21 世纪的决策支持系统

George M. Marakas 著

朱 岩 肖勇波 译

清华大学出版社

（京）新登字 158 号

书　　名：21 世纪的决策支持系统
译　　者：朱　岩　肖勇波
出 版 者：清华大学出版社（北京清华大学学研大厦，邮编 100084）
　　http：//www. tup. tsinghua. edu. cn
责任编辑：江　娅
版式设计：肖　米
印 刷 者：北京国马印刷厂
发 行 者：新华书店总店北京发行所
开　　本：787 × 1092　1/16　印张：24. 75　字数：568 千字
版　　次：2002 年 11 月第 1 版　2002 年 11 月第 1 次印刷
书　　号：ISBN 7-302-05765-6/F・430
印　　数：0001 ~ 5000
定　　价：43. 00 元

译 者 序

决策是为达到某一目的对若干个可行方案进行分析、比较、判断，从中选择最佳方案并予以实施的过程，是各种矛盾、各种因素相互影响最后平衡的结果。它随着人类社会活动的发展而产生。长期以来，决策主要是依靠人的经验，属于经验决策的范畴。随着科学技术的迅速发展，社会活动范围扩大，大企业、大工程的出现，国际关系日益复杂，领导者单凭个人的知识、经验、智慧和胆略来做决策难免出现重大失误。在这种形势下，经验决策逐步被科学决策所取代。

在信息社会，决策水平决定了企业能否在瞬息万变的商业竞争中保持自己的优势。面对决策环境的日趋复杂，我们该如何制定决策？有哪些手段可以为决策提供有力的支持？如何建立有效的决策机制？围绕这些问题，《21 世纪的决策支持系统》这本书给出了答案。

20 世纪 70 年代初美国的 M. S. Scott Morton 在《管理决策系统》一文中首先提出决策支持系统 (DSS) 的概念。到了 20 世纪 80 年代 DSS 迅速发展成为一个依托计算机科学、运筹学、管理科学等多学科的新型学科。从 DSS 概念诞生到 20 世纪 90 年代初，DSS 的理论体系还不完善，在系统开发方面，功能较全的实用 DSS 还没有问世，已实现的 DSS 都是在某些功能上较强，而缺乏另一些功能。20 世纪 90 年代初提出了数据仓库和数据挖掘等决策新技术，为决策支持系统的研究和发展开辟了一条新的方向。

本书就是遵循这样一个发展脉络，全书 16 章分别对决策者、决策过程、决策建模、专家系统与知识工程、数据仓库和数据挖掘、决策支持系统的设计开发等问题进行了深入浅出的介绍。本书的内容具有较强的实践性，通过提供大量的案例讨论，向读者强调设计和开发之后如何来应用决策支持系统。书中还结合了目前一些优秀决策支持系统相关教科书中的内容，同时增加了最新决策支持技术的介绍，如数据可视化和数据挖掘等。因而，本书内容丰富，包含了现代决策支持技术的方方面面，难度适中，应用性强，是一本相当出色的决策支持系统教材。

正如作者所言，本书是面向那些立志于从事与公司管理决策相关的职业的商学院学生，主要使用对象是高年级本科生或者研究生。在学习 DSS 课程之前，学生最好已经完成了管理信息系统 (MIS) 的介绍性课程的学习。当然由于书中内容设计了许多系统分析和设计方法，读者最好学习过“系统分析和设计”课程。

决策由于其较高的不确定性，一直吸引人们去探索好的决策支持方法。现代企业的

决策能力是企业生存的关键。译者衷心希望这本书能够帮助大家对决策支持技术的学习，并希望本书能够为普及决策知识、提高我国企业的决策水平尽到绵薄之力。

在本书的翻译过程中，清华大学经济管理学院 81 班的同学们做了大量的工作，王卉、刘佳、张楠、王小龙等同学做了大量的初译工作，在此对他们的工作表示由衷的感谢！同样感谢清华大学出版社的所有编审人员，是他们才使得本书能够顺利翻译出版。

由于水平有限，兼之翻译时间仓促，译文中还有很多不准确的地方，欢迎读者不吝指正。

朱岩

2002 年 6 月于清华园

前　言

决策是我们日常生活中每天都会遇到的活动。有关决策值得欣慰的一点是，我们生活中的绝大部分决策都是惯例性的，相对比较容易制定。比如，决定穿什么样的衣服，早餐吃些什么，看什么样的电影，以及在杂货店购买什么样的食品，都是我们面临的日常决策。然而，我们在选择职业时所面临的决策就远远不同了，不会因为能够决定上班穿什么样的衣服，就能得到薪水。作为管理人员，我们通过处理诸如是否在某项即将出现的技术上投资（这有可能给公司带来很大的竞争优势）之类的决策，来体现我们对组织的价值。这些决策还包括如何安排项目中有限的人力资源、确定目前和将来市场中最看好的产品等。换句话来说，真正重要的决策往往是很难制定的，需要有大量的信息以及各种各样的决策支持。这正是本书所要讨论的话题：制定和支持管理决策。

概念和目的

本书从感知过程和决策制定的角度提供了决策支持系统（DSS）课程教学的一个基础，主要内容集中于强调管理应用和决策支持的相关技术。

Gorry 和 Scott Morton（1989）曾经非常经典地定义了 DSS：

> 决策支持系统结合个人的智能资源和计算机的能力，来提高决策的质量。它们包含一个基于计算机的支持系统，为管理决策制定者处理半结构化的问题。（P. 60）

我总觉得这有点不太直观，这个定义是从用户的角度写出的，然而，商学院有关 DSS 的教科书一般是从设计者的角度写的。正是因为如此，我在写本书的时候，主要将重点放在帮助学生全面地理解一个 DSS 提供“支持”的方面。本书的内容具有很强的现实倾向性，强调了设计和开发之后的应用和实施中的各个主题领域。将来的经理并不需要懂 DSS 的设计，因为那属于计算机科学家和系统分析员的工作范围。然而，他们需要那些与决策支持技术的有效和战略应用相关的技能，以提高问题识别和相应的解决方案的质量。本书中采用了各种学科用户/管理员的方法，面向 21 世纪的决策和支持决策所必需的技术。决策制定和感知过程所包含的范围包括决策制定的模型、偏好和启发式经验、创造力提高、决策策略、模拟和发现等。书中结合了目前一些领先的决策支持教科书中的最好的内容，同时增加了一些前人没有论及的话题（最主要的是数据可视化和数据挖掘）。

总而言之,本书是基于我个人的一种想法所促动的,即:有必要将我们决策制定的知识与决策支持技术的应用结合起来。对于今天和将来的管理人员来说,应用和理解 DSS 的使用比 DSS 的设计要重要得多。

适用对象

本书是面向那些立志于从事公司(技术的主要用户或者属于技术驱动行业的公司)的管理职能的商学院学生——也就是说,本书适用于商学院的所有学生。主要使用对象是高年级本科生或者研究生。这种类型的课程一般在四年制大学和许多公共学院中开设。理想状况下,在学习 DSS 课程之前,学生应该已经完成了 MIS 介绍性课程的学习,并且有过一个学期系统分析和设计的学习。此外,学生在商业课程中学习得越深入,本书的决策制定的视角将越与他们相关。除了面向学生以外,本书的许多章节还可以给从业者在他们的日常决策活动中提供较好的参考作用。

本书的组织结构

本书中采用了许多传统的教学方法。在写作风格上,尽量在专业的和谈话式的方法上找到一个平衡。贯穿本书始终,围绕介绍的概念,采用了很多图形和示例。在每一章中都包含一个介绍性的小案例,小案例的内容与本章即将介绍的主题相关。在每章的结尾,对本章的关键概念进行了总结,并且提出了一些复习题和讨论题。在本书的最后,列出了参考文献。本书各章中各部分的组成如下:

学习目标

在每章中从表现和行为的角度列出了本章的学习目标。也就是说,目标中阐述了学生在阅读完本章之后,应该能够做什么,应该理解哪些内容。

小案例

所有的小案例都源自于实际生活中的情景,主要给学生一个直观的概念,初步理解本章将要讲解的内容是什么。此外,每个小案例中都列出了案例的来源,这样学生(个人或者小组)如果想进一步探讨出现的状况,可以采用各种研究工具展开调查和研究。

图表

每章都包含一些精心设计的图形和表格,以帮助学生更好地理解材料。在正文当中,一般都对表格的内容进行了具体的阐述。

叙述性的小插图

为了进一步解释与决策制定过程相关的某些概念,我们采用了一些叙述性的小插图。书中,通过构造一些场景,使得学生不仅能理解到讨论的技术是如何应用的,而且也能将它与特定的环境或者上下文联系起来。

复习题

每章都包括 10～20 道问题，用于测试学生对章节内容的理解程度。每个问题的答案都能在正文中找到。在本书对应的网站上，我们提供了问题的示例答案。

讨论题

在每章的最后，我们还列出了几道讨论题。它们是对本章内容的一个扩展，可以让学生进一步地思考，以更好地掌握本章的内容。这些讨论题可以用在开放式课程讨论上，并且可以作为个人或者小组的一个小项目。

WWW 网站

本书有个网站，网址是 http：//www. prenhall. com/marakas。在网站中，包含 4000 多个与 DSS 相关的网站的链接。为了更好地加强学习过程，这些链接在方式上与本书的组织一致，并且根据它们与各章节的相关程度进行了分类。这样，通过使用与某些章节相关的因特网资源，可以根据需要对各章节的内容进行扩展，以帮助学生更进一步了解某些特定的主题。

章节概览

第 1 章：决策支持系统介绍

第 1 章介绍了决策支持系统的概念，并解释了其组成部分。同时，还简要介绍了决策支持系统的发展历史。此外，本章还讨论了目前使用的各种类型和类别的决策支持系统。

第 2 章：决策者

本章的焦点是决策者。我们认为决策者是 DSS 环境中一个必不可少的部分。讨论的问题包括决策风格、决策的效力、可以提供的支持的类型等。

第 3 章：决策的过程

本章的重点是决策及制定决策的过程的概念。介绍了基本的决策理论及相关问题，包括受限制的理性、偏好和启发式规则及基本认知过程等。此外，还介绍了效果和效率的概念，这是在一个决策中经常容易混淆的两个概念。

第 4 章：组织中的决策

本章的焦点是组织。这是因为组织是绝大多数管理决策制定所处的环境。理解组织的一些基本概念，并且探讨组织的各个方面（比如文化、权力和政策，以及可能遇到的“组织层次”决策的类型等）是非常重要的。

第 5 章：决策过程建模

第 5 章介绍了各种决策过程建模的通用技术，介绍了几个通用决策模型结构的实例。同时，在本章中，讨论了概率以及预测概率性事件的方法，这些概率性事件是在复杂的管

理决策中经常会遇到的。

第 6 章：群体决策支持和群件技术

从本章开始，每章将关注决策支持技术的一个方面。我们首先系统地解释群体决策制定的过程，并且识别和解释了各种类型的多人参与的决策制定技术。

第 7 章：经理信息系统

紧接着 DSS 技术之后，本章将讨论的焦点放在了经理以及 DSS 技术在开发和应用经理信息系统（EIS）的应用的领域里。所涵盖的范围包括 EIS 技术的定义、EIS 的简要发展历史、执行层次的决策和决策者的特性，以及与成功地在组织环境中引入 EIS 相关的各种问题。

第 8 章：专家系统和人工智能

本章首先简要介绍了专家的概念，接下来全面讨论了人工智能和专家系统的各个元素。本章最后讨论了设计、创建和评价一个专家系统相关的各种问题。在第 8 章的附录部分按照年代次序列举了各种专家系统技术。

第 9 章：知识工程与知识获取

第 9 章进一步讨论了专家系统，重点放在知识的概念，以及获取知识和在专家系统知识库中编码知识的方法。

第 10 章：学习机

本章的焦点是人工智能和决策支持系统领域最新发展起来的一些内容。我们详细介绍了模糊逻辑和语言模糊的概念，作为介绍人工神经网络和遗传算法的一个先导。本章附录是这个领域两个流行的软件应用。接下来，提供了一个人工神经网络学习算法的数学推导。

第 11 章：数据仓库

本章和下一章介绍了当今决策支持系统最热门的两个主题 :数据仓库和数据挖掘。第 11 章全面包含了数据仓库的基本概念，并且讨论了若干种商业数据仓库产品。

第 12 章：数据挖掘和数据可视化

紧接着第 11 章介绍的概念之后，本章主要探讨了数据挖掘和复杂的模式提取方面的内容，介绍了在线分析处理（OLAP）及其变形。此外，本章还介绍了挖掘数据的技术、它们目前的局限性以及在数据可视化领域中的应用。

第 13 章：设计与构建决策支持系统

如果不讨论实现系统的策略和所需的工具，决策支持的主题就是不完整的。本章讨论了设计和开发一个现代化的决策支持系统相关的问题。

第 14 章：决策支持系统的实施与集成

第 14 章探讨了有效实施、集成和评价一个组织的决策支持系统所需的活动，重点放

在 DSS 的开发问题上。

第 15 章：创新性决策制定和问题解决

在结束对决策制定和问题解决的深入讨论之前，有必要讨论一下最难以捉摸、但是对于成功的管理决策制定非常关键的一个方面：创新性。本章介绍了创新性的概念，以及增强创新的各种方法，接下来讨论了智能主体和其他一些相关的话题。

第 16 章：21 世纪的决策支持

最后一章展望了将来的决策支持系统。通过回顾当前我们已经取得的发展，本章展望了未来的决策支持系统、专家系统、人工智能和经理信息系统。

致谢

这是我第一次尝试着写教科书。在写作过程中，我遇到了很多困难，也学到了不少东西。如果没有众人对我不懈的帮助，纠正我的错误，回答我的问题，在我产生怀疑的时候鼓励我，我相信我不可能完成这本书的写作。我要衷心感谢那些在我写作过程中给我提供了指导和帮助的人们。

此外，我还对那些审查了我的手稿的人们表示感谢，他们提出了很多改进的意见。他们是：William R. Bruyn，DeVry 理工大学；Ranjit Bose，新墨西哥大学；Fatemeh Zahedi，威斯康星大学；Jerry Fjermestad，新泽西州理工学院；Madjid Tavana，LaSalle 大学；B. S. Vijayaraman，阿克伦大学；Bradley C. Wheeler，印第安纳大学；Russell K. H. Ching，加州州立大学；Tome Wichser，DeVry 理工大学和 Gustav Lundberg，Duquesne 大学。

我想要感谢的人很多，但是如果遗漏了我的好朋友和同事 Brad Wheeler 和 Steven Hornik，那我就太粗心了。你们的友谊和真诚给了我很大的动力。

最后，我要把我最真诚的感谢给予我所有的学生，他们选择了我的课程，帮助我发展了很多观点、实例、解释和内容。你们是我真正的源动力，我将永远铭记在心。

有关作者

George M. Marakas 是布卢明顿市印第安纳大学 Kelley 商学院信息系统专业的副教授。他的教学专长包括系统分析和设计、技术辅助决策制定、IS 资源管理、行为 IS 研究方法以及数据可视化和决策支持。此外，Marakas 是系统分析方法、数据挖掘和数据可视化、创新增强、概念数据建模以及计算机自身效力等领域非常活跃的研究员。

Marakas 从迈阿密佛罗里达国际大学信息系统专业获取博士学位，科罗拉多州立大学获取 MBA 学位。在从事学术职业之前，他曾在银行和房地产行业干得相当出色。他的公司经历包括在 Continental Illinois National Bank 和 FDIC 担任高级管理职位。此外，Marakas 还曾任 CMC Group，Inc. 的主席和 CEO。

在马里兰大学和现在的印第安纳大学任职期间，Marakas 在研究和讲课上都别具风格。他曾多次获得国家教育奖项，他的研究成果在很多核心杂志上发表。

除了学术方面的贡献，Marakas 还是一个非常积极的咨询者，他担当过很多组织（包括中央情报局、财政部、国防部、泽维尔大学等）的顾问。他的咨询活动主要集中于工作流再造工程、CASE 工具集成以及组织系统朝 Lotus Notes 的转换。他是一个 Novell 认证的网络工程师，从 1990 年起，就参与了微软公司的程序测试。Marakas 也是很多专业 IS 组织的积极分子。他非常喜爱高尔夫球，也是一个受过认证的潜水员，是 Pi Kappa Alpha 互助会的成员之一。

目　录

第 1 章

决策支持系统介绍

学习目标

- 了解决策支持系统的定义
- 了解使用决策支持系统的好处及其局限性
- 熟悉决策支持系统的发展历史
- 掌握决策支持系统的五个基本组成部分
- 理解数据和模型管理系统的作用
- 理解决策支持系统知识库的作用
- 理解决策支持系统中用户界面的重要性
- 理解决策支持系统中用户的作用以及使用决策支持系统的结构
- 掌握决策支持系统的分类

小案例

Mervyn's 百货公司

面对着来自折扣店越来越激烈的竞争压力，百货公司不得不对自己进行重新定位，想办法来保护并保持其竞争优势。Mervyn's 是美国一家领先的服装零售公司，拥有 280 个零售商店。它正使用一套复杂的决策支持系统，对信息加以分析，以识别正流行的产品，并辅助快速决策。

虽然说 Mervyn's 现有的系统提供了大量的数据，但是仍然面临着很大的挑战。Mervyn's 需要构造一个系统，将数据进行有效地整合并提取成决策所需要的关键信息。在早期的系统中，数据通路不充分，分析能力也相当有限，这些缺陷严重阻碍了经理们使用那些掩埋在交易数据中的信息财富的能力。

将这个问题综合起来，就是 Mervyn's 急需能够快速访问多达 300 ~700GB 的数据。要在一段适当的时间内在如此大量的数据上进行综合分析，目前现有的绝大多数决策支持技术是远远做不到的。于是，解决的途径要求查询性能优化，以及各种并行软件方案，这对决策支持来说是一个全新的领域。

Mervyn's 也需要标准化业务比较分析。Mervyn's 的销售规划和后勤系统的经理 Sue Little 说，"我们的数据在单位和金钱价值上从来没有匹配过。买主都是从金钱价值的视角看待一切，而存货管理则从单位的视角来看待一切。从来未能将这两种数据轻松地连接在一起。"

为了解决这些问题，MicroStrategy（一家 DSS 开发软件商）和 Mervyn's 一道，使用 DSS 智能主体（DSS Agent，是一种开发应用，由 MicroStrategy 制造并销售），着手开发了"决策者工作平台"（The Decision Maker's Workbench，DMW）。这种 DSS 应用可以进行趋势、性能和库存分析。分析者将能够查看销量的波峰和谷底，以及地区与地区之间的差异。这些信息可以帮助决策者进行库存管理的决策。通过使用均衡多重处理硬件技术和并行查询处理技术，DMW 将处理工作从 1 个小时减为不到 1 分钟的时间，从而帮助 Mervyn's 实现了其"通过实时决策保持竞争优势"的战略目标。

Sue Little 说："DMW 使得我们能够在'原子层'上，从金钱价值或者单位的角度，一同看待各种不同的商品。最终我们能够在苹果之间进行比较。现在我们只花 10% 的时间来收集数据，而 90% 的时间用于采取行动，恰好与以前的时间分配完全相反。"

DMW 应用也结合了多种特性，比如数据冲浪、钻取、智能主体以及报警功能等。Mervyn's 的应用还包括基于地理的例外报告功能，以及类似报纸界面方式提供的业务趋势和例外总结的经理视图等。"决策者工作平台"包括预测能力，允许最终用户通过在查询结果上进行某些特殊的分析，来回答一些"What-if"问题。

决策支持系统 (decision support system,DSS) 的学习并不是关于计算机本身的。虽然说它在 DSS 领域里是不可缺少的,但是应该认识到计算机只是其中的一个组成部分。DSS 的学习实际上是关于“人”——关于人如何思维,如何制定决策,以及他们如何作用和反作用于决策的。DSS 是信息系统 (information system,IS) 领域的一个分支,它要求有可以确认的根源以及一个明确的焦点。DSS 经过设计、构造,用于辅助活动——即辅助决策过程。这里需要提醒的是,DSS 并不能自己制定决策,DSS 的真正目的在于对决策者的决策过程提供支持。因此,在学习决策支持系统的时候,我们需要学习的是人、决策以及决策是如何制定的。

1.1　决策支持系统的定义

一般特性

在这一节里,我们将给决策支持系统下一个明确的定义。首先来看看绝大多数 DSS 应用的共同特性。表 1-1 列出了决策支持系统的一般特征。

正如读者所看到的,典型的决策支持系统提供了多种功能。因此,要对 DSS 下一个简单而全面的定义也不是一件容易的事情。事实上,每一本有关决策支持系统的书都有不同的定义。可喜的是,这些不同的定义都陈述了一些共同的主题,我们把焦点集中在这些主题上,就能够得到一个比较好的、能够实际应用的定义。

表 1-1　决策支持系统的一般特性

- 用于半结构化或者非结构化的决策领域
- 用来辅助决策者,而不是取代决策者
- 支持决策制定过程的全阶段
- 着重于决策制定过程的效果,而不是效率
- 为 DSS 用户所控制
- 使用基础的数据和模型
- 具有帮助决策者学习的功能
- 交互式的,用户友好
- 一般使用迭代过程进行开发
- 为所有管理层次 (高层经理到一线经理) 提供支持
- 可以为多个相互独立或者相互依赖的决策提供支持
- 可以为个人、群体和团队决策提供支持

第一个主题是问题结构 (problem structuredness) 的维度,即决策或者制定决策的环境所表现出的结构化的程度。如果决策的目标简单,没有冲突,可选行动方案数量较少或者界定明确,决策所带来的影响是确定的,我们称这类决策是高度结构化的 (highly structured)。与之相反,高度非结构化的 (highly unstructured) 决策,其目标之间往往是相

互冲突的,可供决策者选择的行动方案很难加以区分,某个行动方案可能带来的影响具有高度的不确定性。决策支持系统的作用就是在决策的"结构化"部分为决策者提供支持,从而减轻决策者的负荷,使之能够将精力放在问题非结构化的部分。处理决策的非结构化部分的过程可以看成是人的处理过程,因为我们还不能通过自动化技术来有效模拟这种过程。这一点将在后文进一步说明。

在多数 DSS 定义中,第二个共同的主题是决策结果 (decision outcome)。在当今组织信息丰富的环境中,经常发现决策支持系统的踪影,也正是在这里,决策支持系统做自己的工作。考虑到用于支持决策过程的技术的一个关键元素就是决策本身,我们将花较大精力来探讨究竟决策是什么,为什么在制定决策时,需要一定的技术来支持我们。决策的效果 (即决策达到其目标的程度) 是决策过程中一个最基本的元素。因此,决策支持系统的定义中必须考虑到系统在支持决策目标的实现中所扮演的角色。

第三个共同主题是管理控制 (managerial control)。决策结果的最终职责、义务取决于管理人员。如果从这一点出发考虑,决策可以认为是管理者的武器仓库中最为强有力的工具。决策是所有组织在任何时点上分配和组织资源的手段,它是实现组织的战略目标的管理活动中的一个主要方法。无论把决策认为是一个选择、行动过程,还是战略,它都表现为一种活动,以从多个备选方案中选择一个最佳方案告终。最终选择的控制取决于决策者。为了达到目的,决策支持系统应该能够对选择过程提供支持。但是,最后的选择应该由对决策结果直接负责的管理者来制定。

在给定了上面三个共同主题之后,我们可以提出决策支持系统的一个正式定义。在这个定义的基础上,就能更好地理解 DSS 的构建、应用和使用。决策支持系统是受控于一个或者多个决策者,面向决策环境的非结构化部分,以改进决策结果的最终效果为目的的,辅助决策制定活动的系统。

DSS 可以做什么,不可以做什么

很显然,决策支持系统是一个非常强有力的工具,正在日益成为管理工作的一部分。如今,信息过时的速度简直让人目瞪口呆。未来的管理者面临的机遇越来越少,需要制定有效决策。决策的期限将以天、小时、甚至分钟来计算,而不再是季度、月和年了。决策支持正是为了在变化如此之快的环境中,使得未来的管理者能够有效地工作。为了满足管理工作的需求,决策支持系统必须提供给决策者一些对他们的业绩至关重要的关键元件。使用决策支持系统带来的好处并不是在所有的决策环境下,都能为所有决策者感知的。这种感知取决于决策者、决策内容和 DSS 自身合适的程度。假设三者匹配得好,一般来说使用决策支持系统能够获得很多潜在的好处。当然,我们也应该认识到使用决策支持系统的局限性。表 1-2 列出了这些好处和局限性。

表 1-2　决策支持系统的好处和局限性

好处

- 扩展决策者处理信息和知识的能力
- 扩展决策者解决大规模的、耗时的、复杂的问题的能力
- 缩短制定决策的时间
- 增强决策过程/结果的可靠性
- 鼓励决策者的钻研与探索
- 展现考虑问题空间/决策内容的新方法
- 生成新的证据，以支持决策或者证实现有的假设
- 创建战略或者竞争优势

局限性

- 决策支持系统尚不能包含明确的人类制定决策的潜能，比如创造力、想像力、直觉等
- 决策支持系统的能力受到所运行的计算机系统、设计方案、以及它所拥有的知识的限制
- 语言和命令界面还不够成熟，尚不能处理自然语言方式的用户指令和查询
- 决策支持系统一般是根据具体的应用来进行设计的，因此在不同的决策问题上，缺乏通用性

从上可以看出，决策支持系统通过扩展决策者在制定决策的过程中处理大量信息的能力，为决策者提供服务。决策环境中的许多部分，虽然是结构化的，但是相当复杂，需要花费较长的时间。决策支持系统则可以解决问题的这些部分，减轻决策者的脑力劳动，更重要的是，能够大量节省宝贵的时间。因此，使用决策支持系统可以大大减少达到一个复杂的、非结构化的决策所花费的时间。

在创新性和创造力方面，决策支持系统也具有一定的优势。通过简单地使用决策支持系统，可以提供给决策者很多途径去探讨那些很可能被忽视，或者被认为过于复杂和困难、无法实施的方案。决策支持系统中的工具可以模拟解决问题，达到一些关于解决途径和决策结果的具有创新性的见解。此外，决策支持系统的输出结果往往也能证实决策者的立场，从而促进多方一致意见的生成。最后，决策支持系统也提供了一种竞争优势，可以使得组织超过其竞争对手，或者至少并列市场前茅。要想达到这些潜在的优点，管理者不仅要合理使用决策支持工具，也必须认识到其局限性。

不管一个决策支持系统设计得如何完美，其价值都会受到某些因素的限制。首先，和其他计算机系统一样，决策支持系统仅拥有设计者所给定的知识和工具集所限定的“技巧”。虽然我们在第 8 章中说决策支持系统可以“思考”，在第 10 章中说决策支持系统具有“学习”的功能，但是它永远是受到其设计和设计者的限制的。

决策支持系统的缺点还包括在执行推理过程时，缺乏创造力、想像力和直觉。这些认知活动仍然属于人的经验，不能通过自动化或者机器模拟来进行。还有，决策支持系统必须以一种易懂的方式来与用户进行信息交流。虽然说人可以根据特定环境的需求和约束，非常轻易地改变他们沟通的方法，但是计算机系统（比如 DSS）却不能。因此，我们与 DSS 进行交流，以及 DSS 的响应的方法都是有效使用决策支持系统的障碍之一。我们将在第 13 章探讨界面和交互的问题。

最后，也是最重要的一个问题，就是并不存在一个“通用”的决策支持系统，很可能永

远也不会。一般的决策支持系统都是根据特定的问题环境设计的。因此,要有效解决一个复杂的,或者大规模的问题,可能需要使用多个决策支持系统。在这种情况下,决策者就要协调好多个系统,因为一个系统的输出可能就是另一个系统的输入。这样,整合多个决策支持系统,就成了一个有关复杂性和不确定性的决策问题。

总之,决策支持系统可以使得决策过程更加有效,在决策者的智能范围内辅助制定决策。但是,它们不能克服或者阻止不明智的决策者的行动。用户最终控制着决策的过程,因此,必须明白什么时候使用决策支持系统,使用决策支持系统干什么。更重要的是,要掌握好依赖 DSS 的输出结果和信息的程度。管理者应该认识到决策支持系统只是决策过程中的一个工具,而不是让它自己来制定决策。

1.2 决策支持系统的历史

DSS 的发展

决策支持系统是在管理者应用数量模型解决组织环境中所面临的日常问题和决策的基础上发展起来的。DSS 的概念是在 20 世纪 70 年代提出的,归因于当时发表的两篇文章。其一是 J. D. Little (1970) 写的"Models and Managers: The Concept of a Decision Calculus"。Little 观察到管理科学模型最大的问题在于管理者很少使用它们,他将决策运算 (decision calculus) 的概念描述为"一套处理数据和判断的基于模型的程序,用于辅助管理者制定决策" (p. B470)。Little 提出要成功地使用这种系统,系统必须简单、鲁棒、便于控制、完善、符合用户要求,并且便于交互。

第二篇文章"A Framework for Management Information Systems",作者是 Gorry 和 Scott Morton (1989),正是在这篇文章中提出了"决策支持系统"这个术语。Gorry 和 Scott Morton 提出了计算机支持管理活动的一个二维框架 (见表 1-3),每一维都假定是连续的,而不是离散的。纵向是决策结构的分类,这一点与 Simon 在 1960 年提出的一样。Simon 提出决策根据它们"程序化" (重复性、惯例性和通用性) 或者"非程序化" (异常的、独特的、相因而生的) 的程度进行分类。横向表示 Anthony (1965) 提出的管理活动的层次。这里假定所有的管理活动都可以归为三类之一:(1) 战略规划 (strategy planning),制定与组织目标相关的决策;(2) 管理控制 (management control),制定组织资源配置相关决策;(3) 操作控制 (operational control),制定组织日常工作和活动中的决策问题。通过将 Simon 和 Anthony 的方法结合起来,利用这个框架可以指导 IS 资源的分配,从而发掘最大的投资回报。在这个框架的基础上,就产生了决策支持系统的概念。

表 1-3 Gorry 和 Scott Morton 的决策支持框架

	管 理 活 动			
决策的类型	操作控制	管理控制	战略规划	所需支持
结构化决策	库存控制	生产线的负载均衡	工厂场所设定	MIS,定量模型
半结构化决策	有价证券交易	确定新产品的营销预算	购买资本资产的分析	DSS
非结构化决策	决定一本月刊杂志的封面图片	雇佣新的管理职员	研发项目的决定	人的推理和直觉

从 20 世纪 70 年代概念形成发展到现在，决策支持系统已经在最初概念的基础上进行了无数次扩充。当今学习决策支持系统，内容必然涉及到传统的基于模型的系统、基于知识的系统、人工智能、专家系统、经理信息系统、群体支持系统、数据可视化系统以及全组织的决策支持系统等领域。在后面的章节中，将对这些领域进行深入介绍。

前景展望

如果从决策支持系统及其应用的数量和类型上考虑，DSS 可能是所有计算机信息系统中最成功的一个。如今，信息技术已经渗透到了组织的每一个角落。不远的将来，组织在日常运作的各方面（从产品/服务的研发和营销，到员工、过程和全球活动的协调）都会使用计算机信息系统。当然，也不能过高估计决策支持系统在未来组织中的作用。在管理者的日常活动中，会更加依赖于功能强大、富有实用价值的 DSS 应用。21 世纪，决策支持系统在规模、复杂度和速度上将保持指数增长的速度。在第 15 章和第 16 章里，我们将展望在未来的全球市场 DSS 技术的各种应用。

1.3　决策支持系统的构成

基本 DSS 组件

到这里，读者应该已经认识到决策支持系统绝非一个具有共同特性、目标单一的简单系统。定义决策支持系统需要考虑到各方面的因素，包括其目的、使用环境以及结果目标。根据组成部件对 DSS 进行描述和分类同样是一件很困难的事情。因此，在构造一个一般的 DSS 结构之前，我们先看看根据组成部件对 DSS 分类的各种方法。

早在 20 世纪 80 年代，Alter 根据各个组件对决策的直接影响程度，将决策支持系统组件分为七大类，如图 1-1 所示。最初，Alter 的分类假定各类是一个独立的系统，而不是某个大系统的一个功能部件。从图 1-1 中可以看出，我们可以将七个独立组件采用一个更为简单的分类方式，即分为面向模型的系统和面向数据的系统。在后面我们会看到这种分类更符合决策支持系统的现代分类方法。

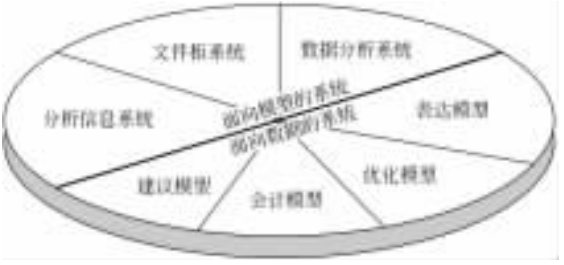


图 1-1　Alter 对 DSS 组件的分类

还有一个方法，从 DSS 提供的用于操纵数据/模型的语言的特性出发，来划分和定义 DSS 组件。根据使用语言的程度，划分为程序化（procedural）的和非程序化（nonprocedural）的语言。

语言程序化 (language procedurality) 的概念相对比较简单,不同于决策支持系统的设计和应用。基本上,DSS 用户应该能够指明他希望从决策支持系统得到的信息,以及信息是否存在于数据库,是否需要从 DSS 中的一个或者多个模型推导或者计算出,或者由 DSS 从其他模型构造的模型计算出。高度程序化的语言要求用户明确数据是如何获得的,从哪里获得的,以及模型 (或模型集) 如何对数据进行处理。非程序化的语言则只要求用户指定所需的信息,剩下的工作由决策支持系统来完成。在这两个极端中间,也有一种情况,以参数的形式预先指定数据检索操作,或者事先决定模型集。根据语言系统程序化的程度的不同,可以进行很多种配置和设计。

在上面讨论的两种分类方法的基础上,我们可以将决策支持系统组件划分为五大部分:

1. 数据管理系统
2. 模型管理系统
3. 知识引擎
4. 用户界面
5. 用户

数据管理系统

决策支持系统的数据管理 (data management) 组件主要管理某一特定决策的相关数据的检索、存储和组织。此外,数据管理系统也提供各种安全功能、数据完整性程序以及与使用 DSS 相关的全面的数据管理任务。数据管理组件中,多个子系统一道来执行这些任务。子系统包括数据库 (database)、数据库管理系统 (database management system)、数据仓库 (data repository) 以及数据查询工具 (data query facility)。在 1.4 节里我们将具体讨论这些元素。

模型管理系统

与数据管理系统的功能类似,模型管理 (model management) 组件主要执行与提供分析功能的定量模型相关的检索、存储以及组织活动等操作。在模型管理组件中,包括模型库 (model base)、模型库管理系统 (model base management system)、模型仓库 (model repository)、模型执行处理器 (model execution processor) 以及模型合成处理器 (model synthesis processor)。在 1.4 节里我们将对模型管理系统和数据管理系统进行详细地说明和比较。

知识引擎

知识引擎 (knowledge engine) 主要执行与问题识别、生成中间/最终方案相关的活动,以及与管理问题求解过程相关的功能。知识引擎是系统的"头脑",数据和模型在这里汇合,以提供给用户有用的应用,为决策过程提供支持。我们将在 1.5 节具体探讨知识引擎。

用户界面

与其他计算机信息系统一样，设计和实施用户界面（user interface）是 DSS 功能中的一个关键元素。DSS 要提供决策支持，而不妨碍其他工作，其数据、模型和处理组件必须能够轻松地访问和操纵。此外，不管是设定参数还是初期研究问题空间，用户与 DSS 交流的方便与否对于 DSS 的成功使用也是至关重要的。在 1.6 节中我们将具体讨论用户界面。

用户

如果不考虑到用户的作用，设计、执行和使用决策支持系统不可能有效进行。用户技能、动机、知识领域、使用方式以及在组织中的作用等相关的问题，是成功应用 DSS 的基本要素。应记住，决策支持系统的基本特征之一是受到用户的控制。如果不将用户看作系统的一部分，剩下的就只是一套计算机系统，光靠它们自身并不能提供任何有用的功能。在 1.7 节和第 2 章里我们来考察用户。

1.4　数据和模型管理

数据库

越来越多的公司已经开始认识到 数据作为公司的一项资产是至关重要的，必须加以有效管理。因此，在最近几年里，数据的收集、存储和传播过程已经经历了大幅度的改进。数据在组织中的价值对于决策支持系统的学习来说尤其意义重大，因为 DSS 数据库组件的质量和结构在很大程度上决定了现代 DSS 的成功与否。

数据库（database）是数据的集合，为了便于检索，以一定的方式加以组织和存储。数据库的结构应该与组织的需求相一致，应该能允许多个用户访问，或者在适当的时候，被用户的多个应用程序同时使用。基于数据的聚合度和粒度，数据库的概念将数据组织为一个逻辑层次的结构，包括如下四个元素：

1. 数据库
2. 文件
3. 记录
4. 数据元素

图 1-2 说明了层次结构中的四个不同元素是如何统一到数据库中的。

正如数据库是以文件的形式组织的数据的集合，文件（file）是以记录（record）的形式组织的数据集合，一条记录对应一条特定的信息。比如说，某个数据文件包含一定时期（不妨说是公司去年的）销售交易的相关信息 或者另一个类似的数据文件则包含整个行业前五年的历史销量信息。不管数据的特性如何，数据必须是围绕一个共同主题的，必须加以组织，这样在文件中可以创建有用的数据子集，并进行比较。

虽然说文件中的数据在结构上进行了组织，但是文件中数据的来源却可以是不同的。文件可以包含来自内部交易的数据，也可以包含来自外部数据源和 DSS 个体用户的

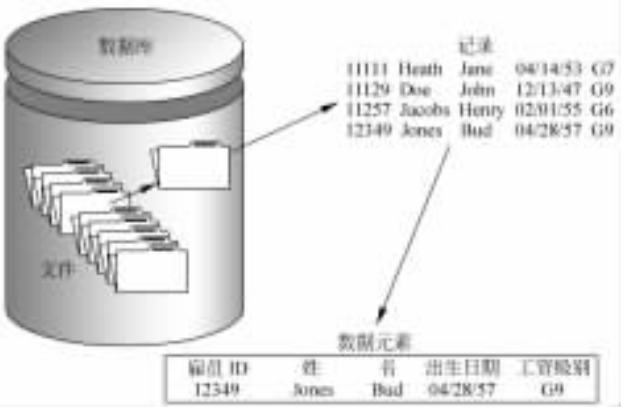


图 1-2 数据的层次结构

数据。

内部数据 (internal data) 一般来源于组织在日常事务中的交易处理系统。比如薪金和工资数据、销售交易、生产级别和产量都属于内部数据源,分布于 DSS 数据库中的一个或者多个文件里。事务级的数据是与组织或者业务运作相关的信息的主要来源。根据 DSS 的问题环境,这些数据很有可能对于提供决策支持都是必需的。当然,内部数据源只是 DSS 数据库信息的来源之一。

外部数据 (external data) 源的范围非常广泛,可以包括行业的某些特定数据、市场调研数据、某地区的就业信息、某产品的价格或者成本数据、地区经济或者人口普查数据或者某一法律范围内 (比如最高法院) 的案例清单等。可以说如果存在一个生成数据的"源",则就存在包含这些数据的数据库。对于 DSS 设计者和用户来说,一个便捷之处在于能够以较低的成本,非常轻松地获取到任何可以想像得到的外部数据。但是数据过多,却需要花费较多的时间来判断究竟该关注哪些方面的数据。幸运的是,因特网和 WWW 的发展,使得访问外部信息变得相当简便了。此外,数据传输服务 (比如 Mead 数据中心的 LEXIS-NEXIS,Dow Jones News Retrieval,CompuServe 以及其他无数的服务提供商) 使得外部数据的下载、综合以及分析成了 DSS 环境中一件相对普遍的事情。

不管数据来源何处,DSS 数据库中文件里的数据都被组织成统一的结构 (或者子单元),称为记录 (record)。同样,每条记录中的数据由数据元 (data element) 或者域 (field) 组成。数据元是逻辑层次结构中最小的数据单元,不能进一步细分。记录中的数据元指明了某一实例 (instance) 的相关信息。我们举个例子来加以说明,考虑一个包含有某组织产品线中不同产品的销售数据的文件 :诸如产品代码、产品名称、销售日期和时间、销售数量、销售地点以及其他一些与交易相关的信息,都可以存储在一条数据记录中。记录存储之后,可以根据数据的某些部分 (比如发票代码) 进行检索,或者以子集的形式进行检索,比如包含某一产品编号的所有记录。更重要的是,可以将它们与 DSS 数据库中的其他数据结合起来,进行某些扩展分析,而不至于被 DSS 用户所忽略。这种从多个数据源联接数据的能力给 DSS 用户提供了一个非常强大的、本质上无止境的数据仓库,用户可以在决策过程中从中提取有用信息。

还有一个数据来源是个体（individual-level）或者私人数据（private data）。在每次使用决策支持系统时，都会生成与决策、所提问题以及特定问题环境相关的数据。此外，不同 DSS 用户使用的个体直观推断（heuristics）或者拇指规则，也可以存储在 DSS 数据库中。这些个体的数据元和记录一般用于创建 DSS 的个性色彩，从而使得系统与某些用户或者用户群在定义的问题环境上的需求相一致。诸如个人对某些数据或者历史状况的评估、“12 月份产品 A 的销量将是产品 B 的销量的 37% ”的推断、或者历史查询等信息，也都可以以个体数据的形式存储在 DSS 数据库中。

数据库管理系统

组织在文件和数据库中的数据必须加以管理，这一重要的“管理”职能由数据库管理系统（database management system，DBMS）来完成。DBMS 主要包括两方面的职责：

1. 协调与数据库中信息的存储、访问以及传播相关的所有活动
2. 保持 DSS 数据库中包含的数据与 DSS 应用之间的逻辑独立性

现代的 DBMS 拥有非常全面的功能，通常由专门的数据库管理员（database administrator）管理。Sybase，Oracle，IBM 以及其他一些厂商开发的商业 DBMS 包一般都提供了一些功能强大的基础应用，利用这些应用可以对 DSS 数据库进行管理。最近 DBMS 的开发包括将大量不相关的或者离散的数据源整合为一个单一的、可以访问的数据库，也就是所谓的数据仓库（data warehouse）。数据仓库以一种便于 DSS 使用的方式，提供大量的数据，我们将在第 11 章和第 12 章详细讨论。

DBMS 的第一个职责“协调与数据库中信息的存储、访问以及传播相关的所有活动”涉及到很多的活动与功能。包括：在发生交易的同时更新（增加、删除、编辑、更改等）数据记录和数据元；从不同的数据源对数据进行整合处理；为生成查询或者报表，从数据库中检索数据等。此外，DBMS 还执行与 DSS 操作相关的许多管理功能，比如数据安全性管理（控制非授权访问、数据库故障恢复、数据完整性维护以及并发控制等）、为用户实验和分析创建临时数据库、跟踪 DSS 的使用和数据获取等等。这些活动大多是在后台执行的，并不为 DSS 用户所见。但是，它们对于使用 DSS 解决问题的成功与否却是至关重要的。

DBMS 的第二个职责，“保持 DSS 数据库中包含的数据与 DSS 应用之间的逻辑独立性”，对于 DSS 也是非常重要的。在一个典型的 DSS 情景中，一个或者多个用户根据从经常变化的数据中提取的信息来制定决策。DBMS 必须管理数据库中数据的物理组织和结构，以及与之相关的文件，而不束缚 DSS 应用或者 DSS 用户的数据的逻辑（或者外观）上的排列。相同的数据在一种环境下以一种形式（比如图表或者表格）表现，在另一个环境下则可能以完全不同的形式表现（比如作为一个分析模型的输入）。此外，找到了新的数据源之后，DBMS 必须对这些分散的数据源进行整合管理，这样它们看起来就具有非常整齐的结构。通过维持物理数据结构和 DSS 应用之间的独立性，DBMS 允许对一个数据库进行更加广泛的使用，比如一个数据库为多个 DSS 应用、多个 DSS 用户、多个问题所使用。表 1-4 列出了 DBMS 的一般功能。

表 1-4 DBMS 的一般功能

数据定义
• 提供一种数据定义语言 (data definition language,DDL),允许用户描述数据实体及其相关的属性和关系
• 考虑到来自于多个数据源的数据之间的相互关系
数据操纵
• 提供用户一种查询语言,来与数据库进行交互
• 允许捕获、提取数据
• 对某些特殊的查询和报表,能快速检索数据
• 允许构建复杂的查询,来检索数据,进行数据操纵
数据完整性
• 允许用户定义规则 (完整性约束) 来保持数据库的完整性
• 根据定义的完整性约束,帮助控制不正确的数据项
访问控制
• 允许辨别授权用户
• 控制访问数据库中不同数据元素和数据操纵活动的权限
• 跟踪记录授权用户使用、访问的数据
并发控制
• 提供一定的程序,对多个用户同时访问同一个数据库进行控制
事务恢复
• 提供了一种机制,在出现硬件故障时,重新启动并调试数据库
• 在某一点记录所有事务的信息,以保证数据库重启符合要求

模型库

模型 (model) 是为了便于研究问题从而更好地理解问题,对某些活动或者过程的一种简化。模型的一个非常重要的特点就是它是对现实的一种简化形式,用于尽可能接近地模仿实际过程或者活动,但是它往往没有现实那么具体。在很多情况下,我们都能够确定过程或者活动的很多特性。此外,在某些特定的条件下,我们可以对活动的性质或者结果进行预测,而不需要进行实际实验或者再现事实以便研究。通过研究模型,而不是活动本身,可以降低成本、节省精力和时间。我们可以想像一下要立在龙卷风中去预测其路线和强度,需要花费多大的努力 (和风险) 想像一下要观察飞机能飞多快 (或者能否飞行) 而去建造一个飞机,需要花费多大的成本 想像一下在销售新产品或者服务之前,为了考察其是否受欢迎而征求每个人的意见需要花费多长的时间。在这些情况中,建模可以成为决策过程所需信息的一个有效来源。虽然说 DSS 中的基本模型是"决策支持"概念的本质之所在,这些模型的仿真程度取决于决策者的需要。但是,如果没有模型,DSS 的作用就会受到很大的限制。我们将在第 5 章重点阐述各种不同的模型及其构造方法。

DSS 中的模型库 (model base) 与数据库的概念比较类似。DSS 数据库用于存放 DSS 使用的数据,模型库则包含 DSS 用于分析的各种不同的统计、财务、数学以及其他一些定

量模型。单独或者联合运行模型，以及构造新模型的能力，使得 DSS 成为解决问题的一个非常强大的工具。

模型管理

与 DSS 数据库中的数据一样，DSS 中的模型可以根据编号、尺寸和复杂度进行排序。DSS 使用模型库管理系统（model base management system，MBMS）来管理这些分析工具。表 1-5 列出了 MBMS 的一般功能：

