

计算机科学与技术方法论^{*}

Methodology of Computer Science and Technology

董荣胜 古天龙 蔡国永 谢春光

(桂林电子工业学院计算机系 桂林 541004)

Abstract In this paper, two documents of "Computing as a discipline" and "Computing Curricula 1991" are briefly introduced, and their main contributions to the methodology of computer science and technology are pointed out. Then based on the general methodology of science and technology, the definition of computer science and technology is given, and its three paradigms (abstraction, theory, design), fundamental problem, core concept, mathematical methods and system approaches are presented. Finally, we conclude that the methodology of computer science and technology is a new theory in the computing cognition field.

Keywords Methodology, Paradigms, Fundamental problem, Core concept, Mathematical method, Systematic approach

1 引言

今天,当我们步入新世纪的时候,计算(Computing)技术作为现代技术的标志,已成为世界许多经济增长的主要动力。计算领域也已成为一个极其活跃的领域,计算学科正以令人惊异的速度发展,并大大延伸到传统的计算机科学的边界之外,成为一门范围极为宽广的学科。如何理解这个学科,引发了长期以来激烈的争论。

为了解决这类争论,学术界进行了大量的工作。1984年7月,美国计算机科学与工程博士单位评审部的领导们,在犹他州召开的会议上对计算教学问题进行了讨论,表面上看这类问题是关于计算教学问题,实际上是关于计算学科的认知问题。这一讨论以及其他类似讨论推动了ACM和IEEE/CS联手组成攻关组并用新的思维方式来理解计算学科。经过近5年的工作,ACM攻关组提交了"Computing as a discipline"的报告,该报告的主要内容刊登在1989年1月的《Communications of the ACM》杂志上^[1]。

ACM攻关组提交的报告得到了ACM教育委员会的认可并批准发行,该报告的重要贡献不仅在于它第一次给出了计算学科一个透彻的定义,回答了计算学科中长期以来一直争论的一些问题,更重要的在于,它找到了一个思考我们学科的"知识框架"(Intellectual framework),而非对策;一个指导方针,而非(具体)指示。对"知识框架"进行研究的思维方法就是报告中所指的"新的思想方法"。这个"新的思想方法"是对计算领域几十年来的概括和总结。从某种意义上说,该报告是计算领域认知过程中的一个里程碑,它为建立计算认知领域的理论体系奠定了基础。该报告所提出的"新的思想方法"其实质就是计算机科学与技术方法论的思想方法^[2]。

1990年,IEEE/CS和ACM联合攻关组提交了"Computing Curricula 1991"(以下简称"CC1991")报告^[3],报告将主领域增加到11个,并提取了学科中12个反复出现的核心概念。"CC1991"报告与"Computing as a discipline"报告(以下简称"两报告")密切相关,并使计算学科的方法论研究更加完备。

然而,正如"Computing as a discipline"报告所介绍的那

样:在学科的划分问题上,专家们颇费心力,最初专家们倾向于选择"模型"(Model)与"实现"(Implementation)相对,"算法"(Algorithm)与"机器"(Machine)相对,这两种方案无疑都可以反映计算学科各主领域研究的基本内容,但是这两个方案不是太抽象就是彼此的界限太模糊,大多数人无法很好地认同这种划分方法。另外,专家们认识到学科的基本原理已被纳入理论、抽象和设计三个过程中,学科的各分支领域正是通过这"三个过程"而实现它们的目标,于是,便选取了"三个过程"这个方案。由于"三个过程"研究的内容就是科学技术方法论研究的主要内容,而从分析该报告中提出的"三个过程"来看,专家们不是自觉地以一般科学技术方法论为指导来建立报告中所说的"新的思想方法"。因此,该报告不可避免地忽视了一些重要问题,如:缺乏对"三个过程"相互作用的透彻分析等,从而在一定程度上影响了人们对报告的理解。

