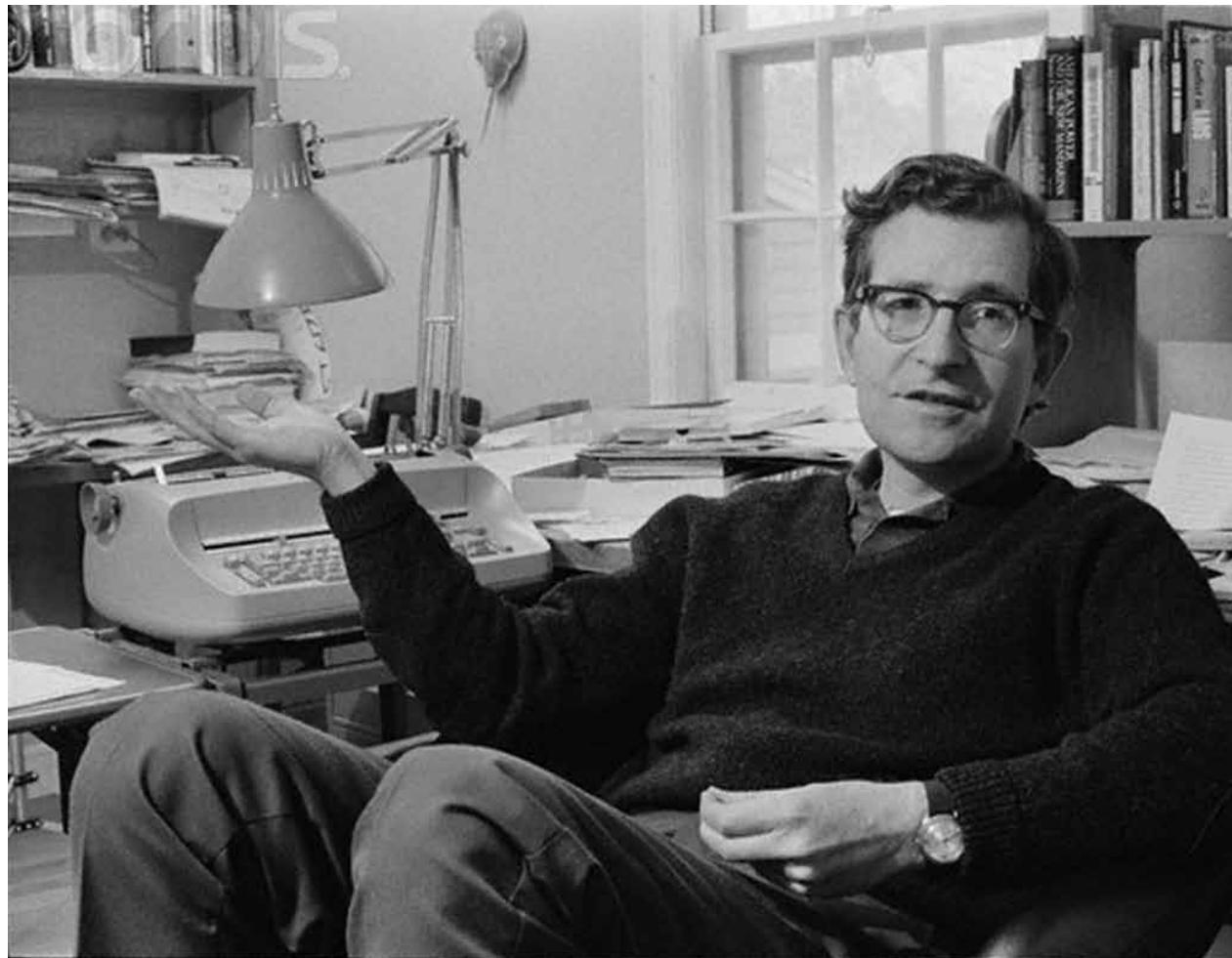


计算科学评论

环球科学

WWW.COMPUTER.ORG

计算历史特刊



《机器、语言和计算》在MIT的诞生历史

合作机构



IEEE
computer
society



ISSN 1673-5153



9 771673 515122

定价：25元

云计算

人工智能

工控机

制造业

电子

嵌入计算



传感器

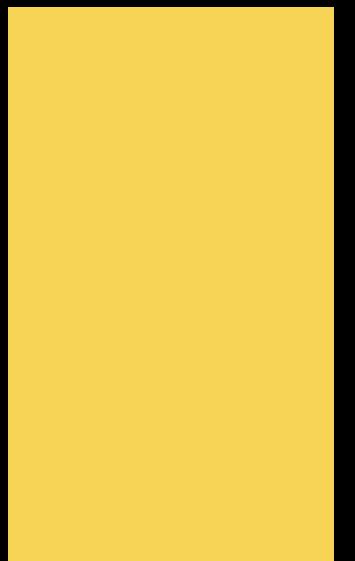
互联网

3D 打印

绿色计算

图形图像

虚拟现实



互联网金融

芯片

大数据

软件架构

纳米架构

人机交互

多媒体

普适计算 MEMS



微信名：计算机人 微信号：jisuanren



Copyright

版权

主管单位 Authorite in Charge

中华人民共和国教育部 Ministry of Education of the People's Republic of China

主办单位 Sponsor

中国大学出版社协会 China University Presses Association

出版单位 Publisher

《环球科学》杂志社有限公司 GLOBAL SCIENCE MAGAZINES Co. Ltd
社址 Address: 北京市朝阳区秀水街1号建外外交公寓4-1-21 Office 4-1-21, Jianguomen Diplomatic Residence Compound, No. 1, Xiu Shui Street, Chaoyang District, Beijing, China. 邮编: 100600
联系电话: 010-85325810 / 85325871

社长 / 总编辑 Editor-in-chief

陈宗周 Chen Zongzhou

副社长 / 副总编辑 Deputy Editor-in-chief

刘芳 Liu Fang

执行出版人 Publisher

管心宇 Xinyu Guan

资深编辑 Senior Editor

马法达 Falda Ma

刘妍 Yan Liu

特约编辑 Contributin Editor

史彦诚 Yancheng Shi

刘大明 Daming Liu

高天羽 Tianyu Gao

费蛹 Yong Fei

王璇 Xuan Wang

运营中心 OPERATION DEPARTMENT

运营机构 Publisher

上海灵宸文化传媒有限公司

发行部 Circulation Department

发行总监 Circulation Director

谢磊 XieLei 010-57439192

市场部 Marketing Department

市场总监 Marketing Director

孔祥彬 Kong Xiangbin 010-85325810-807

广告部 Advertising Department

销售总监 Sales Director

范欢 FanHuan 010-85325871-802 010-85325981

读者服务部 Reader Service

杜珺 Du Jun 010-57458922

印刷: 北京博海升彩色印刷有限公司

如发现本刊缺页、装订错误和损坏等质量问题, 请在当月与本刊读者服务部联系调换(请将坏书寄回)。

国际标准刊号: ISSN 1673-5153

国内统一刊号: CN11-5480/N

广告经营许可证号: 京朝工商广字第8144号

知识产权声明:

IEEE, IEEE Computer, IEEE中文网站的名称和标识, 属于位于美国纽约的电气电子工程师学会有有限责任公司所有的商标, 仅通过授权使用。这些材料的一部分由IEEE Computer英文版翻译而来, 版权归IEEE所有, 并经IEEE授权翻译复制。

IEEE Computer杂志的中文版权, 由美国电气电子工程师学会有有限责任公司授予上海灵宸文化传媒有限公司, 并由本刊独家使用。

本刊发表的所有文章内容由作者负责, 并不代表上海灵宸文化传媒有限公司、美国电气电子工程师学会有有限责任公司的立场。

本刊内容未经书面许可, 不得以任何形式转载或使用。

编辑团队

执行编辑

Carrie Clark

ccClark@computer.org

高级编辑

Chris Nelson

编辑

Lee Garber, Meghan O'Dell

特约编辑

Christine Anthony, Rebecca Torres

多媒体编辑

Brian Brannon

设计和产品

Monette Velasco, Lead

Jennie Zhu-Mai, Lead

Mark Bartosik

Larry Bauer

Erica Hardison

封面设计

Andrew Baker

平面设计

Hector Torres

高级商务拓展经理

Sandy Brown

高级广告经理

Marian Anderson Debbie Sir

产品和服务总监

Evan Buttereld

会员总监

Eric Berkowitz

编辑服务高级经理

Robin Baldwin Manager

编辑服务内容开发

Richard Park

主编

Sumi Helal

University of Florida

helal@csie.u.edu

副主编

Ying Dar Lin

National Chiao Tung University

ydlin@cs.nctu.edu.tw

副主编, COMPUTING PRACTICES

Rohit Kapur Synopsis

rohit.kapur@synopsys.com

副主编, PERSPECTIVES

Bob Colwell

bob.colwell@comcast.net

副主编, SPECIAL ISSUES

George K. Thiruvathukal

gkt@cs.luc.edu

副主编, MULTIMEDIA EDITOR

Charles R. Severance

University of Michigan

csev@umich.edu

2016年IEEE计算机协会主席

Roger U. Fujii

Fujii Systems

r.fujii@computer.org

顾问委员会

Doris L. Carver

Louisiana State University (EIC Emeritus)

Carl K. Chang

Iowa State University (EIC Emeritus)

Theresa-Marie Ryne

Consultant

Bill Schilit

Google

Savitha Srinivasan

IBM Almaden Research Center

Ron Vetter

University of North Carolina Wilmington (EIC Emeritus)

Alf Weaver

University of Virginia

行业编辑

大数据和数据分析

Naren Ramakrishnan

Virginia Tech Ravi Kumar Google

云计算

Schahram Dustdar

TU Wien

计算机架构

David H. Albonesi

Cornell University

Greg Byrd North

Carolina State University

Erik DeBenedictis

Sandia National Laboratories

绿色和可持续计算

Kirk Cameron

Virginia Tech

健康信息学

Upkar Varshney

Georgia State University, Atlanta

高性能计算

Vladimir Getov

University of Westminster

识别科学和生物识别技术

Karl Ricanek

University of North Carolina

Wilmington

物联网

Roy Want

Google

安全和隐私

Rolf Oppliger

eSECURITY Technologies

软件

Renée Bryce

University of North Texas

Jean-Marc Jézéquel University

Rennes

视觉、可视化和增强技术

Mike J. Daily

HRL Laboratories



中国计算机学会(CCF) 会员专属权益

项目	会员	非会员	说明
CCF通讯 (CCCF)	免费	480元/年	全年12期，纸质版，每月邮寄；另有PDF版及IPAD版
中国计算机科学技术年度发展报告	免费	96元/册	权威报告，每年一册，电子版
特价加入ACM	100元/年	240元/年	CCF会员特价加入ACM会员，享受ACM会议优惠、ACM电子版通讯、电子刊等
YOCSEF	免费	付费	除北京总部外，CCF已在24个城市建立了分论坛，每年活动逾百次
会员活动中心活动 (CCF城市分部)	免费	付费	CCF已在24个城市建立会员活动中心，每年活动逾百次
计算机职业资格认证 (CSP)	优惠	付费	一年3次，每年近万人参加，认证结果受到知名高校及企业认可
CCF电子刊	免费	无	每月6期
数字图书馆	免费	无	期刊、培训视频等资料
CCF网站信息发布	免费	无	会员登陆会员系统后在CCF官网发布会员成就、推荐会议、求职、招聘等信息
选举权、被选举权、参与学会治理	专有	无	
中国计算机大会 (CNCC)	优惠	全价	每年一次，每次参会人数逾3000人
学科前沿讲习班 (ADL)	优惠	全价	每年10期，每期邀请该领域国内外顶级专家作为讲者
IEEE CS准会员资格	65元/年	+240元/年	享受IEEE CS会议优惠、每月3期电子刊等
专业委员会会议	8折	全价	CCF拥有35个专业委员会，每年活动近百次
CCF13种会刊论文版面优惠	8.5折	全价	限第一作者

获得以上CCF会员专属服务，只需支付**200**元/年会费！

地址：北京科学院南路6号
电话：(86-10) 62648654
网址：www.ccf.org.cn

通信：北京2704信箱，100190
邮箱：membership@ccf.org.cn



扫码成为CCF会员

计算科学评论



18

22

36

SimH 的故事

撰文 Bob Supnik

摩尔定律的诞生

撰文 Ethan Mollick

两次交互式计算机网络实验

撰文 David Hemmendinger

历史特刊 · 目录



52

《机器、语言和计算》 在 MIT 的诞生历史

撰文 Peter J. Denning
Jack B. Dennis

56

一些往事

编辑 Laurie Robertson

60

专访电子邮件先驱 雷蒙德·汤姆林森

撰文 Dag Spicer

74

一位IBM销售员的宏大战事, 1974–1981

撰文 James W. Cortada

80

深度学习: 人工智能的复兴

撰文 Yoshua Bengio

88

IC卡上的密码进化史

撰文 Jean-Jacques Quisquater
Jean-Louis Desvignes

98

一个数据模型与317种病痛

撰文 W. Wayt Gibbs

106

计算机教育: 美国新战略

撰文 Annie Murphy Paul

114

无人驾驶还需60年

撰文 Steven E Shladover

图片历史

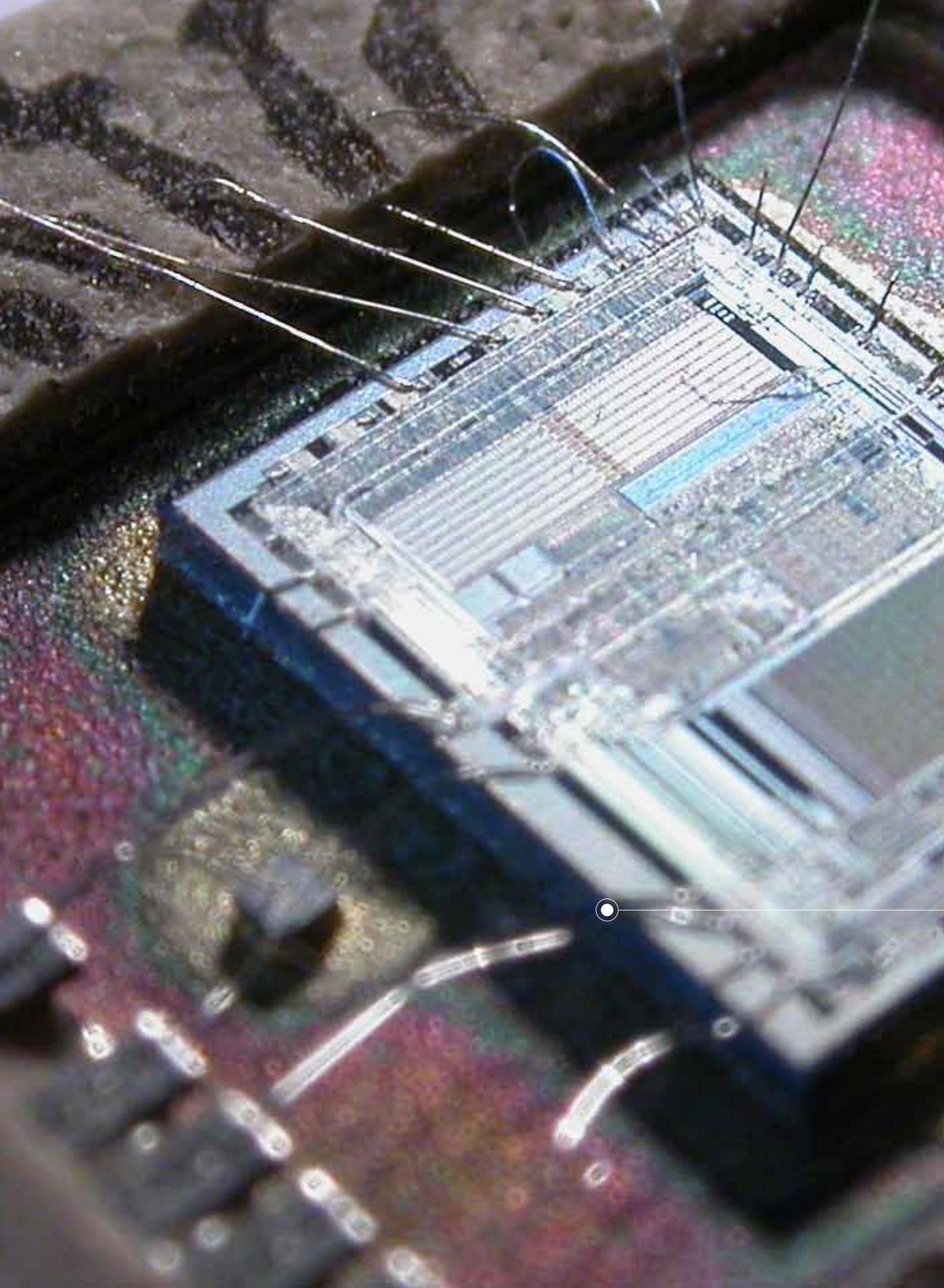
戈登·摩尔

相关文章: 摩尔定律的诞生

戈登·厄尔·摩尔(英语: Gordon Earle Moore, 1929年1月3日-)是英特尔公司的联合创始人。但他最为人知的事迹是提出了摩尔定律,即集成电路上可容纳的电晶体(晶体管)数目,约每隔24个月便会增加一倍。









图片历史

集成电路技术

相关文章: 摩尔定律的诞生

集成电路技术的进展一直在为摩尔定律“续命”。不过现在摩尔定律已经到了一个重要的时刻，因为晶体管密度过大，已经产生了量子效应，导致了漏电等副作用。关于摩尔定律还能延续多久的问题，目前仍有争论。



图片历史

古老的 IBM 主机

相关文章: 一位 IBM 销售员的宏大战事, 1974–1981

1957 年, 人们已经可以利用 IBM704 大型主机协助解决航空研究中的计算问题。

图片来源: NASA



ON
OFF

HALT
INH

W.D.T.

AUTO
RSTRY

MEMORY
PRTCT

1

2

3

4

5

13

14

15

16

17

IMP STATUS

INTERFACE MESSAGE PROCESSOR

Deve
the
by E

T1

1

T2

2

T3

3

T4

4

F

5

I

6

A

7

C

B

P1

9

1

1

ON

OFF

PFI

PFH

POWER

1

2

3

4

SENSE

X

OP

B

A

P/

REGISTER

IMP REGIST



图片历史

接口信息处理器

相关文章: 专访电子邮件先驱雷蒙德·汤姆林森

雷蒙德·汤姆林森于 1971 年首次在 Arpanet 系统中运行电子邮件程序。Arpanet 使用接口信息处理器 (Interface Message Processor, 简称 IMP) 来连接不同的网络。
图为第一台接口信息处理器。





图片历史

诺姆·乔姆斯基

相关文章：《机器、语言和计算》在 MIT 的诞生历史

诺姆·乔姆斯基和他在 MIT 语言学系的学生发现，形式语言和自动机之间存在着未曾被察觉的联系。他们还发现，可以从文法构建自动机，或从自动机构建文法。

SimH的故事

撰文 Bob Supnik

这都是拉里·斯图尔特 (Larry Stewart) 的错, 真的。

1993年初, 我主管的一个项目结束了, 完成了DEC的64位芯片Alpha及其系统程序的开发工作。而我也被“提升”为了公司负责工程战略的副总裁。我在Alpha开发时期做的那些充满乐趣的技术项目也都随之结束了, 例如Raven微程序、筹划中的单片机ECL VAX、为复杂的VAX指令编写Alpha的仿真库、升级冒险与地下城 (Adventure and Dungeon) 的Fortran源代码等。在我的日常工作之外, 我还需要找点别的事情做。

当时在DEC剑桥研究实验室工作的拉里知道我对仿真感兴趣。的确, 我在刚进入计算机行业, 还在应用数据研究 (ADR) 工作时就已经对仿真产生了兴趣, 他建议我编写一个新的针对老式计算机的可移植仿真器, 在PC机上运行, 用来保留过去的一些软件, 让老的系统和架构不至于不断流失。他还指出, 搞一个仿真项目可以让我的编程技能与时俱进, 因为我用得最熟练的那三种语言——microcode、VAX assembler和Fortran, 看起来都没什么前途。

要让这个新的仿真器具备可移植性, 就应该用一种被广泛运用的语言来小心编写它。在1993年, C语言无处不在, 而C++仍处于诞生初期。所以, 在学习了C语言后, 我选择了它来开发这个新项目。我开始着手改写原来的MIMIC仿真系统 (PDP-10 汇编语言, ADR 1969), 使其适应现代的计算机。MIMIC已经把模拟控制台与模拟器本身分离开来了, 从而使得同样的控制台可以在多种不同的仿真器上使用。它还使用了一个规范机器模型, 把处理器, 内存和I/O控制器分离开了, 这个模型是在艾伦·纽厄尔 (Allen Newell) 和C·戈

登·贝尔 (C. Gordon Bell) 的处理器-内存-切换开关 (PMS) 模型的基础上建立起来的。事实证明MIMIC的设计很适合C语言, 再一次证明C其实只是增加了花括号的汇编语言。我先编写了PDP-8, 然后是PDP-11, 来证明控制包是可重复使用的, 然后在1994年发布了SimH (历史模拟器) 的第一个版本。

在这十年剩下的时间中, 我回顾了自己的程序员经历, 模拟了自己在不同时期接触到的机器, 表1中列出了这些机器, 以及其规格参数。

随着新世纪的到来, 我把自己曾遇到的其他机器也加入了仿真器, 有些接触时间非常短暂 (见表2)。

在这个过程中, SimH也逐渐从我个人的爱好项目变成了一个通过互联网相互联系的专业爱好者团队。互联网和Web正好出现在SimH之前, 它们提供了全球合作所需的工具。通过互联网, 我能够联系那些可以获取早已失传了的软件和文档的人, 在调试操作系统和解开棘手问题时获取帮助, 最终还可以招募合作者编写新的模拟器或接管现有的模拟器。Altair、Altair Z80、SWTP 6800、Data General Eclipse、IBM S / 3和IBM1130模拟器都是其他人编写的, HP 2100的则由互联网团队成员接管并完全重写。到了几年前, 很明显, SimH不再是一个单人项目, 它变成了一个GitHub上的开源项目, 由马克·皮佐拉托 (Mark Pizzolato) 负责编辑。

SimH沿着几个方向不断进化。首先, 它可以仿真的系统变得越来越多, 从20世纪50年代的商用计算机到70年代的个人爱好者使用的计算机, 直到21世纪的超级计算机。第二, 控制包变得越来越精密和复杂, 可以提供信号调试功能、一般的断

表 1. 可仿真的计算机，第一组

机器	加入仿真器年份	机器参数	初次个人体验
PDP-8, DEC 1965	1993	12 位, 4~32 KW, 累加器架构	为其编程用作电视台控制台, 1968 年
PDP-11, DEC 1970	1993	16 位可寻址, 8~4096 KB, 通用寄存器架构	为 11/05, 11/40 和 1145 调试架构一致性, 1972 年
Nova, Data General 1969	1995	16 位, 4~128 KW, 寄存器 - 内存架构	为其编程, 用作光谱仪控制器, 1973 年
IBM 1401, 1959	1996	可变字长, BCD 字符, 4~16KC, 内存 - 内存架构	帮助编写了一个基于 PDP-10 的仿真器, 20 世纪 70 年代中期
PDP-4/7/9/15, DEC1962/1964/1966/1970	1996	18 位, 4~32 KW (PDP-15 是 128 KW)	为 PDP-7 DECSYS OS 编写辅助系统程序, 1967 年
PDP-1, DEC 1960	1997	18 位, 4~32 KW, 累加器架构	编写手写识别软件, 20 世纪 60 年代末期
VAX, DEC 1977	1998	32 位可寻址, 256 KB~4096 GB, 通用寄存器架构	在 VLSI 设计中使用, 1980 年后
H316, Honeywell 1969	1999	16 位, 4~32 KW, 累加器架构	为编译器编写代码生成器, 1971 年

点系统、无关字节序的 I/O, 还为常用的外设, 例如磁带和终端多路复用器提供了中心库。

最终, 这个仿真器变得更精确了。随着越来越多的软件出现, 对执行特性的依赖问题也反复出现, 从而有必要根据这些问题来升级仿真器。下面是两个最近的例子。

- ◆ 鲍勃·阿姆斯特朗 (Bob Armstrong) 使用 H316 仿真器, 配合一些他添加的新设备, 来重现当初的 Arpanet IMP 代码。在调试过程中, 他发现处理半双工电传打印机时的不精确, 会导致出现伪中断。我不得不重写了电传打印机模拟器, 修复每处 bug, 使其与硬件兼容。
- ◆ 马特·伯克 (Matt Burke) 发现, VAX Ultrix 在 I/O 空间中对一个未对齐的地址使用了 BLBC 指令。尽管架构手册中严禁这种方式, 但它很明显是有效果的。要修复这个问题就需要把未对齐的 I/O 访问程序彻底剥离出来并重写。

尽管有了这么多变化, SimH 一直保持着相对较高的可移植性。在 2010 年前, 这个项目是不允许

有针对性特别操作系统的代码的。最近, 由于设备仿真器缺乏统一的行业标准解决方案, 例如图形设备, 这个要求需要放宽一点。仿真器需要支持的操作系统也只剩下了 Windows、Linux 和 Mac OS。同样, SimH 一直保持着单线程, 对异步任务进行自身的伪调度, 以免在按照执行指令计量时, 遗失“时间”。这并不适用于现在的多核系统, 但这使得模拟器是完全可预测的——对于调试老软件中的竞争条件和时序问题非常有帮助。

SimH 的非商业性质一直非常明确, 但这并不是意味着它没有商用价值。一个有趣的例子是, 一个 VAX 开发者想利用 SimH 替换他老旧的 MicroVAX 3900, 我试图向他推荐一个商业 VAX 仿真器, 但他坚称非 SimH 不可。于是, 我帮助他从真实硬件上把 VMS 5.5 的镜像下载下来, 然后载入到 VAX 仿真器里。尽管我没有测试过这个版本的 VMS (我测试过 7.2 和 7.3), 但效果很好。那个开发者说, 他的编辑-编译-连接周期从 135 分钟缩短到了 15 分钟, 当时这事就这样过去了。

后来我才听到这个故事的后续部分。那个开发者考虑的这个系统其实是地狱火导弹地面发射站的开发环境。商业 VAX 仿真器不行, 是因为它们都

表2. 可仿真的计算机，第二组

机器	加入仿真器年份	机器参数	初次个人体验
HP 2100/1000 family, Hewlett-Packard 1966	2000	16位, 4~32 KW (HP 1000是1024KW), 累加器架构	无
Ix16b/Ix32 family, Interdata 1967/1974	2000	16位可寻址, 8~256 KB (32位系列是1MB), 寄存器-内存架构	在1968年编写了I4模拟器
GRI-909, GRI Corporation, 1971	2001	16位, 4~64KB, 内存-内存架构	编写程序, 用于控制晶体生长, 1973年
PDP-10, DEC, 1966	2001	36位, 8~1024 KW, 寄存器-内存架构	与人共同编写了MIMIC仿真器系统, 1969年开始
IBM 7094, 1962	2001	36位, 32 KW (装CTSS的是64KW), 累加器架构	MIT的入门编程课程, 1964年
IBM 1620, 1959	2002	可变字长十进制数字, 20~60 Kd, 内存-内存架构	小Fortran程序, 1963年
SDS 940, 1966	2003	24位, 16~64 KW, 累加器架构	无
DEC Alpha (EV5), 1992	2003, unfinished	64位可寻址, 512MB, RISC架构	领导Alpha芯片/系统项目, 1988~1992年
LGP 30, Royal-McBee 1956	2004	32位, 4 KW, 累加器架构	LGP 30是我见到的第一台计算机, 1962年
SC1, SiCortex 2006	2005	64位可寻址, 16MB~2GB, RISC架构, 多核	写入SiCortex从而在芯片设计完成之前促进软件开发
Sigma 32b series, XDS 1966	2007, unfinished	32位, 32~1024 KW, 寄存器-内存架构	无

是俄罗斯开发的, 而且不开源。用SimH仿真VAX虽然速度上跟商业仿真器差得远, 但源代码可以审查, 这是最关键的因素。

有些展示活动会用SimH来形象地表现计算机的高速发展过程。在超级计算2007大会上, SiCortex团队在12个不同核(整个机器有5832个核)上运行了12个SimH仿真器, 用其系统0.2%的计算能力演示了50年的计算发展史, 从IBM 1401直到SC1本身。另一方面, 有爱好者在两个通过以太网连接的树莓派(Raspberry Pi)上运行了VA集群, 用100美元重现了在20世纪80年代需要50000美元的计算设备, 并把占地面积从半个房间缩小到了一个桌面。

SimH也成了处理老式硬件时的参考工具。2011年, 我利用SimH解码了H316操作指令的行为, 并且证明有关这一问题的参考文件实际上并不是很准确。就在今年, 我在澳大利亚的朋友马克

斯·伯内特(Max Burnet)用SimH编写和调试了小型诊断程序, 用来为经典PDP-8桌面电脑的复原机器排除故障。当然, 反过来也一样。计算机历史博物馆中的复原IBM 1401是理解1401行为细节的唯一可靠权威。

尽管我已经从SimH的“无畏领袖”这个位置上退了下来, 但关于它我还有许多事情要做。IBM 7094的仿真器运行的是IBSYS, 是标准的IBM 操作系统, 但不是CTSS——MIT在20世纪60年代早期开发的先进时间共享型系统。(其他7094仿真器运行的是CTSS, 所以不用急。)还没人为PDP-15找到一份MUMPS, 即第一个商用的M 语言。Sigma 32b家庭版仍然不完整, 没有经过测试。还有, 如果我们能找到机器可读的MIMIC源代码, 形成一个闭环, 那不是很好吗!如果在SimH模拟的PDP-10上运行MIMIC, 那它就能运行一个模拟的PDP-7, 反过来又能模拟一个PDP-8……

鲍勃·苏普尼克是硬件顾问兼软件工程领导者，他也是SimH的创始人。他在50年的职业生涯中，曾在DEC担任微处理器开发主管和研究副总裁，并曾在三家创业公司担任工程副总裁。他在2015年退休。可通过bob@supnik.org联系他。

cn Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

微信名：计算人

微信号：jisuanren



《IEEE软件杂志》为软件从业者提供了前沿观点，专家分析和深刻的洞察，让他们跟上日新月异的技术变迁。这本杂志还为软件理论转化为应用提供了权威观点。

[www.computer.org/
software/subscribe](http://www.computer.org/software/subscribe)

SUBSCRIBE TODAY

摩尔定律的诞生

撰文 Ethan Mollick

每个领域都有一些用来进行粗略计算的简单公式或数量关系。这些规则很少得到普及，能像摩尔定律这样独特而又富有影响力的就更少了。摩尔定律是一则 50 年前的预测，其内容是，计算机的运算速度每一两年就增加一倍。下文就是对这条传奇定律的考察，看它如何走上了向自我应验预言演进的道路。

每个领域都有一些用来进行粗略计算的简单公式或数量关系。这些规则很少得到普及，能像摩尔定律这样独特而又富有影响力的就更少了。摩尔定律是一则 50 年前的预测，其内容是，计算机的运算速度每一两年就增加一倍。下文就是对这条传奇定律的考察，看它如何走上了向自我应验预言演进的道路。

摩尔定律的内容是，计算机的运算速度（以单块芯片上晶体管的数量来衡量）每一两年就增加一倍。它以看似无可动摇的精准，被世人誉为电子革命的引擎，视作自我应验预言的典范，而且在学术和大众媒体中极好地展现了技术发展的轨迹。¹ 根据摩尔定律做出的预测成为产业目标的基础，而这反过来又加强了以之衡量行业进程的可信性。在日新月异的行业背景下，摩尔定律成为了企业能够依赖的“唯一固定准则”。²

可奇怪的是，摩尔定律的实际表现却似乎有违此类盛赞。英特尔和仙童半导体公司的创始人之一、物理化学家戈登·E·摩尔，在1965年为《电子学》(Electronics)杂志35周年纪念发表的论文中，首先提出了那条之后以他姓氏命名的定律。摩尔在这篇论文中断言，单块芯片上可安置的晶体管数量，每一年就能增加一倍。仅仅十年后，摩尔就修改了自己的预测，宣称“新的趋势是大约每两年增加一倍，而不是一年”。³ 不过，在最近的技术与大众出版物中，我们常见的是另一种形式的摩尔定

律。某份著名商业与技术期刊对摩尔定律的表述，可作为这一版本的代表：“摩尔定律被誉为现代世界的决定性法则 (defining rule)，其内容是，计算机芯片的性能每十八个月即提高一倍。”⁴ 这些表述之间有明显的分歧，我们很难将摩尔定律看成单独的预测，更难将它当作半导体行业中一个富有影响力的论断。

然而，也正是这些改动，让摩尔定律获得表面上的精准。在过去50多年里，半导体行业经历了很多重大的变化，使得摩尔定律中许多假设都不再恰当。而行业本质发生变化的时机，恰巧切合摩尔定律的修正周期，因此半导体市场变化的同时，摩尔定律也在跟着改变。由此便造成了摩尔定律似乎是一条准确命题的效果，这在它出现的前20年里尤其如此。而实际上，它是一些列不同定律相继接替的结果。

摩尔定律直观上的威力吸引了无数大众化的评论，而学术上的检验却非常少。其他作者也发表过摩尔定律历史的文献，其中Probir K. Bondyopadhyay对定律本源的描述⁵ 以及Robert Schaller就定律自身及其重要性的概述⁶ 尤其精彩。但目前还没有文献详细梳理过摩尔定律从一系列相关定律中演变的经过，以及它在外国竞争介入下、最终确立为行业事实的过程。

追寻摩尔定律的历史并非只是满足好奇心，因为它不仅对半导体行业非常重要，还是技术轨迹和

自我应验预言的重要案例。Donald MacKenzie在他关于技术轨迹的著作中指出，理解摩尔定律有助于我们了解技术模式如何形成：

在所有关于信息技术的经济和社会学研究中，几乎没有一篇公开文献……研究摩尔定律变化模式中的决定因素。不论直白还是隐晦，经济学家和社会学家通常的作法，都是只研究它造成的效果，对它的起因却置若罔闻……在后者这样的研究面世之前，虽说我们无法清楚地知道错失了哪些机遇，但这些机遇绝非无足轻重之物。⁷

从摩尔的预测到摩尔定律，如果将它看作一个动态、而非静态的过程加以考察，或许就可以发现MacKenzie所讨论的某些决定因素。

摩尔定律的技术要素

摩尔定律本身或许已经改变，但有一部分内容却在各个表述中始终存在。在每个版本的定律里，单块芯片晶体管的数量（也称为复杂度[complexity]）都在迅速增长。它的增长可以归结为四个主要因素：核心面积(die size)、电路维度(line dimension)、技术巧聪明度technical cleverness)和技术革新(technical innovation)。它们大体上能够解释摩尔定律对单块芯片晶体管数量快速增长的描述。

核心面积指的是硅片及其制成的芯片的表面区域。核心面积增大，芯片能够安置的晶体管也增多，但在硅片上留下缺陷、致使芯片报废的几率也一同增加。另一方面，电路维度则表示硅片上刻蚀电路的深度、宽度及密度。新型光刻技术的发展及其带来的更细密的刻线，是决定芯片晶体管密度的第二个主要因素。

第三个因素是“技术聪明度”，它不再涉及前两个因素所关注的芯片制造技术，而描述芯片设计的趋势。技术聪明度的第一部分，可以借用摩尔在1975年发明的“设备聪明度(device cleverness)来表示。设备聪明度指的是工程师设计芯片的能力：依靠出色的电路设计，尽可能利用芯片空间，而不依赖芯片制造技术的进步。技术

巧智的第二部分，是开发复杂芯片时计算机辅助设计(computer-aided design, CAD)的贡献。这两部分造就了技术巧智集成电路(integrated circuit, IC)设计，使电路线宽和核心面积的进步得到更加充分的运用。

为芯片密度带来增长的最后一个因素是技术革新，它涉及数字电子学的本质属性。技术革新描述的是制作材料以及芯片制造方式的进步，比如改进逻辑系列(logic families)、开发新基底等。尽管半导体行业也有其它发展，但核心面积、电路线宽、技术巧智和技术革新依然是决定芯片密度的主要因素。而摩尔定律认为，芯片密度会随时间增长。

分析摩尔定律的最后一步，是考察定律中的“时间”部分。半导体行业的演变，根据单块芯片上组件(component)的数量（又称为集成度[level of integration]），习惯上分为以下几个时期：1960年到1965年是小规模集成电路(small-scale integration, SSI)时期，芯片包含的组件不超过100个；之后是中、大规模集成电路(medium-and large-scale integration, MSI及LSI)时期，芯片组件的数量上限为10万个，持续到20世纪70年代中后期；最后，是现在的超大规模集成电路(very large-scale integration, VLSI)时期，芯片组件数量超过10万个。现在的文献几无例外，都将这个分期视为理所当然，但它们各自的时段划分却因为依据不同而千差万别。这是因为这些分期原本就没有精确的定义，而是由半导体产业技术发展的本质衍生而来。

尽管这些时期从名字上看是一个线性的过程（从小规模到中大规模等等），但集成度的过渡并没有这么顺畅，而是暗合了行业基本运作方式的转变。在IC产业中，因为研发周期长，所以行业变化巨大但又多少可以预见，这就迫使企业为将来的需要早做准备。

长程计划的一个结果是，对于当前使用的生产技术，企业能够大致预见其不再适用于生产预定产品的时期。通常，企业可以对当前的生产方法做出简单但变化极大的改善，以此改进产品来满足

未来的产品需求。偶尔，只要四个因素里有一个做出某种新颖的改变，就足以满足既定的生产计划。但在某些罕见的情形下，即使行业在一段时间内不断进行改进，半导体生产还是会遭遇瓶颈，无法通过简单改变当前的工艺来克服。

不同生产机制的变化标志着不同时期的转变，而这些时期通常以行业希望达到的某个标志性产品为代表。例如，SSI时期是在单块芯片上植入一个简单逻辑门；进入MSI时期的标志是用IC构建一个寄存器（电子存储的基本方式）。时期的起止时间很少会有精确界定，不同定义各有差异，在新时期开始的几年里尤其如此。不过，对于新时期将在何时来临、其特征是哪些挑战，工程师和计划制定者拥有一份不成文的共识。

时期过渡虽然来自于技术因素，但它们也可以用时间和集成度来描述。当然，以单块芯片上安置10万个晶体管的时间来标志VLSI时期开始，忽略了突破VLSI瓶颈的经济和技术原因。不过，从历史的角度看，安置10万晶体管的时间正好就是解决VLSI技术难题的时期。标志其它时期的难题也和特定的集成度相关。因此，集成度是半导体行业里用来标志诸多重大挑战的有用指标。

以集成度为背景，摩尔发表摩尔定律三篇主要论文的时机就特别重要了。⁸ 每篇论文都发表在时期过渡之时。最初的论文发表于1965年，处于SSI时期之末；第二篇是在1975年，大概是VLSI投入运用的时候；最后一篇是1995年，正值VLSI技术一段十年剧变期之前。这三篇论文里的摩尔定律都有细微的变化，形式的改变反映了集成度转变的本质。结果摩尔定律维持着表面的精准，但这只是因为它经过了修改，才使得某个时期的定律能够适应下一个时期。摩尔定律自SSI时期出现后，其演变就记录着半导体行业的发展。

SSI时期的摩尔定律

摩尔定律首次出现于半导体行业早期。摩尔1965年发表前述论文时，距离晶体管发明还不到20年。在这篇论文的时代，IC所能实现的功能，用

单独的电子器件也能做到。虽然人们普遍承认IC拥有“成本低、设计非常简单”的优点，但除此之外，它和手工接线的器件没有根本上的区别。⁹ 这样的看法可以说明，至少在短期内，微电子行业都将电子电路视为自己的竞争对手，而没有将IC当做一种独特的产品，来完成只用单个元件无法实现的功能。因此，集成电路和电子电路的经济效益是半导体制造业关注的主要问题。

对摩尔和当时其他高水平的研究者而言，技术问题不如经济问题重要。他们通常认为，只要经济上允许，行业就一定能开发出升级产品所需要的生产技术。比如，摩尔就在论文中提到，“以集成电路所采用的公差”，可以制造安置65,000个组件的芯片，留给我们一个“应该在什么情形下这么做”的问题。⁹ 这个观点恰合当时半导体行业的状况：MSI/LSI时期已经开始，在芯片上安装逻辑门这个最大的技术问题，已于两年前在实验室中得到解决。不过，行业还没有取得很大的经济效益。人们认为，在这个新时期里，必须有合理的经济运作，才能延续技术上的进步。

我们必须在这个暗含的假设下理解摩尔的论文，因为最初的定律来自于经济、而非技术上的分析。摩尔注意到，生产芯片的成本与生产及刻蚀硅片的成本有关，与芯片上组件的数量无关。因此，芯片密度增大，单个组件的成本就降低。还有一个相反的效果，“随着芯片组件增多，产出减少会抵消复杂度增大的效果，导致每个组件的成本上升。”⁹

产出减少的原因和硅片上的随机缺陷有关。硅片使用的次数越多，就越有可能造成缺陷，破坏芯片的有效部分而使之报废。组件成本下降和复杂度升高造成的缺陷增多，共同构成一条U形成本曲线，其最低点表示在两个因素作用下，组件平均成本最低时组件的密度（见图1）。

摩尔由这个发现更进一步。他利用过去几年的数据，画出组件平均成本与芯片上组件数量的U形曲线，注意到了行业生产无缺陷低成本芯片的能力在随时间增强。将这些曲线的最低点连接起来，

就得到一条直线图像，表示复杂度每年即增大一倍（见图2）。摩尔根据这张五个数据点的图像得到下面的结论，它是摩尔定律的基本表述：

当然，可以预计这个增速在短期内会持续下去，甚至还会增大。从长远看，增速不那么确定，但我们有理由认为，至少10年内它都能保持基本恒定的增长。也就是说到1975年，在成本最低的集成电路中，组件数量将达到65,000个。⁹

它是第一个与摩尔定律相关的预想，而且它也非常准确。尽管摩尔预测了达到某个技术水平的具体时间，但他不是通过考察技术因素得到这个结论。摩尔在一次采访中认同了这个说法，说自己的论文是试图找出以最低成本生产微型芯片的方式，获得超出本意的弦外之音，也是出乎他的意料之外。¹⁰ 技术成就作为经济的从属功能，是SSI向MSI/LSI时期过渡的特征。

MSI/LSI时期的摩尔定律

MSI和LSI时期在20世纪70年代中期步入正轨。此时，摩尔定律的预测虽然依旧精准，但它的基本假设却已经名存实亡。此时的情形已经不再是经济驱动技术发展，技术问题和经济问题对半导体行业同样重要。MSI/LSI时期技术上的巨大进展改变了半导体产品生产的经济模式。此外，技术变革也改变了IC的市场行情。IC不再是单个电子器件的替代品，因为新生产的产品只能采用集成功能器件(integrated functions)才能使用。1975年电子元件会议上，摩尔发表了摩尔定律的演讲，其中提到他对技术因素日渐重要的认识。摩尔在发言开始时说：“加工与设计技术从许多方面，使制造复杂单片机或存储电路等器件变得经济可行。”¹¹ 初始定律中核心的经济理由(即集成功能器件必须与电子电路竞争)已经不复存在，所以摩尔稍稍修正了他的定律(或许是无意的)，从而更好地反映出LSI向VLSI时期过渡的本质。

为完成这一转变，摩尔在论述定律的历史与未来时探讨了技术因素的作用。他将复杂度的增长拆分为种种改进过程，并用技术、而非经济上的进

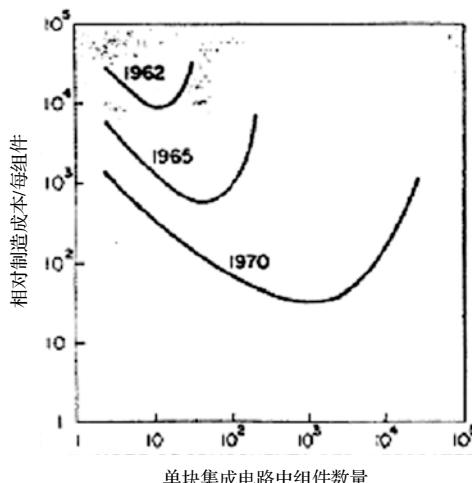


图1. 单块集成电路中组件的数量 (Courtesy Penton Media. Reprinted from G.E. Moore, "Cramming More Components onto Integrated Circuits," Electronics, vol. 38, no. 8, 19, Apr. 1965, pp. 114-117.)

