



软件开发

【特别策划】深入 .NET

大师论道 / 数据库的未来

人物 / SQL 之父 Don Chamberlin

访谈 / DB2 的过去、现在与未来

敏捷前沿 / 敏捷 2013

设计与建模 / 软件架构风格及其应用

【专题】

数据库编程

```
SELECT MIN(Table1.column1)
FROM Table1
WHERE Table1.column1 IN
  (SELECT Table2.column1
   FROM Table2)
```

```
SELECT * from Table1
WHERE Table1.column1 IN
  (SELECT DISTINCT Table2.column1
   FROM Table2
   WHERE Table2.column1 = 5)
```



分析 ▶ 设计 ▶ 管理 ▶ 安全 ▶ 嵌入式 ▶ Java ▶ C++ ▶ .NET

311.52-55
4

Delphi

in association with

Dr. Dobb's
JOURNAL

独家授权中文版

C/C++ Users Journal™

software
development

developer

新的开始

桌上年历的更换，预示着我们又要整装待发，踏上新的征程。

2003年，整个IT业在平淡中度过，然而对于软件研发界而言，这却是富于新意和蓬勃发展的一年。与5月《哈佛商业评论》提出的“IT无用”、“IT已经不能为企业提供竞争优势”的论调恰恰相反，软件技术在2003年成了业界关注的焦点。《纽约时报》指出，硅谷正在反思，由于当年那些只想轻易快速致富而对技术本身缺乏认识和热爱的人大行其道，硅谷的根基动摇了，而如今再造硅谷的行动中，技术人员正在重新获得权威。各大IT技术公司也都认识到，自己在广大技术人员心目中的地位，将成为争夺市场的重要核心竞争力。

2003年对于中国的软件研发界来说，更是难以忘怀：这一年国际厂商研发机构纷纷大举进驻中国，同时大大加强了对国内技术人员的重视程度；这一年国内企业在技术方面投入的力度、决心和取得的成果，也令人鼓舞，多项自有知识产权标准的诞生就是明证；这一年国内举行的各种技术大会、编程竞赛等等社区活动纷至沓来，令人眼花缭乱；这一年有众多的优秀技术图书出版，其中不乏紧跟世界潮流的经典佳作；这一年优秀的中文技术网站不断涌现，内容也愈加精彩；这一年《软件研发》诞生，使中国读者得以更近地接触世界最佳的技术名刊，聆听名刊背后众多大师的声音……作为软件研发人员，我们大概是第一次面对如此海量、更新迅速而且丰富的技术信息，第一次产生望洋兴叹的幸福感觉。

而2003年全球软件研发技术以及环境的发展，也在向我们预示着又一次“模式转变”（paradigm shift）即将到来。不，我不是说设计模式或者重构，虽然这些仍旧是《软件研发》在新年中重点关注的内容，但更应该是我们日常的必需品。平台在转变：除了.NET/Windows和J2EE/Linux/Unix本身的演变之外，我们还将逐渐从桌面转向互连环境，转向更为广阔的普及（pervasive）计算环境，也许过不了多久，我们的工作就会是为高清晰电视、Xbox、数码相机、TiVo（听说过TiVo吗？）、传感虫（知道未来可能遍布全球的传感器网络吗？）或者其他什么设备编写程序；新的工具也在不断涌现：主流厂商的开发工具在飞速发展，功能越来越强大，而且逐渐和软件工程工具融合起来，开源社区更是有无数极具潜力的项目在日新月异地更新着我们的兵器谱；软件开发逐渐从面向对象向面向构件、面向方面、面向服务、面向架构、按约定编程（Design by Contract）、按意图编程（Intentional Programming）……推进；而模型驱动开发的发展和XML对工具、平台的不断渗透，大大提升了开发的整体抽象层次，将使编程更加高效、简便，使建模者、程序员和界面设计者之间的界限变得更加模糊，从而改变我们整个开发世界的图景；从指令性方法到敏捷过程的转变，也渐渐成为大趋势。

从更广的尺度看，我们的整个行业还远未达到成熟，还面临着软件质量和易用性方面的严峻挑战。还记得当年盖茨和通用汽车公司的经典对话吗？

“如果通用汽车像微软那样进行技术开发的话，我们的汽车将每天无缘无故地故障两次……，每款新车上市的时候，司机都必须重新学习，因为操作方式改变很大……”

当软件在我们社会中承担的责任越来越多、越来越重要的时候，通用汽车当年冷冷的反驳，应该成为时刻提醒我们的警钟。

新的一年，新的开始，让我们携手，踏上新的征程。



2004年1月

目 录

专 栏

编辑寄语	1
软件近事	5
人物: SQL 之父 Don Chamberlin	7
大师论道: 数据库的未来	9
访谈: DB2 的过去、现在与未来	11

专 题

数据库编程	17
17 联网、ODBC 与 Perl <i>Robert Kiesling</i>	
23 应用程序级数据缓存 <i>Paul E. Boal</i>	
28 用 MinGW 移植 Oracle C/C++ 代码 <i>Laura Michaels</i>	
30 用 C 程序库管理 Access 或 MySQL 数据库 <i>Robert Walsh</i>	
34 搜集数据库的模式信息 <i>Mick Jones</i>	

特别策划

深入 .NET	39
39 .NET IL 初探 <i>Serge Lidin</i>	
48 在 .NET 框架中创建 Windows 服务 <i>Douglas Reilly</i>	

软件工程与管理

设计与建模	56
56 使用 UML 进行关系建模 <i>Davor Gorni</i>	
63 软件架构风格及其应用 <i>Stephen T. Albin</i>	
敏捷前沿	70
70 敏捷 2013 <i>Scott W. Ambler</i>	
模式专栏	73
73 对话: 构件模式的应用 (上) <i>Mark Völter 等</i>	

软件技术

C/C++ 专家论坛	80
[80] vector 的增长机理 <i>Andrew Koenig & Barbara E. Moo</i>	
C/C++ 技巧	83
[83] 使类不可继承 <i>Shanker Chandrabose</i>	
每月 Bug++	84
[84] Koenig 查找 bug <i>Jeff Claar</i>	
Java 核心技术	88
[88] 用 Java 处理 .INI 文件 <i>Mark Meyer</i>	
[X]ML 档案	92
[92] XML 和关系数据库 <i>Andrew Conrad & Dare Obasanjo</i>	
IBM dW 专栏	97
[97] Delphi DB2 决策支持开发 <i>Bob Swart</i>	
安全实验室	102
[102] 代码安全风险分析 <i>John Viega & Gary McGraw</i>	
嵌入式系统	108
[108] 实时信号分析与实时 Linux (上) <i>Matt Sherer</i>	
算法蹊径	115
[115] 全文搜索与 Burrows-Wheeler 变换 <i>Kendall Willets</i>	

Feature: Database Programming

- 17 Networking, ODBC, & Perl **Robert Kiesling**
23 Application-Level Data Caching **Paul E. Boal**
28 Porting Oracle C/C++ Code with MinGW **Laura Michaels**
30 Managing Access or MySQL Databases With a C Library **Robert Walsh**
34 Gleaning Database Schema Information **Mick Jones**

Column

- 1 Editorial
5 News & Views
7 People: Don Chamberlin
9 Talk: The Future of Database
11 Interview: DB2: Past, Present, Future

Special: Inside .NET

- 39 Introducing .NET IL **Serge Lidin**
48 Create Windows Services in the .NET Framework **Douglas Reilly**

Management&Engineering

- 56 Design&Modeling
Relational Modeling with UML **Davor Gorni**
Software Architecture Style and Its Application **Stephen T. Albin**
70 Agile Edge
Agility 2013 **Scott W. Ambler**
73 Patterns
Conversation: Application of Server Component Patterns: Part 1 **Mark Völter et al**

Technology

- 80 C/C++ Experts Forum
C++ Made Easier: How Vectors Grow **Andrew Koenig & Barbara E. Moo**
83 C++ Tips 5
Making Classes Non-Derivable **Shanker Chandrabose**
84 Bug++ of the Month
A Koenig Lookup Bug **Jeff Claar**
88 Core Java
Creating .INI Classes in Java **Mark Meyer**
92 [X]ML Files
XML & Relational Databases **Andrew Conrad & Dare Obasanjo**
97 IBM developerWorks Column
Drill Into DB2 UDB Tables for Decision Support Using Delphi Studio
or C++Builder Studio **Bob Swart**
102 Security Labs
Risk Trees **John Viega & Gary McGraw**
108 Embedded Space
Real-time Signal Analysis & Real-Time Linux: Part 1 **Matt Sherer**
115 Algorithms Alley
Full-Text Searching & the Burrows-Wheeler Transform **Kendall Willets**