正在起草之中的"CC2001"已注意到这个问题,并要求任务组对这三个过程(理论、抽象和设计)进行专门评述^[4,5]。不管评述结果怎样,学术界当前一个十分紧迫的任务就是要在前人大量工作的基础上,以一般科学技术方法论为指导完整地建立起计算学科自己的方法论,为理解学科提供方法,为执行新的教学计划做准备。

下面,我们先给出计算机科学与技术方法论的定义,并简要介绍计算学科二维定义矩阵的基本内容。

2 计算机科学与技术方法论的定义以及学科二维定义矩阵

由于"两报告"以及关于计算学科本身认知问题的相关工作^[7~13],使我们加深了对学科的认识,并使建立计算机科学与技术方法论的时机成熟。下面,我们就以一般科学技术方法论为指导建立计算学科自己的方法论。首先,我们给出计算机科学与技术方法论的定义:

计算机科学与技术方法论是关于计算领域认识和实践过程中一般方法及其性质、特点、内在联系和变化发展的系统研究。它是计算认知领域的一个理论体系,是我们认知计算技术的一个有效工具和方法。

^{*} 本文得到九五国防预研项目和广西科学基金项目的资助。

在计算领域中认识指的是抽象过程(感性认识)和理论过程(理性认识),实践指的是学科中的设计过程。抽象、理论和设计是具有方法论意义的三个过程,这三个过程是科学技术方法论中最重要的研究内容。从认识论的角度来看,学科的基本内容包含在抽象、理论和设计三个过程(或形态)之中,不仅如此,也正是因为它们之间的相互作用,促进了学科的发展。

显然,“Computing as a discipline”报告遵循了认识论的思想,给出了计算学科二维定义矩阵(以下简称定义矩阵)的概念并细化了其内容。定义矩阵的一维是“三个过程”(抽象、理论和设计),另一维是主领域。特别当主领域仅为计算学科时,定义矩阵便是本文所指的“知识框架”。该“知识框架”反映了计算领域中人们的认识是从感性认识(抽象)到理性认识(理论),再由理性认识(理论)回到实践(设计)中来的辩证唯物主义思想。在这里,这个“知识框架”也称作学科的知识内涵,它是稳定的;“知识框架”的内容(值)指的是学科的外延,它包括主领域以及各主领域“三个过程”的内容,这些内容(值)将随着计算技术的发展而变化。就“两报告”而言,ACM和IEEE/CS攻关组对学科本质的研究采用了内涵与外延相结合的方法。

由于计算学科二维定义矩阵(“知识框架”)是对计算学科的高度概括,因此我们可以将把握计算学科的本质问题归约为把握学科二维定义矩阵的本质问题。

要把握定义矩阵的本质,就是要分别把握定义矩阵的“横向”(抽象、理论和设计三个过程)以及“纵向”(各主领域)共有的、能反映各主领域内在联系的思想和方法的本质。

“横向”关系的研究,即抽象、理论和设计三个过程的内在联系与发展规律的研究,是计算机科学与技术方法论研究中最重要内容,因为学科的基本原理不仅已被纳入理论、抽象和设计三个过程中,更重要的在于,“三个过程”的相互作用,推动了计算学科及其分支领域的发展。

“纵向”关系的研究,即各主领域之间所共有的思想和方法的研究,将揭示各主领域的内在联系与发展规律,是计算机科学与技术方法论中的主要研究内容之一。

以下我们将分别对定义矩阵的“横向”关系和“纵向”关系进行研究。

3 计算学科中的抽象、理论和设计及其内在联系

方法论在层次上有哲学方法论、一般科学技术方法论、具体科学技术方法论之分,它们相互依存、互为作用。在一般科学技术方法论中,抽象、理论和设计是其研究的主要内容。以下我们以一般科学技术方法论为指导,阐述计算学科中的抽象、理论和设计内容。

3.1 学科的抽象形态

在科学技术方法论中,科学抽象是指在思维中对同类事物去除其现象的、次要的方面,抽取其共同的、主要的方面,从而做到从个别中把握一般,从现象中把握本质的认知过程和思维方法。科学抽象的成果有:科学概念、科学符号、思想模型等内容^[6]。