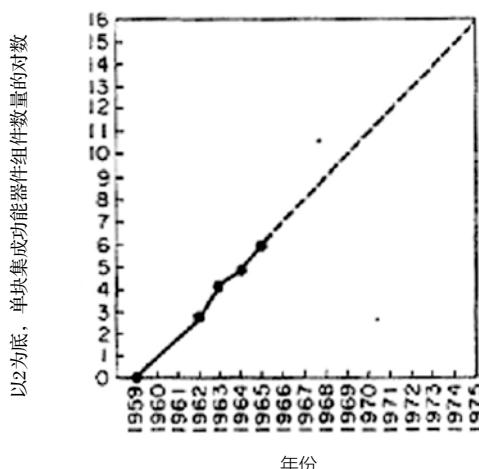


图2. 以2为底，单块集成功能器件上组件数量的对数。 (Courtesy Penton Media. Reprinted from G.E. Moore, "Cramming More Components onto Integrated Circuits," Electronics, vol. 38, no. 8, 19 Apr. 1965, pp. 114-117.)

步，来解释半导体行业的发展。¹² 摩尔对定律背景的改动，比论文中的变化更剧烈。

这些变动来自于芯片生产和IC运用中出现的技术革新。微电子学中出现了三大新应用：动态随机存取存储器(dynamic random access memory, DRAM)、微处理器(microprocessor)和通用异步收发器(universal

asynchronous receiver transmitter, UART), 它们为电子设备之间提供了交流的手段。DRAM 和微处理器迅速成为两大重要的半导体产品, 这个变化反映在对摩尔定律的新解读上。

DRAM在许多方面都是非常重要的产品。首先, DRAM拥有大量寄存器和重复单元, 因而容易设计。DRAM的制造商就可以使用最先进的生产技术, 将芯片的复杂度发挥到极致。此外, DRAM也是半导体制造商的重要市场。纵观历史, 存储芯片一向供不应求, DRAM销售往往在整个IC行业中占据大头。因为DRAM产品非常重要, 又通常是可制成的最复杂的芯片, 所以在衡量半导体行业状况时, 往往以它作为指标。摩尔也不例外, 在VLSI时期开始时, 摩尔就将DRAM芯片的数据点用在了表示摩尔定律的曲线图中。¹⁰

DRAM和微处理器在半导体行业中变得十分重要, 但它们在摩尔定律形成之初还不为人知。定律为适应它们也开始了暗中改变。微处理器虽然是复杂的电路系统, 但比起生产难度而言, 它更是设计上的挑战。它们的晶体管密度很少有DRAM那么高, 因此位置会在摩尔定律图像的下方。而DRAM的特征, 便使它们成为芯片最高复杂度的理想指标。结果, DRAM在复杂度曲线上取代了微处理器。到20世纪80年代早期, 摩尔的曲线被视为“动态RAM密度的”准则, 而不是普遍原理。之后在VLSI时期, 随着时间推移, 微处理器变得越发重要起来, 结论又会发生改变。不过在70年代晚期, DRAM是最重要的芯片产品, 因此也是摩尔定律关注的焦点。

在MSI/LSI时期, 技术革新不只是设计新型的电路, 还有新的逻辑系列。金属氧化物半导体(metal oxide semiconductor, MOS)、互补金属氧化物半导体(complementary metal oxide semiconductor, CMOS)和电荷耦合器件(charge coupled device, CCD)相继出现。其中CCD尤其重要, 因为人们希望它能成为一种新型存储芯片的关键元件。

在技术革新突飞猛进的背景下, 其它三个因

素也有显著发展, 进一步增大了芯片的复杂度。其中核心面积与电路线宽, 在摩尔的论文中被认定为两个增大了芯片复杂度的领域。这两个因素都取得了巨大的飞越, 尽管它们的进展多是缓慢稳定的变化, 而缺少突破。摩尔用图像将这些变化表现出来, 并确定了自1965年以来, 它们为增加芯片复杂度而贡献的程度。根据摩尔的结论, 它们可以解释晶体管数量增长的三分之二强。

摩尔将剩余部分的增长划给了最后一个因素, 即技术巧智。在MSI和LSI时期, 电路设计者与之前相比, 变得非常擅长利用芯片的整个区域。这是CAD以及存储芯片(具有简单、易于设计的单元结构)发展带来的结果。摩尔在1975年对芯片密度的分析中称, 技术巧智“对复杂度的贡献, 比增大芯片面积、使用更细密的刻线更重要”。¹³

虽然技术巧智带来了如此重大的发展, 摩尔却认为, 这一因素此后对决定芯片复杂度的重要性将大大降低。他的理由建立在CCD存储器的基础上, 他知道CCD在不断发展。CCD“几乎可以逼近最高密度”, 为设计上的提升只留下很小的余地。³因为他了解这个行业, 所以知道十年后CCD芯片将得到广泛应用。¹⁰ CCD通行之后, 技术巧智方面的贡献将荡然无存, 摩尔定律中的增幅也出现变化, 从每年增长一倍, 一度变为1980年的每两年增长一倍。³

摩尔没有料到的是, CCD存储器对天然本底辐射很敏感, 最终沦为失败产品。CCD的失败意味着, 本来五年才须变化的摩尔定律, 几乎立即就要改变。结果, 虽然摩尔预测新图像的斜率依旧正确, 但1975年重构的摩尔定律是错的。¹⁴ 到了20世纪70年代末, 翻倍时间大约是两年, 和摩尔预测的一样; 不过单块芯片上组件的数量要低于摩尔的预期, 因为曲线在更快地变平。最终, 这段CCD经历倒没怎么反映出定律的弱点, 反而指出了两个重要的优点。

第一个优点是, 摩尔可以有效地改变定律中的斜率, 使之更好反映向VLSI时期过渡的本质。实际上, 这会给定律带来不同的曲线, 也没有构成

初始摩尔定律可信依据的历史案例。摩尔之所以能这么做，是因为他将自己的修正描述为预测，而不是在斜率变化后改写原先的定律。而且，事实上摩尔修改后的预言非常有先见之明，因为他是在曲线斜率下降前做出预测的。这样，即使修改了定律，摩尔也能保持它的可信度。

摩尔修改定律是无心之举，而非刻意吹嘘自己的预测准确。因为，即使是在1975年的演讲中，摩尔也没有认为自己的预测有多大意义。一次访谈中，摩尔在提及1975年的修改时说：“我不觉得有谁会按照它来制定商业计划，可能是因为我还沉浸在第一次预测正确的喜悦当中。我不觉得还有人会关注它。”¹⁵ 不关注定律是否重要，是摩尔敢于修改它的部分原因。他不用顾虑许多，只是在延续十年前那篇论文开启的预测传统而已。摩尔定律在接下来十年里重要性不断增长，又使这几个预测看上去像是某个无可避免的现象的组成部分，而不是一套不同的评估。

CCD事件还折射出摩尔定律另一个重要优点，这个优点和前一个相关。摩尔在CCD上犯的错，说明他是依靠业内信息来做预测。当时CCD还没有进入市场，摩尔就已经对它的投放时间、以及它会对行业产生的效果了然于胸。期望落空之时，他和行业中每个人都同样感到意外。摩尔承认，业内人士的视角给他带来的业内消息，要比大部分人早“一到两年”。¹⁰ 类似的情况，还有1965年那篇论文中最后的那个数据点。那个数据来源于仙童半导体公司研发的产品，当时尚未面世。¹⁵ 有了这些消息，摩尔做出复杂度曲线会变平的预测，就没那么了不起了。因为他就像其他业内人士一样，知道在同等经济效率下，英特尔的工程师也无法开发出技术上更巧妙的设备。结果，摩尔定律之所以精准，有部分原因（至少来自于半导体行业之外）与他对英特尔的了解有关，并不是纯粹的预测。

结合上述要素可知，1975年后，摩尔定律大体上就是向VLSI过渡阶段的产物，就像1965年的摩尔定律是向MSI/LSI过渡的结果那样。新版的摩尔

定律，不再假设技术进步纯粹是经济因素的结果，而意识到电路密度（line density）等领域的技术进展，也有自己独立的发展速度。诚然，这版摩尔定律的大背景、来源和含义都发生了改变，但它依旧保有一种历史可信性，来自于那条有着十年经历的经验发现。此时，它已经做好进入VLSI时代的准备了。

超大规模集成电路（VLSI）时期的摩尔定律

VLSI时代始于上世纪70年代末。当摩尔在1995年的那篇论文中对摩尔定律下定义时，摩尔定律已经被证实是VLSI时代芯片复杂性的准确指标。随着这点逐渐广为人知，摩尔定律也成为难于预测的计算机产业的少数稳定规律之一。在某种意义上，摩尔定律也成了未来的可靠指导。

对摩尔定律描绘的计算机产业的传统模式进行研究后我们可以发现，摩尔定律的成功部分归因于对该定律的阐释的进步，这让摩尔定律在VLSI时代长盛不衰。让摩尔定律的准确性家喻户晓的关键因素是VLSI时代早期的一系列文章的发表。这些文章在传播摩尔定律的历史有效性的同时，实际上也对其进行了某种程度的夸大。

在70年代末，集成电路（IC）开始获得广泛关注。微处理器的发明和个人电脑的到来使得芯片成为“电脑革命”的风向标，而这使人们对芯片的兴趣激增。在这场浪潮的肇始之初，对半导体的特征进行了通俗科普宣传的主要媒体是《科学》（Science）和《科学美国人》（Scientific American）。这两篇文章都发表在1977年。

在这些出版物中，关于摩尔定律的文章是由罗伯特·诺伊斯（Robert Noyce）撰写的。诺伊斯和摩尔共同创立了英特尔（Intel）以及更早的仙童半导体公司（Fairchild Semiconductor）。在对半导体行业发展的预测方面，这两篇文章以及他们对摩尔定律的独特阐释使其确立了似乎不可动摇的地位。这两篇文章在许多方面都很雷同，但是发表在《科学》上的文章比发表在《科学美国人》上的技

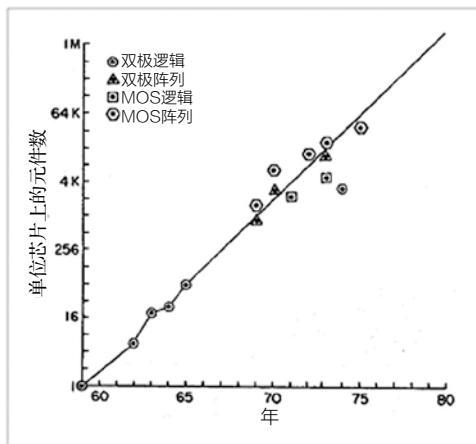


图3. 随着面市时间而变化的电路复杂性

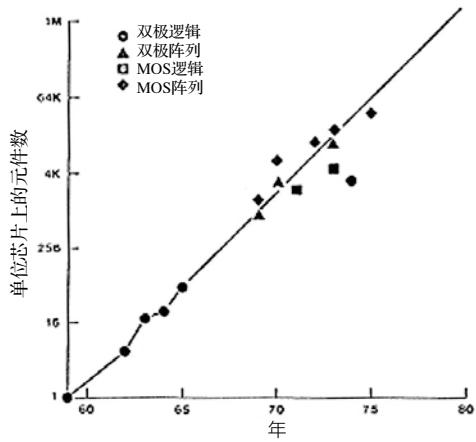


图4. 随面市时间变化的复杂集成电路元件数量估值

术性更强一些。

但是，这两篇文章也讨论了半导体行业愈发复杂的趋势。当然，摩尔定律也考虑到了这些因素。这些文章通过以复杂性、最少维度和管芯面积为纵坐标、时间为横坐标的图，对行业发展趋势进行了解释。在摩尔1975年探讨半导体未来的那篇文章中，也用了同样的变量作图。

这两篇文章和摩尔的论文之间的相似性不止于此。诺伊斯的图（见图3）和摩尔的图（见图4）是一模一样的，甚至都没有更新1975年和1977年之间芯片发展的数据。不过，虽然他们用了相同的图表，但是摩尔定律的本质却由于两个作者的不同处理而发生了分歧。诺伊斯的主要图表显示的是随着

时间增长的元件数量，这幅图的标题是“随着面市时间而变化的电路复杂性”。¹⁶ 这幅图完全复制了摩尔1975年的论文中所用的图。

摩尔1975年论文的图表的标题是“随面市时间变化的复杂集成电路元件数量估值”。¹¹ 这两幅图标标题的主要差异是在“复杂集成电路”的措辞上，两者的差异很大。在摩尔的论文中，他用了某种标准选取了那些数据点，并只描绘了“复杂集成电路”的数据。而诺伊斯的论文似乎把这些数据点当成了所有电路的代表。

摩尔对“复杂”电路的定义的关键信息可以在1975年的图中找到。摩尔的图描绘的是一条将一些数据点连接起来的回归线，60年代初期有4个数据点，而70年代初期则有大量数据点聚成一团。但是位于这两组数据之间的1965到1968年则没有任何IC数据。这非常奇怪，因为那段时期的半导体行业出现了巨大的进步，其中包括整个中规模集成电路（MSI）时期。

摩尔在另一篇论文中对此图进行了解释。这篇文章发表在1979年。他在这篇文章中又使用了这幅图，并宣称：“请注意1965和1968年之间的空白。这是由于那段时期很难找到复杂性能够超越临界的半导体产品”。¹⁷

他的这番话说明了两件事。首先，摩尔认为自己的定律表达的是任何时刻复杂性的上限。第二，摩尔并没有画出没有达到复杂性期望值的数据点。

这种对数据处理的方法从他的角度看自然非常合理。因为他要向世人展示的是从经济角度考虑最有效的复杂性水平随时间递增的方式。对于摩尔来说，经济效率即生产效率。他关注的是将单位元件中电路的生产成本降至最低。但是他的观点却忽略了半导体的可销售性和设计外观方面的因素。

摩尔自己也在1965年的论文中承认，仅仅能够制造复杂的芯片并不意味着就应该去制造。有时候，设计师无法研发出能够最大限度利用最有效的芯片生产手段的商业产品。而摩尔在一次访谈中也表示，他的兴趣在于研究已实现商业化的技术前

沿。¹⁵

这就是60年代末发生的事。那时IC还相对新鲜，生产方面的进步远超设计步伐。这背后的部分因素在于，电子产品生产中的基础性研究尚未完成，而重要的新进展（比如逻辑门的研发）并没有使用大量集成元件。¹⁸ 摩尔没有在图中描绘这类芯片，因为它们无法证明IC不断增长的生产效率。

诺伊斯的文章虽然用的也是摩尔的图，但是生产效率的语境统统消失。这让诺伊斯的文章看起来描绘了芯片复杂性在20年间的持续增长。实际上，如果加入1965年到1967年间的数据，这张图看起来就没有那么平滑了，因为这段时间的数据远低于预测趋势线。¹⁹

因为诺伊斯的文章非常受欢迎、常常被引用，因此他所描绘的普适的、准确的摩尔定律扎根在了大众心中，而摩尔更为严谨的预测则没有获得同等关注。结果就是，摩尔定律的历史准确性广为传播，但它对历史研判的准确性实际上并不符合真实的情况。摩尔定律的含义的转变，实际上也是因为受到了诺伊斯以及后续相关文章的干扰。

虽然很难明确地证明，大众对于摩尔定律的感知来源于这两篇文章，但是有大量的间接证据指向这一结论。诺伊斯的文章是该话题中第一篇流行起来的，而那时微电子学开始升温。这两篇文章也迅速成为相关讨论的基础。

尤其在1981年初，相关的讨论就十分火热。一个工业分析师这样描述《科学美国人》上的那篇文章：“或许它是关于微电子学的论述中最著名的文章…诺伊斯用来解释微电子学的基本‘规律’的图被广泛复制和引用”。²⁰ 诺伊斯也是第一个在大众出版物中使用“摩尔定律”的人。不过摩尔倒是把这个名词归功于加州理工学院的卡弗·米德（Carver Mead），后者大概在一两年前铸造了这个新名词。²¹ 虽然这些证据并不能确凿地证明是诺伊斯将摩尔定律广而告之，但至少它们说明了诺伊斯的文章的重要性。

摩尔定律的准确性应该部分归功于历史的再演绎。而摩尔定律在VLSI时代的另一个因素——

灵活性则是另一个原因。摩尔定律并不是为VLSI时代量身定制的，比如，摩尔最新的文章仅仅指出，新的复杂性翻倍的时间更接近2年，而不是之前的1年。¹³

只要实际复杂性的增长率符合摩尔的曲线，摩尔定律就会被认为对VLSI时代是有效的。而实际上复杂性翻倍的时间（大约是26个月）是在该定律被认可之后才被确定下来的。这个现象以及历史修正主义确保了摩尔定律在VLSI时代的成功。

这些因素助长了摩尔定律从一条解释“经济因素如何影响技术革新”的命题转变为一个产业预言。摩尔定律的所谓历史准确性意味着，摩尔曲线成为了一条自然规律、一个技术发展过程的指标。

因此，摩尔定律常被视为行业规划的重要组成部分，这让它成为自我实现的预言。许多公司根据复杂性增长趋势来设计生产计划表。

与此同时，摩尔定律也附着实际情况而在预测方面进行自我转变。VLSI时代最重要的两样产品——微处理器和存储器芯片成了摩尔定律的主要预测对象。在1995年的那篇文章里，摩尔明确使用动态随机存取存储器（Dynamic Random Access Memory, DRAM）和处理器的数据来绘制图表。

这么做显著增加了摩尔定律的简单性。需要注意的是，DRAM背后的技术十分稳定。也就是说，技术革新并不是增加DRAM复杂性的因素；自从感光耦合组件（Charge-coupled Device, CCD）失败以来，技术本身是否优秀已经不再成为考虑因素了。这就意味着，DRAM的进步仅仅是由于密度和管芯面积的增长推动的，而这两者的增长速度均较为稳定。DRAM的标准化进一步让摩尔定律在VLSI时代的预测变得更容易测量。

在早期，对逻辑系列和芯片种类的区分让IC之间的比较变得困难。不过VLSI时代的DRAM并没有受到这个问题的影响，因此能够直接计算摩尔定律在DRAM方面的准确性。

在摩尔的图中，43个数据点代表着DRAM芯片的每平方毫米的晶体管密度。这些点包括了5代

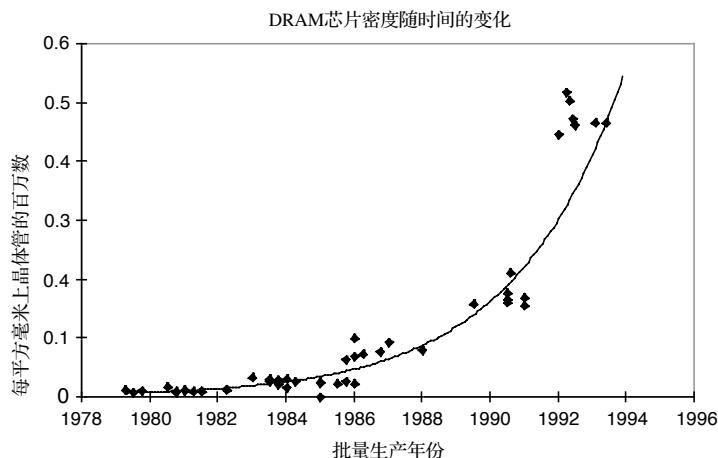


图5. 每平方毫米上晶体管数量随DRAM的批量生产年份的变化 (22)

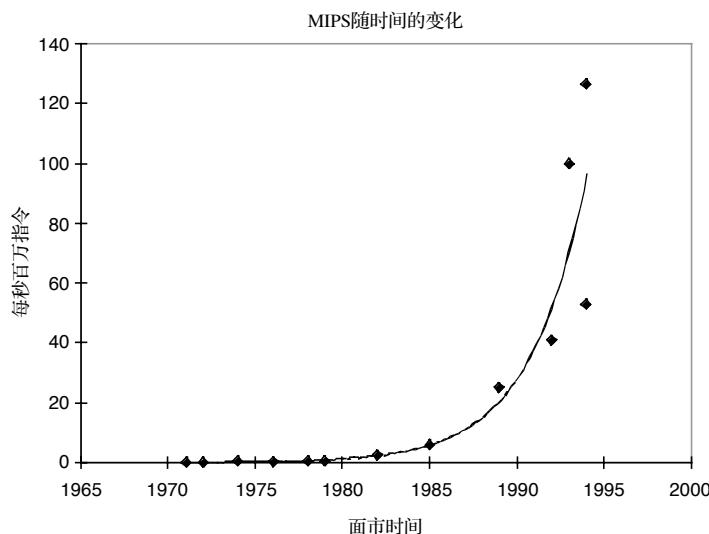


图6. MIPS处理速度随面市时间的变化

DRAM的数据，这5代DRAM上市时间跨越了70年代末到90年代中的这段时期。这些点连成了一条指数型曲线，每年都会粗略增长31%，每26个月会翻番。

这条曲线——也就是所谓的“摩尔定律”——和数据的拟合很好，能够解释数据点的76%的变异。每26个月实现翻番的预测就是由此而来。这就意味着，至少在VLSI时代的大部分时间，被称为摩尔定律的26个月的周期的确是技术革新的准确指标。²³

此外，由于微处理器的重要性日益凸显，它们

也被合并到摩尔定律对VLSI的定义中。对微处理器的强调同时也改变了摩尔定律的测量体系。在并入微处理器后，摩尔定律首次被用来描述趋势而不是不断增加的复杂性。这是因为，和DRAM芯片相比，微处理器的晶体管密度并不高，因此不能用和储存器芯片一样的尺度来测量微处理器。

一般是用芯片上晶体管的总数，或是处理器的操作速度（每秒百万指令MIPS）来作为指标。

一般来说，一家公司在出品新一代的微处理器前会用老一代的设计推出一系列更快、更优化的微处理器。因特尔原创的486芯片就是一个例子。这个20MIPS的芯片是在1989年发布的，接着在1990年因特尔又推出了改良的27MIPS芯片，最后在1991年发布了41MIPS的芯片。后面的几个版本的晶体管数量和486芯片是一样的，只是因为设计的进步而提升了性能。因此，微处理器通常是用性能，也就是MIPS作为作图数据，而不是晶体管总数（见图6）。

图6展示的就是比较符合VLSI时代指数型增长的摩尔定律的曲线。图中年均有30%的增长，拟合度达到了97%。不过，这幅图中的数据点似乎是分成2组。趋势线下面的2个点代表的是更老的芯片（用于经济型电脑）模型的改良版。如果把这两个点去掉，拟合会更好。

这3个测量指标——DRAM复杂性、处理器晶体管总数，以及处理速度都是摩尔定律在VLSI时代的考察因素。摩尔定律的当代版本预测，芯片的性能将以每年30%的速度不断改进。

摩尔定律的准确性并不令人吃惊，因为摩尔定律是在规律被发现后才贴上的标签。不过，摩尔定律在VLSI时代的不寻常之处在于，它在这个时代依然保持了长时间的准确性和一致性。在过去，定律的有效期只有几年，然后就会被修改。²⁴ 而摩尔定律却在20年内屹立不倒。这说明，半导体产业整体以摩尔定律提出的创新速率为指导而发展。

半导体产业的变化

虽然自1975年以来，很多因素促使摩尔定律

成为产业基准，但在向超大规模集成（VLSI）过渡的初期阶段，维持半导体产业遵循摩尔定律快速发展的关键在于国际竞争者的加入。在此之前，半导体产业分散多样，不同种类的芯片使用相互竞争的逻辑系列。多样性意味着产业缺乏公认的标准，这也是摩尔定律在其出现的头15年不断发生变化的主要原因。到了20世纪70年代后期和80年代初期，半导体产品和制程实现标准化，这不仅使摩尔定律更为适用，也使国际竞争愈加激烈。推动这一变化的正是从产品到制程技术的转变。

在VLSI时代初（1979年），半导体制程主要应用几项稳定的技术，这些技术长期以来基本保持不变。²⁵ 制程不断得到完善和改进，无需重大突破也足以维持半导体产业的技术进步。很多情况下，这些技术持续获得成功对半导体公司而言是个意外——他们早已习惯了不断变化的生产方式。

最典型的例子是刻蚀工艺，它驱动了线密度增长。人们预计70年代后期发展起来的光刻工艺到了80年代中期便不再适用，一些公司因此研发了昂贵的新工艺。²⁶ 然而对标准光刻制程的改进使得这项基本技术跨越了预期的障碍，延用至今。²⁷ 渐进而非改革式的发展模式导致半导体制造商对一系列生产方法进行标准化，降低了进入产业的门槛，为激烈的竞争开辟道路。同样的标准化也发生在使用这些技术生产的产品中。

在LSI时代，各有特色的逻辑类型大量出现。那时，至少有11种不同的逻辑系列可供使用。²⁸ 然而，人们渐渐发现每种系列都存在缺陷，达到某个复杂度后便无法继续使用。到了VLSI时代，几乎所有重要的产品都使用MOS技术，接着出现了变种PMOS，再后来是CMOS。²⁹ 摩尔本人认为，逻辑系列的标准化是摩尔定律在70年代后期更为适用的部分原因。¹⁰

但实现产业标准化的不仅仅是逻辑类型。半导体公司长期以来持续生产各种各样的产品。在产业发展初期，生产复杂器件还很困难，任何新产品的出现都有可能推动技术发展。但到了VLSI时代

初，情况大不一样。芯片生产技术变得非常先进，只有两种产品推动了技术创新，使半导体产业的复杂度进一步提升。

这两种芯片就是微处理器和DRAM（动态随机存储器），它们都对提升复杂度至关重要，但在电路密度、晶粒尺寸、技术创新和技术智能化这四大要素中的着力点不同。DRAM由一系列相同的基本元件组成，因此制造的关键在于提高电路的精细度和密集度。而微处理器需要很大的面积来容纳大量功能各异的部件。所以，DRAM通常驱使刻线尺寸不断变小，而微处理器则使晶粒尺寸增加。³⁰

产业界因此更加容易预期未来的发展。通过借鉴微处理器公司的晶片制造技术，DRAM的生产者可以完全致力于获得更小的刻线尺寸。³¹ 而只有在生产几种大家熟知的芯片时，相关技术才需要发展。这样，人们便有可能预见技术发展的方向。这些标准化的集成电路同样改变了半导体产业竞争的本质：大部分利润只和少数产品类型相关。

特别是DRAM——VLSI时代的关键产品。首先，几乎每种和计算机相关的产品都需要内存芯片，这带来了巨大的市场。另外，产业界每2~3年便会推出新一代DRAM，存储能力往往是上一代产品的4倍。³² 大部分消费者会逐代升级，这为最新的芯片创造了持久的市场。根据以往的经验，人们对新产品的需求几乎是无限的，但制造商往往无法满足市场需求。这些因素使得DRAM成为整个集成电路行业销量最大的一支。但DRAM不仅仅是重要的产品，它还是半导体产业的主要推动者，无论是技术上还是经济上。

在经济方面，至少在VLSI时代后期，DRAM的竞争市场是近乎完美的示例：³³ 需求量大，可替代产品多，价格成为区分内存芯片的主要标准。³⁴ 但在VLSI时代初，DRAM还是相对较新的产品，参与市场竞争的公司很少。此外，由于需要发展各种专门的制造方法，进入半导体行业非常困难。这种情况直到制程标准化出现才得以改变。

VLSI时代到来之前，半导体公司依靠声誉和专业研发将产品销售到小众市场。随着一些产品，

比如DRAM，的生产技术逐渐成熟，非专业公司使用标准制程生产的产品也可以进入半导体市场。制程和产品仍在以摩尔定律的节奏快速创新。只不过在VLSI时代，发展是渐进而非改革式的，这为国际竞争者的加入创造了机会。VLSI时代的黎明也是半导体大商业的开端，因为国际公司被吸引入这一行业。

日本的兴起

在LSI到VLSI的过渡期，产业界最值得关注和讨论的变化是70年代后期日本集成电路公司的突然兴起，尤其是在生产内存芯片上。长期以来，半导体产业的大本营都在美国。直到VLSI时代，日本出其不意地从美国手中抢走了DRAM的巨大市场份额。到1983年，日本控制了全球DRAM市场的 $\frac{3}{4}$ 。³⁵

日本成功逆袭的主要原因是：日本公司在进军半导体市场时更注重改进制程，而不是产品。³⁶ 重点的不同区分了两种截然不同的创新策略：产品的改进趋于产生市场领导者；而制程的创新提升了制造能力，直接影响产品的价格和质量。³⁷ 半导体公司在二者之间不可有所偏颇，但是选择哪一种更优先对于企业业绩至关重要。³⁸

例如，日本公司注重改进制程技术。在成熟的市场中，一旦出现某些拳头产品，注重制造技术将为公司带来竞争优势。因为在这种情况下，价格和质量是区分产品的主要标准。³⁶ 因此，制程创新虽然不像产品发展那样引人注目，却是工业策略中的重要一环。

真正为日本开启半导体产业大门的是市场标准化。在70年代后期，无论是产品还是制程，日本都落后美国好几年。³⁹ 为了克服劣势，日本的策略是首先进入半导体市场相对稳定的分支，再逐渐扩大规模。日本公司为此计划合力主攻一种利润丰厚的标准化产品，以此保持市场竞争力。他们选择了DRAM。DRAM在设计上相对简单，并且产量高利润大，为日本公司赶上美国半导体制造商提供了契机。这一计划由日本通产省制定，称为VLSI

项目，于1976~1979年实施。VLSI项目聚集了五家主要的电子公司：富士通、日立、东芝、NEC和三菱，由他们共同进行一系列研发。日本政府在这个项目上投入了2.5亿美元，涵盖了项目总支出的40%，VLSI项目也因此成为日本半导体产业获得政府补贴最多的项目。这个项目成功地改进了日本的半导体制程，提高了日本公司的竞争力。⁴⁰

摩尔定律制度化

在VLSI项目大获成功之前，日本就开始在DRAM市场上崭露头角。在LSI时代末，通过推出16K(16K比特)DRAM，日本的内存芯片第一次获得可观的市场份额。1976年，三家美国公司首先设计并推出了16K芯片。1977年又有两家美国公司和两家日本公司开始制造这种芯片。⁴¹ 到了70年代后期，日本在16K DRAM市场上获得了可观的份额，并且准备研发下一代产品。日本公司获得成功的三个因素是：企业环境不同，芯片质量更好，产品垄断日本国内市场。由此带来的竞争优势后来被证明是日本半导体产业成功的关键。这三个因素也促进了“26月成长周期”（摩尔定律）的制度化。即便如此，64K芯片才真正决定了日本在DRAM制造方面的全球领导地位，它被公认为过渡到VLSI时代的关键器件，也为日本打开了利润丰厚的市场。⁴²

64K芯片同样“见证”了国际竞争的威力，它迫使半导体产业遵循摩尔定律实现高速发展。在DRAM市场中，随着芯片产量提高，生产芯片的成本迅速降低。^{32,43} 首先销售这一代芯片的几家公司占尽先机。由于几乎没有竞争对手，他们能够快速提高产量，降低成本和价格。后来加入市场的公司没有规模经济的优势，将面对更高的成本，但产品价格还必须与迅速降低的市场价格保持一致。就这样，日益激烈的竞争带来的压力促使各公司以摩尔定律的节奏前进，因为位于曲线后端的公司将会失去市场。而这就是美国公司在64K DRAM的市场竞争中实际遭遇的情况。

在64K芯片的市场竞争中，日本公司首次以大优

势打败美国公司，并且这种优势保持了大约两年。美国制造商在发展64K芯片方面落后的原因在于，他们尝试跳过这一代芯片——产品和制程创新都直接为256K内存芯片做准备。⁴⁴ 在全球化市场形成之前，这不失为一条良策，但是愈加激烈的竞争意味着公司的创新节奏既不能比摩尔定律慢，也不能太快。这是因为昂贵的创新会延迟对某一代产品的生产，相比进入市场更早的竞争对手，公司将在削价竞争中处于劣势。成功的策略需要适度的创新，这样公司才既能跟上产业界的前进步伐，又不会因为发展更先进的技术而延迟生产。美国公司没有计划大量生产64K芯片，若是在全球竞争时代之前，导致的后果很可能只是推迟64K芯片的上市日期，放缓摩尔定律曲线的节奏。

然而，日本竞争者的加入使得64K芯片如期上市。VLSI项目成功实施后，日、美在技术方面不相上下。但日本公司选择了和西方制造商不同的发展道路：他们不使用实验性技术，只是在16K芯片的基础上改进、升级，同时也在研发更先进的64K芯片。⁴² 因此，日本公司在市场竞争中打败了美国。虽然两家美国公司——摩托罗拉和德州仪器花费巨大成本迅速重新设计芯片，终于跟上了产业发展的节奏。⁴⁵ 但在美国公司的芯片进入市场时，64K芯片的市场控制权已经牢牢掌握在日本手中。

国际公司的加入不仅使竞争更加激烈，也为芯片的设计和生产提供了更多样的创新途径。不同国家的经济结构、薪资率、投资法律和相对优势不尽相同，这将导致创新的节奏不同。但也许更重要的是，他们选择的创新策略不同。⁴⁶ 日本加入半导体产业在增加竞争压力之外，还带来了不一样的创新途径。

迫使半导体产业的复杂度按指数增长的“幕后推手”是全新的商业气氛，而非摩尔定律。过去，大多数半导体公司生产各种各样的产品，只有少数公司生产类似、适配的芯片。在这种体系下，如果一些公司延迟了某种芯片的上市日期，就像64K DRAM事件中的大多数美国公司那样，整个市场都会受到影响，因为少数能够及时生产芯片的公司

无法满足市场的需求量。但是随着国际竞争时代的到来，这种情况通过两种方式得到了改变。第一，所有公司的生产方式全都相同的可能性不大，这保证了技术问题不像从前那样普遍。第二，某些制造商的发生延迟事件时，其他国家的公司可以抓住机会，扩展市场份额。因此，激烈的竞争导致了半导体产业以自控的方式更快发展。

在美国公司的发展无法与摩尔定律的节奏保持一致时，国际竞争成为维持这条发展曲线的关键因素。我们并不知道摩尔定律开始主导半导体产业发展节奏的确切日期，但到了80年代中期，这两个概念已经不分彼此。⁴⁷ 从那时起，其他因素进一步巩固了摩尔定律的轨迹，其中包括微软-英特尔联盟、美国国家半导体蓝图和其他许多有待分析的经济驱动。在80年代，摩尔定律完全适用于VLSI时代，这一回，它稳定地维持了超过20年。

参考文献

1. See, for example, D. MacKenzie, *Knowing Machines*, MIT Press, 1996, for a discussion of Moore's law as a self-fulfilling prophecy.
2. M.S. Malone, "Chips Triumphant," *Forbes ASAP*, Feb. 1996, p. 70.
3. G.E. Moore, "Progress in Digital Electronics," *Technical Digest of the Int'l Electron Devices Meeting*, IEEE Press, 1975, p. 13.
4. M.S. Malone, "Chips Triumphant," p. 68. Eighteen months is the standard time given in recent publications, both technical and nontechnical. See, for example, T. Lewis, "The Next 10,000 Years: Part II," *Computer*, May 1996, p. 78.
5. P.K. Bondyopadhyay, "Moore's Law Governs the Silicon Revolution," *Proc. IEEE*, vol. 86, no. 1, Jan. 1998, pp. 78-81.
6. R. Schaller, "Moore's Law: Past, Present, and Future," *IEEE Spectrum*, June 1997, pp. 52-59.

7. D. McKenzie, *Knowing Machines*, chapter 3, pp.49-66.
8. Although the 1965 and 1975 papers are generally considered the critical papers on Moore's law, I have included Moore's 1995 work since it contains the technical material on which he has based all of his recent interviews and presentations.
9. G.E. Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, vol. 38, no. 8, 1965, pp. 114-117.
10. G. Moore, personal interview with author, 17 Dec. 1996.
11. G.E. Moore, "Progress in Digital Electronics," 1975, p. 11.
12. G.E. Moore, "Progress in Digital Electronics," 1975, p. 11-12.
13. G.E. Moore, "Progress in Digital Electronics," 1975, p. 12.
14. See for example, J.G. Linvill and C.L. Hogan, "Intellectual and Economic Fuel for the Electronics Revolution," *Science*, 28 Mar. 1978, p. 1107, which uses the faulty prediction.
15. G. Moore, personal interview with author, 16 Feb. 2005.
16. R.N. Noyce, "Large Scale Integration: What Is Yet to Come," *Science*, vol. 195, 18 Mar. 1977, pp.1103-1105. In *Scientific Am.*, this graph is simplified, but it uses the same slope as the *Science* graph; see p. 37.17. G. Moore, "VLSI: Some Fundamental Challenges," *IEEE Spectrum*, April 1979, p. 32.
18. G. Moore, personal interview conducted by R. Schaller, 13 June 1996.
- 74 IEEE Annals of the History of Computing Establishing Moore's Law July–September 2006
75
19. I was unable to find precise transistor counts of the most complex chips from those years, but Moore's article makes it clear that the number of transistors was less than what was expected.
20. T. Forester, ed., *The Microelectronics Revolution*, MIT Press, 1981, p. 29.
21. Noyce wrote: "We have yet to see any significant departure from Moore's law"; R. Noyce, "Microelectronics," *The Microelectronics Revolution*, T. Forester, ed., p. 34.
22. Some data courtesy of M. Iansiti, Harvard Business School, the rest gathered by the author.
23. Note that the 26-month doubling time differs from some published versions of the law. Eighteen months is the number given in much of the popular press, but no authors that I spoke to could trace this number back to its original source. Intel's Web site gives "18 to 24 months," a range that is close to the 26-month cycle.
24. The SSI version in Moore's original paper lasted from 1959 until 1966, when the gap in Moore's graphs begin. The MSI/LSI version as developed in 1975 lasted from approximately 1968 until 1976, ending with the failure of CCDs.
25. US Office of Microelectronics and Instrumentation (OMI), *A Competitive Assessment of the US Semiconductor Manufacturing Equipment Industry*, US Document Printing House, 1985, p. 60.
26. M.S. Malone, *The Microprocessor: A Biography*, Springer-Verlag, 1994, p. 69.27. G.D. Hutchenson and J.D. Hutchenson, "Technology and Economics in the Semiconductor Industry," *Scientific Am.*, Jan. 1996, p. 55.
28. R. Zaks, *From Chips to Systems*, Sybex, 1981, p.23.
29. D. Manners and T. Makimoto, *Living with the*

- Chip, Chapman & Hall, 1995, pp. 102-103.
30. "Processes of the Future," Solid State Technology, vol. 38, no. 2, 1995, p. 42.
31. M. Iansiti, *Technology Integration: Making Critical Choices in a Dynamic World*, Harvard Business School Press, 1998, p. VII-10.
32. H. Gruber, "The Learning Curve in the Production of Semiconductor Memory Chips," *Applied Economics*, vol. 24, no. 8, 1992.
33. M. Iansiti, personal interview with author, 7 Dec. 1996.
34. N.S. Dorfman, *Innovation and Market Structure*, Ballinger, 1987, p. 213.
35. H. Queisser, *The Conquest of the Microchip*, Diane Crawford-Burkhardt, trans., Harvard Univ. Press, 1988, p. 112.
36. W.R. Boulton, E.S. Meieran, and R.R. Tummala, "Japan's Product Development Strategy," *Electronic Manufacturing and Packaging in Japan*, Japanese Technology Evaluation Center 1994; www.wtec.org/loyola/ep/toc.htm.
37. H. Gruber, *Learning and Strategic Product Innovation*, North-Holland, 1994, p. 82.
38. Economist Harald Gruber developed a productprocessmatrix that models the advantages and distinguishing characteristics of these differing approaches in the semiconductor industry. Gruber, *Learning and Strategic Product Innovation*, pp.82-95.
39. The president of Nippon Telephone and Telegraph stated in 1974 that Japan was behind the US by three years in VLSI technology. M. Anchordoguy, *Computers Inc.* Harvard Univ. Press, 1989, p. 139.
40. See K. Flamm, *Mismanaged Trade?*, The Brookings Institution, 1996, p. 98, for a list of statements on the importance of the VLSI program.
41. Electronics Panel of the National Academy of Engineering (NAE), *The Competitive Status of the US Electronics Industry*, Nat'l Academy Press, November 1989, p. 48.
42. M.G. Borrus, *Competing for Control*, Ballinger, 1988, p. 147.
43. This is often attributed to the so-called learning curve, in which costs fall as experience producing a chip increases. In fact, economies of scale, in which higher levels of production result in greater efficiency, cause this effect.
44. D.I. Okimoto, T. Sugano, and F.B. Weinstein, eds., *Competitive Edge*, Stanford Univ. Press., 1984, p.48.
45. "A Chance for US Memories," *Business Week*, 15 Mar. 1982, pp. 126-127.
46. G. Dosi, *Technical Change and Industrial Transformation*, St. Martin's Press, 1984, pp. 272-275.
47. See, for example, R.W. Wilson, P.K. Ashton, and T.P. Egan, *Innovation, Competition, and Government Policy in the Semiconductor Industry*, Charles River Associates, 1984, pp. 34-35.

Ethan Mollick is麻省理工学院斯隆管理学院的博士生。他的研究方向是影响创新过程的因素。他在哈佛大学获得了本科和MBA学位。联系方式: emollick@MIT.edu。

For further information on this or any other computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.

