这对于大型设计系统是一个十分重要的问题。但在 Agent 加入系统时,如何建立系统的初始状态是一个困难的问题。

- Agent 结构表现出良好的伸缩性。

- 多 Agent 结构的实现十分困难,在这一领域鲜有十分成功的成果。

- Agent 间的通讯<sup>[4]</sup>是另一个困难的问题。首先,Agent 间必须有一套能交换信息的最小协议集,其次,Agent 必须拥有共享的语义。目前的研究基本上都是根据言语学(Speech act)进行的。

- 大型系统中,完全的一致是不可能达到的,关键是如何通过协商来消除这种不一致。Agent 的协商能力为消除信息的不一致提供了可能性。

- Agent 系统最吸引人的地方在于它的认知方面的能力<sup>[1]</sup>。Agent 可通过学习表现出一定的专家行为,成为设计助理。

结论 大型设计项目中,设计人员通常使用不同的工具,具有不同的知识,并不十分了解别人的行为,设计是许多设计人员行为的折中,如何构造能包含设计和产品知识的系统来改善设计行为和如何

将设计知识转交给设计产品的使用人员是构造设计系统的一个长期目标。传统的方法很难实现上述目标,基于 Agent 的设计系统似乎已给出了部分答案。在这一领域中,仍需做大量的工作去验证这一问题。

#### 参考文献

- [1] J. Bates, The role of emotion in believable agents, CACM, 37(7)1994
- [2] C. Castelfranchi, Guarantees for autonomy in cognitive agent architecture, In: Intelligent agents, theories, architectures and languages, Springer-Verlag, 1995
- [3] M. R. Cuskosky et al., PACT: An experiment in integrating concurrent engineering systems, IEEE Computer, 26(1)1993
- [4] M. R. Genesereth and S. P. Ketchpel, Software agents, CACM, 37(7), 1994
- [5] J. Macguire et al., SHADE, Technology for knowledge-based collaborative engineering, J. of concurrent engineering, applications and research, 1(3)1993
- [6] Michal P. Case et al., Discourse model for collaborative design, Computer-Aided Design, 28(5) 1996
- [7] H. Park et al., An agent-based approach to concurrent cable harness design, AIEDAM, 8(1) 1994
- [8] M. Worldridge, N. R. Jennings, Intelligent agents, theory and practice, The knowledge engineering review, 10(2)1995

(上接第 89 页)

第二步,是采用“自顶向下”法:教师首先在自己熟悉的领域开设包含并行计算或面向对象专题的选修课程,然后再向不太熟悉的领域转移。

第三步,也就是本文所提出的最终目标,可采用“自底向上”的方法实现,即从导论性课程开始,到较高水平的课程,均把并行计算和面向对象的内容融入其中。这时,教师通过进修或讲授相关的选修课程,已积累了一定的经验。到时,包含合适专题的各种层次水平的教材相信已出版。本文提出的方法终

将被普遍采用。

结束语 并行计算及面向对象的概念应该贯穿计算机科学本科的整个课程体系中,以使 21 世纪的计算机研究开发人员能够灵活运用并行计算和面向对象技术去解决许多实际问题。作者认为,采用文中提出的一点一点逐步积累的方法,是让学生掌握不断出现的新概念、新方法、新技术的可行方法,因为这样不必去除传统论题的大量内容,同时也符合教学规律。(参考文献共 9 篇略)

## 24 浅谈计算机科学本科课程与新技术的融合

88-89, 64 张自力 唐雁

(西南师范大学计算机科学系 重庆630715)

**摘 要** 如何将计算机领域不断出现的新概念、新方法、新技术结合进本科教学,使学校教学尽可能跟上计算机发展的步伐,是大学本科计算机教育中的一个突出问题。鉴于此,本文提出了将并行计算和面向对象的概念等方法融入整个本科课程的课程体系改革构想,并讨论了可能的实施方案,以和同行探讨。