“Computing as a discipline”报告认为:理论、抽象和设计是我们从事本领域工作的三种主要形态(Paradigm),或称文化方式,它提供了我们定义学科的条件。第一个学科形态是抽象(按方法论研究的习惯,我们将报告中理论和抽象原来的次序对调),抽象源于实验科学。按客观现象的研究过程,抽象形态包括以下四个步骤的内容:形成假设;建造模型并做出预

测;设计实验并收集数据;对结果进行分析。

3.2 学科的理论形态

科学认识由感性阶段上升为理性阶段,就形成了科学理论,科学理论是经过实践检验的系统化了的科学知识体系,它是由科学概念、科学原理以及对这些概念、原理的理论论证所组成的体系。

在计算学科中,第二种学科形态是理论,理论源于数学。按统一的合理的理论发展过程,理论形态包含以下四个步骤的内容:表述研究对象的特征(定义和公理);假设对象之间的基本性质和对象之间可能存在的关系(定理);确定这些关系是否为真(证明);结论和解释。

3.3 学科的设计形态

在计算学科中,第三个形态是设计,设计源于工程。按为解决某个问题而实现系统或装置的过程来看,设计形态包含以下四个步骤的内容:需求分析;建立规格说明;设计并实现该系统;对系统进行测试与分析。

设计、抽象和理论三个形态针对具体的研究领域均起作用,在具体研究中,就是要在其理论的指导下,运用其抽象工具进行各种设计工作,最终成果将是计算机的软硬件系统及其相关资料(如,需求说明、规格说明和设计实现方法说明等)。

设计形态(技术方法)和抽象、理论二个形态(科学方法)具有许多共同的方面。这是因为,设计作为变革、控制和利用自然界的手段,必须以对自然规律的认识为前提(可以是科学形态的认识,也可以是经验形态的认识);设计要达到变革、控制和利用自然界的目,必须创造出相应的人工系统和人工条件,还必须认识自然规律在这些人工系统中的人工条件下的具体表现形式。所以,科学认识方法(抽象、理论二个形态),对具有设计形态的技术研究和技术开发是有作用的。但是设计形态毕竟还有其不同于抽象形态和理论形态的特点。其中最主要的是设计形态有更强的实践性;其次,设计形态具有更强的社会性;第三,设计形态具有更强的综合性。

3.4 三个学科形态的内在联系

理论是数学的根本;应用数学家们认为,科学的进展都是建立在数学基础上的。抽象(建模)是自然科学的根本;科学家们认为,科学的进展过程主要是通过形成假说,然后系统地按照建模过程对假说进行验证和确认取得的。类似地,设计是工程的根本;工程师们认为,工程的进展主要是通过提出问题,并系统地按照设计过程,用建立模型而加以解决的。许多有关数学、科学和工程相对优劣的争论都隐含了基于抽象、理论和设计三个过程中某一个更为更根本的假设。

“Computing as a discipline”报告认为,更详细的研究揭示出:在计算学科中,“三个过程”是错综复杂地缠绕在一起的,以至于把任何一个作为根本都是不合理的。

当专家们把注意力集中于“三个过程”谁更为根本的时候,学科方法论中的关键问题——抽象、理论和设计三个过程的相互作用被忽视了。不仅如此,在一定程度上还转移了我们的视线,削弱了我们对报告本质的理解,以致“CC2001”任务组不得不承认,与报告密切相关的“CC1991”计划的执行并没有达到预期的效果。

“Computing as a discipline”报告的实质是学科方法论的思想,其关键问题是一抽象、理论和设计三个过程的相互作用的问题,正是因为它们之间的相互作用,促进了学科的发展。文[2]介绍了这方面的有关内容。

4 计算学科各主领域之间的内在联系

定义矩阵中的“纵向”关系即各主领域之间存在以下内在联系:

1) 各主领域中的某些研究内容是一致的,比如数据库中的并发控制、缓冲区管理的思想与操作系统中的并发控制和缓冲区管理的思想是一致的;