两次交互式计算机网络实验

撰文 David Hemmendinger

对于分时计算机网络的早期实验主要有两次，分别是1963年系统开发公司（SDC）和斯坦福大学研究所（SRI）的分时计算机实验，以及1966—1967年该公司与麻省理工学院林肯实验室的实验。这两次实验都受到里克莱德（J.C.R. Licklider）的影响，他当时对信息共享很感兴趣，两次实验中均包括了对远程计算机程序的交互式使用。

对于分时计算机网络的早期实验主要有两次，分别是1963年系统开发公司（SDC）和斯坦福大学研究所（SRI）的分时计算机实验，以及1966—1967年该公司与麻省理工学院林肯实验室的实验。这两次实验都受到里克莱德（J.C.R. Licklider）的影响，他当时对信息共享很感兴趣，两次实验中均包括了对远程计算机程序的交互式使用。

上世纪60年代的后五年里，美国和英国都对计算机网络进行了大量研究。其中最著名的网络是1969年开始运营的Arpanet，而在英国，唐纳德·戴维斯（Donald Davies）打算建立一张全国网络，该网络的第一个小型版本几乎和Arpanet的最初实验同时进行。¹这两个网络均使用戴维斯和兰德公司的保罗·巴兰（Paul Baran）提出的数据交换方式，后来成为长距离计算机网络的规范。不过，在早期的网络实验中，接收的文件非常小。本文介绍的两次网络实验均使用加州圣莫尼卡的系统开发公司（SDC）的时间共享计算机。第一次实验是1963年SDC和斯坦福大学研究所（SRI）建立的为时不长的连接，第二次实验是1966—1967年SDC和麻省理工学院林肯实验室建立的连接，后者的知名度要大一些，常被视作首次长距离网络连接的实现，网上有一本书将其称为“种子实验”，但把时间误写成了1965年。²这次实验为更多人所知的原因可能是其组织者是拉里·罗伯茨（Larry

Roberts），他后来成为Arpanet项目的负责人。不过很有意思的是，两项实验的结果都没有对外公布，只在内部技术报告和备忘录中提及。

资源共享

上世纪50年代末，一些研究人员提出了分时系统的几种实现方式，使连接至一台计算机的多台电传打字设备或其他终端连接能同时处于活跃状态。^{3,4}这可以让几个用户同时对程序进行在线调试和除错，在保证计算机运行的时间效率的同时实现多线程交互。分时系统出现于上世纪60年代初，其中包括SDC的AN/ FSQ-32（Q-32）系统。⁵1962—1964年任美国国防部高级研究计划局（ARPA）信息处理技术办公室主任里克莱德（J.C.R. Licklider）是分时系统的主要支持者，他在任期间大力资助分时项目。他对这一技术的兴趣是出于对更广层面上人机交互技术的探索，进而产生了对旨在资源分享和共同工作的计算机网络。

上世纪50年代，计算机联网是为了交换信息，当时的SAGE系统利用计算机通信来定位雷达通信。⁶这些联网的计算机没有让管理员远程调用用户层面的程序，而是用固定的一套程序将数字信息通过电话或无线电传输。而本文中介绍的网络实验允许系统用户在远程计算机上调用任何适当的程序。

1963年4月，里克莱德向“星际计算机网络”

(Intergalactic Computer Network)成员提交了一份备忘录，其中提到了在一台计算机上工作时可以从其他计算机上调用程序的设想。但他同时提到，这个从远程计算机导入二进制程序的观点是有问题的，因为这些程序架构是独有的。

当计算机为我运行程序时，我以为这个过程发生在SDC的计算机上。不过我很快就认识到这只是我的推测。凭借成熟的网络控制系统，我不用决定是否要发送数据，以供别处的计算机的程序运行使用，或者引入其他计算机上的程序，用我的数据来运行。我并不怎么反对做出这样的决定，但从原则上来说，由计算机或网络来做这样的决定似乎更加合适。⁷

在阿瑟·诺博格 (Arthur Norberg) 和朱迪·奥尼尔⁸ (Judy O' Neill) 撰写的ARPA信息处理技术办公室的发展史中，将该单位的成果归纳为在里克莱德的信息共享观点的指引下的集体成果，他的继任者伊万·萨瑟兰 (Ivan Sutherland) 和罗伯特·泰勒 (Robert Taylor) 也持同样的观点。

作为信息共享观点的一部分，分时的理念最初源于学术计算机中心在程序开发的编写-测试-除错过程中有意提高计算机使用效率。根据诺博格和奥尼尔的说法⁹，最初的计算机网络就是一台带有远程终端的分时计算机，比如SDC 的Q-32和MIT兼容分时系统。与分时开发不同的是，用于多台计算机间通信的计算机网络开发主要由信息处理技术办公室资助。

计算机网络升级往往采用自上而下的方式，这并不奇怪。由一家机构对自己的计算机进行分时处理可能效率更高，但将多家机构的计算机进行联网可能被视作让他人获得稀缺资源的访问权限。实际上，最初推广Arpanet时的情况就说明了这一点。泰勒曾在报告中提到1967年的一次ARPA计算机网络会议：“我接触的大部分人一开始都不喜欢这个主意，我觉得大多数人最初都将其视为一次机会，让别人使用自家程序的机会。”¹⁰ 我们知道，不久之后人们开始热衷于分工协作的带来的机遇，里克莱德的期望成为了现实。

其他Arpanet和互联网发展史中，包括海蒂·哈弗内尔¹¹ 和马修·莱恩¹² 的记述中也确认了计算机网络最初的发展是由IPTO提供的支持。本研究的剩余部分将考察两个实例，然后简要回顾一下可能存在的早期项目，这些项目的资料比较匮乏。

资源

关于SDC Q-32分时系统的几篇文章中提到，该系统存在网络连接，但是只有少数几篇公开发表的著作中提到了其使用情况，而且都缺乏细节。这些著作包括查尔斯·伯恩 (Charles Bourne) 的书和讲座以及SRI的技术报告。在托马斯·玛利尔 (Thomas Marill) 和拉里·罗伯茨 (Larry Roberts) 的文章中提到了1966年SDC与林肯实验室建立的连接。在《个人工作站发展史》¹³ 一书和加州大学洛杉矶分校的一篇博士论文中也提到了这两次实验。¹⁴

遗憾的是，SDC的历史技术文件并没有全面归档。查尔斯·巴贝奇研究所的Burroughs档案中包括一些SDC文件，但没有一个文件中提到网络实验，虽然1966年的年度研究报告中确有提到。在林肯实验室的历史文件中和国防部技术信息中心的网站的季度研究报告中可以查到更多的信息，后者提供的信息颇为粗略。笔者也参考了罗伯特·泰勒和拉里·罗伯茨的口述历史中的说法，并与当年供职于这两个项目的人互通了邮件，从中受益颇丰。

SRI-SDC实验

SDC的Q-32采用了1963年中投入运行的分时系统 (TSS)。1964年的一篇关于分时系统的论文中简要提到了与SRI的控制数据CDC 160A的网络连接，传输距离超过300英里。实际的连接建立在160A和作为Q-32通信前端的PDP-1之间。SRI的程序使用该网络对存储在Q-32上的书目数据库进行全文搜索。这一网络的设计者查尔斯·伯恩 (Charles Bourne) 与道格拉斯·恩格尔

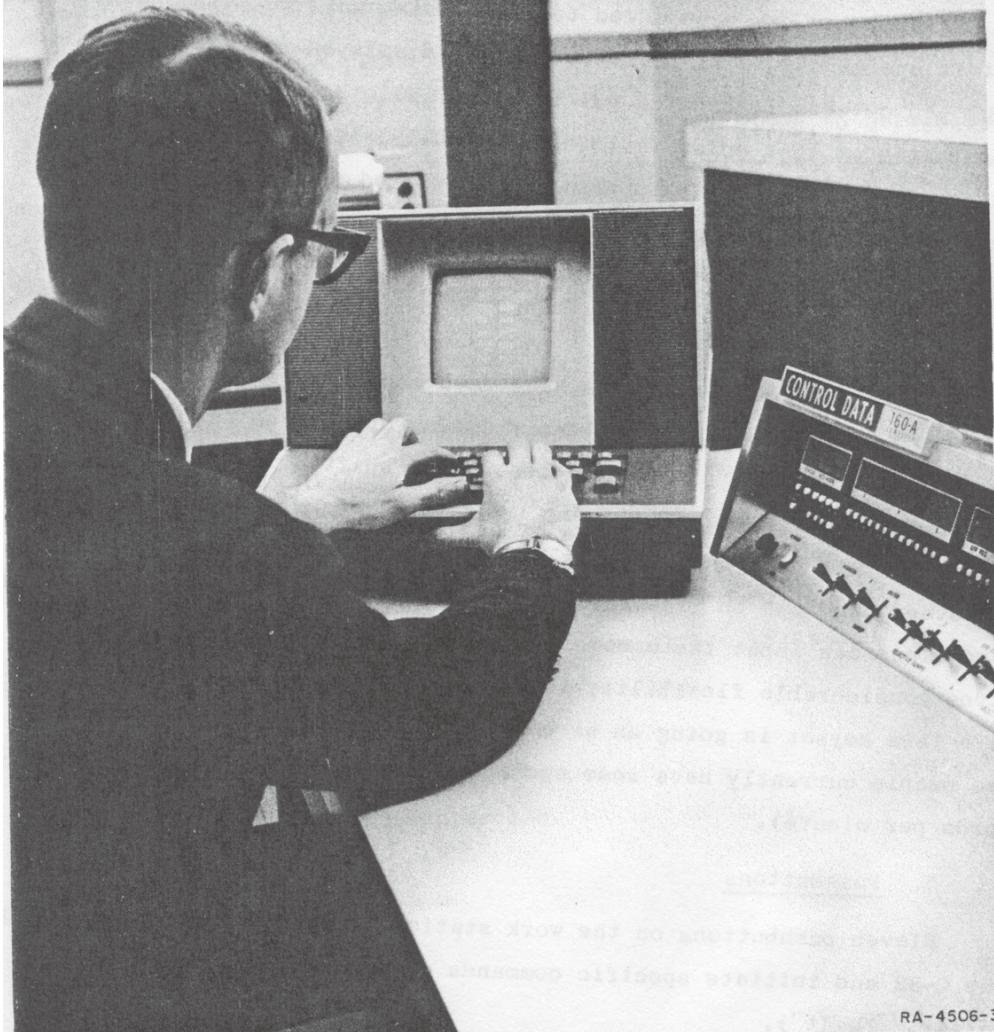


图1. 斯坦福大学研究所的控制数据CDC-160A工作站。操作人员在这个交互式工作站使用了恩格尔巴特的五音“弦键”（未在图中），允许操作人员使用单手输入，另一只手则使用光笔。

巴特 (Douglas Engelbart) 一起从事信息检索工作，伯恩后来担任加州大学图书馆研究所所长。他在一篇SRI技术报告中描述了该项目，¹⁶ 后来又在1999年康拉德纪念馆的讲座上¹⁷ 和2003年出版的书中再次提及。¹⁸这是一个概念验证系统，旨在证明全文搜索词语和书目术语的可行性。该系统是恩格尔巴特主持的增强人类智力项目的一部分，该项目得到了IPTO的部分支持。¹⁹根据恩格尔巴特后来的说法，他们当时建议使用与本地计算机相连的交互式工作站，但里克莱德要求他们将数据库放在SDC的Q-32系统上，并建立一个网络连接。(IPTO也资助了Q-32分时系统。)这种交互式机-机网络在远程Q-32系统上运行搜索软件，由SRI CDC机器提供用户界面。尽管一些计划中

的功能（例如搜索字符串中的布尔操作）没能实现，但这个带远程连接的搜索程序在1963年年底成功完成了对一小组文档的搜索。搜索过程中本应使用SDC的50条记录，但由于记录中存在许多缺陷，最终实际只使用了7条记录。

由于项目失去了资助，对该系统的研究不久后即被放弃。伯恩在他的讲座中表示，当时SRI的项目开始“被资金牵着走”，在SRI的资金流入与冷战相关的项目后，他们没能为这个项目找到资助机构。根据他书中的说法，当时他虽然是项目经理，但已经忘记了这个从未命名的项目，直到他三十多年后写书时才想起来。

恩格尔巴特提到伯恩的短命项目，是我发现的该项目唯一一次公开提及。恩格尔巴特补充说：

“出于各种在开创性投资中常见的原因，与项目的目标和计划相比，第一年的研究是相对低效的”。虽然里克莱德愿意继续提供支持，但恩格尔巴特说自己无法提供足够的资金。图1所示交互式工作站的照片来自伯恩的报告，该报告还提到使用五键“弦”键盘，以便用单手输入，而另一只手则使用光笔。²⁰ 这种输入方式似乎与恩格尔巴特在1968年演示计算机鼠标输入的方式相似。²¹

根据伯恩的说法，网络使用AT&T的201A调制解调器（根据朱尔斯·施瓦茨（Jules Schwartz）的说法，该网络在2kHz线路上运行），传输速度为2000bps。²² 如果使用当时最新的ASCII编码，运行速度可以达到约200字符/秒。虽然在已发布的文献中没有关于网络实现的更多细节，但是笔者在对SDC和SRI的程序员的采访中发现了更多信息。

当时的Q-32系统程序员之一克拉克·魏斯曼（Clark Weissman）表示，SDC和SRI之间的网络使用两条全双工电话线，其中一条用于向Q-32发送命令（例如登录，加载和执行），另一条用于向这些命令启动的程序传输数据。²³（他指出，这种安排类似于CPU的设计，将数据和控制分离开来。）两条线都与处理系统的所有终端输入输出的PDP-1相连。他表示，这种设计的缺点是需要两条线路，但优点是只需要对操作系统进行微小的修改。因为命令和数据流是彼此分离的，所以操作系统执行程序（命令解释器）的执行可以不作变动，除非必须将两条电话线作为单独的终端来处理。它首先接受来自命令行的输入，当输入命令开始在Q-32上运行程序时，该程序将被分配数据线“终端”用于其输入和输出。由于PDP-1负责管理终端，因此只需对其操作系统进行更改，而无需更改Q-32的系统。

与伯恩一起合作开发全文搜索程序的SRI程序员伦恩·蔡廷（Len Chaitin）也回顾了双线设计。²⁴ 魏斯曼和他都表示，网络连接运行良好。蔡廷说，在硬件中只进行了少量的位校验。按蔡廷的说法，当时对网络的设计宗旨是尽量简单，便于以

后供其他人使用，不过之后并没有人使用该网络。然而，当时其他人在使用洲际电话线路时遇到很大困难，而这种简单的网络设计就可以实现几百英里距离内的有效传输，这一点令人感到意外。

笔者没有发现任何SDC文档中提到SRI-SDC实验，也没有发现将2000 bps连接用于其他任何用途的记述。该网络仍然列在1965年的SDC年度研究报告²⁵ 中，该报告还引用了这一网络设计的最终报告²⁶，但该文档不在巴贝奇研究所关于SDC的收藏中，笔者在其他地方也没有发现。

关于“第一”的声明需要持谨慎态度，而且“第一”通常并不是被考虑对象的最重要的特性。尽管如此，1963年的SDC-SRI网络似乎是第一个被设计和用于交互使用的网络，尽管使用方式很简单。在此之前，用于数据传输的计算机-计算机网络已经存在，比如在SAGE系统中的网络，而且人们可能已经巧妙地将其用于交互，但这些网络不是为交互而设计的。更重要的是，SRI-SDC属于网络的早期实例，证明了在一个项目中可以发挥不同计算机的各自优势，比如Q-32的存储优势和CDC系统上使用交互式终端的优势。里克莱德对网络的兴趣就是这样直接催生出了一个实验。

LL-SDC实验

一篇发表于1967年的对TSS性能的评论称SDC-SRI网络已经过时，同时还提到了林肯实验室的Q-32和TX-2之间的连接，²⁷ 即托马斯·玛利尔和拉里·罗伯茨所提到的网络，²⁸ 它常被人们称为第一个交互式计算机网络连接。尽管他们在论文中提供了网络实验的细节，但实验结果却没有公布，这让人感到很意外。后来的人提及这次实验的很少，罗伯茨曾在一次回顾Arpanet项目²⁹ 时表示，实现通信是可能的，但电话线的传输速度太慢、可靠性不足。这个评价和当年的参与者对SDC-SRI连接的评价不同，后者对SDC-SRI连接的评价是传输速度慢，但是相当可靠。也许这一差异反映了三年后人们对网络计算更高的期望。

玛利尔-罗伯茨网络实验是何时开始的尚不清

楚。诺博格和奥尼尔对美国军方对支持信息处理项目的研究³⁰ 表明, 1965年, 玛利尔刚刚成立了美国计算机公司(CCA), 正在寻求当年的合同, 并向IPTO提交了关于网络研究的建议, IPTO将其作为林肯实验室合同的分包合同, 设立了该研究项目。³¹ 其他资料中, 比如DARPA的Arpanet发展史³² 中, 只提到该项目是与ARPA与林肯实验室签订的合同的分包合同。

玛利尔在计算机网络发展史中的角色似乎只限于这个实验。他是一位心理学家, 曾与里克莱德一起工作, 在麻省理工学院获得博士学位, 随后加入了里克莱德所在的Bolt Beranek and Newman公司(BBN), 从事感知心理学、信息检索、分时和人工智能的研究。³³ 随后, 玛利尔开始在CCA开展各种项目, 包括数据库、信息检索和图像识别。在20世纪80年代末和90年代, 他任麻省理工学院人工智能实验室的研究员, 致力于机器人和感知研究。

罗伯茨最著名的头衔是Arpanet项目带头人。应IPTO第三任主任罗伯特·泰勒的要求, 他在1966年底接管该项目。³⁴ 1964年底, 他出席了在弗吉尼亚州Hot Springs召开的关于计算的未来的会议, 会上, 里克莱德说服了他相信了计算机网络的重要性。³⁵ 他在一次采访中回想起那次会议: “当时我对通信很感兴趣, 但我在那时和之前一直不知到应该研究什么方向。”他继续在林肯实验室研究计算机图形。他在1965–1966年间认识了玛利尔, 并出任玛利尔项目的合同经理。

在线互联网档案库³⁶ 的Arpanet信息汇总中³⁷ 载有凯蒂·哈弗内尔(Katie Hafner)为她的互联网著作收集的资料。其中包括玛利尔给她的一组信件, 内容有关他和林肯实验室的合同。³⁸ 他1965年5月向林肯实验室提出的一个网络项目的提案收到了态度含糊的回应。于是他在6月致信罗伯茨, 更详细地讲了他想做这项研究的原因, 他建议连接三台电脑。接着, 林肯实验室在7月提出一项正式提案。从信中我们看不出5月的提案是谁提出的。在没有人给出合理理由, 证明Q-32可用性的情况下, 不管是作为对IPTO提出建议的答复, 还

是在该答复提出的请求中, 玛利尔都不大可能提出使用SDC Q-32。在笔者收集的最后一封信(写于1965年11月)中, 玛利尔向罗伯茨提到他们一周前的讨论, 要求罗伯茨将他的合同延期六个月。

玛利尔1966年6月提交的报告³⁹ 就是在这六个月中写的。报告第三稿的落款日期为2月28日, 几乎与最终版本相同。报告内容反映了当时处于早期的网络状态。与后来的Arpanet的计划一样, 报告将资源共享作为开发网络的主要原因: 使专用系统或程序可以从任何分时系统访问。该报告主要是作为可行性研究提出的, 玛利尔在报告中花费大量篇幅来讨论字符编码和传输, 以及现有的调制解调器及其成本问题。报告还讨论了使用电话线, 并推荐了几种方案, 可根据使用情况进行选择。后来玛利尔和罗伯茨合写的论文提出的方案, 即使用一条4 kHz的西联语音级线路和一台1200 bps调制解调器, 就是玛利尔论文中的推荐方案, 使用时间可达50小时/月。

报告中最有趣的部分是其软件部分。和后来玛利尔和罗伯茨的文章一样, 它给出了三项策略。其中两项几乎不需要对操作系统进行改动。第一种“快而脏”的方法仅需要让程序与两个终端(用户和远程计算机)进行通信。用户将在本地系统上启动程序, 程序首先与远程计算机通信进行登录, 然后在远程计算机上运行程序, 并向用户一样与之进行数据交换。玛利尔表示, 系统的唯一改变是要提供一个命令, 让用户程序拨打电话号码。

该报告补充说, 这样的连接可以达到电传打字机的速度(11字符/秒), 而不需要做进一步变动。速度更快的调制解调器和线路可以提高传输速率。玛利尔认为, 传输速率达到100字符/秒不会造成硬件压力, 但可能会降低其他线路上的服务质量。实际上, 根据PDP-1手册中关于中断处理的描述, 即使以该速率进行稳定传输, 用来处理输入的性能也不会超过SDC PDP-1计算能力的1%至2%。⁴¹

第二种方法将对命令和数据使用单独的线路。如果使用单通道, 程序或操作系统必须解析所

有输入以区分命令和数据。如果使用两个通道，系统就不需要检查数据线路，因为其内容将直接输入程序。虽然使用两条线路可能看起来很贵，但玛利尔认为实际上不一定真的贵，因为只有其中一条线路需要高速传输，而命令线路通常只是短暂连接。读者应该记得，SDC-SRI网络使用的就是这种双线路设计。在最后的建议章节中，玛利尔提到 SDC-SRI 网络是一个仅用于数据的连接，并由第二条命令线路加以补充。由此来看，他显然意识到双线路设计实际上已经成功部署了。

第三种方法只是一种推测，即对命令和数据分别使用单个高速线路。它需要运行一个协议来区分命令和数据，玛利尔建议只有每第n个位或字符可能属于命令流，以便操作系统只需要计数位或字符以便能够识别命令尴尬的设计。

玛利尔建议使用的三台计算机是 TX-2、Q-32 和 MIT Project MAC 用于分时研究的 IBM 7094。他讨论了 LL 和 Project MAC 之间的资源共享的优点，但他没有说明希望使用 7094 的原因，该机从未参与他的网络实验。根据诺博格和奥尼尔的说法，“MAC 机那些人对参与实验不感兴趣”。⁴² 另有消息说，IBM 不想参与。⁴³ 玛利尔的报告中提到，MAC 7094 使用一台 IBM 7750 计算机作为通信前端（和 SDC 的 PDP-1 类似），线路传输速度为 1200 bps，玛利尔的方案不需要更改 MAC 的硬件或软件。不过，一旦实验中添加了通信协议，就需要对系统软件做出一些改变来解释协议，这可能就是 MAC 选择不参与实验的原因，但是真正的原因如何没有确定的信息可用。玛利尔在 1967 年的备忘录只是简单地说了一句“出于各种原因，MAC 并没有参与实验。”⁴⁴

在林肯实验室 1966 年和 1967 年的季度报告中，简要记述了使用 TX-2 的 APEX 操作系统之后的一年内，网络组建的进展缓慢。

目前正在研究 TX-2 的低速、多路复用数据终端序列的硬件、软件和操作要求。初始要求是系统可以通过“拨号”或租用的电话线路通过电传或 2000 比特/秒的速率与其他计算机通信。⁴⁵

程序被编译后，
会在一秒钟左右的时间内给出
结果，这对于洲际通信来说
性能并不算低。

(1965年11月至1966年1月)

连接电话线数据终端与 TX-2 所需的输入输出单元的逻辑设计已经完成。……正在获得西联宽带切换服务，用于每秒 1200 bps 异步操作的数据集和自动应答单元，用于加州圣莫尼卡的 SDC 的数据连接。⁴⁶ (1966 年 2 月 - 4 月)

[TX-2] 计算机现在使用一条低速数据通道与远程控制台和其他计算机通信。现在正在对使用该硬件与 APEX 时间共享系统所需的第一阶段软件进行测试。目前已对 APEX 进行改动，允许使用两个新的控制台，发送和接收字符流的基本路径已准备就绪。首次远程连接的目标将是位于加州圣莫尼卡的 SDC 的 FSQ-32 计算机。⁴⁷ (1966 年 8 月至 10 月)

在 SDC 处的 TX-2 和 AN / FSQ 32 之间的网络链路已经被混合地成功执行。计划在下个季度举行大规模示范。⁴⁸ (1966 年 11 月至 1967 年 1 月)

在 SDC 的 TX-2 计算机和 AN / FSQ-32 计算机之间的网络连接已被用于几个演示。目前已创建并运行了使用 TX-2 的 Lincoln Reckoner 和 SDC 的 LISP 分布式程序。在 TX-2 上键入的语句在被发往 SDC 进行解析，然后被送至 TX-2 确认器用于计算。研究人员正在收集一些关于网络操作的统计数据。⁴⁹ (1967 年 2 月至 4 月)

玛利尔和罗伯茨的论文在通信协议的描述方面对玛利尔的 CCA 技术报告做出了补充。它将消息分成块，每块最多包含 119 个字符，每个块都有针对系统或用户的开始字符、停止字符以及

校验和，同时还有用来确认和重发请求的ACK和NACK字符以及其他一些控制字符。罗伯茨后来将这些消息块称为分组，不过当时并不存在分组交换。⁵⁰该协议的消息块比较小，这样就可以通过对一个或几个块的内容进行重新传输的方式进行错误恢复，而不需要重新发送整条很长的消息，节省了成本。要使用这个方法只需要添加少量控制字符即可，最多只占日常开销的百分之几。在1988年的采访中，罗伯茨说，当时自然就想到了如此命名：“我们想到了块。这就是计算机工作的方式。”⁵¹

在林肯实验室归档的CCA备忘录中提到了一种更高效的协议，它允许在先前的消息块被确认之前就发送第二个消息块，不过在该确认到达之前，第二消息块的发送不会被终止。如果接收机报告第一个块中存在错误，则第二个块的发送将立即终止，重新发送第一个块。⁵²只要在发送下一个块的同时，能够接收到针对该块的每个ACK，该协议就允许连续无延迟地传输消息块。备忘录作者比尔·曼恩(Bill Mann)回想起，⁵³当时在SDC负责网络终端的人员认为最好不要实现这个更复杂的协议。他们采用了玛利尔和罗伯茨论文中较为简单的一种方案，该方案也记录在CCA备忘录中。⁵⁴曼恩表示，他对此表示反对，这种方案的效率太低，而当时罗伯茨的说法是，该项目只是一个概念验证

测试，性能高低并不重要，这和他后来的说法形成了一个有趣的对比，他后来评论该网络速度太慢，不够实用。当然，罗伯茨的这两种回应在相应的背景下都是合适的。

在1966年6月的备忘录中，罗伯茨⁵⁵概述了实验的建议时间表。在6月和7月里，他们将设置TX-2电话硬件，修改APEX操作系统以便和Q-32建立连接，运行一个测试程序，让SDC的调制解调器将字符回传至TX-2，并另外编写一个程序，使TX-2终端看起来更像Q-32控制台。在8月，他们将以一种“有意义的方式”运行一个使用SDC程序的TX-2程序演示。报告中还提到该项目的下一阶段，并表示项目时间表可能还需要修订。但没有迹象表明是否有必要进行修订。

两份进展报告描述了进一步的计划和已经实现的内容。第一个报告内容是关于1967年1月的会议，玛利尔在会上报告了网络计划并列出了16项实验。⁵⁶其中五项实验使用Q-32的TX-2程序，四项使用TX-2的Q-32程序，七项使用哈佛大学的DEC 338图形终端(PDP-8的改进型)连接到这两台计算机之一。其中一些雄心勃勃的实验中试图使用图形显示器。备忘录中还指出，这些测试将需要实施一些不符合网络原理临时措施，比如需要对远程程序进行修改、不使用标准显示语言等。Q-32和TX-2都在开发图形项目(包括一些由罗伯茨主持的项目)，使这两台机器成为远程实验的候选对象。

在1967年的报告中，有评论指出，使用Q-32的三个TX-2程序已经运行，但由于调制解调器存在缺陷，使它们的可靠性不足。研究人员也开始着手Q-32上的三个与之对应的程序的调试工作，但其余的程序，包括图形程序在内，都没有开始运行。DEC338计算机并没有成为这次网络实验的一部分，不过后来在罗伯茨的ARPA办公室装设了一台。⁵⁷

关于SDC这边的网络的消息就更少了。由查尔斯·巴贝奇研究所收集的Burroughs论文包括来自SDC的一些技术备忘录，尽管Burroughs获得

在1967年的报告中，
有评论指出，使用Q-32的
三个TX-2程序已经运行，
但由于调制解调器存在缺陷，
使它们的可靠性不足。
研究人员也开始着手Q-32上的
三个与之对应的程序的调试工作，
但其余的程序都没有开始运行。

了这些备忘录，但是其中似乎没有关于网络的任何内容。SDC的1966年度研究报告⁶¹中的一章是由魏斯曼编写的，内容是关于编程系统，其中提到了两个网络项目。一个是位于加州大学伯克利分校和SRI以及SDC的SDS 940之间正在设计过程中的计算机的网络，但是该网络项目显然从未成功实现。另一个项目是TX-2连接。报告中强调，该项目允许一个系统调用另一个不同的程序，并补充说，这种方法“可能是数据处理社群对他人的程序和工作实现真正共享的唯一途径。”关于分时网络的章节中，列出了玛利尔提到的西联1200 bps线路，以及CCA合作对SDC PDP-1和TX-2的操作系统进行修改一事，这些内容与CCA的文档内容一致。它对TX-2没有使用Q-32程序的解释是，后者的APEX操作系统的资源主要用于图形显示程序，需要为这些程序开发协议。Q-32将在1967年被IBM 360取代，虽然报告认为网络项目将会转移至新系统中，但后来的事实表明，这种预见并未成为现实。

魏斯曼表示，网络确实对命令和数据分别使用了单独的线路，即对前者使用低速线路，对后者使用高速线路。⁶²这对于SDC来说已经是一件自然而然的事了，因为SDC已经在早期的网络上使用了这种简单的设计。然而，在现有的SDC文献中，笔者没有找到对双线网络的记载，而在林肯实验室的材料中，唯一一次提及双线网络的可能就是1966年11月的林肯实验室进展报告中提到用于连接其他计算机的低速通道，也就是指在TX-2上的两个新的控制台。这是对双线网络这一说法的少有的支持，不过如果报告内容是说两个控制台分别用于低速通道和1200 bps的电话线，那么就与魏斯曼的回忆的一致。

魏斯曼还回忆说，罗伯茨不喜欢两条线路的过高成本，认为双线路模式不是网络开发的可行模式。因此，最初可能存在使用双线路网络的尝试，但是这种尝试没有发展到足以需要在进度报告中记录的程度，因为这一模式后来被使用能够分离命令和数据协议的单线路网络所代替。

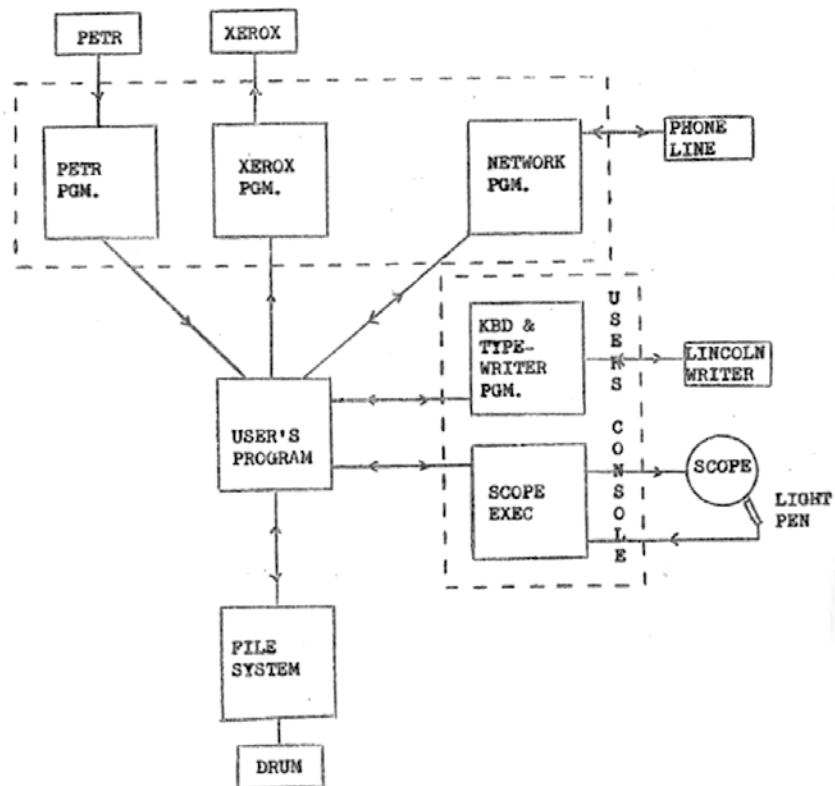


图2. 网络和TX-2系统。此框图中看不出系统是否具有自动拨号功能。（图片由MIT林肯实验室档案提供）

LOG IN CCA

```
CCA IN 425 PGS , THU 16 FEB 67 1320.13
BTR 2
```

AT

```
DIALING SDC
CONNECTED
1967 FEB 16 1021.9 LISP 1.5 M2.6 7JA13 RL1447
(EX CD)
TLST
BOPS
UOPS
(COMP EVBT TEST)
```

X=7.012345

```
(SIN X) * (SIN X) + (COS X) * (COS X)
1.0000000
```

图3. 使用自动拨号的会话示例。粗体文本表示用户的输入，其余部分是APEX系统（如DIALING SDC）或Q-32（如Lisp 1.5线路）的输出。

我们尚不清楚林肯实验室-SDC的网络又继续使用了多长时间。罗伯茨于1966年底离开ARPA，但在1967年10月于田纳西加特林堡举行

Arpanet 和互联网的 运行情况表明，远程计算机 网络设施对大量用户具有 相当大的实用价值。

第一届操作系统原理研讨会上提交的一篇论文中，罗伯茨提到“现在用户已经在利用网络来提高他们的能力”。⁶³ 在1967年剩余的时间里，林肯实验室的档案中没有再出现关于网络的文件。1968年2月，CCA备忘录中再次提到了前文所说的已经实施的三个程序，虽然当时运行这些程序的网络并没有投入使用，因为SDC已将Q-32计算机退役。还有一些早期文件中提到自TX-2建立自动电话拨号系统的计划。玛利尔在关于1月会议的备忘录中说，拨号器还没有准备好，⁴⁴ 不过1967年3月的CCA备忘录中⁶⁴ 报告称拨号器已经完成，而且TX-2可以在网络呼叫时拨出号码，同时还能接收呼叫。图2所示为TX-2系统⁶⁵ 的框图，不过图中看不出其是否具备自动拨号能力。

玛利尔的最后一个进度报告中提到一个使用自动拨号的会话示例。⁶⁴ 图3所示是在TX-2上运行的AT程序。下面是他的程序注释：

输入“CCA”，记录我们现在的时间，东部时间1320.13。

键入“AT”以启动程序。程序将对SDC拨号、登录、载入LISP、发送LISP程序供LISP编译器编译、等待编译完成、然后等待来自Lincoln Writer的输入。

LISP于太平洋时间1021.9加载至Q-32上，加载时间点在登录TX-2后1.8分钟。这1.8分钟包括选择所需程序的时间，“AT”时间，以及将该选择键入系统的时间，（我记得没有人会在这时着急）。在东部时间1325.1时刻，即3.2分钟后，已编译的LISP程序传输完成，并进行了汇编。

我们现在输入X的值和一个待计算的函数。这里我们没有记录时间信息，但我们估计答案输出时间不足一秒。其中包括将代数字符串发送到Q-32的时间，将其编译成基本翻译器代码，将代码发回，对函数求值并输出答案。

在CCA的备忘录中列出了网络文件和收集关于其执行任务的统计数据，但没有迹象表明这些方案已被采用。该研究项目有两个成果，第一是给出了一个可以正常运行的网络连接和消息协议的简单示例，不过当时并没有继续使用下去。第二，该项目与SRI-SDC网络一样，也展示了对远程计算资源的利用。在玛利尔的演示中，程序被编译后，会在一秒钟左右的时间内给出结果，这对于洲际通信来说性能并不算低。如果能够知道发生了多少次网络错误就更有趣了，因为这对早期的SRI-SDC连接来说同样有用，但笔者还没有找到任何记录的信息。

SDC和林肯实验室的文档都涉及使用图形显示。参与者并没有开发必要的协议，在理想情况下，开发这些协议需要用到与设备无关的格式来传递信息，以便在每一端转换为设备特定的代码。虽然这个雄心勃勃的目标并没有达成，但是研究人员还是取得了显著的成就，作为资源共享早期范例的Arpanet在几年后得以建立。

其他建议的实验

在诺博格和奥尼尔关于早期IPTO网络合同的章节中，他们提到了几个其他实验，笔者在这里再多提供一些信息。在1964年与加州大学洛杉矶分校的第一份合同中，内容是加入该校位于健康科学中心的主计算中心，以及西部数据处理中心（一个本地联合机构）。该合同也涉及SDC的Q-32系统，但是三个单位之间并不想彼此协作，⁶⁷ 尽管ARPA建立初期形势乐观，但最终并没有任何结果。

前文提到，1966年有人曾建议将加州大学伯克利分校的SDS 940计算机、BBN和SRI的计算机实现互联，这显然是为了从单一计算机机型

网络的便利性中获益。虽然该建议未包括SDC系统，不过SDC仍于1966年6月举行了一次小组会议，制作了一份由韦恩·利钦伯格(Wayne Lichtenberger)编辑的设计文档。⁶⁸ 笔者没有找到这份文件的副本，编辑本人也不记得了，不过他说他的设计主题与当时的工作是一致的。⁶⁹ 没有迹象表明这项拟议的项目已经实施，笔者也未发现有资料提到SDS参与了这一项目。

IPTO还支持了卡内基梅隆大学(CMU)和普林斯顿大学和IBM公司对IBM 360/67s的项目。关于该项目的一份报告中⁷⁰ 引用了利钦伯格的报告中描述一个类似的项目，可能是因为两者都是用于实现一种计算机模型的联网。CMU的项目于至少作为一个原型于1968年实施。该项目使用点对点连接，这一点与Arpanet不同。网络工作小组在关于Arpanet主机到主机协议一事的征求意见中(RFC 33)提到，CMU的项目没有影响到Arpanet。该意见中还说：

加州大学伯克利分校、麻省理工学院、林肯实验室和SDC(都由ARPA提供支持)的分时研究对网络的设计产生了相当大的影响。智能感知，时间共享计算机的ARPA网络是更早期的分时概念的自然延伸。⁷¹

可能的1965年实验

在包括计算机历史博物馆的互联网时间表在内的几项在线或公开资料中，把SDC和林肯实验室的传输实验时间记成了1965年。⁷² 虽然这是错误的，但还有两次与罗伯特·泰勒的口述历史访谈中提到了1965年的实验。在2008年在接受计算机历史博物馆的采访中，⁷³ 泰勒表示他在1965年曾要求罗伯茨进行一次“位可靠性实验”，以评估网络分时系统的可行性，罗伯茨将这项工作分包给了玛利尔。该实验使用的是SDC和林肯实验室分时系统，泰勒说，因为他正在资助这两家单位。1989年在巴贝奇学院的一次口头历史访谈中¹⁰ 有一个与此类似的说法：泰勒于1965年向罗伯茨问了一些关于网络可靠性问题，并且罗伯茨让CCA“通过电

话线在SDC和林肯实验室之间来回发送一些位信息”。

在最近的一封电子邮件⁷⁴中，泰勒强调，这一测试不是为了网络应用而将Q-32和TX-2相连接，而主要只是将两个系统“作为位发送器和接收器”来使用，只是足以保证跨国长距离传输的可靠性。泰勒在采访中没有给出更准确的日期。1965年底，他曾在一封电子邮件中说，他认为跨国传输是可行的。他认为这一想法是基于SDC和林肯实验室实验的结果，这与他在1966年2月启动Arpanet项目中的观点一致。

程序被编译后，会在一秒钟左右的时间内给出结果，这对于洲际通信来说性能并不算低。

罗伯茨曾在电子邮件中⁷⁵ 提到这次网络测试，他说，他需要在1965年确定电话线传输的突发错误率，来决定网络消息块的适当大小。他说当时很可能根本没有使用调制解调器(TX-2直到1966年中才用上第一台调制解调器)，不过他使用一个模数转换器将电话线连接到TX-2，以前他曾使用过这一技术来帮助Amar Bose测量他的卧室与扬声器设计频率响应。他认为他把测试结果报告给了泰勒。

1965年和1966年林肯实验室报告中都没有提到这次位传输实验。因此，泰勒的这些回忆不足以确切证明关于1965年实验的确切情况。根据泰勒和罗伯茨的回忆，这次测试是规模很小的简短测试，可能时间太短以致没有记录。(今天回过头来，令我们感到幸运的是，任何位传输测试的结果都表明洲际网络连接是可行的，而且这一技术的应用促使我们不断追求更快的传输速度。)

结论

在本文中，笔者一共描述了三个网络实验，这些实验都受到了几任IPTO主任(里克莱德、萨瑟兰和罗伯茨)主持的ARPA项目的推动，目的是增进各计算机中心之间的资源共享和协作。这些实验持续时间都不长，相关记录也很少。1963年SDC-SRI连接实验是有趣的首次实验，但更重要

加州大学伯克利分校、
麻省理工学院、
林肯实验室和 SDC 的
分时研究对网络的设计产生了
相当大的影响。

的是它提供了一种“可能性”：一个简单有效的设计，可以提供连接不同计算机的相关经验，并接着成功使用了远程资源。它还为SDC员工提供了利用网络的经验，尽管关于这种经验没能为后来的SDC-林肯实验室的实验做出多少贡献。

既然这次实验是首次使用分时网络的实验之一，为什么它的知名度相对较低呢？一个可能的原因是，该实验在最初演示之后即告结束。而另一个原因，正如克拉克·魏斯曼在他的电子邮件中所说，尽管SDC也进行了一些信息检索的相关研究，但SDC系统的主要作用是支持在SRI和林肯实验室等地进行的研发。根据魏斯曼的说法，Q-32系统是国家资源，被一些大学的研究人员使用，他们通常通过远程拨号连接。虽然SDC的研究人员确实发表了关于分时问题的几篇论文（比如调度问题），但其中的细节不多，比如没有详细说明双线电话线网络的设置问题。

其他两个实验使用的是1966–1967年SDC与林肯实验室的连接。一个是对洲际电话线传输的简单通信协议的测试。罗伯茨对实验中的传输速度、成本和可靠性都不甚满意，这促进他去找到一种更好的方法。第三个实验是Lincoln Reckoner对Q-32上运行的Lisp软件的实际应用，目的是展示这种网络应用。虽然没有公布关于这种远程软件的任何使用情况，但这次实验可以说是三次实验中最重要的，因为它展示了如何在远程计算机上自动调用软件并将结果应用于本地。据我所知，直到1969年Arpanet建立之后，人们才进一步开发出

了更多的这类用途。

这些由IPTO资助的实验主要是IPTO主任对计算机网络推广的结果。

这些由IPTO资助的实验主要是IPTO主任对计算机网络进行推广的结果。有些参与实验的人，如罗伯茨，是通过与里克莱德的交流对这些实验产生了兴趣，而另一些参与者则是在寻求自己的项目：比如伯恩的信息检索研究和玛利尔的搜索研究，后者是为了寻求合同订单。在很大程度上讲，进行这些网络实验的初始动机是自上而下的。有多个记录表明，当时大型计算机的研究人员并不太愿意分享这些计算机。因此，在追溯这段关于计算机网络的历史时，恐怕要将最初原因归结于少数几个人的远见上，这可能要比其他诸多领域发展史更符合事实。连续性当然也发挥了作用，例如，对事件和需求共同期待，导致巴兰和戴维斯去开发数据包交换技术，韦斯利·克拉克（Wesley Clark）利用IMP（接口网络处理器）的概念来简化Arpanet的实现，此外还有戴维斯的团队在1967年的加特林堡会议上对罗伯茨产生的影响，罗伯茨正是在会上提出了组建Arpanet的想法。玛利尔和罗伯茨的计划进行实验的文章经常被当作第一个长距离计算机网络的诞生，其最重要的原因可能是罗伯茨在论文发表后不久就领导了Arpanet项目。这个关于“第一”的复杂故事有许多问题。早期的SDC-SRI网络所取得的短暂成功，可能影响了林肯实验室随后对双线网络的最初设计，随后该实验室就在利用这一设计来分析如何利用带通信协议的电话单线路。这是一个很好的范例，表明一个没能继续下去的项目本身也可以对后续项目产生促进作用。

