

# 面向对象计算的现状和展望

米 澤 明 憲

这是译自面向对象计算的专辑的一篇领头文章。简略介绍了面向对象计算的背景和历史，以及一些争论的问题，阐述了面向对象的目标、要素和有关论点。

## 1. 引 言

六十年代后半期到七十年代初，在计算机科学的多个领域中，以不同形式相互独立而几乎同时产生了“面向对象”中所指的“对象”概念<sup>[1]</sup>。例如，

•K.Nygaard等人的模拟离散事件的语言 Simula,

•A.Kay的理想多介质个人计算机Dynabook及其用户语言Smalltalk (Smalltalk 80的前身),

•C.Hewitt等人的并行计算模型和知识程序设计的ACTOR形式;

•M.Minsky用于人工智能中知识表示的框架形式;

•J.Dennis等人用于操作系统中的资源保护的“可保持性”概念。

在研究和设计上述一些工作时，起核心作用的概念就是相当于目前叫做“对象”的概念。

面向对象具有这样广泛的背景和悠久的历史，因而进入八十年代后，它很快引起了人们的注意，下面作者就这一问题结合个人的见解，直观而非形式地叙述一下它的要点。标题中用了“面向对象的计算”而没有用“面向对象的程序设计”，原因是，基于“对象”的方法不只限于程序设计风格，而且正渗透到从计算机问题求解和信息处理的基础到应用的广泛领域之中。

## 2. 何谓面向对象

如何刻划“面向对象”，随着对这一方法

的着重点放在何处和应用领域的不同说法不一，其唯一的共同点也许都是都存在“对象”这一概念。所谓“对象”，最广泛的解释是将某一数据（组）和使用它的一组基本操作或过程汇集一起，而将此汇集体看作一个实体。因此，“面向对象”的基本想法就是把要构造的系统表示为对象集。

这里所说的系统，不仅考虑程序和软件系统，而且应广泛解释为计算模型、软件设计、模拟模型、CAD/CAM中的成分模型以及人工智能中的知识表示形式等。上述的“面向对象”的说法是作者的意见。在研究人员中，特别是在程序设计语言专家中，所谓“面向对象语言”用什么语言功能刻划，没有这些功能就不叫“面向对象语言”，这方面的争论是众说纷云的。作者认为这样的争论在积累软件新概念方面虽然很重要，但对构造一般软件，过于强调：“如果不用具有如此这般功能的面向对象的语言编写程序，就不能进行面向对象的程序设计”的说法是不可取的。

## 3. 面向对象的的目标

在构成系统的情况下，以前的办法是，当实体作为系统的成分时，往往把“数据”和“过程”这两种实体分开考虑。而在使用面向对象的方法中，正如第2节所述，不把“数据”和“过程”分开，而是将数据及操作这些数据的一组基本过程看作一个整体。

那么, 为什么会产生这样的想法呢? 这与如何解决一般构造计算机软件的两大问题有关。一个问题是“怎样克服复杂性的障碍”, 另一个问题是“怎样将现实世界模型在计算机中自然地表示出来”。而且, 这两个问题相互之间当然有密切的关系。

作为克服复杂性的手段, 在面向对象中, 可利用如下对象的性质: (1) 将密切相关的数据和过程汇集起来定义为一个实体, 而且 (2) 一旦定义出实体, 即使不知道怎样实现也能使用此实体的功能。(1), (2) 相当于软件工程和程序设计方法论中的抽象化, 抽象数据类型和信息隐蔽等概念所具有的性质。此外, 为把有关功能或知识及操作知识的过程汇集一起定义为一个实体所使用的框架形式也具有 (1), (2) 的性质。

在第1节中谈到, 用计算机模拟离散事件的高级语言Simula是使用面向对象这一概念的一个开端。为了得到现实世界行为的必要信息, 模拟是从计算机中用足够的精度表示出现实世界模型开始的。此时, 现实世界或必要的信息越复杂, 越有必要以自然形式将现实世界的结构模型化。这就是为什么在Simula中要考虑对象这一概念的原因。

另一方面, 人工智能的一个基本问题是: 如何使用最佳的形式在计算机中有效地表示人类具有的有关现实世界的各种知识。对此, 我们在认识世界时, 按照随意使用的某种划分法, 将实体或概念分段落, 又将这些段落各自的功能和性质分别表示为各个汇集体, 这也许是最合适的。这就是为什么提出使用框架形式的原因。这也成了最终要求对现实世界进行自然模型化的基础。

而且, 近几年来, 在数据库概念模型(数据模型)领域中, 也引进面向对象的各概念, 这方面的研究很活跃<sup>[4,5]</sup>。这大概是因为人们开始认识到了在以前的关系数据库形式下, 对现实世界的模型化不能做到自然的缘故。