2) 学科中具有方法论性质的核心概念贯穿于各主领域之中,揭示了学科各主领域的内在联系,使计算学科各主领域合成一个完整的体系而不是一些互不相关的领域。

4.1 学科中的核心概念

“CC1991”报告的一个重要贡献在于提取了具有方法论性质的12个反复出现的核心概念,并认为掌握和应用这些概念是成熟的计算机科学家和工程师的标志之一。

核心概念是学科中具有普遍性、持久性的重要思想、原则和方法,它具有以下特征:在本学科中多处出现;在各主领域及抽象、理论和设计的各个层面上都有很多示例;在技术上有高度的独立性;一般都在数学、科学和工程中出现。

三个学科形态(理论、抽象和设计)与核心概念揭示了计算学科的基本规律。随着研究的深入,并受一般科学技术方法论的启发,使我们能以更宽广的视野来认识计算学科。我们认为,在计算机科学与技术方法论的研究中,还应包括计算学科中的科学问题、数学方法以及系统研究方法等内容。学科中的科学问题、数学方法以及系统研究方法不仅反映了学科各主领域之间的内在联系,同时也有助于加深我们对学科本质的理解,并促使我们更自觉地运用这些思想和方法来从事计算领域的研究。

4.2 学科中的科学问题

科学研究过程是不断提出问题和解决问题的过程,发现和选定问题是科学研究的起点。

提出问题和解决问题的过程与抽象、理论和设计三个过程在本质上是统一的,虽然视角不同,但同样具有方法论的性质,并有助于我们从另一个视角来认识学科。以下我们介绍这方面的内容。

科学问题是指一定时期的科学认识主体在当时的知识背景下提出的关于科学认识和科学实践中需要解决而未解决的矛盾,它包含着一定的求解目标和应答域,但无确定的答案。

科学问题根据角度的不同有不同的分类方法。如:按学科的性质可以将问题分为基础理论问题和应用研究问题;根据问题在整个所要达到的目标中的地位可以分为关键问题和一般问题等。

科学问题有很多,其中有一种是反映学科本质,并在其发展过程中始终追求解决的问题,我们把这种问题定义为学科的根本问题(或本质问题)。学科各主领域正是围绕着学科的根本问题而逐步发展起来的,学科的根本问题有助于我们对学科本质的理解。本节以学科的根本问题为主,兼顾与其相关的内容进行分析。

人们对计算本质的真正认识来源于数学的形式化进程。1930年9月哥德尔(K. Gödel)发表的关于形式系统的不完全性定理宣告了著名的希尔伯特(D. Hilbert)纲领的失败,希尔伯特纲领的失败同时也暴露了形式系统的局限性,它表明形式系统不能穷尽全部数学命题,任何形式系统中都存在着该系统所不能判定其真假的命题。这就启发人们,应避免花费大量的精力去证明那些不能判定的问题,而应把精力集中于解

决具有能行性的问题。

在数学形式化的影响下,并通过对人的计算过程的哲学分析,1936年图灵(A. M. Turing)描述了计算一个数的过程,得出后来以他名字命名的通用计算机概念。根据图灵的分析,直观地说所谓计算就是计算者(人或机器)对一条两端可无限延长的纸带上的一串0和1执行指令,一步一步地改变纸带上的0或1,经过有限步骤,最后得到一个满足预先规定的符号串的变换过程。图灵用形式化方法成功地表述了计算这一过程的本质。图灵的描述是关于数值计算的,不过,我们知道英文字母表的字母以及汉字均可以用数来表示,因此图灵机同样可以处理非数值计算。

图灵提出的基本论点是:可计算性=图灵可计算性。在关于可计算性问题的讨论时,不可避免地要提到一个与计算具有同等地位和意义的基本概念,那就是算法。算法也称为能行方法或能行过程,它是对计算过程的一种刻画,是计算过程的一种能行实现方式。根据图灵的论点,可以得到这样的结论:任一过程是能行的(能够具体表现在一个算法中),当且仅当它能够被一台图灵机实现。图灵机与当时提出的用于解决可计算问题的递归函数、 λ 演算和POST规范系统等计算模型在计算能力上是等价的。在这一事实的基础上,形成了现在著名的丘奇-图灵论题。