最后，由罗伯茨曾经供职的林肯实验室发布的报告中有关图形的一章中曾描述了Lincoln Reckoner对SDC Lisp程序的使用情况：

网络承包商（CCA）收到调查所需的服务设施的要求，以使该连接可以为普通用户所用。在解决关于文档、服务设施和指令方式的问题之前，远程网络计算机对大众用户的实用价值可能不大。⁷⁶

我们还没有完全解决文档和指令方式的问题，但是随后的Arpanet和互联网的运行情况表明，这些网络设施对大量用户确实具有相当大的实用价值。

致谢

笔者在此感谢埃德·科夫曼、拉里·罗伯茨、罗伯特·泰勒和克拉克·魏斯曼对网络的讨论，以及查尔斯·伯恩和伦恩·蔡廷提供的关于斯坦福大学研究所的信息。查尔斯·巴贝奇研究所的工作人员提供了50年前的CCA技术报告，MIT林肯实验室热心人士诺拉·萨尔迪瓦尔提供了CCA备忘录。来自匿名评阅者的评论也提高了本文的质量。笔者在此一并致谢。

参考文献

1. M. Campbell-Kelly, "Data Communications at the National Physical Laboratory (1965–1975)," *Annals of the History of Computing*, vol. 9, no. 3, 1988, pp. 221–247.
2. J. Pelkey, *A History of Computer Communications*, 1968–1988, 2007; www.historyofcomputercommunications.info.
3. J. McCarthy to P.M. Morse, "A Time Sharing Operator Program for our Projected IBM 709," memo, Massachusetts Inst. of Technology, Jan. 1959.
4. C. Strachey, "Time Sharing in Large, Fast Computers," Proc. IFIP Congress, 1959, pp. 336–341.
5. D. Hemmendinger, "Messaging in the Early SDC Time-Sharing System," *IEEE Annals of the History of Computing*, vol. 36, no. 1, 2014, pp. 52–57. This article discusses some interactive uses of the SDC system.
6. R.R. Everett, C.A. Zraket, and H.D. Benington, "SAGE—A Data-Processing System for Air Defense," *Annals of the History of Computing*, vol. 5, no. 4, 1983, pp. 330–339.
7. J.C.R. Licklider, "Memorandum for Members and Affiliates of the Intergalactic Computer Network: Topics for Discussion at the Forthcoming Meeting," memo, ARPA, 23 Apr. 1963.
8. A.L. Norberg and J.E. O'Neill, *Transforming Computer Technology: Information Processing for the Pentagon, 1962–1986*, Johns Hopkins Univ. Press, 1996.
9. Norberg and O'Neill, *Transforming Computer Technology*, p. 156.
10. R.W. Aspray, "Oral History Interview with Robert W. Taylor," CBI OH 154, Charles Babbage Inst., Feb. 1989; <http://purl.umn.edu/107666>.
11. J. Abbate, *Inventing the Internet*, MIT Press, 1999, pp. 48–49.
12. K. Hafner and M. Lyon, *Where Wizards Stay Up Late*, Simon & Schuster, 1996, p. 68. 13. Adele Goldberg, ed., *A History of Personal Workstations*, Addison-Wesley, 1988.
14. G.D. Cole, "Computer Network Measurements: Techniques and Experiments," UCLA-ENG 7165, Univ. of California, Los Angeles, 1971.
15. J.I. Schwartz, E.G. Coffman, and C. Weissman, "A General-Purpose Time-Sharing System," Proc. ACM Spring Joint Computer Conf., 1964, pp. 397–411; doi:10.1145/1464122. 1464163. Two Early Interactive Computer Network Experiments 22 IEEE Annals of the History of Computing
16. C.P. Bourne, "Research on Computer Augmented Information Management," report no. ESD-TDR-64-177, Electronic Systems Division, US Air Force Systems Command, Nov. 1963. NTIS report no. AD-432 098.
17. C.P. Bourne, "40 Years of Database Distribution

- and Use: An Overview and Observation,” Miles Conrad Memorial Lectures, Nat'l Federation of Advanced Information Services, Feb. 1999; <https://nfais.memberclicks.net/assets/docs/MilesConradLectures/bourne1999.pdf>.
18. C.P. Bourne and T.B. Hahn, *A History of Online Information Services, 1963–1976*, MIT Press, 2003.
19. D. Engelbart, “The Augmented Knowledge Workshop” *A History of Personal Workstations*, A. Goldberg, ed., Addison-Wesley, 1988, pp. 191–192.
20. Bourne, “Research on Computer Augmented Information Management,” p. 43.
21. D. Engelbart, “The Demo,” 1968; <http://web.stanford.edu/dept/SUL/library/extra4/sloan/mousesite/1968Demo>.
22. J.I. Schwartz, “The SDC Time-Sharing System, Part 1,” *Datamation*, vol. 10, no. 11, 1964, pp. 28–31.
23. C. Weissman to D. Hemmendinger, email, 10–14 Feb. 2013.
24. L. Chaitin, interview by D. Hemmendinger, 18 Mar. 2013.
25. “Research and Technology Division Report for 1965,” tech. memo TM-530/009/00, System Development Corp., 1966.
26. Y.S. Loy, “System and Logical Design of a 2000 BPS Data Terminal Unit for Q-32/PDP-1 Operation,” final report, SDC document TM-2552, 19 July 1965.
27. J.I. Schwartz and C. Weissman, “The SDC Time-Sharing System Revisited,” Proc. 22nd ACM Nat'l Conf., 1967, pp. 263–271; doi:10.1145/800196.805996.
28. T. Marill and L. G. Roberts, “Toward a Cooperative Network of Time-Shared Computers,” Proc. AFIPS Fall Joint Computer Conf., 1966, pp. 425–431.
29. L.G. Roberts, “The ARPANET and Computer Networks,” *A History of Personal Workstations*, A. Goldberg, ed., Addison-Wesley, 1988, p. 145.
30. Norberg and O'Neill, *Transforming Computer Technology*, p. 158. Hafner and Lyon's book has a similar account, although neither book cites a source. F. Heart et al., “A History of the ARPANET: The First Decade,” BBN report, 1981, says only that in 1965 Marill and CCA were given the networking-study subcontract and that the CCA study reported in 1966 was done in late 1965. It adds, “Later in 1966, CCA received another contract to carry out the linking of the Q-32 and the TX-2.”
31. T. Marill, “A Cooperative Network of Time-Sharing Computers,” tech. report 11, Computer Corp. of Am., June 1966.
32. Heart et al., “A History of the ARPANET,” p. III-10.
33. D. Walden and R. Nickerson, eds., *A Culture of Innovation*, Waterside Publishing, 2011.
34. Taylor had tried to get Roberts to run the Arpanet project in early 1966, but Roberts didn't want to leave Lincoln Lab, where he was working on computer graphics. In his oral histories at both the Babbage Institute and the Computer History Museum, he describes how, in the fall of 1966, he put pressure on the head of the Lincoln Lab, which depended on ARPA funding, to persuade Roberts to take the job. Taylor said that he “blackmailed him into fame,” a remark to which Roberts refers in “The Arpanet and Computer Networks,” p. 145.
35. Norberg and O'Neill, *Transforming Computer Technology*, p. 33.
36. L.G. Roberts, “Interview, 1988-06,” James L. Pelkey Collection: History of Computer

- Communications, lot X5671.2010, accession 102746626, Computer History Museum. p. 3.
37. See <http://archive.org/details/arpanet>.
38. See <http://archive.org>.
39. See <http://archive.org/details/MailToKatieHa-fner>.
40. T. Marill, "A Nationwide Cooperative Computer Network," tech. report 11, draft 3, Computer Corp. of Am., 28 Feb. 1966. Available in MIT Lincoln Laboratory Archive.
41. Digital Equipment Corp., "Programmed Data Processor-1," 1960.
42. Norberg and O'Neill, Transforming Computer Technology, p. 158.
43. B. Williams, "Coercing the Network," Dr. Dobbs J., vol. 26, no. 10, 2001; www.drdobbs.com/coercing-the-network/199200749.
44. T. Marill to J.L. Mitchell, "Progress Report on Networking," memo, Computer Corp. of Am., 14 Feb. 1967.
45. Lincoln Laboratory, "Quarterly Technical Summary, General Research," Feb. 1966, p. 6.
46. Lincoln Laboratory, "Quarterly Technical Summary, General Research," May 1966, pp. 5–6.
47. Lincoln Laboratory, "Quarterly Technical Summary, General Research," Nov. 1966, p. 15.
48. Lincoln Laboratory, "Quarterly Technical Summary, General Research," Feb. 1967, p. 8.
49. Lincoln Laboratory, "Quarterly Technical Summary, General Research," May 1967, p. 6.
50. D.W. Davies et al., "A Digital Communication Network for Computers Giving Rapid Response at Remote Terminals," Proc. 1st ACM Symp. Operating System Principles, 1967. Roberts adopted the term "packet" from the presentation of this paper. A discussion with Scantlebury also persuaded Roberts to use 50 Kbps phone lines for the Arpanet instead of the much slower ones that he'd planned to use. July–September 2016 23
51. L.G. Roberts, "Interview, 1988-06," James L. Pelkey Collection: History of Computer Communications, lot X5671.2010, accession 102746626, Computer History Museum. p. 7.
52. W. Mann, "Message Format and Protocol," memo, Computer Corp. of Am., 23 May 1966. Available in MIT Lincoln Laboratory Archive.
53. W. ManntoD. Hemmendinger, email, 26 June 2014.
54. H. Murray to L.G. Roberts, "Q-32/TX-2 Message Format - Revision II," memo, Computer Corp. of Am., 27 July 1966.
55. T. Marill to L. Roberts, "Schedule for Networking Demonstrations: Phase 1A," memo, Computer Corp. of Am., 13 June 1966.
56. T. Marill to J.L. Mitchell, "Progress Report on Networking," memo, Computer Corp. of Am., 14 Feb. 1967.
57. R.M. Gray, "A Survey of Linear Predictive Coding: Part I of Linear Predictive Coding and the Internet Protocol," Foundations and Trends in Signal Processing, vol. 3, no. 3, 2010, pp. 1–147. This document reports that in 1967 Danny Cohen had connected a DEC 338 in Roberts' Pentagon office to the TX-2. It adds that the point-to-point telephone line was "not yet networking." Another view is in a message by Cohen (11 Apr. 1999) in an Internet mailing list that mentions working on this connection: "It sure was the first ARPA networking" (see <http://seclists.org/interesting-people/1999/Apr/52>). There is no report, however, on uses of the link.
58. A.N. Stowe et al., "The Lincoln Reckoner: An Operation-Oriented On-line Facility with Distributed Control," Proc. AFIPS Fall Joint Computer Conf., 1966, pp. 433–444.
59. Weissman said that the program was basically one that he wrote for his book, Lisp Primer: A

- Self-Tutor for Q-32 Lisp 1.5, SDC, 1965, and that this arrangement worked well, providing a demonstration of the value of networking for program sharing. C. Weissman to D. Hemmendinger, email, Feb. 2013.
60. T. Marill to J. Mitchell, "Progress Report on Networking," memo, Computer Corp. of Am., 3 Apr. 1967.
61. C. Baum, ed., 'Research and Technology Division Report for 1966,' tech. memo TM-530/010/00, System Development Corp., 1967.
62. C. Weissman to D. Hemmendinger, email, 10 Feb. 2013.
63. L.G. Roberts, "Multiple Computer Networks and Intercomputer Communication," Proc. 1st ACM Symp. Operating System Principles, 1967, pp. 3.1–3.6; doi:10.1145/800001.811680. According to Roberts' online version, this paper was written in June 1967.
64. T. Marill to J.L. Mitchell, "Progress Report on Networking," memo, Computer Corp. of Am., 14 Feb. 1967.
65. H. Murray to J. Mitchell, "Functional Description of the Network System," memo, Computer Corp. of Am., 30 Mar. 1967.
66. T. Marill to J. Mitchell, "Progress Report on Networking," memo, Computer Corp. of Am., 3 Apr. 1967.
67. Norberg and O'Neill, *Transforming Computer Technology*, p. 156.
68. ARPA, "Tentative Specifications for a Network of Time-Shared Computers," document no. 40.10.130, 9 Sept. 1966.
69. W. Lichtenberger to D. Hemmendinger, email, 23 Aug. 2013.
70. R.M. Rutledge et al., "An Interactive Network of Time-Sharing Computers," Proc. ACM 24th Nat'l Conf., 1969.
71. S. Crocker, S. Carr, and V. Cerf, "New Host-Host Protocol," Request for Comments 33, Network Working Group, 12 Feb. 1970; <https://tools.ietf.org/html/rfc33>.
72. Computer History Museum, Internet History, 1962–1992, www.computerhistory.org/internethistory/. Other sources include Pelkey's online book, *A History of Computer Communications*; L. Lambert et al., *The Internet: A Historical Encyclopedia*, 3 vols., ABC-CLIO, 2005; and L. Roberts' Internet chronology (<http://packet.cc/internet.html>), which says "This experiment was the first time two computer talked to each other."
73. P. McJones, "Oral History of Robert W. Taylor," oral histories lot X5059.2009, Computer History Museum, Oct. 2008.
74. R. Taylor to D. Hemmendinger, email, 26 Feb. 2013.
75. R. Taylor to D. Hemmendinger, email, 30 Oct. 2014.
76. Lincoln Laboratory, "Semiannual Technical Summary, Graphics," May 1967, p. 2.

David Hemmendinger 是纽约斯克内克塔迪联合大学计算机科学退休教授, 曾在该校教授计算机程序语言、计算机架构、并行处理、计算机史等课程。他曾经从事数学史和科学史研究, 过去 20 年来从事计算机史研究。他是《计算机科学百科全书》第四版的编写者之一。他从耶鲁大学获博士学位, 现任 IEEE Annals 副主编。联系方式: hemmendd@union.edu。



搜索你的工作机会

IEEE Computer Society 招聘可以帮你轻松找到IT、软件开发、计算机工程、研发、编程、架构、云计算、咨询、数据库很多其他计算机相关领域的新工作。



新功能：找出那些建议或要求拥有IEEE CS CSDA或CSDP认证的工作！

点击www.computer.org/jobs,
从全世界的雇主那里搜索技术工作岗位和实习机会。

<http://www.computer.org/jobs>

IEEE  computer society | JOBS

IEEE计算机协会是AIP Career Network的合作伙伴。其他合作伙伴包括《今日物理》杂志（Physics Today），美国医学物理协会（American Association of Physicists in Medicine），美国物理教师协会（American Association of Physics Teachers），美国物理学会（American Physical Society），AVS科学和技术学会（AVS Science and Technology），物理学生协会（Society of Physics Students）和Sigma Pi Sigma。

《机器、语言和计算》 在MIT的诞生历史

撰文 Peter J. Denning、Jack B. Dennis

从前，在麻省理工学院（MIT）的电子工程和计算机科学系，曾经有一门编号6.253，名为计算理论模型的课程。这门课是在计算机科学还不是那么庞大的一个领域时开设的，当时在人们看来，所有的计算机科学家都应该知道包括理论、编程、架构和系统等在内的一切事情。这个领域不同分支之间的竞争才刚刚出现。

在20世纪60年代中期，诺姆·乔姆斯基（Noam Chomsky）和他在MIT语言学系的学生提出了大量新见解。他们发现了形式语言和自动机（之间未曾被察觉的联系。特别值得一提的是，他们提出每一类自动机——有限状态自动机、下推自动机、线性有界自动机和图灵机，都可以用它们可以识别的语言类别来表征。他们发现，每一类语言都可以通过定义它所需的形式文法结构来描述。他们也发现，可以从文法构建自动机，或从自动机构建文法。例如，如果下推自动机可以识别出给定的输入字符串是用某种语言写的，那么这种语言就可以归类为下推自动机可识别。上下文无关文法可以被转换成对能够识别某种语言的下推自动机的描述，同样，上下文无关语法也可以从下推自动机构建出来。这些发现对于构建高级语言（例如Algol）的编译器极有价值。

杰克·丹尼斯（Jack Dennis）当时是MIT计算机科学实验室的教职员，他在研讨会上见到乔姆斯基的学生展示他们的研究成果。丹尼斯深信，计算机科学家很有必要了解这些理论。当时，与此相关的材料只有马文·明斯基（Marvin Minsky）的教案（后来编纂成书，由Prentice Hall出版¹，还有几本关于开关理论或逻辑电路的书。这些书也

没有把它们讨论的自动机看作是语言识别器。正是这种缺失促使杰克决定开设6.263课程。戴维·库克（David Kuck）和他一起设计了课程内容，并在1965年春开始教授这门课。1966年，彼得·丹宁（Peter Denning）成了他的助教，并很快就开始参与改善和拓展教案、编写习题集和讲课。

1967年，Prentice Hall的计算机科学编辑约翰·戴维斯（John Davis）注意到了6.253课的教案，建议我们（杰克和彼得）根据教案写一本书。当时，我们设想可以在1968年完成全部书稿。不幸的是，这个设想未能实现。彼得在1968年要写他的博士论文，然后他在普林斯顿大学获得了电子工程教职。而杰克发现，计算机科学实验室理论部门的成员不相信一伙做系统的家伙能教好一门理论课或写好一本理论书。我们不是“有资格证”的理论计算机科学家！因为想专心设计一门计算机系统课程（编号6.232），在彼得离开后，杰克招来了约瑟夫·夸利兹（Joseph Qualitz）来负责6.263课。约瑟夫继续教授这门课，并帮助完成了那本书。6.253课一直开设到1973年秋季。

因为有各种事情分心，而且我们对这门课程教学的参与也减少了，我们一共用了10年时间才写完了那本书。《机器、语言和计算》（Machines, Languages, and Computation, 简称MLC）在1978年出版，比我们当初估计的出版时间晚了10年²。

在普林斯顿，彼得与杰克在MIT的遭遇一样，也从理论计算机科学家那里收到了类似的反应。普林斯顿的教职员中有一些最伟大的理论计算机科学家（例如杰夫·厄尔曼，Jeff Ullman），

还有一些理论专家与普林斯顿关系密切（如阿尔·阿霍，Al Aho 和约翰·霍普克洛夫特，John Hopcroft）。霍普克洛夫特和厄尔曼也在写一本有关自动机和语言的书，这本书在1973年出版³。彼得从他的MLC书稿中选取了一些内容，在自动机理论课上讲授。但霍普克洛夫特和厄尔曼的书更受普林斯顿师生的欢迎。杰夫和彼得时常谈论MLC这本书的构思。杰夫喜欢这本书的某些部分，但非常讨厌我们把上下文无关文法和下推自动机联系起来的解释。他说，这相对于形式语言中使用的标准范式是一种倒退。有一天，彼得找到杰夫，说他想到了一个简短的证明方法，可以证明优先文法确定无疑地是上下文无关的。此前唯一一个已有的证明占据了苏珊·格雷厄姆（Susan Graham）博士论文的大部分篇幅。彼得想要一个简短的证明过程，在MLC书中只占两三页，当时这本书还是处在草稿阶段。彼得已经用了好几年的时间寻找简洁的证明方法。彼得开始向杰夫演示他的证明，起初杰夫想把他打发走，说：“你的证明里有一个漏洞。

“之后有一天杰夫说：”我想你的证明是对的，你对下推自动机的非常规倒退解释找到了关键。“彼得与阿霍和厄尔曼合作写了一篇论文，在1972年发表，描述了这个更短的证明⁴。做系统的家伙找到了理论计算机科学家没能找到的简短证明，但理论家认为这只是狗屎运罢了，因为我们让一个关键定义倒退了。

在教授这门课程的过程中，我们发现一个简短的证明方法，来证明对应问题的不可计算性⁵。这个证明基于这样一种方法，我们把图灵机所允许的操作赋予一组多米诺骨牌，这样就有了一组可以模拟图灵机计算的多尼诺骨牌，当且仅当这台机器暂停的时候⁶。我们把这个证明写成了一篇论文，并投给了一个理论期刊。很快，我们听说罗伯特·弗洛伊德（Robert Floyd）也找到了一个简短的证明，即将发表。这个证明并没有采用多米诺骨牌方法。因此我们从未把自己的证明发表出来，只是写到了我们的MLC书里⁶。多年以后，我们听说其他教师也在用多米诺骨牌式的方法来向学生教授这个证明，

表明（其实我们早已知道了）我们的证明方法更容易为学生接受，即使理论家们已经有了更简短的数学证明。

6.253课和MLC这本书，可以让人系统地认识自动机和语言。我们发现，我们的学生很难掌握某些定义，诸如“一个有限状态自动机是一个由……构成的五元组”。他们更喜欢让我们绘制状态图，然后用简单指令组成的程序来描述机器的运行过程。在我们布置的家庭作业中，我们让学生为四类机器中的每一种编写程序。我们的方法不像霍普克洛夫特和厄尔曼的书那么紧凑，数学也没有他们那么优美。我们尽力让课程更易为学生接受，并让他们有过一个系统的认识。

我们认为，一个更重要的区别是霍普克洛夫特和厄尔曼的书中有一个关于计算复杂性这个新兴领域的章节。我们一度曾考虑添加一个关于复杂性的章节，但最终还是没有这样做，因为要把原有章节目录写圆满，我们面临的困难已经够多了，而目录是复杂性领域兴起之前就定下来的。现在回想起来，这个遗漏对于我们的书是一个战略性错误。

尽管有这些困难，这本书还是写得不错的，一直到1990年还在继续印刷。对于我们真正关注的重点——计算机系统的建模和分析，我们写得要更好一些。理论计算机科学家会来找我们，以了解关于当前某个模型的假设是否与现实相符，还有我们是否认为有人会对某个形式结果感兴趣。结果还是很好的。

参考文献

1. M. Minsky, *Computation: Finite and Infinite Machines*, Prentice Hall, 1967.
2. P. J. Denning, J.B. Dennis, and J. Qualitz, *Machines, Languages, and Computation*, Prentice Hall, 1978.
3. J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*,

- Addison-Wesley, 1979.
4. A. Aho, P.J. Denning, and J.E. Hopcroft, "Weak and Mixed Strategy Precedence Parsing," *J. ACM*, vol. 19, no. 2, 1972, pp. 225–243.
 5. The correspondence problem concerns two lists of code words for the same message source and asks whether there is a message that encodes the same in the two codes. For example, with Code1 $\frac{1}{4} \{a, ab, bba\}$ and Code2 $\frac{1}{4} \{baa, aa, bb\}$, the message (3,2,3,1) encodes as bbaabbbaa in Code1 and as bbaabbbaa in Code2. Therefore (3,2,3,1) is a solution for these two codes. The general decision problem is for any two codes of the same message source: is there a message encoding the same in both codes?
 6. A domino is a block with a string on the top and another string on the bottom. For the correspondence problem, we make a set of domino types with a codeword on the top from Code1 and on the bottom from Code2. A solution to the problem exists if we can find a sequence of these dominoes where the string spelled out along the tops matches the string spelled out along the bottoms. It is possible to make a set of dominoes that spell out Turing machine tapes and control states and represent a Turing machine move from the top to bottom string on a domino. The tops and bottoms match if and only if the Turing machine represented halts. Thus, a solution to the correspondence problem would solve the halting problem. The Wikipedia article, "Post correspondence problem," refers to a 2005 book chapter by Michael Sipser, "A Simple Undecidable Problem," *Introduction to the Theory of Computation*, 2nd ed., Thomson Course Technology, 2005, pp. 199–205. Our MLC book contains a similar proof discovered over three decades earlier.
 7. Sipser, "A Simple Undecidable Problem," *Introduction to the Theory of Computation*, 2nd ed., pp. 199–205.

Peter J. Denning 是麻省理工学院计算机科学与工程荣誉退休教授。联系方式 dennis@csail.mit.edu。

Jack B. Dennis 是海军研究生院的杰出教授和计算机科学系主任，也是塞布罗夫斯基创新与信息优势研究所所长。联系方式 pjd@denningassociates.com。



会议就在你的手中

IEEE计算机协会的会议发布服务（CPS）现在可以提供组织会议的移动应用了！让会议的日程、会议信息和论文列表在你的与会者手中的设备上显示。

会议的移动应用可在安卓设备、iPhone、iPad和Kindle Fire上运行。



欲知更多信息，请联系cps@computer.org



一些往事

撰文 Laurie Robertson

进入咨询行业

1959年是我进入计算机这个（彼时的）新兴产业的第八个年头。我感到有些厌倦，并且离职了。这八年中，我从事过数据简化工作，参与设计过飞机、汽车和导弹。我设计过一个操作系统，在一个大型服务供职过，还做过一个大型商务汇编程序的项目经理，同时也为一家大型公司提供过一些内部咨询工作。我的工作一直很有意思，收入也属于顶尖水平，但这些工作从来没有完全让我感到过满足。

我曾经和两位同事一起成立了一家公司。作为创始人，我从这份事业中得到了相应的回报。但我很快对此失去了兴趣——我还要负责初创公司的管理和财务，这对我更钟爱的技术工作形成了干扰。后来，我因为遇到一个严重的道德问题离开了公司。我的个性似乎更适合从事咨询。一个公司在签下单一生意之前，必须做好相应的准备，必须先将大部分的官僚问题解决掉，并对你的声誉以及保证人进行核查。签单以后，唯一需要完成的就是合约中的工作。这才是我喜欢的工作环境。

奠定工作基础

在20世纪50年代，没有人教我怎么做咨询，市面上也没有相应的书籍。我必须快速学习如何给人提供服务，学习市场和广告知识、如何组织咨询业务，以及如何跟上这个快速变化的领域。我当时已经结婚，需要支撑我的家庭，所以我一边做着一份全职工作，一边和客户沟通我的第一份咨询合同。打了一个月的电话之后，我得到了这份合同，用一种新的源语言为我的前同事正在开发的一个商业编译程序进行编程验收测试。就在收到这份合同后不久，另一家公司给了我第二份合同。兰德

(RAND) 公司计算机部门的经理认为我无法胜任自由咨询工作（那时候他们都没听说过其他的自由咨询师），所以想抢在别人手下之把我变成他的雇员。

兰德公司对我有些特殊的兴趣。我在进入编译程序开发行业之前曾经做过一些涉密的工作，有一定的特殊权限，可以接触一些保密信息。军费投入使得许多领域（包括计算机）的发展走向高精尖，因此，为了能继续从事这份适合我的有趣工作，我希望能保留自己的至少一部分涉密权限。于是，我和兰德协商，降低了我的咨询日价，他们没有给我的工作限定具体的截止日期，并且接受了我的涉密限制。

兰德公司的咨询工作每周大约会占用我三天，每天的工作时间与我从事全职工作时无异。然而，咨询工作存在额外的风险，也不提供福利和假期。在和太太商量之后，我们决定承担这个风险。出这个决定意味着我必须单枪匹马完成合同。接下来，我还要寻找其他工作，把一周剩下的两天填满。

我曾经为IBM 704设计过一个堆叠式部件的计算机处理系统，还负责过霍尼韦尔 800 (Honeywell 800) 的一个商业编译程序项目。因此，我联系了洛杉矶地区的其他几个计算机生产商，得到了一份授课合同，为布劳斯公司 (Burroughs) 的B-5000团队讲授编译程序开发——这就又解决了一天的工作份额。我的涉密权限帮我填满了一周的最后一天：我得到了一份来自军队企业的合同。系统开发公司 (Systems Development Corporation, SDC) 不仅对我的工程及计算机经验感兴趣，我的身份对他们来说也非常合适：SDC只花了大概一周就将我的涉密权限转移了过去，而如果雇佣一个没有涉密权限的

人，他们大概要多花三到六个月的时间。

扩展业务

我很满意这样的安排。一周五天我都能做有趣事情，我所提供的的服务似乎也是他人无法取代的（至少在我的领域而言）。这种良好的自我感觉大约持续了24个小时，接着我便开始焦虑：如果拿不到新的合同怎么办？如果和客户发生严重的争执怎么办？独立自主的工作对我充满吸引力，可我得到的合同不过是按日工作量计酬的约定而已。合同里没有对我的保障、没有规定交付产品，也没有仲裁条款。我虽然时薪很高，但是否能得到工作全凭甲方意愿。另外，大部分合同在新历的年底就会结束，是否续约也取决于客户。面对这些问题我没有慌张，因为我一直能够从容应对充满竞争的环境。但与此同时，压力也促使我不断寻找新的工作机会。

1960年的时候，个人可以在技术出版物上的分类广告位上刊登名片（这和现在没多大区别）。我花钱登了广告，却从没接到过别人打来的电话。有一阵子，一连好几个星期我都在招聘广告中寻找机会，如果发现有重要的职位找不到合适人选，我就打算自告奋勇担任兼职，直到他们找到雇员。不过这也行不通。我也去参加了一些领域内的专业会议，买了不少酒去四处和人套近乎，我希望通过这种方式让同行们知道我还在这个行当里，好让他们在看到机会时能够想到我。这一招最终奏效了，但有时我得等上好几年才通过这种方式得到工作。我所从事的工作非常有趣，其中一些也非常前沿，所以我也通过给杂志撰稿、接受演讲邀请来保持自己在领域内的存在感。这同样是一个漫长的过程，但它最终确实奏效了。

要建立人脉，最好的方式就是最传统的方式。我请人吃午餐、下班后和人一起喝啤酒，尽全力地为客户勤奋工作。到年底时，我的第一批客户都和我续了约。但在咨询这个行业，能找到新客户才是最关键的事。最开始做咨询的时候，能否得到第一份合同决定了你能否能炒掉当时的老板自立门户，

而能否得到第二份合同（如果第一份合同是全职合同，这里也可以指拿到第一份合同的续约）才真正决定你能在咨询行业存活多久。令我意外的是，我在霍尼韦尔的客户将我推荐给了霍尼韦尔的军用产品部门，他们给我提供了一份工程师工作。他们正在通过全模拟为一艘核攻击潜艇的战术训练器建造一台原尺寸的指挥台样机（这当然是基于H-800的）。训练器类似一个游戏，只不过这个游戏的风险要高得多。

这份合同又将我的日程填满了。新工作刚刚启动，我就不得不加班加点。每周我有五天到客户的公司坐班，需要在24小时之内处理来电和来信。为了跟上行业的发展，我必须阅读相应技术期刊；作为一家之主，家务事也让我忙个不停。除此之外，我每月都要处理客户的账单，有时需要出差，还要定期填报税单。对于没有一定心理承受能力或者单身的人来说，这样的生活是难以想象的。我Hui在家里完成一些额外的咨询工作，我的妻子负责接听电话、打字、运营办公室，并且照料我们的孩子。每当我抱怨自己的工作时间太长，她就会提醒我：我给别人打工的时候同样得加班加点，但收入却没有现在这么多。

乘风破浪

入行大约六个月后，我对自己的状况进行了评估：我提供的服务是无可取代的，我拥有关于大型的IBM系统的经验，开发过一个操作系统、一个编译程序；我对自己新开发的两部计算机非常熟悉；我的日程也非常充实。然而，未来的程序员正在不断涌入这个行业，大学也在教授计算机操作。很快，这个领域就会变得非常拥挤，我必须在其中找到一个无人填补的空缺，并且抢占这个位置。

就在我为美国航天航空公司（Aerospace Corporation）提供咨询服务时，IBM宣布推出了一款可将非IBM设备外接至其大型计算机的通道上的产品。我和几个同行决定一起做点事情，于是就有了Direct Couple 配置器。

irect Couple 解决了IBSYS堆叠式工件环境

在接下来的 20 年中，
我从 IBM 接收到了涉及硬件、
软件，以及计算机中心运行的工作
任务。我为 IBM 的多个部门工作过。
尽管我从 IBM 获得的报酬一般不及总收入的一半，但这段雇佣关系给我的体验一直非常好。

中长久以来的一个问题，它可以将两台计算机的通道连接起来。在一个工程任务繁重的环境中，磁盘对磁盘式的物理转换会降低工作效率，非常不灵活，同时无法对客户变化的优先需求做出响应。我们意识到，如果我们运用刚刚出现的IBM 7040作为与外部（前端机器）交互的界面，并向其中植入一些控制代码，系统就能够按优先级顺序向后端（例如IBM 7094）返回工件，且不用人为干预（或仅需少量干预）就能自主处理许多干扰。简言之，通量越高，系统的优先级响应越好；且事实后来证明，硬件成本也会越低（因为IBM减价了）。做了一些展示报告，并获得许可之后，我们组建了一个团队。来自埃尔塞贡多(EI Segundo)和休斯敦(Houston)的客户团队开发了项目所需的软件。最后的成品可以实现和预期中一样的效果。我们同时改进了运营模式，降低了租赁成本。

美国航天航空公司的工程工作负荷并不是通用的，而我们的产品是一个全新的配置器，所以我们需要进行一些调制。要调制一个大系统，你需要对运行中的系统进行一些测量，然后观察主要的处理引擎是否会被什么事件耽误，并且找出背后的原因。如果延迟效应很严重，你就得重新进行配置。你可以将磁盘或驱动器移至其他通道以消除运行瓶颈；你还可以增加通道、额外购买内存，或者改变操作系统中的一些控制代码。最后一个、同时也是相当重要的一个办法，是重新搭建那些占

用了过多资源却没有充分将其利用的大型应用程序。

正当我在调试用Direct Couple 适配器连接起来的IBM 7040和7090时，IBM推出了一款小型电子提箱。这种提箱可以使用硬件收集那些我们正在使用软件包收集的数据。在空军部门时，我学会了一些为测试中的飞机安装部件方法。这款小型电子提箱的设计功能，正是为测试中的计算机做同样的事情。早年在IBM的卡片编程化计算器部门的一些经历使得我一贯不太信任测试用硬件，所以我自己为这个电子提箱设计了一些校准测试，由我们的一个团队将测试编写出来。出人意料的是，电子提箱的电路给出了和软件包不一样的结果。进一步研究之后我们发现，电子提箱中有一个连接错误的线路；事实证明我们是对的。

差不多同一时间，IBM的传奇人物吉恩·安达尔(Gene Amdahl)造访了航空航天公司，审查了Direct Couple的装配及工作负荷。两年以后我才得知，IBM的小型电子提箱是在为一个彼时尚未发布、后来名为 System / 360的计算机家族进行设计测试，通过输入/输出收集数据，并且计算其特性。几个月后，我惊喜地得知，由于安达尔的推荐，我得到了一份IBM在波基普西市(Poughkeepsie)为我提供的咨询合同。

在接下来的20年中，我从IBM接收到了涉及硬件、软件，以及计算机中心运行的工作任务，我还做了一些演讲。我为IBM的多个部门工作过，还曾出差去过欧洲。尽管我从IBM获得的报酬一般不及总收入的一半，但这段雇佣关系给我的体验一直是绝佳的。

寻求平衡

大约在我开始从事咨询工作的两年后，我发现自己已经在这个领域站稳了脚跟。我的客户分为四种：

- ④ 研究者客户。为他们工作使我能够不断学习。在工作类型的选择上，他们给予了我一定的

自由度。

- ◎ 军队客户。为他们工作使我能够紧跟行业的尖端进展，将现有的研究还原为实际应用。
- ◎ 商业客户/工程师（用户）客户。为他们工作使我能将现在以及未来的技术应用到一个成熟有序的环境中。
- ◎ 计算机制造商客户。为他们工作使我能将自己在真实应用场景中学到的经验应用到下一代产品的开发上。

同时为这四种类型的客户服务可以让我的生活保持一个稳定的状态。在意识到这一点后，我刻意地维持着这种平衡，避免发生任何的利益冲突。于是，当我完成一项工作后，就会开始寻找下一个同一类型的客户。这样，我一直拥有这四种类型的客户：研究者、军队、用户，以及制造商。

证测试委员会，为美国计算机协会（Association for Computing Machinery）前往全国各地进行专业讲演。我还协助撰写并编辑了美国情报处理学联合会（American Federation of Information Processing Societies）1974年出版的安全章程。我还为洛杉矶市交通委员会服务了五年，最后以委员会主席的身份退休。

除了上述志愿服务，我在本文中提到的所有工作本质上都是技术工作。然而万事开头难，这比实际的技术工作、和客户周旋、签下新的合约，以及紧跟技术前沿都要难。令所有人（包括我自己）惊讶的是，我在这行一干就是33年。

罗伯特·L·帕特里克
clpatrick@mac.com

回馈社会

20世纪60年代初期，我开始在《数据信息》（Datamation）杂志上发表文章。写一篇文章并不会使我得到多少收入，但对于想要获得专业曝光的人来说，这是她/他梦寐以求的机会。1962年，《数据信息》的时任编辑桑迪·兰扎罗塔（Sandy Lanzarotta）将行业内的人组织在一起，建立了一个小型的顾问委员会。我是他邀请的第一批成员之一。顾问委员向编辑提供报道线索，审阅他们存疑的文章，编辑同时会向顾问委员约稿。我为《数据信息》及其母公司汤森出版（Thompson Publication）工作了25年。在《数据信息》的编辑受伤时，我曾为三期杂志担任过代理编辑。

在往后的职业生涯中，我还从事了一些其他的公共服务工作。志愿服务虽然不计薪酬，但同样充满挑战。无论有无合约，你做出的承诺必须是一样的。你必须按时出现，除非遭遇紧急情况不能爽约。你必须展现自己的最佳状态，必须在一个你无法实际掌控的环境中尽自己的一份力量。我作为志愿者加入了数据处理管理协会（Data Processing Management Association）的认

征稿启示

让自己出现在人工智能领域的最前沿吧！

在IEEE Intelligent Systems上发表你的文章

IEEE Intelligent Systems 寻找所有人工智能领域的文章，它专注于将最新的研究付诸应用的开发工作。

投稿指南参见
www.computer.org/mc/intelligent/author.htm.