执行主编: 刘江
责任编辑: 贾梅
编辑: 吴怡 陈剑瓯
编辑信箱: editor@ddjchina.com
美术编辑: 锡彬 禾余
排版制作: 金浩
网站: 周明涛

编委会

中文版: 李 维 潘爱民 钱 岭 孙以义 夏 昕
曹晓钢 左轻侯 李会武 武卫东 温莉芳
英文版:

Dr.Dobb's Journal

J.Erickson, A.Stevens, B.Schneider, R.Duncan, J.Woehr,
J.Bentley, T.Kientzle, G.Wilson, M.Nelson, E.Nisley,
M.Swaine

Software Development

A.W.Morales, S.Ambler, D.Cline, W.Keuffel, A.Barnhart,
G.Evans, L.O'Brien, R.Wayne, B.Douglass, M.Fowler,
E.Gottesdiener, R.Martin, B.Meyer, S.Meyers,
M.Poppendieck, G.Pour, J.Rothman, G.van Rossum

C/C++ Users Journal

J.Casad, C.Allison, D.Abrahams, B.Karlsson, D.Saks,
H.Sutter, M.Austern, T.Becker, S.Dewhurst, A.Koenig,
R.Meryers, B.Moo, B.Schmidt, R.Ward

Windows Developer Magazine

J.Dorsey, J.Claar, D.Esposito, G.Frazier, R.Grimes,
P.Hesselberg, P.Tomlinson, V.Volkman

读者服务

电子邮件: reader@ddjchina.com
电 话: 88379061 88379062
网 站: www.ddjchina.com

版权登记号 图字: 01-2003-1691

图书在版编目(CIP)数据

Dr. Dobb's 软件研发. 第6辑 / (美) 凯尼格 (Koenig, A.), (美) 穆 (Moo, B.E.) 等著; 荣耀等译. —北京: 机械工业出版社, 2004.1

ISBN 7-111-12248-8

I. D… II. ①凯… ②穆… ③荣… III. 软件研发

IV. TP311.52

中国版本图书馆CIP数据核字 (2004) 第003228号

出版发行

机械工业出版社

(北京西城区百万庄大街22号 邮政编码 100037)

印刷: 中国电影出版社印刷厂

版次: 2004年1月第1版第1次印刷

889mm × 1194mm 1/16 · 7.5印张 16彩页

定价: 18.00元

版权声明

Dr.Dobb's Journal China contains articles under licenses from CMP Media LLC. The articles appearing on pages 17,23,28,30,34,48,70,80,83,84,88,92,102,108,115 are translated and reprinted by permission of Dr.Dobb's Journal, C/C++ Users Journal, Windows Developer Magazine, and Software Development copyright©2003 CMP Media LLC. All rights reserved.

本书部分文章翻译自 Dr.Dobb's Journal, C/C++ Users Journal, Windows Developer Magazine 和 Software Development. 由 CMP Media LLC 独家授权。未经出版者书面许可, 不得以任何方式复制或转载部分或者全部内容。版权所有, 侵权必究。

最热门的技术项目

技术日新月异,从事软件研发的人,应该心怀天下,博采众长,可不能故步自封。知道目前最热门的高技术项目都有些什么吗?权威技术杂志《MIT技术评论》最新一期广泛调查了全球顶级公司的研发机构,从中选出了7项最具吸引力的项目。也许,几年之内,这些项目的成果就将转化为产品,走入你我的生活:

自动语音翻译系统, IBM 公司, T. J. Watson 研究中心

状态: 2004 年中可能出现在笔记本电脑和 PDA 上

社会学家们总是声称计算机阻碍了人际交流,而 IBM 公司雄心勃勃的计划 MASTOR(Multilingual Automatic Speech-to-speech Translator) 将借助计算机拉近操不同语言的人们之间的距离,使人们能够更自由、更方便地进行交流。目前的原型系统只能实现英语和中文普通话之间的互译,但系统本身是独立于语言的。值得一提的是,项目的协调人是曾在中科院工作过的高玉清(Yuqing Gao)女士,项目的开发使用了大量 C++ 和 Tcl 语言。

脊髓损伤治疗, Biogen 公司

状态: 三到四年之内有望进行临床试验
医学界长期以来认为人类的脑和脊髓神经的损伤是不可恢复的,但是耶鲁大学的神经化学家 Stephen M. Strittmatter 分离出了阻止神经恢复的物质,从而打开了通向拯救之路的大门。

垃圾邮件阻止技术, Microsoft 公司, 硅谷研究院

状态: 已经进入产品开发阶段

垃圾邮件已经成为巨大的社会公害。早在 10 多年前,spam(垃圾邮件)这个词对一般美国人来说还只意味着午餐肉的时候,算法学家 Cynthia Dwork 就开始思考如何阻止 spam 的问题了。在 CRYPTO'92 会议上她和 Moni Naor 提出了一个简单的方法: 提高发邮件的成本。比如,如果要给不认识的人发信,你的计算机需要先花 10 秒钟解决一个数学问题,这样即使全天

运行,也只能发送 8600 封邮件,从而大大遏制了垃圾邮件的泛滥。然而,这种简单的技术离实用还有不小的差距: 对于计算机能力不同的机器显然影响也差异很大。怎么办呢? Cynthia 的同事 Ted Wobber 从内存(各种机器之间内存延迟差异很小)着手,改进了设计: 计算要对内存进行存取。其他方案还有需要人参与的图灵测试等等。19 世纪以前,因为邮寄由收信人付费,计费非常繁琐,而黑便士邮票的问世宣告了付费方式的革命,彻底改变了邮政的面貌。所以 Microsoft 公司的垃圾邮件阻止项目就称为 Penny Black(黑便士)。此项技术有望很快在新版本的 Outlook、Hotmail、邮件服务器和浏览器中应用。



微型超声波设备, GE 公司

状态: 2005 年之前进入评估阶段

GE 公司的科学家们正在用硅代替传统的压电材料,设计出医生能够放在口袋里的超声波设备。

芯片间通信, Sun 公司

状态: 5 年内将用于计算机

如何突破摩尔定律的局限? Sun 公司的科学家正在利用电容耦合实现芯片间通信,从而大幅提高系统速度。

流媒体, HP 公司

状态: 两年内进行大规模测试

如何使流媒体能够在各类设备中流畅地传输? HP 公司的解决方案是在现有网络中加入一些节点,通过相应的硬件和软件,使流媒体内容离用户更近,并实现缓存、路由优化、自适应设备等等功能。

压电控制的燃料喷射, 西门子公司

状态: 已经用于柴油机, 2006 年前用于汽油机

采用压电材料精确控制燃料喷射,可以增加里程数,减少尾气排放。此技术已经在标志汽车中得到应用。

Microsoft 的新年不轻松

2003 年最后一天,微软又遭到重创: 由于降价和套件拆开购买的要求没有得到满足,以色列负责政府采购的财政部发言人发表声明说,将不再升级目前使用的 Windows/Office,同时鼓励低价替代产品的开发。以色列政府计划与 Sun 和 IBM 合作,将办公环境迁移到运行在 IBM 硬件上的 OpenOffice。虽然当地的微软官员嘲笑说 OpenOffice 1.1 的功能只相当于 Word 97,但是它显然已经能够满足需要,因为目前 OpenOffice 支持从右到左读写的希伯来文和阿拉伯文。差不多与此同时,英国的政府财政办公室以及一家大型的医疗服务机构都宣布转而采用 Linux 系统。其他的类似案例还发生在法国、中国、德国、巴西、韩国等等,越来越多的政府开始研究用开源软件替代微软产品的可能性。而按照 IDC 的数据,政府部门在全球 IT 市场中占 10% 的份额,这还不包括其巨大的影响力。面对这样的冲击,微软的全球定价体系在 2004 年可能必须有所改变。

微软的麻烦还远没有结束,最近牵涉的官司之多,让人有眼花缭乱的感觉: 除前几个月我们已经报道过的微软已经败诉的 Eolas 公司控侵犯网页嵌入动态内容专利案(赔偿 5 亿多美元)之外,还有 ImagExpo 公司控告 NetMeeting 软件侵犯其白板专利案(赔偿 6000 万美元),Real Network 公司控告其不正当竞争,索赔更是达到 10 亿美元……

BIOS 将完成使命? EFI 横空出世

也许不久之后我们就会告别 BIOS——自 1982 年以来一直使用的 PC 机启动系统。Intel 和 Microsoft 将大力推广一项旨在取代 BIOS 的技术: EFI (Extensible Firmware Interface)。