图灵机等计算模型均是用来解决“能行计算”问题的,理论上的能行性隐含着计算模型的正确性,而实际实现中的能行性还包含时间与空间的有效性。

在图灵机这个思想模型提出不到10年的时间里,来源于图灵机思想的第一台电子计算机诞生了。

其实,图灵机反映的是一种具有能行性的用数学方法精确定义的计算模型,而现代计算机正是这种模型的具体实现。正如前面所提到的那样,计算学科各主领域均可以用模型与实现来描述。而模型反映的是学科的抽象和理论两个过程,实现反映的是学科的设计过程,模型与实现已蕴含于学科的“三个过程”之中。计算学科各主领域中的抽象和理论二个过程关心的是解决具有能行性和有效性的模型问题,设计过程关心的是模型的具体实现问题。正因为如此,学科中的三个形态是不可分割的、密切相关的。

在进行大量分析的基础上,“Computing as a discipline”报告对计算机学科是“工科”还是“理科”的问题以及学科的定义和学科的根本问题给出如下结论:

计算机科学和计算机工程之间本质上没有区别,两者是一回事,计算机科学注重理论和抽象,计算机工程注重抽象和设计,计算机科学和工程则居中。在统一认识之后,专家组将计算机科学、计算机工程、计算机科学和工程、计算机信息学以及其它类似名称的专业及其研究范畴统称为计算学科。

计算学科的定义:计算学科是对信息描述和变换的算法过程的系统研究,主要包括对其理论、分析、设计、效率、实现和应用等过程的研究。

计算学科的根本问题是:什么能被(有效地)自动进行,学科的根本问题讨论的是“能行性”的有关内容。而凡是与“能行性”有关的讨论,都是处理离散对象的。因为非离散对象,即所谓的连续对象,例如处处稠密的对象,是很难能行处理的^[14]。“能行性”这个学科的根本问题决定了计算机本身的结构和它处理的对象都是离散型的,甚至许多连续型的问题,也必须化为离散型问题以后,才能让计算机处理。例如,计算定积分就是把它变成离散量,再用分段求和的方法来处理。

正是源于学科的根本问题,以离散型变量为研究对象的高散数学对计算技术的发展起着十分重要的作用。同时,又因为计算技术的迅猛发展,使得离散数学越来越受到重视。为此,“CC2001”报告特意将它从“CC1991”报告中的预备知识中抽取出来,并列为学科的第一个主领域来认识,以强调计算学科对它的依赖性。

在弄清学科的根本问题后,我们可以对学科进行更好的理解。计算学科包括计算机和计算过程的理解、设计和使用。该学科的广泛性在下面一段来自 Computing Sciences Accreditation Board 发布的报告摘录中得到强调。

计算学科的范围包括从算法与可计算性研究到根据可计算硬件和软件的实际实现问题的研究。这样,计算学科不但包括从总体上对算法和信息处理过程的根本理解,也包括为满足给定的规格要求的有效而可靠的软硬件的实际设计……它包括所有科目的理论研究,实验方法和工程设计^[5]。

计算运用了科学和工程两者的方法学。理论工作已经大大促进了这门艺术的发展。同时,计算并没有把新的科学知识的发现与利用这些知识解决实际问题分割开来。理论和实践的密切联系给该学科带来了力量和生机。然而,同样也是理论与实践的联系,意味着与计算相联系的知识体(body of knowledge)随着技术的发展而有很快的变化^[5]。

正是由于计算学科理论与实践的密切联系,伴随着计算技术的飞速发展,计算学科现已成为一个极为宽广的学科。

尽管计算学科已成为一个极为宽广的学科,但其根本问题仍然是:什么能被(有效地)自动进行。甚至还可以更为直率地说,学科所有分支领域的根本任务就是进行计算,其实质就是字符串的变换。