最好的人工智能杂志 **Intelligent Systems**
www.computer.org/intelligent

专访电子邮件先驱 雷蒙德·汤姆林森

撰文 Dag Spicer



雷神BBN科技
提供图片

“汤姆林森的电子邮件程序带来了翻天覆地的革命，从根本上改变了人们的交流方式。”

因其在互联网交流及网络化的贡献，汤姆林森获得了乔治R.施蒂比茨计算机先驱奖、国际数字艺术与科学学院颁发的威比奖、《探索》杂志创新奖、电器和电子工程师学会互联网奖。他还获得过阿斯图里亚斯王子奖（Prince of Asturias Award Laureate）中的科技研究奖，以及爱德华·雷恩·库特普瑞斯文华奖（Eduard Rhein Kulturpreis Cultural Award）。

下文根据2009年6月马克·韦伯和加德纳·亨德利对汤姆林森的采访整理而成，（全文手稿请见CHM数据库：<http://archive.computerhistory.org/resources/access/text/2016/03/102702118-05-01-acc.pdf>）汤姆林森已于2016年3月5日去世。

马克·韦伯：您于何处出生、长大，因为什么走上了计算机研究工作，可以讲一下吗？

加德纳·亨德利：我出生于纽约州北部的阿姆斯特丹市。出生后不久我们搬到了更北一点的韦尔磨坊（Vail Mills），这个小地方不属于正规的行政区划，

计算机工程师雷蒙德·汤姆林森有两件成名作：一是TENEX操作系统，二是1971年首次在Arpanet系统中运行电子邮件程序。互联网名人堂（the Internet Hall of Fame）的官方传记中称，

大概在萨拉托加温泉以西30英里。我的小学和高中都在那里上。那时我总喜欢把东西拆开，玩真空管之类的东西。当我要上大学的时候，我差不多已经决定要学电子工程方向了。

然后我就去了位于纽约州特洛伊的伦斯勒理工学院（RPI），并在那儿参与了与IBM的产业合作项目，为期四个学期。众所周知IBM是计算机领域的翘楚，但我对计算机并不感兴趣。那时我学电子工程专业，刚开始是做测试晶体管的设备，两三年后我终于穿过大厅见到了给工程师用的电脑，于是我决定给它编程序。那是台IBM 1620，它的字段长度可调，你可以算很大的数，大到计算机容纳位数的极限。我写了一个计算 π 的程序，那是我的第一个程序，我完全靠自己完成的。这跟我当时在做的东西无关，我单纯觉得这事挺有趣。

随后我去麻省理工读研，在语音部门研究语音合成，这对当时计算机来说计算压力太大了。有一天下午我穿过26号楼的大厅，走廊尽头有间昏暗的房间中传出大喊尖叫的声音，我进去一看，发现有十几个人围着一个阴极射线管显示器玩空间战游戏。我就想，“这玩意真妙，我得学学怎么把这种程序编出来。”

当要做硕士论文的时候，我想：“我要做语音合成，但我要通过计算机来做这件事。”我有了这么个想法：做个模拟电路，用数字计算机来控制。这是个混合体，模拟电路和数字电路各占一半。信号处理全由模拟电路来做，但计算机控制它如何设置放大器的增益、谐振电路的共振等等。用光笔给声音之类的画个基本轮廓从而调控音高升降，也可以用声道控制对声音进行调和等等。

这件有趣的事占据了我大部分时间。我不再去上课，把更多时间倾注在计算机上。我还是拿到了硕士学位，不过我做的程度太浅，要接着做博士课题几乎没可能。过了一段时间，我的导师建议我去BBN（博尔特·贝拉尼克-纽曼）公司工作，他每周在那里做一天的顾问。

BBN的语音项目做得还挺好的。1967年6月我开始在BBN工作，之后一直在那儿。

书们：您刚说您小时候就对电子器件感兴趣。您父母是做技术的吗？

汤姆林森：不是，我父亲是一名通用电气的机械师，他造些东西。我母亲那时不上班，虽然她婚前是一个百货公司的会计。后来他们做杂货店的买卖，我就整整货架。我不想再干这些事，不过这当时还是蛮有趣的。

所以说，他们不是做技术的，不过他们很支持我。我大概12岁的时候，他们给了我个可以用来制造电路的小工具箱。这是我第一次接触电子器件。在此之前我也拆过别的东西，不一定是电子器件，比如说钟表。我还挺擅长拆东西的。不幸的是我基本都没法把它们装回去。所以我意识到，我不仅得会拆东西，还得会造东西。

书们：您喜欢数学吗？

汤姆林森：我确实喜欢数学。我是个工程数学家。我的意思是，我学的数学用来做工程已经够了，但我觉得没有必要弄懂数学中很难的部分，像微分这种。只要时间够，我应该能解出数学家解的所有问题，至少回想起怎么解。不过我并不觉得用数学解决我的日常问题时有必要这样。

书们：您很早就对语音方面感兴趣。来到BBN以后您加入了语音项目组吗？

汤姆林森：没有。我去了计算机中心工作。其实当时语音项目组还没完全成型。公司里有些人对这方面感兴趣，这部分内容在我们当时的心理学项目组之外，不过作为声学的一部分，公司一直有语

音方面的想法。要做语音，当时的计算机还不够强力，所以当时语音都不能成为计算机科学的一个分支。当时的研究所还是有些为特定目的建造的机械、电子设备。我把我在计算机、混合运算方面的经验用到了其他项目上。

书们：比如说哪些项目？

汤姆林森：我们比较有兴趣的点包括人机交互、人们的行为、心理学实验、人们在不同环境下的表现之类的。我们并不造很复杂的设备，就是用计算机对实验对象施加刺激，然后考察他们的看法或反应，比如说在模拟驾驶的环境下扳一个操纵杆。这些研究确实跟我之前的研究有些相似之处，我用到了之前的经验，还蛮关键的。但我很快就不在这些领域了，转而研究基本的计算机问题。当网络出现时我已经全身心投入进去了。

书们：刚开始的时候，BBN内部做计算机的团队跟做系统的团队是不是有很多互动？

汤姆林森：有些互动，但也不多。这两块还是有些不同的。实际上我们在做联网的时候还有个硬件上的难题：早期接口的有效距离只有几英尺，而100码是个很大的障碍。为了克服这个障碍，我们得楼上楼下跑来跑去。为了让我们连上接口，他们不得不设计并建造了一个远程主机接口。后来我们终

**我大概 12 岁的时候，
他们给了我个可以用来制造电路
的小工具箱。在此之前我也拆过
别的东西，不一定是电子器件。
我还挺擅长拆东西的。不幸的是
我基本都没法把它们装回去。
所以我意识到，我不仅得会
拆东西，还得会造东西。**

于连上了，然后又得考虑用什么程序来使用这个接口。

韦伯：您对BBN的第一印象是什么？那里氛围如何？

汤姆林森：很放松的氛围，甚至有点对人满不在乎的感觉。雇我的时候也没有多少手续或者多严格的考核。我大概是跟人力的某个人聊了聊。我没有很深刻的印象了，所以那过程肯定是挺非正式的。当时我们签了个合同，对我的工作内容没什么限制。工作内容的限定很宽泛，就是我以后要做哪方面的研究。我有很大的创新空间，可以自己想一些点子。你可以说，“哪儿哪儿有机器，哪儿哪儿有个人，他们在交互”，这就是人机交互。那个氛围很宽松，我觉得在这种氛围下很容易出成果。

韦伯：有点像学术研究的氛围？

汤姆林森：比其他许多地方都有学术氛围。而且那儿的人都很友好。大家都对其他人做的事感兴趣，你总能找人问“这个想法怎么样？”有些人会说：“不怎么样。”好吧，好想法和不好的想法都是有的。

韦伯：您怎么又转到了BBN的计算组呢？

汤姆林森：我当时是计算机研究中心的成员，运行计算机的。那些研究者是使用计算机的，他们经常运行LISP语言，问题是大部分机器的内存都不够，所以有好多东西得放到虚拟内存上。他们当时用一种自制的虚拟内存系统，我们已经把当时的计算机用到了极致，再往后就须要新的计算机系统了。在许多备选方案中，我们选择了DEC PDP-10硬件。刚开始时我们用的是σ数据系统（SDS）计算机，这就是后来的Xerox。当时还有PDP-1，不过我没用。

韦伯：这是在计算机研究组的事对吧？

汤姆林森：当时我在的那栋楼有两台PDP-1，但只支持2~3名用户。这两台机器有分时系统，不过

并不是每个人都能用得上。分时系统大概是1963年开发的。我是没用过。我们当时在用SDS940。我们用的是伯克利的分时系统TSS-1.85，但也不够用。于是我们打算买个支持虚拟内存的新系统，好让我们用来做LISP应用。主要问题在于绝大多数用虚拟内存的系统页面尺寸都很大，而我们的LISP程序页面很小很小。所以我们决定用PDP-10电脑，但要在内存总线上加上我们自己的虚拟内存系统，为此我们涉及了BBN页面调度程序。大致的工作流程是这样的：数据地址从KA10处理器里面出来，程序对请求进行寻址、调整，再传送到实际内存。

加德纳·亨德利：那些页面有多大？

汤姆林森：有512个字。有个相联存储器来追踪正在使用的，再快速加载其他寄存器的内容，从而呈现出整个物理内存。

亨德利：硬件是您自己设计的吗？

汤姆林森：大部分硬件工作是杰瑞·布彻菲尔和我做的。这个活儿干起来还是挺快的。我们决定基于一个名叫TENEX的蓝本写一个操作系统，这个系统要有虚拟内存，所以我们从分区框开工。这个过程花了几个月的时间，不是很久。我们做好后给它写了个操作系统。然后我就放下了硬件工作，捡起软件工作，开始做监视器。

监视器的大部分工作是丹·墨菲和我一起完成的。大概有十几个人在做操作系统的其他部分，诸如执行控制系统、子系统等等。也有另外一些人做监视器，不过丹·墨菲做了调度器、存储管理、页面调度界面硬件、软件等等。我似乎基本没管这些东西。我做的是文件系统，其中JSYS是到系统指令的飞跃，我们进出操作系统都要用到；还有所有的高级别I/O功能、低级别驱动的界面。我们做得很慢。我记得12月页面调度系统就能用了，然后到次年三五月的时候又出了些问题，总是崩溃，不过后来我们又花了几个月时间让操作系统正常运行了。

有了TENEX之后，我们先运行了我们的LISP和其他研究人员要运行的东西。但我们还要等远程主机接口就绪，好接上网络。连上之后，我们得用网络控制协议(NCP)来与界面信息处理器(IMP)交互。我写了远程登录应用来使用远程登录协议。

这些都就绪之后，我觉得需要一种传送文件的途径。当时没有正式协议，文件传输协议还在制作中。我想，让我们把这些东西组合起来吧。

我试着从用户应用层面让文件系统界面接入网络。你可以用这种方式在打开文件的同时以相似的方式打开一个链接。我写了一个名叫CPYNET(Copy Net)的小程序，用来在一个地方打开文件，在另一个地方打开链接，并把文件传来传去。挺简单的。首先，你输出一个字符串——就是你想命的文件名。另一端接收这一字符串，在那边打开文件，文件数据就在这两边流动了。到这儿我们就很接近电子邮件了。

书们：我们先回来一点。您当时在做NCP，但是为美国国防部高级研究计划局(ARPA)做的。IMP小组是个很独立的小组，您当时跟他们是什么关系？

汤姆林森：主要来说，他们是我们遇到网络问题要找的人。他们就在那儿，我们就这样儿。也许我们会在午餐室见到，但我们做的东西还是挺有区别的。

书们：ARPA是跟IMP小组签的协议，对吧？

汤姆林森：是的，建立阿帕网的协议是跟IMP小组签的——戴维·瓦尔登、弗兰克·哈特那些人。

亨德利：TENEX项目是谁在支持呢？

汤姆林森：计算机中心做的研究也是ARPA支持的，但主要是应用端。就是写个LISP应用，做些AI相关的东西或者探索人机交互的方式，诸如早起的自然语言识别：输入一个短句、分析、理解其含义，然后基于它背后的数据返回给你合适的答案。刚

我试着从用户应用层面
让文件系统界面接入网络。
我写了一个名叫
CPYNET 的小程序，
用来在一个地方打开文件，
在另一个地方打开链接，
并把文件传来传去。

开始我们运行计算机是为了让他们做这些研究，但随着框架发展成熟，就不用我们做太多事了。包括我在内的这群人就逐渐开始做这个领域由ARPA支持的项目。

亨德利：所以这是个附加任务。TENEX是那种周末做的副业吗？

汤姆林森：是的，是附加任务。

书们：Jerry Burchfiel当时在做什么？

汤姆林森：我不知道。(笑)

书们：可你们俩关系很密切？

为了继续做我们之前的项目，我们需要类似于SNDMSG的东西，所以我用机器语言为PDP-10重写了一份。

汤姆林森：我们俩的办公室紧挨着。他好像是在计算机中心，或是丹尼·博布罗或别的什么人的团队工作。至少一定是ARPA赞助的项目。计算机中心的人都知道我们这儿有操作员。每次除了问题，就得有人去给软件纠错或者加些新的软件特征，但大部分软件开发出来就是为了与我们做的SNDMSG(发消息)等研究对接。那是个用伯克利分时系统搞出来的程序。

我们换上PDP-10电脑的时候，问题也随之而来。为了继续做我们之前的项目，我们需要类似

在我看来，@最有意义，
因为at是个介词，
是键盘上惟一的介词。
其他的都是“与”之类的，
没有其他@了，
除非你觉得下划线也算@。
很多人说起早期的事都会犯这么
个错：去数字键2哪里找@，
但在Model 33电传打字机上，
@跟2并不在一起。

于SNDMSG的东西，所以我用机器语言为PDP-10重写了一份。我写了这两份软件，四月份斯坦福研究院(SRI)的迪克·华生写了RFC(请求注解)来探寻用网络发送信息并在行式打印机上打印的方法。大多数高速打印机都是行式打印机或链式打印机，一次打一行的那种。这意味着一张张纸打完之后会被放进一个盒子，直到有人来拿走。

当时他提出的许多协议都是在打印机上排版，如何换页、缩进、制表之类的。我看到这个问题的第一反应是，“他试药让你把信息发到其他地方。为什么不直接发给其他人呢？这样就不需要邮箱了。”每个人都有自己电脑的登录帐号，如果知道一个人的帐号，你就能把信息发给这个人。至少在我们的系统上，就算实际程序不像SNDMSG，大多数机器与其也有相似之处。

那种分时系统还有很多衍生产品。我的想法是干脆把两者结合起来，用CPYNET来传送文件，因为邮箱就是个文件；用SNDMSG来写信息内容、寻址，并发送。如果想要纸质版，可以把信息发到打印机。如果想慢点接收信息，可以用电传打印机。当时阴极射线管显示器正变得越来越流行，不过在很多地方还没有那么普及。

书们：有了SNDMSG，只要人们在同时登录同一个分时系统，就能收发信息，对吗？

汤姆林森：有了SNDMSG，人们不必在同一时间在线。所有必要的事情其实已经都有了。人们可以用SNDMSG写信息、寻址，可以直接在他们的终端打出来，不需要什么特殊的阅读软件。

CPYNET是文件传输的测试版，仅仅是一个开始：你在这里写文件，放进这个程序，文件就在那边写出来，这里和那边之间就是@符号连接的。@是个介词，连接了你和其他用户，使你得以决定将信息发往何处。

第一版RFC诞生于1971年4月。几个月后又进行了修正。那年夏天我知道了RFC的存在。RFC刚出来时我没见过。我做了几周这方面的工作，不过我当时的重心并不在此。制作整个软件耗时并不久，有几周每桌做三四天。

书们：您记得那些想法是何时成型的吗？

汤姆林森：成型挺快的。有些很具体的问题要解决：用户名和主机之间应该用键盘上的哪个键分隔？剩下的东西基本已经就绪了，我不用去想我如何打出一条信息，在用户的名下就可以。TENEX不用数字用名字，其他登录系统用数字或者其他途径指定用户。

书们：在TENEX里面信息是什么样的？

汤姆林森：就是给出用户名、主题、信息正文、日期。日期好像是自动填上去的。

亨德利：所以说，代码也并不复杂？

汤姆林森：在已有内容的基础上，用汇编语言写也就400行的代码吧。语言必须简洁，因为那时候的计算机内存不大。

1971年的某天，第一条信息成功从一台计算机发到了另一台。我们把两台计算机并排放着，硬件基本都相同，其中一台内存大一些，用来开发操作系统、测试新版本等等，以免它干扰其他机器。因为它们就是并排放着，你可以在放了两台计算机

的桌子旁坐着，这边打点字，那边看看，然后来回忆这样。不用两头跑。

刚开始的时候出了很多错。对大部分程序来说，你写点代码、测试下，一般都出错。哪次不出错了得回家庆祝下，太不寻常了。别人问我第一封电子邮件说了点啥，我只能告诉他们没人知道，因为我是想到什么输入什么，手指滚键盘。我要是觉得特别无聊，可能会输：“葛底斯堡演说我还记得多少？”然后接下来输：“八十多……我记不清了。”（译注：林肯葛底斯堡演说第一句是“八十七年前，我们的先辈们在这个大陆上……”）什么时候完成了最后一步？我记不清了。很多很多小细节造就了最终的程序，从首行开始到每一个回车，都是一点进步。

某种程度上讲，接下来就是TENEX的下一次发布。当时，我想阿帕网上大概有六个在运行TENEX的站。在我看来，@符号最有意义，它是键盘上唯一的介词。

书们：我知道你肯定被问过十几次了，不过选@符号……

汤姆林森：是个没过脑子的决定。回想起来，我应该对用Multics系统的人友好一点的。他们把@用作删除行。所以他们一输@符号，键盘就锁定了。他们用IBM 2741终端，这种键盘会锁。如果他们真的想输@符号，得再输一次。这种锁定机制是终端的硬件决定的。他们给终端的控制器编程说：“我们真的是想输入@。”然后终端就会发回信号解锁键盘，你才可以再输入下一个字符。如果你想输入@得输两遍，大概就这样。对他们来说用@符号实在是尴尬，所以他们找了别的办法避免输入@。

在我看来，@最有意义，因为at是个介词，是键盘上惟一的介词。其他的都是“与”之类的，没有其他@了，除非你觉得下划线也算@。无论如何，这是有意义的。这个人就在(at)他的主机这里，@键就在那儿，都不在数字键那一排。很多人说起早期的事都会犯这么个错：去数字键2哪里找@，但在Model 33电传打字机上，@跟2并不在一

起。

书们：那@键在哪儿呢？

汤姆林森：就在字母行，P旁边。因为如果看看所有终端都在用的ASCII码你会发现，数字都在一起，跟键盘上的物理按键以及按键产生的码一一对应，有一个机械编码器，某种意义上讲，你按键的时候，它就发送这三个比特和所在列，然后发送字符串编码的另外四个比特。把@放在数字那行会把布局打乱，因为在ASCII码中它就在A旁边。

电子邮件是随TENEX一起传播出去的，时间大概是1972年1月。

亨德利：TENEX是怎么传播到其他站点的？

汤姆林森：BBN的研究人员跟其他机构的研究者，尤其是大学的研究者经常来往，加州大学在阿帕网有三四个站点，犹他大学和麻省理工各有一个。我们的研究人员跟许多人合作，在这时他们可以在阿帕网更高效地合作了。这是整个阿帕网计划的基础：达成这种合作。因为当时DARPA（译注：即ARPA）并不想给他们的每个研究者买大型计算机，所以他们在摸索共享资源的方式。鉴于DARPA支持TENEX的发展，我相信他们对TENEX也是感兴趣的，它是能让其他研究人员共

他们给终端的控制器编程说：

“我们真的是想输入 @。”

然后终端就会发回信号解锁键盘，

你才可以再输入下一个字符。

如果你想输入 @ 得输两遍，

大概就这样。对他们来说用 @ 符号实在是尴尬，所以他们找了别

的办法避免输入 @。

电子邮件的诞生是量变而非质变。 我能跟隔壁楼的同事发信， 为什么不能跟镇里或整个国家的其他人发信呢？

用的操作系统。

亨德利：所以DARPA很可能是支持TENEX传播的。

汤姆林森：是的。

亨德利：像伯克利软件套件那样(BSD)？

汤姆林森：是的。我们组了一个用户小组，有意向的人都聚在一起，购买BBN的调度器，拿到自己的KA10。我们会帮他们装机、提供软件。到那时，所有活动都是DARPA赞助的了。

韦伯：在电子邮件之后，下一步是什么？

汤姆林森：电子邮件的故事还没完。1972年4月，FTP工作组开了个会，会上我们开始为电子邮件设计文件传输协议。这个协议我们用了很久，直到简单邮件传输协议(SMTP)诞生，戴维·克罗克做出了做出了下一代支持邮件传输的协议。我们继续在BBN做了一段时间的TENEX。1974年，我们在夏威夷的一个操作系统研讨会上展示了TENEX和我们的工作。

在那次大会上，文特·瑟夫和鲍勃·卡恩展示了他们的TCP/IP论文，我被这个作品迷住了。我们参与开发了TENEX系统中的TCP协议。我也写了一篇讲选择序列号的小论文（应该是在1975年），文中指出了这项协议的一些不足。序列号的容量只有32位，如果你长时间运行连接对话，序列号就会自动换行、再循环，我的论文意于指出这个问题。

网络自身会延迟、保存，制造数据包的副本，所以一个数据包可能被发送很多次。旧的数据包跟新的数据包一起过来，就会令人不解，也会出错。

这就是那篇论文的论点。不过后来人们发现TCP还会产生“三次握手”，当时没人想到。序列号问题用另一种方式解决了一一这样也好。我做的东西不完美，需要改进，所以有人去进行改进。我对三次握手并不负全责，不过……

电子邮件的诞生是量变而非质变。我能跟隔壁楼的同事发信，为什么不能跟镇里或整个国家的其他人发信呢？

韦伯：我们回来讨论电子邮件的问题，在您看来这个圈子何时接受了电子邮件呢？

汤姆林森：接受得相当快，因为电子邮件满足了一项很重要的功能需求。那时已经有录音电话了，但还不普遍。你想联系的人不在时，还要他的秘书带话或者用录音电话留言，让他给你打回来——这是比较走运的情况。对方不可能随时随地应答你，电子邮件克服了这一沟通障碍。打电话、写书面信都比不上电子邮件。

人们可能与合作对象处在不同的时区，或单纯有不同的时间安排，电子邮件一出现就为这个受众群体所欢迎。当然，问题在于能接触到计算机和阿帕网的人很有限。当时（1972年早期），我想一共只有28个节点。每个站点最多五十或一百人，一般有系统权限的人没这么多，所以你也就能跟一两千人以这种方式交流。虽然当时电子邮件在这个小圈子里很受欢迎，但一开始还没什么全球性的影响。

韦伯：您在BBN内部会用吗？

汤姆林森：电子邮件在BBN很快就流行起来了，不过并不是每个人都用。在团队中总有些勒德分子觉得要尽可能多面对面交流。两种交流方式都试过的人知道，面对面交流是一种有效的交流方式，你的“带宽”更大，不必浪费时间去纠错、调整语言。但俩人时间表不同的时候，这些好处就毫无意义。

我想，当时应该有早期分析称电子邮件很快会大火，但他们选择晚点接受。电子邮件这个术语应该是在三四年后才流行起来。

书们：当时您管它叫什么？

汤姆林森：发信，信息，邮件；不是电子邮件。但大多数情况下你就是给某人发一条信息。分时系统已经内置了这个功能，所以电子邮件也很类似。不过是使用群体比以前大了些，是量变而非质变。我能跟隔壁楼的同事发信，为什么不能跟镇里或整个国家的其他人发信呢？

书们：早期邮件的内容和现在相仿吗？

汤姆林森：我想是很相仿的。不久之后人们就开始用邮件安排会议、午饭聚餐、发布会议和研讨会通知、告诉大家某人会在午餐时跟大家聊聊，诸如此类的事情。

书们：刚开始时，要抄送一群人容易操作吗？

汤姆林森：容易，最早程序版本就有“抄送”列表。随着时间的推移更高效的群发功能也诞生了。在那时你真的得给每个收件人一封一封发。有群组之后我们继续给每个人发了很久。用邮件列表服务器群发可能容易些，但大家还是倾向于给每个人单发。这主要是带宽要求——至少文本邮件是这样——很少被网络视为一个整体。

就算在当时，电子邮件也不是个很占资源的应用，这点是推动其发展的另一因素。没人会装电子邮件程序之前还再三考虑。跟他们使用的其他程序所占资源相比，电子邮件占的比例实在不算大。

书们：后来电子邮件要跨过多个网络送达还挺费时间的。但刚开始时电子邮件都在阿帕网内部互发，是吧。

汤姆林森：是的，延迟主要跟你所做查询等行为的限制有关。

书们：当时在阿帕网的圈子里，您是否因为发明电

子邮件受到了广泛的认同呢？

汤姆林森：没有吧。这是个很有趣的现象，所有人觉得自己都能写个邮件阅读器，但他们只是觉得有趣在旁观而已。写电子邮件的程序确实不费劲，不过发送器的部分确实琐碎了点。虽然当时大家一共写了三四个版本的程序，所有人还是都认为他们有更好的用户阅读界面方案。你想想，确实如此。写这个程序就跟其他文字处理应用差不多，还算比较简单的。但在阅读端，一旦你收到了大量邮件，组织这些邮件就比每收一封都在你终端打印出来还要麻烦。如果你每天收一封，问题不大；你打印出来就完事了。如果你每天收200封，就是个问题了。所以所有人都觉得他们可以写个更好的阅读器，他们的方案有些也很新颖有趣，但这些方案很可能都没有存活下来。

书们：在20世纪70年代，SRI的道格·恩格尔巴特不是也做了一些阅读器吗？当时也有些协议在探讨中，是吧？

汤姆林森：当时有些对标题字段标准化的探讨，很早就有，很快就决定了。大概有六七个像这样很常用、有待标准化的东西，其他的就是为了特殊需求而存在的了。处于测试中的都标记上了X。有一点要注意，大多数情况下你并不需要理解标题，只需要知道那是一个标题，你可以展示出来它。用不着找二十种不同的办法来做同一件事。

书们：您参与得多吗？

汤姆林森：不多。有个功能我叫做“Re:”（回复），取自一个便笺格式，源于拉丁文，意为“考虑到”、“关于”。还有，回执、发件人、抄送给其他人的副本也是这个整体中很关键的部分；日期也是必需的，因为你要知道邮件是什么时候发出的，就这样。后续的邮件就追寻第一封邮件这条线，“回复：”就成了邮件主题——没什么大不了。邮件主题里面有“回复”是件奇怪的事，让我疑惑了很久。有人说，“这就是个主题，但我们要把‘回复’这个词放进去。”我一直不是很懂。

书们: 什么时候有了邮件附件这个点子？

汤姆林森: MIME编码这种内容方式确定后才有的。这本质上是种把一封邮件分成几个部分，并告诉你各部分是什么。确定机制后，添加附件就是小事了。单纯的文件插入出现得很早，但你得手动把说明这份文件的东西跟文件分开。这大概是20世纪70年代中后期的事情，人们尝试运用这种机制并没有花很久的时间

书们: 您自己经常用电子邮件吗？

汤姆林森: 是啊。我想我一直是一个很朴实的电子邮件用户。有些人把电子邮件用出了许多有创造性的花样。我自己喜欢直接打字，不喜欢HTTP编码或HTML编码邮件。我一直想保持邮件简单。

书们: 你创造电子邮件的时候预期到它会产生多大的影响么？

汤姆林森: 没有，并没有真正预期到。我觉得每个能接触到网络和计算机的人都没有理由不使用电子邮件，因为它显然非常有用。但在当时能接触到两者的人只有一个小群体，大概1000人。

电子邮件的广泛使用真正需要的是网络和计算机的普及。在某种意义上，两者互相驱使。有些人出于基本业务或生活需求，想要使用像电子邮件这样的应用程序，他们会买一台计算机、通过调制解调器连接到网络上，这当然会导致人们对低价计算机和高承载力网络的需求增高。而这些人接入网络就意味着你可以沟通的群体范围也大为扩大，从1000人扩大到1万人、10万人或是100万人。随着数字的增长便会产生再生效应，需求会驱使计算机和网络的价格下降，因为它们只要一经生产，便可以再次用大规模生产技术批量生产，像用模具做饼干一样。这样就发展成一个不断扩大的螺旋，把越来越多有能力使用电子邮件的人覆盖进来。

书们: 在电子邮件之后发生了什么呢？

汤姆林森: 电子邮件之后？那我们可以聊上一整天。之后我们有一个有趣的项目，无线电分组交

换网络 (packet radio network)，邮局把它称作Intelpost，它与发送信息相关，但不是电子邮件。这个想法是在输入端使用传真机发送文档，将文档通过TCP消息一起发送到其他邮局，在那里另一台传真机将文档转回到纸上，然后经由邮递员送达给接收者。这是一种发送大型文档的方式，据测比其他方式更便宜，也肯定比其他方式更为快捷。

如果你想给多个收件人发送10份相同的文件（比如一份10页、12页或50页的合同），邮局可以把文件发送给5个或10个不同的地方，每个地方发送两份文件，然后作为普通邮件在本地投递。整个想法的初衷在于，传真机在当时十分昂贵，而通过结合网络功能便可以获得一些协同增效作用，削减成本。等到传真机变得越来越便宜之后，人人都可以自己这样做了。

我们一度还写出了一个可以将TCP融入TENEX管程的程序，它的作用与NCP（网络控制协议，Network Control Protocol）相同，你打开连接并进行数据传输。比尔·普拉默 (Bill Plummer) 和我一起从事这项研究，我认为他做了大部分的工作。我们用它来连接行式打印机，其中一台机器上的行式打印机实际上是一台测试机，在这台打印机上打印东西成为测试TCP的一种方法，看看实际使用中时候它会如何中断。另外我们做了一些工作，说服国防部IP才是他们在互联网中应该采取的协议模式。当时是20世纪70年代末。

书们: 接下来是三网实验吗？

汤姆林森: 是的，我们建了所谓的工作站，它与每个无线电分组交换网络中的线路布设有很大关系。我们还从事了一些安全和保密通信工作。我记得罗克韦尔·柯林斯 (Rockwell Collins) 也参与了，所以我怀疑这些工作一定与无线电分组交换网络相关。

书们: BBN公司在其中发挥了什么作用呢？

汤姆林森：我们在无线电分组交换网络项目中的作用，是为网络提供一些更高级别的控制功能，也就是工作站。无线电分组交换网络本身是相对自主的，它们也需要自主，因为它们并没有实际连接到其它物体上。然而，它们必须通过一个自举过程（bootstrapping process）才可以运行，接收初始信息并组织其他入网信息。当时，因为中央控制是这种系统中潜在的致命弱点，完全自主的方式似乎不是正确的选择。我们知道如何来处理中央控制，无线电分组网络需要一定的自主性来运行，但要良好运行则需要工作站的引导。事实上，因为这不是一个可以单独研究的课题，我们还参与了无线电分组网络开发其他方面的课题。但这是我们名义上的研究课题是工作站。

们：你们参与了之后的军事实验吗？

汤姆林森：没有，我们后来退出了。我没有参与另一个阶段的无线电分组网络开发。项目终止的时候，我们有了无线电装置，可以架设网络。我们能接入互联网，连进圣弗朗西斯科或者西海岸上其他进行实验的地方，网络基本上一直有效。因此能推测出来军方已经部署了网络，或者至少到了即将完成部署的地步。

们：你在双网和三网实验中起作用了吗？

汤姆林森：我的参与程度在于写了一个网关或路由器程序。这应该在无线电分组网络项目的早期，因为我们使用的是DEC LSI-11小型计算机，我花了一些时间来写出高效的路由程序。程序运行得很好，但有人说：“这个程序还得有其他所有功能。”然后他们接手过去，又把它弄得缓慢不堪。我个人认为如果你做最低限度的事情，它会在效率方面表现很好。它只是不支持你需要的行政目的、访问控制、隐私或任何实际上使现代生活的许多方面复杂化的事情。

这一次，施乐（Xerox）公司的帕洛阿尔托研究中心（Palo Alto Research Center, 缩写PARC）——他们挖我们公司的员工有一段时间

**我们在无线电分组交
换网络项目中的作用，
是为网络提供一些
更高级别的控制功能，
也就是工作站。**

——在开发业务和研究方法方面做得非常好。他们的活动产生的结果之一是Alto工作站，一个具有位图图形和出色计算引擎的桌面单元。我们很羡慕这一点，既然情况不允许研究中心销售这个工作站，也还没有别人进行相关营销，我们决定继续做我们自己的工作站。

于是我们设计了Jericho工作站。它是一个具有内置磁盘和位图图形的位片结构，成品比我们原本预期的要大一些——是一个像文件柜一样大的箱子。我工作了一段时间，完成硬件工作。然后我为它写了一些Pascal语言微码，并开始做一些多媒体开发。我们开发了JADE（发展和执行联合助理，Joint Assistant for Development and Execution），一个军用规划工具，整个用Pascal语言编写。这让BBN公司开始开展多媒体计算业务，这个项目多年来以不同的名字一直在继续。

们：你为Jericho工作站开发了自己的操作系统吗？

汤姆林森：是的，它直接转译Pascal语言或LISP语言，有两个版本的微代码，一个用于Pascal语言，一个用于LISP语言。所以你像是在用子程序运行计算机，而不是用保护模式下的内核来实现。

Pascal语言之后被编译成p-code，我们做了一点点修改，但它只是一种以字节表示Pascal语言代码的方式——这个用一个字节来表示，那个地址或其他什么东西用五个字节来表示。微代码用来转译代码和运行机器。有一些指令相当复杂。

我们开始设计一台拥有
65,000个处理器的
超级计算机。
这将是一系列串联
在一起的计算机。

加德纳·亨德里：谁做了硬件设备？

汤姆林森：吉姆·卡尔文 (Jim Calvin)、吉姆·米勒 (Jim Miller)、鲍勃·克莱门茨 (Bob Clements) 和我做了主要工作。艾伦·贝尔 (Alan Bell) 是实际发起者，我记得他后来去了帕洛阿尔托研究中心。其他人也时不时有所贡献，我可能记漏了谁。

当时是1980年左右，早于Sun工作站问世之前。

加德纳·亨德里：Apollo当时做出自己的工作站了吗？

汤姆林森：我记得他们有了自己的工作站，但是我不确定，应该是在同一时期。Fibernet那时有了我们在6号楼使用的光纤网络。

加德纳·亨德里：你建了很多工作站吗？

汤姆林森：大概有25到30个之间。我记得其中运行LISP语言的机器比运行Pascal语言的机器要稍多一点。他们做了一个非常漂亮的LISP卡片组，用微码编写。CONS、CDR以及所有主要操作都在微码中实现，所以你不会用完整的机器指令来建工作站。而它的构建方式和它的工作方式一样。工作站中有一个很重的磁盘，降低了机器的整体水平。

韦伯：Jericho工作站处理器是什么样的？

汤姆林森：我记得是AMD 2910。

视频硬件配得相当好，它可以运行普通的 640×480 彩色显示器，也能在黑白显示器上运行更大像素数，而且你可以编辑刷新率和扫描率。它

大体上可以适应几乎所有显示器，我们其实用它试过大约三种不同类型的显示器。

韦伯：处理器有鼠标吗？

汤姆林森：有鼠标，也有键盘、存储量可观的磁盘和Fibernet界面。键盘用的是十分常见的RS-232界面，你可以在挂接终端、运行视频之前调适排错。它用的是测试微码，你需要弹出PROM (可编程序的只读存储器，Programmable Read Only Memory) 来更改微码。它不能直接编码到位，所以你只能离线烧录PROM，然后再插入，估计它们大多数后来都是因为这个才坏的。

这些系统我们用了大概两年，然后Sun推出了100和150型号的工作站，之后出现的工作站越来越多。我觉得用LISP语言机型的人比用Pascal语言机型的人使用Jericho系统的时间更长，但是我们之后接手的工作都是用Sun工作站来编码。

接下来我们做了一些从未公之于众但很有意思的事情。我们想到可以设计自己的超大规模集成 (VLSI, Very Large Scale Integration) 电路。当时DARPA (美国国防部高级研究计划局) 在运营一个代工厂，你可以做出设计、寄到铸造厂，然后就可以拿回来你设计的芯片。我们决定将这个优势发挥到淋漓尽致，开始设计一台拥有65,000个处理器的超级计算机。这将是一系列串联在一起的计算机而不是一排互不相干的计算机，所以所有的存储器和处理器之间都会有内部通信。我们认为这是一个好想法，因为在使用大批计算机时的问题之一是如何保持它们的时钟同步。你可以让它们内部保持异步，等到它们需要彼此交换信息时不停地再同步。但是我们认为，如果每个CPU上只需要调整一根管线的话，便可以做出能使所有时钟同步的设计。为此我们研究设计了对应的大规模集成电路。我们做出了这样一个电路，但不知道是否获得了专利。我们的想法是，你可以用电子方式调整延迟，这样它就可以感知到延迟，原理类似于锁相回路。

唯一的问题是它的实体出奇地大，要占掉一个

40英尺见方的房间，周围要有很多冷却机。此外，因为电路之间的相互连接，房间内不能有任何柱子，用掉一卷厚达8英尺的电线，还消耗两万瓦特的电量。显然，这样的体积导致它无法推行。可是这件事还是很有趣，我们在这个过程中逐步研究出来一些很酷的并行处理算法和技术，因为如果你要做这个并行处理的玩意，显然你最好能拿出实际利用到这65,000个处理器的算法。

当时有一个标准检查程序来检测24种问题，由威尔·克劳瑟(Will Crowther)负责。他想出了处理每个问题的方法，问题解决之后，你会发现处理器数量越多，整体线性加速越大。而且你从来不会犹疑要怎么应对这些处理器，非常酷。

当时是80年代中期，可能是1986年或1987年。

书们：在并行处理器工作之后你做了什么？

汤姆林森：我回到了在超大规模集成电路之前一直做的多媒体工作上。对这段时期我没有什么突出的印象。我们做了一些视频会议、多媒体会议。这些是事件驱动的会议技术，想法是把同一个应用程序放在多个用户面前，然后通过网络分配，实现的方法有很多。现在通常是在一个地方操作，然后把屏幕图像发送到各处。这个做法对当时的网络来说是不可行的，所以我们研究了一个方案，基本上是取标准事件流，然后在各地复制相同的计算，并确保每个人都播放事件流。只要一切如常这个方法就奏效，但一旦什么地方出了状况，你就无法恢复同步。该方法有可取之处，但也有局限性。

还有一些与视频处理相关的其他事情。我们有一个早期版本的网络摄像头，有一段时间，在6号楼的四楼窗户上有一台摄像头把录像播到网上，当时大概是1990年或1991年。

书们：你用什么工具来连接摄像头呢？

汤姆林森：用的是叫做PicWin的工具，名字取自“图片(picture)”和“窗户(window)”。有小小的商业化宣传，但并没有获得多少市场。我自己并没有参与其中，但我目睹了整件事。摄像头透过窗

户拍摄十字路口的交通灯，每隔一段时间就会拍到一起事故。要是你真的很无所事事，可以看看是否有人在交通灯那险些发生事故。

书们：我之前看到有人提过接入互联网的电话会议系统。

汤姆林森：Slate，问世时间在1991年或1992年。它不只是视频，而是群件(Groupware)。它有电子数据表和文字处理功能，很难说它后来怎么样了，只是渐渐地就消失了。BBN公司历来在营销产品方面不是很成功。现在我们效益很好，但那时的心态不对。产生一些新潮想法，并把这个想法变成实际可销售、可支持、有利润的产品，需要一群完全不同的人。你需要两个不同的公司，或者至少同一个公司的两个群体来做这件事，很少找到一个同时擅长两者的人。

大约在同一时期，沙漠风暴行动也在进行中，BBN公司的某个人试图向DARPA销售软件，告诉他们我们有某种装备，一种测试平台。我们认为我们可以建立这个测试平台，然后各个大学等可以做出各自的贡献。大概就在此时，伊拉克向科威特或其周边派遣了部队，造成了一些危机，DARPA与我们沟通的人于是说：“我们会需要这样的东西。”

所需软件必须能够快速做出物流决策和后方运输计划。我们快马加鞭，努力实现这个非常具有实验性的演示软件，当时仍有很多漏洞。我们很快把这个工具建成，并把它带到德国，国防部在那里为沙漠风暴行动做了很多后勤准备。

这个实验性软件的唯一问题是它需要手动操作，基本上要有人每天24小时看着，它经常崩溃，需要有人重启。我们通常把它和国防部当时在用的标准物流工具同时运行，当做某种备份。我们可以做一些“假设”实验——假如你有这个计划，我们这样做结果会如何？我们的软件可以相对快速地做到这一点，国防部的标准软件无法做到这么快。两个软件同时运行也可以验证他们的计划是正确的。我曾在那里看守过一阵，这样我了解到了

从德国回来后，
我继续从事了一些多媒体方面的工作，然后接下来就是可穿戴个人支持系统 Pathfinder。你可以用它来沟通和监测身体状况，至少在原则上是可行的。

数据库。

书们：这个军用物流项目叫什么名字？

汤姆林森：我们把它叫做DART（动态分析和再计划工具，Dynamic Analysis and Replanning Tool）。

书们：在沙漠风暴之后呢？

汤姆林森：从德国回来后，我继续从事了一些多媒体方面的工作，然后接下来就是可穿戴个人支持系统Pathfinder。你可以用它来沟通和监测身体状况，至少在原则上是可行的。它带有GPS，所以你可以知道自己在哪里；它带有无线电广播，所以你可以与其他人沟通，告诉他们你在哪里；它带有地图库，以便让你看到自己在地图上哪个地方。虽然我们从来没有实际开发它，你可以把它连接到任何你会穿戴的传感器上。所以理论上，如果你受伤了，你可以让它通知某人。小的排级部队可以把它用于军事行动中。同时我们在制作某种早期的平板计算机。我们研究了一些潜在的应用程序，并使用英特尔486处理器进行原型开发。

书们：那之后呢？

汤姆林森：我们在物流汇总桌面上工作，这是一种获取并展现各种物流问题的位置感知系统。我们主要从数据库得到数据源。有一种或多或少的标准规划格式，叫做TPFDD（时间分阶段部署数据，

Time Phased Force Deployment Data）。它基本上是一行一行的列表，把所有需要物流运输的事物的不同细节记录下来，从人员运输到大宗物品，如石油、水、坦克、履带、修理零件和所有九类供应物品。这是一种将所有信息集中在一个显示屏上的方法，因此你可以了解物流情况、手头有多少东西等等。

这个桌面后来演变成了Cougar项目，但Cougar项目的中间名是ALP。这一系列规划工具基于代理，用来协助物流问题。使用基于移动代理的技术的想法是将计算分布到相对接近的地方，例如补给站或战场中的作战单位，这样你就可以跟踪了解当地情况，让那里的计算机知道你目前的情况，然后其他地方的代理交流并安排运输。

举个例子，你在此处有一个补给站，你要把供给运输到彼处，因此你需要安排运输、制定行程、想出计划，然后在计划执行时开始观察。重新计划是一个整体概念，因为每个计划都会有变化，一旦计划开始执行便不会再准确无误。由于这个原因，我们在基础设施中，也就是它们运行的软件框架中处理移动代理。多个供应商协同合作，我们主要集中在基础设施上以支持各种计划，其他公司则提出解决其他事情的算法。当时有两个程序，一个是ALP，一个是UltraLog。我们的代理系统后来被称为Cougar，它现在仍在服役，并持续多年处于开源状态。

书们：60年代时，军事研究、反战运动和许多人的生活方式之间关系有些紧张，BBN公司中是怎么体现的呢？

汤姆林森：BBN公司里总会有人选择不参与国防部资助的项目。我们所做的大部分工作都与国防部没有直接联系。DARPA本身经历了从DARPA到ARPA，再从ARPA到ARPA反复几次名称改变，这取决于他们想要强调所做的事情是先进性还是防御性。但研究的大部分具有普遍目的，可以用于解决许多可能会出现的问题。它恰好是由国防部资助的，它当然有国防应用程序，不过有时你看到

某些项目会心生怀疑。

大多数人可以在那个多元化氛围中找到适合自己的地方，而那些没有找到自己位置的人则会另投别家。我们很长一段时间以来一直有教育小组，虽然现在的功能比以前削弱很多，这种现象在教育小组中也有体现。很多时候，有些人发现一些商业项目，而有些人去了别的地方。当然，在60年代和70年代的越南战争时期，国内紧张情绪不严重，比起内部人员，外人把我们视为在国防工业中实力更强、地位更加重要的角色。刚才我说过，我们的大部分工作用于普遍目的，不过都偶然具有军事意义。但我们是由国防部资助的，这往往会引起人们感兴趣。某些情况下，我们确实受到了一些从未真的威胁。

Q: 当时BBN公司很多员工看起来像学生或者嬉皮士，是吗？

汤姆林森：这个嘛，这里的人们总是很多元化——当时如此，现在也是如此。有些人光着脚进来，看上去像是刚刚起床，没经任何梳洗就来了，但也有一些总是西装革履的人。每人根据他们的工作性质穿着，统一着装并不是很重要。很明显，在某些情况下，例如与客户进行互动时，客户对于什么样的人来接待他们有自己的期望，他们可能会因为这样一个不修边幅的人来接待他们感到不满。但在大多数情况下，这些问题根本不存在。我不得不承认，我可能相对不受这类事情的影响。我不太会注意到这些事情，所以我可能不是这方面话题的最好发言人。

Q: 对于有兴趣从事计算机工作的年轻人，你有什么建议吗？

汤姆林森：我认为如果你不感兴趣，想用计算机做什么都很难。你需要记住，你没有必要去寻找问题，很多问题就在那里，它们会找到你。你需要知道这些问题中的哪些方面是你能够解决的。想知道这些问题的哪些部分已经被其他人解决了，你需要做研究。你需要知道还有什么其他事情已经发



投稿 Articles

《IEEE软件杂志》希望刊登实用且值得一读的文章，可以吸引专业和非专业人士的兴趣。这本杂志的目标是把可靠、有用和前沿的信息传递给软件开法师、工程师和管理人员，帮助他们站立在技术浪潮之巅。文章主题包括需求、设计、构造、工具、项目管理、过程改进、维护、测试、教育和培训、质量、标准等等。

投稿指南：

www.computer.org/software/author.htm

更多细节：software@computer.org

Software

生了，这样你不会重复同样的错误。如果你已经知道你能做什么、什么问题已经解决了、什么问题还需要解决，你就会知道还需要做些什么。这就是该怎样使用计算机，甚至是怎样处理其他的一般事情。在这方面，计算机并不特殊、并不唯一。它们只是很酷。

Dag Spicer 是计算机历史博物馆的高级管理员。联系方式：spicer@computerhistory.org。



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

一位IBM销售员的宏大叙事， 1974–1981

撰文 James W. Cortada

1973年，我在佛罗里达州立大学取得现代欧洲史博士学位，我以为自己将成为一名历史学教授。但你也看到了，命运自有安排。我在1974年加入IBM，成为一名销售员，直到1981年到IBM销售学校做老师。我确实有些故事可以说给你一听。

让两个销售员一起坐在酒吧里，他们俨然原始社会里围着火堆而坐的两名猎人，动情回忆着盛大的凯旋和可憾的漏网之鱼。IBM的销售文化富有故事成分，它是个人成长和师徒制的一种形式。因为它所扮演的角色鲜少（甚至从来没有）被那些研究IBM的评论家和历史学家讨论过，我想我值得用几个例子来做一番展示。

经济和政治压力

虽然IBM有很了不起的营收数字，它的产品也很棒，但这家公司在1969年到1982年间整体面临着一个生死攸关的问题：美国政府的反垄断诉讼。我于1974年进公司前这场官司就已开打，直到里根执政期开始才落下帷幕——彼时我已离开了销售部。它直接影响了我作为一名销售员的工作，也导致整家公司以及我们所有前线人员在销售策略和销售活动中减少了“侵略性”。主动进攻本是任何优秀销售文化的标志——不论在IBM或其他地方。我认为它甚至使我们放慢了产品推陈出新的速度。谨慎行事和反复检查成了常态。正式的营销准则被贯彻执行，我们不得不阅读并签署一份表格，表明我们已经阅读过它们，若有任何违规行为，将立即被解雇。我2012年底离开IBM时这些准则都还在施行，其中的条例在过去几十年里没有本质的变化。