表 1-5　MBMS 的一般功能

建模语言
• 允许从最开始的或者现有的模块中构造决策模型
• 提供一种机制来链接多个模型，以进行依次处理和数据交换
• 允许用户根据自己特定的偏好修改模型
模型库
• 存储、管理所有模型，并提供一定的运算法则，可以对模型进行轻松访问和操纵
• 对所存储的模型提供了一个目录和组织计划，同时对每个模型的功能/应用进行简要的描述
模型操纵
• 也提供一些与 DBMS 类似的函数（比如，运行、存储、查询、删除、链接等），可以对模型库进行管理和维护

MBMS 具有两个非常重要的职责，即 执行和整合 DSS 的模型，根据用户的偏好进行建模。一般来说，要求用户提供数据作为模型的输入，或者设定模型的参数（参数可能影响到模型的执行）。MBMS 通过提供给用户一定的途径访问模型，以及输入重要参数和数据来关注这一过程。

在执行模型的时候，经常要求用户采用一定的语法或者命令来启动执行过程。MBMS 则生成所需的命令结构，为 DSS 定位模型。同时，数据或者参数的格式化需求由 MBMS 的子系统来处理。最后，在使用 DSS 的时候，经常要将多个模型整合为一个复杂的模型。MBMS 通过提供模型之间的链接，控制模型执行的次序来帮助整合过程。

“根据用户偏好建模”是指收集、组织并整合 DSS 用户的偏好、判断和直觉到模型中的过程。决策者和决策环境的偏好都会经常发生变化，而且往往互相冲突。因此，MBMS 必须提供一种机制，合并各个用户或者问题环境的偏好到分析过程中，这样输出不仅仅包含定量的或者随机的结果，而且具有定性的条件和约束。总之，MBMS 在 DSS 应用中决策支持过程方面起到了非常重要的作用。我们将在第 2 章和第 3 章中讨论用户特性和偏好的问题。

1.5　决策支持系统知识库

在决策和解决问题的时候都少不了推理。没有进行推理的决策实际上不能算作真正意义上的决策。诚然，决策越结构化，需要的推理越少。我们可以想像，随着问题结构化

程度的增加,所需的推理将汇聚于某一点。在交叉点处,决策过于结构化,以至于根本不需要进行推理,这样实际上也就不存在任何决策了。

这里我们得明白究竟什么是推理。推理 (reasoning) 是在现有的或者以前推导出的信息的基础上推导新的信息的过程。在最简单的情形下,推理的依据是事实,虽然我们并没有亲自对信息进行明确的验证。比如说,我们知道某个公司的流动比率 (流动资产/流动负债) 为 2.6,速动比率 (速动资产/流动负债) 为 1.7,我们知道了公司两个独立的事实。通过推理过程,我们可以知道公司的流动性是比较合适的。并没有人告诉我们这个事实,我们也没有亲自验证它,但是我们知道了这个结论。在问题求解和制定决策的过程中可以使用多种形式的推理,我们将在第 8 章、第 9 章和第 10 章中详细讨论。在这里,推理的定义对于我们理解 DSS 的另一个组件"知识库"是足够了。

知识库 (knowledge base) 是存储 DSS 的知识的地方。知识,具体讲包括规则、启发式经验、边界、约束、先前结果以及其他的以程序的形式编入 DSS,或者通过重复使用获取的"知识"。

DSS 知识库中包含的信息与数据库和模型库中包含的信息是完全不同的。首先,知识库包含的信息一般都是与特定问题相关的,也就是说,知识仅仅在一个比较狭窄的问题求解范围内有用。相反,数据库和模型库组件则存储了范围非常广泛的元素。数据库或者与 DSS 相关的数据库不限于某一个特定的问题求解领域,模型库中包含的模型也不受到定义的求解问题的限制。然而,知识库一般并不包含直接相关的或者根据问题内容推导出的知识。我们可以这样看待三者的区别:知识是针对某一特定领域结构的,而数据和模型通常遍及到多个领域或者任务。

那么在 DSS 知识库中究竟都包括哪些内容呢?应该说凡是可能为领域专家使用的知识都可以包括在 DSS 知识库中,这些知识包括:不同目标或者实体及其关系的描述、各种问题求解战略或者行为的描述、相关的约束条件、不确定性、可能性等等。知识库中的知识可以简单地划分为两大类:事实和假设。事实 (fact) 是指在给定时间内为真的表述,而假设 (hypothese) 是我们认为在事实之间存在的规则或者关系。

我们假定有一个简单的知识库,其中包含很多专家认为在评估借款人信用程度时非常重要的指标:

1. 历史信用的年数 (是一个数字事实)
2. 历史信用的质量 (是一个排序事实,可选值为完美、几乎完美、非常好、好、不好、很差)
3. 借款人的工作描述 (是一个范畴事实,比如医生、律师、计算机操作员、大学教授等)
4. 工作的时间 (一个数字事实)
5. 借款人的总收入 (一个数字事实)
6. 借款人的总负债 (一个数字事实)
7. 借款人希望贷款的数额 (一个数字事实)

现在,除了这些事实之外,知识库中还包括它们之间的关系。换句话说,我们也存储

了这些事实之间彼此相联的方式。图 1-3 中说明了知识库中所包含的知识之间的关系。

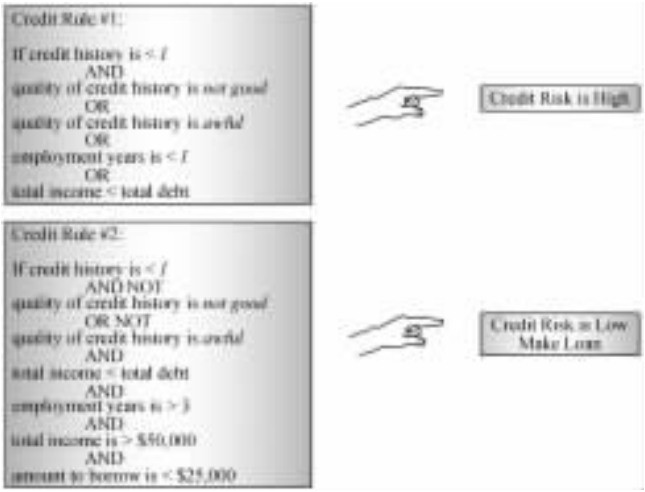


图 1-3　信用规则举例

在这个例子中，我们的目标是制定一个决策，判定是否应该借款给某个人。注意，事实之间的关系并不是单独针对某一个人的，而是以规则的形式加以描述的。一条规则可能陈述的是事实 1 和事实 2 之间的关系，而另一条规则则陈述的是事实 5 和事实 6 之间的关系，依此类推。虽然说也可以将这些规则看成是一系列的模型，但是它们可以以任何方式、按照任何次序组合，来达到一个决策结果。在图 1-3 中就列出了两个这样的组合。比如说，假设借款人历史信用很短（高风险），但是信用历史非常完美（极低风险）她有一份好工作，而且已经工作了很长时间（两者都是低风险），同时，相对收入而言她的负债极高（高风险），申请贷款的额度也极大（高风险）。在这种情况下，知识库将我们提供的事实与事实之间的关系（关系我们已经编写到了 DSS 中）结合起来，可能得出“我们应该反对贷款给借款人”的结论。当然，这个例子是非常简单的，与一般的知识库相比较，就显得太微不足道了。但是，从中我们可以看出知识库是如何在 DSS 中发挥作用的。

知识采集

到这里，读者可能想知道知识和所有的规则是从哪里来的。如果说 DSS 用户已经知道了问题的答案，那么为什么我们还需要 DSS 呢？这样想就对了，一般来说 DSS 的用户并不知道这个问题的答案。在使用 DSS 制定决策的过程中，用户依靠的是数据库中的数据的完整性，模型库中模型的质量和精确性，以及知识库中事实及其关系的价值和可靠性。那么，这些知识是如何进入到 DSS 中的呢？

答案很简单，知识的采集来自于专家。一个或者多个被称为知识工程师（knowledge engineer，KE）的人员通过采访各个领域的专家，来收集知识库所需的信息。要与专家进行交互，充分获取专家某方面的知识及知识之间的关系，知识工程师往往需要进行专业的培训。在挖掘专家的知识到知识库的过程中，可以采用多种知识采集技术，比如采访、原型分析、建模等。采集知识的过程是一项非常繁重的工作，我们将在第 8 章和第 9 章中讨

论采集和组织专家知识的各种技术。

知识检索

把事实及其之间的关系收集到知识库中之后，我们还需要在必要的时候从知识库中取出来。推理机（inference engine，IE）就是用来实现这一工作的，它属于知识库组件的一部分。IE 实际上是一个程序模块，用于激活收集到的领域知识，并执行推理，以在给定的事实及其关系或者规则的基础上得到一定的结果或者方案。在 DSS 中，必须提供给推理机一些有关如何应用规则、如何解决规则之间的冲突、如何验证推导结论是否可靠等等的规则。在第 8 章我们将深入讨论 IE。

读者应该已经认识到了，将 DSS 的知识库组件和现代 DBMS 和 MBMS 结合起来，可以创造出一个非常有效的工具，为今天乃至未来的经理提供复杂的决策支持。但是光靠这三个组件自身，却是几乎不能实现的。它们必须是可以为 DSS 用户理解的，这样才能真正支持决策过程。这里我们需要的是一个用户界面。

1.6 用 户 界 面

界面（interface）也是系统的一个组成部分，它能使用户非常方便地访问系统的内部组件，而不需要用户明白各部件是如何连接在一起，如何协同工作的。界面越是便于用户访问系统，就越好。另一方面，界面越是具有通用性，则需要用户进行的培训越少。在计算机程序（比如 Microsoft Windows）和应用软件背后有一个基本的概念，就是通用界面（common interface）。在一个 Windows 程序中保存和读取文件的过程与其他基于 Windows 的应用是一样的。但是，通用界面的概念并不仅限于基于计算机的系统。现代化的汽车已经在其界面上已经形成了几乎全球通用的一套特征，这样用户没有必要针对每一种汽车都去取得一个驾驶执照。除非驾驶一个界面完全不同的汽车，比如半拖车或者校园公共交通车，一般用户取得一个驾驶执照就够了。因为界面之间具有相似性，只要学会驾驶一辆汽车，也就学会了驾驶其他所有的汽车。

然而，DSS 还未能享受到通用界面的好处。这里需要提醒的是，并不是说 DSS 就不想有一个通用的界面，只是实现起来不是很容易。记住：DSS 并不是一个具有通用功能的应用，也不是为多个用户在多种不同的环境下使用的。相反，DSS 是针对某一个特定领域的，只是为一个小群体的用户所使用。因此，一个通用的用户界面在设计和实施上就不太容易实现了。我们可以这样来看待这个问题：因为一个特定 DSS 的用户基础是有限的，通用的用户界面就变得不太重要了。既然系统是针对用户定制的，界面也是针对特定的用户设计的。

界面组件

DSS 界面主要用于系统与用户进行交互和交流。从这个职责出发，界面就不仅仅包括软件（比如菜单和命令语言）和硬件（比如多重监视器或者输入设备）部件，还必须处理一些与人的交互、可访问性、便于使用、用户技能、错误捕获以及报表、文档等相关的因素。

考虑到其职责的广度和深度，DSS 专家通常认为界面是系统中惟一的最为重要的部件。如果没有一个好的用户界面，DSS 再强大的能力和功能也不能发挥出来。可以这样来考虑：从用户关注的角度出发，界面与 DSS 实际上是同一码事。拙劣的界面等同于拙劣的系统。在过去，许多 DSS 的界面比较拙劣，这样 DSS 反而阻碍了用户的使用，而不是辅助用户制定决策。表 1-6 列出了 DSS 界面的基本功能，以及在设计时必须考虑的因素。

表 1-6　DSS 界面的一般功能

通信语言
• 允许用户以各种对话的形式，与 DSS 进行交互
• 识别输入的形式，以将请求输入 DSS
• 提供多个 DSS 用户之间的通信支持
• 可以通过多种形式来实现，包括菜单驱动、提问/回答、程序命令语言、或者自然命令语言
• 可以捕获并分析历史会话，以改进将来的交互作用
表达语言
• 提供多种形式来表达数据
• 允许用户定义并生成详细的报表
• 允许以表单、表格和图表的形式显示输出结果
• 可以为同时可用的数据提供多重窗口（或视图）

通信语言

通信语言（communication language），或者称为动作语言（action language），主要处理用户与 DSS 直接对话的相关活动。在这个组件中，包含有各种形式的数据入口，数据可以以各种传统的方式（比如键盘、鼠标、触板等）输入到 DSS。近几年，通信组件发展很快，诞生了很多非常规的数据输入装置。虚拟现实装置（比如生物物理输入手套、扫描仪、头部监视器等）等数据输入技术也变得越来越普遍了。此外，很多 DSS 也开始使用游戏杆、语音识别、光学扫描等技术了。从本质上来说，任何可以用于输入的装置和技术都可以用于界面的通信语言组件中。图 1-4 中列出了很多可以用于 DSS 的输入装置。

或许通信语言最共通的方面就是菜单（menu）了。菜单提供了一个经过组织的、直觉的方法，供用户从多个功能、可选方案、命令或者结果中进行选择。经过适当的组织和逻辑设计，菜单可以为非专业的用户提供向导，有助于 DSS 专家更好地认识系统。但是，如果菜单设计不合理，很有可能导致功能强大的 DSS 不能使用。上文已经提到，在本书的始终，我们都会具体探讨如何设计一个好的界面。

表达语言

DSS 界面的表达语言（presentation language）组件是在使用 DSS 的过程中，用户实际看到的、听到的、使用的东西。输出设备，比如打印机、绘图仪、显示器、音频监视器、语音合成器等都属于这一组件。屏幕技术，比如多窗口、图形、表格、图表、图标、信息、以及音频触发器或者警报器等也都是表达语言的一部分。通信语言组件是用户传递信息和命令给 DSS 的基础，而表达语言组件则充当 DSS 与用户交流的工具。如果 DSS 要成为“用户

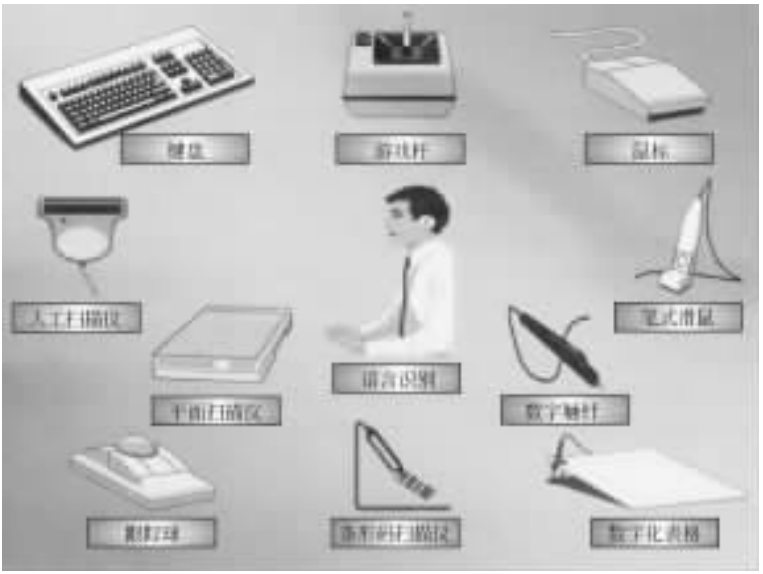


图 1-4 一般输入设备

友好”的,则两个组件之间必须做到无缝协调。

研究表明,信息呈现给用户的方式,对于信息的感知价值和其解释的精确度都具有显著的影响。必须对表达语言进行合理设计,这样 DSS 用户可以根据偏好事先选择,或者在查看阶段结果的时候选择表达的方式。各种图形、表格、图表要能够非常方便地选择,这样用户可以从多个视角来查看结果。这样能够保证输出结果符合不同的 DSS 用户的需求,也能保证输出结果与系统所支持的问题和决策相一致。在本书的始终,与通信语言一道,我们将针对某一特定的 DSS 应用来具体讨论表达的问题。

1.7 决策支持系统用户

没有用户,任何 DSS 都是没有使用价值的,也是不完整的。不像其他计算机信息系统,DSS 中用户与硬件或者软件一样,也是系统的一个组成部分。用户 (user) 一般定义为负责提供问题或者决策的解决方案的一个或者多个人。在系统的整个生命周期中,DSS 用户是参与得最多的。在 DSS 的概念规划、逻辑和物理设计、测试和实施,以及使用过程中,都需要用户的参与。

用户角色

上面定义的用户实际上不一定真正“使用”DSS。Alter 在 1980 年将 DSS 用户划分为五种基本角色,然后进一步将用户角色划分为四种不同的使用模式。图 1-5 中列出了用户角色与使用模式之间的关系。

在 Alter 的分类中,用户定义为直接与 DSS 进行通信的人,不管他采用什么样的方法,带有什么样的意图。决策者 (decision maker) 是在 DSS 输出结果的基础上最终制定决策的人。中介 (intermediary) 是一类特殊的用户,起到过滤或者解释 DSS 结果的作用。在

图 1-5　Alter 的决策者使用模式

决策过程的各个阶段，通过辅助解释 DSS 的输出结果，中介与决策者紧密联系在一起。维护员（maintainer）或者称操作员（operator），负责 DSS 的日常运作，比如保持系统和数据是最新的以及处于运行状态等。最后，输入员（feeder）提供数据给 DSS，但是可能永远也不会直接使用 DSS，将它作为一种决策支持工具。输入员由一个或者多个工作人员组成，定期地收集与问题相关的一些数据。在有些情况下，所有的用户角色可能由一个人来担当，也有可能为多个不同的人或者团队来管理。Alter 提出，不同用户角色涉及到的人员越少，则 DSS 越是简洁、便于使用。读者可能已经注意到，DSS 某些方面的用户可能永远也不会实际使用系统来制定决策。

DSS 使用的模式

Alter 描述的使用模式对于理解各种不同的用户如何与 DSS 进行交互是很有帮助的。在订阅方式（subscription mode）中，决策者一般接受预定报表（scheduled report），这种报表是在日常安排的基础上执行和生成的，通常由 DSS 自动生成，不需要用户进行查询。此外，这些报表一般是在对来自组织多个数据源的信息进行合并的基础上得出的，通过离线（比如硬拷贝）或者电子邮件或者文件传输等方法进行接收。订阅方式的例子包括定期接收预算变动表。

在终端方式（terminal mode）中，决策者在线直接与 DSS 进行交互。用户决定并提供输入，直接管理模型库中的模型，直接接受输出结果，并对结果进行解释。终端方式用户使用接收的结果来制定最终的决策，或者决定是否需要进一步与 DSS 进行交互。终端方式的例子包括使用 DSS 为一个经纪人客户分析/决定最优股票投资组合。

DSS 也可以用于职员方式（clerk mode）。决策者也是直接使用系统，但是并不以在线、实时的方式与 DSS 进行交互。相反，系统的输入、参数和请求是离线形成的，通过输入编码表或者其他电子批量提交过程来提交。在等待系统反应的同时，决策者可以从事其他的工作。在大量用户需要访问 DSS 系统，但是并不要求在线交互时，比较适合采用职员方式。这种系统可以有效处理大量的请求，在反馈结果给不同的职员方式用户之前进行复杂的分析。这类系统的例子包括用于决定保险公司客户的续保费率，或者用于决定某些特定服务行业客户报价的 DSS。

第四类 DSS 用法是中介方式 (intermediary mode)，用户通过一个或者多个中介用户与 DSS 进行交互。在这种方式下，由于分析或者参数输入的过程非常复杂，因此中介是必需的。在一个极端情形下，DSS 定义为中介使用的一种工具，用于准备最终报告给决策者。这种使用模式就减轻了决策者的负担，不要求决策者与 DSS 进行交互，也不要求决策者知道系统是如何工作的。中介方式一个典型的例子是，组织的定价结构是在多个部门的分析的基础上，由一个决策者最终决定的。到目前为止，中介使用方式为时还不长。与管理决策相关的机遇变得越来越小了，对时间的要求也越来越高了。因此，通过中介与技术“交互”所花费的时间对于很多机会来说是非常昂贵的。将来的决策者必须能直接访问支持技术，在配置和使用方面必须熟练。此外，在设计将来的决策支持技术时，必须留意技术配置和获得结果的速度。当然，就目前来说，这四种使用方法都有自己的使用领域，如果加以合理地使用，能够在问题求解上面提供很大的便利。

1.8 决策支持系统的分类

有多种方法可以对 DSS 进行分类，根据 DSS 提供的支持的类型，决策环境，用户参与的程度，面向数据、文本、规则或者模型的程度，以及面向单个用户还是多个用户，都可以对现存的或者正在开发的 DSS 进行分类。目前，理解 DSS 分类的依据就够了。在设计或者实施新的系统时，分类对于选择最佳的方法是非常重要的。我们将在第 13 章和第 14 章具体讨论这个问题。

以数据为中心的和以模型为中心的 DSS

在图 1-1 的基础上，图 1-6 列出了 Alter 对 DSS 的分类。在这种分类方法中，用到了 DSS 两个主要的支持偏向。以数据为中心 (data-centric) 的偏向主要着重于数据的检索和分析支持活动 ;以模型为中心 (model-centric) 的偏向则包括诸如模拟、最大化或者优化，以及那些在规则或者模型的基础上生成建议行动的 DSS 结果的活动。在这两大类内，还可以划分一系列 DSS 类型。

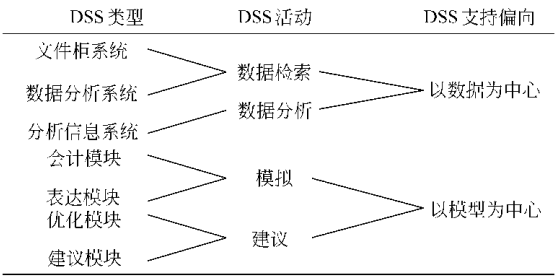


图 1-6 Alter 对 DSS 的分类

一般和专门系统

Donovan 和 Madnick 于 1977 年根据求解问题的属性提出了 DSS 的一种分类方法。一般 (formal) 或称常规 (institutional) 的 DSS 用于组织中周期性的或者重复发生的决策。这

种决策问题要求经常与 DSS 交互，以保证决策结果的一致性与有效性。在石油行业中周期性定价和与有价证券管理或者季节存货控制的决策中都可以找到这类系统的踪影。一般 DSS 在设计上一般非常稳定，经过几年的时间，发展为高度精炼的、可靠的支持机制。

相反，专门（ad hoc）DSS 一般是针对范围狭窄的问题背景或者不重复再现或不易预见的决策进行设计的。在这些 DSS 中，决策条件的性质和直接性就成了设计和实施方面的直接驱动因素。有关收购或者公司合并的决策都属于专门系统的范畴。以前，开发一套专门系统的成本相当高，因此仅用于极端原始的决策环境。然而，DSS 生成器（一套综合软件开发环境，提供了 DBMS，MBMS，知识管理等基本 DSS 功能）①的出现，使得专门系统成为了一种可行的、低成本的方法，可以用来为习惯已经成为日常议程的环境提供高质量的决策支持。

导向和非导向 DSS

Silver 根据系统提供决策引导（decisional guidance，即 DSS 在用户构造并执行决策的过程中，通过援助用户选择和使用操作符，来引导用户的方式）的程度对 DSS 进行了分类。Silver 定义操作符（operator）为在制定决策的过程中，可以/必须为用户操纵的 DSS 单元，比如菜单、按钮、模型、算法、工具栏等。这种分类方法表明用户可以从 DSS 的引导中获益。在图 1-7 中，以二维矩阵的形式，对不同决策支持方法的决策引导进行了分类。

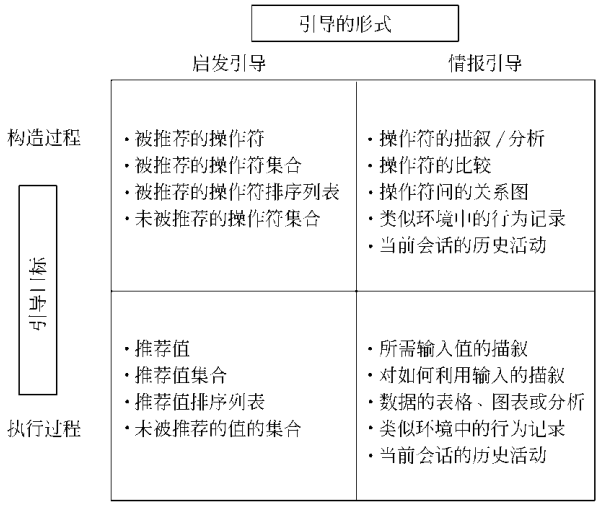


图 1-7　Silver 的决策引导分类

资料来源：MIS Quarterly，15 卷，第 1 册，1991 年 3 月。明尼苏达州大学信息管理和管理信息系统研究中心版权所有。

Silver 最初根据提供的引导的类型将 DSS 分类为机械的（mechanical）和决策的（decisional）的。机械引导是 DSS 命令中较为常见的形式，一般通过运行系统的“机构”（比如菜单、按钮和命令）来辅助用户。相反，决策引导主要帮助用户处理与问题内容相

① DSS 生成器及其应用我们将在第 13 章中讨论。

关的一些决策概念。这类引导可以进一步细分为用于构造或者执行决策过程的引导。最后,每种决策引导又可以根据引导采用的形式分为启发(suggestive)引导和情报(informational)引导。前者为用户提出行动过程方案,比如 DSS 建议用户在下一步调用哪一个操作符,或者推荐初始值作为输入选择算法等。后者提供给用户与环境相关的信息,但是并不指示用户应该根据信息进行何种操作。Silver 认为使用的引导的类型可能对如何使用系统和系统决策结果都产生重大影响。

程序化和非程序化系统

在 1.3 节中已经说到,程序性是指用户从 DSS 以任何形式获取任何想要的信息的程度。与 Silver 的分类方法类似,不同的 DSS 可以根据程序化的程度进行定位。比如,使用高度程序化的数据检索语言(比如 COBOL)的 DSS,可能要求用户明确提供数据检索和执行子程序的规范。严格的句法哪怕只有细微的差别,比如缺少一个括号或者逗号,或者颠倒了变量的次序,都可能导致 DSS 无法识别命令,或者更糟的是,产生一些不可预知的结果。这样的系统就属于极端程序化的。

使用不太程序化的语言的系统,比如用第四代编程语言(如 Oracle 或 Smalltalk)编写的系统,采用结构化的查询语言(structured query language,SQL),允许用户非结构化地构造命令或者信息请求。这些系统位于程序性频谱的中间位置。虽然说非程序化的命令结构易于用户理解和构造,但是仍然要求用户遵循一定的语法规则。因此,要有效使用基于命令的 DSS,用户需要学习一套新的词汇和语法规则。

近来,出现了一种朝着自然语言(natural language)命令语言发展的趋势。这些利用现代开发环境(比如 Lotus Notes)编写的系统,提供了一种非常类似于自然英语语言的命令句法。对非程序化的概念进行扩展,自然语言命令处理器以通用结构英语句子的形式接受命令和指令,这样极大地方便了用户的使用。我们在说话和进行书信交流时,词语的次序、标点符号、拼写和语法对于意思的传递来说都是非常重要的因素,但是有些时候与规则适度的偏差也不会影响我们理解句子的基本意思。在写作时,如果省略了一个逗号,或者错误拼写了一个单词,我们仍然能明白表达的意思。在面向命令的语言中,不管程序化程度如何,忽略了一个逗号或者颠倒了命令参数都可能导致命令处理器无法解释信息。自然语言 DSS 在这些问题上则具有较大的容错能力,而且随着时间的推移可以学习用户的意图。也就是说,命令处理器可以根据用户以前的命令或者请求来解释一条新命令的意思。此外,自然语言系统也可以采用多种输入方式,比如语音识别或者甚至可视化模式匹配等。这意味着可以直接通过说话或者对着连接到系统的照相机做动作来输入命令。虽然说自然语言命令处理器具有这么多的优点,但是这类的 DSS 还是处于幼年时期。技术要发展到用户直接以自然的、人的方式与系统交互还需要很长一段时间。我们将在第 8 章和第 16 章探讨这个问题。

超文本系统

另一种方法根据系统提供问题所需的知识管理功能时,所使用的技术对 DSS 进行分类。有一种技术是以文档为中心的超文本(hypertext)系统。这种系统通过跟踪大量不同

的知识库来支持决策过程，这些知识主要是文档或者文本的形式。在基于文本的 DSS 中，通常都包括文档的创建、修订、检索、组合、合并、编目、传递等活动。使用超文本的概念，基于文本的 DSS 有助于用户从多个文档中形成观点。一个文档中的不同部分可以链接到其他文档中的相关片断，这样可以研究某些特定的思路。这种思想在万维网（World Wide Web，WWW）中到处可见。

用户可以从一个网站出发，通过使用超级链接浏览到成千上万的文档（或者称网页，见图 1-8），对某种观点展开调查。在任何时候，用户都可以重新构造调查，或者放弃当前的研究重新开始。在相关文档和文本片断中的逗留完全取决于用户，与人的思维方式比较类似。超文本系统通过减轻用户记忆的负担来辅助决策过程。在记忆方面计算机的功能远远超出了人的大脑，因此用户可以专心进行思考和评估的相关决策活动。在日常生活的非结构化问题中，这些活动仍然是人完成得最好。

图 1-8　带有超文本链接的网页

资料来源：DSS Research Resources home Page，© 1997。经 Daniel Power 许可使用。

电子表格系统

DSS 用于知识表达的另一种技术是电子表格（spreadsheet）系统。电子表格概念简单，但是功能强大，通过使用一个行列交叉的二维表格来表达数据之间的关系，不仅便于用户从数字和位置两方面表达数据之间的关系，而且可以直接管理链接，查看最终结果。很多应用软件（比如 Microsoft Excel 和 Lotus 1-2-3 等）都提供了功能强大的电子表格环境，可以用作复杂的决策支持系统的基础。最近，超文本表达和电子表格系统的概念已经整合到了一些通用开发环境中，比如 Lotus Notes。使用这种方法，DSS 设计人员可以将电子表格知识表达与超文本文档结合起来，这样用户可以结合多种技术对数据进行操纵。

个体和群体 DSS

根据 DSS 为单个还是多个决策者提供支持，是 DSS 使用得最为广泛的一种分类方法。在现代商业领域中，由某个单独的决策者来制定多个决策是不太现实的，一般的决策

都是由多个决策者来制定的。因此，群体决策支持系统（group decision support system，GDSS）与单用户的决策支持系统在设计、实现和使用上都是不相同的。我们将在第 6 章和第 7 章讨论两者的异同点。

1.9 小　　结

本章介绍了决策支持和 DSS 的相关术语。通过使用本章提到的基本定义和分类框架，希望读者能深入认识后文即将讨论的各种 DSS；同时，在实际工作中，针对特定的问题能选择使用合适的 DSS。

DSS 远非传统的专业信息系统，它是一个需要有知识的动态系统。将来的决策者将依赖这种技术来支持日常活动，辅助管理制定有效决策所需的、日益膨胀的知识库。当然，本书也不可能涵盖计算机决策支持的所有方面，希望它能为读者更好地理解、更加有效地使用决策支持技术打下坚实的基础。

复习题

1. 决策支持系统具有一些什么样的共同特征？它们是如何与决策过程联系起来的？
2. 为什么说决策支持系统是决策者的一个强有力的工具？
3. 有没有可能存在一个“通用的”决策支持系统？为什么？
4. 为什么“程序”的概念对于设计、实施决策支持系统至关重要？
5. 列出决策支持系统的五个基本组成部件，并作简要描述。
6. 什么是数据库？数据库中收集的数据有哪些可能的来源？为什么说从不同的数据源综合数据记录的能力对于 DSS 用户是非常重要的？
7. 什么是 DBMS？解释 DBMS 的两个主要职责，并说明它们在 DSS 中起什么样的作用。
8. 什么是模型？在解决问题的过程中，模型库如何支持决策者？
9. 阐述 MBMS 的两个主要职责，并说明 MBMS 在决策支持系统中的作用。
10. 什么是推理？为什么说推理在决策过程中是非常重要的？
11. 什么是知识库？知识库在 DSS 的作用是什么？
12. 知识是如何进入到 DSS 中的？如何将知识提取、组织成为有用的信息？
13. 为了使用户能够与 DSS 进行良好的交互，需要什么部件？
14. 解释一下 DSS 中各种不同类型的用户的作用。
15. 举例说明各种使用 DSS 的模式。
16. 比较说明数据驱动和模型驱动的系统、一般和专门系统、导向和非导向系统，它们分别具有什么优点，什么缺点？
17. 比较一下高度程序化的语言和非程序化的语言各有什么好处，什么局限性？

讨论题

1. 分析一下市场中的某个 DSS 应用，描述其主要组成部件，并总结其功能。
2. 观察你所熟悉的某个组织中的决策过程，讨论一下决策支持系统如何可以帮助这个决策过程，判定哪种类型的决策支持系统适合这个决策过程。
3. 比较 DBMS 和 MBMS，它们具有什么共同点，有什么不同点？
4. 马里兰大学的招生办公室需要一套决策支持系统来辅助评估申请表，他们需要为 DSS 建立一个数据库。请思考可以使用的数据源和数据都有哪些。
5. 采访某个组织中使用 DSS 的成员，说明 DSS 中用户的作用，断定其使用 DSS 的模式。
6. DSS 专家们都赞同“界面是系统中最重要的组成部件”，为什么用户界面在决策支持系统中具有如此重要的地位？什么样的界面算得上是“用户友好”的？

第 2 章

决 策 者

学习目标

- 理解决策过程的组成和框架
- 熟悉决策者的类型的形成
- 基于决策者认知的复杂性及价值导向，理解决策类型的分类及其相关的三个因素：决策问题所处的背景、个人洞察力与价值观
- 为了设计好系统以提供恰当的决策支持，理解“决策问题所处的背景”与“决策类型”之间的交互作用
- 理解什么是好的决策，以及在决策过程中促使决策者采取行动的力量有哪些
- 学习决策支持系统可以提供的几种决策支持的类型

小案例

特内里费岛的灾难

这场灾难是航空史上最严重的灾难之一。人们可以从这次灾难中吸取很多教训，因此，机舱资源管理领域里的专家都会对此进行研究。远离摩洛哥海岸线的 Canary 岛是有名的旅游胜地，而位于特内里费岛上的 Los Rodeos 机场在 1997 年 3 月 27 日这一天是异常繁忙。由于位于 Canary 岛中心的 Las Palmas 机场前一天下午发生了爆炸事件，归航的飞机只能转向 Los Rodeos 机场降落，糟糕的是：特内里费岛上的 Los Rodeos 机场的承载能力并没有 Las Palmas 那么大，飞机只能暂时挤在机场的坡道上。在这些转向到 Los Rodeos 飞机中包括了 Pan Am 班机 1736 和荷兰皇家航空公司（KLM）的 4805 班机，都是满载乘客的波音 747 客机。Pan Am 班机在 KLM 班机之后降落到停机坪上，就停在其后，恰好是 12 号跑道的末端，不能起飞。

在 KLM 飞机的甲板上，受过很高训练的机长 Jacob van Zanten 正焦急地等待飞回去，因为他对全体机组人员负责的时间正在减少。当控制台无线电广播说，Las Palmas 机场已经重新开放的时候，van Zanten 决定等在 Los Rodeos 机场的坡道上给飞机加油，而不是飞到 Las Palmas，因为那里必定由于刚刚重新开放而拥挤。现在该 Pan Am 1736 号飞机起程离开了，但惟一可以到达起飞跑道——30 跑道的路径是：从坡道进入 12 跑道，然后沿原路返回。不幸的是 KLM 4805 号机刚刚开始加油，在 Los Rodeos 机场根本没有足够的空间让 Pan Am 1736 号滑过去，Pan Am 的第一指挥官 Bragg 电话询问 KLM 机组人员要用多长时间能够加完油，对方回答道："大约还要 35 分钟。"这样，1736 号航班除了原地等待外别无选择。就在 4805 号航班加油的同时，雾气慢慢笼罩了整个机场，等到 4805 机加完油的时候，机场的能见度在某些区域已经降到了 900 英尺。KLM 机组人员开动引擎，准备起飞。就在他们滑向 12 跑道起点的时候，控制台传来通知："一直向前滑……嗯……沿跑道……原路退回。"就在这时，Pan Am 1736 号班机已经开动引擎，占据着本就不是很长的跑道。能见度低使得控制台既看不清跑道，也看不清飞机。Bragg 于是呼叫控制台请示进一步的命令，控制台回答说："滑入跑道，……嗯……然后从第 3（Third）跑道起飞，第 3 跑道就在你们左边。"

显然，Pan Am 机长 Grubbs 并未听清楚控制台播音员的发音，他说，"我以为是'First'"。Bragg 回答道："我会再询问一下。"与此同时，控制台再次通知 4805 班机全体机组人员。"在跑道尽头做 180°转弯，然后再报告，……嗯……，准备好 ATC 通行证。"这次通话过后，Bragg 回呼控制台讯问："能确认一下，您是让我们在第三个路口左转吗？"控制台回答是："是的，先生！第三个路口，一、二、三……的三"。而 1736 号班机在跑道上滑行时，要辨认清滑行跑道还有一些困难。这时候，4805 班机已经到了跑道末端，正进行 180 度大转弯。转过弯后，van Zanten 打开节气门，飞机开始向前滑动。首席工程师 Meurs 说："等会儿吧……我们还没有得到 ATC 通行许可"。对此，van Zanten 的反应是："我知道！是没有！继续走再问。"因为他握着刹车，他不制动，别人

也没办法。Meurs 呼叫索要通行许可，就在他还没读出来时，van Zanten 再次打开节气门，说，“检查牵引力！我们走！”Meurs 重复完通行许可，试图让控制台知道目前的状况，报告说，“我们正在起飞！”控制台显然把他的意思理解成了：他们正准备起飞。说道：“OK！站在旁边等待起飞……到时候我会呼叫你们。”在 Pan Am1736 的飞行甲板上，全体机组人员正为 4805 班机的转移相牵连而焦虑，Bragg 报告说，“我们还在跑道上滑行！”控制台回答：“Roger，Pan Am 1736 请报告跑道情况。”不幸的是，这次信息传递因为控制台信息阻塞而并未传到 4805 班机上，以至于所有 KLM 机组人员都只听到“OK”！而 1736 班机的转移干扰了 4805 班机飞行工程师 Schreuder 的工作，他大声问道：“他们不清楚自己的跑道吗？”van Zanten 由于全神贯注于起飞，回问道：“你说什么？”Schreuder 重复道：“他不清楚自己的跑道吗？那个 Pan American！”van Zanten 与 Meurs 都回答说，“不！他清楚！”可 1736 班机仍旧沿跑道滑行，试图找到合适的地点停下来。但显然现在也牵扯到 4805 班机的转移。Grubbs 道：“要是我们安全驶出这里，我们都下地狱好了。”Bragg 回道：“是啊……，你焦急了，不是吗？”几秒钟后，Grubbs 打开 4805 班机的照灯以透射尘雾，突然他大叫起来：“看！那个家伙就在那儿！天哪！那个狗娘养的正朝这儿开来！”van Zanten 也看到了 1736 班机正在跑道上滑行并企图拉回时，他制动了所有的四个刹车，试图让飞机脱离跑道，以避免影响其他班机。Bragg 大喊：“停下！停下，停下！”

调度人员也想让 1736 班机脱离跑道，但还有其他很多班机挤在 Pan Am 班机的右侧，4805 班机在空中停了几秒钟，就撞击地面引发爆炸；Pan Am1736 受到冲撞马上就着火了。KLM 4805 班机上所有人无一幸存；Pan Am1736 班机上全体机组人员幸存，而且都没有受到损伤，因为幸好没被 4805 撞击。令人惊奇的是，Pan Am 班机上还有其他 66 名乘客幸存。不幸的是，特内里费岛那一天的灾难有 583 人丧生，至今还是航空史上最严重的一次事故。

研究者头脑中最大的疑问是：像 van Zanten 这样受过很高训练、经验丰富的机长怎么会在未得到控制台起飞许可时决定起飞呢？是 Meurs 在 van Zanten 已经松开刹车时才向控制台发出请求许可的。很显然，van Zanten 在 Meurs 制止他时，他很清楚还没得到起飞许可，他当时还回答说，“我知道！是没有！继续走再问。”很可能是 van Zanten 急着赶到 Las Palmas，因为他们的停留延误了很多时间，他的机组人员没有额外的责任时间。然而，即使在得到许可之后，控制台也是命令 KLM4805 班机“停在旁边等待起飞！”关键的是，机组人员都没有听到这一命令及 Pan Am 1736 班机仍在跑道上的指示。另外，Meurs 除了提醒 van Zanten 还没有起飞许可之外，其他什么也没做。也可能，凭借 Meurs 的经验水平并不足以向 van Zanten 挑战。进一步使形势恶化的是：1736 班机机组人员的所有的努力又都因为能见度降低而白费。他们手边只有一张很小的机场图示，从他们滑行方向可退回的跑道只有一个：一个 135 度的大转弯。这对于波音 747 来说，相当具有挑战性。他们显然还认为在 45 度方向上是不是还有一个跑道，是控制台计划让他们使用的。因此他们在经过第三个路口之后继续前行。事实上，Los Rodoes 机场并不存在另一条标记的可以使用的跑道。最后一个要考虑的因素是：

控制台工作人员与 KLM (荷兰) 机组人员用英文交流存在着一些困难。随着天气状况恶化,仅仅靠无线电交流使得双方对形势的判断出现明显的不一致。荷兰方面的调查团把责任全部推到 Los Rodeos 机场控制台工作人员身上,而美国方面的调查团认为 van Zanten 机长才是造成事故的主要因素。

你能判定吗?