4.3 学科中的数学方法和系统科学方法

文[11]对学科中一些具体的数学方法和系统科学方法做了归纳和整理,并将数学方法中的公理化方法,以递归、归纳和迭代技术为代表的构造性方法,内涵与外延的方法(内涵与外延虽是哲学中的两个基本概念,但它们在计算学科中,主要是以数学方式来表示的,因此,内涵与外延的方法属于学科中的数学方法)以及系统科学方法中的快速原型法、演化方法和展开与归约方法等统称为学科中的典型方法。文[11]的整理工作无疑是值得肯定的。但是,这种把学科中的数学方法和系统科学方法两种不同性质的方法合为典型方法的提法值得商榷。我们更赞成在学科中使用一般科学技术方法论中的数学方法和系统科学方法来阐述学科的有关内容,这样更有利于学科方法论的研究。

数学方法、系统科学方法与核心概念一样,反映了学科各主领域的内在联系,并具有一般方法论的性质。其中,数学方法主要出现在计算学科各领域的抽象和理论形态中,系统科学方法主要出现在抽象和设计形态中。

4.3.1 学科中的数学方法 数学方法是指以数学为工具进行科学研究的方法,即用数学语言表达事物的状态、关系和过程,经推导、演算和分析,以形成解释、判断和预言的方法。

数学方法具有以下基本特征:高度的抽象性;精确性;应用的普遍性。

数学方法在现代科学技术的发展中已经成为一种必不可少的认识手段,它在科学技术方法论中的作用主要表现在以下三个方面:为科学技术研究提供简洁精确的形式化语言;为科学技术研究提供数量分析和计算的方法;为科学技术研究

提供了逻辑推理的工具。

在计算学科中使用的数学方法有很多。一般认为数学中的公理化方法,以递归、归纳和迭代技术为代表的构造性方法,内涵与外延的方法,模型化与具体化的方法是计算学科中较有代表的数学方法。文[11]已对前三种方法进行了概括,本文仅给出模型化与具体化方法的思想。

模型化或模型方法是通过抽象、概括和一般化,把研究的对象或问题转化为本质(关系或结构)相同的另一对象或问题,从而加以解决的方法。模型化方法要求所建立的模型能真实反映所研究对象的整体结构、关系或某一过程、某一局部、某一侧面的本质特征和变化规律。

具体化是把抽象的概念、原理和规律体现于具体的对象或问题的方法。具体化有助于加深我们对概念、原理和规律的理解。

以上介绍的是数学中的模型化与具体化的思想,一般来说,数学模型是静态的,而计算模型则是一种过程性的模型。

在计算学科中,模型化反映了学科的抽象和理论两个形态,具体化反映了学科的设计形态。因此,同样可以用模型化与具体化(计算模型与实现)来反映计算学科各主领域研究的基本内容。然而,抽象、理论和设计三个学科形态较模型化与具体化而言更具有一般方法论的性质,也更易于理解。并且,从一般方法论的角度来看,抽象、理论和设计三个学科形态还蕴含了模型化与具体化的思想。因此,ACM的专家们最后选择了抽象、理论和设计三个学科形态来划分学科。

4.3.2 学科中的系统科学方法 系统科学方法,或简称系统方法,即按照系统科学的观点和理论,把研究对象视为系统来解决认识和实践中的各种问题的方法的总称。运用系统科学方法一般应遵循以下原则:整体性原则;动态原则;最优化原则;模型化原则。其中,前两个原则是基础,第三个是目标,最后一个手段。

系统方法的作用主要有以下三点:(1)系统方法是认识、调控、改造、创造复杂系统的有效手段;(2)系统方法为人们提供了制定系统最佳方案以实行优化组合和优化管理的手段;(3)系统科学方法为人们提供了新的思想模式。

系统科学方法种类很多,如系统分析法、黑箱方法、信息分析方法、功能模拟方法和整体优化方法等。在计算学科中最常用的方法是系统分析法。

系统分析法是为确定系统的组成、结构、功能、效用,而对系统各种要素、过程和关系进行考察的方法。计算学科中常用的结构化方法、原型法以及面向对象的方法均属于系统分析法。