这些准则以相当的篇幅详述了法律允许或不允许的事。它禁止说竞争对手的坏话，虽然我们可以拿自己的产品和别人的产品做事实上的比较，比如对成本、速度、同类设备的性能、硬件和软件的功能做比较。

不论是否被允许与人竞争，都有定价方面的规定要遵守。比如，某个客户已经成为我们竞争对手的忠实买家了吗？如果是，那么我们就不能再去搅和破坏这种忠诚度了。营销准则搞得整家公司都神经兮兮的，在销售第一线也是，因为我们不愿冒被解雇的风险。不难理解，我们的管理层就更是小心翼翼了。

我们公司的架构方式是这样的：如果政府要求拆分公司，我们也能做到，还是能在行业里存活下去（到了七十年代末我和其他许多员工都相信分拆是可能的）。大客户（我们的企业和政府部门客户）被聚集到了数字处理部（DPD），中小客户被划入通用系统部（GSD）。产品部门也按大、小型系统划分，产生了各自的销售、售前售后服务和产品部门。到了七十年代末，这种划分已经减少了各个销售和服务部门间的合作，甚至使得大家各有自己的销售分部、售前售后工程服务部，以及员工组织和人事部门（有些是跨部门的！）。我们多少都经历了这一分拆过程。此外，到了七十年代末，每月的分公司会议上似乎永远都在宣布更多的改组决定，一些是出于明智的业务上的原因，另一些则可能是因为反垄断诉讼。

某天，我和销售部的一个同事埃里克·史密斯（Eric Smith）一同到访位于纽约麦迪逊大道590号的地区总部。反垄断案的审理就在附近进行。

我们决定前去一探究竟。几年来，那里有一间法庭专门用来审理IBM这桩官司。法庭的一侧从前到后是一整排靠墙的柜子，里面陈列着政府文件。另一侧是IBM的一两台终端和一台打印机，它们连接的主机里存有超过十亿个文件。我们还去瞧了瞧IBM及其法律顾问设立的一个反垄断指挥中心，那里看起来像一个巨大的IBM分公司。我猜那儿有两三百名号人吧，包括律师、律师的下属以及IBM的员工。反垄断官司就像一大片乌云笼罩在整个公司之上。它让我们花费了数亿美元，耗费了管理层大量时间。不过，我常常惊叹，那段时期我们还不断地推出新产品，究竟我们是怎么做到的？

然后就是整体经济和整个计算机行业的状况了——尽管这不是我和其他销售代表们通常会去关心的事，但它们毕竟影响了我们的业绩。随着技术的不断应用，以及各种各样的设备和供应商涌入市场，数据处理的运用在扩大。一年年过去，我注意到自己手头的每个产品都有了更多竞争者，甚至当美国无线电公司（RCA）和阿姆达尔公司（Amdahl）试图成为市场主导力量时，我为了守住客户打了不少惊心动魄的肉搏战。到了七十年代末，我的销售范围里到处可见日本计算机企业，此外还跑出来很多不靠谱的二手数字设备销售员，企图以比我低二到四成的价格出租老旧的IBM设备。

不过，我们是在高通胀时期做销售，到了1981年初的几个月，我的客户要面对两位数的通胀。这是非常严重的问题，尤其对我而言，因为我在卖东西时会更强调产品在经济上的合理性，而不会在客户关系、技术特点或辉煌的销售业绩这类上扯嘴皮。七十年代中期，一轮全球性的计算机行业衰退已经过去几年，此时资金预算已开始松动，像我这种销售方式变得尤其重要，但到了我大概要离开销售部以及PC产品开始推出时，资金预算又开始吃紧了。

国家环境保护局（EPA）在1978年左右出了新招，我那些身在造纸和林产品行业的客户被要求花数百万美元购买控制污染设备，原本用来实现

电脑化的资金看来要被挪去改造工厂设备以求保护空气质量了。我不得不帮我的客户学习如何用计算机测量各种污染物、如何将我们的多用途计算机和那些正被安装起来的高度专业化过滤系统连接起来。与此同时，我努力确保人们需要更多的计算机和外围设备来完成这些工作。从确认问题、成本、解决方案到确保执行，轻轻松松就花了我一整年时间。

冷战还在持续，所以我们还得担心一些设备不翼而飞。有相当一部分从法国流入了中东欧地区，这违反美国的法律。我的客户手里有两整套设备（一大一小）“找不到了”或者“发错了地方”，我怀疑都去了那里。

设备丢失是冷战时期计算机历史的重要内容，虽然在事情发生当时我们大多蒙在鼓里。苏联当时决定以美国产计算机为基础，统一东欧和中欧的计算机标准，在大型计算系统方面则完全依赖IBM的硬件和操作系统。因此，东欧人重新改装了我们的产品，把尽可能多的机器偷运到自己的国家，通常把法国作为从西往东的通道。多年后，我的同事们总结道，多达100套IBM系统可能都是这样消失的，而其中两套就是我的客户经手的！

此外我还有那么两三个客户是我的老板不想了解太多的，其中包括一间药店，它买了全套配置的System/370 Model 145放在地下室，还配备了操作人员。那只是一家药店啊！我已经学乖了，对某些客户不要太过好奇。在新泽西卖东西有很多好玩的事，哪里都有冒险之旅。

销售关系

1981年，我快要去IBM销售学校做老师了。8月，公司推出了个人电脑，我们分公司有一个三四人的团队要在分公司会议上宣布推出新的PC产品，我是其中一员。我们准备了幻灯片和小册子来介绍产品的开售日期、产量和定价等。我已在分公司工作多年，却头一回碰到这种事：我和同事要宣布推出新产品，却不知道这产品是要卖给谁、怎样证明成本的合理性（幻灯片上有针对这个问题的提

议，但太过理论化）、它在商业中能有怎样的应用。不过，我们仍能意识到这是个设计极为出色的系统：恰当的容量、适用的编程语言、妥当的应用软件，或许也是合理的价格，而且外观很棒。所以我们在会议上尽力推销它：“让你们的客户来拿几台去捣鼓捣鼓、学习学习。”过了一阵，分公司里装起了几台PC，这样销售代表们自己也能学习了。效果很不错。我感觉很快将发生翻天覆地的变化了，我很高兴自己将要离开销售第一线，做一名旁观者了。

1975年12月，我在纽约恩迪克特的销售学校接受首次培训。我们住在托马斯·约翰·沃森（Thomas J. Watson Sr）和他太太盖的一个宅子里。这里很像家庭旅馆，专门用来为IBM的员工和客户提供食宿。某天晚上，我们这一群人听说美国著名的热狗和加工肉类制造商Oscar-Meyer的CEO就坐在这间宽敞的餐厅里附近的一个座位上，他的一些高管和客户团队也在。这家公司在电视上和电台里有一首广告歌曲放了几十年，大部分美国孩子都知道歌词，更重要的是他们都有一个香肠形状的Oscar-Meyer口哨。我们这一桌上有人出了个好主意：到这个CEO面前去唱这首歌。然后我们就都站了起来，走过去唱了。我记得餐厅里的美国人都和我们一起唱起来，欧洲人则一脸茫然。他们桌上的客户团队可能很烦我们，但那个CEO很开心。

第二天，我们因为对客户做了这样“愚蠢透顶”的表演而受到严厉批评，并警告我们不得再犯。后来，我听说数字处理部的总裁第二天来这里开会，那名CEO对他说，IBM的人太棒了，他为昨晚的大合唱感动不已。而这首歌之所以会出现在广告中，正是他在多年前批准的——那会儿他还是Oscar Meyer的CEO！

新上的漆

在一家大型制泵公司，我一再劝他们升级主机，但他们的DP经理（数字处理部经理）不同意，因为他不敢再问他的老板要更多钱了。于是我让

IBM的一位系统工程师仔细检查了这台机器的容量，看看它离宕机还有多远。而他告诉我已经到达极限了。我知道这家公司在12月和1月要总结一年的财务数据，需要做更多运算。这让我惊出一身冷汗：他们没法做这事了。那位擅长捣鼓计算机的工程师向我确认了这一点。更糟的是，因为这家公司的会计和财务系统是一体化的，所有运算处理都必须在这台主机上完成，不可能有任何运算被外包。然而我还是没法让那位又蠢又倔的数字经理听我的。我知道，如果那些运算完成不了，我会被这家公司的CFO（天知道还会有什么人）质问为何知情不报。我想我不得不去找这个CFO了，尽管我可能会得罪了DP经理而丢了这个客户，然后因为搞砸了事而被我的老板惩罚，但我觉得我别无选择。

我和CFO私下碰了头。我用大白话告诉他出了什么问题，解释了不作为的后果。我还递给他一封信，用白纸黑字（用公司的正式信纸信封）重申了我的意见。这样，如果他们没有任何动作，我可以据此证明IBM已经提醒过他了。我猜那封信让他很震惊，同时也有些恼火——我竟然背着他的下属做这件事。我扔给他许多技术数据，建议他绕过DP经理找人探讨一下我发现的问题，然后再决定怎么做。我就不再管这件事了。

几天后，他请我参加电话会议。会上，他告诉我他按我的建议做了，公司的技术人员认同我的发现和建议，他们得马上弄一台更大的机器。他还找DP经理谈了，两人大吵一架，最后CFO说了算，自己向我发来了订单，要买我们的机器并寻求指示——我说该做什么就做什么。

通常，这样一台机器和必要的配套设备的订货交付时间是半年到一年。现在他想要我们在11月底前全部安装完毕。也就是说，还剩两个月。

我们全力以赴。下了订单后，看到了可怕的发货日期。我、系统工程师、分公司的行政人员，以及工程服务人员于是开始动用一切关系，把全美和加拿大所有的IBM仓库都问了个遍。我们甚至还去打扰那些快要收到硬件的客户，问他们能不能把东西让我们先用。我们从附近一个仓库“借来”一

台打印机——公司过了半年才发现它丢了。我从一个仓库里拉了些电缆，又从工程部那里再找了些电缆和零部件（据说这个部门在遍布新泽西的众多数据中心都盖了架空地板，下面储藏着零部件）。我借来的一辆小货车已经被这些东西塞满了。

我们找到了所有需要的设备。有一台机器是明黄色的，另一台橘色，其他还有蓝色的、灰色的。真是个五彩斑斓的美妙景象。与此同时，那名DP经理被炒了鱿鱼，一个比较正常的经理接手了他的工作——后来我们发现此人是一名真正具有远见卓识的领袖。当我们听闻制泵公司的CEO和CFO都要下来看看“新”设备时，我们的工程服务人员把所有设备上盖着的金属罩子都扒拉下来，把东西全部拖到停车场，一股脑儿都喷上了IBM蓝——机器看起来就和那家公司以前用的一个样了。那位CFO给的唯一评论是：数据中心闻起来怎么有股新漆的味道。没人再吭声了。

这个故事值得一说，有几点原因。首先，我们总是一心想着用计算机真正解决客户的问题，这次显然做到了。第二，它展现了创造力、进取心、团队合作，以及让IBM系统正常运作的能力，或者说克服这一系统的能力——看你从什么角度看了。第三，我需要那4万个点的业绩来达到我的销售目标，并成为当年度的IBM“百分百俱乐部”成员。9月份的时候，这个目标看起来是不可能实现了，除非我有魔法。我去会见那位CFO就是这个魔法。这笔生意也非常契合销售员的信条：“客户说不之时，销售才真正开始。”它也证实了“最好的销售员排除一切障碍敲定交易”的神话。这场经历堪称完美。

私售时间

我的另一个故事比较严肃。在那些日子（1979），我们把主机出租给一家老客户，收取固定的月租费，对方可以每月使用这些机器一定量的时间。我记得是每月200小时不到，超出的时间会产生额外费用，以小时计。我注意到，在某个时段，机器的使用量会迅速上升并保持高速运转，但我

我们的人
把所有设备上
盖着的金属罩子都扒拉下来，
把东西全部拖到停车场，
一股脑儿都喷上了
IBM 蓝。

并不知道这家公司对这些设备还有什么新用途。我当时想，会不会IBM弄错了，多算了人家钱，那我的客户真是倒霉。所以我让负责这个客户的系统工程师去给那套系统装了个监测器。她回来后向我确认说，运算量确实增加了。我让她晚上再去看看机器在处理些什么东西。

两周后，她汇报说这个数据中心在为一个地方工会处理医疗福利数据。但这家公司并不是医疗相关企业啊。我让她打印些运算数据出来，算算处理这些数据会产生多少额外的小时数。我还注意到我的客户最近开了一辆全新的凯迪拉克，还有几套新西装。单凭他的薪水吗？我现在相信这名DP经理在干些非法的勾当。我该怎么办？如果我向自己的上级汇报或者去找IBM的律师，他们会叫我啥也别管。我把证据拿给一个律师朋友看，他说有足够的证据显示其中必有猫腻。

与此同时，我借口要审核IBM的账单，请这家公司的财务部查查看。他们发现，公司内外没有任何人为那些额外的运算时间买单，DP经理从他的预算里列支了这笔费用。

我决定去找这家公司的老板。我担心如果不这么做，某天IBM会被问责。我知道这个老板有加班的习惯，所以某晚七点，我假装很随意地路过这里，我很清楚这时其他人都下班了。我敲开他办公室的门，一五一十地告诉他，他的下属可能干了什么事。这名下属可是个老员工，在这家公司工作了35年。

他听了很不开心，明确告诉我说，如果是我搞错了，他要去找IBM的董事长，让他开除我！然后他打电话给自己公司的律师，把他叫到办公室，我们又一起把证据过了一遍。这个律师同意我的评估。老板要他尽力证明那名雇员的清白，并请我继续收集证据，而我们对外都要三缄其口。

长话短说。三周后，他们和那名DP经理当面对质了，对方承认私自出售公司计算机的运算时间，自肥了腰包。他被解雇了。那位老板显然很感激我，想要告诉IBM的所有人我做了件多么出色的事。我请他不要声张，不能告诉IBM的任何人，因为我没有遵照公司的规定向我的老板和律师汇报，恐怕会受到严厉的训诫。

悠久传统

所有IBM销售员都有一堆故事可讲。我说的这些已经足以展现这家公司的销售文化和形态。通常，我们的故事都是关于：如何以杰出的客户服务解决客户难题这类英雄行为（比如制泵公司的故事）、做过的蠢事（热狗公司之歌）、不得不调和IBM和我们的客户之间的矛盾（刚刚说的那个故事）。

当我升职要离开分公司时，按照IBM延续至今的传统，我的部门为我开了个派对，我所有做过的蠢事都被一次性挖出来笑个够。这种派对有时也会邀请一个客户（通常是DP经理）一起来闹。但当我来到派对时，吓了一大跳：除了分公司的同事和工程服务部的同事，还来了两家客户公司的总裁、制泵公司的CFO，和至少五六个DP经理。几乎所有人都想敬我一杯（说我的笑话）。唯有两名总裁很专业，彬彬有礼、言语得体。他俩称赞所有的IBM人（不仅仅是）既专业又正直。就这样，我在IBM最初的销售生涯划上了句号。

回顾我在IBM“前线”工作的二三十年，我可以总结出一点：我在七十年代做销售的所有经历，也正是在美国和其他国家的成千上万IBM人的经历。我记得那时有人告诉我，我们工作的方式、我们受到的压力和面对的问题，也都呼应了我的前

辈们昔日的经历，甚至在IBM于五十年代中期开始专注于计算机之前就是如此。我那时的导师、资深营销代表格斯·凯恩（Gus Kane）在我们一起驱车拜访一个又一个客户时，告诉了我许多他在六十年代的往事（也是在传授经验）。还有些人告诉我四十年代的故事，或者来自其他部门的不那么久远但同样似曾相识的故事。那些故事的结局，还有我的经历，也都是其他人的遭遇的写照。

James W. Cortada 是明尼苏达大学查尔斯巴奇研究所（Charles Babbage Institute）的高级研究员。在IBM度过文中所述的时期后，他担任了一系列销售管理职务，包括担任分公司经理（在威斯康辛州的麦迪逊）和更多人事管理职位，而后在九十年代转战咨询部门。到九十年代末，他晋升高层主管，负责管理IBM全球咨询经理网络。2004年，他加入IBM商业价值研究院，当时的计划是两三年后退休。不过他一直在那待到了2012年。要联系他可致信 jcortada@umn.edu。



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.



讨论
最新文献

寻找
求职机会

吐槽
科研生活

就到

一站式科研社交网站



—— 科研圈 www.keyanquan.net



北大研发疫苗“万能”制备方法



010-85321181

contact@keyanquan.net

《环球科学》旗下产品

深度学习：人工智能的复兴

撰文 Yoshua Bengio

上个世纪50年代，计算机在西洋跳棋(checkers)上击败了人类，还可以解决代数题，在世间引起了巨大轰动。60年代时，人们充满信心地认为科学家很快就可以通过硬件和软件用计算机模拟人脑，而且这种“人工智能”很快就能在任何任务上与人类匹敌。1967年，麻省理工学院的马文·明斯基(Marvin Minsky)甚至宣称，人工智能问题将在一代人的时间内被彻底解决(明斯基教授于今年1月去世)。

现在看来，人们当时的乐观情绪显然过于幼稚了。尽管软件工程师已经设计出了旨在帮助医生更好诊断疾病的软件，开发了用于识别图片内容的模仿人脑的网络模型，但是效果均不理想。早期的这些算法不够复杂，而且依赖的海量数据也是当时无法提供的。另一方面，人类智能十分复杂，要想近似地模拟人脑功能需要巨大的计算量，可当时的计算机实在太慢，无法胜任。

到2005年前后，科学界几乎已经没人还看好机器可以达到人类的智能水平。在那个时候，“人工智能”似乎都已经变成了科幻的代名词。科学家和作家们往往将20世纪70年代到2005年前后的这段时间称为“人工智能的冬天”——梦想幻灭的日子。

10年之后，一切都不一样了。从2005年开始，人工智能的形式出现了巨大变革。这一切源于“深度学习”(deep learning)技术的出现，这一方法原本指的是从脑科学中汲取灵感以制造智能机器，但后来已经自成体系。近年来，深度学习已经成为了驱动人工智能领域发展的最主要力量，各大信息技术公司在这方面共掷下了数十亿美元的资金。

深度学习指的是用计算机模拟神经元网络，以

此逐渐“学会”各种任务的过程，比如识别图像、理解语音甚或是自己做决策。这项技术的基础是所谓的“人工神经网络”，它是现代人工智能的核心元素。人工神经网络和真实的大脑神经元工作方式并不完全一致，事实上它的理论基础只是普通的数学原理。但是经过训练后的人工神经网络却可以完成很多任务，比如识别照片中的人物和物体，或是在几种主要语言之间互相翻译等等。

深度学习技术彻底改变了人工智能研究的面貌，让计算机视觉、语音识别、自然语言处理和机器人领域重新焕发生机。2011年，首个语音识别产品面世，也就是大家非常熟悉的Siri。不久之后，用于识别图像内容的应用也成熟了起来，该功能现已被整合进了Google Photos的图片搜索引擎。

如果你嫌手机菜单操作起来太麻烦，那你一定会喜欢Siri或是其他智能手机助手给你带来的使用体验的飞跃。假如你对于几年前的图像识别软件还有印象，那你一定记得当时的识别效果相当糟糕，有时甚至会误把吸尘器认作犰狳。而现在这个领域早已发生了翻天覆地的变化，现在的软件在某些情况下识别出猫、石头或人脸的准确率甚至和人类相当。人工智能软件如今已成为了数百万智能手机的标配，拿我自己来说，我已经很少用手输入信息了，都是直接对手机说话，有时它也直接用语音回应我。

这些技术进步一下子为进一步的商业化打开了大门，人们对人工智能的热情日益高涨。各大公司竞相招揽人才，擅长深度学习的博士一下子变成了供不应求的稀缺商品。这个领域的很多大学教授(可能是绝大多数)都已经从象牙塔被拉到了产业界，企业为他们提供了一流的研究环境与设施，以及优良的待遇。



在利用深度学习解决各种问题的过程中，科学家们取得了惊人的成就。几个月前，神经网络模型击败了世界顶级围棋选手李世石的新闻占据了各大媒体的首页。机器在其他领域超越人类的步伐也在加快，而且不局限于竞技游戏。最新的深度学习算法可以从磁共振图像中准确检测出心脏是否衰竭，准确度甚至能和心脏病专家相比。

智能、知识、学习

人工智能为什么在之前的几十年会遇到这么多挫折呢？因为对机器来说，学习新事物是很难的。编写计算机程序需要把任务用很规范的方式写成一条条具体的规则，但世界上大部分知识并不是这样的形式，所以我们很难直接给电脑编写程序，让它实现人类可以轻而易举完成的任务，例如理解语音、图像、文字或是驾驶汽车等。科学家之前尝试过直接编程，将复杂的数据集按照某些规则进行细致的分类，希望以此让计算机渐渐拥有类似“智能”的东西，可是这种尝试并无建树。

但是深度学习突破了这个困境。深度学习属于人工智能领域内一个更宽泛的概念——机器学习，即根据某些基本原理训练一个智能计算系统，最终使机器具备自学的能力。其中一条基本原理，就涉及到人或者机器如何判断一个决策是好是坏。对动物来说这就是演化原理，“好”的决策意味着提高存活和繁衍几率；在人类社会中，“好”决策可能是指提高地位或产生舒适感的社会活动；对机器而言，以智能汽车为例，一个决策的“好坏”则可能被定义为其行驶模式在多大程度上接近优秀的人类司机。

在特定情况下做出好决策所需的知识并不一定能轻易地用计算机代码表示出来。比如说一只小鼠，它会熟悉自身周围的环境，而且本能地知道该去嗅哪里，该如何移动四肢，该怎样寻找食物或配偶，并避开天敌。但没有哪位程序员可以用一行一行的代码写出一个能做出这些行为的程序，虽然这些行为所需要的知识都存储在小鼠脑内。

在让电脑学会自学之前，计算机科学家必须搞

清楚一些基本问题，比如人类是如何获取知识的？人类的知识部分是天生的，但大部分都是从经验中学习而来。自上个世纪50年代以来，科学家就一直在寻找动物和人类（或是机器）从经验中学习知识的基本原理。机器学习的研究致力于建立一系列所谓“学习算法”的流程，可以让机器从样例中学习。

机器学习在很大程度上算是一门实验科学，因为还没有普适的学习算法出现，对于不同的任务，科学家需要开发不同的学习算法。每个学习知识的算法都需要在相应的情况下，用相应的任务和数据进行测试，有时候任务是识别一副图像是不是日落，有时候任务是把英语翻译成乌尔都语。我们没法证明一个算法可以在任何任务中都比别的算法表现更好。

人工智能专家用正式的数学语言描述了这个原理——“免费午餐”定理。这个定理指出，不存在可以解决所有实际问题的“万能”机器学习算法。可是人类的行为似乎违背了这个定理：我们的大脑中似乎存在一些通用的学习算法，能让我们学会各种任务。要知道，这并不能完全归结于演化，因为很多任务我们的祖先都没有做过，比如下棋、建造桥梁，或者是研究人工智能。

这说明，人类智能利用了关于世界的一般性假设，启发科学家创造出具有通用智能的机器。正因如此，科学家以大脑为模仿对象，开发出了人工神经网络。

大脑的计算单元是一些叫做神经元的细胞，每个神经元可以通过叫做突触间隙的小缝隙向其他神经元发放信号，神经元跨越间隙传输信号的效率叫做突触强度（synaptic strength）。在神经元“学习”的过程中，它的突触强度会增加，也就是说当它受到电信号的刺激时，它更有可能向邻近神经元传输信号。

受脑科学的启发，计算机科学家开发出了人工神经网络，用软件或硬件实现了虚拟神经元。人工神经网络是人工智能领域的一个分支，该领域在早期又被称为“联结主义”（connectionism）。计

算机科学家认为，神经网络可以接收视觉或听觉输入，渐渐改变神经元之间的连接强度，最终网络连接会形成一种特定模式，让神经元的电信号可以提取出图像内容或是对话短语的特征，从而实现一些复杂的任务。随着人工神经网络接收的样例越来越多，学习算法不断调整神经元连接的突触强度，可以更准确地对输入数据（如日落的图像）进行表征。

给自己上一课

最新一代的神经网络则在“联结主义”的基础上更进一步。在神经网络中，每个突触连接的强度都由一个数值决定，代表了这个神经元向其他神经元传递信号的可能性。以图像识别为例，每当网络接收一幅新图像时，深度学习算法就根据当前输入对连接强度做一点小调整。随着见过的图像越来越多，网络连接强度就会一点一点地变为最优连接模式，使网络预测图像内容的准确度不断提高。

为了得到最好的结果，目前的学习算法仍然需要人的密切参与。大部分算法均使用了叫做“监督学习”（supervised learning）的技术，这是指在训练过程中，每个训练样本都配有一个与内容相关的标签，比如对一幅日落的图像，同时还会附上标注“日落”。监督学习算法要做的，就是让网络以这幅图像作为输入，并输出图像中央的物体名称。在数学上，把输入变成输出的过程叫做函数（function）。机器学习的目标就是找到可以实现学习任务的函数所对应的参数，比如突触连接强度。

用死记硬背的方式来记住正确答案固然直截了当，但也毫无意义。机器学习的终极目标是让算法明白“日落”到底指的是什么，然后可以认出任何一幅日落的图像，不管这幅图在训练过程中有没有出现过。

要获得能够识别出任何日落图像的能力，也就是所谓的从具体示例中进行概括并推广的能力，这才是所有机器学习算法的主要目标。事实上，科学家评价网络的学习效果时，就是用该网络从未见过

**机器学习的终极目标
是让算法明白“日落”
到底指的是什么，
然后可以认出任何
一幅日落的图像。**

的图像来测试其识别准确率。由于在每一类别下面都有无穷多种变体，要想识别从未见过的图片并非易事。

想让深度学习算法从大量样例中概括出抽象的概念，除了输入样例以外还需要其他信息，比如一些关于数据的额外假设，或是对特定问题而言最优解大概该是什么形式的猜想。具体来说，图像识别软件中常用的假设之一，就是当给特定网络输入相似的图片时，网络的输出也不该有太大变化。比如把一张猫的图像中的某几个像素进行改动，那么图像依然是猫而不会变成狗。

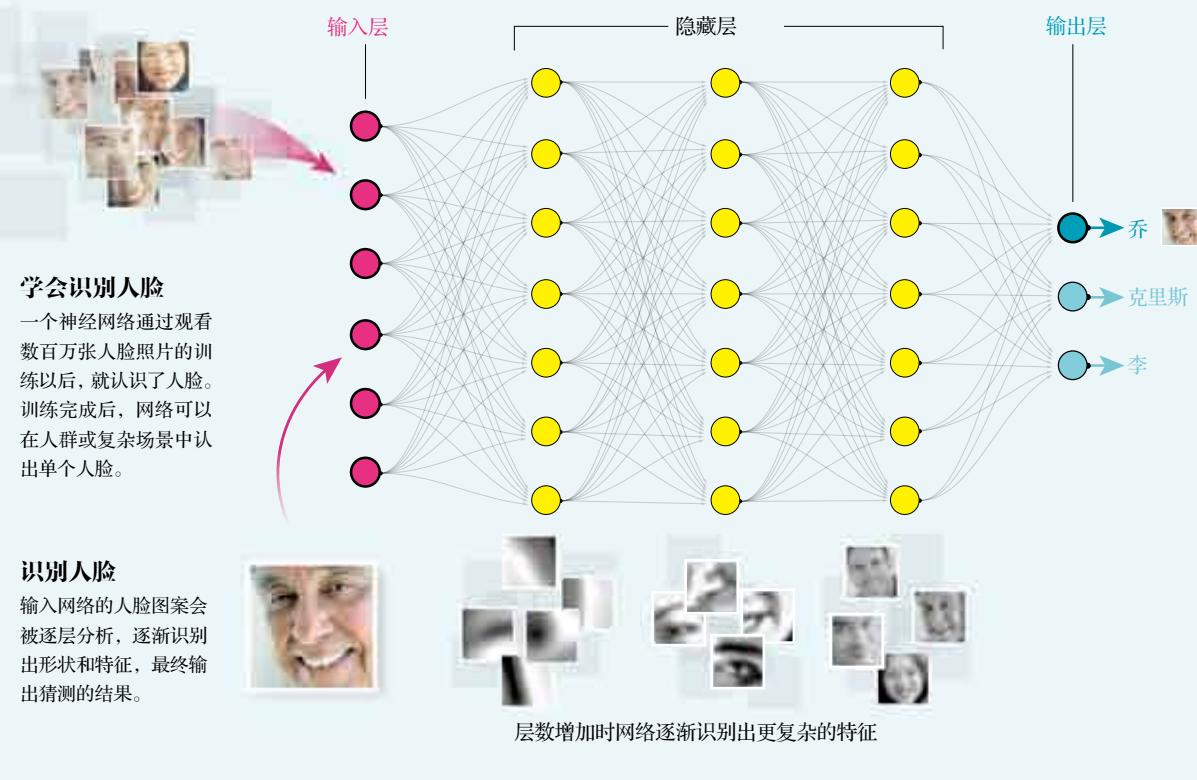
有一类基于这些假设的神经网络叫做“卷积神经网络”（convolutional neural network），正是这项关键技术推动了人工智能的复兴。深度学习中的卷积神经网络包含很多层神经元，层与层之间通过特定的方式连接，使得输出结果对图像中央物体的变化（例如位置稍稍偏移）不会非常敏感，这样的网络在训练完成后可以认出从不同角度拍摄的人脸。卷积神经网络的设计灵感源自人脑视觉皮层的层状结构，它在人脑中负责处理从眼睛接收的图像信号。卷积网络中包含很多层虚拟神经元，“深度”学习的“深”指的就是这个。这类网络涉及到的计算量十分巨大，但是它可以学会很多任务。

深度学习

从实践角度看，大概是10年前出现的一些研究进展，让深度学习从一些技术创新中脱颖而出。

不断进化的类脑网络

人类大脑皮层中的神经元之间有着复杂的连接，这启发研究者设计出了模仿这种连接的算法。神经网络可以通过大量图像的训练，学会识别出图像中的人脸。当网络学习了“人脸”这一类别（例如，区别于“手”），并识别出不同的人脸后，网络可以认出以前见过的人脸，即使新的输入图像和训练时用的图像存在细微差别也没问题。识别人脸时，网络首先对输入层接收到的图像的每个像素进行分析，接下来网络会在下一层认出人脸的一些几何形状特征。随着层数变深，“眼睛”、“嘴”以及其他面部特征会在网络中渐渐出现，最终在高层形成一幅复合人脸图像。这时，网络可以在输出层猜测面孔的身份，是“乔”还是“克里斯”，或是“李”。



当时的人工智能领域正跌入几十年来的最低点，但一个由政府和私人资助者支持的加拿大组织重燃了科学家的研究热情。这个名为加拿大高等研究所（Canadian Institute for Advanced Research，简称CIFAR）的组织资助了由多伦多大学的杰弗里·欣顿（Geoffrey Hinton）领导的一项计划，计划参与者还包括纽约大学的杨立昆（Yann LeCun）、斯坦福大学的吴恩达（Andrew Ng）、加利福尼亚大学伯克利分校的布鲁诺·奥尔斯豪森（Bruno Olshausen）、本文作者本希奥以及其他

一些科学家。当时这个研究方向并不被世人看好，导致相关工作很难发表，研究生们也不愿意在这个方向发展。但是我们中的一小部分人还是有强烈的预感：这是一条正确的道路。

人工智能之所以在当时不受待见，部分原因是科学家普遍认为训练好一个神经网络几乎是不可能的，很难找到可以高效地优化网络以提升其性能的学习方法。“最优化理论”（optimization）是数学的一门分支，它所做的事是尝试找到能达到一个给定数学目标的参数组合，而在神经网络中，这

些参数被称作“突触权重”(synaptic weight)，反映了信号从一个神经元通向另一个神经元的强度。

深度学习的最终目标是做出准确的预测，也就是将误差控制在最小的范围内。当参数与目标之间的关系足够简单时——更确切地说，目标是参数的“凸函数”(convex function)时，可以逐步对参数进行调整，直到接近最优解(或称“全局最小点”)，即整个网络的平均预测误差最小时。

然而，神经网络的训练过程并不简单，需要另一种“非凸优化”(nonconvex optimization)过程。非凸优化就更麻烦了，很多研究者甚至认为这一障碍是不可克服的。学习算法在运算过程中可能会陷入所谓的“局部最小点”处，此时，轻微调整参数值就无法减小预测误差，也就无法进一步提升模型性能了。

科学家一直觉得神经网络难以训练的罪魁祸首是局部最小点，这个误解直到去年才得到澄清。本文作者在研究中发现，当神经网络足够大时，局部最小点造成的麻烦会小很多。对于一个稳定的网络，大部分局部最小点都反映了模型已学到的一定程度的知识，而且几乎和全局最小点处的情况一致。

即使知道最优化问题在理论上是可解的，研究者在训练包含两三层以上的神经元网络时依然常常失败，当时很多人认为继续这类研究毫无希望。从2005年开始，由CIFAR支持的一些工作不断取得突破，成功解决了这个问题。2006年时，我们找到了训练深层网络的算法，所用到的技巧就是逐层训练。

后来，在2011年我们又发现，只要改变一下每个虚拟神经元的运算，让它们表现得更像真实的神经元，就可以训练包含更多层神经元的深层网络。我们还发现，在训练过程中给网络中传输的信号上加上噪音，也可以让网络更好地识别图像或声音，这一点也类似于大脑中的信息处理流程。

深度学习技术的成功依赖于两个关键因素。首先是计算速度的大幅提升：科学家借用了本为电子

游戏所设计的图形处理器(GPU)，使计算速度提升了10倍，这使得在有限的时间内训练大规模网络变为可能。让深度学习研究开展得如火如荼的第二个因素是海量带标记数据集的出现，这些数据集内的所有样例都配有正确的标记——如一幅图中有猫这个元素，便会被标上“猫”。

深度学习近年来的成功还有一个原因：它可以通过学习算法找到一系列运算操作，对一幅图像、一段音频或其他信号进行逐步的重构或分析，这里的步数，就代表着网络的深度。目前人工智能可以高效地处理很多视觉或听觉识别任务，都要求网络的深度足够大。事实上，最近的理论和实验的相关研究表明，假如网络深度太浅，是无法完成这些复杂的数学运算的。

深层神经网络中的每一层把接收的输入进行处理，再把输出传向下一层。越到深层，其所表征的概念就越抽象，因为这些层离最初的输入信息距离太远(见图框“不断进化的类脑网络”)。实验发现，深层的虚拟神经元往往对应了一些抽象的语义概念，比如代表了“桌子”的图像。即使在训练过程中网络从来没有看到过单独被标注有“桌子”的样例，一些深层虚拟神经元在处理信息时依然可以认出对应着桌子的图像。而且“桌子”这个概念还可以作为中间步骤，在更深的神经元层中产生更抽象的概念，比如“办公室场景”。

模式识别之外

直到不久前，神经网络的成功还大部分局限在识别静态图像方面，但是也有一些神经网络擅长处理随着时间变化的任务。以递归神经网络(recurrent neural network)为例，这种网络可以连续执行一系列运算，处理语音、视频或其他数据。时序数据由一个个单元(不管是音素还是词语)依序排列形成，递归神经网络处理这些输入信息的方式也和大脑的工作模式相似：当大脑在处理感官输入的信号时，神经元传递的信号也不断变化。大脑内的这种神经状态会把输入信号视作周围环境中发生的各种事件，进而输出合适的指令，

告诉身体该如何行动以达成特定目标。

递归神经网络可以预测一句话中的下一个词是什么，由此产生连续的词语序列。它们也可以完成更复杂的任务：在“读”完整个句子之后，猜测出句子的意思。然后再使用另一个递归神经网络，就可以利用前一个网络做完的语义分析，把句子翻译成另一种语言。

递归神经网络研究在上世纪90年代末到本世纪初也经历过低潮。本文作者的理论工作表明，这种网络在处理时序信息时，很难回忆起太久之前的输入，即输入序列中最早的部分。想象一下当你读完一本书时，是否还能逐字逐句回忆起书的开头？但后来的几项技术创新在部分程度上解决了这个问题，使网络的记忆能力得到显著提升。现在，神经网络会在电脑中留出一块额外的临时存储空间，专门处理分散的多段信息，例如分散在一个长文档中的各种句子的含义。

在经历了人工智能的寒冬之后，深度学习神经网络的强势回归不仅仅代表着技术上的胜利，也给整个科学领域上了一课。回顾人工智能的历史，我们必须意识到，哪怕一些好的想法受到当时的技术状态限制而看似前景渺茫，我们也不应放弃。鼓励多样化的科研课题与方向，或许有一天就会让一时处于低谷的领域迎来复兴。

Vol. 521, pages 436–444; May 28, 2015.