**关键词** 并行计算, 面向对象, 计算机教育

计算机科学作为一门高新技术新兴学科,在短短的几十年里获得了空前的发展,近年来,发展变化愈加神速,新概念、新方法、新技术层出不穷。据权威人士预测,在本世纪末下世纪初,在计算机领域,有5种关键技术:大规模并行、多媒体、面向对象、开放系统以及网络技术,应引起广大的计算机工作者极大的重视。

另一方面,大学计算机科学(专业)本科的课程内容与教学严重滞后于计算机本身的发展,这就自然引出一个问题,我们应当如何更新课程内容,以培养出具有坚实的理论与较强的应用技能,适应计算机飞速发展、面向21世纪的新型计算机专门人才?

要解决这一问题,我们认为,应该将已成为主流的方法与技术,贯穿进整个本科课程的始终,从基本的导论出发逐步深入,最后达到较高的水平。当这些主流的概念、方法与技术成为几门课程的一部分之后,每一门课程均可达到了解这些主流的概念、方法与技术的总体目标作出部分贡献。经过这样的课程体系训练的学生,在其毕业后,才不会面对已经成为主流的一些概念、方法与技术而惶惶不知所措。对于前面提到的5个关键技术,已经成为主流、应率先融入整个本科课程的技术是并行计算与面向对象。

## 1 并行计算与面向对象

并行计算已不再是一种只在实验室里才能找到的实验性方法,促使并行处理进入计算主流的三个关键领域:并行计算的计算模型、并行系统结构中的处理机间通信以及将并行系统加入通用计算环境的系统集成已取得显著进展<sup>[1]</sup>。许多计算密集型问题

均是采用涉及许多处理机的并行程序解决的,并行程序可在各种各样的环境中加以实现,从大型昂贵的大规模并行处理机(MPP)到较小规模的带共享存储的对称多处理机(SMP)。与对称多处理和分布式计算相关的领域现也逐步变得重要起来。因此,要对计算及计算机科学有一个全面的了解,对并行计算的了解就必不可少。

在 ACM/IEEE 计算教程1991<sup>[2]</sup>中,包含一些知识单元,与并行计算相关的有:AL9,并行与分布式算法;AR7,系统结构;OS3,进程协同与同步;OS10,分布与实时系统;及 PL12,分布与并行程序设计构造。实际上,计算机科学的绝大多数领域都包含了与并行计算密切相关的一些概念。

另外,1991年美国政府批准实施由戈尔副总统主持的高性能计算和通信计划 HPCC(High Performance Computing & Communication),其中有两部分:高性能计算系统 HPCS 和先进软件技术与算法 ASTA,均要求下一代计算机科学家——今天的本科生,懂得并行计算的基本概念<sup>[3,4]</sup>。

让我们再来看看面向对象方面的情况。本世纪剩下的几年,软件的主要发展是面向对象。90年代,面向对象的技术已开始为许多软件开发者和用户所理解和接受。不少软件都同时出现了面向对象的倾向,其中包括程序设计语言、用户界面、数据库、人工智能和操作系统等。“面向对象”的趋势如一股洪流,势不可挡;热遍全球的 Java 语言是面向对象的;《Computer》杂志在为纪念计算机诞生50周年举办的一次“计算机未来50年”的预测活动中,不少人士估计,“面向对象”应是软件开发人员必须掌握的一门技术<sup>[5]</sup>,IBM 公司的一项调查表明,现在许多项目都

涉及到面向对象。因此,项目经理在寻找开发人员时,应考察其在面向对象方面的技巧与经验<sup>[6]</sup>。

既然并行计算和面向对象对未来的计算机工作者如此重要,那么他们现在接受这方面训练的实际情况又如何呢?