计算学科中的系统科学方法揭示了计算学科各主领域的内在联系,有助于我们从总体上进行思维,实现复杂系统的设计。

数学方法和系统科学方法在科学技术方法论中占有重要的地位,如何更好地借鉴其它学科现有的科学方法,是值得我们学科思考的问题。

计算领域中的三个学科形态、科学问题、核心概念、数学方法以及系统科学方法反映了计算领域各主领域的变化发展规律以及各主领域之间的内在联系,构成了计算机科学与技术方法论的主要内容。

结束语 综上所述,由抽象、理论和设计三个过程以及学科各主领域组成的计算学科二维定义矩阵,它不仅隐含了学

(下转第12页)

于信息发布,其上的绝大多数内容都是只读的,而现在 WWW 逐渐成为一个网络计算的平台,“只读”的假设不再成立,此时保证分布式 Web 服务器系统中各成员服务器的一致性就要付出更大代价。又如,随着网络带宽的提高,WWW 上的大粒度对象(如视频对象)越来越多,它们必将对分布式 Web 服务器系统的请求分配、负载均衡和内容分布产生重大影响。这些都是迫切需要研究解决的问题。

参考文献

- 1 Iyengar A, et al. High-Performance Web Site Design Techniques. IEEE Computing, March-April 2000. 17~26
- 2 Cardellini V, Colajanni M, Yu P S. High Performance Web-server Systems. In: Proc. of 13th Int. Sym. On Computer and Information Sciences (ISCIS'98), Ankara, Turkey, Oct. 1998. 286~293
- 3 Egevang K, Francis P. The IP Network Address Translator (NAT), RFC1631, 1994
- 4 Aversa L, Bestavros A. Load Balancing a Cluster of Web Servers Using Distributed Packet Rewriting. In: Proc. of IPCCC'2000: The IEEE International Performance, Computing, and Communication Conference, Phoenix, AZ, Feb. 2000. 24~29
- 5 Kwan T T, McGrath R E, Reed D A. NCSA's World Wide Web Server: Design and Performance. Computer, 1995, 28(Nov): 68~74
- 6 Dias D, et al. A scalable and highly available web server. In: Proc. of the IEEE Computer Conference (COMPCON), Santa Clara, March 1996
- 7 Damani O P, et al. ONE-IP: Techniques for hosting a service on a cluster of machines. Computer Networks and ISDN Systems, 1997, 29: 1019~1027
- 8 Bestavros A, et al. Distributed Packet Rewriting and its Application to Scalable Server Architectures. In: Proc. of the Intl. Conf. on Network Protocols, Austin, Texas, USA, Oct. 1998
- 9 Fielding R, et al. Hypertext Transfer Protocol--HTTP/1.1, Network Working Group, RFC2616
- 10 Andresen D, et al. SWEb: Towards a Scalable World Wide Web Server on Multicomputers; [Technical Report TRCS95-17]. Department of Computer Science, University of California, Santa Barbara, 1995
- 11 Baker S, Moon B. Scalable Web Server Design for Distributed Data Management; [Technical Report 98-8]. Department of Computer Science, The University of Arizona, Aug. 1998
- 12 Shirazi B A, Hurson A R, Kavi K M. Scheduling and load balancing in parallel and distributed systems. IEEE Computer Society Press, Los Alamitos, Calif. 1995
- 13 Casavant T L, Kuhl J G. A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems. IEEE Transactions on Software Engineering, Feb. 1988. 141~154
- 14 See <http://www.squid-cache.org/>
- 15 Wessels D, Claffy K. Internet Cache Protocol (ICP) Version 2. RFC2186, Sep. 1997
- 16 Genova Z, Christensen K J. Challenges in URL Switching for Implementing Globally Distributed Web Sites. In: Proc of the Workshop on Scalable Web Services, Aug. 2000. 89~94
- 17 Mitzenmacher M. How Useful is Old Information. In: Proc. of the 16th Annual ACM Symposium on Principles of Distributed Computing, 1997. 83~91
- 18 Cardellini V, Colajanni M, Yu P S. Dynamic load balancing on Web-server systems. IEEE Internet Computing, 1999, 3(3): 28~39
- 19 Dahlin M. Interpreting Stale Load Information. In: Proc. of the 19th Intl. Conf. on Distributed Computing Systems, May 1999
- 20 Colajanni M, Yu P S, Cardellini V. Dynamic load balancing in geographically distributed heterogeneous Web-servers. In: Proc. of the 18th IEEE Intl. Conf. on Distributed Computing Systems, Amsterdam, The Netherlands, May 1998. 295~302