由上面看出, “面向对象”的目标可归纳

如下。对试图利用计算机进行问题求解和信息处理的目标领域, 尽量使用对象概念, 以自然形式在计算机内进行模型化<sup>[6]</sup>, 由此所得的自然性就可克服软件系统的复杂性障碍, 从而得到问题求解和信息处理的更高的能力。

此处所谓的“自然性”, 意味着系统设计者从现实世界所得到的图象或设计者头脑中形成的模型里所出现的物理图象或概念图象与构成系统的一组对象之间有近乎一对一的对应关系。

#### 4. 面向对象的要素

本节稍为具体地叙述一下刻画面向对象的计算的一些要素。不过, 事先声明一句, 我们不推崇这样的狭隘观点: 如果不具备下述一切要素, 就不能叫做“面向对象”。同样, 对各要素也没有理由要它们各自都必须具有下面要说明的性质和功能。应该允许有各种各样的变通形式。

**对象** 一收到消息就被激活, 并完成由该消息所指定的任务之独立的过程实体。对象内部一般具有记忆, 记忆内容对应于对象的状态。因此, 状态随着记忆内容的变化而变化。

更具体地说, 如第2节中所谈到的那样, 对象是数据和对其进行操作的一组过程的汇集体。不过, 从上面的“收到”、“独立”等词所知, 收到送来的消息后, 是否开始完成指定的任务, 可由对象本身的判断, 即对象的当前状态和对象本身所具备的功能(或者所定义的一组过程)所决定。此外, 对象记忆内容的引用和更新只由对象本身进行(用自己定义的过程), 而其它对象不能随意从外部访问。

为定义某个对象的具体动作, 可以指定此对象在何种状态下收到哪种消息, 就起动哪种过程。同样, 也可以指定在对象具有内部状态的情况下, 状态可以用哪种要素来构成。随着“对象概念”被使用的目的不同, 描述对象动作的语言也有各种级别和种类。有

用某种代数的数学语言描述的，用设计软件的语言描述的，以及用可由计算机执行的程序设计语言描述的等等。

**对象的动作** 被激活了的对象通常可能执行哪种动作？首先必须把消息送给对象，作为和其它对象相互作用的唯一手段，这是必要的。对象的消息传送方式，在能同时激活多个对象时可以考虑多种情况<sup>[7]</sup>。而且，结构会发生动态变化的现实世界作为对象集合加以模型化时，一个对象能生成新对象的功能被认为是重要的。作为对象的基本功能通常事先假定具备条件转移、重复等普通命令型程序的控制结构，以及算术运算、表处理的基本运算等等。但是，如果开始假定有某种类型的基本对象存在，则这些控制结构和基本运算都可以通过这些对象间的消息传递和对象的动态生成组合所构成。

**消息** 消息含有激活对象、指定委托任务的标记。除此之外，消息还可以包含完成委托任务所必需的信息，例如运算需要的参数等等。特别是，对象名也可以包括在消息之中。

**类和实例** 通常把具有同样性质和功能的东西所构成的集合叫做类。在面向对象中，也可以把具有相同内部存储结构和相同一组过程的对象看成属于同一类。在指定一个类后，往往把属于此类的对象叫做它的实例。如果假定一对象具有能生成新对象的功能，则往往考虑这样的对象，该对象将具有生成属于某一类的实例之功能的对象对应于该类（或者将该类模型化）。

**继承性** 一般说来，上位概念具有的性质和功能可以由下位概念所继承。在面向对象中，属于下位类的对象当然具有刻划上位类特征的性质和功能。因此，定义下位类的对象时，如果指定上位类的名称就能自动继承该上位类（的对象）的性质和功能，则虽然之后没有该上位类的对象，但只要附带描述一下为定义下位类对象所需要的功能和性质就行了。允许这种对象（的类）的定义法

之机制叫做继承性。而且，将相互无上下关系的若干类指定为某类的上位类的机制叫做多重继承性。这种机制对于把现实世界中概念之间的上下关系作为对象类之间的关系而模型化，以及简明描述这种模型都是有效的。继承机制从模块重用观点看来也是重要的。

## 5. 面向对象的论点

现在可以从各种不同角度进行面向对象的研究和实践。例如，在面向对象的软件设计法、面向对象的数据库、面向对象的并行计算、程序设计教育和机器人学等方面的应用。下面，简单讨论一下这些方面的同时，提出了与其它面向对象的计算有关的几种论点。

**“面向对象设计法”** M. Jackson的系统开发（JSD）<sup>[9]</sup>作为一种方法也是基于面向对象的设计法。还报导了以ADA为基础来考虑的设计法和设计实例<sup>[9]</sup>。这些设计法虽然没有考虑继承性，但可以认为包含了这种意义的今后会大力提倡的所谓“设计法”。

