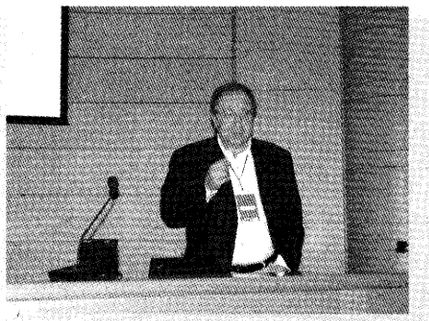


寻道

当IT爆发式膨胀的同时,有一些人曾默默地奉献,他们可能开创了一个时代,他们的作品可能是IT发展中的一个里程碑……

程序基于精确的数学
——STL之父Alex Stepanov访谈录

●特约记者 赵玉勇

了解Alex和STL

Alex Stepanov,STL(标准模板库)之父,并因此而荣获第一届Dr. Dobb's 程序设计杰出奖,现在是Adobe公司首席科学家。他曾是康柏电脑公司的副总裁和首席科学家,AT&T实验室副总裁和首席架构师,SGI服务和超级计算机业务首席技术官。

背景:Dr. Dobb's程序设计杰出奖

从1991年起,每年《Dr. Dobb's Journal》都会将荣誉给予那些对软件开发技术的发展做出了重要贡献的人。这就是著名的Dr. Dobb's 程序设计杰出奖(Excellence in Programming Award)。

什么是STL呢?STL就是Standard Template Library(标准模板库)的简称,它是由Alex Stepanov和Meng Lee在惠普实验室工作时开发出来的。现在虽说它主要出现在C++中,但在被引入C++之前,Alex和David Musser已对该技术进行了很长一段时间的研究。

STL是多年来编程应用中常用组件的集合,STL的贡献是将各组件和接口泛化及标准化。这样可以极大地提高编程效率。STL现在是C++的一部分,任何C++平台都包含着它。

中国·数学·未来

对于STL之父Stepanov先生来说,他更像一位数学家。和在上个月我们采访的C++之父一起,这次他是第一次来到中国,而他对古代中国和中国数学家的了解,更是让人佩服。

Stepanov先生告诉记者,计算机科学是建立在数学之上的精确的学科,他说:"程序设计就像同未理顺的复杂性问题的一场战斗,既然要进行这场战斗,首先需要运用数学这门工具,几个世纪以来,数学的作用正在于此。如果将现在生动的数学体系作为实验证据,对于解决人类遇到的复杂性问题,数学还是最有效的。"

当我们问他对中国和中国的程序员的认识时,他的回答还是和数学相关。他告诉记者:中国是一个伟大的国家。曾经有许多伟大的数学家:秦九韶的《数书九章》就是古代数学中的经典;《孙子兵法》中已包含现代西方称之为中国余数定理的内容。现代中国也产生过许多真正的伟大的数学家,如对哥德巴赫猜想做出杰出贡献的陈景润先生。

Stepanov先生的编程信条是:程序设计是基于精确的数学训练的。他的建议更是浅显而简洁:好好学数学,好好学计算机,好好学英语。

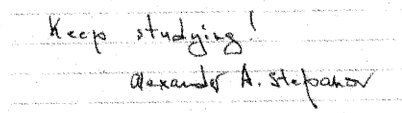
谈到未来时,他虽然并没有明确回答什么,但对美好未来的憧憬,还是和大家的期待一样。关于将来要用的计算机语言,他也有自己的答案。他说:C++和Java会在近几年得到最广泛的应用,我更期望最终会有某些新的、更完美的语言出现。

大师和成长历程

1950年11月16日,Stepanov先生生于苏联莫斯科。他曾在莫斯科大学研究数学,但未成为一名数学家。因为他实在不能对Tamagawa算术、Coxeter群等一些纯数学的东西感兴趣。Stepanov先生的想法很单纯,他要脚踏实地地干事。对他来说最幸运的事情是,他能够看到很多伟大的数学家是如何做学问的,也就使他更清楚地看清计算机科学中一些司空见惯的伪数学。因此,能成为一名程序员对他和计算机科学来说真是一件好事。

另两位大师对他的影响是显而易见的,一位是计算机程序设计艺术教授高德纳(Donald Knuth)先生,另一位则是计算机科学大师Edsger Dijkstra。他深情地说:"前者告诉我答案,后者则引导我深思。"而他对高德纳先生的《计算机程序设计艺术》一书的推崇,也使我们找到了提高自己编程水平的杀手锏。