虽然 BIOS 的设计非常简单,但是 20 多年的发展、不断的补丁添加,它已经面目斑

驳。毕竟它所服务的操作系统已经更换了4代,硬件更是早已经今非昔比。

新的EFI本身其实就是一个小的操作系统,完全用C写成(而BIOS是用汇编语言写的),可以用标准程序设计工具编写扩展。它还提供了高清晰度的GUI,具有自己的文件系统和网络功能,在操作系统崩溃时,仍然能诊断硬件系统。当然,至关重要的数字权限管理和安全方面的功能更是强过BIOS多多。

取代过程当然不会在一夜之间发生。BIOS主要厂商Phoenix就表示,在EFI成为标准之前,该公司不会很快跟随。它也在开发下一代的BIOS: Core Management Environment。

Microsoft将在下一代操作系统Longhorn中,Intel也将在下一代芯片组中支持EFI。两公司表示会尽快成立一个论坛组织对其进行标准化。

为Internet画地图

我们都喜欢把Internet称为虚拟世界,但是Internet到底长什么样呢?最近,Opte计划完成了对整个Internet地图的绘制,其中各种颜色表示地理上不同大洲地域的网站。整张地图与宇宙星云和礼花是不是很像?炫目的画面,就算是《软件研发》编辑部送给大家的新年礼物吧。



红色表示亚太地区,绿色表示欧洲、中东、中亚和非洲地区,蓝色表示北美,黄色表示拉丁美洲和加勒比海地区,青色表示RFC1918 IP地址,白色表示未知。

标准中国造

2003年5月12日发布、7月做过大规模宣贯推介活动的中国国家强制性无线局域网标准(WAPI: WLAN Authentication and Privacy Infrastructure, 无线网鉴别和保密基础结构),却在新旧年交替之际成为了媒体和业界关注的热点。

12月4日,华尔街日报报道:“中国无线技术加密标准引发业界慌乱”。12月9日,IEEE 802与城域网标准委员会主席Paul Nikolich给中国标准化管理局局长李忠海和信息产业部部长王旭东发来一封信,信中指出:“新标准的强制执行会对无线网络产品市场造成很大的冲击,将全球市场一分为二。”

事情的起因是,11月26日,国家质量监督检验检疫总局和国家标准化管理委员会联合发布一份公告:12月1日开始,禁止进口、生产和销售不符合强制性国家标准的无线局域网产品。12月1日,国家质量监督检验检疫总局联合国家认证认可监督管理委员会宣布,根据相关规定,决定对无线局域网产品实施强制性产品认证。

无线局域网安全问题,已经成了这项技术应用推广的屏障。Wi-Fi联盟也意识到了事情的紧迫性,他们把向802.11i过渡的安全解决方案WPA定为强制性认证项目

并于今年8月份开始实行。但是,WPA技术核心仍然和WEP技术相同,它们的区别在于在通信的过程中不断地变更WEP密钥,变换的频率以假设目前的计算技术无法将WEP密钥计算出来为依

据。但是对称加密的不足在于AP和工作站使用相同密钥,包括变更密钥在内的信息会在相同的简单加密数据包中传输,黑客只要监听到足够的数据包,借助更强大的计算设备,同样可以破解网络。

中国国家标准将要求用WAPI协议取代IEEE802.11标准中先天不足的WEP协议。与其他安全机制相比,WAPI要更胜一筹。它已由ISO/IEC授权的IEEE Registration Authority审查获得认可,是我国目前在该领域惟一获得批准的协议。

让我们祝愿Made in China的技术标准:WAPI、EVD、AVS等在新的一年里都一路走好。

名人动态

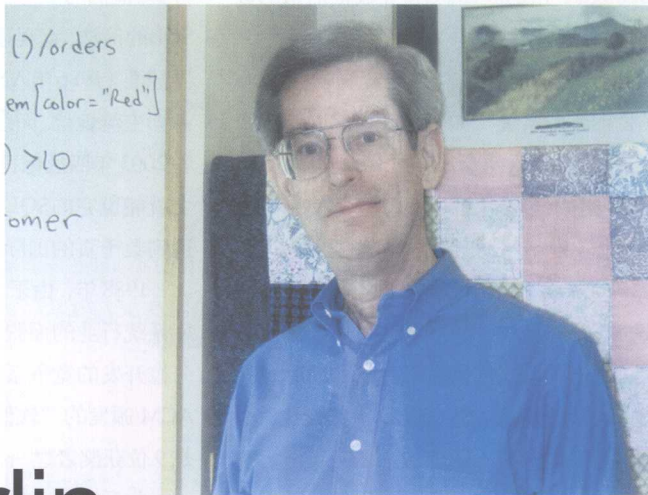
Tim Berners-Lee 英国白金汉宫在新年前公布,英国女皇伊丽莎白二世册封Tim Berners-Lee为英国骑士团(the Order of the British Empire, KBE)二级骑士(Knight Commander)爵位,以表彰Tim Berners-Lee发明WWW以及对其发展所作的贡献。祝贺你, Tim!

Grady Booch: 自加入IBM以后,Booch的头衔从首席科学家变成了IBM名士,工作内容主要是与IBM研究中心联络,过得非常清闲。他形容自己在过去一年的经历仿佛是一个孩子来到了琳琅满目的糖果店。至于新的一年,他的工作将更有“目的性”,因为Rational将在开发和设计工具方面有较大投入。

Ward Cunningham 大概很多人都想不到,模式、极限编程和Wiki的先驱,一直过着闲云野鹤般日子的Cunningham,最近居然加入了Microsoft,负责指导架构、模式和企业级开发实践方面的技术和推广工作。目前Java和J2EE企业级开发、模式等方面发展得非常迅速,此领域中近年来好书迭出,就是明证。Cunningham这样大腕的加盟当然将有助于.NET模式研究和应用迎头赶上。

(更多新闻和评论,请访问 www.ddjchina.com)

SQL 之父 Don Chamberlin



2003年对于Don Chamberlin来说,可以说是收获之年:这一年他获得了IBM公司技术方面的最高荣誉IBM名士(Fellow)称号;获得了ACM SIGMOD颁发的创新奖;获得了母校Harvey Mudd学院的杰出校友奖;更重要的是,他还获得了《软件研发》英文版《Dr. Dobb's Journal》颁发的Dr. Dobb's程序设计杰出奖(Excellence in Programming Award),与Linus Torvalds、James Gosling、Anders Hejlsberg等一起载入史册。

Don Chamberlin获得这些殊荣当然是无愧的:是他发起了数据查询的两次革命,他是SQL语言的创造者之一,也是XQuery语言的创造者之一。今天数以百亿美元的数据库市场的形成,与他的贡献是分不开的。

Don Chamberlin似乎天生与数据库、信息检索有缘:小的时候,家里的一本100多磅重的百科全书是他的最爱,在他看来,这大概是数据库的最早形式。作为地地道道的硅谷人,他的本科是在规模很小但是声誉很高的Harvey Mudd学院度过的,这个学校至今仍然保持每年从1600多名申请者中仅招收100多名学生的制度。

在斯坦福大学获得博士学位以后,Chamberlin加入了位于纽约的IBM T.J. Watson研究中心。那里汇聚了当时计算机界的大批精英,主要研究方向是操作系统。Chamberlin一开始从事的项目是System A,一年后,项目最终失败。当时担任项目经理的Leonard Liu(后来曾担任Cadence公司首席运营官)很有远见地预见到数据库的美好前景,他转变了整个小组的方向。Chamberlin从此如鱼得水,在数据库软件和查询语言方面进行了大量研究。他成了小组中最好的网状数据库CODASYL专家,整日徜徉在复杂指针和集合选择规则的海洋中。与此同时,20世纪60年代晚期,在美国西海岸IBM圣何塞研究中心(Almaden研究中心的前身)工作的数学家Edgar Codd独自创造了关系数据库的概念。但是,由于这种思想对IBM本身已有产品造成了威胁,公司内部最初是持压制态度的。当然这也与Codd采用了太多数学方法(以谓词演算为基础),不容易理解有关。在Codd奠基性的论文“A Relational Model of Data for Large Shared Data Banks”以及许多重要的后续论文公开发表之后,IBM研究

中心从事数据库的研究人员中,还有不少人以怪异思想视之,或者半信半疑,或者完全摸不着头脑的,甚至有人做出如此轻蔑的评论:“这东西过十年再说吧。”