**“面向对象的数据库”** 以扩大灵活性和表达能力为目的，在设计数据模型中，正在引进对象和类/实例的方法<sup>[10]</sup>。此外，人们开始研究作为访问面向对象数据库的查询语言，应具备哪些要求<sup>[11]</sup>。

**“面向对象的并行计算”**<sup>[14]</sup> 为克服并行计算和分散处理中的复杂性障碍，以及将现实世界本身所具有的并行性用模型自然地表示出来，研究了能同时被激活多个对象的假定。计算模型、描述语言、体系结构、程序设计环境及应用等的各种水平上的讨论很热烈，国际会议也在筹划之中（当年9月在美国圣地亚哥就要召开OOPSLA会议）。

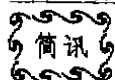
**“带强类型和无类型”** 在生成Lisp, Actor和Smalltalk系统的面向对象语言中，程序内所出现的名字等一般都不带类型。另一方面，Simula、CLU系统的语言却带强类型。带类型的方法是否好，在专家之间有不同意见。不过，一般说来，在质量要求高的

设计中，很多人编制一个大型软件时，用普类型的语言也许安全些。另一方面，用原型方法编制软件时在灵活性上，类型的限制也成了障碍。

“执行速度” 偶尔听到“用面向对象语言编写程序，在执行速度上有问题”这样的忧虑，不过此事没有必要太耽心。现在被称之为面向对象的语言已有多种，在执行速度上也有和普通命令式语言差不多的，也有慢几倍的。对执行慢的语言，正在认真研究其改进（请回忆一下以前执行速度慢的Lisp语言，其速度飞跃提高的历史事实）。而且，由采取面向对象的方法所得到的软件系统的设计质量水平上的提高，可以认为已抵消了目前速度上的问题。进一步提高所构成系统的速度也比较容易。

“约束程序设计” 在描述要解决的问题及其解法时，使用所出现的同类事物中必须成立的约束条件有时是方便的。在这种情况下，只用面向对象的方法来表示有时不自然。对此，把约束条件描述为对象的方法，也就是约束表示和面向对象的方法如何很好地融合一起的研究正在活跃起来<sup>[12]</sup>。

“有关面向对象的出版物” 本文中引用的论文和报导只是最低限度的，所以文末集中列出了比较容易得到的论文集、会议录之类的文献，请参见[13—20]。



## 计算机科学与逻辑学术讨论会胜利召开

由计算机学会直属计算机理论学组与软件分会程序理论学组举办的计算机科学与逻辑学术讨论会于1989年12月12日—15日在汕头大学与厦门大学举行。参加此次会议的代表有96名，在会上宣读与交流论文71篇。

此次会议是我国计算机科学工作者与数理逻辑工作者首次聚会共同探讨计算机科学中的逻辑问题的一次新的尝试。会上所交流的论文有人工智能理论、软件理论、NP完全问题以及数理逻辑领域中的一些问题。会议邀请了我国数理逻辑界老前辈莫绍揆教授、国防科技大学陈火旺教授以及其它一些知名计算机科学专家作了“论证式”、“计算机科学的发展对理论工作的要求”、“NP完全问题的研究动态”、“用逻辑方法提高数据库研究水平”、“λ-演算与组合逻辑”等专题报告。

与会代表经过讨论一致认为，为推进计算机科学的发展，迅速提高我国计算机科学工作者的理论水平特别是数理逻辑水平是极端必要的。同时也认为，数理逻辑学者要与计算机科学工作者结合为计算机科学发展提供理论支撑。

根据代表们的要求，会议决定委托南京大学徐洁磐教授筹建“计算机科学与逻辑”专业学组，以推进计算机科学与逻辑间的相互渗透与共同提高。

## 6. 结束语

回顾从六十年代末到七十年代，在“结构程序设计”的这一口号下，从理论研究到新语言设计，…，软件开发的组织和管理等，进行了各种各样的研究和实践。而且，在这一过程中，从乐于编制短程序到从事大型软件的开发的大程序员们，一致得出结论：“各种过程或例程都必须明确写出使用的算法及其表示，以便易于理解”。

面向对象是一种适用于直观的模型化或直接表示其模型的形式体系。八十年代到九十年代前半期，在面向对象这一口号下，必将进行各种研究和实践。为提高克服复杂性从而提高问题求解/信息处理的能力，有必要将现实世界进行自然模型化并将其直接表示出来，这一思想也许将渗透到设计和实现较大型的系统的人们中间。

今后可以预料；随着计算机硬件/体系结构的进步，能廉价地提供极大的计算能力和存储容量。这时，计算机系统有可能用来提供各应用领域中“现实世界”的广义模拟装置或假想的现实世界。因此，使此装置运行的软件就是表示了现实世界模型的程序。如果这一观点成立，也许要重新认识面向对象和面向并行对象的方法和见解的重要性。

〔闻 译自日刊《情报处理》1988, No.29, No4 土 口校〕

计算机理论与程序理论学组供稿