## 2 现状

在并行计算方面,R. Miller 曾对美国各大学并行计算的教育情况作了全面的调查分析<sup>[7]</sup>。他在报告中指出,大多数规模较小的学院一般在其课程体系中都包含有一门并行计算方面的课程,作为较高水平的选修课,课程内容比较典型的是介绍各种并行系统结构及程序设计技术。一些规模较大的大学有时则开设了几门研究生水平的课程,如并行算法,并行系统结构和并行程序设计等,但为本科生一般仍只提供并行计算方面的一门导论性课程。

在我国,一些实力较强的大学一般也在高年级开设了并行计算方面的选修课,如国防科技大学1996年修订的本科教学计划中,在第4学年开设了“并行程序设计”和“分布式系统”的选修课。

在面向对象方面,虽未见有较全面的调查报告,但情况与并行计算方面大致相同。在目前国内各院校执行的教学计划中(主要是参考 ACM/IEEE 91教程和中国计算机科学93教程),对本科学生,一般只在高年级开设有“面向对象的程序设计”或“面向对象技术”这样的选修课,而程序设计语言一般也只开设 C 语言,没有开设 C++, Java 或其他面向对象的语言。

总的来说,现状是:并行计算和面向对象的概念、方法与技术仍被处理成一个专题,而没有作为典型的计算机科学本科课程的一个有机部分,这势必减弱学生将来工作的适应能力。

## 3 本科课程中的并行计算与面向对象概念

要使并行计算和面向对象概念成为计算机科学本科课程的一个有机组成部分,就必须在各种水平的课程中分别予以介绍。以 ACM/IEEE 计算教程1991(附录 A 中实现 D)<sup>[8]</sup>为例,建议将并行计算结合进如下所列的几门课程中,以便让学生懂得并行计算在许多应用中都是—种强有力的工具,这些课程是:CS101,102;计算导论;CS201;计算机系统导论;CS202;算法分析与设计;CS301;计算机组织与汇编语言程序设计;CS302;软件系统;CS303;软件工程;CS304;程序设计语言;CS305;操作系统;

CS306;计算机系统结构。

此外还有如下一些选修课:计算机通信网络;容错计算;并行与分布式计算;分布式数据库;人工智能;计算机图形学;计算机系统设计;计算理论。

### 3.1 导论性课程

在 CS101和102中,重点介绍的是有关算法的基本内容,可将并行算法和并行搜索引入,同时与传统算法进行比较,以便让学生建立起对并行计算的初步印象,但目前国内外尚无类似的教材。

### 3.2 计算机组织与系统结构

在 CS301和306中,可包含对各种并行系统结构的讨论,如 SIMD, MIMD, 共享存储型多处理机,分布存储消息传递型多计算机等。在这方面,由 Hennessy 和 Patterson 合著的 Computer Organization and Design, The Hardware/Software Interface (Morgan Kaufmann 1993年出版)一书是一本较好的教材。

### 3.3 操作系统

在操作系统课程(CS305)中,本身就包含了并行计算的一些基本概念,如同步、死锁、进程间通信等,因此可以自然地融进对并行计算机的讨论,如共享存储模型中存储访问的同步,分布存储模型中消息传递的协同等问题。由 A. S. Tanenbaum 著的 Modern Operating Systems (Prentice-Hall 1992年出版)一书是这方面较好的教材。

在程序设计语言及其他选修课程中如何融入并行计算有关内容的讨论,可参见[7]。

同样,面向对象的概念、方法与技术也可融入程序设计、软件工程、数据库原理、人工智能等课程中,有关这方面较详细的讨论,请参见文[8]。

## 4 实施问题

要具体实施上述构想,主要需解决两个方面的问题:一是配套的教材,二是教师。目前,虽已有一些教材将并行计算或面向对象作为原教材扩充的章节,但显然不能满足本文所提出的将这两个概念融入整个计算机科学本科课程的要求。同时,也有为数不少的教师本身,对这两方面的论题不甚了解。鉴于此,建议分三步来逐步实施上述构想。

首先,在高年级开设“并行处理”、“面向对象技术”这样的专题选修课程。通过讲授这样的课程可使教师逐步熟悉相关的内容,并总结出有效的教学方法。

(下转第64页)