一天,Codd到Watson研究中心访问,在讨论会上,他几乎用一行语句就完成了类似于“寻找比他的经理挣得还多的雇员”这样的查询。“我的天!”Chamberlin知道,这个查询用CODASYL来表示的话,可能要超过5页纸。这种强大的功能使Chamberlin醍醐灌顶,幡然醒悟,从此转向了关系数据库。

在其后的研究过程中,富于程序员气质的Chamberlin相信,Codd提出的关系代数 and 关系演算过于数学化,无法成为广大程序员和使用者的编程工具,这个问题不解决,关系数据库也就无法普及。因此他和刚刚加盟的Ray Boyce设想出一种操纵值集合的关系表达式语言——SQUARE(Specifying Queries as Relational Expressions)。

1973年,IBM在外部竞争压力下,开始加强在关系数据库方面的投入。Chamberlin和Boyce都被调到圣何塞,加入新成立的项目System R。当时这个项目阵容十分豪华,有Jim Gray, Pat

Selinger, 和Don Haderle等数位后来的数据库界大腕。

System R项目分成研究高层的RDS(关系数据系统)和研究底层的RSS(研究存储系统)两个小组。Chamberlin是RDS组的经理。由于SQUARE使用的一些符号键盘不支持,影响了易用性,Chamberlin和Boyce决心进行修改。他们选择了自然语言作为方向,其结果就是“结构化英语查询语言(Structured English Query Language, SEQUEL)”的诞生。当然,后来因为SEQUEL这个名字英国已经被一家飞机制造公司注册了商标,最后不得不改称SQL。有趣的是,他们的野心很大,想借此实现非程序员也就是普通大众也能广泛应用计算机的梦想(当然,那时候还没有图形界面)。因此,项目组找来了一位语言学家,她跑到圣何塞州立大学,找了许多不懂计算机的大学生,教授他们SEQUEL和SQUARE两种语言,像白居易当年对老姬吟诗那样,寻找改进的

方案。他们的梦想当然并没有实现,然而,SQL的简洁、直观还是使它迅速成为了世界标准(1986年ANSI/ISO),30年后仍然占据主流地位。而经过了1989、1992、1999和2003年四次修订,当初仅20多页的论文就能说完了的SQL,如今已经发展为篇幅达到数千页的国际标准。

1988年,由于“革命性地改变了数据库系统行业的面貌”,System R和伯克利分校开发的竞争系统INGRES共同获得了ACM颁发的“软件系统奖”,Chamberlin是9位获奖者之一。

此后,Chamberlin曾一度顺应个人电脑的大潮,对桌面出版发生了兴趣。他领导了一个小组开发了一种文档编辑/格式化程序,名为Quill。其中文档的逻辑视图是用SGML保存的。这是Chamberlin第一次接触标记语言。

20世纪90年代,Chamberlin再次返回数据库世界,开始从事对象——关系数据库的开发,其成果在DB2中得到了

体现。其间他曾撰写过一本专门讲DB2的书《A Complete Guide to DB2 Universal Database》(Morgan Kaufmann, 1998)。在网络时代到来,XML日益成为标准数据交换格式的时候,Don看到了自己两方面研究经验——数据库查询语言和文档标记语言相结合的最佳时机。他成为IBM在W3C XML Query工作组的代表,并与工作组中两位同事Jonathan Robie和Dana Florescu一起开发了Quilt语言,这构成了XQuery语言的基础。而后者经过多年快速发展,即将成为W3C的候选标准。对于Chamberlin来说,XQuery语言标志着自己“整个职业生涯中的又一个高峰。”他深信Web数据技术的发展将带来第二次数据库革命。

Chamberlin的学术成就,使他1994年当选为ACM会士,1997年当选为美国工程院院士。他对于教育一直很有兴趣,多年来一直担任ACM国际大专程序设计竞赛(ICPC)的出题人和裁判。

Dr. Dobb's程序设计杰出奖

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

- ④ 1991年: Alexander Stepanov, C++ 标准模板库(Standard Template Library, STL)的开发者。
- ④ 1992年: Linus Torvalds, Linux, 操作系统背后的关键力量。
- ④ 1993年: Larry Wall, Perl 语言的作者。
- ④ 1994年: James Gosling, Java 首席架构师。
- ④ 1995年: Ronald Rivest, 教授、作者和计算机安全专家, RSA 算法设计者之一, 2002年图灵奖获得者之一。
- ④ 1996年: Gary Kildall, 操作系统、程序设计语言和用户界面等领域的先驱。
- ④ 1997年: Erich Gamma, Richard Helm, John Vlissides 和 Ralph Johnson, GoF, 《Design Patterns: Elements of Reusable Object-Oriented Software》一书的作者。
- ④ 1998年: Guido van Rossum, Python 之父。
- ④ 2009年: Donald Becker, Linux 联网功能开发者, Beowulf 项目的首席研究者。
- ④ 2000年: Jon Bentley, 计算机科学家, 本刊英文版特邀编辑。
- ④ 2001年: Anders Hejlsberg, Turbo Pascal 开发者, C#和.NET 框架架构师。
- ④ 2002年: Adele Goldberg 和 Dan Ingalls, Smalltalk 和面向对象语言先驱。
- ④ 2003年: Don Chamberlin, SQL 语言和 XQuery 语言的开发者。

大师论道： 数据库的未来

从最早用文件系统存储数据算起，数据库的发展已经有50多年了，其间经历了20世纪60年代的层次数据库（IBM的IMS）和网状数据库（GE的IDS）的并存，70年代~80年代关系数据库的异军突起，90年代对象技术的影响。50年后，关系数据库依然处于主流地位。在关系技术之后，在新形势下，数据库应该如何发展？又有哪些重要的趋势呢？最近，《Software Development》杂志举办了一次专家论坛，讨论数据库的未来。论坛由咨询师和作家，《Dr. Dobbs's Journal》XML与Web Services技术频道的编辑Ken North主持，与会的有六位大师：Rick Cattell，Don Chamberlin，Daniela Florescu，Jim Gray，和Jim Melton。



Rick Cattell 是 Sun 公司杰出工程师。主持了 JDBC API 的设计，他创办了对数据库技术产生

了很大影响的 ODMG 组织并担任主席，是对象数据库方面的世界权威。目前，他担任着 Sun 公司 Java 企业级技术集团首席架构师的要职，执掌着自己参与开创的 J2EE 的未来技术方向。



Jim Melton 长期担任 ISO/IEC SQL 标准的编辑，SQL 标准委员会的长期成员，先后供职 DEC、

Sybase，现任 Oracle 公司高级工程师。他还是旨在结合 SQL 和 XML 的项目 SQLX 的发起人。

讨论首先从回顾历史开始。North 回忆起当年一辆 IBM 1410 计算机的价格得上 300 辆野马跑车，而现在一台双 CPU 带 RAID 的服务器，价格还不到一辆野马跑车的 10%，大家都笑起来。相对硬件的发展速度而言，数据库的功能似乎进展得还不够快。毋庸置疑，今天，信息的存储、组织、管理和访问，依然是计算机技术最重要的领域之一。近年来，随着科学技术、电子商务尤其是 Internet 的迅猛发展，信息的来源、数量、形式和使用方式都发生了很大变化，使传统关系数据库面临着巨大挑战。各位大师面向未来展望了各种重要的趋势。

Jim Gray 认为：“如今，普通使用者也开始有了在桌面电脑上方便地存储和访问任何形式海量信息的需要，比如自己喜欢的一首歌曲、家里拍的一段视频、

一幅扫描的图片，这些现在非得保存为文件的东西，就需要能够直接存入数据库。而检索时，既可以通过一些标识文字，也可以通过乐句查询、人脸识别、甚至是 OCR（光学字符识别）等等更自然的方式。微软目前有 3000 人的开发团队，正在致力于将所有这些集成在桌面操作系统中。我的小组里，有三个人在开发类似的但是相对廉价的版本。所以，这样的梦想可能不久后就会变成现实。当然，整个过程难度很大。”

“这的确是一个大趋势，” Jim Melton 表示同意。“事实上，和过去数据库处理的结构化信息不同，我们今天首先要解决的是如何处理复杂类型的数据：文本、时间性、空间性、声音、图形和视频数据等等。最有效的方式，显然是数据库本身就支持它们，将它们作为内置数据类型



Jim Gray 是 1998 年图灵奖得主, 早年在 IBM 参与过 IMS、System R 和 DB2 的开发, 后曾供职于 Tandem 和 DEC 公司。1993 年微软想尽一