2.1　谁是决策者

在第 1 章我们已经从内到外分析了 DSS 的组成部分和关键因素。我们从 DBMS, MBMS 及知识库入手,从系统的内部机制到用户界面,最后到用户都有所涉及。这种从内到外研究的目的是为理解 DSS 是什么及不是什么奠定基础。关于这两个主题有很多可讨论的东西,但每个讨论都需要有一个切入点,这个点就是从您——决策者开始。

第 1 章简要涉及到了关于设计与使用 DSS 的目标或者说目的这个主题。实际上,它非常简单:我们知道我们需要做决策,但却不能确切知道怎样做。在这样的背景下,尽管 DSS 在决策制定领域的那个部分有很高的价值,"怎样决策"并不总是过程与方法,而是要能够处理大量数据、管理无数现实模型,能对一些关键结果给出"预测"的过程。本章我们从关注为什么需要决策支持系统的根本原因——决策者开始。要不是问题解决者,在结构化决策过程中某些经常是冗长而复杂的阶段需要决策支持的话,我们就不会开发 DSS,同时,也就不需要谈论了。让我们多少感到高兴的是:决策者需要决策支持,而 DSS 恰好就可以提供支持。因此,我们就从探讨哪些是决策者及什么使他们彼此不同开始。

决策剖析

在我们把关注焦点直接转移到决策者之前还有最后一个问题。我们需要决策者在决策过程中完成任务的工作模型。

有关这个主题的文献很多,涉及到怎样决策、决策过程分为哪些步骤、做决策的动机以及决策步骤中又分成了哪些小步骤等等。提醒一句:所有这些关注都没什么坏处。对于管理者吸收和理解知识有很多方法,这里我们只把注意力放在一到两个决策过程的理论框架,并用这些框架引导我们对它所涉及到的过程以及人员的研究。

图 2-1 展示了决策制定的许多路线中的一种的剖面,这个模型一定要如此描述的原因在于:在现实中做决策时,恰好用模型中建议的程序或是用到模型中的所有步骤的决策是很少见的。而且,这里是用静态的过程来表现显然是动态的操作。要解决的问题都处在特定的条件和环境下,这从根本上限定了决策应采用的方式。尽管如此,我们还是可以使用图 2-1 中描述的模型作为理解典型决策背景的基础。

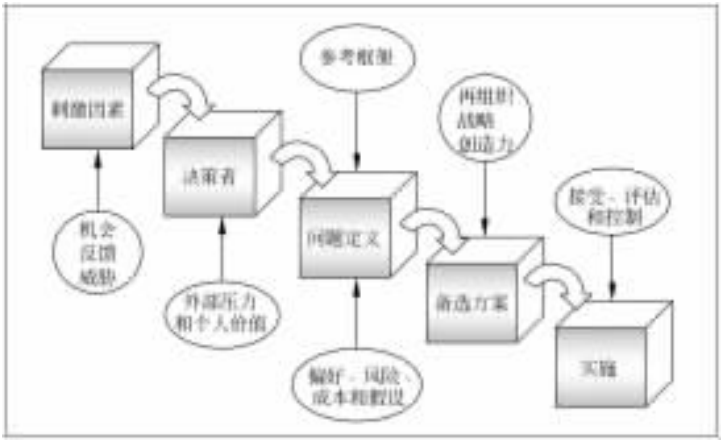


图 2-1 决策制定过程举例

资料来源：Rowe，A. J.，Boulgarides，J. D.，(1994)，Managerial Decision Making，Englewood Cliffs，NJ：Prentice-Hall。经作者许可引用，© 1985 年版权所有。

在决策进行过程中，决策者扮演某种分裂的角色。决策不仅是决策过程中的要素或决策步骤（图 2-1），而且，还是决策过程中各个步骤不同程度的参与者。第 3 章中我们还会更进一步理解这种双重作用是如何使得决策过程复杂化，而且增加管理该过程的复杂性。然而，在这里我们只需简单考虑过程中的每一个要素以及它们是如何集成在一起的。

刺激因素

过程的第一步是某些外在因素或力量促使决策者感知到存在一个或多个问题需要做出一个或多个决策。问题（problem）可以被简单定义为：对事情的当前状态与期望的状态之间差距的感知。如果决策者认为事态需要改变或状况需要改变，那么问题的存在就被感知到了。这种对问题背景的感知是由一个或多个刺激因素引起的。这些刺激因素可能有不同的形式，不同的决策者感知也可能不同。换句话说，一个决策者感知到的问题背景，同一组织中的另外一个人却不一定感知到。这种不一致可能会发生，原因在于问题实际上并不存在，而决策者对现有事实或信息曲解了；或是问题实际上存在，而组织中的某个或多个成员没有认识到。复杂吧？

刺激因素的形式可能会经常影响对问题的理解，通常，一个问题本身并不能一目了然地看出它是个问题。但它一定有一系列的表征显示出它是个潜在的问题。在第 3 章我们会探讨问题与其表征之间的差别。这里，足可以说某些事情“发生”引发了决策过程。

决策者

图 2-1 已经显示了决策者是决策过程中的一个部分，但我们从头到尾的讨论都是假定决策者是决策过程的参与者。在接下来的几章中，也都是如此。这里，决策者定义为：刺激开始后的下一步或接下来的事件。

决策者这个词真正是个“黑箱”，我们可以把真正“知道”的所有事情都嵌入决策者的意念里去。有人可能会争辩说，嵌套不就是装篮子吗？我们确实知道很多，但是不管怎

样,我们只是不知道那么多关于 DSS 用户的。这是一个方面的研究领域,它对将来的管理者会有很大的贡献。

问题定义

就像前面提到的,问题经常会显现出一系列与实际问题相关联的表征或是问题的结果。但是不管情形如何,问题解决者在对备选解决方案进行有效研究之前都必须定义问题。这一阶段对于成功地得到决策过程的结果是至关重要的。如果问题的正确定义能被明确地表达出来,那么,就能正确地解决问题;反之,如果由于对要解决的问题的真实属性没有正确领会而错误地定义问题的话,那么就会有很大可能是向一个错误的问题提供解决方案。更糟糕的是,它可能还会被实施。于是希望 DSS 能够指导我们去除错误的问题定义,识别出正确的那个。

备选方案

这是决策过程的核心,也是 DSS 最有用的地方。从一系列可行的备选方案中选择有效的解决方案是决策过程的关键。这本身就是决策。用 DSS 可以提供定量化的方法来分析这些可行的备选方案,以帮助管理者选出最佳的可行方案。然而在第 3 章我们还会看到,这种选择过程是我们最无力解决的部分,因此,一定要高度依靠 DSS 的帮助。

实施

用户从备选方案集中选出最有效的解决方案,实际的工作才刚刚开始。决策过程之后,组织内部就集中精力来执行方案以解决问题,因此会触发一系列的相关行动和事件。这些行动包括:达成共识和认可、谈判、制定战略、政治活动及认真的规划。世界上最伟大的解决方案,如果不能得到很好的贯彻实施的话,也是毫无价值的。DSS 在帮助与执行直接相关问题的决策上,只能提供有限的支持;这也是决策者本身成为最重要参与者的一个阶段。一个方案实施的成功与否,根本上取决于决策者自身。我们会在第 4 章详细探讨这方面的问题。

决策者的类型

现在我们已经有了一个决策模型来帮助我们理解为什么要首先把注意力放在决策者身上。有那么多不同类型 DSS 的原因就在于:存在那么多需要支持的决策者。图 2-2 中给出了不同类型决策者的一个模型。

个体

个体决策者,顾名思义,就是在决策过程当中独一无二的决策者。这类用户本质上来说,在决策过程中单独工作,也就是说,对信息的分析以及最终决策的生成都完全掌握在他们的手中。根据定义,由于个体决策者是一个具有独特性格的个体,其性格可能与如下因素有关:知识、技能、经验、个性、认知风格以及个人在决策过程中的偏好。所有这些特点除了直接影响决策者最终决策以及在决策过程中需要什么样的支持外,相互之间还有

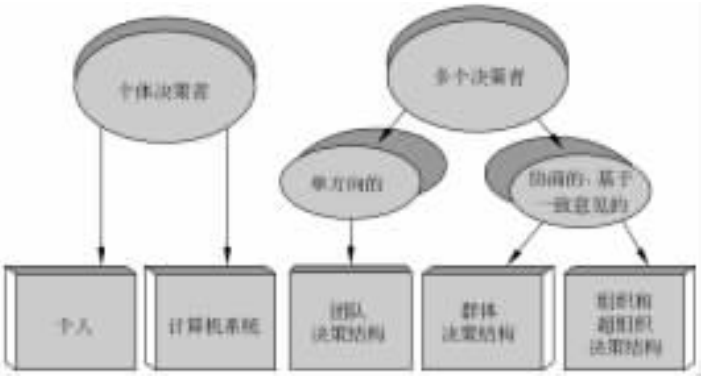


图 2-2 决策者的分类

交互的作用。这也是我们为什么通常把 DSS 设计得非常复杂的一个原因。为达到真正的有效，专门给个体问题解决者使用的 DSS 在设计时一定要考虑决策者本身独特的特点和要求。如果有多个个体使用同一个 DSS，那么这个 DSS 一定要设计成能反映每个人的独特之处和要求的系统。这个问题的组合特性应该是显而易见的。设计一个个体使用的 DSS 系统是一项复杂而困难的活动。我们这里只是提到它的复杂性与艰难性，更进一步的叙述将在第 13 章和第 14 章涉及。

多人

这类决策者是由多个个体组成的，这多个个体相互作用、共同做出决策。我们在下一节会区分多人决策者、群体决策者以及团队决策者。在这里，多人决策者是指：每个人在某个详细的决策中都占有一定的份量，最后达成意见一致及共同的约定。这类决策者每个成员都可以有自己独特的动机或目标，也可以从不同的角度进行决策。此外，每个成员还可以使用同一个 DSS，也可以使用不同的系统来支持他在决策过程中的贡献。最后，多个决策者还经常出现这样的情形：多个决策者对做某个决策的职权不等，但他们中的任何一个都没有单独做决策的权力。在这样的问题背景下，多人决策者并不需要作为一个单位用正式的方式或公开论坛/讨论的方式碰面。相反，交流的制度化及组织内部不同的职权等级决定了参与各方之间的相互作用，就是以这样的方式最终达成“决策”并开始实施。给这类用户一个最好的概念是：一个动态的用户联盟，在这个联盟中，决策者们都以个体的身份参与决策，最终形成一个决策或是对某特定问题背景的解决方案。

群体

与多人决策者相比较，群体决策者以成员之间存在一个正式的组织结构为特征，在这个结构中，群体中的每个成员都对决策结果有类似的既定影响，并且在构成上有相同的发言权。群体决策者在整个决策过程中都是在正式的环境下工作，参加例会，在决策过程每个部分都有正式的日程安排和议事程序，经常还会有决策最终完成的期限。群体决策者最普通的例子是组织的委员会与评审委员会。依据群体多数人意见做出决策时，每个参与者都要参加，但是任何一个参与者都没有特权。由于决策过程的正式化是全体决策者

区别于多人决策者的基本特点，因此如果由 DSS 提供决策支持的话，就必须理解群体过程的独有的特点。我们将会在第 6 章和第 7 章深入探讨这个话题。

团队

还有一种类型的决策者是团队决策者，这类决策者可以被看成是：个体与群体类型的组合。一个组织的结构经常是这样的：尽管做决策的特权掌握在某个个人的手中，但是他还有一些为共同目标工作的助手的支持。在团队中，决策支持可以来自于经核心决策者授权的若干个体收集信息，最后做出某个方面的决策；另外，支持还可以来自于一个或多个 DSS，系统可以由核心决策者与其助手组成任意组合的团队使用。在这样的情况下，团队产生或“制造”了最后的决策，但决策是正式的，而且，做决策的特权是落在个体决策者身上的。图 2-3 说明了个体、多人、群体及团队决策之间的区别。

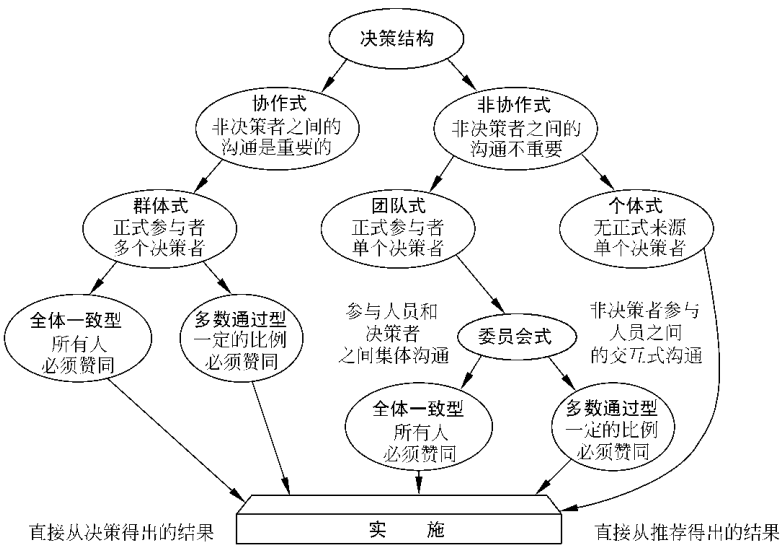


图 2-3　决策结构的分类

在群体与团队决策之间一个最独特的差别是决策的类型。群体决策的结果通常是协商的结果，这是因为外部力量经常首先决定了群体形成的必要，而且，在这样的情况下，群体面临的选择经常是有争议的。因此，群体决策者倾向于寻找那些能达到最初的目标、又能让所有相关方都能接受的折衷方案。而在团队决策中，决策通常实质上是单方的。尽管可以说有很多人影响最后决策的形式，但只有一个决策者有特权和责任做出一方的决策。因此可以说，这类决策最最基本的，通常拥有许多与个体决策相似的特征。

组织与超组织

50 多年前，著名的管理科学专家 Chester Barnard 把组织定义为：一个由两人或两个以上的人组成的可以自我调和行动的系统（Barnard 1938，73）。尽管自这个定义之后，很多东西已经发生了变化，但今天看来，这个定义仍是成立的。把组织成员粘在一起的东西是什么呢？那就是：组织成员共同的目的与目标。用这个定义我们能很容易想像得到，组

织内部还会有组织 群体和团队,他们也能为达到共同的目标而自觉调和行动。组织层次的决策者是那些被授予特权、代表组织整体负责决策的人。根据定义,这些决策及决策过程的特点很大程度上与个体、团队和群体决策者类似。既然这样,我们为什么还要分出一个组织层的决策者类型呢?

原因之一是组织决策者所需要的信息的广度和深度不同。这类用户包括高层管理者——CEO,在第 6 章和第 7 两章我们会看到,已经出现了一种专给这个层次用户使用的 DSS,我们称这种特殊的 DSS 为经理信息系统 (executive information system,EIS)。

另一个我们区分团体或群体与组织层次的原因在于,组织决策者所做出的决策通常需要得到整个组织的支持才能成功地实施。这类决策在很多日本公司里都能找到典型的实例。所谓的 nemawashi 这个概念,就是一个核心决策者,在组织的所有层次上都遵循服从多数人意见的约定,最终的决策所有人达成了共识。如果没有大部分下属的支持,组织的决策者,一般来说,没有大范围实施决策结果所需的资源。于是,这些决策者就有不同于个体、群体以及团队的特定支持的需求,因此应该单独归为一类。

在组织层次的决策者之外,还存在另一类决策者。世界上所有组织构成企业系统。在这个层次上活动的用户所做的决策倾向于社会福利、生活质量、受调节或有限资源的分配、社会秩序或社会正义等。换句话说,这个层次的决策者不像企业内部的决策者那样把经济目标放在首位。相反,他们更关注对社会秩序影响的决策。由于组织是存在于社会经济大系统中的,因此,所有的决策者以及他们各自的决策,都应该考虑超组织层次的决策。更重要的是,也像其他所有的决策类别一样,超组织决策者也需要特定的支持,因此需要特定的 DSS。

2.2 决 策 风 格

为了更进一步理解决策者,就需要更好地理解直接影响他们行为的因素——就需要对决策风格进行研究。决策风格 (decision style) 是用来形容决策者决策方式的一个词汇。管理者的设计风格可以通过他对某个给定决策问题背景的反应反映出来——认为什么是非常有价值/非常重要的?信息如何解释?外部事物和力量如何处理?某个特定决策者的决策风格对决策结果的最根本的影响是由如下一些因素决定的 问题背景、决策者的洞察力及个人在特定境遇下的价值观。幸而存在着对决策风格进行测量和分类的方法,这些测量得到的结果可以报告给 DSS 的设计与实施人员,甚至比这更进一步。

背景、洞察力与价值观

背景、洞察力与价值观三个因素共同作用于决策风格。问题的背景 (context) 包括了决策过程当中与作用于决策者的各种力量相关的因素。组织的与环境的力量,比如政府管制、新技术、市场竞争以及内部力量斗争等等,都会影响到问题的背景。另外,一些个人在群体中的特征,比如技能、精力、动机、感知能力等,也能够塑造问题的背景。在问题解决过程中,决策者对以上所有提到的力量都应该很好地管理和平衡。

另外一个影响决策风格的因素是洞察力 (perception)。决策者经常把个人偏好带入

决策问题的背景中，就会经常颠倒事实以和他们自己认为的事实相匹配。管理者有按照个人的目标而不是严格意义上的现实来感知问题形势和潜在的解决方案的趋向，这已经得到了证明。为此，感性的偏好与问题所处的背景交互作用，一起决定了决策者解决问题的方法，而不管这种方法是否有效。我们将在下一章进一步探讨个人偏好对决策结果的影响。

最后，决策者个人的价值观对决定其决策风格是很重要的。价值观是：对世界的看法和全面的信仰，它指导个人的行动、看法以及期望的结果。一个人的价值观不与任何特定的对象或形势相联系，是可以根据一个人的所有行为方式归纳出来的。此外，价值观从一出生开始就已获得了，而且在人的一生中保持一种强大的力量。为此，个人价值观形成一个永久的框架，不断地影响个人的行为及决策的方式。

这三种因素在决策风格形成过程中纠缠在一起、交互作用的复杂性是非常明显的。然而，我们使用已经建立起来的决策风格的测度，可以建立一个决策风格的分类方案，来为我们设计给持有某种特定决策风格的用户在特定问题背景下使用的 DSS 服务。现在我们就来看看决策风格的类别及其相应的特征。

图 2-4　决策方式模型

资料来源：Rowe，A. J.，Boulgarides，J. D.，(1994)，Managerial Decision Making，Englewood Cliffs，NJ：Prentice-Hall。经作者许可引用。© 1985 年，版权所有。

图 2-4 显示的就是来自于著名的精神病专家 Carl Jung 的决策风格分类图。这张分类图是很多其他决策风格测量的基础，这其中就包括非常流行的 Myers-Briggs 类型指标试验 (Myers 1962) 和 Alan Rowe 发展的决策风格清单。就像图中所显示的那样，决策风格的分类依据两个组成部分：认知的复杂性和价值导向。用这两维坐标，把决策风格分为如下的四种类型：

指令型

指令型 (directive) 的决策风格对结构有很高的要求,背景中对含义的模糊性容忍度较低。具有这样特征的决策者对技术性的决策更为关注,通常,他们并不需要大量的信息或考虑大量的备选方案。他们通常被认为是有效率的管理者,但安全和身份都需要在可见的层次上。最后,通常认为指令型的决策者的口头交流能力比起写作或其他多种媒介是最好的。

分析型

分析型 (analytical) 风格对背景的模糊性有较高的容忍度,而且,决策过程中需要大量的信息,考虑许多备选方案。分析型的决策者最擅长的就是 :处理新的、而且常常是未预料到的环境和问题背景。他们只是以解决问题为乐!与指令型的决策者相比,分析型的更喜欢用笔头来交流,而且并不急于做出决策。他们对细节的关注往往会拖延最终决策之前对问题背景研究的时间。

概念型

很像分析型的决策者,概念型 (conceptual) 的管理者对背景的模糊性有很高的容忍度,但倾向于从多个人的角度对问题进行分析。概念型的管理者对其下属往往表现出开放的心态,通常是强调价值观和伦理道德的唯心论者。这类决策者也是"长期的"思想家,通常对组织非常忠诚。另外,这类决策者非常注重成就感、价值观得到推崇以及组织的认可。他们的管理风格通常是参与者,而且放松控制。顾名思义,概念型是思想家而非实干家。

行为型

决策风格的第四种类型——行为型 (behavioral),虽然对认知的复杂性要求比较低,但他们对组织负有很深的责任,是雇员导向的。这种类型需要相对较少的数据量输入,故具有相对短浅的眼光。行为型的人本性就反对冲突,倾向于依靠会面、舆论等与下属交流或组织起来。

怎样分类

Rowe 把决策风格分成四种类型的拓扑,给我们提供了一种依据决策者特定风格来区分决策者类型的方法。但我们还需要把这些知识融入到 DSS 的设计当中去的方法。为此,有几个关键点需要提一下。

首先,我们必须认识到,人是非常复杂的,没有任何简单的分类方法,比如说"决策风格清单"就能彻底反映这种复杂性。即使这样,我们也不能希望把管理者都能列入清单中的某一类。通常管理者的反应会指示出他们目前占主导地位的决策风格,但是也还有一个或两个后备风格。因此,我们最好是把管理者决策风格理解为"风格的方式 (style patterns)",而不是确定的、可以预测的严格行为的集合。尽管这样,知道一个人占主导地

位和备选的决策风格可以为设计界面和指令提供大量有用的信息。

背景/决策风格交互

表 2-1 给出了多种决策风格的比较以及四种类型的行为结果，这些在为特定类型的决策者设计特定的 DSS 时可以提供信息。

表 2-1　决策风格行为的一般特征

基本风格	压力下的行为	动　机	解决问题的战略	思想的性质
指令型	暴躁的、无常的	权力与地位	政策和程序	集中精力
分析型	集中于规则	挑战性	分析和见识	逻辑性的
概念型	古怪、不可预知	认可	知觉与判断	创新性的
行为型	回避	同辈的接受	情绪与本能	情绪化的

我们关注决策风格的关键论点就在于：决策风格与 DSS 的设计与使用相关，它强调决策者的特定的反应以及问题解决的一般方法。如果我们在设计某一专为某已知决策风格的决策者使用的 DSS 时，使用所有这些特征的知识，那么，我们就能向用户提供与他们期望的方法相一致的支持。反之，如果我们忽略掉决策风格，或者是没有为其提供适当的支持，DSS 就可能实际上阻止了过程的成功，同时增强了决策者决策问题方法的偏差和弱势。一些简单的实例可以说明这一点。

现在我们来说一个使用 DSS 的用户，其主导决策风格是分析型的。我们知道这类用户对模糊性具有很高的容忍度，在做出决策的过程中考虑大量数据，努力找到优化后的方案，具有创新性，而且通常是在一种结构化的、规则导向的环境下。这类用户使用程序化的 DSS 往往能够很有效率，而且在使用 DSS 的过程中往往需要获取大量的数据资源与模型。考虑创新性与违反直觉分析，并结合模型的系统，对分析型管理者是最有用的。

相反的情况是：特定决策风格与系统配对时，系统的设计与那种风格的显著特点相反。考虑一个高度程序化指令型的系统被一个决策风格是高度指令型的管理者使用，指令型管理者对模糊性的容忍度较低，而且，并不善于处理紧迫的情势。他需要控制力与权力，而且，需要一个这种控制力在组织中得以实施的界面。DSS 系统中构造一个精确的、语义准确的询问或命令的乏味性对指令型决策者来说，不久就会消失。另外，指令型采用最原始的以口头传递信息的形式，这需要对结果进行情境分析，把来自 DSS 的指令性的响应解释成简单、明了的表格或叙述性文字。总之，把系统设计得与用户决策风格相互补，能够在决策过程中提供额外的支持，如果没有这样做，就会阻碍过程的成功性。

2.3　决策的有效性

何以谓之“好的决策”

在过去的汽车行业中有个古老的说法，“一笔好生意”就是你刚好同意的生意。这可以推出一个假设就是：购买者从不会故意同意“一笔坏生意”，因此，如果他们同意了，就

一定是笔好生意。我们可以把一笔生意当成一个决策,而且可以假设购买者同意购买的决策就是好的决策,但我们怎么知道一个决策是否为好的决策呢?

假设做决策是管理活动的核心,好的决策与组织长期的成功高度相关,每个人都希望找到知识的宝库,能对好的决策进行剖析,找到成功决策的决定因素。但不幸的是,这与现实差距太远。大量研究表明,能改进决策成功机会的有用的规则几乎没有。那么,我们的面临就是:定义出"好"决策,然后用定义去判断决策是否为好的决策。

下面给出了"好"决策的一个定义:"好"的决策就是在问题背景的边界和约束条件下,达到需要提升的目标结果的决策。换句话说,如果我们在环境的限制下,达到了目标,那么就可以说我们做了个"好"决策。另一种理解是:我们的决策解决了已有问题而未再造成新问题,那么我们的决策就必定是"好"的。问题就是,我们只有在做完了决策,才知道它是不是个好决策。因此,我们就需要根据决策的目标对我们即将做出的决策进行分析。这也是为什么决策通常很难做出的原因之一。下一章,我们会进一步探讨这个问题,那时,我们会考虑与决策相关的各种理论。这里,我们需要进一步理解影响问题背景的各种力量,这些力量通过对问题背景作用,进一步影响决策过程与决策结果。

决策的力量

有关解释作用于决策问题背景,并在决策过程中进而影响决策者的各种潜在力量和约束条件的模型,已经提出了好几种。每种模型都指出,决策者在决策过程中要权衡各种力量,而且,在形成和实施最终决策时,一定要进行动态的调整。图 2-5 描述了这些力量,尽管每个问题的背景都不同,但图中列出的多数(如果不是全部)力量都是典型决策过程中出现的。下面让我们简单回顾一下在典型决策过程中出现的大多数力量和约束条件。

图 2-5 决策者的驱动力量

个人的和情绪的

决策者并不是生活在真空中的,他们一样也有人类普遍的需要。因此,与他们的感

情、健康、安全、报酬、挫折以及忧虑相关的情况，可能是最重要的。认知的局限性可能在挑选决策方案的过程中是非常有影响的。个体力量（personal force）能增强或者弱化管理者做出准确决策的能力，为此，决策者在制定决策战略时一定要认真考虑个体的力量。

人类思想的能力和力量通常是无限的，然而，我们理解人类认知过程的实践表明，我们目前存取和处理知识的能力是有限的。与知识进步的速度相比，我们只是拥有少量知识，而且，研究表明，我们处理已拥有的知识的能力是有限的。Miller（1956）提出了这样的概念：不可思议的数量 7，加或减 2。我们一般都不能同时处理多于 5 到 9 条截然不同的信息。为此，决策者在构造决策战略的时候，一定要仔细考虑作用于决策者的个体力量。

经济/环境

这一类的力量及约束包括了资源限制、政府管制和社会价值观，比如：对要考虑的决策的伦理观或道德观、市场上的竞争压力、顾客的需求、决策结果潜在影响的利害关系个体的需求与需要，以及新技术的出现。单个或混合作用的力量要求决策者以某种方式对导致考虑影响的最终决策的变更做出反应。

1994 年，Intel 发现在它的新奔腾处理器当中存在一个缺陷，会造成某个计算的不精确（见图 2-6）。其根本问题是简单的除法运算需要精确到 16 位有效数字造成的。Intel 的执行长官决定把这个问题告诉给所有的用户。但是实际上，可能发生问题的概率很小，对一般用户来说，只在使用 25000 年才会发生一次，而且，这并不需要重新制造或改造。然而 Intel 在做这个决策的时候，没有考虑到某些市场力量。IBM 是 Intel 处理器最大的客户，而且库房里有大量 486 处理器的库存。市场对新的奔腾处理器非常看好，因此，486 处理器已经无人购买了。当 IBM 得知奔腾机器内存在缺陷的消息时，就开始公开开展商业活动诽谤 Intel 奔腾处理器，宣称会每 24 天发生一次，一般的用户最有可能受到影响了。尽管任何一方都没能提供证据，但是公众开始对奔腾失去了热情，开始转向购买 486——这正是 IBM 高兴发生的。Intel 最初宣告只是想更换处理器，IBM 的反击却对消费者造成了很大的影响。因为 Intel 没有考虑到与竞争相关联的环境因素，而随着新技术的出现，它除了失去宣传新的奔腾处理器的契机外，还造成了第 4 季度 4.75 亿美元的经济损失。尘埃落定，Intel 从他们“糟的”决策中吸收很多教训。尽管如此，这还是被经常作为一个需要考虑经济/环境力量的典型实例。

组织的

在制定决策的过程中，管理者所在的组织的各种力量与约束因素都应该考虑。政策与程序、群体的一致、组织文化、全体职员的协调性以及其他一些资源都可能影响决策过程。管理者与上级和下属相合的程度，加上组织的全部的政策结构，能够明显地改变一个决策过程，进而改变决策结果。假设一个组织趋向于不对新的或创新的想法给予鼓励，而是奖励那些顺应已有的政策或程序的行为，那么这个组织的决策者很快就会对创新性的决策结果不断拒绝来进行破坏。随着时间的推移，组织决策过程忽略创新的现象就会形成习惯，而且，管理者所做的决策就会被组织缺乏想像力和预见性的组织文化所束缚。

FROM：Dr. Thomas R. Nicely
Professor of Mathematics Lynchburg College
1501 Lakeside Drive Lynchburg，Virginia 24501-3199
Phone：804-522-8374 Fax：804-522-8499
Internet：nicely@acavax. lynchburg. edu

TO：Whom it may conern
RE：Bug in the Pentium FPU
DATE：30 October 1994

It appears that there is a bug in the floating point unit （numeric coprocessor） of many，and perhaps all，pentium processors. In short，the pentium FPU is returning erroneous values for certain division operations.

For example，1/824633702441.0 is calculated incorrectly （all digits beyond the eighth significant digit are in error）. This can be verified in compiled code，an ordinary spreadsheet such as Quattro pro or Excel，or even the Windows calculator （use the scientific mode），by computing

（824633702441.0）*（1/824663702441.0），

which should equal 1 exactly （within some extremely small rounding error；in general，coprocessor results should contain 19 significant decimal digits）. However，the Pentiums tested return 0.999999996274709702 for this calculation. A similar erroneous value is obtained for x*（1/x） for most values of x in the interval 824633702418 < = x < = 824633702449，and thoughout any interval obtained by multiplying or dividing the above interval by an integer power of 2 （there are yet other intervals which also produce division errors）. The bug can also be observed by calculating 1/（1/x） for the above values of x. The Pentium FPU will fail to return the original x （in fact，it will often return a value exactly 3072 = 6 * 0x200 larger）.
The bug has been observed on all Pentiums I have tested or had tested to date，including a Dell p90，a Gateway p90，a Micron p60，an Insight p60，and a Packard-Bell p60. It has not been observed on any 486 or earlier system，even those with a PCI bus，if the FPU is locked out （not always possible），the error disappears；but then the pentium becomes a “586SX”，and floating point must run in emulation，slowing down computations by a factor of roughly ten. I encountered erroneous results which were related to this bug as long ago as June，1994，but it was not until 19 October 1994 that I felt I had eliminated all other likely sources of error （software logic，compiler，chipset，etc.）. I contacted Intel Tech Support regarding this bug on Monday 24 October （call reference number 51270）. The contact person later reported that the bug was observed on a 66-MHz system at Intel，but had no further information or explanation，other that the fact that no such bug had been previously reported or observed. . .

图 2-6 一封有关 Intel 奔腾处理器除法问题的电子邮件

资料来源：Dr. Thomas R. Nicely **和** Lynchburg **大学。**

背景的和紧急的

决策过程中一个最明显的限制与因果来源就是决策问题的背景本身。在我们所列出的清单中发现，技能、所需时间、决策动机以及洞察力对决策者都是非常重要的。也许，这其中最重要的，也是最能引起大家注意的就是所需时间。即使作用于环境的其他所有因素都得到了有效的处理，时间上的限制还是会对决策者产生很大的压力。如果不注意或忽略这个因素，显然就会增加错误或劣质决策的可能性。认知性要求高的活动通常如此分类，并不是因为活动本身要求高，而是因为活动中每个决策都有时间上的限制。考虑一个飞行员必须要做出多个决策，涉及到的活动包括决定：上升还是下降、向左转还是右转、增加还是减少能量，这些好像它们都能自动完成，对认知上没有苛刻的要求。但是，当这

些决策是由一个以 350 英里/小时飞行的飞行员,迎面有一同样速度飞行的飞机就要接近时做出的话,时间上的限制就完全改变了决策过程的性质。这只有二三秒的时间就使得飞行员考虑所有相关信息或多个决策策略变得不可能。记住这种情况并没有越出我们讨论的管理决策的背景范围。飞行员的办公室就是他的驾驶员座舱,他所做的工作就是代表他所在的组织做出这种类型的决策。在某种相对的意义上,地面上的管理者面临的许多决策也都是有时间约束的,就像飞行员遇到即将碰撞的情况一样的紧急。

2.4　决策支持系统如何帮助决策

这一整本书都在试图回答 DSS 是怎样提供帮助的问题,但是至少考虑我们前面刚刚讨论过的各种力量,列出 DSS 是如何在决策过程中提供帮助的方式还是恰当的。

回顾第 1 章提到的 DSS,那时说 DSS 非常有用,但在决策过程中要支持所有需要的却不是个通用的解决方案。DSS 并不是替代决策者本身,而只是在决策过程中向决策者提供一个或多个活动的支持。因为这种局限性,我们必须意识到 DSS 能提供决策的本性。然后,我们看到,DSS 技术要和其适用的问题背景结合起来。表 2-2 列出了 DSS 能够提供的决策类型。

表 2-2　DSS 提供的支持的一般类型

- 从多个角度研究决策问题的背景
- 生成多个高质量的可供考虑的备选方案
- 探讨并测试多个解决问题的战略
- 促进头脑风暴以及其他创新性的问题解决技术
- 对于特定问题背景,探讨多种分析方案
- 对不断弱化的偏见与不恰当的试探法进行指导与约简
- 增强决策者处理复杂问题的能力
- 改进决策者的反应时间
- 阻止未成熟的决策与方案选择
- 对大量的、不同的数据进行控制

表中所列的支持的类型并不总是互相排斥的,也并不总是适用于特定的问题背景,这也正是设计、实施与应用 DSS 如此具有挑战性,与其他所有的信息系统不同的原因。然而,正确地应用 DSS 在创造“好的”决策方面是基本的要素。

2.5　小　　结

本章说明,纵使所有人生来就平等,但决策者不是。决策者有许多不同的形式,有些是个体,有些来自群体,还有一些是来自团队或组织。每种类型都面对着不同的问题背景,而且每一种为做出某个特定的决策,都需要各自独特性的资源及支持。与不同类型决策相关联的独特特征与各自的问题背景相结合,就能增强对一般性的 DSS 并不存在这一陈述的推理。下一章我们开始研究决策过程及其所使用的各种模型。

复习题

1. 请列出决策过程的主要组成部分。
2. 定义刺激因素，并描述其在决策过程中处于怎样的位置。
3. 描述一下决策者在整个决策过程中所扮演的双重角色。
4. 在决策过程中，DSS 在哪些部分对决策者有很大的帮助？为什么？
5. 简要概括决策者的类型。
6. 给出个人决策者的定义，并描述一下哪些特征会影响个人决策者的决策方式，
7. 具体阐述一下群体和团队决策有何不同。
8. 识别组织层次上决策者的特征，DSS 中的哪种类型是专为其设计的？
9. 描述影响某个特定的个人决策风格有哪三种力量？
10. 简单描述基于问题背景、认知复杂性以及价值导向考虑。决策风格有哪些类型？
11. 为什么理解决策者的决策风格对设计和实施 DSS 是非常重要的？
12. 是什么使得决策成为好的决策？
13. 简要描述在决策过程中作用于问题背景和决策者的力量有哪些。
14. 个人与感情的因素在决策过程中是如何作用的？
15. 组织是如何推动和限制决策过程的影响的？
16. DSS 对于决策者来说，是怎样的一个角色？

讨论题

1. 在决策过程中，你认为哪些要素是非常重要的？请说明理由。
2. 讨论一下：多人决策者与群体决策者各有哪些特性？各举出一个例子。它们之间有何不同？
3. 观察你熟悉的一个组织当中的决策者，基于问题背景、个人洞察力和价值观的特性，确定他的决策风格属于哪一类。
4. 我们都知道：设计系统是对决策者决策的辅助，能够对决策过程提供额外的支持，如果做不到这一点的话，就会限制决策过程的成功。请找一个案例来支持这一说法。
5. 分析一个由组织做出的好的决策。讨论一下，为什么那是一个好的决策。决策者是怎样来决策的？

第 3 章

决策的过程

学习目标

- 从不同角度理解制定决策的困难性，比如问题结构、认知的局限性、决策结果的不确定性以及可选的多目标
- 学习决策的分类，并理解这些类型在决策支持系统设计中的重要作用
- 理解 Simon 的问题解决模型
- 学习理性决策理论
- 理解 Simon 的“可满足”策略与有限理性
- 区分问题与表征
- 熟悉选择的过程
- 理解决策者的认知过程及其对决策过程的影响
- 学习四种最常见的启发式偏好以及它们对决策过程的影响
- 区分效果与效率

小案例

菲利浦石油公司的产品定价

美国菲利浦66公司是菲利浦石油公司附属的子公司。在1984年11月到1985年6月间，菲利浦石油公司面临着将被两家公司收购的困境：其中之一是T. Boone Pickens，另一家是Carl Icanhn。同时，菲利浦石油公司资产值减少了22亿多美元，负债却达88亿多美元，从而，其负债比率从20%提高到了80%。

美国菲利浦石油公司的总裁Bob Wallace为应对财务危机，开始精简组织，裁员40%（包括管理层）。而且，他还认识到改进获取信息的渠道对公司度过危机是至关重要的。Wallace支持开发并实施（经理信息系统EIS），希望在各个层次上提高管理者对信息的关注以及处理信息的时效。

Wallace知道新的DSS最起作用的决策活动之一就是产品定价过程中的决策。在石油天然气工业中，定价可能是最关键的、时间价值最明显的业务过程。市场状况要求管理者每天都要对产品进行详细定价。在实施EIS之前，定价是集中进行控制的，更重要的是手工完成的。通常先把地区性的销售数据汇总起来，形成区域性的销售状况数据，再考察一下市场上竞争的油泵价格，那么，再把这些数据提供给集中定价的决策群，于是，他们就用这些信息定出一个国家内三个主要区域的价格。而每个区域内的所有的服务站，油价都是一样的。通常要花3～4天的时间，这种价格的影响才能遍布整个地区。而且，很多情况下，基于地区的信息可能是不确切的或是根本上就是已经失真的信息。如果定的价格少了1个美分，那全年累计起来，就可能达到4000万美元的损失。因此，这项决策活动对菲利浦公司来说非常重要。Bob Wallace清楚，如果不为改进这些过程提供支持的话，就很有可能造成代价高昂的重大错误。

Wallace也知道定价部门的决策者已经是行业内最优秀的一批人了，试图通过改进他们的技能，进而提高定价的准确性是不可行的。惟一能改进价格决策过程的效率和效果的方法就是通过战略上部署计算机与交流信息的系统来支持决策者，即实施DSS。新的系统与商品市场上各个地区每天每个服务站竞争对手的价格信息相整合，菲利浦公司来自地区内各个服务站的价格信息，也自动进入EIS。信息经过分析以后，就向决策者提供一系列的报告，并给出地区内有竞争力的定价。它所提供的报告是一系列易于理解而且具有高度准确性的报告。

通过引入EIS，菲利浦公司的管理者获得了必要的信息和分析工具，也就使得他们能够准确而快速地协调定价活动，进而形成每天的定价策略。另外，管理者有必备的交流报告，确保了定价信息能够迅速、可靠、准确地传遍整个公司。

EIS通过改变公司以前集中定价的方法，完全改变了菲利浦公司商业运作的方式。在新系统下，液体甲烷在公司82个不同输送管道终端实施了不同的价格。这就使单个管理者对定价活动负起控制的责任，而且，可以对特定的市场变化做出迅速的反应。顾

客感到，菲利浦66公司对市场反应敏锐，而其竞争对手面对新的价格结构好像是束手

无策，难以作出反应。

据公司市场部副经理 Charlie Bowerman 保守的估计：由于系统对顾客需求及不断变化的市场状况能迅速地作出反应，因此已给公司带来了 2000 多万美元的利润增长。更重要的是，依靠信息技术的支持，菲利浦公司的决策者改进了关键决策的结果，同时，使得公司度过了危机，成功地摆脱了被其他公司收购的危机。

——Yu-Ting“Caisy”Hung

3.1　为什么决策这么困难

通常我们怎么才能预先知道即将做出的决策有多大的难度？仔细考虑一下，就会发现我们快速把众多决策划归为是容易或是困难的能力，是非常惊人的。如果是决定与朋友在海滩花上一天，还是留在家里去割一天草，我们都能立刻觉察出，决策不是非常难以做出。相反地，如果我们负责把新技术引入组织内部，那就会降低运作成本，同时还能改进产品质量，但这会解雇上千的工人。我们直觉上也知道，这样的决策不同于上面所讲的决策。真正的问题是：我们怎样在还没开始收集信息、未做出决策的时候就知道一个决策是易于还是难于做出？

这个问题的答案依赖于结构的、心理上的、物理的及环境的因素。决策当中的困难可能是无数的复杂性、不确定性、组织与环境的压力以及单个决策者的局限性组合的结果。本章我们将探讨四种单独或者混合作用决定一个未决决策问题相对难易的领域：结构、认知局限性、不确定性以及可选择的多目标。

结构

回顾第 1 章里 Simon (1960) 提出的从完全结构化到完全非结构化的连续分类问题。图 3-1 描述了这种分类，其中包含着在连续决策结构的末端各种决策结构的实例。

Simon 通过讨论决策按程序进行的程度，对整个领域两端之间的过程进行了如下区分：

> 程序化的决策是那些反复地按照一定的程序执行的决策，这类决策都有事先规定好的处理程序，以至于每次决策发生时，不需要重新处理程序问题。非程序化的决策是那些异常的、非结构化的和重要的决策。这些决策以前从未出现过，或是其确切的本质与结构难以捉摸或比较复杂，或是因为它非常重要以至于需要专门处理，故没有事先安排好的处理方法。对非程序化决策问题，系统没有一个特定的程序可以用来处理手边决策问题的情势，而是必须退回到做出明智的、适应性的以及问题导向的行为时所具备的一般能力 (p. 5 ~ 6)。

换句话说，程序化决策因为我们能迅速获得决策所需的各个部件，因而比较容易：我们有制定决策的现成的程序。而非程序化决策问题就很难做出，因为一方面我们做决策

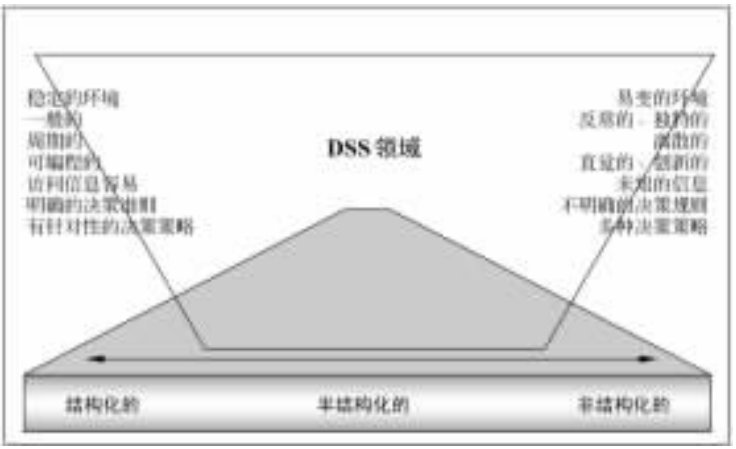


图 3-1 决策的结构化分布

时必须先收集所需的信息,然后才能做出决策;另一方面,我们好必须“写出程序”,或是设计决策的过程。使用 Simon 对 DSS 的分类方法,Keen 与 Scott Morton (1978) 把这个连续过程的两端分别称为结构化决策与非结构化决策。在这里,我们沿用这种约定。

从图 3-1 可以看出,一个决策问题结构化的程度是由很多因素决定的。在正常的活动中,经常重复要做的决策是结构化的决策。飞机飞行员经常要制定起飞、降落、路线选择等方面的决策。这些决策都是平常而又反复做的,因而是相对的结构化的决策;而当飞行员面对一个发动机熄火,他本身对机场又不是很熟悉,身边又没有地图的情况时,所做的决策就成为非结构化的了。

认知的局限性

我们不可能知道所有的事情,有时候,我们甚至不能处理完手边情势所迫不得不知道的所有事情。更糟的是,我们偶尔还回忆不起来以前遇到过或是已经存储在大脑中的信息。同样糟的,有时候即使我们成功地追回了记忆,却不完全准确。尽管人的大脑具有非凡的推理、计算和储存的机能,但我们目前的理解力表明,我们处理和储存信息与知识的能力确实是有限的。

认知局限性的研究当中,最著名的,也是引用率最高的理论之一是 Miller (1956) 的工作。他的研究一直被认为是认知局限性的典型例证。Miller 使用大量的样本,进行了一系列的试验,得出结论:人的大脑快速意识区在处理 5 ~ 9 条信息的时候就受到了限制。换句话说,在一个特定的时候,我们在决策过程当中,只能自觉地明了“不可思议的数字 7 加或减 2”条激活的知识。Miller 用“相当大数量的”这个词汇来描述在“工作内存”里组织好的信息。从本质上来说,Miller 的结论表明,人脑处理信息的能力受到接收刺激数量的限制。同时,也证明了人脑趋向于依据过去的经验把信息组织成大块,专家比起某个知识领域的新手更擅长于处理大块信息。尽管我们已经发明了很多聚合信息的方法以减小局限性,但 Miller 研究的基本结论还是成立的。因为决策过程是一项非常认知性的训练,个体决策者认知的局限性就会从根本上提高特定决策的难度。本章在后面会进一步探讨

认知的局限性问题。

不确定性

“20/20 的后见之明”这个概念是理解不确定性的一个很好的途径。当回过头去看决策结果与决策过程时,可以看到,那样的一个决策过程得出的结果是完全确定的。如果我们能够准确地预测出某特定决策的结果,那所有的决策就都很容易做出来了。

完全确定性意味着与待决策问题相关的完全的、准确的知识。名副其实的不确定性意味着待决策问题的结果,即使在概率框架中也不能确定。不幸的是我们大多数时候面对的都是处于不确定情势下的决策问题,很少存在完全确定的。值得庆幸的是,完全不确定的也是非常罕见的。通常的情况是,决策者可以对预期决策结果赋予主观的概率值,这样就假定某种程度上存在确定性。大多数情况下,这些主观概率准确与否,依赖于过去赋值信息的完备性与准确性程度。所有这些暗含着,大多数决策结果都包含着决策者风险这个因素。在第 5 章,我们将会着眼于帮助决策者赋予主观概率值的各种方法,以及各种用来减少特定决策背景下不确定性的各项技术。在这里,让我们采用如下公理:决策结果对决策者来说越不确定,决策就越难做出。

可选择的多目标

不管决策是在何种背景下做出的,决策的目标都是获得想要的结果。我们可以把决策结果类比成决策过程中制造的产品。事实上,不管决策是个体的、团队的还是组织层次上的,决策目标都是一样的,那就是从备选集中进行筛选,以制造出最大数量想得到的结果以及最小数量不想要的结果。由于我们并不知道任何结果是否会确定发生,所以,我们必须仔细考虑每个与给定决策战略相关的可能结果以确定想要的结果是否能达到。

这里我们要说的是,多个备选方案或是目标能明显地提高给定决策问题的复杂性与难度。每个新方案都要进行透彻地分析,并与其他方案进行比较,很显然,多一个备选方案加入到可选集,就更显著地增加决策难度。决策者在某一时刻可能有多于一个的目标,这就进一步加重了问题。为此,选择出来的某一个备选方案本身,在满足决策目标标准的同时,可能也阻碍了另一个相关目标的改进。现在决策者的共享价值与目标,外加多个备选方案的组合问题,共同使决策过程复杂化了。

管理者用来处理这种情势的常用方法之一就是:在选择过程中尽早放弃那些在可行边缘上的方案。通过这种方法,决策者就能够把主要精力集中于剩下的看起来好像更贴近于想要结果的方案。尽管这个淘汰的过程直观上看起来是做出正确决策的富有成效的方法,但是在本章的后面,我们还会看到,这可能并不正确,事实上,这种方法可能实际上用于依据目标与约束条件排除某些方案被选为最优决策方案的可能性。

3.2　决策的分类方法

世界上有无数个决策,却没有两种完全相同的决策。任何一个人的管理生涯都面临着数量惊人的决策活动。实际上,典型的经理生活的一天充斥着似乎无止境的许许多多

个决策。一个决策所需要的支持种类可能和下一个决策的需要完全不同。我们需要一种分类方法,它可以将大量的、处于不同环境的决策划入一些更容易管理的分类集合中。

决策的分类有许多种不同的方法。我们可以按照管理的层次划分,例如作业层决策、中层管理决策和高层管理决策。我们也可以根据决策不同的侧重点划分,例如战略型决策和管理型决策。通过决策所使用的策略分类也是一个不错的选择,如计算型决策和判断型决策。可见,决策可以从各种不同的角度进行逻辑分类。即使不考虑所采用的分类方法,如果想成功地为一个给定的决策环境设计并实现一种支持技术,我们也必须能够区分(或者更贴切地说是识别)期望的决策支持系统将适用于哪些决策活动,哪些活动又将是它不适用的。总之,我们需要一种决策的分类方法。图 3-2 给出了决策的分类方法的一个例子,它综合了各种从决策过程和决策支持系统的文献中挖掘出的方法。我们现在观察一下这张图所包含的分类方案。

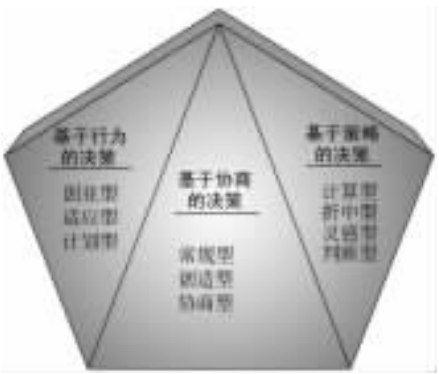


图 3-2 决策的分类方法

基于协商的决策

Delbecq (1967) 提出了一种三类的分类方案,它是从协商的角度考虑的:

1. 常规型决策 (routine decision)。这种决策的一个或多个决策者非常清楚他们希望达到的目标,以及为达到目标所需要采用的策略、步骤和技术。常规型决策类似于 Simon 提出的程序化决策。
2. 创造型决策 (creative decision)。有些决策非常复杂,解决它们需要采用新颖的方法。这些决策缺乏一致性的策略,又因为没有已知的策略和完备的知识,结果也变得不确定。创造型决策类似于非程序化决策和半结构化决策。
3. 协商型决策 (negotiated decision)。对于某些决策,大家的目标和方法存在着冲突。这时,各个对立的派系必须面对彼此,努力解决这些差异。在这些决策活动中,管理者可能根本就不是决策者,而只是决策过程的一个参与者。

基于行为的决策

Mintzberg (1973) 提出了一种注重与决策过程密切相关的行为的分类方法:

1. 企业型行为 (entrepreneurial activity)。这种类型的决策通常具有高度不确定性。

需要主动地考虑如何选择方案，注重短期利益而不是解决长期问题也是这种类型决策的一个特点。

2. 适应型行为 (adaptive activity)。这种类型的决策同样具有高度不确定性的特点，但通常是被动地考虑如何选择方案，它也更注重短期利益。
3. 计划型行为 (planning activity)。高风险是这种类型的决策的环境特点，制定决策既需要主动地考虑也需要被动地考虑。这时更注重长期的利益和效用。

从 Mintzberg 的分类方法可以看出，对于按照这种方式分类的决策，主要应该根据其决策环境确定为它们提供的支持，而不是根据决策本身的结构化程度。从这个角度考虑，Mintzberg 的分类方法和 Simon 与 Delbecq 提出的按照决策的结构化程度分类不同，它提供了一种基于决策策略的分类方法。