(上接第4页)

科中的科学问题、核心概念、数学方法以及系统科学方法,其实它还隐含了学科中一些有趣的问题。其中一个问题就是计算领域的工作者应该具备什么能力?报告指出有两类能力:

1. 面向学科的思维能力:发现本领域新的特性的能力。这些特性将导致新的活动方式和新的工具,以便使这些特性能被其他人所利用。

2. 使用工具的能力:使用本领域的工具有效地进行其他领域实践活动的能力。

报告建议:把面向学科的思维能力作为大学计算专业课程设计的主要目的。同时,计算专业工作者必须充分熟悉工具,以便与其他学科的人们有效地合作,进行那些学科的设计活动。

在面向学科的思维方式这类问题上,世界著名计算机科学家、图灵奖获得者 Dijkstra 教授在充分的论述后,告诫我们在计算科学的教学过程中不要用拟人化的术语,而是要用数学的形式化方法。我国学者赵致琢博士也提出了思维方式的数学化的思想。我们认为,面向学科的思维能力应包含两层意思:其一是面向学科方法论的思维能力;其二是面向学科的数学思维能力。

现在,计算学科有了自己的学科方法论,有了一个关于学科认知领域的理论体系。为此,我们向学术界推荐这种计算认知领域的理论体系,并认为,随着这种理论体系的不断完善,必将提供给人们一种完全超出自己粗放的想象的认识计算学科的能力。

参考文献

- 1 Denning P J, et al. Computing as a discipline. Communications of the ACM, 1989, 32(1): 9~23
- 2 董荣胜. 计算教育哲学初探. 计算机科学, 2000, 27(1): 93~97
- 3 Turner A J, et al. A Summary of the ACM/IEEE-CS Joint Curriculum Task Force Report; Computing curricula 1991. Communications of the ACM, 1991, 34(6): 68~84
- 4 IEEE-CS/ACM Joint Task Force. Computing curricula 2001 (draft). <http://www.computer.org/education/cc2001>
- 5 Chang C K. Curricula 2001: Bringing the Future to the Classroom. Computer, 1999, 32(9): 85~88
- 6 国家教委社会科学研究与艺术教育司. 自然辩证法概论(修订版). 高等教育出版社, 1991
- 7 Denning P J. A debate on teaching computing science. Communications of the ACM, 1989, 32(12): 1397~1397
- 8 Dijkstra E W. On the cruelty of really teaching computing science. Communications of the ACM, 1989, 32(12): 1397~1404
- 9 Scherlis W L, et al. Colleagues respond to dijkstra's comments. Communications of the ACM, 1989, 32(12): 1405~1413
- 10 Dijkstra E W. Dijkstra's reply to comments. Communications of the ACM, 1989, 32(12): 1414~1414
- 11 赵致琢. 关于计算机科学与技术认知问题的研究简报(I, II). 计算机研究与发展, 2001, 38(1): 1~15
- 12 王浩. 数理逻辑通俗讲话. 科学出版社, 1983
- 13 吴允曾选集. 北京科学技术出版社, 1991
- 14 莫绍揆. 递归论. 科学出版社, 1987
- 15 Computing Sciences Accreditation Board Inc. Annual Report, Stanford, Conn., 1998