切办法, 甚至为他专门在湾区开设研究院, 终于请他加盟。SQL Server 7.0 的成功想来离不开 Gray 的强大支持。



Daniela Florescu 是著名的 XML 查询引擎开发商 XQRL 公司的 CTO, 在公司被 BEA 购并后进入后者任高级软件工程师。她曾在法国计算机科学研究

院 (INRIA Rocquencourt) 任研究员, 是 W3C XML Query 工作组成员, 与 Chamberlin 等合作开发了 XQuery 语言。



Don Chamberlin 是 IBM 名士, SQL 之父 (生平参见本辑文章《人物: SQL 之父: Don Chamberlin》)。

中的一等公民。”

来自对象数据库技术阵营的代表人物 Rick Cattell 三句不离本行: “对象数据库或者对象-关系数据库已经很好地实现了对这些扩展数据类型的支持。目前的数据库已经同时支持对文本和结构化数据的查询了, 但是在同时支持信息检索的概率推理方面还有差距。”他还谈到了数据库中对编程的支持, 毕竟他是 Sun 公司 J2EE 的技术掌门人, 是 JDBC、JDO 背后的力量。“数据库如今已经加入了用户定义过程的功能, 允许添加带有一定行为的数据类型。但是, 过程以及其他可执行代码实现方式比如触发器, 都还是数据库中的二等公民, 这一点需要做出改变。”

而来自 BEA 的 Florescu 则提到了目前比较热门的传感器网络计划, 将在很大的地理范围内部署巨量节点的传感器, 监控一定区域内各种物体的状态和位置。这个项目本身蕴含着巨大的潜力, 有可能掀开军事情报和指挥、环境监测、资源勘察、气象预报、交通控制等等领域的新篇章, 甚至改变我们的日常生活: 如果家里的每个物体上都有一个传感器的话, 找不到或者丢失东西, 都将成为历史。这对数据库技术当然提出了极高要求。想象一下, 成千上万, 甚至数以十万、百万计的传感器实时发送的流信息, 该是何等景象?

对于 P2P 计算能否取代客户端-服务器模型成为主要数据库存储方案, 专家们

取得了一致。Chamberlin 说: “P2P 可能更适用于更新较少的应用场合。”而 Jim Gray 认为, 如果没有任何网络带宽限制的话, P2P 模式才是有可能的, 因为它需要大量的数据传输。他们对所谓元组空间 (tuple space) 计算模型都给予了负面评价。

在 Jim Gray 看来, 缺乏元数据是目前数据库技术无法解决 Web 中问题的主要原因。举例来说, 寻找满足某种条件的一张照片或者一段音乐, 就目前而言, 是难以解决的。

当听到有人问及“如何在文档中进行查询”时, Don Chamberlin 当仁不让地接过了话筒。他比较了传统数据库中的结构化数据和 Web 上非结构化或者半结构化、异构、稀疏型的数据。他说: “自从上世纪 90 年代以来, 似乎一夜之间, 所有计算机都连接起来了, 信息空前激增。但是这些突然出现的信息并不都是由行和列组成的, 有跟顺序无关的, 还有流式数据比如股票行情报告系统。”有着数据库查询和文档处理两方面背景的 Chamberlin 深信, XML 在未来的数据库中将成为中心地位: “XML 将成为文档交换的主要格式。它是灵活的, 而且能够自我描述。许多程序都需要用交换数据的格式进行查询。其他数据都将映射为 XML, 通过接口与外界联系。但是关系数据库和 SQL 不会退出历史舞台, 因为它们在数据的无关性方面, 在处理关系上有着天然的优势。”当然, Chamberlin 也承认: “还有许多工作要做。比如更新

和事务处理问题、XML 数据的索引问题就需要认真解决, 这也是 W3C XQuery 工作组目前的重点。我们必须像多年来开发关系数据库那样, 围绕 XML 构建健全的基础设施。”

对于数据库未来的发展方向, Cattell 显然有不同的看法: “我可并不认为全 XML 数据库会有美好的未来。人们难道会完全用 XML 格式来存储和查询数据吗? 性能肯定是大问题! 我在 10 年前就站到了另一边, 赞成通过面向对象技术解决问题, 实际上目前的主流厂商, 都已经关系数据库中很好地支持了对象。互操作性和兼容性当然重要, XML 作为通用的数据交换模型, 因为它的可扩展性, 可能会非常适用于各种特殊目的的专门市场, 但是绝不会替代关系数据库。”

而作为 Chamberlin 的亲密战友和 XML 查询引擎方面的专家, Daniella Florescu 表明了自己对于 XML 数据库的信心: “我认为 XML 查询的优化潜力是很大的, 其性能完全有可能超过 SQL。目前, 在 XML 元素、Java 对象和数据库表之间进行数据编组 (marshalling) 和解组是性能的瓶颈所在, 这也就是所谓 ‘Web 服务阻抗’。凭我的经验, 我相信未来如果直接用 XML 编程, 就可以克服这种问题。”

短短的 1 个小时时间, 当然不可能将数据库未来的发展完全涵盖, 但是与会者都从中受益匪浅。类似的讨论会, 今年 3 月份举行的 SD 大会上还会举行, 让我们翘首关注吧。

访谈:

DB2 的过去、现在与未来

Pat Selinger、Don Haderle 和 Don Chamberlin 是第一个 DB2 产品的主要缔造者,并在 20 年的变革中不断对其进行完善。这三位现代 RDBMS 的创始人一同铸就了 DB2 的发展史并继续制定着未来的发展计划。

从 20 世纪 70 年代开始,“更快、更好、更廉价”就已经成为了关系数据库的代言词,当时 IBM 软件先驱们第一次开始着手开发 System R。System R 是 IBM 研究部门开发的一种产品,这个原型系统促进了技术的发展,并最终在 1983 年以 DB2 的形式进入到了商业市场。历史表明, System R 并不仅仅是一种原型系统,它还代表了更多的含义:它是一群才华横溢的编程人员、开发人员和系统工

程师共同开创的。

Don Chamberlin、Don Haderle 和 Pat Selinger 就是上述功臣中的几位。这三位专家在 IBM 数据管理解决方案的领导和开发方面的地位是不可替代的。很难想象如果没有他们, DB2 的 20 年发展史会是什么样子,但当前和未来的挑战很少有时间让他们来回忆过去。

最近, DB2 杂志编辑部主任 David Stodder 利用 DB2 20 周年纪念的机会,

与这三位数据库先驱进行了一次非正式讨论,讨论的范围涉及 DB2 的过去、现在和未来,这次谈话的主题是 DB2 发展的“转折点”:改变了客户软件使用方式的那些关键技术进展。从一开始,关系数据库技术的一个关键目的就是帮助客户显著降低企业重要应用程序和决策支持系统的成本,缩短上市时间。这一目标持续至今。

IBM 名士 Don Haderle、Don Chamberlin、Pat Selinger 小档案



Don Haderle 是 IBM 数据管理解决方案部数据管理的 CTO 兼数据库技术副总

裁,他以前是 DB2 的架构设计师,与 IBM 研究部门在工程 and 开发方面紧密协作。Don Haderle 仍在负责指导 DB2 的开发,致力于研究联邦数据管理技术和扩展 DB2 对数据分析、对象模型和其他新技术的支持。



IBM Almaden 研究中心研究员 Don Chamberlin 是众所周知的 SQL 数据库语言创始人(与 Ray

Boyce 合作),他是《A Complete Guide to DB2 Universal Database》一书的作者(Morgan Kaufmann, 1998 年)。Chamberlin 担任万维网协会(W3C)XML 查询工作组的 IBM 代表兼 XQuery 语言规范编辑,他于 1997 年当选为美国国家工程院院士(NAE)。



Pat Selinger 是 IBM 数据管理解决方案部数据管理架构和技术副总裁。Pat Selinger 是 System R 小组的主要成员,因其在查

询优化方面的创新工作而著称。Selinger 是 Almaden Research Computer Science 部门的负责人,主持创建并成立了 IBM 数据库技术研究院(DBTI)。Selinger 还负责 IBM 为所有数据提供集成式访问的计划。在 1999 年,由于其对关系数据库技术的杰出贡献,她当选为美国国家工程院院士(NAE)。

DB2 杂志: 2003 年是 DB2 在 MVS 平台上第一次发布 20 周年。就你们看来, DB2 演化和客户采用这一软件的过程中, 重大转折点有哪些?