基于策略的决策

Thompson (1967) 提出了我们的决策分类学中的另一种分类方法。这种方法主要根据做出最终选择所使用的策略分类：

1. 计算型策略 (computational strategy)。相关的输出和结果/效用相当确定，而且对于可能的输出存在强烈的偏好。
2. 判断型策略 (judgmental stragety)。对于可能的输出存在强烈的偏好，但是相关的输出和结果/效用高度不确定。
3. 折中型策略 (compromise strategy)。相关的输出和结果/效用相当确定，但是对于可能的输出和选择方案的偏好很微弱或者不清楚。
4. 灵感型策略 (inspirational strategy)。对于可能的输出和选择方案的偏好很微弱或者不清楚，同时相关的输出和结果/效用也高度不确定。

我们可以在 Thompson 的方法中找到两个维度 确定性和对输出的偏好。这种策略正是建立在这两个维度上。在这种情况下，对于一个给定的决策，必须根据哪种策略看起来最有效来确定必要的支持。

无论采用哪一种分类方法，我们会发现所有的方法都具有一些共同的特点。正如图 3-1 所展现的，它们可以被更深一层地划分为一类或者两类——第一类具有常规性、反复发生以及高度确定性的特点，而另一类则是非常规的、并不反复发生而且高度不确定。当面临的决策类型具有给定的基本特征时，采用这种元分类方法可以降低决策的复杂程度。换句话说，对于在特定的环境中所面临的决策，如果能够更好地理解其分类方法，我们也将能够更好地确定具有哪些特点的决策支持系统对于当前环境下的决策支持更有效。

3.3　决策理论和解决问题的 SIMON 模型

如果想了解决策者到底在“黑箱”世界中进行了什么操作，就需要通过一些参考点或者“透镜”观察各种不同的表面行为，并且解释它们之间的联系。这个“透镜”通常是以决策过程理论的形式存在的，因此人们提出了许多试图解释决策过程的理论。Keen 和 Scott

Morton 在 1978 年提出了一种决策理论的分类方法，这种方法试图根据观察角度的不同，将大量的决策理论划分成五类。表 3-1 列举了他们根据观察角度划分的五类决策理论，并且为每一类都提供了简略说明。

表 3-1 Keen 和 Scott Morton 根据观察角度划分的五类决策理论

从理性管理者的角度
• 传统概念和制定决策
• 理性假设，信息完备，个人决策者
• 偏好成本 - 收益分析
• 需要对决策变量分析定义
• 需要精确的、有针对性的选择标准
从过程导向的角度
• 注重有效地利用有限的知识和技能
• 强调用搜索法寻找“足够满意”的解决方法
• 决策支持系统设计的目标是帮助改进现有的解决方法，而不是寻找最优的解决方法
从组织过程的角度
• 试图理解作为标准运作过程的结果的决策
• 决策支持系统设计的目标是确定需要支持或改进哪些过程
• 强调对组织角色的识别
• 注重交流的渠道和联系
从政治的角度
• 制定决策的过程被看作是业务单元间讨价还价的过程
• 假设权力和影响决定了任何决策的结果
• 决策支持系统设计的目标更注重对决策过程的支持，而不是对决策本身的支持
从个体间差异的角度
• 注重个人解决问题的行为
• 决策支持系统设计的目标视决策类型、决策背景和将来用户的性格而定

虽然已经存在多种理论方法，开发一种能较好地描述决策过程的方法仍然很重要。例如决策支持系统的设计和实现就必须以一种可操作的、健全可靠的描述性理论为基础。为了我们的目标，我们必须研究并依靠 Simon 和他的同事的成果，把决策过程作为一系列的理性行为 (rational behavior) 来研究。

图 3-3 展示了 Simon 在 1960 年提出的解决问题的三阶段模型①。虽然图形的构成非常简单，但是 Simon 的模型经过了时间的考验。即使在今天，它仍然是大多数管理决策模型的基础。我们发现模型将解决问题的过程描述成一个事件流，它可以是直线型，也可以是反复型。这说明无论位于过程的哪一处，为了更加精确，问题的解决者都可以选择回到

① 虽然 Simon 的模型是在解决问题的领域内构造的，但是制定决策也可以看作是解决问题的一种类型。模型既包含注重抓住机遇的事件，也包含注重选择问题的有效解决方法的事件。此外，Simon 最初的模型并不包含实现阶段。这一步是 Sprague 和 Carlson 后来加上的。

上一步，甚至前几步。我们接下来将更加详细地观察这个过程的各个阶段。

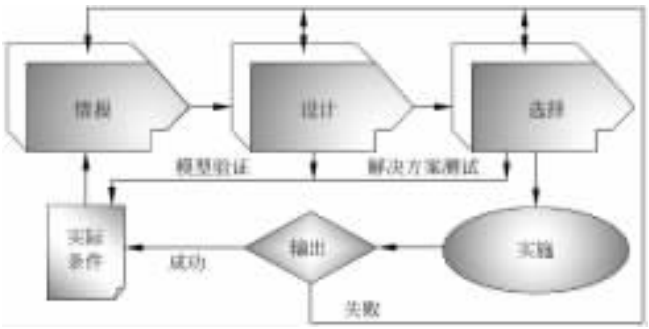


图 3-3　Simon 的问题解决模型

情报收集

这个过程从情报（intelligence）收集阶段开始。在这个阶段，决策者“警惕”地寻找需要的信息和知识，它们表明了问题的存在，或是制定决策时需要的。这个搜寻的过程并没有周期性或连续性的特点。问题（无论真的还是假的）常常表征在当前状态和希望状态的显著差别上。探查出问题通常只是情报收集阶段的搜寻活动的结果。解决问题过程的活动还包括探查出问题解决者“拥有”的问题①。如果决策者以及他的组织不能解决这个问题，那么就不拥有这个问题，也就不会触发过程的下一个阶段。但是在解决所拥有的问题时，这种情况的问题就成为必须考虑的约束或条件。环境条件是这种情况的一个例子，如税率和利率。决策者可能会认为对其进口或出口所征的税率过高。即使察觉出的当前状态和希望的状态存在差别，但是这并不由他直接控制，所以这个“问题”就必须作为解决其他“拥有”的问题时必须考虑的约束或外部因素。

通常能在搜寻活动中察觉出决策者力所能及的范围内的问题。例如，每天检查一遍生产日志将揭示必须提供的维护的情况。商业航班的驾驶员座舱是另一个例子。飞行员间流传着一条古老的谚语，它将飞行描述成“一阵极度疯狂的恐怖打断了数个小时的无聊”。飞行器驾驶舱中的大部分活动都是检查那些普通的装置，这些装置为飞行员提供关于飞行器以及飞行环境的反馈信息。飞行员能够估计出各种环境下的一组信息的价值。如果一个甚至多个装置传递的信息不在估计的范围内，就说明存在问题，飞行员必须离开当前阶段，进入接下来的问题解决阶段②。

---

① 这里的所有权是指问题在问题解决者能够解决的范围之内。一个问题可能被探查出来，但不归问题解决者拥有。

② 我必须对经常提及的飞机驾驶舱做出解释。它们是驾驶它们的经理（飞行员）的"办公室"。此外，它们为各种决策提供了丰富的环境，从没有约束的高度结构化决策到似乎有大量约束的非结构化决策（最重要的约束是时间）。虽然大多数读者可能不会去“管理”一条商业航班，但可以将从研究面临的各种问题中轻松获得的教训带入地面上公司的办公室。

设计

一旦规范地定义了识别出的问题，决策者就必须开始进行下一步的活动。这些活动关系到形成并分析备选方案，这些备选方案是问题潜在的解决方法。这个阶段是过程的关键阶段。必须挖掘出大量的问题潜在的解决方法，并且将它们压缩成可行的子集。还必须仔细分析每一种潜在的解决方法，比较它们的期望输出、成本、成功的概率，以及决策者认为关键的其他指标。

分析并选择解决问题的策略是这个阶段需要执行的另一项关键活动。确定并实现最终选择的方法，是和确定可行方案的范围完全不同的一件事。正如人们所看到的，有时候解决问题的过程是决策中还含有决策，问题中还有问题。这就是做决策通常很困难，并且充满了风险和错误的一个原因。决策者必须确定问题建模、识别模型包含的关键因素和变量，以及确定变量间的相互联系的最佳途径。最后，还必须确定哪一种分析工具最适合检验所选择模型的完整性以及它可能的输出。在这个阶段，决策者可能发现他需要更多关于当前问题的信息，因此可能为了满足这个需要，在转入设计更高层次的活动或进入过程的第三个阶段之前重新返回到情报收集阶段。

选择

凭直觉，这个阶段应该是三个阶段中相关活动最清晰的一个 然而我们很快发现恰好相反。这个阶段仍然是将决策作为一个理性行为，需要从细节上挖掘它的概念。

在这个阶段，决策者从在前一个阶段提出并分析的可行解决方案中选择一个。我们又一次发现决策中的决策的复杂性。例如在决策者挑选问题可行的解决方案之前，他必须再一次地检查内部和外部环境，从而保证将要实现的方案仍然是在当前约束下的最佳选择。虽然在大多数情况下，决策的环境足够稳定，符合期望，但是在某些情况下，决策的环境非常多变，所以我们不能以当前条件为基础进行选择，而应该基于实际实现解决方案时的期望或估计条件。石油工业是这种情况的一个绝佳例子。回顾本章的小案例中菲利浦石油公司所面临的问题。石油和其精炼产品的价格应该在不断变动的环境下确定。即使一加仑石油的价格仅仅存在一美分的误差，也会给公司的未来收益带来数百万美元的损失。此外，确定的价格（解决方案）必须在将来的一段时间内保持不变，所以有时需要对将来的市场情况做出估计。

在这种环境条件下使用二分法时（稳定和易变），通常需要返回到前一个设计阶段，从而进一步改善选择的策略和定义的可行方案集。此外，二分法的内涵是需要做出一个决策，确定可行方案集的构成。在当前情况下是一定需要一个“最优”的解决方案，还是“可接受”的方案就足够了？在选择备选方案时，我们愿意接受哪一种程度的风险？我们需要了解决策环境的稳定性或易变性才能回答上述问题，因此才能选择决策的策略。表 3-2 识别了多种与决策相关的模型和分析策略，既有最优化的，也有满意化的。

表 3-2　策略的模型和分析

满意化策略	最优化策略
激励	线性规划
预测	目标规划
假设分析	简单队列模型
马尔可夫分析	投资模型
复杂队列模型	库存模型
环境的影响分析	运输模型

注意：马尔可夫分析和排队论等技术是确定事件可能性的高级方法。读者可以在任何一本优秀的管理科学教材中找到对于这些方法的详细介绍。

3.4　理性决策

在深入讨论之前，让我们进一步观察一下可以选择的两种策略①。最优化(optimization)的概念清楚地表明了决策者将选择可能是最优的方案，它一般提供最大的价值和输出。最优方案还可以体现在成本最小、目标完成度最高以及性价比最高。如果不考虑结果的结构，最优方案将是“所有可能的方案中最佳的一个”。但问题是我们如何能知道“所有可能的方案”？我们过一会儿再讨论这个问题。

最优化在许多领域中被认为是理性的行为，尤其是在经济学和管理科学领域。换句话说，在检验过所有可行的方案之后选择最优方案是理性的，它将给决策者带来最大化的满意度和效用。“根据经济学理论，理性行为通常的结果是企业利润最大化和人类效用最大化”(Katona 1953，313)。大多数常见的经济学理论都建立在人类行为绝对理性的假设上，认为人们通常选择问题最优的解决方案，例如市场供求的均衡定理。

虽然最优化的决策策略很有吸引力，但是要想在管理决策的环境中实际应用它还存在一些问题。管理决策领域内的很多情况使得最优化即使可能，操作也会非常困难。首先，管理者面对的大部分决策都明显是定性的问题，因此只能采用和最优化相关的定性方法。例如通用汽车公司在 1984 年面临的一个决策，需要确定它的新的占地 400 万平方英尺的 Saturn 制造公司的位置。这个决策就具有复杂、定性的特点。诚然，诸如土地成本、劳动力成本、交通运输问题、必要资源的可利用性以及税收等定量的问题都关系到最终的决策，但是诸如雇员的生活保障金、可以提供的教育以及其他维持生活的服务同样关系到最终的决策。虽然可以准确地计算定量的问题，但是定性的问题却不可以。因此，最优化在这些情况下价值不大②。

Simon 在 1960 年提出了最优化作为一种可行的决策策略的另一个问题。他的角度是不可能为决策搜寻到所有可行的备选方案。显而易见，这样做的成本将会非常高。再说，由于认知能力的限制，人们很难清楚地理解所有的可行方案，因此无法有效地对它们

① 文献中有很多不同的决策策略，主要有层次分析过程 (Saaty 1987)、特征排除法 (Tversky 1972)、逐步完善法 (Lindblom 1959) 和 Etzioni 的混合搜索法 (1967)。但是为简单清楚起见，这一节将主要讨论两种常见的策略类别。

② 最终将 Saturn 的总部设在田纳西州的斯普林希尔。

进行比较（记住，7 加上或者减去 2）。我们将在下一节讨论一种定义更加严密、也更加实际的理性决策方法。这种方法适合实际操作，似乎是一个能够对我们所听到的那些发生在“黑箱”中的行为做出更加准确的描述的模型。

3.5 受限制的理性

为了获得一种关于解决问题的行为的理论，它和理性模型相比应该更加适合于实际操作，我们再一次关注诺贝尔奖获得者 Herbert A. Simon。Simon 应该为他“破坏”了理性的、经济的人的概念受到大家的称赞。他早期的研究就对普通经济模型中最大化和最优化的概念提出了质疑。Simon 提出：人们是利益最优化实体的概念是站不住脚的，否则将不存在价格尺度。试想：如果人们真的想使利益最大化，那么为了使价格尽可能高，每一笔交易都将成为一场谈判。如果有些买主想以低一些的价格买进产品，其他人则可能愿意出更高的价格。理论上在一段时间后将达到最优的利益结果。所以一个“经济”的人为什么会不这么做呢？

但这样太难了，这就是人们为什么不这样做的原因。Simon 认为，人们的认知局限使得考虑一个专门问题所有可行的解决方案不切实际。即使我们能够考虑所有相关的备选方案，我们也不可能理解所有的信息，所以无法做恰当的决策。因此，Simon 提出了一种“简化实际”的理论，我们应该集中精力找出一种可以接受的解决方案，它能满足所有预想到的需求。一旦找到这种方案，我们就应当马上采用它，而不是继续寻找更好的方案。在价格尺度的例子中，当我们确定了一个可以接受的毛利后，就公开产品的价格，反映出它的毛利。虽然这样做将得不到最优的利润，但是能够满足我们销售的利润标准。和对每一笔交易都进行谈判以求得到一个确定的结果相比，这样做将简单得多。Simon 以“满意”的形式提出了这种有限搜索策略。进一步，决策者由于人类普遍的认知局限，实际做出的理性决策常常受到诸多不可控约束的限制，Simon 称其为“受限制的理性（bounded rationality）”。

问题空间

有一个例子很好地解释了 Simon 的受限制的理性的概念。图 3-4 包含了对一个典型的问题空间（problem space）的图形解释。

假设左边的图形展现了当前识别出的问题所受到的限制，以及问题所有可行的解决方案：好的、坏的、很好的、不错的，等等。此外，问题空间也包含了当前问题“最好的”或者最优的解决方案。决策的理性模型指出：问题解决者必须找出并检验问题空间的所有解决方案，直到所有的解决方案都经过检验和比较。这样将寻找到并识别出最好的解决方案。

Simon 认为在典型的决策情景下这种情况不会真正发生。右边的图形展现了他的方法。决策者真正要做的是开发一个模型，它包含了问题看起来可行的解决方案。然后他搜索问题空间，找到一个和预想的解决方案模型足够匹配的解。一旦找到就结束搜索，完成这个解决方案。这样，决策者的选择很可能不是最优的解决方案，因为搜索的问题空间缩小了，使得永远也遇不到最优的解决方案。使用这种方法，决策者将采取给定问题令人满意的解决方案，而不是继续寻找最优的解决方案。

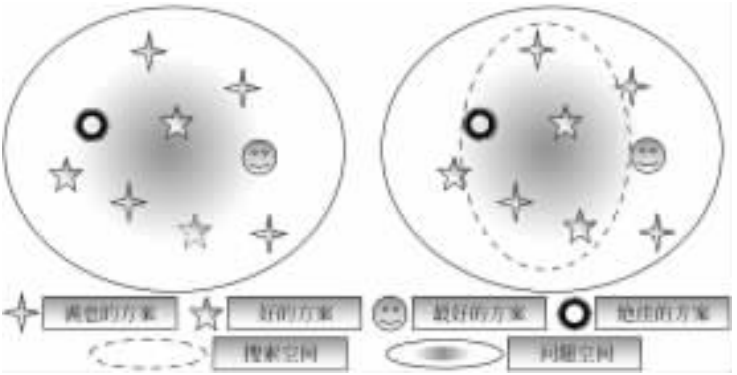


图 3-4　问题空间和搜索空间

但是这又和决策支持系统的设计和使用有什么关系呢？首先，我们发现人类决策者不可能花费所有的精力去收集和给定决策相关的可用信息。其次，即使假设决策者收集了所有可用的信息，他也不可能全部理解，因此无法使用这些信息。最后，根据受限制的理性的概念，典型的决策者似乎在开始搜索解决方案之前，就希望能够预想到解决方案的结构。所有这些都说明了在识别问题和选择满意方案时对指导和结构的需求，也就是对决策支持系统的需求。

另一个需要讨论的重要问题是决策者在探索问题空间和选择最终方案时使用的策略。必须有一种方法能够“度量”所遇到的预想模型中的每一个解决方案。Simon 提出：决策者为了有效管理他们的搜索，应该开发一个启发式规则（heuristics）的集合，或者“拇指规则”。这些启发式规则来自经验、理解、观点、直觉、偏好等。虽然提出这些启发式规则是为了提高搜索过程的效率，但是它们反而常常降低决策者的效果。我们将在本章的最后一节详细讨论这个问题。

问题和症状

典型的决策者容易混淆症状和问题，这是受限制的理性对于决策过程有负面影响的一点。不理解它们的区别，没有区别它们的意识，可能会导致灾难性的后果。

正如前面的定义，问题表现在希望的情形和察觉到的情形之间的差异。如果你感到头痛（察觉到的情形），而你又的确不想这样（希望的情形），这就说明你遇到了问题。但是什么是症状呢？

对症状的注意是以起因和影响为基础。让我们回到头痛的例子。这个例子非常简单，却是理解概念间差异的有效途径。感到头痛是不希望的情形，所以被归入问题的类别。我们确定这个情况是不希望的以后，就想采取行动“解决问题”，也就是消除头痛。结果是去看医生。这时，症状和问题间的差别将逐渐清楚。

如果医生选择将头痛看做一个问题，他将为病人开药，例如阿司匹林，头痛不久就会消失。一切似乎都好转了，直到第二天我们坐在教室里开始抄黑板上的笔记。突然，头痛又复发了。我们已经知道了头痛和阿司匹林之间所希望的关系，所以我们又服用了一些阿司匹林，头痛再一次消失。不幸的是，第三天它又复发了。问题出在哪儿？

“问题”在于受限制的理性允许我们暂时对“治疗症状”满意，而不去“治疗疾病”。

头痛仅仅是表象,它隐藏了一个更深层的问题 我们需要眼镜!这儿就存在症状和问题的差异。症状是问题存在的证据,但绝对不是问题。可以把症状想像成仅仅是正常情况的背离。这样定义的症状不必像症状一样坏。如果在审查周期性的预算和实际的报告时,检查者发现有一个特殊科目的成本超出了预期,这就暴露了一个问题。进一步的审查显示,那项高成本科目涉及的产品在某个地区的销售额明显提高。消息听起来很好,这个"有利的变化"会触发潜在问题的出现吗?可能不会,但是它应该触发。这两个环境下的变量都偏离了正常预期,这些偏差至少应该引起更加详细的审查。虽然它们看起来可能不相关,或者说起源于两个完全不同的部分,但是它们可能恰恰是高度相关的,并且指出了一个更深层次的问题。假设只是因为一个高度热忱的市场经理大幅度地将该产品的价格降低到市场价格以下,这导致了这个地区高于预期的销售。如果没有一个用来增加市场份额或者和当地其他的卖主竞争的合理的短期策略,这种降低价格的行为是不可取的。再说,这种销售额的突然增加也导致和制造该产品相关的材料成本突然增加。如果把每一个变化都当作核心问题诊治,那么真正的问题可能反而得不到解决。进一步说,依靠治疗症状的方法,症状可能仅仅是消失,但却隐藏了需要进一步调查的问题。

可见,清楚地区分问题和它相关的症状对于成功的决策来说极为重要。我们可以把症状定义为一个内在起因的表面影响。真正的差别在于和消除症状相关的结果。对症状的治疗通常能够消除症状(影响),但不能消除相关的问题(起因)。反之则不然。如果我们愿意花费时间和精力去完全识别并定义根本的问题,则会得到一个不仅能消除问题,还能消除所有相关症状的解决方案。我们将在第 5 章探讨决策支持系统如何帮助决策者通过症状解决问题。现在只需要记住 如果拥有眼镜,我们将看得更清楚,并且不再头痛 如果只是服用阿司匹林,我们只能在短时间内消除头痛,但永远都不会看清楚。

3.6 选择的过程

从某种意义上说,问题解决模型的选择(choice)阶段是整个决策过程的最高点。Simon 解释道:

> 整个过程的顶点是决策本身。无论问题、备选方案、决策辅助工具和随之而来的结果是怎样的,一旦做出决策,就有事情要发生了。决策触发了活动、转移和变化……所有这些现象都有一个共同的重要特点。其中,决策者是……在选择的刹那,准备好在这条或另一条路上做出引导人们走出歧途的脚标。所有的现象都通过将注意力集中到最终的那一刻歪曲了决策。他们都忽略了长期的、复杂的定义、探索和分析过程,正是这些过程引导出最终的那一刻。

换句话说,如果我们的目标是把所有的精力都集中在选择阶段,我们就不能对整个过程做出正确的判断,因此很可能也无法做出一个好的决策。要在解决问题前先识别问题,以上只不过是 Simon 指出其重要性的方式。

选择阶段主要关注那些从半结构化到非结构化类型的决策。再说一遍,如果一个决策是高度结构化的,那根本就不算是一个决策。有不确定因素的地方才有真正的选择。

决策者面临着选择，备选方案源自一个主要靠判断的决策策略。可以用定量的模型比较并评价那些方案，而且在某些情况下可以降低决策者面临的不确定程度。虽然如此，不确定因素最终还是一直存在的，决策者必须在面对这一点时做出选择。

标准选择与描述选择

理解制定决策的标准方法（normative）和描述方法（descriptive）（或者说行为方法，behavioral）之间的差异非常重要。对于标准方法，选择是其中的一种理论，或者就是标准方法本身。但是对于行为方法，选择只是过程的一步。这个区别对于成功地设计和使用决策支持系统非常关键。选择理论具有一个前提假设：可以从某种确定程度上预计未来的结果，决策者必须在将来可能出现的多个结果之中做出猜测。选择理论使用概率分布来简单地表示未来结果的不确定性。但这些理论不涉及与开发备选方案、确定目的和目标，以及制定决策后的实现相关的活动。虽然这似乎过于狭窄地集中到了过程上，但我们还是必须对辅助标准决策的开发工具有透彻的理解，因为使用计算机辅助决策工具本身就暗含着标准的观点。这并不说明决策支持系统不能支持行为决策的过程，它很可能可以支持。这只是说明当代决策支持系统所提供的大部分支持都明显具有标准的特点，其他大部分与行为方法相关的活动通常不以决策支持系统的方式提供支持。我们将在第 5 章更加详细地探讨这些标准支持方法的优缺点。

可度量的约束

我们在设计决策支持系统时还必须考虑选择阶段的另一方面，那就是大量的可度量的约束对决策者构成的挑战。表 3-3 列出了多种通常和选择模型相关的特点。诸如认知局限、不完备或不准确的信息、时间限制和成本约束之类的障碍，都在选择阶段限制了可行方案的集合。此外，决策者又进一步被他自身的心理障碍约束。为活动过程背负的可察觉的压力，对失败或报复的恐惧，以及个人偏好，进一步增加了选择阶段的约束。“毫无疑问，人类的脆弱弥漫了选择活动，使得整个决策过程的每一点都应该接受推敲和详细审查。”（Harrison 1995，58）

表 3-3　选择模型的普遍特点

• 模糊	决策者对决策任务不清楚的程度
• 复杂	决策任务的各种不同组件的数量
• 生疏	决策者对决策任务不熟悉的程度
• 不稳定	决策任务的构成在选择阶段及以后变化的程度
• 可取消性	如果没出现希望的结果，选择能够被取消的可能性
• 重要	选择对于决策者和组织双方的重要性
• 责任	决策者应对选择结果承担的责任
• 时间/金钱	对于决策过程和可行方案集的约束
• 知识	决策者具有的相关知识
• 能力	决策者的智力和竞争能力
• 激励	决策者对于做出成功决策的愿望

资料来源：摘自 Beach 和 Mitchell（1978）。

3.7 认 知 过 程

迄今为止,我们已经接触了许多认知的过程,但是仍然需要进一步了解其中的一些。记住:人类决策者的认知过程在多个方面都存在局限,这是决策者在初始阶段就需要决策支持系统的根本原因之一。我们在决策活动中需要对适当的认知过程有全面的理解,成功地设计并实现一个决策支持系统在很大程度上依赖于这一点。

认知局限

表 3-4 包含了一系列心理因素,它们是在研究人类决策者的认知局限时确定的。

表 3-4 导致认知局限的因素

- 人类只能在短期记忆中保留少量的信息单元
- 决策者的智力水平和类型存在很大差异
- 决策者有一个闭塞的信念体系,这不正常地限制了信息的收集
- 决策者在信息处理方面往往受到一定的限制
- 决策者对风险的偏好不同,风险偏好者需要的信息比风险规避者少
- 决策者对信息的需求和他的愿望程度正相关
- 一般来说,年长的决策者受到的限制比年轻的多

资料来源:摘自 Harrison (1995)。

从各种因素的列表中可以看出,决策者克服,或者至少弥补这些认知局限的任务非常艰巨。典型的决策者一般使用多种工具,并不限制在选择一种理想方案的常规概念中。努力化简的自然趋势可能是和试图克服这些认知复杂性相关的最重要的工具。

不管在什么环境下对一项活动施加约束,都会增加活动的复杂程度,而且通常会增加成功地完成这项活动所必需花费的精力。在森林或牧场中砍倒一棵大树要比在房屋旁容易得多。在考虑如何砍倒大树时,房屋的存在成为一个重要的约束。只有当知识和分析能够提供一种不对房屋造成任何损失的可行方案,才允许把树砍倒,否则不能砍树。当存在约束时,大多能找到一种达到目标的方法,但通常和标准的方法不同,也有可能更简单。如果知识和分析说明不能克服这个约束,那么仍然可以砍树,只不过房屋受到损失的风险很大。在这种情况下,决策者可能会试图通过简化实际情况来减轻一些和问题相关的认知压力。从本质上说,他可能只是"希望"房屋消失。

上面的例子并不像它看起来的那么牵强。认知局限通常会引发决策者选择一种简化决策环境的方法,直到它的复杂程度比较合理且容易管理。创造一个现实的简化模型(simplified model of reality)是这样做的一个方法。一旦建造出这种现实的简化模型,决策者就能够集中精力去解决模型反映出的问题,而不是现实的问题。如果简化的模型准确地暴露出了问题的主要特点,就能够通过这种策略得出一个可行的决策。反之,如果简化的现实模型是在抛弃了许多细微、但是重要的因素上建立的,那么得到一个成功结果的可能性就大大降低了,这也是通常的情况。这时,可能有人会怀疑一个理性的人怎么可能在砍树时忽略了旁边的房屋。我们将在 3.8 节看到这种看似不合理的行为是如何发生的。我们现在只需要相信它能够发生,而且会发生。

感知

感知 (perception) 是认知局限的一个特例。决策者趋向于按照他们“看到”的去做。这种形式的“选择歧视”可能会在特定的决策环境结构下对决策过程造成限制。而且,内部或外部力量形成的约束和限制的程度,通过感知而增强其重要性。

感知对于决策过程非常关键,因为它是所有现象必须经过的“过滤器”。一旦现象通过了感知过滤器,它们就变得更加符合决策者对现实的看法。换句话说,现实就是一个人的感知。这个现象的过滤器可能会导致决策以感觉到的问题环境,而不是它真实的结构为基础。决策者“过滤”现实的程度决定了给定的解决问题的环境被扭曲的程度。当这种事情发生时,能否成功地做出决策就值得怀疑了。

大量的因素构成了决策者的感知过滤器。决策者本人的独特性就是最常见的因素。感知是经验、个人偏向、目标、价值观、信仰、激励和本能的偏向等等累积的结果。此外,个人的认知局限常常导致他无法准确地感知当前状况。这些感知障碍 (perceptual block) 会成为现象不可渗透的过滤器。表 3-5 列出了多种感知障碍。

表 3-5　常见的感知障碍

- 隔离问题的困难
- 划分的问题空间太接近
- 不能从不同的角度看待问题
- 循规蹈矩
- 充分的、甚至过度的认知

资料来源:摘自 Clemen (1991)。

决策者无法隔离出当前问题是最常见的感知障碍。复杂的症状通常导致识别出潜在的问题相当困难。如果没有一种结构化的方法用来收集和组织症状,那么很可能检查不出问题。如果我们不能将问题隔离出来,我们就无法解决它。

受限合理性是感知障碍的另一种形式。为了使问题符合感觉到的现实,决策者可能会被蒙蔽双眼,制定一个和期望的解决方案相悖的潜在方案。这种感知障碍通常被称作“在盒子内思考”。Adama 于 1979 年提出的经典的九点难题是阐述这一点的绝佳工具。图 3-5 显示了难题的初始布局。

问题是要找出一种方法,用不超过四条直线连接所有的九个点,同时你的笔不能离开纸。你将在本章的最后找到这个问题的“标准”解法,以及很多非常有创造性的解法,它们是 Adams 多年收集的结果。

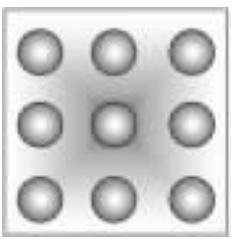


图 3-5　九点难题

资料来源:J Adams, Conceptual Blockbusting, (第 24 和 25 页)。© James L. Adams 1986 年版权所有,经 Addison Wesley Longman 许可引用。

可是这个问题与感知障碍和决策支持系统的设计有什么关系呢?决策者通常会给一个问题强加上实际不存在的约束或规定。这样做的时候,他们为解决方案集制造出人为的界限,严重限制了潜在方案的范围。正如从这个问题的普通解法上看到的,如果要有效地解决问题,就必须保证不制造默认的或不言而喻的约束。在设计决策支持系统时必须认识到

使用者给问题强加上人为约束的倾向,因此设计的决策支持系统必须能够识别出实际的约束,并对人为的或强加的约束提出质疑。我们将在第 15 章详细讨论这一点。

还有一种感知障碍,就是习惯性的循规蹈矩 (stereotyping)。我们都会在一定程度上存在这个问题,它就像一道屏障,使我们无法看清真实的现状。举个例子,假设你安静地坐着,读一本杂志,身心愉快,偶尔凝视着窗外的白云和天空。它们有时在你上方,有时在你下方。你并不为这个世界忧虑,并且感到非常安全,因为你知道飞行员相当专业,而且受过一流的个人训练。实际上,你还回想起飞机升降的平稳,并且把它和你原来乘坐的飞机相比较。

这时你抬起头,注意到机舱内的乘务员正在和一个男士说话。那个男士嚼着口香糖,穿着蓝色牛仔裤和 T 恤衫。他的头发及肩长,还在一个耳垂上带了两个耳环。乘务员似乎在仔细地听着他的每一个字,面带尊敬的微笑。这时他打开门,走进飞机的驾驶舱并消失了。这一切引发了你的好奇心,你向乘务员询问那个男士的身份。“那是 Conway 机长,”乘务员回答,“他是我们最有经验的驾驶员之一。能够成为他的员工通常是一种光荣。”这时你会怎么想?又会有什么感觉?

当时的真相是 Conway 机长的确是一个有经验的飞行员,而且其能力到此刻为止都不容质疑。那么问题出在哪儿?问题在于 Conway 机长的角色并不是你事先预料到的,而且就因为这件事,你开始怀疑所有和飞行员有关的现象。我们认为一个商业航班的飞行员需要身着制服、修饰得体,但他并不是这样的事实改变了现状。你开始为一个高度熟练的专业飞行员焦虑。循规蹈矩仅仅是一个自我施加的对现实的感知障碍。

判断

虽然存在很多关于比较和评估解决方案的策略,判断 (judgment) 似乎是最受欢迎的一种。通过判断,决策者在经验、价值、感觉和直觉的基础上做出选择。和详细分析相比,判断更加快捷、方便,压力也更小,这是它受欢迎的一个原因。如果能被恰当地和其他选择策略一块儿使用,判断将是选择过程的一个有意义的有用工具,而且能为决策的质量做出重要贡献。但是如果单独使用,判断将只是猜测而已。

判断不能在选择过程中单独使用的一个主要原因是,它主要依靠决策者对事件的重复收集和认知。如果能够准确地回顾过去的经验,并且能够根据那些经验正确地识别当前事件的集合,那么在这种情况下可以通过判断有效地确定恰当的选择。但是在大多数情况下,认知局限会使重新收集变得“模糊”,那么判断就存在问题。这时为了确认或否决决策者的判断,我们就应该采用其他的分析策略。表 3-6 列出了多种认知局限影响判断的途径。

表 3-6 认知局限影响判断的途径

- 判断更依赖于感觉和偏好,而不是相关的新信息
- 直觉的判断常常会造成错误的引导
- 直觉的判断可以用于考虑事件发生的频率或目标所占的比例 (见 3.8 节)
- 通过表象给概念分类通常不可靠
- 固执的判断解释了一个人对于结果的考虑

资料来源:摘自 Clough (1984)。

我们将在下一节探讨各种不同的偏好和搜索方法，这些通常是决策者的策略中固有的，而且它们会从感知和/或判断上导致“错误”。

3.8　决策中的偏好和启发式方法

启发式搜索和“拇指规则”方法

无论条件怎样，无论我们是谁，无论我们在制定决策和解决问题方面如何接受很好的训练，在制定决策时我们都会依赖某些“拇指规则”。“7 月 4 号不会下雨”，“只买市盈率低于 9 的股票”和“教授从来不会在第一节课上讲授很重要的内容”等都是我们在制定决策时使用“拇指规则”的例子。这些规则的另一种说法叫启发式推断（heuristics）。heuristics 这个词来自希腊语的“发现”。这个来源十分恰当，因为启发式推断往往是在反复的尝试和失败的经验中发展而来的，这样的经验经历了各种不同的分析和实验。如果一个启发式推断被验证过是可行的话，它就能作为一个可靠的工具，在更便于管理的程度上减少搜索过程。这种方法叫做启发式搜索（heuristics search）或者启发式编程（heuristics programming）。这类搜索方法一般是遵循一系列的步骤，直到得到一个满意的解决方案 这些步骤建立在决策者已经知道的规则的基础上。与按顺序检验所有可能策略的穷举搜索相比，启发式搜索往往成本更低，也更有效率。而且，几个研究者，如 Zanakis 和 Evans（1981），已经证明启发式搜索可以得到与穷举搜索非常接近的解答。表 3-7 列出了启发式搜索的主要优势以及启发式搜索比最优化策略更适用的情况。

表 3-7　启发式搜索的优势和适用情况

启发式搜索的优势：

- 简单易懂
- 容易实现
- 所需的构思时间短
- 只需要较少的努力去认识（对人而言）和较少的 CPU 时间（对计算机而言）
- 可以有多种解决方案

适用的情况：

- 输入的数据不确切或者很有限
- 求出最优解决方案的计算时间太长
- 问题经常或者重复遇到，花费了不必要的时间
- 涉及符号的而不是数字的问题
- 与问题的复杂度相比过于简单的模拟模型
- 还没有可靠的、准确的方法来解决的问题
- 使用好的初始解可以提高求最优解过程的效率
- 最优的解决方案从经济上不可行

资料来源：Zanakis 和 Evans（1981）。

一个用于说明组合的复杂程度的普通例子也可以作为启发式解决方法很好的例子，这就是推销员问题。

问题很简单：一个推销员必须拜访几个城市里的顾客。由于效率和成本的限制，他只能到每个城市一次，而且在他回到原来的城市以前必须到达过所有的城市。因为他的成本主要是与他经过的总路程联系起来的，所以问题的解决方案应该是符合公司政策的最短可能路径。听起来够简单的，是不是？

图 3-6 描述了推销员要去的城市和每两个城市之间的距离。

	纽约	底特律	亚特兰大	迈阿密	达拉斯	塔尔萨	拉斯韦加斯	芝加哥	盐湖城
纽约	-	338	841	1308	1552	1344	2548	802	2182
底特律	338	-	810	1165	1266	1211	2439	748	2110
亚特兰大	841	810	-	665	795	772	1964	674	1878
迈阿密	1308	1165	665	-	1329	1398	2526	1300	2532
达拉斯	1552	1266	795	1329	-	257	1221	917	1242
塔尔萨	1344	1211	772	1398	257	-	1228	883	1172
拉斯韦加斯	2548	2439	1964	2526	1221	1228	-	1772	433
芝加哥	802	748	674	1300	917	883	1772	-	1390
盐湖城	2182	2110	1878	2532	1242	1172	433	1390	-

图 3-6 推销员问题——要到达的城市

问题的复杂程度随着城市数的增加而增加。如果我们假设只能沿着一个方向前进（因为规定了不许到同一个城市两次），那么所有不同路径的数目可以由下面的式子决定①：

$$\text{不同路径数} = 0.5 \times (\text{城市数} - 1)!$$

如果城市数为 9，如图中所表示的那样，那么所有的路径数是 21060。如果我们把城市数增加 1，有 10 个城市，路径数就会跳到 181440 个。再增加一个城市，总路径数就会超过 180 万。12 个城市的话，路径数就上升到接近 2000 万。即使用计算机来处理，这个问

① 注意“!”表示阶乘。$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

题的求解也是非常耗时而且困难的。求解的一个办法是定量的分枝定界法[①]。这种方法虽然有效，但是在处理规模较大的问题时效率却不一定高。用启发式方法可以得到这个问题令人满意的解。一个可能的启发式是："从总部出发，选择最近的城市，然后继续前往最近的城市直到最后一个城市为止，返回总部。"图 3-7 显示了由此得到的解。

图 3-7　初始的启发式解决方案

图 3-8　修正后的启发式解决方案

经过简单的修改，我们改进了初始的解决方案，总路径缩短了 575 英里。更重要的是，我们不需要测试所有的 20160 种方案去寻找一个可以接受的解。在这个例子中，如果能恰当地使用启发式搜索，这种方法将成为一种很有价值的求解工具。

启发式偏好

但是在某些情况下，用启发式搜索去寻求有效的解是会造成危害的。Geofrion 和 Van Roy (1979) 解释说：

> 常识方法和启发式推断会失败，是因为它们的武断。它们在出发点的选择

① 分枝定界法是一种常见的搜索方法。该方法开始时把问题界定在可行域中，然后重复地把原来的界定域分割为子域，在子域内搜索最优解，直到找到一个解或者所有的可能解都被否决。

上,在分派和做出其他选择决策的顺序上,在约束的确定上,在确认条件的选择上,在论证最终结果是最优的或者接近最优的所花费精力的程度上,都是武断的。结果是不稳定和不可预料的——在某些应用上表现很好,而在其他应用上则相反。

最终的结果似乎是,典型的决策者以启发式决策的形式发展出一套简化的现实模型,然后他们依靠这一套规则为当前问题做出决策。如果这些规则是像 Geoffrion 和 Van Roy 所指出的那样被武断地发展而来的话,将不会得到很好的结果。

20 世纪 70 年代初 Tversky 和 Kahneman 发表了一系列的论文,主要论述他们观察到的与武断地发展起来的启发式方法的使用有关的偏好。他们的结果显示,决策者倾向于依靠一套有限的启发式原则,这些原则往往导致他们的思想出现严重的系统性错误。表 3-8 描述了个人决策者表现出的常见偏好。

表 3-8 个人决策者的常见偏好

- 他们往往高估低概率事件,低估高概率事件
- 他们似乎对他们观察到的真实样本的规模感觉迟钝
- 他们不断地根据另外的证据调整他们的初始估计
- 他们倾向于对自己预计事情的可能性的能力过于自信
- 他们倾向于高估别人预计事情的可能性的能力
- 他们倾向于两两比较选择方案,而不是全体考虑
- 他们倾向于尽量减少对直观的权衡或者其他数值计算的依赖
- 他们经常做出前后不一致的选择
- 他们倾向于观察相互排斥的事件

资料来源:摘自 Harrison (1995)。

在 Tversky 和 Kahneman 提出的大量启发式偏好中,四类最常见的是:(1) 有效性偏好,(2) 调整和固定偏好,(3) 代表性偏好,(4) 动机偏好。每一类偏好都关系到对一个有效的决策支持系统的设计和使用。

有效性

有效性偏好 (availability bias) 的产生,是由于典型的决策者不能准确地评估特定事件发生的概率所导致的。个人倾向于根据过去的经验评估一件事情的概率,而经验可能并没有代表性。换句话说,我们倾向于根据我们所能想起的一件事情上次发生的容易程度去判断它将会发生的概率。1992 年 8 月份安德鲁飓风来袭以前,大部分人 (包括我),甚至大部分保险公司,普遍低估了它带来重大破坏的概率。我原以为自己失去拥有的所有东西的可能性非常小。事情很可能是这样的。尽管我低估了这种可能性,但我为我的房子和财产买了全值保险。安德鲁飓风袭击后的第二天,我意识到失去所有东西的概率突然上升到 100%!保险公司也突然意识到这一点了,这让它们大为沮丧。这件事情的关键在于,所有的估计都是基于对上次同样规模的飓风的回忆而做出来的。我从来没见过这样规模的飓风,甚至大部分佛罗里达州本地人都没见过。尽管如此,它确实发生了,只

有很少一部分人对此做好了准备。

图 3-9　1992 年 8 月安德鲁飓风造成的损失

有效性偏好的一种叫相关性错觉偏好。在这种情况下，决策者因为两种事件经常一起发生而错误地认为它们彼此高度相关。1996 年末到 1997 年初，大量黑人教堂成为明显的纵火案的作案地点。看起来这些纵火案显然是全部相关的，但是经过一番深入调查后，没有任何证据表明它们是有关联的。几个嫌疑犯被逮捕并且被判定为纵火者，但是他们之间并没有任何关系。事实上，有几桩火灾事后证实根本就不是纵火的结果。换句话说，证据表明这些纵火案并不是相关的，起码它们不是由同一个人或者一伙人放的。相关

性错觉可以使正确解的搜索过程产生偏差。

对目标数据的结构性回顾和分析可以减少有效性偏好的产生。在缺乏这些数据的时候,决策者至少可以尝试诚实地估计他的回忆潜在的偏好,然后相应地对估计做上下调整。在任何情况下,对决策过程的支持必须包括在合理的地方确认和控制有效性偏好和相关性错觉偏好。

调整和固定

人们在制定决策时往往首先选择一个初始的出发点,然后对出发点进行上下调整,直至得到一个最终的估计。不幸的是,人们常常低估这种调整的必要,保持着偏好,或者说固定在他们开始的估计上。例如,一个人可能估计参加某个会议的人数约为 22000 人。这个开始的估计可能是基于观察或者推测,如知道现场有多少座位然后估计空座位的百分比 ;也可能是基于一个出现在决策者前面的代表出席人数的数字。不管哪种情况,对估计的改变都是以一种增加或者减少初始估计的形式进行的,并且容易趋向初始的估计。由于这种调整不足的倾向,大部分凭主观得到的概率分布都太狭隘,不能估计事件真正的变化。

决策支持系统可用于减少由这种偏好所引起的错误的一种方法,要求用户估计与多个点相关的区间。这意味着,决策支持系统要求用户估计一个从 0. 05 分位点到 0. 95 分位点的区间,而不是从一个点去估计变化①。蕴含在区间估计中的一个事实是,真正的结果可能出现在区间的两端的外面。接着决策支持系统要求用户估计四分位数 (0. 25 到 0. 75) ②。然后用这些估计的组合来决定最终估计的中点。这种方式可以帮助用户尽可能地避免固定在一个中间值上以提高正确性。

代表性偏好

第三种常见的启发式偏好,代表性 (representativeness) 偏好,与相关性错觉偏好很相似。决策者往往根据一个人或者一件事物与一个群体或一类事物的典型概念的一致程度,来确定它们属于某一特定群体或种类的可能性。两者的相似程度越高,两者相关联的可能性就越高。Tversky 和 Kahneman 提出一个简单的问题作为代表性偏好的后果。考虑下面的问题:

> "Tom W. 虽然缺乏创造性,但是智商很高。他很有条理,思路清晰,对每一细节都处理妥当。他的文章显得迟钝甚至呆板,偶尔一些朴素的相关语和科幻想像的火花会带来一点活泼。他的好胜心很强,对别人没有多少感觉和同情,不喜欢和别人交流。他以自我为中心,却有很强的道德感。"

上面对 Tom W. 的性格描画是一个心理学家在 Tom 读高中时以一个心理测

---

① 区间 (Interval) 是指一条线上任何两点或者一个列表中任何两个数字的数学距离。分位点表示从开始点到这一点那一段区间的比例。

② 第一个四分位数以下有四分之一的数比它小,第三个四分位数有四分之三的数比它小,而第二个四分位数有一半的数比它小,一半的数比它大。

验为根据写下来的。现在 Tom W. 是一个研究生。下面有 9 个研究生专业，请根据 Tom 最可能的选择顺序对这 9 个专业排序。

(a) 工商管理　　(b) 计算机科学
(c) 工程学　　(d) 人文教育
(e) 法律　　(f) 图书馆科学
(g) 医学　　(h) 物理和生命科学
(i) 社会科学

(资料来源：Tversky 和 Kahneman，1973)

通常的对 Tom W. 的分类是根据“他是个随波逐流者”这一主观评定。这样，大部分人会把他归类到计算机科学或者工程学。但是，如果我们使用不同的方法去衡量，如根据某一专业的研究生的基本比例，我们可能得出完全不同的结论。事实上，尽管上面提到的两个专业十分受欢迎，更多的研究生是人文教育和社会科学专业的。简单地说，Tom W. 是 (d) 或者 (i) 专业的可能性要比他是 (b) 或 (c) 专业的可能性要大。不考虑基本比例而支持代表性会给决策结果带来危害。