4. When Computers Surpass Us. Christof Koch; Consciousness Redux, Scientific American Mind, September/October 2015.

参考文献

1. ImageNet Classification with Deep Convolutional Neural Networks. Alex Krizhevsky et al. Presented at the 26th Annual Conference on Neural Information Processing Systems (NIPS 2012), Stateline, Nev., December 3–8, 2012.
2. Representation Learning: A Review and New Perspectives. Y. Bengio et al. in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, No. 8, pages 1798–1828; August 2013.
3. Deep Learning. Yann LeCun et al. in Nature,

Yoshua Bengio 是蒙特利尔大学计算机系教授，也是深度学习技术的先驱者之一，而深度学习技术正是人工智能复兴的关键因素。

《IEEE计算机图形及应用》(IEEE Computer Graphics and Applications, 简称CG&A)把计算机图形学领域的理论和实践联系在一起。《IEEE计算机图形及应用》提供了包括从某个特定算法到全系统实现在内的同行评议的深度报道。它为那些处于计算机图形技术前沿的人们提供了必不可少的资料。无论他们处于商界还是艺术界, 这本杂志都能让他们受益。

请点击: www.computer.org/cga

无论你在哪里，
都能紧随
IEEE计算机协会
的脚步。

IEEE computer society

| @ComputerSociety
@ComputingNow

| facebook.com/IIEEEComputerSociety
facebook.com/ComputingNow

| IEEE Computer Society
Computing Now

| youtube.com/ieeecomputersociety

IC 卡上的密码进化史

撰文 Jean-Jacques Quisquater）、Jean-Louis Desvignes



自从在4年前面世以来，非接触式支付获得了巨大的成功，并且正变得越来越流行。Visa欧洲宣称，从2015年5月到2016年5月，欧洲共进行了30亿笔非接触式Visa卡交易，这个数量差不多是一年前的3倍多。在商店中，有1/5的交易采用非接触式支付，而2013年这个比例仅为1/60。



在法国，非接触式支付的交易数量在一年内就增长了230%，在12个月内进行了1.94亿笔交易，交易额超过20亿欧元。消费者对于非接触式支付极为信任。

然而，非接触式支付所采用的近场通信技术（在不超过10厘米的范围内，读卡器通过无线电波与带有微型天线的银行卡芯片通信）并非无懈可击。当然，银行提供了众多防护措施来保卫你的钱包，比如每次取款限额为20欧元、每天非接触式支付数量设有上限等等。不过，卡片本身没有任何内置的防盗刷安全措施。

也没有什么措施能够阻止黑客用读卡器贴近你装着卡片的口袋，拦截你的银行数据。

百万美金赏金

实际上，非接触式支付面临的这些麻烦只是由来已久的IC卡安全问题的最新表现而已。本文的两位作者既是这段历史的见证者，也是这段历史的参与者。让-雅克·齐斯夸特对技术进步做出了贡献，而让-路易·德维涅则大力主张在IC卡中使用加密技术。在那时，加密技术还是专为军方和政府服务的。

历史总是在不断地重演。在IC卡使用的初期，也就是20世纪80年代左右，人们相信这种卡片是万无一失的，完全值得信赖。这种信任如此根深蒂固，以至于有人悬赏100万美元邀请黑客来破解。

1985年左右，在布鲁塞尔一场大型展览会上，飞利浦公司的一个高管打算分发几万张IC卡，让参与者来破解它们的4位密码，也就是个人识别码（PIN）。首位破解成功的人将获得100万美金的奖励。但是，考虑到分发的卡片和参与者的数量，简单的概率统计计算就能证明总会有人能够破解PIN码。而这和卡片的安全性毫无关系。本文作者齐斯夸特向飞利浦高管说明了这一点，但是他却不相信。和IC卡安全先驱之路易·纪佑（Louis Guillou）一道，我们打算黑掉一张卡片，来证明我们能够破解IC卡的密码——潘多拉魔盒就这么被打开了。



图中两张卡片是银行卡的前身。上面的卡片是飞利浦公司制造的双芯片卡；下面的卡片是布尔电脑制造的CP8卡片。它们都带有保障卡片安全性的关键：处理器。



法国萨基姆公司（SAGEM）生产的DCS500保密电话是第一款利用IC卡储存密钥的数字加密器设备。插卡输入密钥就能加密或解密信息。从20世纪90年代开始，法国当局和军队就开始使用这个设备。

我们不打算在这讲述IC卡的整个研发史，因为要全面介绍IC卡研究者和他们进行的试验，得一整本书才行。我们将要讲述的是，IC卡是怎样通过引入加密技术而变得值得信赖，同时成为密钥的守护者的（而密钥本身就是加密方法的核心）。

万无一失的安全性？

当IC卡出现的时候，由于性质使然，它看上去天然就是安全的，不会受到恶意查询和金融诈骗的侵害。的确，集成电路对于普通人来说是无法逾越的技术壁垒，那些可以读取或是改写当时常用的磁条卡的技术，对IC卡是无效的。

最初开发的IC卡和相关专利并没有涉及加密技术。不管是法国科幻作家赫内·巴赫札维勒（René Barjavel）的小说《冰人》（1968年）中的贡达瓦人（在书中，贡达瓦是消失的文明，贡达瓦人用一把钥匙登录到中央电脑里就可以付款），还是日本、德国或美国的第一批相关专利，抑或是罗兰·莫雷诺（Roland Moreno, IC卡发明者）在1974年设计的智能“戒指”，都没有在芯片中使用加密技术，尽管当时已经有人提出了安全方面的考虑。

很自然地，银行卡成为了IC卡首选的应用领域。印有凸起字符的老式银行卡和它的读卡器，或是磁条卡，很快就都显得不够安全了，至少对欧洲的银行来说是这样。

但是，另外一个意想不到的应用却使IC卡技术实现了大规模工业化生产——电话卡。IC电话卡里面存有通信时长，这解决了投币电话亭兑换硬币难的头疼问题。

IC卡的另外一项应用并不太广泛，但是却触及了具有影响力的人群，因此这个应用——付费电视，也为IC卡的普及作出了贡献。此外，当我（本文作者德维涅）掌管法国计算机安全服务中心（法国国家计算机安全局的前身）时，正是付费电视让我开始思索IC卡的安全性，并向当局发出有必要尽快处理IC卡安全问题的警告。

1996年，英国天空电视台（BSkyB）就成为一起大型盗版事件的受害者，盗版者为一家爱尔兰公司。这家爱尔兰公司公开廉价销售可以看天空电视台节目的伪造电视卡。电视台此后为了打击盗版而出台的一系列补丁也无力对抗盗版者的快速应对。

正是由于在开发新技术之初对安全问题缺乏

严肃考虑，才产生了这个盗版黑色产业链。那时，我用这个例子说服法国卫生部使用加密技术来保护该部门所使用的卡片（医保卡和职业健康卡）。复制或伪造的卡片在外泛滥会带来多么严重的风险，根本无需多言。

公众对IC卡的使用仅仅是风险的一方面。10年前，当我是国防部“密码官”的时候，我的工作就是管理加密设备的密钥，当时这方面的状况不太令人满意。

像那个年代所有负责这个领域的官员一样，我脑子里总想着汉斯·提罗·施密特（Hans-Thilo Schmidt）背叛的往事。施密特是德国密码机构的工作人员，早在第二次世界大战开始前几年就向法国情报机构出卖了德国恩尼格玛密码机的使用手册、密码本和破译加密信息的机器参数。在法国人和英国人（包括著名数学家阿兰·图灵）的先后帮助下，波兰人破解了这台密码机。施密特提供的情报极大地促进了盟军的破译工作。

直到1990年，政府间的加密网络（如北约）一直没有什改进。它们用的是对称密码系统，即用来加密和解密的都是相同的私钥。密钥的初始化（将同一个密钥分配给所有机器）是手工操作的，操作人员会在物理上接触密钥。

不管选拔操作人员的过程如何严格，都无法完全排除操作者泄露其接触的机密的可能。当然，为了减少这种可能性，保密机构当时也采取了一些不同的预防手段，比如用密码活页纸，每天撕去一页换新的，防止同样的密码长时间暴露。

保密机构也曾用过从某些游戏中借用来的刮涂层。后来又出现了在读取后无法倒带的打孔带盒子。还有电子密钥输入器，就是一个简单的储存器，但操作界面外行根本看不懂（不过成本很高）。

而我为了提高大规模加密网络的安全性，曾经反其道而行之，延长了一些密钥的密码有效期，减少了能够接触这些密钥的操作者数量。我把密钥原本的更新周期从24小时延长到1周，通过这种方式，把因操作人员走上歧途而泄露重大机密的可

为了用 IC 卡安全存储密钥， IC 卡自身必须是 真正的安全系统， IC 卡的芯片 应当具有一定计算能力。

能性缩小到原来的1/6到1/5左右。

这就是为什么，IC卡一出现（例如20世纪80年代出现的布尔CP8 IC卡），我就试图利用它来研发成本适中的新型加密设备。一开始IC卡只被用来安全存储密钥，比如法国萨基姆公司为政府开发的军用保密电话DCS500中的IC卡就是这个功能。

直到20世纪90年代末，法国国防部才开始在“密钥管理基础设施”项目中采用IC卡来储存安全证书，也就是说这时候已经开始采用非对称加密系统了。

这些安全证书一般被称为公钥（这类系统有2个密钥，公钥用来加密，私钥用来解密），是在70年代中期出现的。这种系统的广泛应用是与互联网的发展交织在一起。

但是，为了用IC卡安全存储密钥，IC卡自身必须是真正的安全系统。如果想要避免受到某些攻击的话，IC卡的芯片就应当具有一定计算能力。

举个简单的例子，储存在IC卡上的密码可用 来获取某种服务，比如付费电视和Minitel（法国的网络终端服务，已停用）一开始采用的就是这种方法。假设这张卡通过PIN码和某个用户联系在一起，这个用户就可以用卡中存储的密码来证明自己的身份。要储存该密码的话，芯片就必须有一个无法篡改的储存器，芯片本身也应该是无法篡改的。

必要的加密技术

另外，如果原封不动地传输密码，盗取密码显

IC卡的发明

IC卡是一张塑料卡片，上面装着至少一个存有信息的集成电路。IC卡可以追溯到1958年，当时美国德州仪器(Texas Instruments)公司的杰克·基尔比(Jack Kilby，集成电路的2位发明人之一，2000年诺贝尔物理学奖获得者)发明了第一个集成电路。把更多的晶体管集成到硅基片上，就能在单个芯片上制造出真正的计算机，例如后来出现的微处理器(英特尔公司在1969年制造的Intel 4004微处理器，以及在1972年制造的8008微处理器)。



集成电路令计算机实现微型化，让人们很早就开始设想具有支付功能的移动设备。20世纪60年代，科幻作家赫内·巴赫札维勒(René Barjavel)在他的小说《冰人》(La Nuit des temps)中，就曾描述过一种使用“戒指”的支付系统。从1968年到1972年，在日本、德国、英国和美国都有人注册了专利，描述的都是带有电子存储器的塑料卡片。

带有储存器的IC卡的发明者常被认为是法国人罗兰·莫雷诺(Roland Moreno)。1974年，在让·穆兰(Jean Moulin)的帮助下，他为自己设计的储存器卡片申请了专利。但是布尔公司的法国工程师米歇尔·于贡(圆图中的人)才是真正的“智能”IC卡之父。

于贡在1977年发明了集成了微处理器的卡片。微处理器对于保障卡片安全性所需的计算能力来说是必不可少的。另外，多年来，于贡为整个法国IC卡领域的安全化付出最多。

1977年和1978年间，于贡为微处理器卡申请了两项重要专利。1979年3月，他造出了第一张(双芯片)微处理器卡；1981年10月，他又造出了单芯片微处理器卡，也就是布尔CP8卡，这种卡片的安全性更高。

接着第一批用于电话卡或银行卡的工业订单就出现了。虽然当时研究者已经考虑了加密技术，但还并未达到比较理想的水平。

不过，在20世纪70年代出现了一个著名的特例。在卡尔·迈耶(Carl Meyer)领导下的IBM公司团队很早就开始围绕着密码学的应用进行研究。他们当时设计出了一款实验性的卡片，与现在的IC卡很不一样，用的是DES算法的前身Lucifer加密算法。在伦敦的一次数据传输试验中，他们对这种卡片进行了测试。

但是IBM并没有沿着这条道路继续走下去，因为它的银行赞助商并不信赖这些并不完备的卡片技术。于是，IC卡的初步摸索就这样夭折了。

然易如反掌。因此，密码的传输不能通过直接的方式进行，而是要用迂回的方式，不提供密码。

实际操作中采用的方法是，用某种加密算法来配合随机数或者密码。比如，不传输密码，而是传送一个加密算法的函数，这样就能让使用者在不透露密码的情况下证明自己知道密码；反过来，这种方法也能保证另一方在不知道密码的情况下验证密码是否正确。

因此，IC卡中有必要使用强加密技术的理念在1980到1984年间迅速得到了认同。法国布尔公司(Bull)的米歇尔·于贡(Michel Ugon，Telepass算法的发明人)、法国电视广播及通信研究联合中心的路易·纪佑(Louis Guillou，加密字节发生器GOC算法的发明人)，以及当时任

职于飞利浦公司的本文作者齐斯夸特在这方面做出了巨大贡献。1976年，当美国斯坦福大学的惠特菲尔德·迪菲(Whitfield Diffie)和马丁·赫尔曼(Martin Hellman)的文章《密码学的新方向》(New directions in cryptography)刚刚发表时，我(齐斯夸特)曾有幸拜读。因为在密码学领域的贡献，这两个作者在40年后的2015年获得了图灵奖。

那个年代还只有对称密钥系统，这两位密码学家在这篇文章中提出了公开密钥系统的概念，并证明了大质数计算在这个系统中的价值。我当时是飞利浦在比利时布鲁塞尔的实验室的研究员，致力于应用数学研究。看到这篇论文后我着了迷，向上司申请进行加密数学的研究。

我的申请得到了批准，在第一个公开密钥系统——RSA加密算法出现前，我和其他6人就已开始探索这个领域。RSA加密算法是以3个发明人的首字母命名的，他们分别是罗纳德·李维斯特(Ronald Rivest)、阿迪·萨莫尔(Adi Shamir)和伦纳德·阿德曼(Lenard Adleman)。时至今日，RSA加密算法仍在被广泛使用。马丁·加德纳(Martin Gardner，美国著名业余数学大师，曾在《科学美国人》上开设了延续20多年的数学游戏专栏)曾于1977年在《科学美国人》上介绍了这个算法。

1979年12月，飞利浦法国公司和我们联络，问我们是否愿意参加法国经济利益集团存储卡项目(Groupement d'intérêt économique Carte à mémoire)的招标。这个项目背后是法国电信总局以及一些法国银行组成的财团。这项招标旨在测试在现实生活中，IC卡在技术以及经济方面的可行性。飞利浦公司的另外两个竞争对手斯伦贝谢公司(Schlumberger)和布尔公司也参与了竞

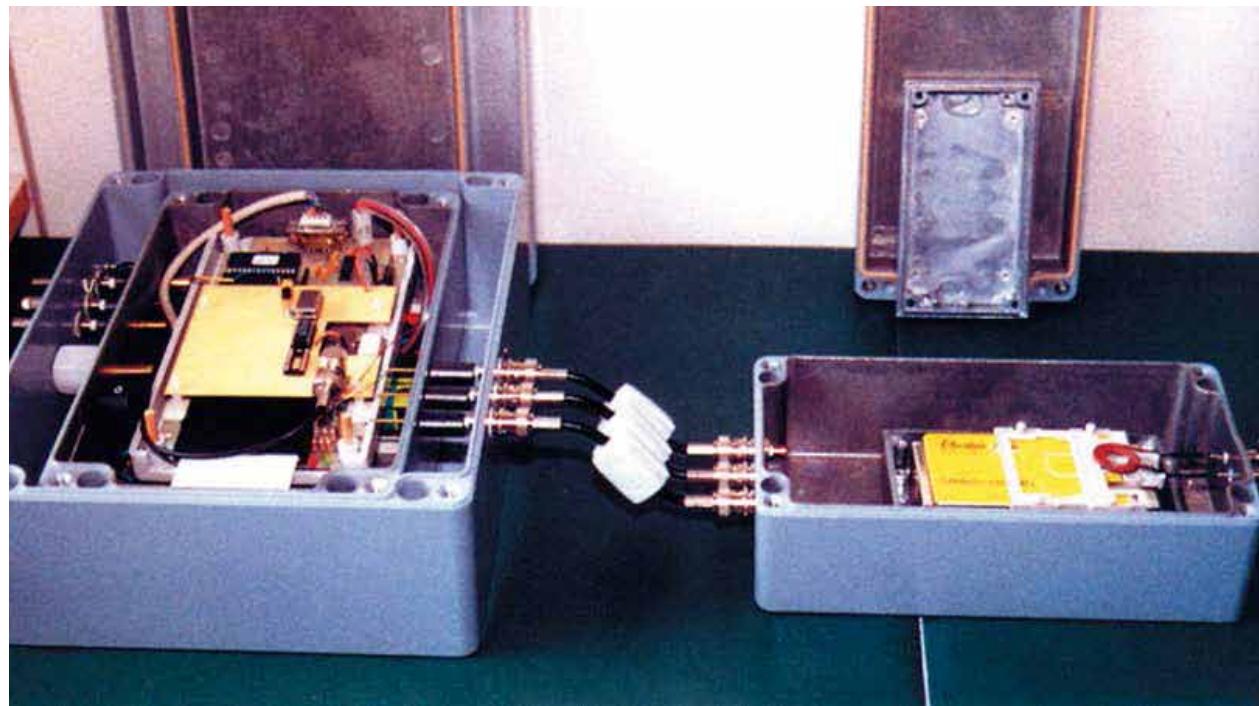
标。

负责这个计划的团队认为，应该在IC卡中植入加密技术，至少在身份验证时应该加密。从这次竞标起，我们就在飞利浦公司成立了致力于密码学研究的实验室。

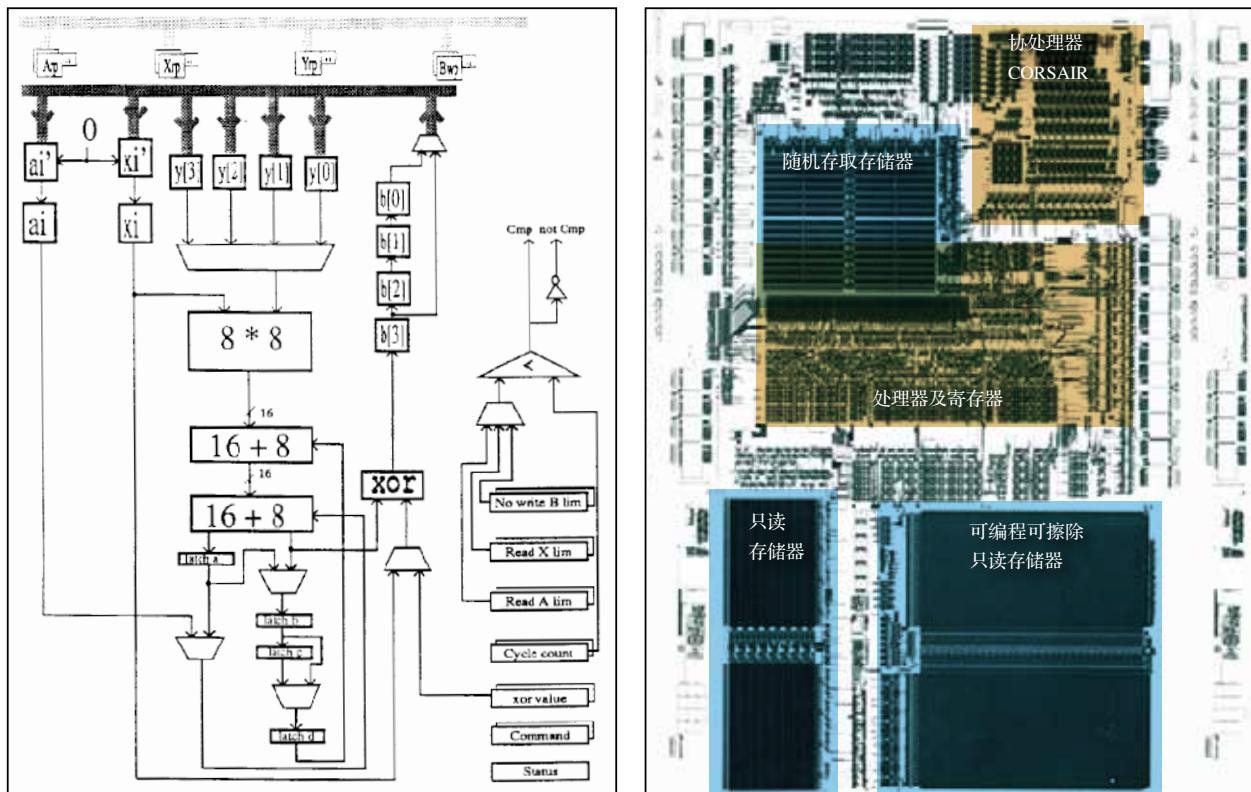
1981年，经济利益集团存储卡项目开展了对IC卡的3次大规模试验。第一次是布尔公司在法国布卢瓦进行的，第二次是飞利浦公司在卡昂进行的，第三次是斯伦贝谢有限公司在里昂进行的。每次试验都用上了几百台终端和几万张IC卡。在准备进行试验的时候，我们开了几次会，为IC卡增添了加密功能和安全协议。

只有在布卢瓦进行的试验取得了令人满意的結果，于是布尔公司赢得了招标，后来为这个项目开发了IC卡。但是，当时的IC卡还远不够安全。原因很简单：要给IC卡增加加密协议的话，加密算法，尤其是公开密钥算法对于当时卡片的容量来说太大了。

而法国计算机安全服务中心的要求是非常高



只要用这台和计算机相连的平平无奇的IC卡分析设备，我们就能够方便地测量芯片计算用时以及耗电量。利用这两类数据我们就能破解卡片上RSA加密系统的密钥。



第一个 RSA 芯片叫做 CORSAIR (右图)。除了不同的储存器 (ROM、RAM、E2PROM) 以及一个处理器以外，它还包含一个能够增强大数乘法计算的协处理器。RSA 算法就是建筑在协处理器之上。左边这张图表现的就是协处理器的算法。

在图中我们可以看到一个8位 (一个字节) 乘法器，以及两个多字节加法器。剩下的部分包括选择器、指针和寄存器链。它们能够管理所有的参数和要处理的数据地址。这是世界上最小的协处理器。在一个时钟周期内，该协处理器就能够实现 $x * y + a + b$ 的运算，其中 x 、 y 、 a 、 b 均为一个字节。

这个运算十分神奇，因为其中每个元素都可以是一个0到9的十进制数。用这个算式得出的最大值将是 $9 * 9 + 9 + 9 = 99$ ，这个二位数无需向左再进位。这个算式已经足以应付所有的大整数乘法，以及带余数的除法计算。

的：要防止不法分子挪用使用了强加密技术的物品。要用IC卡存储密钥或提供安全身份认证的话，必须满足这些条件。换句话说，IC卡的加密算法必须强到无法通过加密过程反向推导出解密过程。

加密算法之争

一开始，IC卡带有2个芯片，其中一个是处理器，另一个是储存器，这样就可以为算法提供更大的存储空间。但是我们觉察到，两个芯片间的连接是透明的，因此窃听处理器与存储器之间的数据交换是可能的（例如监测电流强度变化）。单芯片解决了这个问题，但是并没有解决容量的问题。

第一批IC卡的加密算法要么是单向算法，要么是可逆算法，但都只用一个密钥（比如Telepass

1卡、TDF卡、Telepass2卡、Videopass卡等）。对于当时IC卡的8位处理器，以及有限的计算速度和储存空间，这些算法很合适。完整安装的情况下这些算法大概只占用200个到300个字节，非常了不起。但是，由于这些算法并没有公开发表，因此它们在法国之外的使用和传播都受到了非常大的限制。

使得这些算法变得广为人知的是1984年欧洲密码学大会，这是关于密码学的第一次公开国际会议。纪佑在该会议上展示了IC卡以及条件接收系统。有人问他为什么他不用DES算法。DES算法是一种密钥长度为56位的加密算法，在1976年成为行业标准。他回答：“这是不可能的，因为ROM（随机存取存储器）太小了。”

这是我和我的团队当时准备着手应对的挑战。实际上在比利时当时正有一个名为TRASEC的银行间安全传输项目，这个项目使用IC卡来储存交易密钥，用户可以用自己的PIN码来使用IC卡进行交易。但是，密钥使用的DES算法要在一台没有保护的计算机上运行。如果将算法也植入IC卡中就能填补这个安全漏洞。在几个月里，我们一个一个字节地，将DES算法里的置换操作提取出来，并将其压缩。我们通过这种方式把安装算法占用的空间减少了总共668个字节：每个字节大约花了每个人1天的工作时间才完成。

芯片上的RSA加密演算法

但是，对于我们每个人来说，IC卡加密技术的圣杯并非DES算法，而是RSA算法。RSA算法的基础是大整数的因数分解，这种算法已经成为密码学的标准。在卡片中集成这个算法就能进行生成数字签名和密钥分配等操作了。英国牛津大学的保罗·巴雷特（Paul Barrett）于1986年为发明了一种新形式的RSA算法，该算法是专门为信号处理器设计的。

我们从巴雷特算法出发进行研究。在对IC卡进行了几次测试之后，我们终于实现在100秒之内执行512位的RSA算法。借助于一些技巧（对公共模数的存储，以及在求模运算中使用中国剩余定理——用于求解同余方程组的定理），我们就能用512位RSA模块在20秒内生成一个数字签名。这个结果已经非常漂亮，但是对于实际应用来说还是太慢，而且消耗过多计算资源。

1989年，飞利浦法国和德国公司的亨利·莫尔科（Henri Molko）向我们提出了一个大胆的建议：在芯片中加入一个RSA协处理器，与80C51处理器联合起来，可以在1秒内完成512位RSA数字签名。实际上后来我们仅用了0.2秒。在没有经费支持的情况下，我们参与到了这个名为 CORSAIR（Coprocessor RSA In a Rus）的项目中。

在5周内，我们用我在不久前发现的一种新算法设计出了几个协处理器的版本。这是对IC卡

RSA算法的首次提速，提速后的运算速度达到之前的500倍。1990年初我们制作出了第一个原型硬件芯片，一切运转正常，唯一的缺憾就是在我们项目的后期，对卡片的可编程可擦除只读存储器（E²PROM）的写入速度“优化过度”了。

我后来另外设计出了一个叫做齐斯夸特算法的新算法。它与其他算法的不同之处在于，协处理器的从头到尾的执行时间并不取决于密钥的长度，而是固定的，因为在算法执行期间没有进行任何其他校验。这种算法能够防范通过测量算法执行时间来推测密钥的攻击行为。

芯片制造商间展开了一场第二代协处理器的竞赛。Fortress、西门子、英飞凌（Infineon）和飞利浦等公司都参与到了混战之中。1995年，飞利浦为了这场竞赛启动了一个新项目：FAME。这个项目的成果就是，我们制造出了速度比CORSAIR协处理器快500倍的新一代协处理器。它有何秘诀呢？改良了电路，增加了能将频率提高16倍的乘法器（该乘法器能够计算32位，而不是8位的编码数字）、带宽更大的总线（bus，信息传送线路）、端口宽度更大的双端口RAM，协处理器的处理链也改善了。时至今日，该协处理器的几种优化版本仍在被使用。和10年前的第一代8位软件版本相比，FAME 协处理器的速度提高了250 000倍。

1989 年，飞利浦法国和德国公司的亨利·莫尔科向我们提出了一个大胆的建议：在芯片中加入一个 RSA 协处理器，与 80C51 处理器联合起来，可以在 1 秒内完成 512 位 RSA 数字签名。实际上后来我们仅用了 0.2 秒。



IC卡是武器？

在法国，IC卡安全化还面临的一个法律问题。因为按照1939年的法令，密码技术被归为战争武器中的一类。

因此在1985年左右召开的法国密码学部门会议上，法国工业部的代表在进入马蒂尼翁宫会场时，为携带第二类武器——银行卡进入该会场而表示歉意。

当时的法律是不允许他拥有这张银行卡中使用的加密算法的。后来上述法令得到了修订，对使用该技术的人并不加以处罚，那时加密算法的潜力已经开始慢慢凸显出来。

加密算法到底是不是战争武器？当IC卡被用于通信广播时（如付费电视或移动电话），这个问题又重新浮上水面。

IC卡应用的扩张适逢一些国家正要寻找评估计算机系统安全性的方法。美国最先推出了橙皮书（Orange Book），向北约提出了计算机安全性要达到一定水平所应遵照的标准。

很快各国发现，美国的提议实际是为了大力推销美国的产品，与他国竞争。作为回应，欧洲的一些国家各自提出了不同的议案，德国提出了绿皮书，英国提出了黄皮书，法国提出了蓝白红皮书。最终，欧盟控制住了局面，催生出了欧洲的安全评价标准，即ITSEC。这套标准让欧洲列国能够执行他们自己的认证计划。要注意的是，ITSEC与橙皮书不同，橙皮书确认的是安全机制是否存在，而ITSEC则会确认安全机制的有效性以及效率。

不久后，在大西洋两岸，以及世界上的其他地方就出现了一套各国都能接受的、一般性的安全标准，这就是信息安全通用标准（Common Criteria, CC）。不久，CC被转化为国际标准化组织（ISO）的标准。法国作为工作组主席，在该标准的编写过程中承担了决定性的工作。

实际上，当美国和英国主要关注不同操作系统以及类似于防火墙一类的安全工具的认证时，法国把精力投入到了IC卡领域。法国在这个领域的专长使其能够形成一定的垄断之势。

于是，那些著名的韩国、日本和美国公司（如摩托罗拉）纷纷委托法国对其产品进行评估。而法国在该领域的傲人能力，实际上源自法国电信（France Télécom）一个人数不多的研究团队。这个团队大约有十几个成员，他们与SCSSI合作，研发硬件和软件以及自动搜索漏洞的检测方法。这个团队后来被转移到法国原子能和替代能源委员会（CEA）旗下的信息技术实验室（Laboratoire des technologies de l'information, LETI）中。

保障IC卡组件的安全性需要生产企业付出大量努力，有时甚至需要对生产线进行大改造（如洁净室）。这些工作耗费的投资是巨大的。

1999年，Moneo卡（这个电子钱包卡片没有得到预想的成功）在法国图尔正式发布，这家公司的首席执行官在现场正式接受了自己的安全证书，他在随后的演讲中称，这个安全证书让软件和开支都额外增大了20%，使产品超时30%才交付，并且让研发人员的汗水和压力也增多了50%。但是他对此后研发速度提升，以及最终的产品非常满意。

长久以来，加密技术保护下的IC卡一直被认为是最安全的支付方式。公开数据显示这种卡片的盗用率为0.03%。因此，该行业缺乏进一步加强安全措施的动力。尽管密钥的长度被专家认为太弱，但还是被原样保持了太长时间。因此，当GIE银行卡遭遇了亨佩琦事件，原本的乐观迅速演变成了恐慌。

亨佩琦事件

1997年，自学成才的芯片工程师塞尔日·亨佩琦（Serge Humpich）成功地破解了RSA数字签名。当时该数字签名是静态的，和加密过程

一样都储存在卡片中。后来他在巴黎大众运输公司(RATP)的yescard卡终端机上成功地使用了假卡，这些终端机对于小额交易不要求强认证。他试图靠这个技术赚上一笔，但因触犯了高德佛兰(Godfrain)法而被逮捕。2000年对他进行的审判为人们敲响了警钟，因为亨佩琦和类似的黑客是有能力通过传播RSA密钥而危害系统安全的。

实际上，纯靠技术手段盗用IC卡的案件是相当少的。多数案件主要归咎于受害者令人难以置信的天真，以及诈骗犯在社会工程学(指利用人的心理弱点进行欺诈和控制的手段)方面的高超技能。但是，当卡片具备了“非接触”功能后，IC卡真正的弱点才暴露出来。

已经有人证明，有办法在距离较近时黑入非接触IC卡，从而窃取高度敏感信息。法国信息与自由委员会(Commission nationale de l'informatique et des libertés, CNIL)得知后大为震动，并命令法国的银行在卡片持有者请求的情况下关闭这项功能，或者提供一个“法拉第笼”，以避免不必要的电磁干扰。

然而，还有更值得担忧的地方。不管IC卡片是接触的或非接触的，它本身也是一种安全设备。而现在网络和电信业巨头们打算用新的产品，把它替换掉。

例如，使用带有新功能的SIM卡，或是本身就具有这些新功能的手机。在这类手机上，通信工具以及身份识别和认证方法已融为一体。黑客对这样的趋势肯定是喜闻乐见的。这也为谷歌、苹果、Facebook和亚马逊等网络巨头提供了控制尚未被美国公司完全占领的领域的机会。

参考文献

1. *Carte à puce et cryptographie :jet' aime moi non plus, conférence de J.-L. Desvignes et J.-J. Quisquater au Musée des arts et métiers du Cnam, 2015* :<http://bit.ly/29mT1Lk>
2. S. Bouzefrane et P. Paradinas, *Les Cartes à puce*,

Hermès Science Publications, 2013.

3. C. Tavernier, *Les Cartes à puce. Théorie et mise en oeuvre*, 2e édition, Dunod, 2011.
4. W. Rankl et W. Effing, *Smart Card Handbook*, 4e édition, Wiley, 2010.
5. M. Éluard et S. Lelievre, *YesCard, entre mythe et réalité ou un aperçu du système bancaire français*, MISC, hors-série n°002, novembre 2008.

Jean-Jacques Quisquater 是比利时天主教鲁汶大学密码学荣誉退休教授。

Jean-Louis Desvignes, 法国退役上将、法国数字和信息安全预备役军人协会(ARCSI)主席。

一个数据模型与317种病痛

撰文 W. Wayt Gibbs

在克里斯托弗·默里(Christopher Murray)10岁那年,他和家人一起收拾东西,离开明尼苏达黄金谷的家,踏上了去英国的旅途。他们只带着几个旅行箱还有一台便携式发电机。后来,他们乘坐车辆和渡轮又从英国来到西班牙、摩洛哥,最后穿过撒哈拉沙漠抵达了位于尼日尔偏远地区的迪法村。第2年,父亲(医生)和母亲(微生物学家)带领整个5口之家,在当地建立和运行了一所医院。小默里经常去药房跑腿,在这期间他注意到尼日尔的居民总会染上各种怪病,而明尼苏达的人则不会。他觉得很奇怪,“为什么有些地方的人会比其他地方的人更容易染病,情况也更严重?”

默里变得很沮丧,他想和家人一起帮助迪法的居民。但在一年后,默里意识到,“当我们离开时,这里的情况还是不会变好”。默里说,“那么问题来了:怎么做,才能使这里的情况发生根本性的改变呢?”

在接下来的40年里,这个问题一直促使默里着手推动一系列的项目,他希望医生和卫生部官员们更加注重大局和长期趋势,从而关注为什么这么多人死于本来可以预防的疾病。他把大部分精力都投入到了建立一个优质的系统中,一旦成功,这个系统就能为全球卫生的决策提供真实可靠的信息,在此前,相关数据十分缺乏。

默里意识到,让我们生病的原因往往描述得很模糊,因为政客会隐瞒或回避这些让人尴尬的数字。比较一个又一个国家的统计数据也可能非常棘手,现在,他们终于找到解决这些问题的办法了。

在很多人的帮助下,默里开始研发一种新的工具,用来消除数据间的差异,反映世界卫生的真实状况。这套工具中还会提供一些建议,比如,需要

做些什么才能减少后代的苦难。

从19世纪开始,一直到20世纪,显微镜的广泛使用掀起了一场关于人类健康的革命,它让科学家把目光聚焦在那些细小的却又能影响人们健康的微生物上。当时科学家们发现了细菌,因此,他们提议改善卫生,加快寻找合适的抗生素,找出可以预防疾病的疫苗。可以说,默里的新工具更像“显微镜”的反义词,它从最宏观的角度阐述有用细节,将国家,大陆甚至整个人类看作一个整体,现在我们可以把它称为“宏观视野”。

新的契机

1993年,默里在世界银行(the World Bank)工作,他和几位同事共同编著了对全球人类疾病具有里程碑意义的报告,当时是他首次尝试使用类似的工具。2007年,他在西雅图的华盛顿大学创办了健康指标评估研究所(Institute for Health Metrics and Evaluation, IHME),并在全球范围内招募合作者,他想构建一套更加复杂和全面的“宏观视野”。

和许多21世纪的创新发明一样,“宏观视野”也是由软件和大数据组成的。一端是从世界各个角落收集的数据,这些数以千兆的数据会由来自世界各地的1000多名科学家审验核查。另一端是比以往任何时候都要详细和精确的交互式图形和表格,它们几乎涵盖世界范围内各种生病,受伤或死亡的情形(从心脏病到被驴咬都有)。中间则是运行在超级计算机上的代码,这些程序会运用巧妙的统计方法校正误差,识别和剔除异常数据,对全球许多缺乏合格数据的地方进行智能评估。

经过10年上千万美元的投入和开发,这套系统已经能够做出和每年人体健康检查类似的推断



癌症——这也是发展中国家的苦恼

通常，癌症被认为是富裕国家的问题，但宏观计算得出的结果表明，这种观点是错误的。在墨西哥和中美洲，随着生活水平的提高和妇女寿命的延长，乳腺癌在健康支出中的比例大幅增加。例如，自1990年以来，墨西哥妇女的死亡率已经飙升了75%——校正人口老龄化的影响后，比例为17%。相邻国家也有类似的增长。

从更广阔的角度来看，在中等收入和低收入国家中，因各种癌症致死的人数已占总死亡数的一半左右，超过了艾滋病、结核病和疟疾致死人数的总和。但国际社会在预防癌症方面花费仍然相对较少。比如，对癌症相关的人乳头状瘤病毒疫苗（HPV）或癌症风险因素（吸烟、不良的饮食和缺乏运动等）的教育还不够。在更富裕的国家中，这些减少癌症发病率的措施都实施地较好。

什么是 DALY？

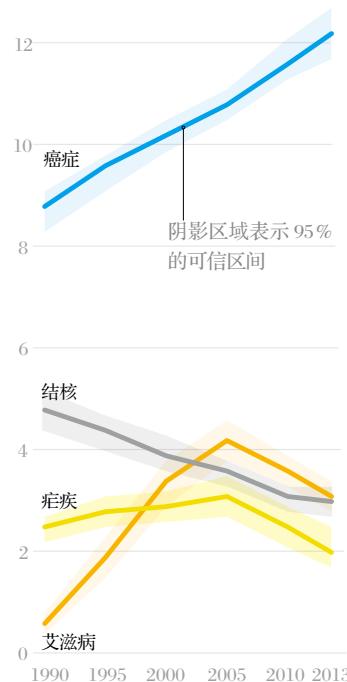
默里说，“良好的健康不仅仅是为了延缓死亡。”

为了更有效地量化非健康状况，默里和阿兰·洛佩斯长期使用DALY指标，它也被称为健康生命年损失。

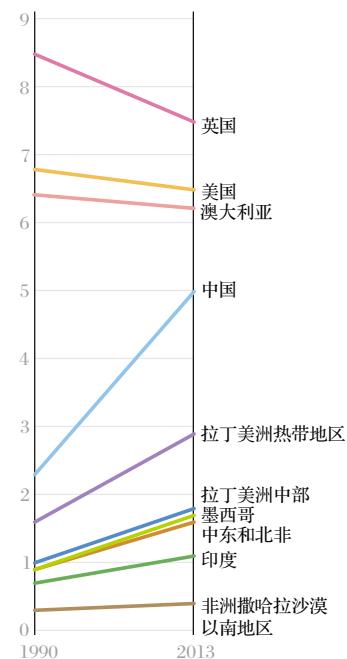
一般认为，一个DALY相当于损失一年的健康生活。要计算它，必须先假设每个人，无论贫富老少，都能达到最大的预期寿命（最新的研究数据是86岁），而且没有患病。如果一位5岁的孩子因患疟疾突然死亡，就造成了81个DALY损失，如果一位85岁的老人在之前一直健康地活着，但过马路时因事故死亡，就造成了1个DALY损失。伤残虽然没有实质性地减少寿命，但是会恶化生活质量。

因此也需要转化成DALY。以前，加权方法主要参考有医生参与的公共调查，而现在，各行各业的人都开始加入其中了。来自不同领域的专业人士会对不同的病痛做出评估，比如，失明带来的失能或前列腺癌造成的痛苦。

A 发展国家的死亡情况
(占总人数的比例)



B 由乳腺癌、肺癌和结直肠癌损失的寿命数
(占伤残期等效寿命的比例)



了。在2012年和2014年，IHME分别发布了2版统计数据集。2015年收集更新的数据集经过年度审核后，也将在今年9月发布。随着数据不断更新，这套涵盖了历史信息和最新信息的系统将变得更加准确、详细和完整。报告也统称为全球疾病负担、伤害和风险因素研究（简称为GBD）。

不管在反映富裕或贫穷国家的空气污染、肾脏疾病和乳腺癌等疾病的发展趋势时，报告都引起了人们的关注。不过，该报告的一些调查结果也引起了争议，因为有时报告的结果会与世界卫生组织（WHO）和联合国（U.N.）的其他分支机构发布的数据相矛盾。

和许多发展中国家的人们一样，卢旺达卫生部长阿格尼丝 (Agnes Binagwaho) 也是“宏观视野”的粉丝之一。在2013年的官方发言中，她说：“这可不是一个简单的工具，而是一场革命。我们将有一门科学家的通用语言，一门有关卫生政策的通用语言，我们的人民将拥有更美好的生活。”

启动项目

开发“宏观视野”的动力无疑来自20世纪90年代初的迪安·雅米松 (Dean Jamison)。他是一名经济学家，在世界银行工作时，汇集整理了全球范围内关于健康和经济相互作用的深度报告。当时，雅米松曾委托默里调查1990年时由于疾病和伤害造成的经济损失。之后，在世界卫生组织工作的流行病学家阿兰·洛佩斯 (Alan Lopez) 也加入团队，和默里一起工作。

“我们大部分的工作是在缅甸克里斯 (Chris) 的谷仓中完成的，”洛佩斯说。他现在在澳大利亚的墨尔本大学工作，从事跟GBD相关的项目。“大家疯狂地搜索所有能够查到的数据，大概找到了关于120种疾病和10种危险因素的资料。我们完全陷入其中，甚至每天工作20个小时。”

最新的GBD将覆盖317种致死和致残的原因，平均每个国家有50 000多人贡献数据。与它相比，1993年问世的世界银行报告显得非常基础。尽管如此，这份报告依然有很大的影响力。正是因为它，微软联合创始人比尔·盖茨 (Bill Gates) 和妻子梅琳达·盖茨 (Melinda Gates) 才决定把大部分慈善基金（迄今为止超过370亿美元）用于遏制传染病。1998年，WHO干事格罗·哈莱姆·布伦特兰 (Gro Harlem Brundtland) 还曾聘请默里为WHO建立类似的系统。

当时，默里和他的同事们试图对各个国家的卫生保健体系表现排名。一些排名靠后的国家对此提出了抗议，他们中很多本身就是U.N.的成员。

“它变成了一场特殊的政治斗争，”默里说。回应政治风波总会让人分心，而整个评估项目也因此失败。2003年他离开WHO回到了哈佛大学。默里回

忆说，“最大的教训是，应该学会如何规避政治影响。”

为了避免政治干预，他开始寻求私人资金建立学术机构。在去西雅图的旅途中，默里与比尔·盖茨见了面，盖茨立马同意了默里的提议。2007年，为了启动IHME项目，比尔及梅琳达·盖茨基金会捐赠了1.5亿美元，于是，围绕着“宏观视野”的工作迅速铺开了。

克服建模困难

全球范围内的数据都面临缺乏有效统一管理的问题，要想统计人类患病的信息，是一件极其困难的事。默里发现，有很多研究者像他和盖茨一样，还对信息统计怀有信心。因为，虽然来自政府、WHO和其他联合国机构或的数据统计存在缺陷，但这些问题都可以被修正。从长远来看，优质的数据可以拯救更多的生命。他们需要克服的只是：有问题的测量方式、系统偏差和数据缺失。

来自卫生部门、救援组织和科学文献的原始数据都会被收录到系统中，但首先，他们需要严格控制数据的质量。“每得到一个新的数据集，我们首先会问，‘它存在什么问题？’”洛佩斯说，“我们用一些非常智能的程序处理数据：不合格的数据就会被排除在外。此外，我们还会用科学的方法将处理后的数据重新分配到已经定义好的数据列表中。”

这个过程有助于控制异常值。比如法国人的脂肪和胆固醇摄入高，但他们心脏病的死亡率看上去却不高。事实证明，当人们死于心脏病时，法国的医生倾向于记录其他的协同因素。IHME的首席科学家特奥·沃斯 (Theo Vos) 说：“不同的文化习惯造成了近半数类似的‘法国悖论’。”

全球有数百名科学家参与了工作，他们擅长各种不同的疾病，对数据里疾病定义之间的差异进行了校正。在统一数据的标准后，洛佩斯说：“我们可以对比匈牙利和萨尔瓦多的癌症发病情况，也可以对比南非的情况，甚至可以对比任何有癌症发生的地方。”

慷慨的资助者

与比尔·盖茨对谈

一直以来，比尔·盖茨和梅琳达·盖茨都是全球许多卫生行动的主要资助者之一。其中包括努力减少艾滋病和结核病的行动，以及根除脊髓灰质炎和疟疾的行动等。比尔及梅琳达·盖茨基金会同时也资助创建了健康指标评估研究所(IHME)，好让研究所对各种疾病的发病率和原因进行评估。这个项目就是我们熟知的全球疾病负担、伤害和风险因素研究(简称GBD)。在2014年接受《科学美国人》编辑W·韦特·吉布斯(W.Wayt Gibbs)的采访中，比尔·盖茨回顾了为什么会支持这项工作，到目前为止这个项目又取得了哪些进展。采访摘要如下。

《科学美国人》：您是默里的早期支持者，同时还资助了他建立的独立组织IHME。这个组织很大程度上是独立于WHO的，它谨慎地汇总了全世界大量和健康相关数据。当时你们两人是如何见面的，为什么会想到开展这个计划？

比尔·盖茨：在2001年时我遇到了默里，当时他在WHO工作，第一次尝试为各国的卫生系统排名。但是有些国家不愿接受排名，因为他们不喜欢默里做的事。当时他们想把各种数据汇总，试图以更好的方式理解全球(尤其是贫困地区)的卫生状况，我们认为这个想法很有吸引力，所以就把钱给了华盛顿大学，建立IHME。

《科学美国人》：但WHO和U.N.的其他机构也会收集和发布世界各国的卫生统计数据。为什么还需要有一个完全独立的组织去做这个事？

比尔·盖茨：我热爱WHO，WHO的总干事陈冯富珍已经做了很多伟大的事情。但它是一个联合国机构，有一定的复杂性。当默里在WHO里试图给各国的卫生系统排名时，他发现外界的资金限制了他们站在有争议的立场上，他们对此无能为力，因为，要给自己的资助者评分是一件十分困难的事。

《科学美国人》：目前收集到的人类健康数据质量都非常高，您对此有什么印象？我们比20年前更健康了吗？

比尔·盖茨：让人振奋的好消息是，现在我们比过去健康了不少。可能这是一件没有什么争议的好事，所以也没得到足够的注意。不过，如果你看到像越南、柬埔寨、斯里兰卡、卢旺达和加纳等国家的情况，就会发现基本上所有的传染病都在减少。现在唯一一种发病率还在上升的传染病是登革热，我们正在想办法克服它。目前发展中国家的非传染性疾病的负担正在加重。糖尿病正在流行，我们的医疗费用也在激升。还好，在过去的20年中，全球范围内的健康状况仍然发生了巨大的转变。

除此之外还有政治影响力。“政府向WHO和U.N.提供数据变得越来越困难，”默里说。比如，联合国艾滋病规划署(UNAIDS)每年都会发布预估的艾滋病毒/艾滋病(HIV/AIDS)流行率，但一些国家并不认同这些报告。所以相关机构也就不再公布这些国家的数字。截至2015年，已有83个国家和地区不在数据公布范围内了。

数据缺失是最棘手的。世界上很多疾病肆虐的地方并没有做过全面的统计。GBD的团队也在着手补足这些缺陷。

首先，他们安排了一拨人在类似地区挨家挨户做调查，从手写记录中收集信息，必要时，还会对家庭中的患病和死亡情况做深入访谈。在印度实施的“百万死亡研究计划”(The Million Death

Study)就大规模地采用了这种方法。2010年，初步结果表明，疟疾致死的人数是之前WHO报道的10倍以上，因为WHO的主要资料来自医院病历，漏算了很多死在家中的病例。

第二种方式是根据前人的研究，总结各种疾病、伤害和风险因素的一般模式，以此推断，填补空缺数据。例如，在雨季或雨季刚结束时，疟疾发病更严重；在老龄化国家中，癌症发病率更高；如果一个国家HIV阳性的比例较高，邻近国家的艾滋病也更流行。通过这种相关性，可以利用附近地区的规范的统计指标，合理估计其他数据不完整地区的情况。

“我们有200个这样的数据库，库里的数据类型非常丰富，从纬度、人口密度和降雨量到抽了

多少烟，消耗了多少猪都有（变量都经过了仔细修正），“沃斯说道。系统会生成无数的变量组合，将它们嵌入到各种形式的模型中，看哪些疾病能和哪些组合配对，做出一致性最高的预测。