Selinger: 我们必须从 DB2 之前 —— System R 原型的某些转折点开始。波音是我们较早的重要客户。我们向他们提出: “这就是原型关系系统, 你认为可以用它来做什么事情吗?” 波音回答道: “是的! 我们可以询问以前不能询问的问题。这太奇妙了!” 这是在真实环境中使用 SQL 关系数据库的一次试运行, 波音认为它是一次重大突破。

我记得有一天坐在飞机上休息时, 突然想到: “我将乘坐一架使用我开发的软件来计算重量分布的飞机!” 那时, 我真正认识到了这项工作的重要性。

DB2 杂志: 波音的 System R 是基于 IMS 实现的吗?

Selinger: 不是。IMS 虽然扮演了重要角色, 但波音希望使用关系技术来进行工程设计, 以便在整个设计和制造飞机的流程中分析数据, 从而可以计算出重量分布之类的重要参数。

Haderle: 1976~1977 年是研究和开发部门之间形成密切关系的重要阶段。Norris Van den Berg 领导 IBM 研究部门朝着客户产品项目要求的方向发展。这极不寻常, 因为在此之前研究部门从不与客户打交道。这个改变是研发部门与客户开始共同协作的关键, 比如 System R 就是与波音、Pratt & Whitney 和其他公司开展协作的结晶。

在开发 System R 的同时, 我们还开发了一系列查询工具, 比如 Query By Example (QBE) 和 VS/Query (查询管理工具, 或 QMF)。客户为能够使用这些软件而高兴不已, 因为它们为客户提供了一个与 System R 相匹配的询问问题的环境,

System R 成了能够回答问题的数据库系统。在秉承这一传统的基础上, IBM 在 VM/VSE 上引入了 SQL/DS。这种操作系统主要用于决策支持系统 (DSS) 和商业分析。在美洲银行, 你仍将看到在 MVS 上运行的交易处理系统以及围绕 VM 构建的 DSS。因此, 它很自然地适用于对 System R 所作的改进以及查询工具, 并把它转向 VM 的 DSS 市场。

现在我们再回到大型机和 MVS 操作系统。MVS 环境忙于应付事务处理和批处理这类操作性行为, 只给 DSS 留出了极少的空间。当我们在 1983 和 1984 年开发了 DB2 之后, 我们的目标是同时运行三类任务, 但事务处理仍是主体。当时 DB2 的性价比不能与使用相同硬件构建的 IMS/DL1 相提并论, DB2 的事务处理成本是 IMS/DL1 的 3~4 倍。

1968 年上市的 IMS 已经成熟。它可以提供事务处理的高可用性, 虽然还没有达到 24*7 的水平, 但在当时, 能够达到那种程度, 用户已经非常满意了。

DB2 杂志: 是什么导致了 IMS 的衰落?

Haderle: 软件故障。内存是有限的, 你有可能因为耗尽内存而不得不覆盖掉前面的内容。当事务中的多个操作相互关联而又没有被恰当地隔离的时候, 程序错误将导致整个系统崩溃。

事实上, IMS 上任何系统管理任务都将触及 MVS 的内核, 并且由于错误的指针使用, 很容易扰乱整个数据结构。到 20 世纪 70 年代末, 很多企业都遇到这样的困惑。那就是虽然可以比较容易地添加硬件, 但是却不得不付出成倍的工作量来进行系统管理, 这是因为操作系统本身并没有带来任何益处! 人们希望系统一年 365 天都能运作, 这需要进程之间有严格的隔离机制, 而在那个时代, 我们并没有做到。

Chamberlin: 在那些日子里, 使用相同的语言和相同的系统来处理事务和提供决策支持是一种非常激进的想法。行业的标准做法, 是尽量使用完全不同的应用程序, 并根据不同的标准对它们进行优化。借助于关系系统, 我们正在努力证明可以集成这些不同的应用程序并实现数据共享。

Haderle: 是的。即使 DB2 针对于操作性和 DSS 应用程序而设计, 但事实是当它进入到市场时, DB2 还没有成熟到能够长时间支持高容量事务处理系统需要运行的大量并发进程。迄今为止, 前四年的大多数客户主要使用 DB2 来运行 DSS, 大约有 30%~40% 在处理事务。

为什么当时客户应用 DB2 来处理事务呢? 这是因为他们可以使用关系系统来迅速开发和发布应用程序。它能加快产品上市的速度。客户会选择能够使产品尽快上市的系统, 而不是事务处理成本或性能需求将备受关注的系统。

在 DB2 面世的头四年内, 我们遇到的问题: 如何使 DB2 能够经济高效地处理事务? 如何能够使其强大到能够提供高级别的并行可扩展性? 1988 年推出的 2.1 版本在性能方面进行了显著改进; 在成本方面, 我们从比 IMS 高 3~4 倍下降到只高出 30%, 而且这个差距还主要是由于 CPU。客户需要在发布新应用程序的速度和运行成本之间做出选择。20 世纪 80 年代末, 我们密切关注着随时间不断变化的硬件价格曲线, 持续的改进将成本差距缩小到大约 5%。在 1988 年, 我记得《Computerworld》使用了这样一个醒目的封面标题: “关系数据库步入黄金时代!”, 这一年也成为了关系数据库的转折点。我们看到, 到 80 年代末, 部署 DB2 来处理核心事务的客户大幅攀升。从那个时候开始, 我们可以降低 CPU 成本, 显著改进并行运行机制, 包括行

级锁 (row-level locking), 这都得归功于研究部门 C. Mohan 所做的工作。

DB2 杂志: 那么下一个转折点是什么?

Haderle: 依我看来, 应该是分布式数据库。在 20 世纪 80 年代末和 90 年代初, 客户机/服务器架构开始浮出水面, 它可以演变为各种各样的网络模式。此时 IBM 推出了分布式关系数据架构 (DRDA), 它支持同构和异构的分布式计算。

在分布式数据库之后迅速出现的另一个转折点是并行处理。在 1989 年, 它开始与分布式技术融合。

Chamberlin: 在 20 世纪 90 年代初, 我们主要扩展了 DB2 支持运行的平台范围, DB2 发展到开放系统领域, 这也是一个转折点。

DB2 杂志: 这同样也是 IBM 的另一个巨大转变吧?

Haderle: 是的。在 1986 年以前, 我们一直隶属于存储处理器分部。IBM 原则上没有软件业务。我们只开发处理器或硬件产品。大约在 1986 年, IBM 开始考虑把软件作为自身的一项业务。为了使这一点可行, 我们必须支持开放系统。在 1992 ~ 1993 年, 我们使 DB2 做到了这一点。

DB2 杂志: 对象/关系技术是一个转折点吗? 即使它并未在市场上真正站稳脚跟?

Haderle: 是的。对象/关系技术必须利用数据库的可扩展性, 也就是如何提供一个可以添加你或其他人开发的新功能的平台。虽然对象/关系技术并没有在市场上得到广泛应用, 但它成为了我们实现多个小组协作、添加功能和保持整体向前发展的基本的数据库引擎基础结构。在 1995 年和 1996 年, 我们开始第一次在 DB2 和 DataJoiner 中实现来自 Garlic

研究项目的技术。

Selinger: 触发器、BLOB、用户自定义函数 (UDF), 所有这些都伴随着对象/关系技术的浪潮一同出现。现在每个人都可以充分利用这类技术, 因为我们已经将它们整合到数据库引擎中。我认为对象/关系技术与人工智能 (AI) 有几分类似: 一旦 AI 开始发挥作用, 人们也就不再称它为 AI 了。剩下的是那些发展越来越缓慢的功能了。但我认为, 正如我们看到在对象/关系数据库结构中引入了 EJB 和其他新技术一样, 我们将发现有越来越多雄心勃勃的技术得到应用。

谈到数据可用性, 我们还必须提到数据共享和多系统功能——联邦和 Parallel Sysplex。大约在 1994 年, 我们在大型机和开放系统平台上引入了这些功能。我认为这类技术也是一个转折点。

Chamberlin: 我想我们现在正在经历信息集成的转折点。回顾过去, 我发现两种截然不同但同时存在的技术潮流: 一种专注于处理结构化、同构数据, 另一种专注于处理非结构化的数据。如果你正在处理银行账户, 所有记录就都是同构且结构化的。今天, 我们可以用数字媒体来展示书籍, 但书和银行帐户不一样, 每本书的结构都是不相同的。

因此, 历史上各种各样的软件都围绕这两类不同的数据技术潮流来发展。现在, Web 正在把它们融合在一起。我们需要通用的工具来同时处理这两类数据。这正是我们目前面临的挑战。

DB2 杂志: 早期, 我们讨论了使用能够处理 DSS 和操作、事务处理行为的 SQL 语言是一种激进但可行的想法。现在 SQL 是否面临类似的考验, 即是否能够支持结构化和非结构化的数据系统?