这种偏好的另一种形式是常说的博弈谬论 (gambler's fallacy) 或者样本大小的不敏感性 (insensitivity to sample size)。在这种情况下，人们从具有高度代表性的小样本例子中得出事情发生的可能性，而忽略它们对统计错误的极端敏感性。以一个站在赌场里轮盘旁边的人为例。你走到轮盘旁边，那个人转过来对你低语：“把所有钱押在黑上！”你问为什么你要这样做，他回答说：“听着，朋友。我过去半个小时一直站在这里，红色已经连续出现 15 次。来吧，黑色一定准！”

他犯了双重错误。首先，他没有意识到轮盘的旋转 (假设它是公平的) 是与前后的旋转独立的。这样，红色连续出现的次数与这一次是红还是黑的概率一点关系都没有。其次，他的概率估计是基于小容量样本的。我们从概率论知道，如果轮盘转动很多次的话，两种情况的真正概率会很明显。小容量样本会远远偏离它们的真正意义。

另一种代表性偏好是不能认清平均数回归。Tversky 和 Kahneman 叙述一件佚事来解释这种偏好。他们回忆起一个飞行员教练，他总是为一次成功的着陆夸奖学生，而为一次笨拙的着陆责备他们。他注意到当一个学生被夸奖，下一次着陆总是不如上次好；而一个学生被责备时，下一次着陆往往会大有进步。这个教练得出结论：责备是一种有效的方式，而夸奖则相反。

这位教练的结论很明显是错误的。当一种现象是随机出现的时候，极端的例子往往会跟着一般的例子出现。这是因为真正的平均数处在两个例子之间的某处，概率论表明只要试验足够多次，就能得到真正的平均数。所以一次极好的着陆的后面往往会跟着出现一次没那么好的，而一次极差的着陆后面常常会变好。

动机偏好

到目前为止，讨论过的偏好在认知的根源上是相关的。它们都源于个人在解决问题过程中处理取得的信息的方式。然而，还有一种非认知的偏好对解决结果有着重大影响。

动机,不管是实际存在的还是感觉到的,经常导致决策者估计的概率不能正确地反映他的真实信念。这叫动机偏好(motivational bias)。在动机偏好的影响下得出的估计常常反映决策者的个人利益。这种估计也会来自被认为是特定领域的专家的信息提供者。在这种情况下,决策者必须意识到动机偏好的存在,并且决定在多大程度上相信专家的估计。

动机偏好会在很多场合表现出来。例如,一个销售员可能被要求提供一个每月的销售预测,指示预期的销售量。动机偏好会导致销售员的估计比真实的低,这样当销量超过预测时,看起来是因为销售员的表现异常出色。在别的例子中,有证据表明天气预报中下雨的机会要稍为比实际可能性要大。这种偏好主要是无意识造成的。天气预报员不自觉地让人们为坏天气做好准备,在看到天晴时感到惊喜,而不愿意他们期望着好天气而失望。

这种偏好是最难通过设计一个决策支持系统而解决的。一个可能的办法是让决策支持系统询问大量的估计,这些估计来自相似的来源,与问题背景可以有关或无关。给定足够的信息,用户可能可以从相关团体的估计中检测出动机偏好的趋势。这种方法并不是很有效,但是它可以提高决策者对潜在的动机偏好的警觉。

3.9 效果和效率

在第 1 章,我们已经知道决策支持系统的一个主要目标是提高决策结果的效果。这也许是决策支持系统著作中最重要的一个主张。紧接着的第二个目标就是在给定的背景下提高制定决策的效率。以上两点,虽然在本质上是互为补充的,但却是两个不同的概念,而且在决策支持系统的设计和实现中往往是相互冲突的。只有区分开这两个概念,才能充分地理解它们对决策支持系统的作用的影响。

在制定决策的时候,效果(effectiveness)集中于应该做什么,而效率(efficiency)集中在我们该怎样去做。要想有效决策,需要仔细考虑影响决策的各种条件。如果我们是在设计一个帮助决策者提高决策效果的决策支持系统,我们必须了解他对决策背景的理解。相反地,提高效率意味着我们只需要注意能最大限度减少时间、成本和精力的问题。这两个目标之间会存在一种张力,因为在一方面投入的注意力增多会导致在另一方面的减少。Keen 和 Scott Morton 提供的一个例子很好地说明效果和效率之间的区别:

> 一个公司的计算中心……很自豪地吹嘘自己的效率:它能比国内任何其他计算中心产生更多的输出——管理报告;计算机的停机时间短;机器的使用率高,并且故障时间最小;输入数据能立刻得到处理,输出结果能迅速交付。不幸的是,用户并不认为这些报告非常有用。经理们指示他们的秘书,要么把那些没读过的上百页的上个月的经营总结放到档案柜里,要么就把它们扔到废纸篓。这个中心在追求一个无效果的目标上是有效率的[强调]。(Keen 和 Scott Morton,1978,7)

表 3-9 列出了使用决策支持系统对效果和效率的贡献。

表 3-9　使用决策支持系统对效果和效率的贡献

效果
• 更容易得到相关的信息
• 更快、更有效地认识和确认问题
• 更容易得到计算工具和验证了的模型，以便对选择标准进行计算
• 更强的产生和评估大型选择集合的能力
效率
• 决策成本的减少
• 减少分析同等规模的细节问题的决策时间
• 为决策者提供的更好的反馈

这两个目标之间最根本的差异在于，效果需要不断的适应、学习和再考虑，经常冒着进度缓慢和不能运行的风险，而效率只是简单地注重用更经济的方法做同一件事情。这种张力的一个很好的例子是一个组织的研发工作。研发可以看做为了在未来提供效果而进行的一笔无效率的资源支出。研发可以被除去，因为现存的经营任务并不需要它，而且排除它可以对当年的利润有显著的有利影响。如果只从效率的角度来衡量的话，这种观点无可辩驳；但如果人们的目标是效果的话，那么排除研发就是一个组织的灭亡。为了保证效果，效率往往必须做出牺牲。对于决策支持系统的设计者来说，如何在效果和效率这两个经常矛盾的目标之间取得一个合适的平衡点，是一个很重要的任务。经验说明，环境越不稳定，越要注重效果。如果环境是稳定的，那么一个公司就可以把它的注意力放在使第二年更有效率的经营上面。在未来的年月，经理们会发现他们越来越重视效果而不是效率。决策支持系统也会成为支持这一重点的基本工具。

3.10　小　　结

这一章讨论了制定决策所包含的过程。使一个决策程序化的难度和决策的结构化程度有关。简单（结构化）的决策包含了反复性的常规步骤。相反地，当认知局限和相关的不确定性包围了整个决策过程时，制定决策就很困难，或者说决策是非结构化的。如果随机因素将决策引向一个不确定的结果，未决决策的准确性就降低了。

决策可以按照类型划分：基于行为、基于协商、或者基于策略的决策。基于行为的决策关注于和决策相关的活动；基于协商的决策又可以按照对协商的看法分为三类；最后，基于策略的决策可以根据做出最终选择所使用的基本策略划分。

制定决策的过程是复杂的，而且即使是最有经验的决策者通常也不清楚。了解有效地制定决策的天然障碍既能够帮助改善决策过程，也能够改善相关的结果。

复习题

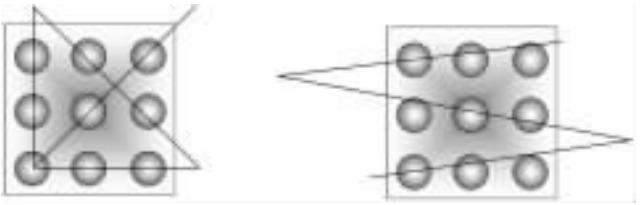
1. 从问题的结构化的角度陈述制定决策的困难。
2. 为什么理解决策的分类方法对于决策支持系统的设计非常重要？
3. 描述决策基于行为的分类方法，分别为每一类举出例子。
4. 简单描述 Simon 的问题解决模型的构成。
5. 能够做出最优化的决策吗？解释可以或者不可以的原因。
6. 什么是满意（satisficing）？
7. 为什么受限制的理性的概念对于决策过程非常重要？
8. 问题和症状的区别是什么？
9. 在决策过程中混淆问题和症状会产生什么影响？
10. 简化决策环境的危害是什么？
11. 描述决策者凭感知制定决策的影响。
12. 决策者的判断会给决策造成什么影响？
13. 使用启发式搜索法（启发式编程）的优点有哪些？
14. 定义启发式偏好。
15. 列举并描述四种最常见的启发式偏好。
16. 比较并对比效果和效率的概念。

讨论题

1. 假设你正准备购买一辆小轿车。使用 Simon 的解决问题模型，列举出你各个阶段的活动。
2. 假设你正准备制定一个关于新产品价格的决策。使用本章描述的分类方法为这个决策分类，陈述任何需要的假设。
3. 根据你自己的经验，举一个受限制的理性概念的例子。
4. 分析一个你最近遇到的问题，列出它的症状，并且将这些症状的起因分类。哪一个才是真正的问题？
5. 分析你在制定决策的过程中遇到的认知局限。决策支持系统能够提供帮助吗？如果可以，它又是如何提供帮助的？
6. 分析组织的一个错误决策，讨论失败的原因。这次失败是和认知局限有关，还是和决策者的启发性偏好有关？

九点难题的答案

左图显示了这个难题的普通解法。右图则展现了一种三线连接的替代解法。



Adams 还收集到了另一些解法，部分列举如下：

- 将纸张折叠，因此所有的点排于一列，然后可以简单地通过一条线连接所有的点。
- 将纸张卷起，边缘接合成纸筒，然后可以用一条螺旋线连接所有的点。
- 将纸张裁成小条，再将它们卷在一起，因此也可以只用一条线连接所有的点。
- 在一个小区域内画九个非常小的点，然后用一条很粗的线一次连接所有的点。这个解法是由一个十岁的小女孩提出的。

资料来源：J Adams，Conceptual Blockbusting，（第 24 和 25 页）。

第 4 章

组织中的决策

学习目标

- 学习组织和组织中的决策制定活动的定义
- 熟悉组织决策的五个维度：小组结构、小组角色、小组过程、小组风格和小组规范
- 理解不同组织决策层次上的不同类型的支持系统的需求
- 理解组织文化及其对组织中决策制定活动的影响的意义
- 明白组织文化和组织变化的能力之间的关系
- 学习组织中权力和政策对决策制定活动的影响，理解它们对决策支持系统的设计与实施的作用；熟悉组织决策支持系统的功能

小案例

古巴导弹危机

1962 年 10 月 16 日，美国空中照相技术侦察到苏联开始在距离美国边界 90 公里的古巴岛上建造导弹基地，并且在那里安置中程（1000 公里）和中远程（2000 公里）导弹。仍处在对自己内阁在古巴的猪湾外交政策的惨败的懊恼之中的约翰·肯尼迪总统，害怕苏联对西半球未加制止的干涉将破坏美国的全球性的可信性。他决定做出强烈的反应。在和他的最信任的顾问进行了一周的激烈的秘密讨论之后，肯尼迪对克里姆林宫发出了最后通牒。他要求，导弹必须立刻撤除，否则美国将施行海上封锁以防止更多的导弹进入。而且，美国将武力拆除古巴现有的导弹装备。这个通告使得全美和全世界都为之震惊，因为从前从没出现过两个超级大国在核武器战争上的直接交锋。美国国务卿 DeanRusk 说："我们现在是相持不下。"美国军事部门提交了侵占古巴的计划，然而，做出的决定是和平地解决危机问题。

根据苏联总理赫鲁晓夫的回忆，在 1962 年的 5 月，他想到要在古巴放置中远程核导弹，作为对美国的导弹战略进行反击的一种方法。他也提出：可以以这个计划来保护古巴以避免美国入侵（例如美军 1961 年对猪湾的失败的侵略尝试）。

在得到卡斯特罗的同意之后，苏联快速、并且秘密地在古巴建立导弹基地。10 月 16 日，肯尼迪观看了苏联在古巴施工的导弹基地的空中拍摄图片。美国内阁经过了 7 天激烈的讨论，在这期间，苏联外交官否认在古巴建造导弹基地。肯尼迪在 10 月 22 日通过电视发表了演说，在演说中宣告已经发现古巴的导弹基地，并且申明，任何来自古巴的核导弹攻击都会被认为来自苏联，美国将就此事做出强烈的反应。他也对古巴进行了海上封锁，以防止苏联运送更多的导弹进来。

危机期间，双方进行了很多的书信往来和联系，有的是正式的，也有暗中进行的。赫鲁晓夫在 10 月 23 日和 24 日给肯尼迪写了信，信中称在古巴建立导弹基地只是起威慑的作用，并且申明了苏联的和平愿望。在 10 月 26 日，赫鲁晓夫给肯尼迪写了一封散漫的长信，好像是说如果美国保证美国或者是美国的代理国不侵占古巴，苏联将撤离在古巴建造的导弹基地以及人员。在 10 月 27 日，另外一封来自赫鲁晓夫的信中称，如果美国撤销在土耳其的导弹基地，苏联就撤销在古巴的导弹基地。美国内阁决定不采纳 10 月 27 日的信中所写的内容，而采纳 10 月 26 日的。然后赫鲁晓夫在 10 月 28 日宣布撤销古巴的导弹基地，把导弹运回苏联，并且还表达了对美国不侵略古巴的信任。两国之间在 10 月 28 日做了进一步的协商，包括美国要求苏联的氢弹从古巴撤离以及美国保证不侵占古巴的具体条款等等。

以下是赫鲁晓夫在 10 月 24 日给肯尼迪写的信：

尊敬的总统阁下：

总统阁下，想像一下，如果我们也给你们发了一个同样的通牒，那会怎么样？你们会对此采取什么行动？我想您会对我所做的事情感到出奇的愤怒。这个，我们很理解。

在给我们提了这些条件之后，总统阁下，您也就是对我们发起了挑战。是谁让您这样做的？您有什么权力这样做？我们和古巴共和国的关系，同我们和别的国家的关系一样，不管这些国家是什么样的政治体系，只跟这两个国家之间的关系有关。而且，如果事情真是您在信中提到的封锁问题，依照国际间的惯例，只能通过两国之间的协商来确立，而不是通过某个第三国。封锁是存在于像农产品之类产品之上的。然而，现在我们并不是谈论封锁问题，而是一些更严重的问题，您自己清楚。

您，总统阁下，并不是宣布封锁，而是发出了一个通牒，并且您恐吓，如果我们不遵守您的命令，您将会实施武力。想想您自己说的话！您想说服我们同意！同意这些要求意味着什么？这将意味着处理我们和其他国家的关系不是通过讲道理，而是通过暴权。您不是对我们讲道理，而是在胁迫我们。

不，总统阁下，我不会同意的。我想在您心里，您也会认为我是对的。我深信，如果您是我，您也会这么做的。

美国没有什么权利或者基础来批准您信中所说的那些决定。因此，我们也不会接受这些决定。现在还存在国际法，存在广为接受的处事原则。我们坚决拥护国际法的条例，遵守在公海以及国际海域的航行管理。我们遵守这些规则，并且享有被所有国家承认的应有的权利。

您想强迫我们放弃每个主权国家应享有的权利 您在试着怀疑国际法，您在违反广为接受的国际法的条款。所有这些不仅仅是因为是你们对古巴人民和他们政府的仇恨，还跟美国的总统大选有很大的关系。什么道德、什么法律能够证明美国政府这种对国际事务的做法是正确的？这种道德、这种法律是找不着的，因为美国对古巴采取的行动完全是出于自私。这就是后帝国主义的疯狂。不幸的是，所有国家的人民，包括美国人民自己都将会受到这种疯狂的伤害，因为美国拥有了从前没有的现代武器。

因此，总统阁下，如果您冷静地思考一下，您将明白苏联负担不起不拒绝美国专制的要求。当您把这些条件摆在我们面前的时候，试着站在我们的立场考虑一下，换成美国会对此做出什么反应。我敢肯定，如果任何人对您和美国，提出相似的命令，您会拒绝。我们也一样会说 不。

苏联政府认为对领空和领海自由权的侵犯是一种将使得人类进入核武器战争深渊的侵略性行为。因此，苏联政府不能指示驶往古巴的苏联船舰的船长们听从美国海军关于封锁古巴岛的命令。我们对他们的指示是坚决遵守广为接受的公海的航海标准，并且绝不退让。如果美方违反了这个标准，美方要为违反付出代价。你们要清楚，我们不会眼睁睁地看着美国海军在公海领域的海盗行为。我们会采取必要的措施来捍卫我们的权利。这一切都是必要的。

赫鲁晓夫

莫斯科

1962.10.24

4.1　理 解 组 织

到现在为止,焦点主要是放在两个比较微型的层次概念上:决策以及个人决策制定者。如果没有对这些基础知识进行完全透彻的认识,就得不到宏观的看法。在这一章,重点就会从决策制定的微观看法转移到宏观看法,即组织。从上往下观察整个流程,使得我们可以从不同的层面看待决策制定和问题解决的一些行为和特性。

教科书上对"组织"这个术语有很多不同的定义。大多数的定义都提到了:"组织是为了共同的目的或目标而产生的社会实体之间相互交互的系统模式。"在这本书中对组织的定义不仅包括了这些共同的组织的特性,还把概念扩展到结构、资源、政策以及环境。依照我们的目的,一个组织可以被看成在一组松散定义环境中,由资源组成的统一系统,而这些资源是被一系列具有特定目标和功能的功能型和政策型子系统定义和组成。这个定义暗含的一点就是假设整体大于局部之和;它使得组织成员能够集体地追寻和获得比单独行动更为伟大的目标。通过这个定义,我们可以看出决策制定和决策支持的不同方面对于这样一个暗含合力协作的运作是必要的。

"决策网"

Stohr 和 Konsynski (1992) 提出:把组织看成一个"决策网"是很有用的。组织的目的是在一个商业的环境里制定决策,而且为决策机会、权威人士和职责所限定。在这一点上,决策以及决策制定是组织的本质,是组织边界、组织政策、组织过程、组织风俗以及运作环境的标识符。然而,要注意到这个观点和个人决策制定者是组织的物理表现方式的观点是无关的。在这里,组织是由个人、团体、团队以及电脑化的决策支持系统组成的复杂网络构成。怎样使它们相互之间好好配合是这章的主题。

组织决策的维度

如果我们把上面对组织的定义分解开来,就会发现里面存在若干个因素。图 4-1 描述了能够影响组织中制定的决策种类以及有效的决策方法的五个维度。

小组结构

我们在第 2 章讨论了决策结构的概念。在这个方面,我们着重讲了决策点 (指的是制定最终决策的人) 和决策过程中的参与者之间的关系。如果决策制定者单独行动,而且除了通过独立的或者第三方的信息源,不和任何正式的参与者一起工作、交流,那么这个结构可以被看成是个体型的。如果决策制定者保留了惟一的决策制定权,同时他还雇了一群人,这些人从事特定的角色,并且他们很和谐地工作以向决策制定者提供一些有用的信息和服务,那么这样一个结构就差不多是团队类型的。最后,如果这一群人都有责任和权利来制定决策,而且每个人的决策制定的权限都是一样的,那么这样一个结构就可以被看成一个真正的团体型结构。给定这三种决策结构,就很容易识别决策类型是怎样制定的,同时也可以看出使用的决策方法很容易发生改变。第 8 章将着重谈到这些结构,当

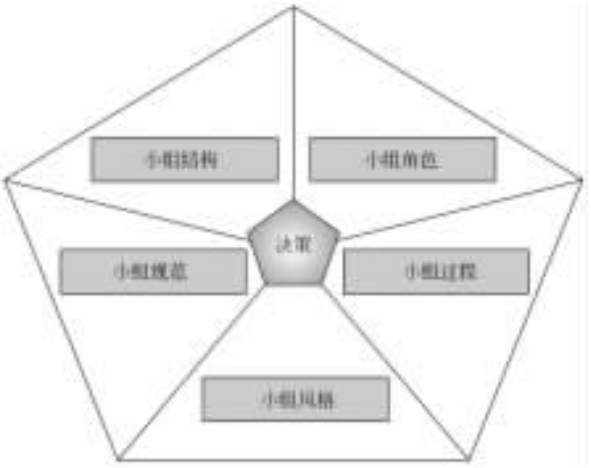


图 4-1 影响组织决策的因素

这些结构和特定的 DSS 设计需求联系起来的时候。

小组角色

每个决策制定团体的成员都有一个或多个角色。在个体型的结构里,所有必要的角色,例如信息收集者、分析者以及 DSS 使用者等等,都归到那个惟一的决策制定者身上。当系统向团队或者小组结构发展的时候,这些角色扩散到那些不同的成员中去,这样每个参与者就会变成决策过程中更为实在的成分。采用的决策战略以及指定决策类型的范围都可以本质地被改变,但这些改变是依赖于组织怎样定义这些角色(以及谁扮演这些角色)。

小组过程

被决策制定团体采用的过程对团体制定的决策类型有很大的影响。它会断然影响到制定这些决策所采用的方法以及过程。过程影响过程,这就像是一个"是先有鸡,还是先有蛋"的问题。这个循环的问题其实告诉我们:决策过程与决策类型必须与决策制定团体的观点相协调。如果采用一个有多个决策参与者的结构,那么特定的决策过程对团队来说不再是有效或者是相关的。例如,在美国,最后决定是否要参战只取决于一个人,即总统的意思。虽然总统必须在形式上要求国会宣战,但是决定权还是在一个人身上。原因就是宪法的制定者意识到,集体不能很好地制定这样一种决策类型。多渠道的吸取信息和建议是必要的,但是要所有人同意就不必要了。如果做这样一个决策的时刻到来,大家最好还是使用只有一个人做决策的决策过程。

小组风格

在第 2 章,我们也介绍了关于决策风格的概念。在任何一个决策制定的情况下,不同的人际关系以及压力、决策结果、权力、政策等,都会决定一个特定的决策制定团体最好采

用的决策方法和决策类型。正如前面所述，决策者的风格会影响到决策过程，他在特定条件下的行为以及成果的质量。例如，宣战的决策环境对于那些对宣战有着不同观点的，且持 A 观点和 B 观点的人数基本一致的团体是不适合的。这样一个决策最好还是留给那些对那些明显概念有着相似观点，并且可以基于现有的信息，而不是个人信仰或者是政治立场做出必要决策的团体。组织必须认识到这些因素，并且要为给定的决策条件创造出最好的决策团体。

小组规范

这个维度是最重要的维度。关于决策的社会心理学对决策制定小组有很大的影响，下至所有个体，上至整个小组都受其影响。成员之间的价值共享、个人和集体的社会压力、行为类型和行为指示、个人信念、潜在的认可都会塑造一个组织的决策制定环境。

以关联的角度来看待这五个因素，我们会发现组织中的决策制定其实是一个连续的过程。也就是说，这些因素是基于与它们无关的动力学进行变化的，它们在时时刻刻调整以寻求平衡，但是不管领导者有多大能耐，它们还是不能达到平衡。这里有很多关于最优化是无用的证据。使这个问题更为复杂化的是，我们必须意识到平衡行为不是发生在二维，而是发生在三维：(1) 参与者人数，(2) 参与者结构以及角色，(3) 参与者在组织层次中的位置。

组织决策层次

如果从决策制定的观点来看待一个组织，那么可以认为组织有三个层次：(1) 操作层，(2) 战术层，(3) 战略层 (参见图 4-2)。所有的组织决策都可以被安置在这三层中。

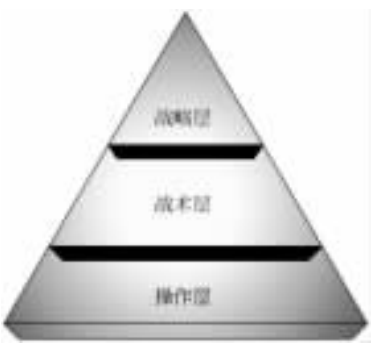


图 4-2　组织决策的层次

在组织中，决策制定者是由很多，而且各不相同的个人组成的。在基层，即操作层，工作人员主要是做出一些和日常的生产以及服务相关的决策，而操作层的管理员主要是配置可用的资源以及做一些应付指派限额和进度表的必要决策。战术 (或者是计划) 层主要是执行组织最高层所做的决策以及为保持操作层要求的能力和产出，做一些获取资源的决策。大部分和组织内部管理有关的决策也是在这一层制定。

战略层的成员包括世界上大工业公司或者是服务公司的高层经理，以及大型政府机构的高层管理人员，军队里的将军和高级军官。这些决策制定者中的中坚分子，虽然人数在这三层中是最少的，但是他们是社会的主要决策的制定者。他们制定的决策往往是在这三个层次中最具有深远意义。

当设计和使用 DSS 的时候，要考虑决策是在组织中的哪一个层次制定的。不同的系统和组织的不同层次相关。某个层次的 DSS 的设计者面临的一个问题，就是每个层次人员能力和技能的流动性和多样性。当工作人员从低层升到高层，这些人必须改变自己决

策制定的观点以及战略,扮演新的角色,担当新的任务。然而,有时候,他们旧的习惯仍然存在,而且通常这些必要的改变(指的是决策制定观点的改变)并不发生。在这里,DSS 的设计者就应该解决一种现象,这种现象是 Peter 原则,它表明一个人会一直升迁到不能胜任的职位为止。当这种情况发生时,DSS 设计者所期待的使用者应有的技能和能力可能就会不存在,这样应用的决策辅助工具可能就会有问题。我们将在 13 章和 14 章深入讨论这种问题。

4.2 组织文化

什么是组织文化

组织文化(organizational culture)这个术语在关于组织行为、组织理论、组织结构、组织管理的文献中简直是无处不在。好像也不缺乏对这个术语的定义。对组织文化最有说服力的一些定义是:组织支持的主导价值观(Deal 和 Kennedy,1982 年);引导组织政策面向员工和客户的哲学体系(Pascale 和 Athos,1981 年);组织成员共享的基本假设和信念(Browers 和 Seashore,1966 年)以及我们的这本书中所写的。不管这些定义来自哪里,它们都好像有这样一个共同特性:"价值共享"。我们将采纳著名的组织理论家 Edgar Schein (1985) 的定义。他把组织文化定义成由经过多年演化的信念、符号、仪式、神话、惯例构成的共享价值的系统。通过他的定义,我们可以探究组织的这个方面以及它和决策制定的关系。

组织文化不是一个抽象的、没有明确要素的环境。研究表明:组织文化对于组织活动(包括决策制定)很有影响。Robbins (1991) 定义了十个要素,用这十个要素可以定义一个特定的组织文化。表 4-1 列出了组织文化的这十个特性。

表 4-1 组织文化的十个特性

• 个人主动性	个人所拥有的责任感、自由、独立的程度
• 风险忍耐力	鼓励员工积极进取、革新、冒风险的程度
• 方向	组织创立清晰目标和业绩期望的程度
• 整合	鼓励组织中各个单元相互协调运作的程度
• 管理支持	管理者给下属提供畅通的交流、帮助和支持的程度
• 控制	规则、管理条例,以及用来监督和控制员工行为的直接监督条例的数量
• 认同感	成员把组织看成一个整体而不是自己的一个工作团队或者是专业领域的程度
• 报酬系统	报酬分配基于员工工作成效,而不是基于资历、偏好等的程度
• 争论忍耐力	鼓励员工公开批评、争论的程度
• 交流模式	组织交流受形式上的权力层次限制的程度

注意这十个特性并不仅仅限于结构上的问题,还包括一些行为维度。有一点很重要,组织文化不仅仅是历史结构进化的产物,还是组织成员行为过程进化的产物。

文化与业绩以及变化之间的关系

业绩

在第 2 章,我们说过个人的决策类型会影响到决策战略及其成果。决策类型还会影响到个人在组织环境中的行为习惯。在一些环境中,个人会支持他人,但是在另外一些环境里会争夺控制权。一些组织对于创新和改变方面是很严格而且很顽固的,而其他的一些组织鼓励大家在想问题和解决问题的时候要创新。因此,DSS 的设计者必须意识到组织文化对决策制定的影响,以及在特定的文化环境中应采用的战略。

观察组织文化和决策制定之间关系的方法就是关注文化匹配 (cultural fit)。成功的组织显示出它们的文化和企业内外部环境之间具有很好的匹配。一个好的外部匹配 (external fit) 表明企业文化既适合组织的战略,同时也适合市场。例如,市场导向的战略在那些易变的环境中更为成功。这样的文化气氛重视组织对争论的忍耐力、个人积极性、冒险精神、高度的部门整合,以及水平的沟通渠道。在这种文化中制定的决策以及相关的战略比较重视效果、创新以及创造力,但是也是比较松散的。同时,它的信息来源也要求有一定的广度。同样的,对这些决策以及战略进行支持的 DSS 的功能应该能被系统支持的技术使用和强化。相反,如果一个组织是产品导向,就会重视效率以及在稳定的环境中有好的表现,并且比较倾向于推行高控制、低冲突、低风险的文化。在这样的环境中所做的决策会向连续的结构化目标发展。而在这样一个组织里所利用的决策支持技术,就必须被组织的基本模型库和分析工具高度结构化。重要的一点就是,很难说这两种导向中,哪一种导向会比另外一种导向好。只要组织文化和外界环境相匹配,那么组织就有更多成功的机会。

内部匹配 (internal fit) 是组织成功的另一个重要元素。它指的是组织文化和组织主要技术的匹配。那些想把个人积极性、承受的风险最小化的组织文化 (例如产品导向的文化),倾向于采用更多的能使组织更加稳定的常规技术。这些技术有完善的控制机制,自动化的装配机器,以及常见的支持办公的处理机制。另外,因为这些技术普遍都很完善,对个人干预几乎没有什么要求,所以它们也趋向于支持一个更为集中的决策重点 (即由权力中心制定操作步骤、政策、进度表等等)。在那些比较看重个人积极性、创造性以及风险的组织文化中,有适应能力的非常规技术更为流行。就是在这样的文化中,DSS 技术才可能成为问题解决领域中的一部分。把 DSS 技术用在错误的组织文化中,就像是把一个方的木桩放入一个圆孔,虽然可能适合,但是它不会运作的很好。

改变

组织文化另外一个重要的方面就是,它常常决定一个公司在环境条件下对改变的变动能力和适应能力。很明显,生存就等同于组织的变化能力。如果企业的文化不考虑适应性和变革,那么企业将不能继续保持自己的市场份额,也就不能生存。那些想维持现状,倾向于稳定型的文化,将发现自己很难对环境的变化做出应变。在 Peters 和

Waterman 的 In Search of Excellence 一书中，指出了成功的公司的八个特性。表 4-2 列举包含了这些特点。

表 4-2 成功公司的特性

• 它们偏好行动	• 它们是价值驱动的
• 它们接近顾客，并且了解顾客	• 它们只管自己的事
• 它们提供自主性，鼓励创业	• 它们的工作人员是精简的
• 它们认识到生产率应该以人为基础	• 它们采取既松又紧的控制机制

注意 Peters 以及 Waterman 指出的这些特点都是行为导向的。这样，所采用的决策和战略是多种多样的，并且支持很多种支持技术和方法。相比之下，那些官僚型的组织更倾向于使用“会议室的决策战略”，也就是实施和管理集权控制系统，而减少组织中对个人 DSS 和团体 DSS 的需求。

4.3 权力和政策

在大多数组织中，都存在这样一个事实：权力和政策是每天所做的大量决策中不可分割的一部分。虽然，“权力”这个术语给人一种贬义的感觉，Zaleznick (1970) 认为它很重要，是有效决策的必需元素：

> 不管组织是什么类型，解决问题的手段、社会技术系统及报酬系统等等都是政策结构的。这就意味着组织通过分散权力以及为权力的执行创造一个舞台来运作。因此，毫无疑问，想寻求和利用权力的个人总能在商业中找到一个似曾相识而且友好的环境。

从目前讨论的组织的许多其他概念中看出，组织权力的定义很多。因为我们的重点在于决策制定和决策支持，所以我们认为权力是由五个因素组成的。表 4-3 对这五个因素进行了概括性的描述。

表 4-3 权力的五个因素

• 权力分享	决定经理怎样行使自己的职权以及他是否愿意行使自己的权力
• 职权	经理对他管理领域和直接监督下发生的事所拥有的控制权的合法性
• 非正式权力	使得相互依赖的任务和合作活动得以完成的关系集合
• 影响	直接或间接干预结果，在这种干预中，经理们是根据自己的偏好制定决策
• 政策	与协商、舆论形成、影响断言以及为了某个特定目标而采取的曲折的实施方法相关

资料来源：摘自 Rowe 和 Boulgarides (1994)。

当权力和决策制定相关的时候，我们可以把权力看成组织中的一个人和一个团体的多方面能力，来防止其他人来取得自己想要的目标。另外，它还包括一个人或者一个团体对其他人在特定方向上的行动和改变的影响。职权、非正式权力、影响以及政策都会给权

力对组织的决策的影响施加力量。

在一个组织中的管理决策权力能够在很多场景中体现出来。这种权力的范围很广，并且它的支持物常常必须以这些权力的控制和约束的形式出现。Harrison (1995) 指出了管理的决策权力的十二个重要元素。

1. 决定工作种类和数量的权力。
2. 决定 (有法律限制) 组织的运作时间、地点以及执行方式的权力。
3. 决定从其他的公司或者转包商购买什么服务、产品和原材料的权力，以及决定 (有法律限制) 从哪里、什么时候、怎样来购买它们的权力。
4. 决定提供什么样的服务和生产什么样的产品的权力。
5. 制定和管理价格的权力。
6. 决定新投资的投资数额，以及决定怎样投资、在哪里投资、什么时候投资的权力。
7. 通过大众媒介来塑造和调节国民的价值体系的权力。
8. 影响现行法律的立法机构的权力，以及通过支持政府官员的候选人以及游说议员来制定以及通过新法律的权力。
9. 通过使用科技以及制造有毒的传播物质影响这个社会和全体民众健康和福利的权力。
10. 分红利和不分红利的权力。
11. 决定给教育部门以及慈善机构提供多少捐赠的能力，以及怎样捐赠、在什么地方、在什么时间捐赠的权力。
12. 保留权力的权力。

从上面的列出的各条可以看出，决策和权力是联系在一起的。每一条对决策支持机制的需求都与其他的完全不同。而且，在每一条中，我们都可以找到很多的子范畴。例如，在决定投资额的权力中包含着多种问题解决的背景以及对决策支持的需求。不同的投资需要用不同的模型和不同的启发式算法来进行分析。每个决策的子范畴都需要一个为满足决策制定者的需求而定制的 DSS。问题就在于，虽然设计和完善一个 DSS 主要是依赖于 DSS 使用的环境，但是如果想做一个有效的决策，就必须考虑组织及其权力中心的总体情况。

政策

虽然权力和政策之间的关系是共生的，但是它们还是很不同的，用不同的方式渗入决策制定的过程。更为重要的是，政策是形成决策战略中很重要的成分。

Patz 和 Rowe (1977) 把政策描述成一个过程，一个“在减少组织不确定性成分的基础上，增加组织的确定性”的过程。在这个意义上，政策是通过协商、舆论形成以及影响，与决策联系在一起的。这些影响力对决策战略和成果都有本质上的影响，很显然，在任何一个问题解决的环境中，都对政策能力具有一定的需求。在决策的政策方面的技巧和能力对组织有难以衡量的利益，而如果没有这方面的技巧，这些利益根本不可能实现。

Mintzberg (1985) 注意到在决策制定中的组织政策对组织并不会自动的功能失调，相

反，组织政策还是很有用的：

1. 政策可以更正一些不足之处，使之变成一个更有影响力的合法的系统，并且提供一些灵活的准则。
2. 从达尔文主义的角度来说，政策可以让组织中最有能力的人充当组织中决策制定最重要的角色。
3. 当不存在其他合理的方法时，政策常常能够确保客观地听从多方面的意见。
4. 政策可以促进那些受到传统方法阻碍的必要的组织变化。
5. 政策可以加快决策制定过程，特别是在实现服务于特定利益的抉择时。

在一个组织中设计、实现以及使用 DSS 时，要仔细考虑总的政策环境以及政策对特定决策环境的影响。给定决策的政策要求决策制定者要完全明白和决策结果有关的风险承担者之间的不同关系。这样的政策因素还包括风险承担者团体支持还是拒绝决策战略的可能性，也可能包括特定约束的信息，这些约束是被政治单位加到另外一些可能阻碍决策集的执行的政治单位上。例如，为了使得成本减少，将两个以前是自治的组织单位合并成一个的决策可能会带来政策上的事情，譬如说会产生对于某项组织资源或组织过程权力的争吵，行为政策上的不一致或者对立，或者对新产生的部门的任务缺乏一种共同的愿景。不管决策是在哪个层次上被制定，是在个人、团队、团体还是组织，权力和政策都会塑造决策环境及所需的决策支持类型。

4.4 支持组织决策制定

我们将在第 6 章和第 12 章讨论组织决策支持系统（ODSS），这里首先作个简要的介绍。这一节描述这一层次决策支持的范围。

ODSS 可以看成主要对多人决策进行支持的技术。这个概念是基于团体决策支持系统（GDSS），即支持个人小团体制定一些有共同利益和重要性的决策（GDSS 将在第 9 章中进行介绍）进行扩展的。ODSS 和 GDSS 不同的一点就是管理控制。在 ODSS 的环境中，定义和使用支持性的技术是为了使基于决策的组织人力划分在一定程度上被技术定义和控制。虽然传统的决策结构倾向于一种传统的等级型的结构，并且常常类似于组织的表面上的职权配置，较为现代化的决策结构认识到很多决策都有跨越传统职权范围的趋势。当这种情况出现时（出现的频率还是很高的），就产生了对暂时的或实质的决策部门的需要。在这种环境下，决策的需求可能会把自己显示成由一系列决策者制定的一系列相关决策，而且并不是所有的决策只和某个人或某个团体相关。这些决策都是相关的，所以就要求决策者的相互配合。当这种情况出现时，就需要 ODSS 来支持这种决策。Culnan 和 Gutek（1989）提出了四种基本活动，这些基本活动为组织创造了通过 ODSS 收集，存储，以及利用信息的需求。表 4-4 包含了这四个基本活动，并且还包含了与支持这些活动相关的 ODSS 的功能类型。

表 4-4　ODSS 应用环境和功能

• 解决特定任务问题	包括新产品设计、R&D 以及改进生产过程
• 信息收集	信息收集不仅仅针对于特定的需求或者危机，而且还作为一种“储备”的形式。这样，收集和使用不一定是同时进行的
• 子单位间的沟通	包含为了正在进行的合作以及决策执行的信息使用
• 政策行为	利益竞争、资源不足以及含糊都会增加信息的象征性和战略性的使用

Swanson 和 Zmud (1989) 给我们提供了另外一种基于组织决策支持的战略分类法：

1. 决策订阅：决策可以被评级或者被订阅，这样那些高级别的决策就对组织有高的优先级。
2. 信息共享：决策被独立制定，但是所有的参与决策制定的人都共享相关信息。这样的信息共享可以被管理层指引，或者以一种自下而上的、特别的方式存在。
3. 商议的决策：因为分配方式的决策制定常常会引起一些冲突，一些决策就必须在冲突的各党派之间商议解决。一个商议的决策，实际上是由商议的团体共同制定的。

正如我们将在以后看到的一样，不是技术本身会在一个组织里提高决策制定的效果，而是支持技术、决策战略以及组织权力和政策相互配合的程度会提高决策制定效果。在这一点上，你就会感到团体和团队，以及组织和团队之间的概念区别并不是很清楚。这个不清楚的问题在对组织流程研究时就暴露出来了，同时也引起了大家的争论和重视。虽然存在这种不清楚的概念，但是 ODSS 真正的好处将在组织间很好地显示出来。作为决策支持技术的设计者，当公司与它们的顾客、供应商、政府机构甚至竞争者打交道的时候，将需要管理大量由相关但是不相同的决策者制定的决策的平台。而 ODSS 将充当这样的一个平台。我们将在第 6 章对此展开讨论。

4.5　小　　结

在第 3 章，我们讨论了个人决策类型的概念，以及它对决策战略和决策成果的影响。而且，我们看到在组织的环境中，决策类型能够影响个人的处事方式。

这一章中介绍了组织的定义和组织中决策制定活动的定义。组织中有很大一部分是决策者。经理们应该清楚权力和政策对决策活动的影响，同时也应该明白在组织的不同层次需要不同的决策支持系统。

复习题

1. 定义组织。
2. 列出并且简要介绍组织决策的五个维度。
3. 在组织中有哪三个决策层次？

4. 定义组织文化。
5. 组织文化对业绩有什么影响？
6. 在组织文化和变化之间有什么关系？
7. 指出在组织中定义权力时需要的五个基本因素。
8. 描述在组织中权力对决策的影响。
9. 什么是政策？为什么它对 DSS 的设计很重要？
10. 政策和决策制定之间有什么关系？
11. 描述一下多人决策涉及到的支持技术的种类。
12. ODSS 和 GDSS 最主要的区别是什么？
13. 列出能被 ODSS 支持的活动。

讨论题

1. 学习和观察一个组织，试着描述组织文化。通过例子自由地解释你的观察。在这个环境中什么样的文化对组织的业绩以及它适应变化的能力有最大的影响？
2. 描述组织的各个决策层次上的主要特点，分别给出一个例子。
3. 组织环境怎样影响决策支持系统的设计和实施？
4. 在组织中找到一篇描述 DSS 的成功实施和不成功实施的论文。指出组织文化、组织权力和政策的影响。
5. 分析市场上的 DSS 应用软件，描述它的功能，并说明它是怎样支持组织决策的。

第 5 章

决策过程建模

学习目标

- 学会使用三种主要部件来描述问题 :事件当前的状态、期望状态和目标
- 学会识别问题的范围
- 理解问题构建的三个基本要素 :选择、不确定性和目标
- 熟悉两种决策建模工具 :影响图和决策树
- 在理解问题结构组成和工具的基础上,学习通用的决策结构及其变形
- 熟悉各种不同的决策模型 (抽象的或者概念的) ,它们是决策者分析以及随后的预测和预言的基础
- 理解概率的三个必要条件
- 理解概率的不同理解 :长期频率、主观概率和逻辑概率
- 学会使用直接概率预测、几率预测或者比较预测来估计和预测某一不确定性事件的概率
- 识别估计可测量的倾向 (标度) 、分析敏感性 (敏感性分析) 和评价信息成本 (价值分析) 的技术

小案例

Exxon Valdez 油轮——这个事件有什么蹊跷之处？

1989 年 3 月 24 日午夜过后 4 分钟，装有 1 264 155 桶原油的 Exxon Valdez 油轮在阿拉斯加的 Prince William 海峡东北的 Bligh Reef 暗礁处搁浅了。大约 1/5 的货物共 1120 万加仑原油倾泻进海里。海面在经历了三天的平静后，刮起了强烈的东北风，风吹散了海面的原油，也吹掉了把它们再回收的希望。大部分原油由于风力作用与海水混合成一种泡沫奶油似的乳状液，既不能燃烧也很难从海面或海滩上清除掉。这些倾倒的原油形成薄薄的一层，并伴有厚厚的泡沫，继续向西南扩展，在从 Price William 海峡到南方的 Kodiak 群岛和阿拉斯加半岛大约 750 公里（470 英里）的沿岸线上蔓延着。科学家估计倾倒的原油中有 35％自己蒸发消失，40％沉积在 Price William 海峡内的海滩上，另外 25％进入阿拉斯加海湾后或沉积在海滩，或在海里消失。1989 年夏季进行的实地考察确认，Prince William 海峡 790 英里的海岸线被油污染，其中超过 200 英里的地方被列为重度污染。在 Kenai Peninsula-Kodiak 地区，人们发现超过 2400 英里的海岸线已被污染。

事后的努力包括卸载未倾倒的原油、抢救货物、海岸地带调查评估、高空飞行追踪浮油、撇去浮油、清理污染的海滩、拯救野生动物以及废物管理、后勤支援、改善公众关系等。清理工作主要在 1989 年到 1992 年每个春、夏季进行。这些工作及其后勤在以地面为基地的强大基础组织的支持下，共有成千上万名工人参加，并动用了数百辆船只和飞机。1989 年的清理工作就动用了 11000 人力和 1400 辆船只。多年的清理花费超过了 20 亿美元。用于清除或净化原油的技术包括燃烧、化学分解、高温高压水冲洗、冷水洗涤、人工和机械清理以及沉积法。一些值得关注的统计数据如下：

在 1989 年通过撇取回收的石油仅占初始倾倒总量的 8.5％。

在最初的四个夏季，海滩的清除工作共回收处理了大约 31000 吨的固体污染物，估计里面含有占初始倾倒总量 5％到 8％的石油。

在 1989——1990 年冬天，海滩表面大约 90％的沉淀物（小于 25 厘米）被自然过程（风雨腐蚀和生化分解）清除，然而只有 40％的深层油污被清除。

到 1992 年，通过自然过程和人工清除活动的结合，尽管在海峡的许多海岸线上还有零星数量存在，表面的油污已经基本清除了。

1992 年 6 月 10 日，联邦的现场协调人 Admiral Ciancaglini 发表了一份正式声明，宣布结束海岸线的清除工作。这项因为 Exxon Valdez 石油泄漏展开的清理工作，持续了 3 年时间并花费了 21 亿多美元的资金。在后来的调查里，人们确认，负责阿拉斯加州管道铺设和维护的石油公司联盟 Aleyeska 采取的石油意外泄漏补救计划根本不足以应付数量如此巨大的一次泄漏。调查人员发现有证据表明，与各种石油泄漏假定有关的风险概率，包括像 Exxon Valdez 这样的一类，总体上被低估了。进一步确定，如果当初计划能更现实地考虑灾难的可能性，在灾难来临时将有更多可拨出并可用的资源。

5.1　问题的定义及其结构

回忆一下，DSS 的许多功能，以及现代基于计算机的 DSS 所能采取的各种形式的关键，是一个或多个可用于研究手头上的问题并由此获得相关信息的内在模型。在本章，我们将集中讨论问题的各种定义以及为 DSS 设计人员提供的决策模型。

问题定义

不知身处何地，就不可能到达目的地。这句格言实际上成为了解决问题和决策制定领域里的定律。如果不先花时间全面地识别和定义问题，那么尽管我们非常希望解决问题并得到一个决策，仍不会取得成功。这听起来似乎过于简单琐碎，但典型的决策者常常会在开始解决问题的步骤之前疏忽了对手头上问题的正式和简明的识别，其程度出乎人的意料。是否记得第 3 章的“头疼”问题？缺乏对问题全面的识别和定义，会得到一个看起来很好但解决不了问题的方案。

在第 2 章，问题被定义为识别当前的事件状态和希望的事件状态的差别。如果我们能识别和描述当前事件状态和希望达到的事件状态，我们就能进一步得到一个完全成型的问题阐述。这个问题阐述应该包括三个关键部分：(1) 当前的事件状态，(2) 希望的事件状态，(3) 一个能区分以上两者的主要目标 (或多目标) 的阐述。人们更容易通过考虑一个典型问题的阐述并分析其内容来理解这三个部分的价值。