1984年,他成为纽约布鲁克林理工大学助理教授。Stepanov告诉记者:"教授计算机科学使我受益匪浅,我要对付各种研究生课程。在此过程中,我学到了很多新东西。我还用Scheme语言开发了一个巨大的数据结构和算法库,这项工作导致了Ada泛型库的诞生(这是和David Musser合作的)。"在贝尔实验室短暂地研究了一段时间,Stepanov设计了一个新的C++算法库,在1998年他又去了位于Palo Alto的惠普实验室。在那儿,他先花了四年时间研究存储系统。1993年,Stepanov得到了一个回头研究泛型编程的机会。而STL就是这次研究的结果。1995年他又到了Silicon Graphics,在此,组建了一个小组继续进行STL的开发工作。"



Alex和中国程序员们共勉:继续学习!

Stepanov现在是Adobe公司的首席科学家。Adobe是一家生产诸如Acrobat和Photoshop之类桌面软件的公司。

高德纳先生的《计算机程序设计艺术》一书,正是Alex Stepanov先生极力推荐的一部作品,不论电邮中还是面对面时,他反复强调这部著作是一个珍宝库,想要什么,里面便有什么。而他在和这本书打交道的三十多年中,从中受益无穷。从他对高德纳先生《计算机程序设计艺术》的极力推崇上,也对他的个性有所印证。

STL故事

STL是Stepanov先生一生中浓墨重彩的一笔,而这一笔在他的描述中却是那样地不经意。STL也是他个性的重要体现,独立思考,缜密逻辑。要了解STL之父,首先要知晓什么是STL。

有一位意大利记者曾向Stepanov先生问过此问题:STL究竟是指Standard Template Library(标准模板库)还是 Stepanov and Lee?

Stepanov先生笑着解释他的玩笑:"哦,它真的是指Standard Template Library。我曾经在Dr. Dobb的杂志做的那个专访里开玩笑说,STL是指'Stepanov and Lee',但它只是个玩笑而已。"

而STL实质上包含了二者的意思。Meng Lee是他的一位无可挑剔的合作伙伴,她使Stepanov先生更专注,她在代码和文档上花了大量的令人精疲力尽的时间。正是由于像Lee这样的合作者,使得STL广为世人传播。

STL代表什么呢?Stepanov and Lee,这不完全是个玩笑,Meng Lee架起了我们与Stepanov先生沟通的又一座桥梁。这篇采访中浸透着她的汗水……

G13版我们的记者直接对话STL之父Alex Stepanov。

寻迹

编者按:在开发业界,每一个时期总有一些代表的人物,代表的技术。STL就是这样的一门技术,它是那样深刻地影响了一个时代的C++开发。和一群聪慧的C++程序员谈STL,就如同和资深的会计师谈珠算一般。

今天我们就有幸请到了STL之父Alex Stepanov,为我们溯源STL,你甚至还能读到对OO与众不同的理解等内容,希望读者能有所收获。

对话STL之父Alex Stepanov

当我面对面见到Alex Stepanov先生时,和像片中的感觉完全不同。从像片上看他,我们不难对他有威严的感觉。但实际上那并非生活中的他——幽默风趣、平易近人才是真正的他。

Alex先生和C++之父Bjarne先生是很好的朋友,Bjarne的推荐对我们的采访起了重要的推动作用,他告诉我:Alex先生经常有一些有趣的和重要的东西要说。

Alex先生有着非常强烈的个人魅力,他的言辞尖锐,甚至让人想到"猛烈""激进"等字眼,这可能也正是他对计算机内涵的深刻体会和精深的数学涵养所致——他的眼里容不得半粒沙子。

他给我们的忠告很简洁:"好好学数学,好好学计算机,好好学英语。"

溯源STL

追根溯源,STL起源于什么?意大利记者的提问或许对我们有所帮助。

问:STL一开始被设想成今天这个样子吗?即所谓的C++标准库,或者,它是从别的什么项目发展变化来的?告诉我们一些关于STL的历史好吗?

Alex:1976年,又要说到原苏联了。我因为吃生鱼片严重食物中毒而住院,在精神恍惚中,我忽然意识到并发的加法计算能力是基于加法是结合性的[译注:比如说a+b+c+d=(a+b)+(c+d)]。因此,STL可以说是细菌传染的结果。同时,我意识到并发的减法运算是和半群结构类型有关联的,这就是最基本的重点:算法是定义于代数结构基础之上的。我又花了一些年头,意识到必须在正规公理上加入复杂性必要条件以扩展结构的概念,接着又花了15年之久才完成全面的架构(我直到现在都不能确定我是否成功地让我朋友小圈子之外的任何人理解了这一点)。我相信迭代器理论是计算科学的中心,就像环或Banach区间理论是数学的中心一样。