Selinger: 是的, 这就是为什么你看到 Don Chamberlin 在 XML 查询语言 (XQuery)

委员会上强烈建议 IBM 来定义这一能够处理这两类数据的下一代查询语言的原因。展望未来, 客户将希望引入更多的数据, 而不仅仅是常规的结构化信息。

Haderle: 我同意 Don 的说法。正如我们所说, 我们正处于这一转折点的浪尖上。它“诞生”了虚拟数据库概念以及支持这一概念的联邦技术。现在面临的主要问题是存在一组分散的数据, 客户希望引入这些数据并将它们全部链接在一起, 我们该怎么做呢? 答案是提供虚拟层。

Selinger: 在开发了所有 SQL 创新和关系方面的技术之后, Don [Chamberlin] 花费了几年的时间转而进行文档管理的研究。我们认为这是他一次很大的转变, 但我们当时并不知道这一切工作都将殊途同归。

Haderle: 在 20 世纪 90 年代初, 我们对 Don 说: “Don, 这是个问题。”他说: “是的, 我知道这是个问题。”我们问他: “SQL 是否是我们打算驾驭的工具? 它是否有足够的表达力?” Don 说: “不。我需要不断地完善它。”因此, 他成了 XML 的拥护者, 并提出了这样一种理念: 我们需要通过现有的基于 SQL 的应用程序和基于 XML 的应用程序来提供数据访问。Don 一直在研究 XML 数据模型, 它是创建虚拟数据库层和提供允许人们获得无数信息的信息实用程序的核心。

在 DB2 XML Extender 的帮助下, 我们才真正开始了这一过程。XML Extender 实现了一个 Native XML 数据库, 它可以将 DB2 封装起来。在不受太多关系规则约束的情况下, 我们开发出了越来越多的功能, 它将提供 XML 模式、XML 查询和其他工具。但是我们还有可以用于支持 XML 的关系功能——我们可以整合 XML 和关系功能, 同时提供这两种技术的特性。

之后, 我们的重点转移到联邦技术

上来。Pat 主攻元数据，它将使我们能够查看不同数据源和数据结构发送的信息并无缝地对其进行联合。对元数据的理解可以帮助我们开始研究描述数据在不同数据源环境中的语义规则，以及如何能够无缝地对它们进行联合。这就是用户所需要的，不是吗？我们正在为这个问题而努力，我们把这种技术叫做“信息集成”。

Selinger: 你可以这样来考虑：SQL 和关系系统只需要你指定希望从数据中获得的信息，而无需关注数据在物理上是如何存储的，是否对某些数据进行了索引。SQL 和关系系统在物理数据存储上提供一个透明的层次。同样，信息集成为你提供另外一个透明的层次——这次是在数据源的位置和语言上。信息集成能让你上升一个层面并使你能够与分布的异构数据源交互，就好像它们都在同一个位置，都使用 SQL 和 XQuery 语言一样。

Chamberlin: 你必须以一种截然不同的方式来处理元数据。在纯粹的关系数据库中，数据是同构的——每个银行账户看起来都是一样的。你可以归纳出描述银行账户结构的元数据并将它存放在目录中，而无需为每位账户复制它。但是如果数据是异构的，例如一个数字图书馆，因为每个对象都必须进行自我描述，所以元数据必须与数据密切结合。由于每个对象都是不相同的，你不能归纳出数据的描述并把它放在一个公共位置。元数据与数据是混合在一起的，数据模式也是动态的，这样，系统要存取一组非常不同的数据所使用的查询语言必须能够同时查询数据和元数据。

Haderle: 在使用 XML、XQuery 和联邦技术时，你必须提供非常强大的复制功能。即使部署了联邦系统，并不能保证你总能够通过网络来获取到信息。为了使其能够正常运行，你必须将某些信息保

留在本地。我们如何能够实现数据的缓存？我们如何实现自动复制？我们如何设置不同的模式来保证事务更新、信息验证和异步连接的一致性？这都是我们正在研究的课题。我们正处于这一转折点上。

DB2 杂志: 在 V8 中，将 *Content Manager* 整合到 DB2 似乎是这一演变过程中的关键一步。

Selinger: Content Manager 为我们带来了一次重要的转变。这种强大的内容管理应用程序能够保存包括文件夹、版本在内的所有数据库不能理解的各种信息的语义和功能。这一切都将基于 DB2，关系数据库管理系统在 Content Manager 中可以起到类似于图书馆中卡片目录的作用。

在 DB2 V8 中我们所做的工作是在存储过程和 UDF 中实现相应功能以将它们集成到数据库中，然后就可以通过数据库提供的特性来使用这些功能。例如，如果我们需要，我们可以并行运行它。这就是目前我们将功能添加到 DB2 中采用的方法。我们正在致力于将更多的语义融合到数据库引擎中。

Haderle: 这就是我们目前的途径。坦白地说，如果 DBMS 不能够实现版本管理和其他内容管理功能，开发商将基于 DB2 来编写自己的内容管理查询系统。开发商必须依赖 DB2 来构建自己的查询系统的原因是因为他的客户需要查询与时间相关的或者其他的一些特殊问题。通常的情形是，开发商构建的层面会变得越来越复杂，直到他将几乎整个查询系统都建立在我们的查询系统之上。接着，开发商开始担心并行机制和其他一些因素，如果脱离了我们的系统，他可能需要 200 位员工来开发本来 DB2 能够完成的功能。

因此，我们的工作添加越来越多的语义和结构知识，从而使内容管理系

统可以充分利用数据库并最大限度地减少单独构建一个完整查询系统的需求。如果我们不能把某些语义和功能添加到数据库引擎中，结果又将如何呢？借助于我们的可扩展性，如存储过程和 UDF，我们能够让内容管理系统将其他的功能加入到我们的结构中并同样能够享受查询编译、并行机制等特性带来的优势，而无需基于 DB2 来另外编写一个层面。

Chamberlin: 我认为有一些对象/关系的观念正在奏效，它们在数据库层为我们解析复杂对象的语义提供了良好的基础。

Selinger: 是的，通过提升数据库能够理解的抽象层次，可以减少上层应用程序的规模。这就是这场游戏的主题：减少员工工作，从而人们可以加速应用程序的交付和降低成本。你可以腾出更多的人手来开发下一个应用程序并使企业更具竞争力。

DB2 杂志: Starburst 优化器的引入是否使得强大可扩展性成为可能？与 Starburst 一同推出的新型查询重写和查询编译技术为 DB2 引擎增加了更多的智能。

Selinger: Starburst 是一个完整的系统——一种全功能的数据库系统。我们的口号是“现在我们已经走到了这一步，我们知道我们正在做什么，让我们从头开始，设计一个可扩展的系统。”我们希望实现高度的可扩展性和优化，这将对未来的发展起着决定性作用。Starburst 融合了对象/关系理念，以及可扩展性和可扩充性，并开始具有以查询重写和新型编译为代表的自主功能。Starburst 使 DB2 在开放系统上实现了功能上的显著跳跃。

Haderle: Starburst 是 DB2 成为开放系统的巨大转折点。但当我们谈到 20 世纪 90 年代中期的转折点时，我们有点不负责任，因为我们没有提到数据挖掘。我们在 20 世纪 80 年代后期引入了查询并行机制，它使复

杂的查询能够以非常快的速度完成。至于数据挖掘，我们的口号是：“现在要致力于我们一直试图解决的业务问题。” Rakesh Agrawal 在研究方面的初期工作为我们提供了解决问题的其他方式。

最近，使用 DB2 可扩展性，我们已经将这一功能整合到数据库引擎中。用户可以规定不同的数据挖掘功能；数据库引擎使它们能够以更低的成本高速运行。

DB2 杂志：这样的结果是，最初原本需要数据挖掘方法的大量业务问题，实际上可以由更传统的数据库查询来解决，对吗？

Chamberlin：是的，使数据挖掘极具魅力的是你不知道你正在寻找什么。SQL 善于回答事先精心确定的问题：但是如果你不知道问题是什么，结果又会怎样呢？你可能有大量的数据，但你希望找到不寻常的信息，而你又不知道“不寻常”真正意味着什么。这就是极具挑战性的问题，数据挖掘功能对应这类挑战效果显著。

Haderle：数据挖掘功能与其他功能（如并行机制）的结合，形成了 DB2 的另一个转折点。与谈论如何能够拥有更好地处理事务的基础设施不同的是，商务智能开始成为我们的关注点，也就是，我如何能够从信息中获得智能并解决遇到的商务问题？我们想要解决的问题不再只是询问：“我们在数据库中有多少位秃头？”而是：“我想销售假发；谁是最合适的客户？”没有数据挖掘，你可能永远不会知道，答案与快要秃头的人无关。