这种方法称为集成建模，已经广泛地应用在天气、金融和保险等领域中了。沃斯认为，许多流行病学家忽略这种方法，可能是因为它需要强大的计算能力。

近年来，GBD对188个国家的1000多项卫生指标进行了跟踪调查，时间至少向前追溯了25年。这也是对20到40个统计模型的复查。他们还将每个模型都运行了1000次，从而获取波动范围内所有的可信度。

这种技术会结合预估值和（表示不确定性的）误差值，综合做出判断。运行这些算法的IHME超级计算机有12 000颗连续运转的高性能内核，能在4天内算出一张地球快照（一张关于人类健康的成绩单）。

“事实上，无论是想发表这些数据，还是想规范数据的误差范围，都能让更多人参与讨论这个项目，也能让它成为人们关注的焦点，”盖茨说，“现在IHME创建了一个中央资料库，因此人们不用阅读数百篇文章，再自己整理出一个领域的宏观情况了。他们可以把精力更多地放在对某些具体数据的讨论上，如果方法运用得当，就能进一步减小误差，达成研究目标，使知识变得越来越完善。”

与权威数据的矛盾

2012年，GBD发布的第一版数据引发了不小的波澜，甚至一些对本国医疗信息系统颇为自豪的国家也受到了刺激。例如，当英国官员看到英国人的健康水平已经落后于欧洲邻国时，他们感到十分惊慌。默里说：“在看过风险因素分析后，他们调整了策略，更加重视饮食方面的问题。”包括中国、巴西、德国和俄罗斯在内，至少33个国家都开始尝试对本国人群进行类似的研究，从而改善卫生统计数据的质量和细节。同期，“宏观视野”也会收录这些数据。

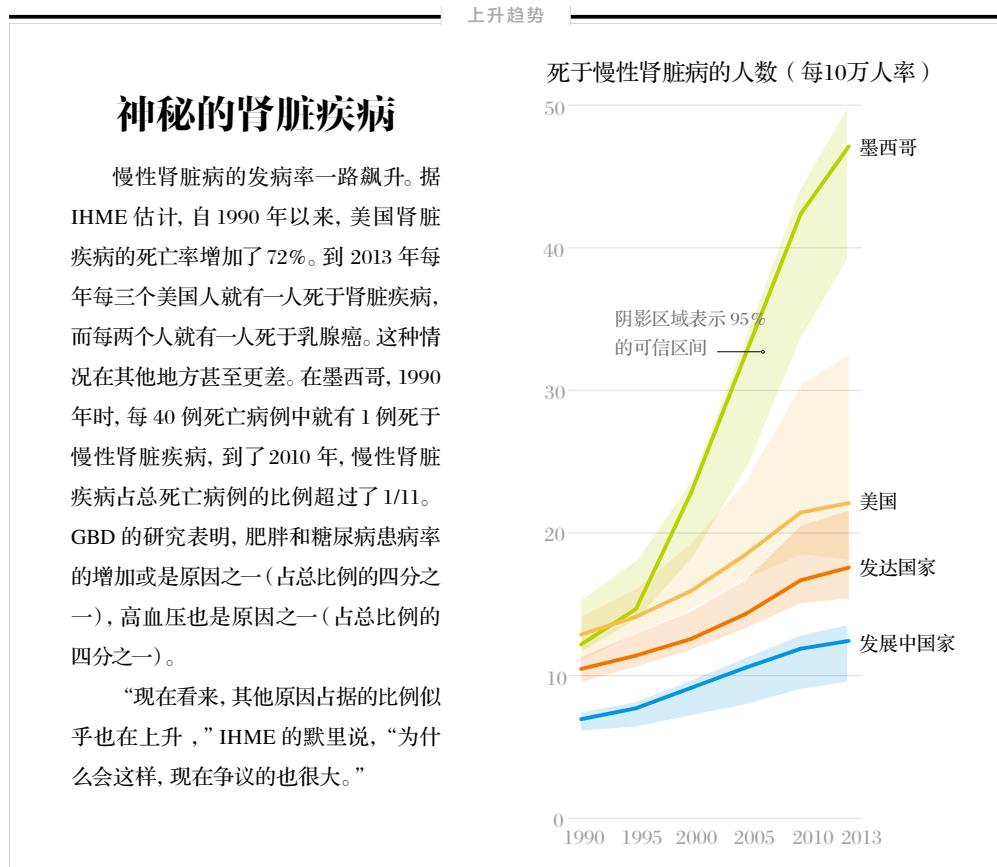
GBD系统的新发现还引发了争议，因为里面有些数据与一些国际机构的权威数据相矛盾。例如，2014年GBD发表的艾滋病病毒感染率分析表明，UNAIDS高估了2005年到2012年期间艾滋病的感染率，幅度高达17%到19%，这意味着真实数据与纸面数据相差了660万感染人数和63.5万死亡人数。如果GDB的数据是正确的，就意味着当时采取的预防和治疗策略的效果好过预期，就可以考虑当时的方法是否值得大面积推广。

另一个有争议的调查是，根据IHME估计，在2013年约有1/3因疟疾丧生的人是成年人。而长期以来人们一直认为，蚊虫传播的疾病会让各种人患病，但致死的情况大多发生在儿童身上。“百分之九十的人都认为默里错了。”盖茨说。目前这个争议还没有得到解决，只有在未来几年收集更优质的数据，才能决出高低。

GBD公布的数据也有很多令人欣喜的部分。比如，报告称，从1990年到2013年腹泻病（主要由饮用不卫生的水引起）造成的致残和死亡率下降了70%。实际上，与其他风险相比，不卫生的水带来的威胁已经非常小了。因此默里和同事们提出，在某些地区应该更改援助方案，比如更明智的做法是把投入到净水工程的资源投入到预防道路伤害。在过去，很多孩子死于水源性疾病。但现在，越来越多的孩子存活下来，成为行人，骑自行车或开车的青少年。

还有，系统也强调了全球监管体系中存在的盲点。“让我们觉得特别糟糕的是，我们并不清楚哪些地方有伤寒和霍乱。”盖茨说。现在盖茨和其他地方的官员或救援人员见面时，都会随身携带GBD的表格。据IHME估计，由于疾病记录质量参差不齐，每年因伤寒致死的情况会造成大量健康生命年损失年（DALY），至少有600万年，多的时候可达1830万年。百日咳、麻疹、A型和C型肝炎也存在这样的不确定性。

默里始终抱有希望，他相信，“宏观视野”会随着时间变得越来越清晰。领导者之后可以将他们的重点从每种疾病致病或致死的人数转变成年



复一年的趋势。他说, 这就是宏观经济学现在正在做的事。个人不再把注意力放在收入中值、工作或者缴纳国内生产总值(GDP)上; 而是放在变化率上。我们不再问, “我们能不能做得更好?” 默里说, 现在的问题变成, “我们如何才能更快完善相关问题?”

November 28, 2015.

3. Interactive visualizations of the Global Burden of Disease results: www.healthdata.org/results/data-visualizations
4. Six Billion in Africa. Robert Engelman; February 2016.

参考文献

1. Epic Measures: One Doctor, Seven Billion Patients. Jeremy N. Smith. Harper Wave, 2015.
2. Global, Regional, and National Disability-Adjusted Life Years (DALYs) for 306 Diseases and Injuries and Healthy Life Expectancy (HALE) for 188 countries, 1990–2013: Quantifying the Epidemiological Transition. Christopher J. L. Murray et al. in *Lancet*, Vol. 386, pages 2145–2191;

W. Wayt Gibbs 是一名科学撰稿人、编辑, 现居西雅图。他在比尔 & 梅琳达·盖茨基金会(Bill & Melinda Gates Foundation)资助的投资公司——高智公司(Intellectual Ventures)担任执行主编。



《环球科学》旗下微信公众号

创新地标

技术解读 | 投资指南 | 行业分析 | 前瞻报告



扫码关注“创新地标”



关注全球具有商业前景的技术与创新

计算机教育：美国新战略

撰文 Annie Murphy Paul

艾莉 (Allie) 和劳伦 (Lauren) 在加利福尼亚州洛思阿图斯市罗耀拉小学 (Loyola Elementary School) 读6年级，她们在第5期课程开始前2分钟，就蹦跳着跑进了希娜·维迪雅南森 (Sheena Vaidyanathan) 的教室。她们一直在和班上的另外两位小伙伴威廉 (William)、布莱克 (Blake) 比赛——只要比他们到得早，艾莉和劳伦就是最先进入编程教室的学生了。

从小学生开始

这两名女生坐在一排崭新的台式苹果电脑前，立马做起了维迪雅南森布置的作业：找出某一电脑程序中的错误，然后修正错误。这是一种十分重要又很费力的工作，专业工程师的大部分时间都在做这件事。

上课铃响后，学生们拖着脚步陆续走进课堂。身材娇小的维迪雅南森脸上带着微笑，热情友好地欢迎他们。

“V老师，今天我们要做什么呢？”有个男生一边走进教室一边问。维迪雅南森回答说：“今天我们要修复一个程序的漏洞。”

“太棒了！”他大喊一声，奔向一台电脑。普及编程教学的拥护者自然希望这种现象能够遍布全国。在这些拥护者中，除了有渴望雇到新一代工程师和程序员的商业领袖，还有希望美国在全球经济中保持竞争力的政府官员。今年年初，微软创始人之一比尔·盖茨 (Bill Gates) 就发推特说，“在21世纪，每一名学生都应该有机会学习必备技能（编程）。”很多人都和盖茨一样，坚信编程是一项新的认知能力，与阅读、算术一样重要。马克·普林斯基 (Marc Prensky) 是一名作家兼演说家，他创造了“数字土著” (digital native) 这个术语，在2008

年的一篇文章中他写道，“我相信，现在区分一个人有没有文化，就看他会不会编程”。

拥护者们都强调，现在需要很多懂电脑编程的人，从事这个领域也更有可能获得晋升，他们表示，这些机会在未来几年还会持续增加。

奥巴马政府的官员为美国计算机科学教育的未来勾勒了一幅宏伟的蓝图：一是美国学生与别国学生齐头并进——他们指出，早在2014年英国就要求每一名学生都学习编程了；二是在所有学生平等接受编程教育的情况下，有助于减少富有的白人学生与少数族裔和条件困难的学生在成绩上的差距。

然而，要想达成目标并不是一件容易的事。在全美学生中普及编程教育还面临着严峻的后勤挑战，上到缺少合格教师，下到缺乏课程资料。由于美国的学校系统是分散的，要实施全国性改革十分艰难。而且，大多数学校至今都没有更新适应信息时代的守则与规范（比如，修读计算机科学课程通常不是理科生毕业的条件之一）。同时，来自不同民族、不同社会经济背景的学生，具备的计算机能力也相差巨大。调查显示造成以上现象的原因是：学校在向富裕的白人学生讲学时，通常会深入指导，教授有实质性的内容；而面对非白人学生和低收入家庭的学生时，就只会给予非常粗浅的指导。

而且，越来越多的人认为，编程技能只是美国学生准备在应对21世纪的生活和工作时，需要掌握的一小项本领。编写代码是计算机科学的实践形式，但是学生也需要了解理论知识，理解计算机操作的根本原理。普及“计算思维”也要求校内校外双管齐下：在校内则是培训教师、改进课程、将计算机科学融入到其他学科中、更改毕业要求；在校外进行课外活动、搜集课外资源。





计算机科学班：南希·塞在南洛杉矶的奥古斯都 F. 霍金斯高中任教，全部学生都是非裔美国人或拉丁美洲人，75%的学生经济困难。

面积接触计算机科学相关的内容——在学术意义上具有挑战性、创新性和合作性的课程。相反，贫困学生和非白人学生只能使用简陋的设备，接受最基础的计算机教学（类似“剪切复制”操作）。

即使这些贫困学生和非白人学生所在的学校拥有了足够的设备，也存在“有设备，没课程”的问题。用马戈利斯和同事的话说，这里的计算机科学教育不够成熟，有很多缺陷，比如，没法为学生提供“连续性的课程，让他们拥有有效的学习体验”，也没有提供“深入的、明确的指导与支持”。马戈利斯和合作者认为，课程内容不能以偏概全，而要因材施教，但也必须考虑两个群体受到过的教育。他们还说，这两个群体“非常不一样，他们的平台完全不在同一高度，分歧太深太大了，这些人根本就生活在两个不同的世界。”富有的学生理所当然地认为他们能够轻而易举地熟悉计算机操作，而处于劣势的同龄人却完全不敢这样想。马戈利斯告诉我，真正平等地得到计算机科学教育，是“21世纪每一位公民都需要关注的问题”。

发现了这种极其不公的形势后，马戈利斯说：“我们觉得必须做点什么来改变现状。”因此，她和同事设立了一个从幼儿园到高三，再到大学都适

用的项目——“探索计算机科学”（ECS）。这个项目为学习计算机科学的高中生开设了入门课程，还为洛杉矶联合学区（现在已经扩充到将近24个学区了）的教师提供了专业发展的机会。参与项目的教师将学到如何提供严格但具有吸引力的计算机科学教学。目前，加入项目的老师在不断增加，他们还能通过互联网相互合作。而其他组织（包括“黑人女孩学编程”和“黑客社区”在内），则集中精力利用校外时间开展活动，比如设立周末工作坊和暑期训练营。马戈利斯和她的团队希望为处境艰难的学生服务，他们发现经济和后勤问题经常让这些学生无法利用课外机会学习。

南希·塞（Nancy Se）也决定参与其中。2013年，塞参加ECS项目的培训后，成为了ECS的“教师主管”，帮助其他老师学习如何让学生深入参与计算机科学课程。塞在洛杉矶的奥古斯都·F·霍金斯（Augustus F. Hawkins）高中任教，学生都是非裔美国人或者拉丁美洲人，75%的学生都是“经济困难生”。塞清楚地知道，让“全体学生学习编程”面临的挑战正在一天天逼近。“大多数学生家里都没有电脑，”她说，“他们接触互联网的唯一途径就是用来玩游戏、发短信的手机。”



塞的学生接触科技的机会太少，有很多方面都需要顾及到——不仅要学习计算机的各种功能，还要了解人类设计机器并为之编程的历史渊源。在塞看来，最大的问题是，学生们必须改变他们对计算机科学的刻板印象。“对于一些高中学生来说，相对于他们的家庭出身和自我身份认同，用计算机工作，知道如何编码，这些事情是非常陌生，甚至怪异的，”塞说道，“在我的班里，我和学生不仅要专研这门学科，还要学会重塑自我认知，从而应对如潮流般涌来的文化冲击。”在翻看高三学生所写的文章时，塞特意指出其中一篇，在那篇文章中，学生表达了能为自己和孩子获取稳定收入的愿望。

“在学习计算机科学之前，我从来没想过自己会涉入甚至主修这个领域，”这名学生写道，“作为一个想为孩子提供一个舒适环境的单身妈妈，我也认为做一份自己热爱并能提供稳定经济来源的工作非常棒。”另一名学生写道，“经济困难充斥着我的生活，然而，我是不会让有限的资源拖垮自己的。相反，这会激励我通过学习走向成功。我渴望有一天能成为动漫制作人，达到平均年收入50 281美元的标准。我立志结束现在的经济困境，能自己照顾全家”。

的确，推广编程教育常用的一个理由是，缺乏编程经验的年轻人很难踏入一些高收入的行业。但是许多批评家认为，狭隘地将精力投入到学习某项专业技能（例如编程）也不是长久之计。斯坦福大学教育学名誉教授拉里·库班（Larry Cuban）提到了上世纪七八十年代，美国用早期计算机标记语言做的试验。这项试验由麻省理工学院教授西摩·佩珀特（Seymour Papert）领导，他们发现，教导学生学习标记语言的努力付诸东流。库班说，这是因为这项教学没有给学生传授他们真正需要的计算机知识。库班预言，“全体学生学习编程”的运动也会无果而终，并认为，这场运动是商业领袖推动学校开展的，他们清楚地知道：这是一场“纸巾改革”，“用了一两次之后，就会被撕碎扔掉。”

如果编程教学不是最终的答案——或者说不全是——那什么才是答案？珍妮特·M·温（Jeannette M. Wing）认为答案是：计算思维。温是卡内基梅隆大学计算机科学咨询教授，也是微软研究院的全球副总裁。2006年，她在一本杂志上发表的一篇文章迅速成为经典之作。她大胆宣称：“计算思维是每个人的基本技能，不单单属于

计算机科学家。这种思维模式是先理解计算机科学基本概念，然后将它们应用在解决问题、设计系统和理解人类行为等方面。”

“计算思维包括抽象能力、逻辑推理和符号推理能力，以及拿下一个大难题，将它分解为若干小问题的能力，”温说，“不管有没有使用电脑，这些技能每个人都会用到。”

温在文章中写道，“为了更好地阅读、写作以及运算，我们应该让每个小孩都具备计算思维。”艾琳·欧文斯 (Aileen Owens) 是一位教育家，她看到了温的呼声。欧文斯采用温的想法，把计算思维作为课程的重点部分，用在了匹兹堡附近的南费耶特乡学区，她是那里的科技与创新理事。在欧文斯眼中，培养计算思维要尽早开始，通过一系列分层次的、相关的项目进行，逐步深入，不断加深难度。

在南费耶特，从幼儿园到2年级的学生都会用一种模块化的编程工具，学习这些工具背后的原理。利用“喵爪”(Scratch)程序，学生们可以拖放一些指示模块，例如“移动10步”，“等待5秒”，“左转90度”等。用精确的顺序排布这些指示模块，学生就能够使屏幕里的对象按要求做出反馈（一个卡通形象会向前进，再停下，然后左转），于是，他们就能大体了解电脑是如何接收指令的。指令的内容会随着学生年龄的增大而变得复杂。3年级到5年级的学生会给发动机和传感器设计程序，

**用马戈利斯和同事的话说，
这里的计算机科学教育不够成熟，
有很多缺陷，比如，
没法为学生提供“连续性的课程，
让他们拥有有效的学习体验”，
也没有提供“深入的、明确的指
导与支持”。**

搭建可以受程序控制的乐高机器人。6年级到8年级学生会学习计算机辅助设计 (CAD)；学生利用 CAD 软件设计自己的发明，再用3D打印机制造出原型。上7年级时，学生所学内容从基于模块化的编码转变为基于文本的编码，这就和专业程序员一样，使用更为复杂却更灵活的语言来编写代码。

“每一阶段的目标都是巩固计算思维，只有这样，才能在前一个水平的基础上向上发展，”欧文斯解释说，“这不止是编码那么简单。这是一种思维习惯的教育，这种思维习惯可以用来解决任何领域的问题。比如，把一个大问题化解成若干个小部分；进行若干小型实验，从而得知哪些方法会失败，哪些方法会成功；与他人合作，共同想出最好的主意。”在不同环境下采用这些策略，可以让学生明白计算思维远比计算机更有用。

即使最小的孩子也能学到这种思维习惯。“我们正在教小孩攻克难题，让他们有逻辑地思考、抽象思考、找出模式、辨认替代物。”梅利莎·昂格尔 (Melissa Unger) 说。她是南费耶特学区幼儿园到2年级学生的科学、技术、工程、艺术和数学 (STEAM) 老师。“我们首先解决的问题是，‘什么是指令？如何下指令才能让电脑知道你想让它做什么？’我们让学生给自己的同学‘编程’，举起标有箭头的卡片来指导他们走出迷宫。”

和洛思阿图斯一样，南费耶特是一个富有、教育水平较高的社区，在匹兹堡新兴科技产业区工作的专业人士和许多在附近大学任教的教授都居住在这里。但是，正如华盛顿州肯特市的伊克赛尔公共特许学校 (Excel Public Charter School) 所提出的那样，计算思维可以教给所有孩子。伊克赛尔招收的学生群体中，37%是非裔美国人，19%是拉丁美洲人；超过一半的学生可以享受免费或减价午餐。尽管学校离科技巨头亚马逊和微软公司所在的西雅图只有几十千米，但伊克赛尔的学生仍然感到无法进入计算机世界。

伊莱·谢尔登 (Eli Sheldon) 是这所学校的计算思维项目负责人，他打算改变这种状况。“我和老师们一起计划，希望把他们教授的课程（不管

给大学教师搭桥

很多刚进入大学的年轻教职员都没有教书经验，如何才能让他们迅速适应新的教学环境？

撰文 **Jennifer Frederick**

年轻的科学家总会忙于处理研究生院和博士后职位的相关工作，常常需要撰写经费申请书，就连建立实验室的压力也超过了教学工作的压力。当这些研究生和博士后最终走上讲台时，他们几乎没有撰写教案和实际教学的经验，而学校又希望他们立即开始教学工作。这种情况在高中是不允许的，为什么在大学却是可以接受的？

指导研究生如何开展教学工作是个不错的机会，可以给这些未来即将走向教职的专业人士提供一些与实际教学有关的经验：什么样的教学方法是有效的，什么样的又是徒劳的。比如，尽管有证据表明，当本科生积极主动地参与课堂讨论和活动时，他们的学习状况最佳，但仍有许多大学教授（多为基础科学方面的）一直把讲课作为主要的教学方式。

最好的教学策略也跟科学的研究的技能是相通的。一位指导老师猜想，有种特殊的练习方式能帮学生更好地学到某一概念，后来他还专门整理了相关评估结论，检查学习效果。跟科学一样，每项“实验”的结果都会影响后续课程的设计。相反，那些传统的，采取低难度考试的课程，以及采用讲课为基本教学方式的课程，既无法挑选出表现优秀的学生，也无法让老师注意到学习有困难的学生。

根据反馈制定的教学策略，与学生成绩的提高息息相关。而且，对于弱势群体，以及第一代大学生（自己是家里第一位上大学的学生）来说，主动学习可以缩小与别人在成绩上的差距。2014年，华盛顿大学的斯科特·弗里曼(Scott Freeman)与合作者研究了几百个样本，开展了一项研究分析。他们发现，参与主动学习的学生与参加传统讲课的学生相比，前者在成绩上要高出6%，挂科率也要低很多。其他研究也发现，主动学习能缩小成绩差距，缩小的程度甚至高达45%。现在还很难解释为什么这种学习方法对某些学生特别有效。主动学习之所以有效可能是因为：更重视常规化的课前准备；频繁的交流也能帮助学生及时意识到自己知道什么，不知道什么。这些发现也说明，结构合理的大学科学课程能鼓励弱势的群体追求科学。

研究院所也逐渐意识到，向研究生和博士后提供教学机会的重要性。为此，美国专门设立了一些奖学金，其

中包括美国国立卫生研究院院校研究(NIH Institutional Research)奖学金和学术生涯发展奖(Academic Career Development Awards)奖学金。设置这两个奖学金的目的是给博士后提供有价值的培训，使他们获得在以少数族裔为主的研究所开展教学的经验。研究、教学、学习网络一体化中心也把研究生和博士生看作未来教职工人群的发展对象，让他们参与校园和虚拟学习社区的活动，获得提高职业技能的机会。

引导博士后从事教职员工作的正式项目并不常见，但是耶鲁大学教学中心（由作者担任中心理事）最近成立的一个教导项目，似乎具有转折性的意义。经过2年培训和课堂辅导，博士后会在第3年正式投入教学工作。他们会在有合作关系的地区性研究所里负责相关工作。让整个教导项目与社区大学、为弱势群体和第一代大学生服务的私立研究所建立战略合作关系，可以确保博士后在多样化的教学环境中收获实用的技能。

目前，关于教职工职业培训项目与本科生教育成果之间的相关性，还没有研究论文专门阐述。但是，最近的一项研究表明，参加教学培训项目超过55小时的研究生在教学时会更自信，而且更能成功守住自己的教职工岗位。后续的表现情况也说明，教学发展项目对博士后产生了有利影响，他们所教的学生也能从这种改变中受益。

经过反馈论证的教学培训，博士后具备了高效、全面的教学能力。这项行动不仅会使学习更加高效，还会惠及很多人，比如，对弱势群体和第一代大学生而言，这是鼓励他们涉足科学的关键因素。

Jennifer Frederick是耶鲁大学教学中心执行理事，她重视推进优秀教学人才培养的实施方案，支持学生进行跨校园学习。弗雷德里克还是一名化学家，还领导执行过夏季科学教学研究所(Summer Institutes)的项目，这是一个为大学科学教师设立的国家级培训项目。



学生们展示了动画放映（1），故事板（2-3）和艺术作品（4），他们把这些用在自己开发的游戏中。塞说她的学生不仅仅在编程：他们还在重塑自我认知。

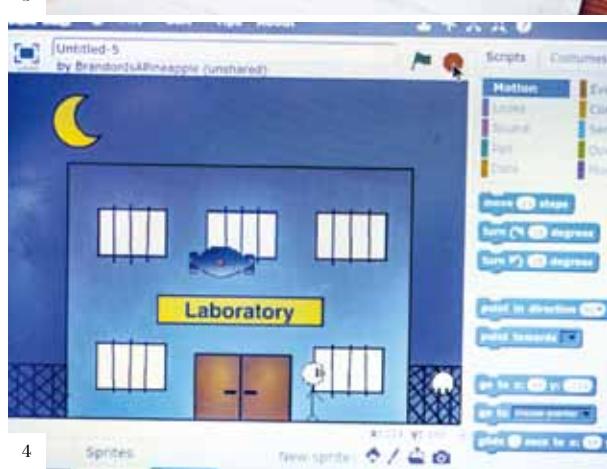
是英语、数学还是生物）与计算思维结合起来，”谢尔登说，“让学生在不同课程中使用相同的思维工具，就会使他们意识到这些工具是普遍适用的，即使到了校外也能用。”谢尔登和一名数学教师合作，创立了一个分析职业篮球比赛中运动员移动数据的班级团体；他还与一名人文学科教师合作，通过分析刑事审判系统，开展互动案例研究。

计算思维教学的拥护者认为，这种教学方式具有很多单纯的编码教学所不具备的优点：内容丰富、知识有深度；可以给学生一套灵活的思维工具，可以用在很多场景中；不论是在学校，还是走上了工作岗位，教给学生的知识和技能都长期有用。

在进入伊克赛尔学校之前，谢尔登在微软担任了4年的项目经理，在那里他见证了在现实世界中运用计算思维。“我看见过工程师们接手了一个个无



3



比复杂的难题，然后用计算思维解决了它们，”谢尔登说，“工程师们对拆分难题真的非常在行。他们会把各个部分按逻辑排列，每次测试其中一个部分，看看这一小部分的变化如何影响结果。我观察着他们，心想‘世界上每个人都应该知道如何运用这种方法’。”

美国普及计算机科学教育的行动仍在持续发酵。白宫科技政策办公室负责学习与创新板块的副理事库马尔·加格 (Kumar Garg) 也在推行这项行动。他解释了为什么这类项目可能会取得成功：

“奥巴马总统就职时，只有11个州同意将计算机科学课程纳入毕业条件，”加格说，“之后，情况发生了翻天覆地的变化，另外28个州和华盛顿特区都同意将计算机科学课程归入理科生毕业要求。”加格还称赞了预修计算机科学课程的“重启”政策，这项措施也将由相关大学委员会着手筹办。新课程与考试将重点放在编程和计算机科学基本原理上。“学生需要了解计算机科学的基本原理，不过，编程才是帮助学生明白这门课程内涵的方式。”他解释说。

加格说，在奥巴马政府提出“全民学习计算机科学”(Computer Science for All Initiative)的倡议下，政府要求学区提交扩展计算机科学教学方式的五年计划；政府会对计划精良的学区发放实施经费。当然，一些学校并不会等联邦政府走完缓慢的程序后再启动计算机科学教育的创新；然而，也有学校还没有准备设立这门学科。这种情形不禁让人想起科幻小说作家威廉·吉布森 (William Gibson) 的警句，他的作品探索了人类与科技之间的相互影响。吉布森已经看到了，“未来已然来临，只是分布还不均匀。”

2. Stuck in the Shallow End: Education, Race and Computing. Jane Margolis. MIT Press, 2008.
3. Learn about President Barack Obama's Computer Science for All initiative to empower a generation of American students with the computer science skills they need to thrive in a digital economy: www.whitehouse.gov/blog/2016/01/30/computer-science-all
4. Machine Learning. Seth Fletcher; August 2013.

参考文献

1. Computational Thinking. Jeannette M. Wing in Communications of the Association for Computing Machinery, Vol. 49, No. 3, pages 33–35; March 2006.

Annie Murphy Paul 是一位科学作家，她的文章经常发表在《纽约时报》、《时代周刊》等媒体上。她写的《个性测试与起源的崇拜》(The Cult of Personality Testing and Origins) 被《纽约时报》列为 2010 年度百部图书之一。她的下一本《才华：如何变聪明的科学》(Brilliant: The Science of How We Get Smarter) 将由皇冠 (Crown) 出版社出版。



无人驾驶还需60年

撰文 Steven E Shladover

很快，电子司机就能随时带我们去往我们想去的地方了，只要不强行左转弯穿过拥挤的车流，就能100%保障安全。变化多端的路面情况和冰雪覆盖也是问题。对于电子司机来说，在行驶中避让交通警察、交通协管员和急救车辆将是非常关键的环节；要是在行人可能突然跑到汽车前面的闹市区，

我们可能只能暂时和电子司机说再见，选择步行或者乘坐地铁。

就像前面讲到的一样，人类驾驶员在日常驾驶过程中能够轻松解决面临的问题，但是，对计算机来说，这些决策却是巨大的挑战。我们需要投入时间、金钱和大量的工作才能让计算机解决这些问题。



目前这种混乱的阐述十分糟糕。自动化驾驶技术就要来了，但是这种可以挽救生命、降低污染、节约能源的新技术，却不会以你之前理解的方式到来。

定义自动驾驶

驾驶车辆远比人们想象的复杂，这涉及很多技能和行为。其中有一部分非常容易实现自动化，例如，利用过去几十年在传统的巡航控制系统中积累的经验，可以让汽车在公路上自动保持匀速行使。随着科技进步，工程师们还成功实现了其他一些任务的自动化：目前广泛应用的自适应巡航控制系统可保持适当车速和车间距；在梅赛德斯—奔驰和英菲尼迪的新车型上也使用了车道保持系统，它可以协调摄像头、传感器收集的信息，利用方向控件使车辆维持在车道中间行驶。现在，汽车已然非常聪明了，可是即便如此，从目前的汽车自动系统到完全的自动驾驶，还有非常巨大的鸿沟需要跨越。

国际汽车工程师协会（SAE International，前身为汽车工程师协会）定义了关于自动驾驶的五个阶段，这种划分有助于我们厘清关于自动驾驶的很多问题。以自动化程度的不同为依据，前三个阶段（不包括零级，即无自动化）的技术全部需要依靠人类来处理行驶过程中的紧急情况。在第一阶段中，包括自适应巡航系统、车道保持系统等类似系统；在第二个阶段中，系统整合了第一阶段的技术（比如车道保持和自适应巡航控制系统中的车辆横向纵向控制），从而实现更复杂的自动驾驶任务。目前在市场中销售的自动驾驶车辆只能达到这一阶段；第三阶段的系统允许驾驶员在特定场景中切换到自动驾驶状态，比如在高速公路上堵车时。

最后的两个阶段与前面的阶段大不相同，完全不需人类协助。第四个阶段的（高自动化）系统可处理所有与驾驶相关的任务，但是使用场景严格限定在封闭停车场或高速专用车道上。顶级的第五阶段就是完全自动化的汽车了。日本日产汽车公司首席执行官卡洛斯·戈恩（Carlos Ghosn）等人

题。然而，现在有很多人笃定地认为，完全自动化的汽车马上就要来了。

是什么引发了这种分歧？一部分原因在于用词，大众媒体不加区别地运用“自主”、“无人”以及“自动驾驶”来描述这些不同的技术，却模糊它们之间的区别。此外，汽车行业也没有帮忙澄清事实。汽车制造厂、设备供应商和技术公司的营销者精心策划宣传材料，以便更好地诠释自家产品的自动化程度。行业内的记者在报道相关进展时，也会主动选择最乐观的预测，因为这样才会令人兴奋。信息在这样的环境中传播后，用户对产品不现实的预期很快就呈螺旋式上升了。

曾信心满满地宣布，2020年自动化汽车将能上路驾驶，想必很多人心里设想的自动驾驶汽车，就是这种了。

然而，顶级自动驾驶系统不可能很快上路，相反，它还有很长的路要走。目前距离第三个阶段就已经很近了，发展到第四阶段还需要几十年的时间。为了认识目前所处的混乱环境，也为了说明为什么会有以上预测，我们可以从软件的角度来谈一谈。

软件梦魇

无论公众怎么看，人类驾驶员还是非常善于避免严重车祸的发生。2011年美国交通安全统计数据显示，综合所有司机的驾驶情况，大概驾驶330万小时会发生一起致命碰撞；驾驶64 000小时会发生一起致伤碰撞。这些数字为自动驾驶系统设立了重要的安全目标，即自动驾驶的最低安全标准不得低于人类驾驶时的标准。想要自动驾驶的可靠性达到这个标准，还有很长的路要走，这中间的距离可比很多自动驾驶爱好者认为的长。

想想你的笔记本电脑多久死一次机，如果这样的软件用于汽车驾驶，“蓝屏死机”将不仅仅是一句抱怨，软件延迟十分之一秒响应都有可能引发交通事故。所以，自动驾驶涉及的软件必须使用截然不同的标准来设计开发，不能以消费市场上现有的设备为基础。

要达到这些标准极其困难，需要在软件工程和信号处理上都取得根本性的突破才行。首先，工程师需要使用全新的方式设计软件，设计出的软件还需要通过高标准的测试，确保在复杂和变化迅速的条件下也能准确运行。目前，在简单的应用场景中，已经采用了一种有趣的新方式：在真实应用一段代码前，分析所有可能出错的地方。这种方法就像用数学证明验证计算机程序一样。科学家已经开始考虑扩大这类测试，因为自动驾驶系统可能涉及到一些超级复杂的代码，用新的方式做测试，可以预先检验软件能否完全实现自动驾驶的要求。

一旦写好代码，软件工程师就需要尝试新的办法来调试和验证，因为现有的方法既繁琐又昂贵。客观来讲，在开发新型商用或军用飞机时，一半的成本都来自软件调试和验证，然而，飞机上使用的软件远不如自动驾驶系统需要的那么复杂。工程师设计飞机自动驾驶系统时，不需要考虑在飞行过程中，附近飞机的精确速度和位置，因为距离太远，总有足够的时间做出反应。飞机在飞行中，系统可以在命令下达数十秒后才制定出适宜的执行方案。反观自动驾驶的汽车，行驶时前后都有很多车辆，还有可能出现突然闯入眼帘的障碍物，面临突发问题。在任何情况下，汽车的自动驾驶系统都需要在几微妙之内做出决策，因此，和飞机上使用的系统和代码相比，这个系统复杂了几个数量级。

一旦代码通过验证，生产商就需要“证明”这套系统的安全性足以满足各方（公司风险管理人员、保险公司、安全倡导者、立法者和潜在客户）的要求。在这种情况下，常规使用的“验收测试”将变得不切实际。为了确保成千上万的消费者在日常使用中，车辆能够应对不同的危险场景，测试者需让汽车行驶数十亿千米。即使利用统计学的方法减少测试的距离，也要跑上很长的距离。为了解决这个问题，行业内部人员已经开始做一些新的尝试。德国的政府部分和行业机构已经启动了一个几百万美金的项目，但不得不说，努力才刚刚开始。

除了控制车辆的代码（通常比喻为大脑）需要接受严格审查，给大脑提供决策数据的传感器也需要接受同样严格的审查。工程师必须开发新的处理传感器信号的算法和数据融合算法，使假阴性（意外出现、没有登记在案的威胁性对象）和假阳性（非威胁性对象，但被错误分类，导致车辆出现急转弯和紧急制动等不恰当反应）的情况趋于零，高效区分道路上威胁性和非威胁性的对象，以便及时做出反应。

工程师不能求助商用飞机系统采用的强制冗余系统，因为自动化汽车是消费性产品，它的价格应该让大众能够负担。不过，求助于人工智能也不能显著解决安全问题。也有些人建议利用机器学

驾驶系统

自动化程度阶梯

汽车行业和媒体发明了很多术语来描述自动化无人驾驶系统。“自主”、“无人”和“自动驾驶”这些名词的出现，不仅没有理清各种概念间的不同，反而使它们更加模糊了。为澄清事实，国际汽车工程师协会编写定义，对不同的汽车的自动化程度做出了解释，并对自动化程度排出完整的序列。有意思的是，第四阶段的自动化技术要比第三阶段的自动化技术易于控制，而第五阶段的自动化技术（不需人类输入指令即可掌控一切驾驶条件的电子驾驶员）仍需几十年才能实现。

人类驾驶员监视环境			系统监视环境		
0 无自动化	1 驾驶员协助	2 部分自动化	3 有条件的自动化	4 高度自动化	5 完全自动化
不含任何辅助功能，例如自适应巡航控制。	帮助驾驶员保持车速或停留在车道内，由驾驶员控制。	上一阶段中一项或多项功能的混合，例如巡航控制和车道保持。	自动化系统驾驶并监视环境，但依赖于人类驾驶员支持。	可做所有事情的自动化系统——不需人类支持。当然，这仅限于部分环境。	真正的电子驾驶员：完全控制车辆，不需人类支持，所有条件下可自动驾驶。
谁负责转向、加速和减速	人类驾驶员	人类驾驶员和系统	系统	系统	系统
谁负责检测驾驶环境	人类驾驶员	人类驾驶员	人类驾驶员	系统	系统
出问题时谁控制车辆	人类驾驶员	人类驾驶员	人类驾驶员	人类驾驶员	系统
总体上辅助驾驶程度或自动化驾驶程度	None	人类驾驶员和系统	人类驾驶员和系统	人类驾驶员和系统	所有驾驶模式

来源: SAE International (www.sae.org/misc/pdfs/automated_driving.pdf); For documentation, including original definitions in their entirety, see Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems. SAE International, January 2014

习，这样就能使自动驾驶系统利用数百万小时的驾驶数据，训练出一个身经百战的系统。在消费者使

用时，这个系统还能一直学习，一直更新。但是，机器学习本身的问题就在于非确定性。如果用不同的



1

交通情况训练驾驶系统，在一年后，两辆从一条生产线出来的汽车也会产生截然不同的控制代码。

第四阶段的未来

我曾经说过，在2040年前完全自动化的驾驶系统很难出现，然而，不知从什么时候开始，就有人引用我的话说自动驾驶的最高阶段在2040年就能实现。我想说，到2075年完全自动化的驾驶系统都很难实现。真正实现的时间会比这个时间节点早吗？也有可能，但不会提早太多。

要实现第三阶段的自动化驾驶，也有很多问题需要解决。在出现紧急情况时，应该如何使驾驶人员重新集中注意力？比如，驾驶员欣赏风景开了小差，或者更糟，在开车时睡着了。我听过一些汽车制造商代表的讨论，他们觉得这些问题太令人头疼，以至于不想开发第三阶段的系统。在交通拥堵，车辆走走停停时，堵塞辅助系统就能接管驾驶，这时车速很低，即使发生碰撞也很轻微。但是，除了这种有限的使用场景外，第三阶段的自动化驾驶很有可能直接被跳过，永远也不会实现。

或许，在下一个十年中就会出现自动化驾驶程度非常高的汽车。现在，几乎所有大型汽车生产商和信息技术公司都在盯着这一块，他们对第四阶段



2

的自动化技术（受限于特定环境，但不依赖人类的完全自动驾驶）投入了大量资源。如果在一些特定的场景中，必须使用自动化汽车系统，那么第四阶段技术的可行性是非常高的。比如在机场里，一些专门设计的自动化载人轨道已经应用很多年了。

我们可以畅想，在未来十年间极有可能出现自动泊车系统。它允许驾驶员在车场入口处下车，让车自动进入装备完善、不允许行人和非自动化汽车进入的场地。此时，自动驾驶系统会与遍布车库的传感器通讯，找出可停放地点，并自动导航到目的地。可以想象，由于没有开关车门的需求，车位两边的空间也可以比现行标准更窄，这就能留出更多的空间，容纳更多车辆。

在城市中的人行区、商业区、大学校园和其他



明年，沃尔沃将对超过100辆SUV进行实地测试，这些SUV可以监测周围环境并自行导航。这些车辆已应用于欧洲公路列车测试。

不允许高速车辆通过的地方，低速的无人驾驶客车也很适用。这些环境中，性能有限的传感器也能有效探测行人和骑自行车的人。如果传感器探测到假阳性，在不必要的的情况下选择了刹车，顶多使车上的乘客感到不愉快，不会伤害其他人。2014年，欧盟委员会推出的城市出行计划(CityMobil2)中就一直在展示这类技术，近期，计划展示的内容又更新了一次。

把公交专用道和卡车专用车道与其他车道分开，就能很快使商用的自动驾驶车辆上路。按这种计划运行，就能让自动化驾驶的可行性得到明显提升。人车分离的方式能极大简化危险探测和反应系统，无人驾驶的卡车和巴士就能够形成节能车队，跟随领头的人类驾驶汽车行驶。全球有大量研究人员都测试过巴士和卡车车队自动化驾驶系统，其中包括加利福尼亚大学伯克利分校的加利福尼亚高级交通技术联盟项目、日本的能源智能交通系统项目、欧洲的Konvoi和安全道路列车项目(SARTRE)。

当我们期待下一个十年时，第四阶段的自动化高速公路系统很有可能会广泛使用，它能为个人用户提供直接服务，允许汽车在指定的高速路段自动行驶。这套系统中，车辆还包含了冗余的部件和子

系统，即使在出错时，也能自己能“跛脚回家”，不需人类操心。当然，这种行驶方式一般需要限制在晴天，车道上详细标明了各种指引标志。当出现故障时，车辆还可以自动驶入这些路段上设置的“安全港”。现在，大量汽车生产商都在努力开发这类系统，明年，沃尔沃将在瑞典哥德堡推行一项实地测试，到时候，100辆样车将在公路上测试自动驾驶系统的真实性能。

这些场景可能不像拥有私人电子驾驶员那样未来感十足，但是，这些技术一定能实现，甚至很快就能实现。

参考文献

1. Technical Challenges for Fully Automated Driving Systems. Steven Shladover. Presented at the 21st World Congress on Intelligent Transport Systems, Detroit, Mich., September 7–11, 2014.
2. Towards Road Transport Automation: Opportunities in Public-Private Collaboration. Summary of the Third EU-U.S. Transportation Research Symposium, Washington, D.C., April 14–15, 2015. National Academies of Sciences, Engineering, and Medicine, 2015.
3. Summary of definitions for SAE International's 2014 report Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems: www.sae.org/misc/pdfs/automated_driving.pdf
4. Driving toward Crashless Cars. Steven Ashley; December 2008.

Steven E Shladover 是位训练有素的机械工程师，拥有麻省理工学院(MIT)的学士、硕士和博士学位。20世纪80年代，他曾帮助加利福尼亚大学伯克利分校交通学院创建加利福尼亚高级交通技术联盟(PATH)项目。



保持联系。

无论你在哪里，都能紧随IEEE计算机协会的脚步。

在Twitter、Facebook、Linkedin和YouTube上关注我们。



@ComputerSociety, @ComputingNow



facebook.com/IEEEComputerSociety
facebook.com/ComputingNow



IEEE Computer Society, Computing Now



youtube.com/ieeecomputersociety