每次当我找到一个算法时,我都要努力去寻求它所定义的结构基础。我想做的就是泛化地描述算法,并乐此不疲。我可以花一个月时间去精确地描述一个众所周知的算法的泛化表示。迄今为止,在向人们解释我这种行为的重要性方面,我是异乎寻常的失败。然而,不知何故,这种行为的结果—STL却是如此成功。

关于STL还有很多故事:STL如何成为C++标准的呢?Alex先生有下面的表述,表述中也有他对C++之父Bjarne先生的精彩描述。

问:有一件事情我一直都很惊奇——C++标准委员会那么快就采纳了STL?我的意思是,这些委员会成员可都是以谨慎和保守而出名的。这一点,你怎么解释?

Alex: Bjarne的支持是至关重要的。如果说Bjarne想要什么东西的话,那就是他真的想把STL弄到标准里,他办到了。他像骡子一样固执。甚至逼着我去改STL——我从来都不会为第二个人这么做——我也是个顽固分子。但他是我所认识的最有主见的人,他花了一些时间去理解STL是干啥的,当他理解之后,他决定使它成功地被大家接受。

他对STL的贡献还在于他支持"不止一种编程方法是'合理的'"的观点——这对立于十来年认同"唯一"方法的无休止的争执和夸大——而坚持把弹性、效率、重载、类型安全结合在模板里以至于使STL成为可能。我很乐意明白地声明Bjarne是我这一代人里卓越的语言设计家。

问:您是由于什么原因开发了STL?开发的过程是什么样子的?它对您的生活和研究有什么影响吗?

Alex:STL的开发源于我多年对程序灵活性和参数化探索的结果。我对此仍不太满意:因为C++语言上的一些缺陷,使得我不能完全表达我要表达的意思。STL应当做将来要设计的库的草稿。我不知道在我有生之年是否能看得到,但我坚信最终会有一个关于算法和数据结构的标准架构,说不定作为读者的你就是这一发明人呢!

我觉得很难形容STL对我生活和研究的影响。STL并未从本质上改变我的生活,因为我没有通过它赚到任何钱。我想向我的几个朋友展示一下正确的编程方法,而STL正是为他们而做的。事实上将这项工作完成带给我很多快乐,当我听说它帮助某人解决了问题时,我更加快乐。当然也有人说如果让他们设计,他们可以设计出更好的东西来,也有人对STL嗤之以鼻,这些攻击也确实伤害到了我。

OO与编程

和STL形成对照的是OO——面向对象。OO的思想困扰着许多程序员,OO的思想一直困扰着我,从未有人对OO表达过如此强烈的想法——除了Alex先生。我想:作为一个程序员,这是一定要掌握的,但是……,请看下面Alex先生的观点:

问:您对面向对象是怎样理解的?它是不是一种好的可接受的编程思考方式?有没有学习OO必须的有用的工具?

Alex:我尽量避免用OO思考问题,我对他们编程的方法不感冒。在意大利的一家期刊采访我时我曾说过:"我发现OOP在技术上是有问题的,它妄图用基于单一类型的不同接口来分解世界,为了处理不同的实际问题,你需要不同种类的代数方法以横跨不同类型的接口族;我发现OOP在思想上是不健全的,它声称一切都是一个对象。

即使真的是这样,也没什么意思——说一切都是对象跟什么都没说一样;我发现OOP的方法论是错误的。它从类开始,就好像数学要从公理开始一样。你不是从公理开始——你是从证明开始的。直到你找到了一大堆相关证据你才能归纳出公理,以公理结束。编程上存在着同样的事实:你要从有趣的算法开始。只有很好地理解了算法,你才有可能提出合理的接口让其他组件共同工作。"我再重复强调一点:程序是描述算法和数据结构的,而不是描述继承性和多态性的。

问:您认为编程的好方法是什么?对于编程来说,一种工具是不是必需的?

Alex:我认为学习多种不同的编程语言是非常重要的。我用过Algol-60, Common Lisp, Scheme, Ada, C, C++, Java, 和多种汇编语言。然而,也不能仅仅局限于程序语言,它仅仅是种表达算法和数据结构的工具——并且是种有缺陷的工具。如Niklaus Wirth有句精辟的见解:程序=算法+数据结构。

问:您认为计算机语言和人类的语言有什么区别?

Alex:没有人尝试过用计算机语言写出诗歌来。计算机语言发展到能允许我们解决一些真正美好的现实生活中的东西,还有很长的路要走。

提示:如想了解STL之父更多,请看本期G1版。