在构造问题的阐述时常犯的错误是过早地把目光集中在解决方案的选择上，而不是集中到问题本身。当这个情况发生时，本来存在的合理性被剔除，问题的空间被限制在那些看起来符合嵌入在问题阐述中的预想的方案上。决策者得到的不是一个完全成型的问题阐述，而是一个找问题去解决的方案的阐述。举个例子，我们用上面完全成型的阐述的三部分要点来分析以下语句：

是否应该给证据确凿的杀人犯判处死刑？

这句话实际上讲了一个寻求问题的方案。首先，这个问题的说明根本不是阐述 (statement)，而是询问 (question)。当说明以这种方式来构成时，其答案就是对所提供的解决方案的肯定或者否定。如果答案是“是”的话，实施死刑就解决了问题；如果答案是“否”，免去死刑也解决了问题。然而，两种情况下都没有真正的问题得以说明——这样就没有真正的问题得以解决。有一种方法可以确定真正的内在问题，就是寻找考虑选择两者之一的原因：

为什么应该给证据确凿的杀人犯判处死刑？

从这个观点出发探索第一句话，我们把注意力重新集中到选择的原因而不是选择本身。我们从这个询问的答案中进一步得到包含三个必要组成部分的阐述：

许多证据确凿的杀人犯被释放回到社会后继续行凶。必须作出一个关于如何防止此类事件的发生的决断。

按照当前状态、希望状态和目标来阐述问题，我们就能得到一个清晰的定义和解决问题的方法。

问题范围

当问题完全得到成型和阐述时，决策者必须检查问题的范围。在检查的基础上，如果问题的范围不超出可用资源、已知限制或时间约束，就可以决定问题是否确实可以解决。这些情况下，当考虑到面向决策集的可用资源的有效配置时，问题的范围必须缩减到某个重心上。通常说来，问题的范围由决策者采取的优先权顺序来决定。

为了约束问题的范围，必须确定它的幅度。这样做的一个办法是通过一系列的提问来确认问题的细节。多少人受到影响？当前它需要多大的成本？什么时候开始发生？之前尝试过什么办法？还有谁知道存在这个问题？还有谁正在解决这个问题？问题的真正规模有多大？类似这些问题有助于决策者确定问题的幅度，从而决定一个合理的范围并集中精力去解决。

问题结构

问题结构的设计就像设计其他结构化的实体一样，比如一辆汽车、一幢建筑、一幅风景甚至是一件艺术品。任何情况下，基本的步骤都是一样的，即必须同时详细地考虑和分析：(1) 最终的面貌，(2) 要素的细节，(3) 要素之间的关系。这种类型的设计过程常常要反复进行，有时变得相当复杂。当需要解决某个步骤的变化时，设计者需要多次修订在另外一个步骤的决策。这种变动一直持续下去，直到设计者对以上三个方面产生的平衡感到满意为止。

不管决策的背景情况如何，问题的结构都可以按照三个基本组成部分来描述：(1) 选择，(2) 不确定性，(3) 目标。用这三部分，我们将探索一些构造问题的基本工具，通常的结构以及对当前问题选择模型的情况。

选择

选择 (choice) 这个概念隐含着存在多种可供挑选的方案。如果只有一种方案，那么就不存在选择以及决策了。当考虑到需要至少两种可选方案时，一种方法是假设所有的决策都有一个简单的替换方案——“什么也不做”。虽然这种办法并不解决问题，但这是一种可行方案，并且经常可以考虑。我们假定所有的问题和决策都含有这个不起作用的方案，以后在构造模型时就不把它包含进去了，但是记住它是存在的并至少在分析任何问题的开始应该给予考虑。

建立问题的结构时最常用的一种方法是影响图 (influence diagram)。用这种方法，问题结构的三个部分由被合并和关联的特殊图形来表示，进而描述已被建模的问题。另一种常用方法是决策树 (decision tree)，它致力于选择集和预期结果上。我们将在本节后面详细介绍这些建模技术，现在就开始用它们来说明这三个组成部分。图 5-1 从技术上说明了对一个具有四个可选方案的决策树例子，用于每个结构化要素的图形。

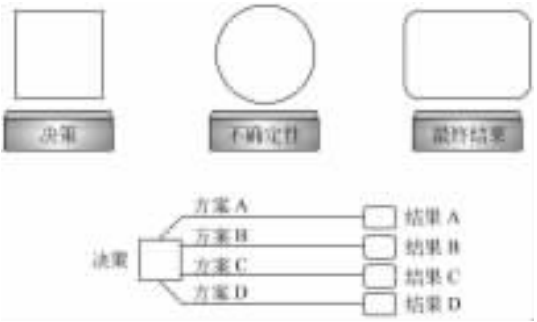


图 5-1　影响图和决策树的组成

不确定性

如果一个人能够确切地知道在问题前后的所有事件（意味着发生的可能性是 100%），那么制定复杂的决策就如同小孩游戏一样简单。同样，制定决策的过程就变得乏味且没有必要研究了。幸好对于讲授解决问题技术的教师和对这方面感兴趣的学生（比如你自己）来说，这不会是实际情况。决策者必须解决所有带有不确定性（uncertainty）的问题结构。记住，选择和不确定的主要区别是事件发生的概率，这是很重要的。对于选择，有 100% 的可能至少其中一个方案被选择，同时也有相同的可能性选择任何一个特定的方案。当一套可行方案被确定时，它们中的每一个在各种不确定性得到识别和解释后就已经被放进方案集中。这样，从决策树上选择一个可行方案的“分枝”就完全由决策者来决定。然而，关于不确定性的情况就不同了。按照定义，不确定性是决策者不能直接控制的情况，它们各自发生的概率最多在一定的范围内可以被估计到。因为如此，发生一个不确定事件会要求重建所有的未来事件。从这里才开始解决真正复杂的问题，才找到问题构造的真实情况。通过图解每个不确定性，把每个不确定性发生的期望概率包括在内，决策者可形象地查看不同的未来事件的路线，从而更容易比较不同的可能情景。我们将在 5. 2 节看到这些类型的模型。图 5-2 显示了一种不确定状态和三种可能结果以及它们之间的概率的模型。

图 5-2　包含结果的不确定性模型

目标

一旦最后一个不确定性得到解决，各种选择得到了分析和选取，决策就固定下来了。此时，不管结果如何，我们需要分析结果将如何根据其期望或成功系数来评价。目标

(objective) 为我们提供了建立标准来衡量某个特定结果的值或期望的方法。目标的例子包括诸如减少疾病、增加市场份额以及增加收入或减少成本等。注意一个好的设计目标必须考虑到量化的比较。类似“增加利润”的目标难以去评价。如果今年的利润比去年增加了 1 美元,那我们是否达到了预期目标?按照目标的阐述,我们达到了。然而,这很可能不是我们制定决策的目的。一个设计得更好的目标,比如“比去年利润增加 10%”或者“比前期利润提高 50000 美元”,能更好地反映我们的意图。决策可以用多个目标去衡量结果,但作为一个完全模型化的决策,至少需要确定一个可衡量的目标。

构造工具

影响图

影响图是一种图形化分析决策的简单方法。它包括了决策的三个组成部分 (如图 5-1),并用箭头连接起来建立它们之间的关系。图 5-3 包含了几个简单的影响图,以及用来构造它们的各种图形和线条的含义。

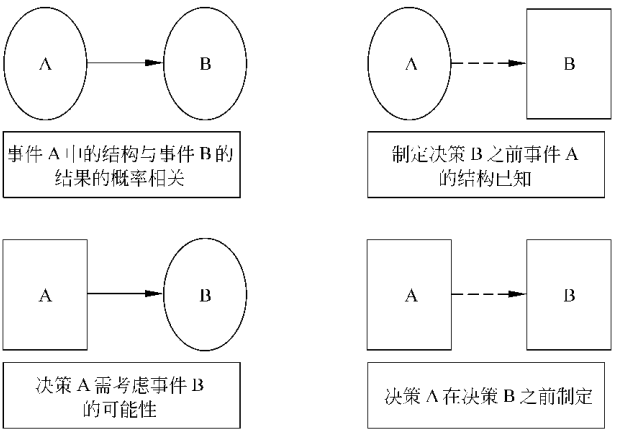


图 5-3 影响图中的箭头

如图所示,有两种类型的箭头用来连接结点 (node) (决策模型的组成要素)。实线箭头总是指向不确定性和目标,并描述了一种相关关系。换句话说,用实线箭头来表示前者与对后者的评价是相关的。在图的左下方,注意到用一个实线箭头来联系决策 A 和事件 B。这表明决策 A (前者) 关系到事件 B 的发生概率。举个例子,某学生在某个考试得到 A 等的成绩在一定程度上取决于他是否选择了学习。同样注意到事件 A 由实线箭头和事件 B 联系起来 (图的左上方)。这种情况下,事件 A 发生的可能性关系到事件 B 发生的可能性。比如,在暴风雨中被淋湿的概率取决于下雨的概率。

虚线箭头只用来指向决策,表示作出决策是基于来自前面结点的结果的知识。这样,事件 A 和决策 B 的关系暗示了决策者基于事件 A 的结果的全部知识做出决策 B (图的右上方)。从另一个角度来说,决策者利用决策前所有的知识做出一个决策。如果一个虚线箭头用来连接两个决策 (右下方),表明制定第一个决策在制定第二个决策之前。这样,我们通过不同结点的连接来描述在图中依次做出的决策。

影响图的另外一个特征是当构造正确时，它没有回路；不管从图的哪一点出发，没有道路能使人回到出发点。道路的箭头是单向的。一旦某个结点落在后面，将没有道路返回到它那里。这个构造规则用于遵循决策制定和问题解决的实际情况。一旦做出了决策，决策就定下来了，我们没法在后来去“不做”这个决策。

决策树

尽管影响图是分析某个决策前后结构的有效工具，但它没有考虑到当前决策的许多相关细节的描述。另一种能弥补其不足的构造工具是决策树。在这个图里，只分析了选择和不确定性。这两个组成部分由叫做“分枝（branch）”的线条来连接。从一个选择结点延伸出来的分枝表示决策者选择集中的各个项。从不确定性结点散发出来的分枝表示从某个事件出来的可能结果。每个结果的值附在每条结果分枝的末尾。图 5-4 包含了一个简单的决策树例子。

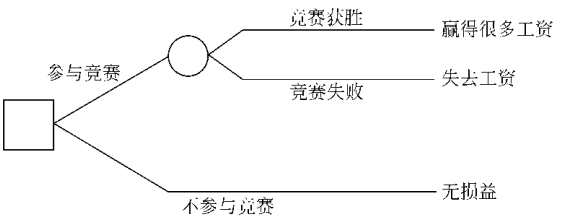


图 5-4　简单的决策树

类似于影响图，决策树可以比作一条铁路。通常设计成从左边开始，不断向右延伸直到结果出来为止。另外，设计决策树必须遵守几个规则以保证它能清晰地描述决策前后的细节。

首先，从选择结点出来的分枝使决策者只能有一种选择。不能有分枝设计成多种选择的结合。其次，从不确定性结点散发出来的分枝必须描述一个互斥（mutually exclusive）并且完备（collectively exhaustive）的结果集。第一个条件是互斥，意味着只有一个可能结果发生。这样的一个例子是一个不确定性结点带有表示成功和失败的两条分枝。这种情况下，遵循这个规则是因为一个项目或者成功或者失败，而不能同时是两者。第二个条件是完备，意味着这些分枝描述了所有的可能结果，没有其他可能存在。这两个条件一起，要求当解决了不确定性时，有且仅有一个结果出来。

第三个设计决策树的规则是：随着时间推移，决策者可能改变所有可能的选择和不确定性结果，所有的可能路线要能重新绘制。容易看到，当决策的环境变得更复杂时，决策树的规模将呈指数上升。例如，如果我们刚开始用一个带 3 条分枝的选择，每个不确定性具有 2 个结果，我们一共有 6 个结果。然而，如果我们在模型的每个不确定性分枝后加上一个带 3 条方案的选择结点，我们就把可能结果的总数从 6 增加到 18。这个指数增长通常被称作“树变得茂盛了”。

最后，随着时间推移，决策树必须能描述一个精确时间上的事件。这可以通过严格执行从左到右的构造方式来完成。通过这种方式，一个典型的决策树从选择结点出发，除非

考虑了所有的可能路线，否则必定要遇到选择结点或者需要解决某些不确定结点。

常见决策结构

关于所有的决策的环境是否惟一这个问题并没有争论，答案是惟一的。虽然如此，正如我们生活中面对的许多其他情况一样，决策前后包含有某些特定的共同特征和模式。在这部分，我们检查在管理活动过程中经常发生的决策结构。通过了解这些常用的结构，一个决策者能更快地确定一个新决策前后的大体结构。

基本风险决策

决策者面临的许多问题在面对一些不确定性时有必要作出一种选择。在基本风险决策（basic risky decision）结构里，一个成功的决策是从选择集中正确的选择以及从不确定性中选择正确结果的函数。从图 5-5 的例子看出这个决策结构是怎么应用于分析购买股票的决策的。下面是另外一个例子：

> 产品经理 Stephanie Essington 正在考虑如何促销她公司的新品软饮料 Blaster。她希望把目光集中在夏季常去海滩的消费者身上。促销活动的名称将是“Blaster 的快乐阳光夏季”，将通过小型音乐会、免费分发、沙滩排球和其他活动来促销这种新软饮料。沙滩活动的基本道理就是当人们在太阳下娱乐后，通常会享受清凉、提神的饮料来解渴。如果这个夏天将特别热的话，Blaster 和 Stephanie 都会取得成功。然而，如果在海滩地区经常下雨的话，Blaster 和 Essington 都会“湿透”的。

在上面的例子中，决策的前后情况由三部分组成：(1) 选择——是否要打促销战，(2) 不确定性——天气将是晴朗还是阴雨，(3) 目标——让 Blaster 品牌取得成功。正如所看到的，让 Blaster 成功这一目标是选择促销战以及海滩下雨量的函数。基本风险决策的结构可以用来描述在管理程序上遇到的许多常见情况。然而，常遇到的这种结构有两个变形：基本风险政策（basic risky policy）和基本风险多目标决策（basic risky decision with multiple objectives）。

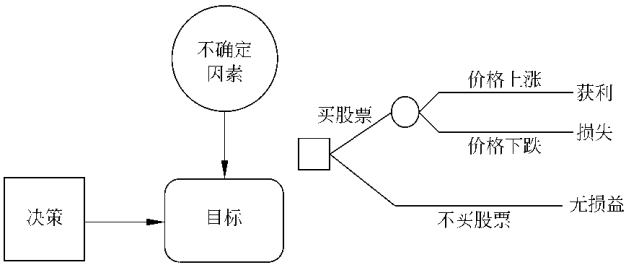


图 5-5 基本风险决策

如图 5-6 所示，基本风险政策描述了这样一种问题的环境，方案集中的选择对一个不确定性的概率有着直接影响，正如对目标有直接的影响一样。考虑以下例子：

Stephanie Essington 的夏季海滩 Blaster 销售战取得了辉煌的胜利，新产品正在销售和流通。Essington 仍然是产品经理，但她的奖金支票已足够让她买一艘新游艇了。现在她正在进行一场新的战役，目的是在产品首次亮相的海滩市场上继续扩大销售。Stephanie 相信 Blaster 的包装可以重新设计得更“适于户外”。她打算采用一种内置、可重新封口吸管的新包装，比一般软饮料包装密封性好得多。推出这种新包装可以极大地扩大市场份额。然而，新包装比传统的包装要贵许多，同时会在一定程度上减少利润。

这个问题仍然用三部分来分析，但它们之间的关系和基本风险决策的结构稍有不同。这里，选择（是否选择产品新包装）和不确定性（市场份额）都对目标（利润）有直接的影响。另外，选择也直接影响到与给定的市场份额增长概率相关的不确定性。如果 Stephanie 不对产品重新包装，那么作为重新包装带来的市场份额增长概率就会降到零。

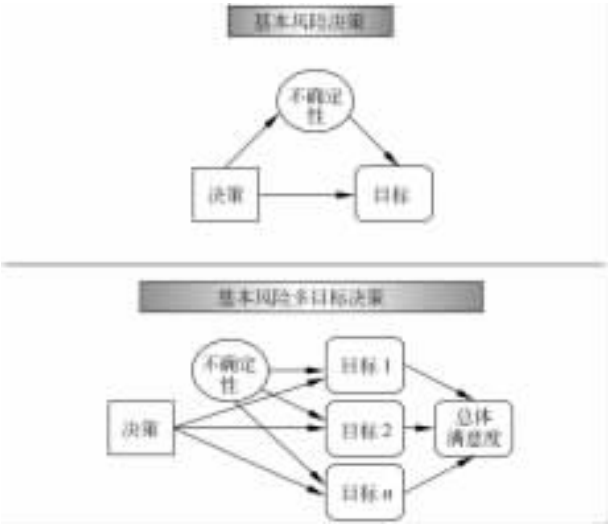


图 5-6　基本风险决策模型的变形

在第二个变形——基本风险多目标决策中，决策者为了管理当前决策前后的不确定性，面临从多个目标作出一个决断。最好的解决方案是能导致可量化的最大成功度。我们的软饮料例子示范了这个变形：

Blaster 已经是全美国海滩地区最畅销的软饮料，正在迅速成为全国的一个热销货。现在 Stephanie 被同事们亲切地称为“超级 Steph”。公司已经全权委托她把 Blaster 带进国际市场。因为如此，她必须扩大产品管理队伍。她考虑过关于队伍成员数量和地域的不同配置的几种方案。队伍新成员越多，Blaster 的市场覆盖率越大，但总的产品管理成本就越高。

这里，选择（选择哪种队伍配置）直接与多个目标相关（市场覆盖率和管理成本）。不

确定性(销售)也直接与多个目标相关。另外,这个选择同样影响到不确定性的概率分布[①]。如果一个目标得到最大化,另外的目标就有损失。决策者必须通过从选择集中选取适当的可行方案,来达到多目标的适当平衡。

确定性

另一个常见的决策结构包括那些在多个目标取得平衡的情形,在这里风险不是重要的考虑因素。在这个结构里,多目标、无风险决策、可衡量的成功度以及对最终选择的满意度取决于从多个目标中权衡产生一个适当的策略。这种结构的一个变形是多目标/多方法和无风险决策。这个结构表明其本身非常广泛和复杂,单纯一个方法不能完全解决问题。在处理公共政策问题的情形,比如吸毒、犯罪、失业以及无家可归时,这是很普遍的。在商业上,管理人员面临的情形中存在诸如员工士气、新产品成长以及提高效率和效果等问题。多数情况下,这些情形很复杂,设计单一的决策结构是不可行的。为了完成分析,决策者必须把问题的前后情况分解成多个管理子模型。在这点上,决策者必须确定在各个子模型的重点程度,然后从每个子模型的选择成分上选取可行方案。让我们回到 Stephanie 那里看看 Blaster 是怎么做的:

> Blaster 的新产品管理队伍已经成立,而且非常成功。高层管理者基于 Stephanie 的成功而给予再三提拔,Stephanie 同样给她的队伍成员予以晋升。然而,现在公司需要 Stephanie 和她的队伍再次"重做"这件事,也就是引入一种新产品。Stephanie 被授予设计和营销新产品的权力,因为 Blaster 队伍是如此受高层管理者的尊重。为开始这个进程,她必须根据队伍成员个人的优缺点来分配不同的任务,必须根据近来的消费调查数据来决定目标市场,必须安排测试和展示新产品的时间,必须做好研究开发工作的初期预算。这些决策之间有些相互影响,但基本上是独立的,而且总体相当复杂。

在这里,重点不是在每个决策情景中可能出现的风险因素。其焦点在于把各个独立的目标结合起来以期望取得可衡量的最大收益。

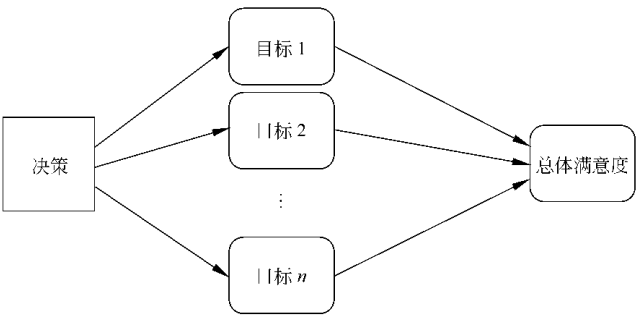


图 5-7 多目标、无风险决策

① 这种特殊的决策结构是一种变化的变种,表示具有多个目标的一种基本风险政策。

序列

决策过程不会总是显然地能构造出一个清晰的开头和结尾。通常，随着时间推移，条件发生变化，先前的一个决策可能不再合适或不再认为有用了。一个在过去某个时期里不可行的方案在目前新的条件下显得更为可取。序列（sequential）决策结构是一系列连续时期的基本风险决策集合，每个时间集之间用箭头来表明它们之间的关系。图 5-8 说明了这种类型的决策结构。

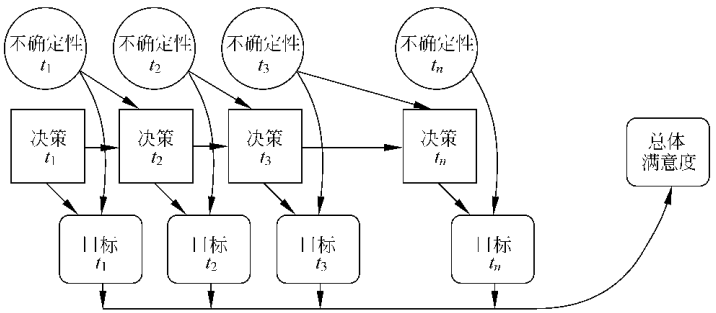


图 5-8　多时期序列决策

可以看到，正如会影响当前时期的量化成功目标一样，某个时期所做的决策会影响覆盖所有时期的总体量化目标。另外，下一个时期的决策（有可能是主要的条件）受到前面决策一定程度的影响。用序列结构模型，决策者能清楚地知道对普通目标的决策之间的联系，就有机会在将来某个时间点上改变某些过程，有可能在某个时期采取稍微激进的战略 如果失败，接下来能用较保守的方法来减轻前期带来的损失。同时，这个模型结构使决策者能看到将来，从而可以决定推迟行动，直至未来某个时期。

5.2　决 策 模 型

除了典型决策支持系统模型库里众多的用于解决个别问题或问题类型的分析模型之外，决策者必须关心到问题的环境的结构。在这点上，一个问题可以想像成一组可按决策者期望功能分解的子系统的集合。每个子系统都可被建模，这样对该部分问题的准确描述就可以被符号化。一旦这个简化的现实描述是恰当的话，决策者可以用这些模型来作为他们后续分析和预测的基础。在这部分，我们来看各种不同类型的决策模型以及它们在特定问题类型中的应用。

决策模型可用多种方法分类。一种常用的高层次的分类方法是按照它们直接包含的时间因素来划分模型。没有直接涉及到时间的模型被称作静态模型（static model），反之称作动态模型（dynamic model）。另一个普遍的划分方法是通过技术上是注重数学还是逻辑来划分。

抽象决策类型

抽象决策类型 (abstract decision model) 致力于用数学方法预测各种可能的结果。这种类型的模型理论上能帮助测定哪个方案会导致有利的结果。基于这个预测,决策者可以制定一个战略来执行被提议的方案,这样就解决了问题。这听起来很好,但是,当进一步分析并分解每个问题前后的子系统时,对决策者的一些信息细节会丢失。这种在建模过程中的信息丢失是因为我们试图建立一个现实的简化描述,为了简化的目的我们在提取信息时去掉了某些片段。决策者经常需要在以下两种模型之间权衡:抓住问题复杂性的模型与足够简单、易于分析解决的模型。这种权衡永远不会达到完全的平衡,这样决策者需要注意到数学建模上的"谦逊信条":所有模型都是错的;但一些模型是有用的 (Golub 97)。尽管需要注意,但数学抽象仍然是有用的,尤其是当决策者知道他们的局限性后。

与功能分解的概念一致,抽象模型可以划分为四个子系统,每个都显示特有的性质 (见表 5-1)。

表 5-1 决策模型分类

• 确定性的	最优化,线性规划,财务计划,生产计划,凸性规划
• 随机的	排队论,线性回归,逻辑分析,路线分析,时间序列
• 模拟仿真	产量建模,运输与后勤分析,经济计量
• 特殊领域	标准经济模型,经济批量采购,约束建模,行为建模,技术扩散,气象模型

确定性模型

一个确定性模型 (deterministic model),其变量在任何时候都不能取超过一个的值。在这种模型中,给定的输入变量总是产生相同的输出值。多数标准经济理论的传统模型都是确定性的,正如在金融分析里应用的许多常用模型一样。这些类型的模型在一个问题含有大量因素和复杂关系时显得特别有用。常见的确定性模型包括线性规划 (linear programming),非线性规划 (nonlinear programming) 和微分方程 (differential equation) ①。

随机模型

在一个随机模型 (stochastic model) 里,至少有一个随机变量被某些概率方程所描述。这些模型通常称为概率模型,因为在它们的结构中直接体现了不确定性。当构造一个随机模型时,决策者把一到多个变量的输入数据分布在某个平均值左右,得到的输出变量是一个概率分布而不是一系列离散值。通常说来,和确定性模型相比,在其他都相同的情况下,一个有着相同元素的随机模型要难处理得多,解决起来也更复杂。通常的随机模型构造技术包括:博弈论、排队论、线性回归、时间序列分析、路径分析和逻辑回归 (或叫做逻

① 有关这些建模技术的详细内容超出了本书的范围。在应用这些数学工具时,可以参考一些管理科学的相关教科书。

辑分析)。

模拟模型

对决策者来说,不幸的是问题的结构并非轻易属于严格的确定性或随机领域。更通常的情况下,一个问题的上下子系统,有些是确定性的,有些是随机的,有些是两者的结合。考虑到这种在问题结构中结合的复杂情况,人们开发了一种叫做模拟模型(simulation model)的技术。一次模拟可以看作进行一到多次实验,用来测试当这些结合起来的子系统处于一个动态环境中时,产生的不同结果。模拟仿真的意图就是要仿造现实而不是简单地做现实的模型。模拟建模主要用于完全改变条件来选择可行方案需要付出极其昂贵的代价的情况下。

另外一些已证实模拟确实有价值的领域,包括对观念测试的检验——用最少的资源和时间检验一些新的创意——研究当前程序和运作以期得到更深入的了解并且寻求一些改进的方法。以下情景描述了一次模拟,这里模拟建模将是一种合适的解决问题的技术:

> 到现在为止,Steve Hornik 在邮政局长的位置呆了一年多的时间了。自从他来到这里,克服了不少运作上的不足之处,邮局的顾客对雇员们提供的服务一般都感到满意,业务也增加了。然而,一直困扰这位新邮政局长的是每天在不同的时间里,顾客们经常在两个仅有的服务窗口外排成长队。尽管雇员看起来尽可能地有效和高效率地为顾客服务,Steve 揣测是否需要更多的窗口以及精简在这些窗口上的流程,来减少排长队导致的等待时间。经过深思熟虑,局长把他的目光集中在以下四种可能策略之一上:
>
> 1. 增加一个只负责检索并递送包裹的窗口。在这个窗口用一位专职雇员同时处理包裹的放置和检取工作。
> 2. 采用和策略 1 相同的策略,只是用了两位雇员:一个专门负责放置,另一个负责检取。
> 3. 增加两个能办理任何业务(包括包裹检取)的窗口。这些窗口在任意规定工作时间内都开放。
> 4. 采用和策略 3 基本相同的策略,只是在业务高峰时期开放这些额外的窗口。
>
> 局长相信他所设计的四个策略之一能有效地解决问题。但问题是他不知道选择哪一个,因为他不能老是在邮局的墙上开洞建新窗口来验证哪个方案最可行。

在这种情况下,问题不能通过静态的抽象模型来得到有效的分析。在这个模型里,许多变量不仅是动态的,而且它们之间的关联随着每天、每周甚至每季度而变化。此外,实地调查每种场景不仅花费大量的时间和金钱,而且消极地破坏了邮局现在的正常环境。然而,运用模拟模型,这些不同的场景就能得到相互比较,在几个小时内能校验到随着天数、周数甚至年数的变化所发生的情况。决策者很容易设计一系列模型来说明每天不同服务要求(比如包裹检取、放置、购买邮资等)的期望频率、顾客到达和离开的分布、窗口

数、完成单个服务的速度等。根据不同模拟模型的运行结果，决策者就能确定合意的可行方案，在规定的目标中达到最好的均衡。

尽管有它的吸引力，但模拟模型也不是没有缺点。首先，一个模拟模型不能保证结果是最优的。这个模型仅仅告知决策者可能的结果和模拟场景之间的比较（是否记得有限合理性？）。其次，构造一个模拟模型，虽然通常是最合算的，但非常耗时和耗钱。再次，模拟的结果很显然与特定的前后背景有关而不能普遍适用于其他问题领域。最后，由于模拟对于管理者来说是一个吸引人的、有趣的方法，比它能得到相同甚至更好、更准确结果的分析方法经常被忽略。不过，当认识到模拟的缺陷时，它确实是一种有力的抽象分析模型技术，适当地运用它能得到积极的结果。

特殊领域模型

第四种抽象模型是特殊领域模型（domain-specific model）。科学的发展产生了许多高度专业化的决策制定技术。在这些情况下，每个学科都开发出自己的一系列抽象数学模型来满足自身需要。供应需求模型是独特的经济学工具。其他广为人知的领域，比如运筹学、社会学、生态学、医药学和气象学等，都开发出自己的抽象数学模型。在为一个专门的问题领域设计决策支持系统时，设计者必须知道模型库中所提供的任何特殊领域模型。

概念模型类型

尽管有多种抽象数学模型可供选择，并且它们在辅助构造问题中具有独自的和结合的能力，但有时用正规的数学方法来解决问题并不是最合适的策略。比如，抽象模型也许会过度简化了问题的结构，数学建模过程也许会太耗费时间和金钱。另一个例子是当决策者非常熟悉问题的结构时，一个抽象的表述就不会起作用。这些情况下，一种概念模型（conceptual model）在预测不同的可行方案集的结果时，显得更为合适。

概念模型可以看作对问题前后情况的类推。这个观点隐含的想法就是所有问题都是独特的，但没有问题是全新的。从过去一个问题的前后结构的经验，可以用来帮助预测新情况的事件和结果。用这个方法，决策者可以回忆和运用过去的多种经验和前后情况来建造当前情形的概念上的模型，并给那些不确定的事件成分赋予一个概率。

概念模型常常被批评为过度主观，并偏重于决策者个人的观点。虽然这个争论当然有些价值，但我们也必须认识到选择某个抽象模型方法同样也是一个主观决策，取决于决策者在技术选择上的意见。此外，抽象模型同样基于主观的事物联系的数学假设。我们可以说，两种方法在建模和使用上都含有个人的偏向和假设。决策支持系统的设计和使用者应该完全知道一种专门技术应用于给定的问题背景时，它的可取（或不足）之处。概念模型和类推论证将在第 9 章应用于人工智能和知识工程时详细探讨。回到我们的邮局问题，我们来看一个应用概念模型的例子：

> 邮政局长 Steve Hornik 已经完成了对四种顾客服务情景的模拟，结果证明对改善排长队是很有效的。因为他对邮政大厅成功的重新设计，Hornik 被任命设计并测试一个免下车（drive-up）的邮件窗口。已经有五个不同地点得到财力

来测试这个新概念，每一处都告知是具有创造性的，并试验哪里适用。他对有这个机会感到兴奋，但还不知道怎么实行它。因为这个邮局为一个较小的社区服务，大部分顾客是步行到达的。那些坐车来的顾客通常在商业区的大停车场放好车后，步行来到目的地。Hornik 不知道这是因为中心停车设施带来的便利还是因为以前没有一种类似免下车设备的选择。他设计了三种情景来区分不确定性：

非常接受：邮局的顾客将会广泛使用这个新的免下车设备，因为使用它传送一些大的包裹和邮件很方便。

拒不接受：邮局的顾客将拒绝使用这个新设备，因为它要求他们改变去邮局的时间安排，比如在停车购物之前或购物之后。

一般接受：一些老顾客会发现这个新设备能带来好处。总的说来，免下车窗口将填补业务范围的一个空白，并将是比较有用的。

Hornik 得出结论，在三种可能的场景中，非常接受的可能性最小，一般接受的可能性最大。因此他给下面三个结果的概率赋值：P（非常接受）= 10%，P（一般接受）= 60%，P（拒不接受）=30%。

Howard 的清晰度测试

不管决策者是用抽象模型还是概念模型或是两者的结合，建模过程的最后一步都是保证模型上下结构的所有组件都已清楚地定义了。Howard（1988）提议用一种有效的方法来确定何时变量和结果的概念都已经被充分定义和区分开：

> 想像有一个能得到将来所有信息的先知……这个人是否能确定结构图中发生任何事件的结果？没有对先知要求任何解释和判断。（引自 Clemen 1991，55）

给出以上情景，在没有解释的情况下，模型每一个结点的事件都能完全确定吗？如果肯定地回答这个问题，那么这个决策模型就通过了清晰度检验。不能对决策结构的基本构件误解。应用清晰度检验一个结构化模型，是一项对各个独立组件不断改进的过程。只有当所有的因素都得到改进后，决策者才继续考虑选择一个解决方案。

5.3　概率的类型

到现在为止应该知道，假如没有不确定性存在的话，决策便不称为决策了。如果所有的事情都像水晶般透明，那么决策就变成一个机械的过程，逻辑和常识就是主要的工具。这里重要的一点是，事实上所有的决策都含有不确定因素，要想在做决策时有一点成功的把握，那些不确定性应该以某些方式来量化。这就是理解概率的概念的重要性所在。

现代决策分析一个主要的原则就是通过恰当应用基本的概率，我们能符号化任何一种不确定性。尽管概率论明显适用于描述不确定性，但它是制定决策中最被误解和误用的工具。应用恰当的话，概率论在帮助决策者进行问题的构造、解决和选择方案时有着不

可估量的作用。但如果应用不恰当,会导致决策的其他方面都很好,惟有结果不对。在下面的几部分,我们将探讨概率论的基本原理,并把目光集中在各种估计和预测不确定事件概率的方法上。

概率的三个必要条件

无论在哪个场合,对概率论的适当应用需要满足三个条件。表 5-2 包含了这一系列的要求。

表 5-2 概率的三个必要条件

1. 所有的概率必须在 0 到 1 的范围内
2. 所有独立事件发生的概率总和必须等于它们联合发生的概率
3. 一个完备的事件集合总的概率必须等于 1

第一个条件暗示了所有的概率都是非负的,没有事件能以小于 0 的机会发生。反过来,很容易知道也没有事件能以超过 100% 的机会发生。

第二个条件给一个特定事件多种结果的概率增加了一个数学限制。假定这些结果是互相排斥的——就是说,如果一个结果发生,那么另外的一定不会发生——那么它们其中之一发生的概率一定等于它们各自概率的和。如果公司实行一项新政策有 25% 的机会增长员工的士气,35% 的机会保持原来的状态,那么士气增加或者保持不变的概率就是 60%。(士气低落的概率是多少?)

第三个条件恐怕是对成功决策最重要的了。如果我们知道给定一个事件的所有结果集,至少我们知道哪些不可能发生:不包含在所有可能结果中的结果是不会发生的。如果打一场棒球赛,我们不知道哪个队能赢,但我们确实知道可能的结果只有两者之一会赢或输。因为规则不允许出现平局,我们就知道这件事发生的概率是 0。一旦我们知道一个事件互相排斥的结果集,我们就说有一个完备的集合。这个所有可能结果集合的概率和必定等于 1。换句话说,如果我们知道一个事件必定发生的所有可能结果,那么其中之一发生的概率是 100%。如果包含在一个决策环境中的不确定性是以这三个要素来构造的,那么概率将是适用的。一旦我们建立了合适的概率,我们就将注意力集中在各种估计的准确性上。我们将在以下几节中详细讨论准确性的问题。

概率的类型

不管一个问题环境是通过抽象方法还是更概念化的方法来建造,最后我们都需要估计各种不确定性的概率。到现在,有三种表达概率的方法:(1) 长期频率 (long-run frequency),(2) 主观概率 (subjective probability),(3) 逻辑概率 (logical probability)。

长期频率

许多概率统计书籍的介绍都以概率作为长期频率的一种衡量。这种方法依赖于所谓“大数定律”,说明如果一个事件或试验重复很多次,那么观察到的某个事件结果的频率

将是这个结果真实概率的很好的估计。最通常用来解释长期频率方法是掷硬币：

从你的口袋或钱包里取出一个硬币并掷向空中，让硬币掉在地上并平放着。硬币正面向上的概率是多少？

对这个问题的回答应该不会引起较大争议或需要认真思考去决定。这个事件有两种可能的结果：正面或反面。我们假定硬币是均匀的，那么两个可能结果是互斥的。给定这个条件，每个可能的结果有 50% 的概率发生。注意到这个简单的分析一样满足了概率的三个必要条件。如果我们重复这个实验很多次，我们将会发现每种结果发生的次数各占 50%。

概率学家经常主张这种长期频率方法。这种方法主张通过试验和重复来估计概率。但有时，按照长期频率来思考会不实际或不可行。考虑需要估计一个核电厂接近完工时发生一次大事故的概率。我们把范围缩小到将来 15 个月内发生核事故的概率，用重复的试验来估计概率是不实际的（即使它是可行的，我们也不会这么做）。虽然如此，我们必须始终需要这种估计，并能合理地依靠这种估计。为达到这个目的，我们必须用另外的方法来思考。

主观概率

考虑以下问题：

- 在 1986 年美式足球超级杯比赛中芝加哥熊队和新英格兰爱国者队开始比赛前，投掷一枚硬币正面向上的概率是多少？
- 1843 年 James K. Polk 当选美国总统的概率是多少？
- Button Gwinnett，Caesar Rodney 和 William Whipple 都是美国独立宣言的起草者的概率是多少？

如果以上问题的答案一时没有想出来，不要紧。在每种情况下，事件都已经发生，其真实概率都已经知道。虽然如此，如果你不知道答案，那么你仍然不能确定。让我们回到掷硬币的例子中：

用前面例子的那个硬币掷向空中。然而，这次在它掉在地上之前用手接住并用另一只手盖住。不要看这个硬币。这个硬币正面向上的概率是多少？

如果你像多数人那样，首先的反应就是，这次出现正面的概率和前面的那个例子一样：50%。你的理由是，在没有提供新的信息，只知道掷向空中后用手接住，这并没有改变结果的概率。这样，概率仍然是 50%。然而，你判断问题概率的第一感觉错了。在这个试验里，你的不确定性不在于硬币随机出现的面，而是你不确定哪个结果会出来。你的概率估计是在事实之后作出的。这个不确定性完全是你自己的想法。这种情况下，概率的估计不是结果本身的衡量，而是你对结果不确定的衡量。从一个概率学家的观点来看，答案是 100% 或 0，而你就是不知道是哪一个。

类似这种情形需要一个和传统长期频率方法不同的观点。主观概率方法把概率看作一个人对一个特定结果会发生与否的“相信程度”。采用这种方法不只是把概率估计作为“信念的艺术”。接受它就意味着一系列隐含的假定，这些假设如果满足的话，就能推

出一系列描述我们信任程度的数值。更重要的是,与概率的必要条件范围内,它们结合起来能够推断出其他的一些概率值的概率。几个研究人员已经成功地开始对这些假设进行发展和正式化(比如 de Finetti,1964 和 Savage,1954)。表 5-3 包含了这些概率公理,使我们能主观地估计概率。

表 5-3 主观概率估计的公理

- 任意两个不确定性事件,或者 A 的可能性大于 B,或者 B 的大于 A,或者两者相等。
- A_1 和 A_2 是两个互斥事件,B_1 和 B_2 是另外两个互斥事件。如果 A_1 的可能性不比 B_1 大,A_2 不比 B_2 大,那么(A_1 和 A_2)不比(B_1 和 B_2)大。进一步,如果 A_1 小于 B_1,A_2 小于 B_2,那么(A_1 和 A_2)小于(B_1 和 B_2)。
- 一个可能事件的概率不能比一个不可能事件的概率小。
- 假设 $A_1, A_2, \cdots, A_n$ 是一个无限增长的事件序列,就是说,对任意 n,如果 A_n 发生,那么 A_1 一定发生。进一步假设对任意 n,每个 A_n 的可能性大于等于其他某个事件 B,那么所有事件的发生 A($n = 1, \cdots,$)小于等于 B。
- 能试验出一个数字的结果,使得结果的每个可能的值在一定范围内是相等的。
- 如果一个人能或愿意根据以上公理表达他对可能性的判断,那么必须有数字对他所感知满足概率微积分条件的事件进行描述。在定义时对这些规则的一致性看法在对不确定估值时是合理的。

资料来源:Watson, S. R. and Buede, D. M., Decision Synthesis: The Principles and Practice of Decision Analysis. Cambridge University Press, 1987.