DB2 杂志：现在我们开始谈谈 SQL——从研究到开发的关键进展。考虑到它的重要性，我们可以将整个会谈都围绕这一主题展开。Don，你如何看待它的演变？

Chamberlin：在很早的时候，我们把 SQL

视为向新用户（非编程人员）开放数据库的方式，他们不是受过培训的计算机专业人士。我们预计人们希望一种更自然、新奇和更少程序化的方式来询问问题。

在相当大的程度上，我相信 SQL 可以做到这一点。它降低了应用程序开发的培训要求并加快了产品推向市场的速度。或许我们未曾想过的是 SQL 将如何嵌入到所有不同的宿主语言环境中——我们把它作为一种单独的事情来看待。但是现在它嵌入到宿主语言中，用于存储过程，与 Java 级界面协作，等等。这么多年来，当看到 SQL 强大到可以应用于多个领域，是件多么令人愉悦的事情啊！

使用非程序化语言，如 SQL，你将需要增加优化方面的前置成本。为了将钱存储到银行账户中，你编写了 SQL 语句并通过优化器来运行，这将增加你的成本。System R 有待解决的其中一件事是如何进行优化、保存生成的优化包并在无需再次增加前置成本的情况下经常运行它。为了做到这一点，我们不得不记录相关性——优化的程序包中包含的假设：“对这一账户使用这一索引。”

这种通过非过程化查询，经过一次优化就可以在以后的运行过程中享受优化带来的好处，这一理念使得使用同一种语言来处理事务和提供决策支持成为可能。我想这大概就是我们在那些日子里实现的重大突破吧！

DB2 杂志：随着我们向开放的 XQuery 标准发展，我们是否面临相同的挑战，使得很难去制订 SQL 标准？

Chamberlin：W3C 有很强的推动力。IBM 和其他数据库厂商在 XQuery 工作组都有代表，全面参与了标准的制订。客户也要求我们密切协作。

Selinger：与 SQL 标准的体验相比，两者的不同之处在于：XQuery 是在厂商实

施的同时来定义；而大多数厂商是先开发它们自己的 SQL 版本，然后才开始制订标准。如果以在标准制订期间语言变动的程度作为衡量标准，XQuery 发展的速度要快于 SQL 标准。Don，你不是说 XQuery 仍是瞬息万变吗？

Chamberlin：是的，但我希望 XQuery 的主要框架会比较稳定，现在已经基本上井然有序了。想像一下在同一间办公室内由 30 位极其聪明然而雄辩的人来设计一种语言，他们有不同的背景和各自的拥护者，你就可以知道 XQuery 是多么有趣。你可以理解为什么可以在短期内完成这项工作。但我认为在这段时间内，我们达成了共识，这使得成果将更完美。

DB2 杂志：现在我们来谈谈未来的一些想法，以结束这次讨论。在接下来的几年内，DB2 将如何发展？是否将推出 DB3？

Selinger：IBM 投资的一个关键点是自主运算。我们在 V8 中提供的功能仅是以后推出的产品的前奏。我们设想会有更多的数据库配置参数将实现自动化，DBA 将非常自信地说：“交给你们去处理吧！”与在操作系统领域关注数据页面的交换以实现虚拟内存的概念相同，我们希望 DB2 能够随着工作量的变化重新配置内存和资源。当你从夜间的批处理转变成早上的事务处理时，系统将调整资源来控制性能和可扩展性。当假期来临时，系统为繁重的事务处理作好了准备；当一月份的计费周期来临时，系统准备提供决策支持。

自主的理念将从单机系统扩展到大规模并行处理器系统，乃至处理器网络中联邦、异构的系统。动态配置调整将超出传统数据库的范畴，扩展到负载管理、调度等。与这一点的结合将在自我管理、自我修复技术方面具有更多的优势，在 V8 中我们已经看到了这种技术的苗头！

Haderle: 我并不认为“DB3”的意义能与DB2所代表的从层次型到关系型的转变相提并论。我们不希望用一种新的语言和某些完全不同的功能来改变数据模型。今天使用SQL的人明天也将使用SQL, XML只是能够帮助我们解决更多的问题而已。

在接下来的10年内,我们将继续密切关注性能、可用性、可靠性这些对于Charles Schwab & Co.等企业来说是激动人心的特性,因为像这样的公司平均每天可能完全更新整个数据库高达8次,而且系统将全天候运行事务处理任务。在这种情况下,我们一直在努力解决的问题将进一步扩大,这将是我們一直努力的目标。

我们看到,在作为整体IT预算的一部分——人力成本持续增加的同时,仍然存在严重的知识工人的缺乏。这就是我们必须解决的问题,而自主运算是我们解决这个问题的重要手段。这也是关系查询编译器所要完成的工作;换句话说,想想在编写和进行特定查询可能使用的无数种方式,为什么不告诉系统你想要的结果而让系统替你把它演算出来呢?这就是我们在节约人力资源方面的首要工作,也是在接下来的几年内你能看到DB2带来的变化之一。

另一个主要挑战是企业购买了越来越多的资产,这意味着企业有数千(或者说成千上万)个数据源,而且这些数据源的形式各式各样。企业将如何处理他们拥有的整个信息资产?这其中有一个挑战是确保系统整体有效运行,另一个挑战是如何为决策者提供有用的信息。为了解决这些问题,并且基于我们用于描述现有对象的联邦方法和可扩展性特性,我们制订了信息集成方案。强大的复制功能将通过联邦系统来回传送数据。Pat所做的元数据工作有助于为信息资产提

供更全面的理解和集成。

最后的结果是信息资产的虚拟:我如何能够得到整个资产的统一视图,或者是我资产的一个聚集?我如何确保一流的信息质量并及时提供信息?我们把这看作是中间层,但这个层面不仅仅是ODBC或JDBC存取,而是真正的信息实用程序。

自主将是至关重要,因为有大量复杂的系统需要进行整合、描述和管理。我的意思是你会部署不同和新型的事务处理模式,这导致你将不得不执行松散型事务处理,而不能运用紧密的事务处理模式。

我们把网格计算视为重要的虚拟化驱动因素。如果我部署了一组异构的数据库和处理器,并且我需要将信息输入到所有不同种类的系统中,我需要知道它的规则和属性、安全模式、目录基础设施等。网格在科学领域已经很重,而且它在商业领域也将变得重,因为它有助于实现计算环境虚拟化。

在传统的现有数据库领域内,我认为我们将看到集成与编程的联姻。我的意思是我们将考虑客户正在努力解决的问题:什么是客户将使用的应用程序模型?什么是与这类模式相关的数据结构?我们把Java 2 Enterprise Edition (J2EE)视为强大的开放系统容器模式,因此,我们必须把J2EE服务与数据库相集成。但问题是:什么是一个行业中使用的主要模板?我们必须做什么以使它们与数据库相关联并正常运行?我认为模板有助于推动我们将在数据库中完成的大量工作,以便更迅速地把新应用程序推向市场。

Selinger: 所有这一切都意味着对WebSphere技术的紧密结合,因为它们可能是运行期间应用程序开发人员和用户的首要结合点。客户不希望能够在WebSphere层面上解决的问题还要在DB2层面上再重复同样的事情,客户并

不关心XML、关系理论或者其他技术方面的变化,他们只是希望所有的系统行为能够自始至终地自动表示。

DB2 杂志: Don, 你通过SQL引发了第一次革命,现在,你正在用XQuery来进行第二次革命,是吗?

Chamberlin: 是的,你知道我一直在考虑这件事,我认为全球每隔25年需要一种新的查询语言。能够经历两次革命是件令人非常愉悦的事情。DB2将同时支持SQL和XQuery,这是正确的方向,它充分体现了我们正在努力完成的信息集成理念。

Haderle: 你是否认为,考虑到目前占统治地位的互联网链接导航方式,Charles Bachman [网状数据库模式创始人]所说的“我最终将击败关系结构!”是否会成为现实?

Chamberlin: 全球有许多超级链接,不是吗?我经常在大学内演讲“数据发展简史”,在演讲中我把Web称做“Bachman的复仇之剑。”

Haderle: 我知道IMS的人在说:“我早就告诉过你会这样了。”

DB2 杂志: 我们的谈话到此结束,谢谢诸位!

(本文由IBM developerWorks网站供稿,原文链接http://www-900.ibm.com/developerWorks/cn/dmdd/library/techarticles/mag_0307stodder/stodder1.shtm)