"共同意义"和"数字需求" 当研究主观概率估计这个课题时,我们需要简要回顾一下在第 4 章介绍的"共同意义"概念。如果我用某种表达或短语来描述某事物,你也能和我一样地理解,我们就说在看待这个短语时我们有一个"共同意义"。如果你不理解我的意思,那么我自己需要改述或重申。当用于概率估计时,"共同意义"的概念变得极其重要。我们可以把一个事件的概率描述成类似"普通"、"很少"、"罕见"、"经常"、"典型"等许多词语。然而,在使用这个方法时的问题,就在于当我们确实需要沟通时在我们之间需要有一个"共同意义"。来自心理感知的研究表明不同的形容词和短语,对不同的人在不同的场合有着不同的意思。Beyth-Marom (1982) 研究了某些描述性的短语和特定的概率估计之间的联系。对于短语"有一个不可忽视的机会……"给出具体概率估计在 0.36 到 0.77 之间。另外,我们可以根据估计的上下关系来改变我们的主观估计。考虑一下,"今晚可能要下雨",这个描述包含的主观概率估计以及含义和下面这句话完全不同:"今晚可能来五级暴风"。"共同意义"在概率估计中是重要的,并且由数字加以描述。5.4 节包含了几种得出主观概率的数字估计的方法。

逻辑概率

对第三种概率类型应用通常限制在那些准确性与成功等同并且涉及到概率的情况下。逻辑概率的概念表明,即使可以得出一个概率,其准确性在任何情况下都是不可接受的。下面的例子阐述了这个概念:

Betty Boggess 收到了来自她的医生的不幸的消息。根据近来一次胸部 X 光

透视，医生发现她的肺里有一个小肿瘤。医生解释说，这个肿瘤可能是良性的，也可能是恶性的。如果是良性的话，对她的健康没有什么危害，就没有必要去除。但如果是恶性的，那么它必须立即去除掉，否则将是致命的。这种肿瘤有 90% 的可能不是恶性的。医生然后接通电话，安排 Betty 立即来进行手术，为什么？

初看起来，医生决定立即安排手术，似乎是违反直觉并且是急躁的。一次组织切片检查能确定肿瘤的性质，就没有必要对身体进行大手术了。这确实如此，为什么医生在动手术之前不来一次切片检查呢？

这个答案就在于逻辑概率的概念。尽管切片检查能确定肿瘤的状况，但准确性不是 100%。如果从肺提取的病理组织检查是恶性的话，那么就应该立即进行手术。然而，如果提取的不是肿瘤的一部分（甚至是实际肿瘤外部的一个细胞），那么检查其是否恶性起了负面作用，手术也变得无实际意义了。问题不在于检查本身的准确性，而在于正确做法的逻辑概率。确认肿瘤的位置是任何一点小错误都不能容忍的。因为医生不能 100% 确定这个检查的组织是从实际肿瘤中提取的，他没有别的逻辑选择只有动手术了。应用逻辑概率的情形通常在某种意义上表明，对一个概率的估计小于 100%，就和 0 是相同的。

5.4　预测概率的技术

所有类型的概率都必须用数字来表述。使用长期频率的概率预测技术（比如统计回归、时间序列分析或者网络优化等）有许多，对它们的描述和应用也是极其繁多的。同样地，讨论这些技术超出了本文的范围，我们将不把注意力集中到这里。本节集中引出了一些可以用概率定律来和其他估计方法结合的技术，如数字化的主观概率估计等。这里包括了三种这样的方法：(1) 直接预测（direct forecasting），(2) 几率预测（odds forecasting），(3) 比较预测（comparison forecasting）。

直接概率预测

直接方法是所有引出主观概率的技术中最简单的。正如其名，它非常直接：向决策者或专家询问从而估计某个时间一个结果的概率。如果信息来自一个非常熟练和有经验的人，他经过大量的预测，比如一个专业气象学家，那么预测准确的可能性就很大。反之，如果来自一个经验不够充分，那么结果的准确性就令人怀疑，其有用性也受到很大限制。尽管如此，适当得到一些直接估计对主观概率的数字赋值还是有用且有效的。

几率预测

第二种引出主观概率估计的常用方法主要是一种赌博的观点。几率预测的目标是寻找赢或输一定数量的金钱，让决策者愿意接受这个打赌的任意一方。一旦确定了这些数量，它们就可以转化为几率，然后从那里来对概率进行估计。这就是几率设置的基本概念，这样的例子有比赛事件。

如果一个人开赌局，把某个比赛事件的赔率设为 2 比 1，暗示这人对每个队伍胜出的

机会看法。几率只是对主观概率的另一种表述方法。上面这个例子中他相信看好的队(几率大的那一个)胜出的机会是输掉的机会的两倍。此时,他对你赌哪个队是无所谓的。如果你选择看好的队并且他们胜出,你得到平均的奖金;如果他们输了,你就输掉了赌注。然而,如果你选择不看好的那支并且他们获胜,你就赢得双倍奖金;他们输了,你也输掉了赌注。每种情况下,无论选择哪一方的期望收益都是一样的。考虑以下场景:

迈阿密热队本赛季的情况非常好。他们如此出色地闯进了第二轮的 NBA 总决赛。他们面对的是七次进入总决赛并四次夺冠的芝加哥公牛队。尽管热队从来没有在决赛中赢过公牛队,但他们是惟一一支在常规赛中击败过公牛队的队伍,一次还是在公牛队的主场。决策者面临两种可能的打赌:

如果公牛队赢就赢得 X 美元,如果公牛队输就输掉 Y 美元。

如果公牛队赢就输掉 X 美元,如果公牛队输就赢得 Y 美元。

X 与 Y 的数额可以想像成放进赌注的金钱。赢家得到所有赌注而输家什么也没得到。这两个打赌可以说是对称的,因为它们彼此互为镜像。图 5-9 为此例的决策树。

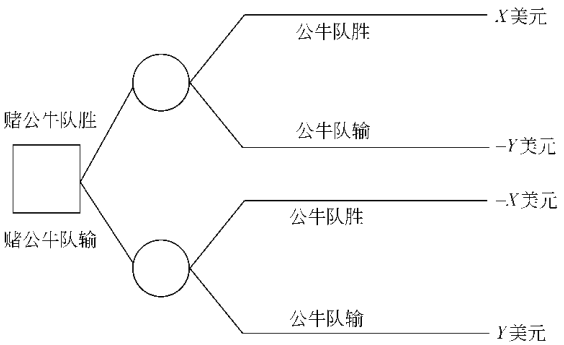


图 5-9 几率预测方法的决策树

如果 X 和 Y 的值都已设定,让决策者觉得选择哪一个打赌都无所谓,我们可以用下面的数学形式来描述:

$$X \cdot P(\text{公牛队赢}) - Y \cdot [1 - P(\text{公牛队赢})]$$

$$= Y \cdot [1 - P(\text{公牛队赢})] - X \cdot P(\text{公牛队赢})$$

从这里,我们得到:

$$2\{X \cdot P(\text{公牛队赢}) - Y \cdot [1 - P(\text{公牛队赢})]\} = 0$$

如果两边除以 2,并把左边展开,我们得到:

$$X \cdot P(\text{公牛队赢}) - Y + Y \cdot P(\text{公牛队赢}) = 0$$

现在合并同类项:

$$(X + Y) \cdot P(\text{公牛队赢}) - Y = 0$$

上面的方程简化为:

$$P(\text{公牛队赢}) = \frac{Y}{X + Y}$$

现在我们需要做的所有事情是确定 X 和 Y 的值,这样我们就能计算一个主观概率的

估计了。假定如果公牛队赢的话，决策者愿意赢 100 美元 如果公牛队输，愿输 225 美元。他的主观概率估计可以计算如下：

$$P(\text{公牛队赢}) = \frac{225}{100 + 225} = \frac{225}{325} = 0.6923 \approx 0.70$$

根据以上估计打赌的几率是 7 比 2，看好公牛队。

建立一个无差异赌博的步骤

寻找决策者认为无差异的打赌数额需要一个简单的反复过程。开始为两个打赌的对象中有利的一方建立一个值，然后询问偏向哪一方。根据这个回答，改变不利一方的值使它变得有利。继续这个逐步调整的过程，直至决策者认为哪一方都不比另一方强为止。这时，就确立了无差异的数额，并且可以计算出一个主观概率估计。

比较预测

类似几率预测方法，比较方法（有时称为彩票预测）表明决策者在参与两个彩票游戏中作一个选择。这两个游戏是这样构造的，一个是不确定事件，另外一个是知道确定的赢取概率的参考博弈（reference game）。想法就是确定赢取的概率使得决策者对两种博弈都无所谓。不确定事件通常构造成能赢得一大笔奖金或输掉一小笔金钱。比如赢得一次免费的欧洲度假或输掉 100 美元。在一般的参考博弈中，通常用圆框来表示，有阴影部分表示赢，没有表示输。图 5-10 描述了这种方法的决策树以及两种彩票的结构。

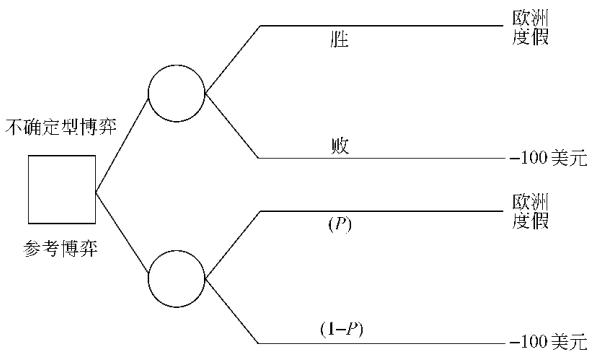


图 5-10　比较预测方法的决策树

和几率预测方法一样，在参考彩票的赢得概率不断地上下调整，知道决策者认为选择哪一个都无所谓为止。假设如果决策者无所谓，这就是他对不确定游戏的主观概率估计。

分析复杂的概率

至今我们看到的例子都相当简单，它们中没有取决于其他概率的概率。然而，实际中发生的许多情况是一个概率取决于一个或多个不确定事件。这种情况也表明自身对前一系列事件的依赖或对任意不同先前系列事件的依赖。在这些情况下，短语“那取决于……”常常用在需要主观概率估计的开头。

当面对一个包含条件概率的复杂决策结构时，决策者必须用一种叫做分解(decomposition)的技术。在这个方法中，我们靠概率规则来确定问题每个可能结果的概率，然后用计算总概率的公式来确定问题中事件的概率。下面的例子解释了这个方法：

LB Industries 是一家位于新英格兰地区的公司，生产一种叫做 HeaTrap 的热回收装置，用于家居电力或煤气炉。HeaTrap 用了一种回收热量的专利技术，每年能节约 30% 的能源。销售副总裁 Debra Herbenick 尝试确定在即将到来的冬天中，HeaTrap 销量超过 25000 套的概率。她需要把她的估计通报给产品处，以便他们能对产品需求作出安排。她从过去年份中知道，HeaTrap 的销量很大程度上取决于他们的市场地区冬天的状况。她估计如果冬天非常寒冷，销量超过 25000 套的概率是 0.80 如果天气一般，概率是 0.50。通常冬季严寒的概率预测是 0.70，一般天气是 0.30。Herbenick 需要快速作出一个决策。

为了确定有一到多个结果所限制的事件的概率，我们需要确定决策树上每条分支的概率。在这个例子中，为了确定销量超过 25000 套的概率 P (A)，三个附加的概率必须先确定：(1) 严寒冬天的概率 P (B)；(2) 如果冬季寒冷，销量超过 25000 套的概率 P (A| B)；(3) 如果冬季一般，销量超过 25000 套的概率 P (A| B')。

Debra 建立了解决问题所需的如下决策树 (图 5-11) 来解答她的问题。为算出销量超过 25000 套的概率，她用以下公式来计算。

$$P(A) = P(A|B) \cdot P(B) + P(A|B') \cdot P(B')$$

通过她所知道的概率估计，她计算出在冬天中销量超过 25000 套的概率，并用她的估计减少了产量。

$$P(A) = 0.80 \times 0.70 + 0.50 \times 0.30 = 0.71$$

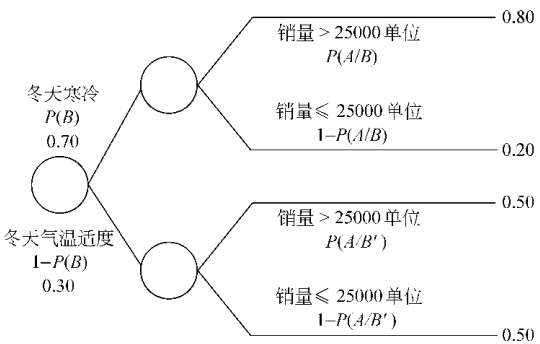


图 5-11 LB industries 问题分解的决策树

DSS 中的混合预测技术

应该比较清楚地知道大部分概率预测技术，无论是客观还是主观，可以很容易地包含在一个典型的 DSS 的模型库中。设计者面临的实际问题是该包括哪些，在什么情况下提供给用户使用。

实际中,在考虑选取一项合适的预测技术时,没有“绝对的真理”。研究表明,决策者对工具的选择并不在于其独特的洞察力,而是感到适于他使用就行了。因此,培训用户恰当、有条件地使用技术的责任就落在 DSS 的设计人员身上了。尽管这种选择不总是能够清楚地划分,但有一些要素可以促进选择其中一种,而不是另外一种。表 5-4 包括了这些能影响选择的要素列表。

表 5-4　影响选择一项预测技术的因素

问题的结构	复杂度	问题的规模
精确	有效性	DSS 设计者偏好

资料来源:摘自 Davis (1998)。

尽管列出的每个要素能影响选择某项预测技术,其最恰当的应用也只是解决了问题的一半。即使选择了正确的技术,估计的来源也必须足够熟练,能可靠地提供一个预测。当一种技术比如直接预测被认为是合适时,这就变得更为重要。本章最后一部分来处理这个问题。

5.5　把握和灵敏度

良好地把握

在多数情况下,我们并不需要某个结果或事件的确切概率。实际上,即使我们能知道一个事件发生的确切概率,它仍然是一个概率。一个概率只是详细说明一个平均值的专门方法,而一个平均值仅意味着给定足够的时间,事件会经常按此发生。尽管平均值是一个有用的描述值,但它同样容易误导人。举个例子,一个人把左手放在一桶沸油里 (492. 7°F),右手放在一桶液氮里 (- 300°F)。取个平均值,他会感到很舒服 (98. 6°F)。也许这有点夸张,但你应该能知道这一点。如果我们要一个完善的决策的话,我们就需要比一个简单的概率估计更多的信息。

不管预测者的技巧如何以及他估计的准确性,意外总会发生,事情并不是按照我们所想的那样顺利进行。和单个概率估计相比,也许我们通过最坏和最好的场景估计能得到更多的益处。适当地进行构造,我们可以断言真实的概率在于这两个值之间的某个地方。这个值的范围被称作置信区间。一个决策者可以有 90% 的信心指出,事件结果的概率会落在估计的最好和最坏场景的值之间。用更多额外的信息我们可以可靠地进一步决策。

一项估计有一定的置信区间,但不要仅仅因为这个就觉得可靠了。决策者或专家的作类似估计的能力,对可靠性将有重大影响。为检测这个,我们让决策者对同一个主观概率作 10 次独立的估计。假设有 90% 的置信度,概率上 10 个估计中应有 9 个以上包含了真实值。如果这个情况发生,我们说决策者关于这种估计是有良好的把握的,我们可以假设评价者清楚地知道他的预测能力。反过来,如果估计准确的数量小于 9 个,决策者对自己在该领域内提供有效估计的知识和能力就是过于自信了。

一个好的把握程度来源需要多年的经验和反馈来提高。专业气象学家被发现通常能很好地做出把握,恐怕这是一个原因。他们已经天天、年年作出同样的估计。每次他们的

估计都以天气确实发生的方式得到回馈。随着时间增加,他们在作类似估计时变得非常熟练,他们的置信区间变得很小。

然而,许多决策者在通常的知识领域里没有如此好的把握。我们在作通常知识的估计时置信区间一般设定的很小,这意味着我们对世界的基本认识通常过于自信。尝试一下表 5-5 的系列问题,看看你对常识有多少把握。

表 5-5 常识的把握检测

对每个问题,为你的答案提供 90% 的置信区间。一些问题的答案可能对于你来说相当熟悉,然而其他可能不是。根据这些改变你的置信区间。

1. 美国 15 岁以上公民受过教育的百分比是多少?
2. 当前美国人口总的期望寿命是多少?
3. 美国国旗上有几个条?
4. 在 1993 年,美国总共拥有多少电视?
5. 美国共有多少个机场?
6. 西奥多·罗斯福担任过美国几届总统?
7. 在 20 世纪 80 年代哪个 NBA 球队在开始常规赛时创下连续输球的记录?
8. 除了弥撒之外,在天主教堂最流行的活动是什么?
9. 在 1987 年 8 月谁辞去 PTL 主席的职务?
10. 前白宫参谋长约翰·苏努努曾在哪个州执政三届?

提示:答案在本章末尾。

灵敏度分析

一旦所有的估计都已经作出,决策者就可以根据手头上的知识来选取最合意的可行方案。假设模型都真实地反映了现实,每个隐含的假定都是合理的并且模型都适当地构造,决策者应该处在一个适当的位置来作出一个明智的决策。剩下的问题是"对假设的假设"必须合理,否则整个过程的成功就会受到怀疑。

不要因为模型提供了某一种方法或行动路线,就认为它是惟一的选择。即使模型内在的假设看起来很合理,也许存在其他同样合理的假设集。灵敏度分析是一种用来确定改变内在假设对问题结果实质性影响的程度的方法。这种方法的另一个常见名称是 What-If 分析,因为决策者对"如果"改变了这种或那种状况所产生的不同结果很感兴趣。

对一个或多个模型进行灵敏度分析可以实现许多有用的目标。一些模型稍稍改变输入,其结果就有巨大的波动。如果需要的话,这样的模型可以识别和调整。对输入敏感的变量可以用来测试稳定性的范围。模型可以调整来反映改变问题条件时真实世界的灵敏度。不管目标如何,一个模型的灵敏度和一个模型的准确构造对一个决策情景的成功是同样重要的。

尽管现有许多方法引导决策者进行灵敏度分析,但它们都遵循同样的基本过程。第一步是把模型的所有变量设为常量或期望的状态,然后选择一个变量来研究,在它的变动范围之内或范围之外变化,记录相关的结果。如果对这个变量一个小的变动导致结果变

化显著,那么我们称它是高度敏感或显著敏感的,把它包含在模型内不但必需而且是恰当的。反之,如果对一个变量变动超过预期范围也只对结果产生一个小的影响,我们就称它是相对不敏感或不显著的,可以把它从模型中完全除去或定为常量,这样从影响可能结果集的变量中去掉了一个:

> 一个变量在以下两种情形可变成常量:或者它单位变动产生重要的影响,但它处于极小的范围内;或者它单位变动产生无关紧要的影响,尽管它有一个很大的范围。(Howard 和 Matheson 1968,719)

可以看到模型的变量越多,潜在的灵敏度结合交互作用越大。这个自动决策支持是另外一个对决策者无用的领域。

价值分析

在制定一个决策之前,最后的考虑是判断是否有足够可靠的信息来做一个成功的决策。因为我们许多日常决策需要面对不确定性,我们通常需要仔细考虑是否能改进对一个不确定事件的认识。如果能确切知道一个不确定性导致的结果,我们就说拥有这个事件的完全信息(perfect information)。然而,如果我们拥有估计一个不确定事件概率的信息,并且能通过得到附加信息来得到更好的估计,但没有达到确定的程度,我们就说拥有了"非完全信息"(imperfect information)。问题变成我们能提供多少时间和金钱(实质上是一样的)来获得额外的非完全信息促进成功的机会。

回答这个问题的一个方法是用期望值(expected value)来确定各种相关信息的价值。一个称为完全信息的期望价值(expected value of perfect information,EVPI)的方法,可用来确定我们是否有必要花费额外的资源来获取额外的非完全信息。

假如我们咨询一个先知(是否记得 Howard 的清晰度分析?)关于决策树中每个不确定事件的真实结果。这种情况下,决策者可以确切知道在发生某事之前会发生的一切,这样,就可以制定一个 100% 准确的决策。这将导致制定一个决策的期望值和实际值是相等的。换句话说,如果一个决策的期望值是 100 万美元,我们有 100% 的成功率去实现,这样,实际值也是 100 万美元,两者是相等的。因为我们知道不可能有 100% 的信心去作一个决策,我们必须根据模型中每个可能结果的各种概率计算一个期望值。考虑以下例子:

> 某城市可能答应加入一个由其他城市结合的联盟,这个联盟的目的是和 Televiz Cable 公司协商取得将来有线电视经营权。Doug Peters 议员有点犹豫是否该投这一票。他不敢肯定付给联盟 5 万美元会费所取得的收益,是否会大大高于独自与 Televiz 公司协商经营权所取得的收益。他决定用一个 EVPI 计算来确定它的价值。

假设完全信息,即能从联盟得到的确定收益是 37.5 万美元,而不确定情景的期望收益是 20 万美元,那么 EVPI 的成本是 17.5 万美元。因为加入联盟的费用比 EVPI 少得多,

正确的决策是投赞成票。

5.6 小 结

决策和建模是有效研究一个问题的基本技能，更重要的是获取和组织所需的与它相关的信息。清晰地定义和系统地阐述是致关重要的。本章着重讨论了识别和解释构造数学模型以及为 DSS 设计者提供的用于优化模型库的技术。

另外，要理解如何阐述一个问题，一个 DSS 设计者必须也要了解解决问题可选途径。现代基于计算机的 DSS 可以有多种形态，但其本质总是内在的模型库，没有内在模型就没有 DSS。

1. 一个完全成型的问题阐述的三个主要部分是什么？
2. 列举并简要描述问题结构的三个基本部分。
3. 是否所有决策至少有两个可选解决方案？试解释你的看法。
4. 比较两种常用的问题结构分析方法 影响图和决策树。每种方法各侧重什么？
5. 选择和不确定性最主要的区别是什么？
6. 当分析决策时，建立目标的方法有什么？
7. 解释影响图没有回路的原因。
8. 列举并简要描述为了清晰说明决策前后的细节，构造决策树必须遵循的规则。
9. 描述基本风险决策的结构及其变量。
10. 简要描述多目标、无风险决策的结构及其变量。
11. 序列决策的结构是怎样的？说明先前、当前以及未来决策之间的关系。
12. 解释抽象模型的概念。
13. 列举并简要描述抽象模型的类型。
14. 简要描述模拟模型的缺点。
15. Howard 清晰度检验说了些什么？
16. 说明概率的三个规则。
17. 列举并简要描述概率的不同类型。
18. “共同意义”以及数字需求对于主观概率估计是很重要的，解释其原因。
19. 比较几率预测和比较预测的主要概念。
20. 定义灵敏度分析并说明它对决策者的价值。
21. 价值分析的主要内容是什么？

讨论题

1. Tiberon Foods 是一家有 55 亿资产，生产牛肉、海产品和冷冻食品的公司，必须依靠及时、准确的报告来计划其销售和运作。然而，用普通邮件把报告寄给销售队伍不是一个有效的流程。当报告在决策者手中时，通常已经过时一个星期了。因此，Tiberon 决策层决定采用一个平台，用于支持更多的及时分析以及能影响当前数据仓库的远程存取。根据以上说明，用完全成型的问题阐述来描述此问题。
2. 假如你是 Tiberon 食品公司的信息主管（CIO），根据可用的资源来确定问题的范围。如果需要可以作出假设。
3. Alice 是 Maryland 大学的高年级学生，她决定毕业后攻读硕士学位。现在她面临的问题是该申请哪个学校。用影响图或者决策树来分析她的决策。说明其选择、不确定性和目标。如果需要可作出假设。
4. 讨论用决策模型的好处以及限制。
5. 研究你熟悉的某个组织的决策。用决策模型来描述此决策。你选择什么类型的模型？为什么？
6. 尝试用直接、几率或者比较预测方法来预测最近要进行的一场比赛中，你喜欢的队的胜出概率。解释你的预测结果。

常识把握测试答案

1. 97%
2. 75.9 年
3. 13 条
4. 2.18 亿部
5. 13363 个机场
6. 2
7. 迈阿密热队
8. Bingo
9. Jerry Falwell
10. 新罕布什尔

第 6 章

群体决策支持与群件技术

学习目标

- 理解群体决策者（MDM）的概念、基本的 MDM 结构以及基本通信网络的类型
- 理解不同类型的群体问题
- 熟悉 MDM 支持技术的基本概念和定义
- 学习两类不同特色和技术的 MDM 支持技术
- 熟悉六种类型的群件
- 理解常用的 MDM 协调方法
- 领会虚拟工作场合的含义

小案例

普莱诺警察局

由于现在的罪犯们善于利用各种资源巧妙安排以逃脱法律的制裁，警察局总是在想方设法对现有的技术手段进行升级，以便更好进行信息的共享。现在警察局的执法工作中还包括很多与其他权力机构之间的合作，这使得信息的共享几乎不太可能。

得克萨斯州的普莱诺警察局现在使用了 Lotus Notes 系统，并利用系统中提供的一些强大工具对警察局进行了更好的组织，克服了原有的组织中存在的日常交流方面的障碍。这个项目的开发者因此获得一枚勋章，而这样的结果当然也令他非常满意。

“几年前，我看过一篇文章中说到波士顿警察局使用 Lotus Notes 来记录团伙犯罪行为，”兰迪·罗杰斯说。他现在正经营着 Technology Helps 公司，这家公司专营执法部门所用的软件的开发和销售。

“作为 Lotus 的地区销售经理以及前任代理州长，我很有兴趣看看 Notes 是否能够像在企业界那样，在执法部门中帮助组织信息以加强交流。因此我设法获得了 Lotus 和波士顿警察局合作开发的代码，并把它们投放到公共领域，开始开发相关应用，然后交给警察局。”

住在普莱诺的罗杰斯觉得当地的警察局适合利用 Lotus Notes 开发附加的网络应用，以便给所有地方的警察局提供一套完整的跟踪工具。

他先着手建立记录犯罪团伙的活动的数据库，这个数据库可以让警官们跟踪团伙成员的详细情况，其中包括外形描述、活动区域和行动方法的描述，甚至还有照片。很快，北得克萨斯州的邻近的城市也安装了这套犯罪团伙活动记录数据库。最初的效果很明显，在几个月内就帮助警察确定了开车射击和入室抢劫的嫌疑犯。但是罗杰斯知道他还需要提供更多的应用，而不仅仅是追踪犯罪团伙的行动。

“我对整个系统进行了扩展，使系统可以追踪一系列广泛的犯罪行为，例如抢劫、入室抢劫、伪造、行窃、性犯罪等等，”罗杰斯说，“这个系统可以提供给警察局一种完整得多的方法进行文件和数据的交叉检验。它还能够简化内部的管理任务。比如说，现在一个领导可以电子化地查看对某个警官的投诉，并且几乎可以立即做出反应，而不需要经过一系列手续和文件。”

罗杰斯还说，在警察局中使用 Notes 的另一个好处，就是可以轻松地从其他地方性的法律执行机构共享并复制信息。现在普莱诺通过拨号上网的方式同其他 4 个警察局相连接，但在不久的将来就准备增加到 12 个甚至更多。

“我们对结果很满意，”普莱诺警察局的局长布鲁斯·格拉丝考克说，(普莱诺市有 19 万人口，)“Notes 系统的一个最大的好处就是它可以让我们所有的调查单位网络化，所以调查单位可以迅速确定其他人是不是也在调查相同的人。

格拉丝考克还补充说 Notes 系统对于警察来说非常好用，他们只需要握住鼠标点击就可以了。他说，曾经通过一项新法律，对个人记录中的一些资料如何归档产生了影

响,这时候 Notes 系统的适应性就体现出来了。警察局可以在几个小时之内创建一个带参数的数据库来适应这些新规则。

罗杰斯还用 Notes 系统为全国儿童保护中心网络开发了一套全国案件追踪系统。他也同意格拉丝考克的评价。他说:"Notes 是一个很好用的平台,可以开发应用程序、数据库、支持整个工作流程的平台以及通信系统,包括因特网上的通信。"

"警察执行法律的工作方式已经有 200 年没变了,"罗杰斯补充说。"调查人员拿起电话问'你听到枪声了吗?'这就像打自动电话赚钱——就是撞大运。有了 Notes 数据库,一切都改变了。通信方面的障碍不复存在。这使得警方的工作效率提高了很多。"

askibm@vnet.ibm.com

6.1 群体决策的制定

直到现在,我们所关注的都是从个体的观点来看待的决策制定过程。通常一个管理者所面对的大部分决策都是由个人完成的,但在现在的工作环境下也有很多的决策要靠群体来制定。群体决策的方式能给选择的过程带来一些好处:更广泛的知识和经验,更多样化的视角,潜在的协作行为。同时也会带来一些不利之处,如果不加以足够的重视,决策的制定就会有问题,甚至产生很严重的后果。

决策群体常常被比喻为"骆驼"。尽管这种说法不那么准确,但观点却很明确:过多的决策参与者会导致要么得到的决策很差,要么干脆没有结果。许多的高级管理者都厌倦了小组会议,尽量避免这种会议。大多数情况下,问题并不在于有这么多人来制定决策或者参与制定决策,而是在于参与者的组织结构不适合具体决策问题。在这一章中,我们要探究一下群体决策制定的结构和决策问题的内部结构,群体决策的优点和缺点,以及群体决策的决策支持技术。

什么是群体

我们从一些定义开始。直到最近,关于群体的作用以及决策制定的文献中,还没有对很多人广泛参与到问题的解决中的行为进行清晰的定义。群体被简单地定义为由个体集合起来的不依赖于个体属性的实体。还有一些文献把群体描述为一系列的个体,而它们的联合属性就是群体的属性。还有一些文献就用决策问题的结构来定义群体。可以想像,如果对群体的构成问题没有足够的关注,必然会带来一些认识上的混乱和争执。

1991 年时 Holsapple 在这方面迈出了一大步,建议用术语"多人参与的决策制定者(multiparticipant decision maker, MDM)"取代"群体决策的制定者(group decision maker)",从而消除了由于"群体"这个术语所带来的混乱,并且考虑到了各式各样的

MDM 结构的分类。一旦建立起这些结构，相应的也就能反映出各自的特征。现在的研究已经开始注重参与者之间的相互作用，以及在某种条件下选用哪种结构比较合适。

我们的目的是要在 Holsapple 和其他人的基础之上建立起“多人参与的决策”的定义。我们把多人参与的决策定义为“一种集合体的行为，这种集合体由两个或两个以上的个体组成，并且其性质由集合体的公共属性和组成成员的个体属性共同决定”。根据这个定义，我们可以根据组成结构和决策内容对 MDM 进行分类。图 6-1 就是一种对 MDM 等级化的分类方法（在第 2 章有简要介绍），这种分类方法是由 Marakas 和 Wu 在 1997 年提出的。

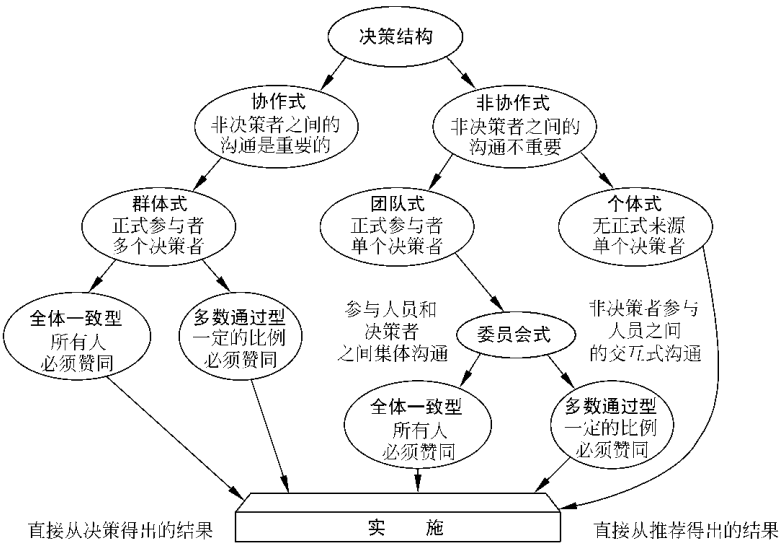


图 6-1　多人参与决策群体结构的层次划分

从图中可以看出，各种不同的结构之间的差异显而易见，很容易确定。使用这种分类方法，可以很容易地确定一种多人参与的决策的结构，并且搞清楚它是否适合解决某个给定的问题。每一种 MDM 类型在决策者以及参与者之间的相互作用上都有特定的结构。图 6-2 给出了几种基本结构。

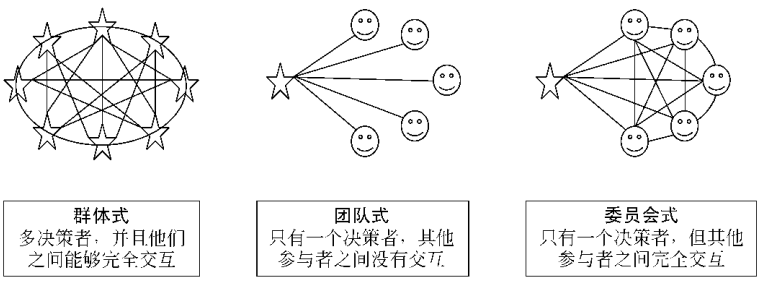


图 6-2　多人参与决策群体的基本结构

使用图 6-1 中的分类法，我们就可以确定三种主要的 MDM 结构的各自的特点。群体式结构是一种协作式的结构，而团队式的结构和委员会式的结构都是非协作式的结构。

这样的分类是基于非决策者之间的交互作用对最终结果的影响程度。而团队式结构和委员会式结构之间的区别,则在于参与决策过程者之间的相互作用。

沟通网络

MDM 的结构首先基于各个成员之间的沟通和交流。在这个意义上,“沟通(communication)”可以理解成在决策群体的成员之间传递各种类型的信息的任何途径和方法。这里包含了两个维度:⑴ 信息流的方向;⑵ 信息流的结构。在图 6-2 中,我们在每一种结构中都能看到这两个维度。

早期的群体内相互作用方面的研究建立起了沟通网络四种基本模式:⑴ 轮式结构;⑵ 链式结构;⑶ 环形结构;⑷ 全连接网络(见 Bavelas,1948;Leavitt,1951)。从图 6-3 中可以看出,每一种 MDM 的分类都可以用这四种基本模式来反映。我们可以根据四种基本模式的集中度来确定。

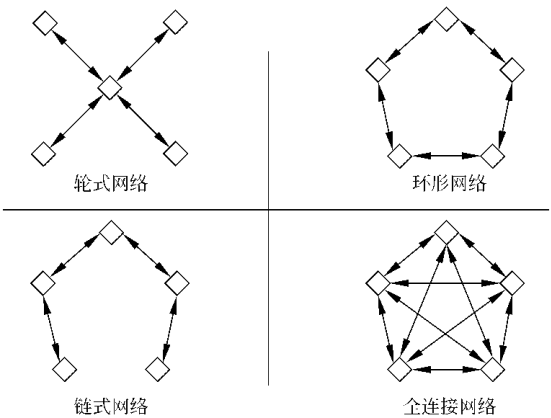


图 6-3 基本的沟通网络结构

轮式网络

一般认为这种模式是四种模式中结构化和层次化最强的。在我们的 MDM 的分类体系中,团队结构最能体现这种模式。在轮式网络中,决策参与者可以和决策制定者沟通交流,但不会和其他的参与者进行沟通交流。而决策的制定者会根据自己的需要和任何一个决策参与者或者全部参与者进行交流。在两个或者多个参与者之间的沟通,都是通过决策制定者和每一个人相互交流达到的。通常情况下,轮式网络比较适合于制定日常的或重复性的决策,在需要优先考虑时间和成本的情况下常可得到高质量的决策。但通常并非所有的参与者都能满意,因为这种模式内部相互沟通有限。

链式网络

这种模式与轮式的相似之处是注重决策的制定者,而在决策的参与者之间缺乏自由的沟通。这种网络可以看成是一种规范的传播系统,某个参与者从其他参与者处得到信息,再把它传递给其他人。最后一个参与者把信息传给中间参与者,而中间参与者则把得

到的信息和自己的信息结合起来，并把结果传递给决策制定者。决策制定者制定出决策，再通过这些传播者传播到最后一个参与者处。以一个五个人的链式网络为例，决策制定者或直接或间接地与所有的参与者发生联系，而且在这个链中，中间参与者与其他两人发生联系，而终端参与者只与另外一个人发生联系。正因为如此，通常相比较而言决策制定者对这种模式最为满意，中间参与者次之，最后一个参与者最不满意。与轮式网络相类似，链式网络也具有很高的集中度，适合于日常的或重复性的事务中做出高质量的决策。

环形网络

与前面两种模式相比，环形网络为每一个成员提供了一个均等的参与沟通交流的机会。在这种模式中，每一个参与者都能得到前面所有信息的汇总，就像决策制定者一样。研究表明，这种模式能够使参与者产生比轮式网络中的参与者更高的满意度。这种模型通常用于群体式或者委员会式这两种 MDM 结构中，或者用于需要将谈判和达成一致意见作为最终决策的基础的某些特定环境中。

全连接网络

这种模式内在的特点就是成员之间的充分交互和完全没有限制的交流。正是由于如此的自由，所以这种模式通常能让所有的成员都对自己的角色满意。但从另一方面来说，由于这种网络中信息充分交流的可能，信息传播相对于其他三种网络更慢，出现错误和信息失真的可能性更大。这种网络是纯粹意义上的群体式结构的基础。表 6-1 列出了这四种网络结构，并且根据集中度进行了划分。

表 6-1　四种网络结构的主要特性

高度集中式的——轮式、链式
• 对例行事务性和重复性的决策比较有效
• 会加强决策群体中心位置的成员的领导地位
• 参与者之间的交互方式比较固定
• 会降低参与者的满意度
• 高度分散式的——环形、全连接型
• 能提高决策群体中参与者的满意度
• 有利于非例行性、非重复性问题的决策
• 有利于做出创造性的决策

群体行为及其规范

根据我们对多人参与决策群体的定义，这种集合体兼具组成个体和整体自身的特点。正因为如此，人们对多人参与的决策建立了一些规范或者行为的标准，以指导决策制定过程并表现了“持续的成员资格的价格”（Harrison，1995）。Homans（1950）对群体标准提出了如下表述：

规范是群体的组成成员的思想中的一个观点，这种观点可以表述为在给定

的条件下成员们应该做什么、可能做什么、期望做到什么。(p. 123)

大多数情况下,决策群体的参与者在群体内的名望,在很大程度上是由他们对已经建立的规范的执行情况来衡量的。决策群体成员的行为以及决策群体的运作流程都要和规范的行为进行比较,来指导以后的行为。这里,"普及规范"的过程显得很重要。在这个过程中,决策群体的成员要互相提醒什么样的行为是合乎规范的,是可接受的。规范的树立和普及可以通过下面的这些方法完成。比如事例法(领导者举出一些正确行为的例子),同事的评论(每过一段时间,互相评价过去这段时间内的行为),奖惩制度(对正确的行为予以褒奖,对错误的行为予以惩处)。不管采用那种方法,一旦建立起规范并在组织内普及开来,就要在组织内组织学习这些规范并且强制执行,以保证这些规范能够长期得到遵守。对于多人参与决策群体的成功来说,最重要的是决策群体的结构、规范和决策问题的内容是否匹配。在为某一个特定的决策群体设计一套支持机制时,设计者必须在组织内几股突出势力间巧妙地取得平衡。我们将在 6. 2 节中详细讨论这个问题。

如何做出决定

正如前面所讨论的,每一种 MDM 结构都有优点和缺点,一旦采用都会影响到决策的产生和成功的可能性。采用哪种方法必须根据具体的决策问题。Vroom 和 Yetton 在 1973 年提出了决定采用哪种决策结构时必须要考虑的若干因素(见表 6-2)。

表 6-2 确定决策结构时要考虑的因素

- 决策质量的重要性
- 决策制定者拥有的制定决策需要的专门知识和经验的多少
- 潜在的决策参与者所具备的必要信息的多少
- 决策问题的结构化程度
- 决策被接受的程度对决策的成功实施的重要性
- 群体成员接受专制式的决策的可能性
- 决策参与者对实现组织目标的渴望程度
- 决策参与者在选择方式时可能产生的矛盾

通过这些因素,可以用一系列试探法告诉管理者如何恰当地把正确的 MDM 结构和决策问题结合起来。表 6-3 中是一个矩阵,两个维度分别是 Vroom 和 Yetton 提出的因素以及 Marakas 和 Wu 提出的各种 MDM 结构。

表 6-3 MDM 结构选择的矩阵

因 素	个人决策	团队式结构	委员会式结构	群体式结构
高度重要		✓	✓	
决策制定者有专家经验	✓			✓
决策参与者有专家经验		✓	✓	✓
高度结构化	✓	✓	✓	
被接受程度非常重要			✓	✓
有可能被接受	✓	✓		
实现组织目标的动机强				✓

存在潜在的矛盾	✓

让我们看几个例子,比如下面的条件对决策产生的作用:相对不重要、高度结构化;决策者可以直接获得必需的信息;决策制定者的专家经验至关重要。这一类的问题通常有:聘请雇员,日常事务安排,或者单独采购的决策。在这些情况下,管理者可以迅速而有效地做出决策,不需要进行咨询和磋商。

相反,如果选择委员会式的结构,则说明需要一定程度上的意见一致。假定决策制定者没有足够的信息、经验或时间独自做出决策。他需要下属的经验以及组织成员或者外部顾客的赞同。用这种方法处理的问题包括:独立的对组织内部的行为恰当与否的调查(例如议会对竞选过程中不正当行为的听证会),采用某种基层组织支持技术的决定(比如采用某种 e-mail 和数据库平台的决定),或者需要广泛的支持和认同的决定(比如任命一个新的部门领导)。从这些例子中可以看出,结构和内容的结合与得到一个成功的决策同样重要。在 6.3 节中,我们会介绍可以提供给各种多人参与决策结构的决策支持技术。在此之前,我们必须先看看决策支持系统的设计者在对多人参与决策提供支持时面临的问题。

6.2　决策群体的问题

对于决策群体,有一点我们可以肯定的是,一个群体的各个成员集中起来会发生一些独特的作用,但我们却不清楚这些作用是什么。尽管在多人参与的决策群体以及各种各样的群体的作用方面有很广泛的研究,但对于管理者是作为个体做决策更有效,还是作为多人参与的决策群体的一员时更有效这个问题并没有普遍的一致意见。很多文献都称赞集体决策达成一致的好处,另一方面也有很多证明这种方式的缺点的依据。在这一节中,我们将集中讨论这个问题,为我们讨论支持多人参与决策群体的方法做准备。

规模

一个特定的多人参与的决策群体的成员数量可能是研究的最广泛的问题,也是研究群体决策制定时自然产生的问题。与个人决策时决策成员的数量固定为一个人相比较,多人参与的决策群体的成员数量可以从最少两个到大得不可想像(比如每四年美国总统选举时的多人参与决策群体的成员数量)。

多人参与决策群体的规模同许多行为科学现象高度相关。研究表明,当群体的规模增大时,个体的满意度就会下降。其中一个原因就是,多人参与的决策群体的规模同成员个人的参与度是联系在一起的。当规模增长的时候,那些不够主动的成员就会明显变得对决策没什么价值了。而且,逻辑学证明,对于需要意见达成一致或者多数达成一致的多人参与决策群体的管理来说,管理一个 5 ~ 7 人的小群体要比 25 人甚至更多的大型群体要容易得多。

另一个使得规模成为多人参与决策群体成功与否的一个非常重要的因素的原因是,群体规模的增大会导致成员的凝聚力的下降。研究表明,当成员很多时,群体终究会出现子群体或者内部组织,这些子群体会把成员的注意力从集体的共同目标转移到子群体的

自身利益上。在有些例子中,子群体的力量非常大,因此在制定决策时必须予以考虑。大公司的董事会经常会碰到这种问题,董事会中的某些成员结合成了一个小团体。政党也经常会遇到这类问题。Shull 等 (1970) 发现,当群体的成员个数增加到 6 到 7 个时,群体中一个个体成员把其他成员仍看成独立个体 (而不是某个小群体的成员) 的情况会迅速减少,而且群体的共同目标的确定变得愈加困难。

还有一个与多人参与决策群体的规模有关的问题是,大的群体的成员越来越感觉受到威胁,或者不愿参与,因为规模的扩大会导致决策问题更加客观甚至没有人情味。通常来说,成员数量越多,成员越有可能得到负面的回应。这种情况会加剧小集体形成的可能性以及成员的不满意度。表 6-4 列示了多人参与决策群体的不同规模的基本效果。

表 6-4 多人参与决策群体的规模的影响

- 决策参与者之间的交互作用会随着规模的增大而减少
- 规模增大时,群体的凝聚力就会降低
- 规模增大时,中央成员的领导地位和支配地位就会增强
- 规模增大时,群体内的矛盾更多的是靠制度来解决,而不是靠分析和协调达成共识

尽管缺点很明显,但大型的多人参与决策群体结构有时对某种特殊的决策问题还是很有好处的。举例来说,在一些必须要定量判断的情况下,决策群体的成员数量越多,判断的结果就越能反映所有人的看法。这可以从统计学加以理解。如果美国国税局需要评估一项新的有关抵税的规则的影响,此时,集合一批在这方面有经验的国税局人员作为决策群体比较有益。

即使不考虑多人参与的决策群体的结构,组成人员的数量也是构成决策群体时必须仔细考虑的因素,它会严重影响到决策支持结构的设计和有效程度。

小组思维

Irving Janis 是一个非常著名的社会心理学家。他是最早发现和多人参与决策群体相关的现象的人之一。他创造了术语“小组思维”(groupthink),来表述“当人们与某个小集团深深联系在一起,并且这个集团的成员们力争达到一致而不是现实地评价采取各种行动路线时人们的思维模式。”(Janis 1982. 9)。小组思维导致了集体思维效率、检验真实性的倾向以及对多人参与决策群体整体的道德上的评价的下降。Harrison (1986) 对 Janis 的定义进行了扩展,提出“群体内成员越友好协作,那些个人的想法就会越接近群体的规范和惯例。”(第 184 页)。小组思维最好的情况下会使决策群体的效率相对降低,而最坏的情况下会导致一系列不利后果。表 6-5 总结了一些和小组思维有关的不利后果。

表 6-5 小组思维的潜在后果

- 导致不可能对目标实现的机会进行虚心的、全面的分析
- 使得决策所需的信息的搜索带有偏见，受到个人实现欲望的影响
- 限制了决策参与者公平、公正地评价其他人的意见的能力
- 经常会导致决策者对失败的可能性及其成本的考虑完全错误，从而使得最终决策的风险超出了其报偿
- 会将偶然性信息完全排除

在小组思维中，“以任何代价达成一致”的思想导致了很多的不利的决策的产生，尤其是在公共领域中，因而受到了指责。

比如说，许多研究人员和历史学家都认为，小组思维这种思维模式在尼克松内阁内非常明显，是造成水门事件的重要因素，并且导致了 1974 年美国总统的辞职。1980 年营救伊朗人质的努力的失败也与此有关。更近一点的，许多研究人员认为在 1986 年“挑战者”号航天飞机失事之前的决策过程中，这种思维模式的存在也很明显。显然，任何用于支持多人参与决策的工具，都必须能够处理并且减轻与小组思维相关的各种问题的影响。

其他的交际上的问题

矛盾

还有一些与多人参与决策群体结构以及组成成员的行为有关的问题需要提到。寻求相互间的支持通常会带来很多的负担。比如说，由于小组思维的存在，希望被别人看成好的成员和被别人接受，通常会导致躲避矛盾。然而矛盾是人类行为中不可缺少的（管理上的决策过程也是这样），矛盾的躲避通常是由于人们意见不和时各自退让。如果这种不和成为决策过程甚至是决策群体本身的一个突出问题，那么这种多人参与的决策群体的效果势必会弱化。躲避风险或者抑制风险的做法最根本的问题在于是隐藏问题而不是解决问题。群体内的一些自然的现象，比如对权力的争夺、对个人角色的不满意、个性相异、目标的不同都会导致某种形式的矛盾（或者躲避矛盾），以及决策群体做出决策的效果的弱化。

匿名

一种常用的控制潜在的矛盾，并支持其他的 MDM 过程的方法就是匿名。大部分支持群体决策的技术都可以让参与者表达观点，做出分析，最后投票做出选择——并且都是匿名的。当个人的身份和他的意见分开时，这些意见的优点就能得到客观的评价而跟他的地位无关了。不仅如此，如果决策群体的成员能够“安全地”表达个人的意见而不用担心后果或者担心产生矛盾，那么他们就能更平等地参与决策，做出更多的贡献。在很多例子中，匿名可以带来更多更好的信息，也因此带来更好的决策。

性别因素

在工业革命早期的一段时间内，女人们是不能参与管理决策的。那时女人被限定在某些特定的职位上，从根本上被排除在组织的日常管理职能之外。但在数字化革命到来

后,女人的角色发生了根本的变化。现在,妇女已经占到了工作人数的 40%,并且她们中报名攻读 MBA 的人数呈天文数字般的增长。由于妇女越来越多地进入到管理岗位上,对与性别有关的问题的研究也就越来越多。有一点很特别的是,人们越来越关注的一个主题不是男人和女人究竟有怎样的差别,而是人们认为他们有怎样的差别。人们能够认识到的性别之间的差别,看起来对个人决策和群体决策领域有关的问题产生了显著的影响。

许多著作都论述了各种社会环境下性别之间的不同。研究表明男人比女人更乐意冒险。同样研究也发现,大体上在多人参与的决策群体的决策环境下,女人比男人更乐于参与其中,并且证明其与男人完全不同的价值(正如第 2 章所讨论的,价值是所有的决策环境中非常重要的组成部分)。

部分研究显示出与性别相关的价值观差异,比如女性通常对能力、雄心、技能、协作与适应性等问题比较看重。一般来说,女性对能力比较看重,而男性则认为成就是最重要的。与传统看法相反,研究发现女性比男性更加具有工作导向。另一些研究发现,女性通常倾向于根据价值给予经济援助,而男性通常喜欢根据需要给予经济援助。无论如何,有两个重要问题必须考虑:(1) 决策方面的性别差异必须根据实际和推理进行分类(这两种类别对决策结果的影响显然不同);(2) 决策方面的性别差异与其说是多人参与决策群体的环境的缺点,还不如说这种性别差异不会增强实力。这里不是刻意要给这种性别差异分类,只是要表明它们对决策过程和结果的影响的可辨别的程度。有一点很重要,值得注意:许多衡量决策成功性的研究发现,不同性别之间的实际表现差异并不明显,而人们感觉上的或者期望中的差异却很大。这说明,所谓的在决策方面"性别的影响"其实更多的是社会造成的,而不是心理的或生理的必然现象。

谈判与决策

如果有两个以上的人处于决策制定者的角色中,谈判就很有可能出现。不止一个人负责一个给定的决策,这种情况的经常出现说明决策特性便是如此,完全交给一个人制定决策并不合适。决策很有可能包含彼此不协调的多个观点。因此,需要谈判商议。谈判商议的过程中,相互对立的观点和派别对目标和方法等一系列问题进行对质。一个常见的谈判场景就是管理层和工会之间的合约谈判。没有一方希望无果而终,但两方都有义务为各自的委托人争取到最好的待遇。可以把谈判想像成许多参与者之间相互拖拉的战斗。每个人都拽着一条绳索,而这条绳索的另一端和其他的所有绳索相连。每个人都希望把其他人的绳子的绳端拉得离自己近一些。当谈判结束时,中心的位置变了,但大家都同意这个新的位置。

设计一套多人参与的决策群体的支持机制时,很重要的一点是要提供一些必需的谈判行为。在采用多人参与的决策群体的支持技术时需要处理很多问题,比如组织和控制相互不一致的标准及参数选择,创建对所有的相关信息的公平的访问通路,以及对各种各样的组织内沟通结构的支持等。6.3 节和 6.4 节将会介绍各种流行的支持技术及各自的特点。

多人参与决策中的变量

到目前为止，我们已经清楚地知道当两个或两个以上的人一起进行决策时，就会使用一些多人参与的决策环境特有的方法，并建立起很复杂的一系列变量。

Fellers 在 1987 年提出，当很多参与者一起决策时，我们知道一定会发生一些事，但我们不能确切知道是什么。从好的一面说，我们开始对很多的变量以及它们之间的关系有了清晰的认识，并以此逐步了解发生的事情究竟是什么——至少可以通过支持技术的设计和运用的角度进行理解。图 6-4 中是设计和运用多人参与决策群体的支持技术时需要说明的各种变量以及它们之间的关系的分类。

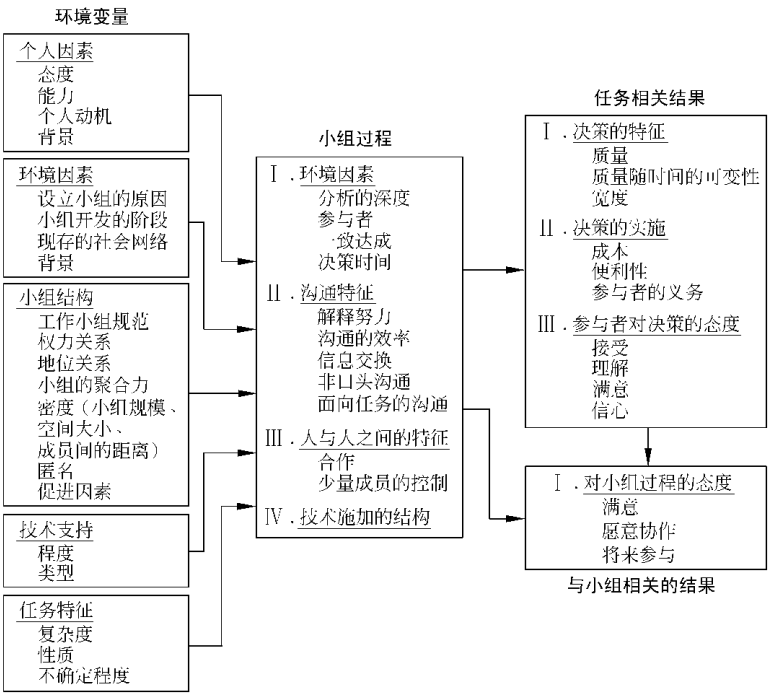


图 6-4　多人参与决策支持技术中的变量

6.3　多人参与决策的支持技术

在我们讨论各种支持多参与者的决策的技术之前，最好暂时停顿一下，考虑一下这种技术的完备水平以及发展速度。

“现在，大多数的商业组织中，当经理们开会时，他们集中在一个房间内，这个房间和 100 年前甚至更早的时候他们的前任开会时所在的房间没什么区别。只有在电灯、空调、也许还有电话上才能看出新的技术。商议过程中得到的信息写在一些便笺、笔记本、财务报告或其他报告上。参加会议的人能听到一些口头简报，再辅以图表和幻灯片的帮助。然而，尽管会议在进行，很多其他人提出的

方案也进行了讨论,决策的制定者们还是不得不依赖他们自己的想法以及别人告诉他们的东西。”(Gray 1983,123)

上面关于集体会议的描述出现在最早的关于群体决策支持系统的论文中。这里有两个重要问题:(1) 对多人参与的决策群体的理解,以及能够获得的技术和知识都已经发生了很大的变化;(2) 上面的描述对世界上的大部分组织还是准确的。多人参与决策的成熟度水平呈指数型增长,但以支持技术的广泛应用的角度衡量,成熟水平还处在“幼年”期。在这一部分,我们将介绍多人参与的决策群体的支持技术过去和现在的水平,并为理解下一代技术的发展打下基础。

基本概念和定义

在最近的 20 年里,出现了许多与多人参与的决策群体的支持技术有关的术语和缩写语。然而又有一个问题就是这些术语同样有了许多定义。很多的研究人员对这些术语应该如何定义以及应该包含哪些组成部分,提出了许多不同的意见。即使在需要达成一致的地方,人们也会给出很多的定义,并且努力把这些定义应用于多人参与的决策群体环境中的各种条件下。本文将采用一种简单的分类方法,这种方法基于一系列同心圆,这些同心圆代表了不同类型的多人参与决策支持,这样相反的意见就不会被忽视或表述不准确。所幸随着时间的推移,研究人员和业内人士对这些各种各样的定义进行集中,并以此来推动多人参与决策的发展,使之作为一个领域更加成熟。

图 6-5 描述了多人参与的决策技术的四个基本层次,适合于 6.1 节中概述的多人参与决策群体的结构分类。绝大多数 (即使不是全部) 现有的多人参与决策的支持技术的应用都可以用这种简单的方法划分。但是,如果不给出每种类别的清晰的定义,没有哪一种分类计划是有效的。因此,本文中,我们对四种基本层次给出如下的定义:

1. **组织决策支持系统** (organizational decision support system,ODSS)。一种基于计算机技术的复杂系统 (包括那些促进沟通的系统),这些系统能够让决策制定者跨越组织内的所有角色以及所有的功能级别,提供了组织内所有人员和单位共同参与的决策环境 (George,1991)。
2. **群体支持系统和群件** (group support system and groupware)。一个基于计算机的技术集合,帮助参与者确定和处理决策过程中的困难、机会和问题 (Huber,Valacich 和 Jessup,1993)。
3. **群体决策支持系统** (group decision support system,GDSS)。一个以计算机为基础的技术集,通常用于支持多人参与的决策行为和过程。
4. **决策支持系统** (decision support system)。在一个或多个决策制定者的控制之下的系统,提供一系列的工具以帮助决策行为,改善决策结果最终的效果。

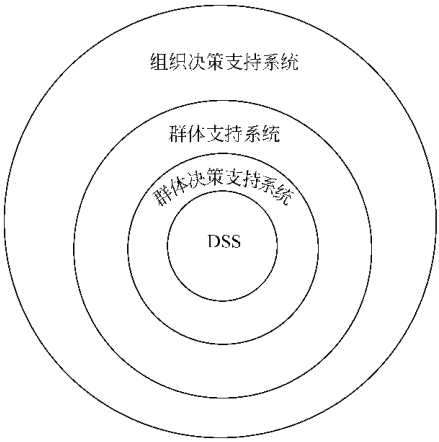


图 6-5　多人参与的决策的支持技术的分类

有了这些定义,我们就可以开始研究和定位决策支持技术的各种类型和结构。

历史情况

多人参与决策群体的行为的支持机制的存在,远远早于现代术语学对这些机制进行术语描述。温斯顿·邱吉尔内阁的战时指挥室就是一个例子。房子的设计非常简朴,包括了一个矩形桌子,战时的首长们就坐在这张桌子旁边进行各种个人的或集体的决策。最主要的支持技术就是墙上的各种地图,这些地图用钉子钉在特定的位置上表示前线和军队部署的情况。而外部的工作人员则负责不断把最新的信息传递进来。

到 20 世纪 60 年代和 70 年代,许多组织把一些新的媒体技术引入到它们的"指挥室"中,比如幻灯放映机、头顶式投影机以及图形描述技术等等,使得决策制定的过程得到了增强。

20 世纪 70 年代末 80 年代初,计算机的引入必然带来多人参与决策支持技术的进步。1971 年,美国紧急事件处理办公室安装了 EMISARI 系统,以支持有关国家紧急时期近 200 人分散到全国各地的决策。70 年代初斯坦佛·比尔和一个英国编程小组在智利实施了一个计算机系统,这个系统的目标是实时监控智利经济。很不幸,在这个系统就快要完成时,萨尔瓦多·阿连德(智利总统)被推翻,这个系统也就再也没有正式投入使用。

现在,全世界有很多的软硬件解决方案可以支持群体决策或者多人参与决策,许多大学开发的解决方案,例如亚利桑那州立大学、克莱蒙特研究生院、加利福尼亚州立大学都提供了内容广泛的技术支持以适应各种不同的群体决策行为。图 6-6 和图 6-7 给出了两个典型的解决方案的物理层实现。

资料来源:"Group Decision Support Systems", Gray/Nunamaker, Decision Support Systems 3/E, edited by Sprague/Watson,© 1982。经 Prentice-Hall, Inc. 和 Upper Saddle River, NJ. 许可引用。

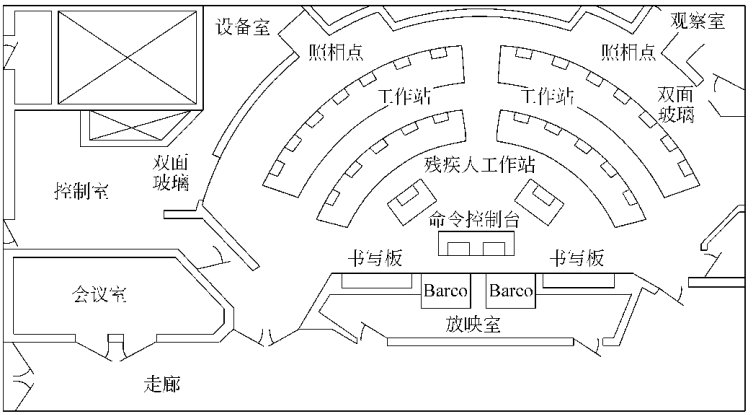


图 6-6　亚利桑那州立大学的大型群体决策支持系统

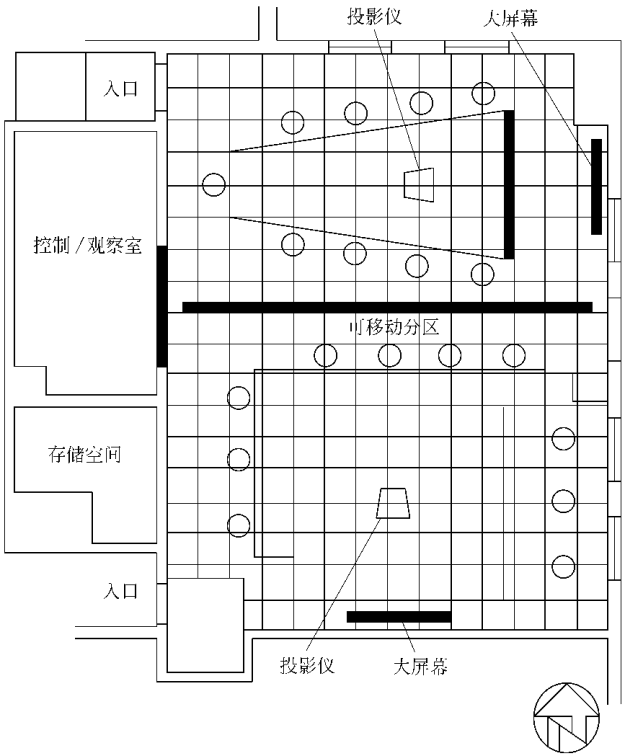


图 6-7　克莱蒙特的系统

资料来源：“Group Decision Support Systems”，Gray/Nunamaker，Decision Support Systems 3/E，edited by Sprague/Watson，© 1982。**经** Prentice-Hall，Inc. **和** Upper Saddle River，NJ. **许可引用。**

多人参与决策群体的支持技术的目标

从前面的论述，我们已经知道，多人决策既有优点也有缺点，会对成功决策产生影响。我们可以把这些优点和缺点看作多人参与决策群体决策时的收益和损失（Numamaker

等，1993）。表 6-6 概要介绍了多人参与决策行为的收益和损失的各种来源。

表 6-6　多人参与决策群体的行为的收益和损失的原因

收益
• 相比于个人，集体的知识更丰富
• 能够解决那些只有相互协作否则无法解决的问题
• 成员间的沟通和交互有利于新的知识和信息的产生
• 通过相互学习，决策参与者的个人表现都能得到提高
• 相对于个人决策结构，这种方法得到的结果价值更大
损失
• 由于多人参与的决策群体的规模，个人的相对谈话时间减少
• 会阻碍观点的产生
• 会导致信息迅速过剩

续表

• 参与者无法记住其他所有人的贡献
• 迫使参与者服从的压力会增大
• 会增加参与者做评价时相互理解的程度，降低了意见的独立性
• 容忍了那些懒惰的不积极参与的人的存在
• 增加了思维的惯性或者“小组思维”的现象，降低了意见的独立性和价值
• 增加了思维的惯性或者“小组思维”的现象，降低了意见的独立性和价值
• 增加了决策问题间的协调行为
• 允许对问题进行部分分析

在此领域内任何支持技术的基本目标，都是使用某些方法使得利润最大、损失最小。这听起来简单，但我们知道多人参与的决策群体决策的复杂性。

有四种基本的方法可以提供这样的决策支持：(1) 过程支持；(2) 过程构造；(3) 任务支持；(4) 任务构造 (Numamaker 等，1993)。根据具体问题，这些方法可以单独也可以综合使用。

过程支持 (process support) 提供了各种机制推动参与决策者之间的交互作用，促进交流、知识的获取与存储。这些机制在现在的群体决策支持系统应用中非常普遍。

过程构造 (process structure) 主要是参与决策者之间各种各样的沟通行为的管理机制，包括沟通的形式、时间、甚至内容。这种方法最基本的好处就是减少多人参与决策群体的行为中由于协调的问题而带来的损失。

任务支持 (task support) 包括一些获取、选择或组织那些与决策任务相关的信息和知识的机制。任务支持方法增强了决策群体对知识的访问，促进了参与者之间的协作，减少了由于对任务本身或者相关的必要知识的分析的失误而造成的损失。

最后，任务构造 (task structure) 机制帮助参与决策者使用各种技术对与任务和问题内容有关的知识的过滤、组织、综合、分析，以及控制知识产生的时机。与任务支持机制类

似,任务构造机制减少了问题分析中的损失,同时增加了多人参与的决策过程的收益。

多人参与决策支持技术的分类

正如我们在前面章节中讨论决策支持系统时所说的,要设计和实施一个多人参与的决策支持系统,首先必须全面地理解系统应用的问题以及支持此问题必需的特点。所以,多人参与决策技术可以按照其内在的特点来划分,或者根据多人参与决策群体的结构的需要划分。

根据特征分类

表 6-7 是 DeSanctis 和 Gallupe 于 1987 年提出的三层分类法,这种分类法基于多人参与决策群体的行为的特征。每一个层次都包含了前一个层次的特点以及本层次特有的特点。此表还显示了这三个层次的特点与相应的决策参与者的需求的关系。

第一层次　这个层次的系统主要用来促进决策参与者之间的沟通。它的主要目标是推动和促进信息的传递,减除多人参与决策群体内的交流障碍。

第二层次　这个层次的系统主要用来减少决策群体的行为的不确定性。过程构造方法和任务构造方法通常一般都归入这一类。从表 6-7 中可以看出,第二层次的系统相比于第一层次的系统,多了一些与分析行为有关的特点。

第三层次　这个层次的系统包含了前两个层次的特点,并且把过程构造的方法扩展到参与者间交互作用的控制。这个层次的系统中多人参与决策群体的行为比前两个层次更严密精确。

表 6-7　DeSanctis 和 Gallupe 的多人参与决策系统的分类

多人参与决策支持技术的层次	决策参与者的需求	系统特征
1. 减少沟通障碍	• 决策参与者之间的信息传递 • 会议进行期间对数据文件的访问 • 阻碍参与者做出贡献的因素的减轻 • 控制免费乘客现象 • 观点和投票的组织和分析 • 偏好的量化 • 议程安排 • 协调时间	• 电子信息传递 • 计算机网络 • 大型的公共显示屏幕 • 匿名地发表意见 • 主动发表看法 • 概要和图标 • 对方案分级排序 • 议程模版 • 流程的连续显示

2. 减少不确定性	• 构造问题,安排问题解决的进程 • 不确定性分析 • 分析资源的分配问题 • 数据分析 • 偏好分析 • 对问题的商讨的组织和引导	• 自动化制定计划的方法(PERT 等等) • 决策表、决策树等等 • 线性规划和最优化方法 • 统计学工具和方法 • 主观概率的方法 • 多人参与决策群体的协调方法
3. 控制决策过程	• 正式规范的决策过程的强制执行 • 决策过程中的选择更加清晰明确 • 对所有的消息进行组织和过滤以保证组织的原则得以实施 • 协商过程的控制规则的发展	• 自动化的决策过程和机制 • 能自动对各种方法和步骤提出建议 • 规则系列的构造和推理机制

资料来源：Management Science，“A Foundation for the Study of Group Decision Support Systems，”DeSantis，G.，Gallupe，R. B. Vol. 33，#5，p. 589～609. © 1987.

根据技术划分

另一种分类方法由 Kraemer 和 King 在 1998 年提出。这种方法主要考虑各种多人参与决策问题中使用的技术,以此来分类。表 6-8 列出了以此方法分出的 6 种类型。

表 6-8 Kraemer 和 King 根据技术的划分方法

多人参与决策支持系统的类型	硬件设备	软件	特点
电子化的会议室	计算机控制的具有大屏幕的多媒体会议室	能够存储和取出事先准备好的展示材料的软件	同时同地的实时交流。需要多媒体技术的支持
远程电话会议	计算机控制的具备在多个地点之间进行视听传播的会议室	控制音频视频数据的数字化传送的软件	同时异地进行同步交流。需要电话会议技术的支持
群体网络	若干办公室通过计算机网络连接起来	既可以在计算机上进行实时会议,又可以不同步地进行音频视频数据的传递	同时或不同时 - 异地的交流。需要某个参与者充当会议协调者或会议主席的角色
信息中心	会议室里有大型屏幕和视频放映设备,而个人计算机上有视频显示终端	具备数据库管理、统计分析、图表生成、文字处理等功能的软件	同时同地的交流。需要专家建模,还需要专门的软件
合作研究室	具有电子显示中心和联网的计算机的会议室	支持信息交流和协作互动的软件	同时或不同时—同地的交流。需要多人参与的决策方法
决策中心	具有大屏幕视频放映设备和联网的计算机的会议室	支持头脑风暴、主题阐释、投票、建模、决策分析、协作交流、数据交换的软件	同时和非同时 - 同地的互动行为。需要多人参与决策方法

资料来源:ACM Computing Surveys 20, No. 2, "Computer-based Systems for Cooperative Work and Group Decision Making" by Kraemer, K. F. and King. J. L. ,© 1988.

在这种分类方法中,各种技术的复杂性是各种类型的系统的分界线。其中最简单的一种系统就是"电子会议室",这种系统中最基本的就是视听和多媒体技术。而最高层次的系统是"决策室"系统 (decision room),在这种系统中计算机技术可以支持 MDM 几乎所有的行为和需要。这些系统提供了一些工具支持头脑风暴、问题分析、问题解释、舆论评价与投票决策等行为。表 6-9 列示了现代的"决策室"系统中各种各样的决策支持机制。

表 6-9 "决策中心"系统的支持功能

- 电子化形式的"头脑风暴"
- 问题阐释
- 问题分析
- 投票选择,并显示出偏好
- 形成政策
- 风险分析
- 对观点和思想进行组织
- 其他方案的评价
- 调查问卷的制定,调查的管理
- 读取多种格式的文件
- 供决策参与者参考和查阅
- 对产生的决策和计划进行分析
- 多人参与决策群体的会议的管理和控制

资料来源:Vogel 等 (1989)。

协作支持技术

现代组织内的协作趋势正在逐步加快。这种趋势是由于多种因素形成的，其中包括网络。网络大大促进了互联互通。另一个因素是日益严重的全球竞争。激烈竞争的环境中企业面临的经济压力是促使其转向协作的第三个因素。最后一个方面，企业都在努力吸取新技术所带来的效率和效果的提高，它们发现一些主流的基础设施不再适合了。因此，今后的企业将会推动群件技术的使用，彻底改造企业自身。

群件

群件 (groupware) 指的是一种特殊的多人参与决策群体的支持技术，主要关于人们之间的协作方法问题。从某种意义上说，群件技术就是人本身。它是一种工具，如果能够正确地安装和使用，对人们相互沟通的方式产生了影响，那么它一定能够改善人们工作的方式和做出的决策的效果。

群件的概念其实只是用计算机对传统工具的一种简单扩充。任何组织要成功，一个关键因素是组织存储。商务管理过程中获得的知识必须用某种方式收集存储起来，以便现在或将来的决策需要时易于使用。这样的工具随处可见，比如：人、政策和程序指南、公告牌、时事通信、文件柜、活动箱子、计划牌，以及计算机。

现在群件产品之间的竞争非常激烈，只有少数几种产品占据了较大的市场。现在市场的领先者有：

- Lotus Notes and Domino——Lotus 公司
- Microsoft Exchange——微软 (Microsoft) 公司
- GroupWise——Novell 公司
- Oracle Office——Oracle 公司
- Team Office——ILC 公司
- Collabra——Netscape 公司

图 6-8 显示了 Lotus Notes 软件中的典型界面。图 6-9 展示了 Notes 软件的通信系统的各个部件及功能。

市场上主要的群件产品的特征及组件都在不断地变化。此外，某些厂商还会和其他的一个或几个竞争对手联合起来打压其余的竞争对手。例如，1997 年末，微软和 Lotus 公司达成协议，在以后所有版本的 Notes 产品中，都捆绑安装微软的 Internet Explorer 4.0。这样做，微软在群件产品的竞争中可能会有所损失，但巩固了在网络浏览器市场中的领导地位 (在这个领域内，微软和 Netscape 竞争激烈)。

群件产品不断在发展，以适应决策制定和商务管理越来越快的速度。这就需要组织内各个层次的人员对组织的存储做及时的、并发的、完全的访问。群件产品的许多功能和类别都使得可以用各种方法很容易地访问组织的存储数据。Ellis，Gibbs 和 Rein 在 1991 年提出了一个根据群件产品对组织的直接支持的不同类型，而将群件产品划分为 6 个部分的方案。表 6-10 所示的就是这种分类方法。

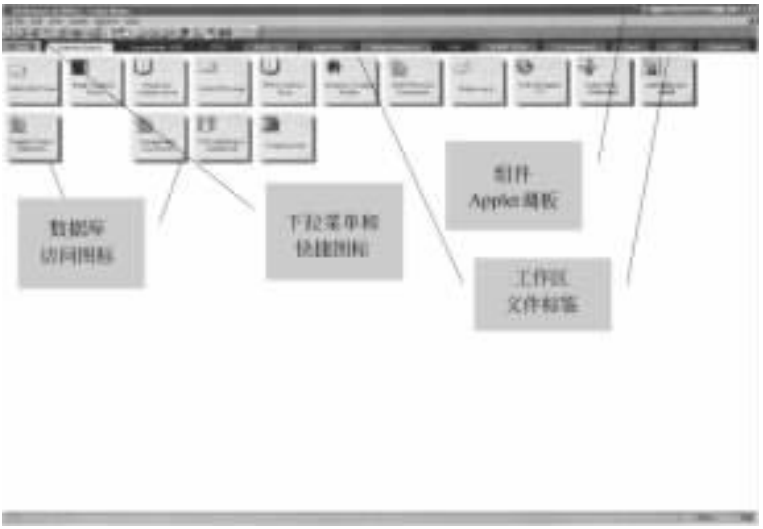


图 6-8　Lotus Notes 系统的典型界面

图 6-9　典型的 Lotus Notes 系统的消息数据库的结构和界面

表 6-10　群件产品的分类

• 通信系统	• 群体决策支持系统
• 会议系统	• 协调系统
• 协同工作系统	• 智能主体系统

通信系统　这种分类体系中的第一个层次就是主要是用来促进信息流动的技术。这个类别中最常见的就是电子邮件。一个以文字为主的信件再加附件 (文件,可以包括图表) 可以在很短的时间内写成,并且发给任何地方任意数量的接收者。与传统的纸质信件 (现在看起来像是"蜗牛信件") 不同,收信者的地理位置不再是影响收信的重要因素。电子邮件可以从世界的任何地方的计算机上发出,只要这台计算机连在因特网上。由于电子邮件的非正式性、快捷性、使用性、方便性,它成为了决策制定过程中提出问题、收集意见的理想工具。电子邮件可以让组织内的成员从很广泛的网络中寻找特定的内容。

这个层次的群件系统只支持异步通信。也就是说,信息的发出和接收没有什么联系,不是同步的。这种异步的特点限制了它在某些决策问题和环境中的应用,但是它对于早期的决策支持非常有用,可以帮助所有参与决策的人搞清决策的需要,收集决策所必要的信息和知识。

会议系统　第二类的群件产品通过将传统的面对面会议电子化,从而克服了通信系统中的异步局限性。这种系统最大的好处是,在进行实时会议时不需要再坐在一起。使用这种系统,人们可以在任何方便的地方与其他人进行实时交流。这种系统使用的媒体可以是音频、视频或者两者兼有。而且,这种会议可以在一个专门的会议室中进行,也可以在个人办公室中进行。而当计算机上的桌面会议技术出现后,参加会议的人可以看到其他人,也可以听到别人的声音,同时他们还能够用计算机展示图表,运用各种分析工具,相互传递数据。

协同工作系统　这个层次的群件系统可以让所有决策参与者以同步或异步的方式协同工作,共同创作或修订一些公共文档。像文档章节的修改、注释、排版以及图表的设计都可以由一个小组管理完成,而其他的所有参与者都可以看到结果。另外,系统会把每个人的工作都记录下来以便以后回顾,而且将来的工作的时限要求也会适当地传达到个人,所以这种系统能够改善决策过程的管理。

群体决策支持系统　这种层次的群件技术的最关键的特征就是,它是以帮助多人参与的决策为目标的。正如前面所讨论的,一个群体决策支持系统能够帮助思考、注释、分析以及最终达成一致意见。

协调系统　协调系统的主要功能就是协调综合个人的工作以利于集体目标的达成。比如,系统会告诉某个参与者其他人的工作完成情况,并且告诉他这项工作的完成对下一阶段工作的开始至关重要。同样,它还可以告诉某个人他的工作是否及时完成。这类系统有个常用的功能称为"业务流程管理",其中包括文件的发送和批准、多层次的数据收集以及信息的传递。

智能主体系统　这是群件技术的最后一类。它引入了一些人工智能的技术来处理某些特定的任务。这套系统的功能,简单的可以管理电子邮件信息,复杂的可以成为个人助手,帮助安排会议时间,转发消息,或者执行一些和决策有关的后台任务。这种类型的决策支持产品还处在初级阶段,但毫无疑问,它将成为非常重要的决策支持工具。我们将在第 15 章更细致地讨论这种技术。

推动群件产品使用的动力

行业分析家认为群件产品在 2000 年前将成为一个具有几十亿美元价值的市场。这种看法建立在一系列促使企业转向群件技术的压力或动机的基础之上。表 6-11 列出了导致群件产品得以大量使用的许多因素。

表 6-11 有利于群件技术的采用的因素

- 成本控制的加强
- 生产力的提高
- 客户服务的增强
- 对全面质量管理的支持
- 日常业务流程的自动化程度的提高
- 供应链延伸的需要
- 地理上不相邻的机构之间的联系的需要
- 提高竞争优势
- 更好地进行全球协调的需要
- 创造个性化服务
- 专家经验和知识的支持
- 基层组织内网络的广泛应用
- 对某些特殊团队的更多利用

资料来源：Coleman 和 Khanna (1995)。

从此表我们发现，在这么多的因素中，没有一种和决策行为直接相关，大部分都是和组织行为有关的。这说明群件产品更多地成为了基础设施，而不仅仅是一个决策支持系统。

6.4 多人参与决策群体行为的管理

一个群体内自然有领导者和被领导者。这样既考虑到了那些希望发挥自己的领导能力带领组织向前发展的人，又考虑到了那些希望接受别人的领导等待接收命令的人。问题是这种协调人际关系的方法常常未必合适。当很多人一起决策时，决策问题的特性决定了这个群体的结构，也就决定了应采用何种协调方式。表 6-12 列出了若干常用的多人参与决策群体内的协调方法。

表 6-12 常见的多人参与决策群体的协调方法

- 记名 (不匿名) 方法
- Delphi 方法
- 仲裁方法
- 基于问题的信息系统方法
- Nemawashi 方法

表中所列的方法绝对是不完全的，这些方法还有一些变种。而这些变形的方法需要不断地测试，改进，最终实施。只有经过多次试验发现是有效的方法，再根据具体问题进行修改，才是最合适的。

记名方法

由 Van de Ven 和 Delbecq (1971) 提出。记名方法 (nominal group technique) 最适合于需要达成一致意见的环境，比如群体式结构或委员会式结构。这种方法需要每个人按照下面的程序履行其职责：

1. 每个参与者都写下自己的观点,应该如何决策如何做出选择。
2. 每个人都把自己的方法写在列表中。这些观点将会汇总在一个表中,这样所有的人都能看到。这一步中不对各种方法进行讨论。
3. 当所有的观点都列出来后,所有的人可以相互提问,以比较各种观点。
4. 按照事先确定的等级和顺序,每个人对所有的意见进行投票,然后统计投票结果,集体的方案就产生了。

这种方法可以按照上表中的步骤和顺序手工地实现,也可以很方便地用计算机实现。

Delphi 方法

由 Lindstone 和 Turoff 在 1975 年提出。这种技术和前面的记名技术比较相似,最基本的区别在于使用这种技术时,参与决策者不需要真的在一起。使用这种方法的步骤如下:

1. 根据在决策问题方面的专门知识和经验把决策群体的成员集合起来。
2. 对所有的成员进行调查,收集他们对决策问题的看法。
3. 对调查结果进行整理和分析。
4. 根据第一次调查的结果进行第二次调查,要求所有人对第一次的结果进行思考,并在思考后填写第二次的调查问卷。如果某个人的结果还是和大多数人的结果不一样,他必然要解释原因何在。其他人也会看到这些解释。
5. 重复步骤 2 到 5,直到达成一致意见。如果在规定时间内无法达成一致意见,支持率最高的意见就将成为最终方案。

我们将在第 15 章讨论多人参与决策群体内创造型思维的培养时,再次考察记名群体方案和 Delphi 方案。

仲裁方法

在决策群体的成员间有矛盾或意见无法一致时,仲裁 (arbitration) 是非常有效的方法。通常进行一系列的商谈,试图找到相同的意见。所有人都明白如果不能在限定时间内得到结果,就会进行仲裁。如果他们仍然坚持自己的意见不妥协,那么第三方就会进入充当仲裁者,从所有的候选方案中挑出一个他们认为最合理的方案。这种选择并不只是在发生争论的方案之间,而是在所有的可能方案之间。仲裁者可以使用任何的方法和标准挑出一种最合适的方案。这种方法的一个改进是,每一个参与者都可以给仲裁者提供一个最终方案,仲裁者可以从这些提交的最终方案中做出选择。

基于问题的信息系统方法

基于问题的信息系统方法 (issue-based information system,IBIS) 是一种指定了决策群体成员的讨论和争论的方式的结构化的方法。这种方法通常用图形 (图中包括点以及点与点之间的连线) 来描述问题、形势和争论。创建一个基于问题的信息系统,首先要选择一个起始问题作为根节点,然后其他的点代表各种候选的解决问题的方案,都与根节点相连。每一种候选方案的被认可程度将会成为最终决策的基础。

这种方法通过超文本链接的技术可以很容易地实现电子化。每种候选方案都可以通

过超文本链接到相应的节点上,这样一点击节点,就可以看到相应的方案的内容和解释等等。通过这种方法,所有的方案都能实现电子化组织和存储。每种方法的合理性可以通过事先确定排列方法很容易地列示出来,以便选择。

Nemawashi 方法

在第 2 章中我们已经介绍过,Nemawashi 方法是在日本很常用的管理多人参与决策过程的方法。使用这种方法时,应该遵循以下步骤:

1. 一旦决策问题的内容确定后,就要从决策群体中选出一个或多个成员作为协调人。他们从剩下的成员挑出一部分参与到决策过程中,并开始询问他们对决策问题的看法以及他们的结论。挑出的人的数量可以很多,也可以很少。
2. 协调人从这些方案中初步挑选出一批,然后咨询专家的意见来制定一个评价标准,并根据此标准对选出的方案和其他的方案进行评价。
3. 协调人根据第 2 步的结果定下一个初步方案。
4. 将第 3 步中的初步方案写成非正式的文件,在所有的决策群体成员内传阅。通过沟通、商议、说服最终达成一致意见。当然,为了达成一致,协调人可能需要对文件进行修改,甚至回到步骤 3,重新挑选其他方案。
5. 一旦达成一致意见或基本一致,协调人就要准备将最终方案及其细节内容写成一份正式文件,并给所有人传阅。这份文件首先要给组织内最基层的人员看,他们看完后才能给更高层职员看。每个人看过文件后都要在文件上签字,表示已经看过并支持文件中的方案。如果某人有非常充足的反对理由,他可以反向签字(表示反对)。如果有足够多的人支持,这个方案就可以通过。这个"足够的"到底是多少,不同的组织间甚至同一个组织内的不同项目之间都是不同的。

在某些多人参与决策群体的结构中(比如群体式和分层次的小组式),因为在不同的层次上都会有权威的存在,这种方法会不太合适。尽管如此,如果条件合适,这种方法能够得到一个很好的决策结果,这个结果能够得到不同阶层的广泛支持。

6.5 虚拟工作环境

作为决策群体行为及支持技术的讨论的结束,我们还要再讨论一下前面介绍的这些技术使企业发生了哪些变化。最普遍的变化就是"虚拟办公室(virtual office)"。请看下面一段文字:

> "位于匹兹堡的美国铝业公司大楼具有全铝制的外表和像电视屏幕那么光滑的窗子,充分地展示了传统的摩天大楼的魅力。1950 年当这栋楼初建成时,每天有 2000 名员工进入这栋 31 层高楼工作,他们每人都一个自己的 12 英尺 × 15 英尺的办公室。
>
> 但是现在你去看美国铝业公司的首席执行官奥尼尔的办公室,你会发现他没有真正意义上的办公室。经理们的办公地点再也没有固定的墙壁或门。所有的高级经理都在一个开放的环境中工作,周围全是电视机、传真机、报纸以及用

于临时会议的桌子。真正属于奥尼尔的房间是厨房，在那儿他和他的同事们一起煮东西吃，围在一起讨论工作。“在厨房，你就像在家里一样合桌而坐。”他高兴地说……保持私人工作空间和个人秘密已经被追求生产率和灵活性所取代。在任何时候任何地点都能工作已经成为了新的标准。对速度和效率的追求使得员工之间团队工作和信息共享变得非常必要。”（Hamilton，Baker 和 Vlasic，1996）

自从协作技术出现以后，全世界许多组织都在将自己的实际资产抵价来购买新的技术。为了具备能够协调由世界各地大量参与者组成的决策团体的行为的能力，它们都在快速转向前面的例子中的模式。简单地说，人们越来越强调工作是你所做的事情，而不是你每天去的那个地方。我们需要理解这种变化对个人决策和群体决策过程的特点带来的影响。短期来看，会显著增加员工在传统的工作环境中的工作效率。宝洁公司在俄亥俄州辛辛那提新建的面积 130 平方英尺、价值 280 万美元的产品开发中心，就完全设计成开放协作式的环境。相应地带来建筑设计上的变化，比如全面采用电动扶梯取代升降梯以便于交谈。在走廊上增设沙发，便于工作人员简短的休息聊天。传统的办公室完全被新的开放式的集中办公环境所取代，中央有大型的白色书写板，这样可以让过去的随意书写转变为电子邮件。初步结果显示，大楼的结构设计得好，能够使生产力水平提高 20% ~30%，因为数据共享更加快捷，可以在更短的时间做出更有效的决策。

当然，这种向虚拟工作环境的转变可能还有很多的未知特点，也可能会有一些缺点。最重要的一点，这种虚拟工作环境所带来的更多的是文化上和社会学意义上的变化，而不仅仅是技术上的。而且，技术发展日新月异，今天看来很有意义的技术（方法），很快就会被新的更有效的方法所取代。尽管如此，企业追求高楼大厦的时代已经过去，新的多人协作技术和多人参与决策支持技术把我们带入了新的时代。

6.6 小　结

本章揭示了群体决策支持的概念。现代工作环境中面临的问题越来越多地涉及到很多人，而不再是一个人。多人参与决策的方法尽管有一些好处，也会给决策过程带来一些缺点和问题。

使用各种各样的多人参与决策支持技术，管理者可以更好地管理多人参与决策群体的行为，还能更加清楚地知道有哪些潜在的问题，以便获得最大的收益。

复习题

1. 给出“多人参与决策群体”的定义。
2. 列出并简要描述群体内沟通网络的四种基本类型。
3. 多人参与决策群体的规模有哪些影响？

4. 什么是“小组思维”？举一个例子。
5. 什么是谈判后的决策？简要描述一下。
6. 说出一个典型的“决策室”系统的支持功能。
7. 给出群件的定义。适用于什么地方？
8. 列出并简要描述群件产品的各种类别。
9. 列出并简要描述常见的多人参与决策群体内的协调方法。
10. 传统的办公室和虚拟办公室之间有什么区别？

讨论题

1. 分析市场上处于领导地位的几种群件系统。描述这些产品的主要组成部分，并比较其功能。
2. 列出几种控制多人参与决策群体内的潜在矛盾的方法。
3. 比较群体式、团队式和委员会式三种决策结构。它们之间有何异同？每种机构各自适合于什么环境？
4. 调查一些使用群件系统的组织中的人。搞清楚使用群件系统有哪些开销，又有什么收益。
5. 哪一种群体内协调方法你最常用？为什么？

第 7 章

经理信息系统

学习目标

- 理解经理信息系统的定义
- 了解设计执行信息系统的两个条件
- 理解经理信息系统的能力
- 了解经理信息系统的发展历史
- 熟悉决策行为及其基本类别
- 了解高层决策所需要的信息的种类
- 熟悉确定决策信息的方法
- 掌握经理信息系统的软件和硬件组成
- 熟悉当前的执行信息系统技术种类
- 理解经理信息系统的开发框架
- 了解执行信息系统的一些缺点和局限
- 了解决策制定时的可能变化情况
- 了解现在和将来的执行信息系统的潜在特点

小案例

AlliedSignal

现在，AlliedSignal 公司的航空部门的经理和其他 500 多名 AlliedSignal 公司的雇员，可以在很短的时间内得到他们想要的商业信息。但在两年以前，他们却不得不为了拿到所需要的商业报告而等待几天的时间，有时还有可能根本得不到这些东西。

AlliedSignal 是一家销售额达到 120 亿美元的全球性制造商，它的产品包括飞机和汽车零部件以及光纤、化学品、塑胶、电路板等专业原材料。经由航空部经理 Dan Burnham 建议，AlliedSignal 公司最近采用了一套覆盖全公司的执行信息系统（EIS），这套系统彻底改变了员工使用信息的方式。

Dan Burnham 希望得到诸如列交易量前 10 位的客户名单、净销售额和净收益、国外销售量与国内销售量的比较、军用品与商用品的销售量的比较等信息。这些信息在员工个人的层级上是可以获得的，但是在部门的层级上却从未被收集过。Dan Burnham 说道："为了计算质量成本，我们不得不找来每一个单位，询问他们哪里出现了返工成本，再把它同其他资料结合起来。"

对于 Burnham 来说，获取信息最大的障碍在于信息的分发。很多在偏远地区的分区主管需要得到那些能很快整理成简单系统的信息。

Allied 公司采用 Comshare 公司的"Commander"OLAP 系统来针对 Burnham 需要的信息开发一个系统原型。短短 30 天之内，该套系统已经准备好向 Allied 的财务主管做演示。3 个月之后，Allied 公司的开发人员在 Dan Burnham 的计算机内安装了第一套 EIS 系统。之后，经过 1 个月的培训和针对用户反映做出的调整，他们将这套系统扩展到 15 个不同地区的 150 多人的电脑中。

初始的用户中很多是负责向经理的 EIS 提供信息的主管人员。但在每一个地区至少有一人被指定为技术支持人员。初始安装之后，重点转移到调整系统，使其为每一个人服务，而非专为经理。起初，分区主管只把这个系统看成是向 Burnham 传送信息的工具。但开发人员很清楚，如果 EIS 不能帮助每一个人工作的话，它将不会有发展前途。

原型是关键所在

传统的系统开发方法需要预先定义好的系统规范，Allied 公司的开发团队在使用这种方法开发系统时是非常谨慎的。他们认为，最有效的使终端用户获得价值的方法，应该是快速地构建系统并将其交付用户使用，然后寻求用户反馈并根据需求做出调整。通过这种方法，系统将会建立在实践反馈而非理论规范的基础之上。他们之所以能够部分地采用这种较灵活的开发方法，是因为"Commander"系统能够适应如此快速的应用层的升级。"Commander"的可靠性对于快速原型开发程序同样具有关键意义。使

用这种开发方法，EIS的使用率大幅度的提高。18个月内，用户由航空部的150人扩

展到覆盖 AlliedSignal 的三个部门的 500 名用户。

初期在 EIS 内部开发的应用软件之中，有一套使用 Comshare 公司基于 Windows 的“Prism”软件的飞机生产预报软件。Burnham 解释道：“我们追踪未来 10 年内将要生产出来的每一驾飞机的信息，我们和每一家制造商、引擎生产厂以及销售商都有联系，有很多种方法处理我们的信息。”

自从系统的首次展示后，用户们关于应用的新点子已经源源不绝地提出，开发团队的每一个人都在面对如此众多的建议。开发团队能如此之灵活，是因为他们使用的工具也具有相似的灵活性。“Commander”给团队提供了很强的应变能力，从而使得它面对用户需求时显得很灵活。

当 EIS 的初始功能全部实现时，超过 750 名 AlliedSignal 的雇员将会使用这套系统。面对这种形势，Allied 的开发团队依然认为 EIS 仍处于幼稚期。人们刚刚开始对这套系统给予关注。

现代组织的 CEO 与几十年之前他们的前辈距离并不遥远。从主管的角度看信息是不可或缺的资源，而组织中大部分资源的分配原则也是为了收集和利用信息。CEO 所需要的信息的特征同样也没有发生太大的变化。高层主管所需要的信息面，比起中层和低层管理者所需要的信息来要宽广得多。与以前相比不同的是信息产生的速度以及 CEO 获得信息的速度。典型的决策支持系统设计的目的，是为了在公司的中低管理层上为解决专门的问题以及决策提供有效的支持，它不能为高层管理者提供所需的多样化的信息和决策支持。为了满足 CEO 的独特的信息需求，一类专门的决策支持技术出现了。本章将重点介绍高层管理者需要的信息，以及为了满足这种需求所使用的经理信息系统（executive information system，EIS）的结构。

7.1　什么是经理信息系统

定义及特征

一般意义上说，EIS 是 DSS 的一种特例。它是专门设计用来帮助分析信息对于整个组织的运作所具有的关键意义，并对高层管理者的战略决策提供支持工具。更具体地说，一套 EIS 系统不仅能帮助 CEO 精确而又快捷地了解自己组织的运营状况，甚至连竞争对手、顾客和供货商的活动也能尽收眼底。EIS 系统可以对公司内部和外部的重大事件及趋势进行连续的跟踪监视，然后根据现阶段的需求将处理过后的信息提供给高层决策者。EIS 还可依据决策者的喜好提供对于所涉及问题的全面概述或是细节。比如，CEO 可以利用 EIS 对于销售情况按产品类别、地区、子分区、月份、本地市场或是其他方法进行快速浏览；同时，CEO 还可用同样的方式对竞争对手的行动进行监视。这种高度概括的浏览可以使管理者对公司以及市场的概况有一个简明快速的了解。一旦在这种简要印象之中出现了不符常规的反常现象，决策者可以对数据进行更深层次的挖掘，以获取更详尽的资料。如果需要的话，这种对数据的分解操作可以延伸到单个交易的层级上，直到 CEO 找

到可以解释这种反常现象的原因并决定采取相应的行动为止。EIS 系统的设计使得它可以获取各种各样的信息资源,并对这些资源进行联系和概括。它还向用户提供检验、分析所获取信息的工具,以便用户能快捷、明智地做出决策。不管 EIS 系统被安装在怎样的环境之下,它们都具有相同的一般特性,表 7-1 列示出了 EIS 技术的一些通用特性。

表 7-1 EIS 系统的一般特性

- 直接被高层决策者使用
- 为个体决策者量身定做
- 操作简便,使用前无需培训
- 专门用于支持高层管理决策
- 可以通过图、表、文字等形式输出信息
- 从包含内部及外部资源的很广泛的范围内获取信息
- 提供选择、析取、分离、追踪信息的工具
- 可以提供各种各样的报告,如状态报告、异常情况报告、趋势分析、数据挖掘研究报告、特别询问报告等

基于 EIS 系统的普遍特性以及各种文献中对于 EIS 的定义,我们对 EIS 采用以下形式的定义:EIS 系统是一套计算机系统,它通过更便利地获取企业内部和外部与组织目标相关的信息,为高层决策者提供决策支持。按照如上的描述,我们通常所说的 EIS 和"电子支持系统"(electronic support system,ESS)是可以通用的,但是通常我们提及的 ESS 系统暗含着比 EIS 更多的功能。随着分布式群件技术的出现以及对现有信息收集、决策制定技术的改进,这种区别正在变得越来越模糊。在这里,我们将会使用术语 EIS 代表所有的经理支持系统。

在我们所讨论的深度上,将各种 EIS 系统结构进行分类并不是十分重要的(虽然我们最终确实要这么做)。重要的是,应该认识到 EIS 系统在规模和设计目标上是不尽相同的,就像 DSS 系统一样,EIS 必须根据特定的决策环境和特定的信息需求来设计和实施。因为这些设计要求是非常容易变动的,组织中的 EIS 必须是一个能自我完善的工具,因而需要企业对其发展完善给予支持。

一个典型的 EIS 工作流程

为了更好地理解 EIS 的特点,让我们看看一个典型的 EIS 工作流程。流程开始是一个关于公司财务和商务运营状况的报告。这份报告将会包含关于地区销售利润和成本的几张图表,其中还会有一部分给出一些关键指标,从传统的财务比率(如资产负债比)到更硬性的指标(如顾客等待电话援助的平均时间)。报告的主体部分将会使用箭头和不同的颜色来展示出那些上升的、保持稳定的和下降的指标。另外还可能用一些图来展示那些表现超出了预想的指标。只需简单地看上几眼,管理者就可以对整个组织的现状有一个整体的评价。

回顾了总体信息之后,管理者可能会注意到一些看起来很糟糕的指标。我们假设流动比率正在以不正常的速度下降。管理者将会使用 EIS 的"钻取(drill down)"功能来研

究那些用来计算流动比率的数据,这项功能可以在屏幕上分别显示资产项和负债项的各个类目。如果这个层次上的分解还不能给出答案的话,“钻取”操作将会继续进行。对财产项的继续“钻取”将会详细列出现金、存货、应收账款以及其他的财产项目的具体金额。这种过程将一直持续下去,直到分解的详细程度足以说明比率变化的根本原因为止。

“钻取”功能是 EIS 的几项重要特征之一。这种分解的过程对于管理者来说是可以选择和控制的,并且在不同的环境中也是有区别的。管理者只需敲击键盘或是点击鼠标来选择他们所认为合适的详尽程度,就能发现在屏幕上显示的信息的背后所蕴含的故事。如果一个管理者对于公司总体的销售情况以及各个地区的销售情况都有所了解的话,他就会发现某类产品的颓势可能是由于特殊的地域因素造成的。对于这个特殊的地区继续“钻取”的结果可能会暴露出促销手段的匮乏。对于那些销售状况良好的地区“钻取”的结果,告诉我们如果销售商对商品做了足够的宣传,那么销售量会比预期的还要好。

对数据的“钻取”功能使得管理者可以使用一种“由顶到底”的分析方式来寻找问题的解决办法。EIS 可以简明地指出潜在的问题,而“钻取”功能使得对深层问题的研究成为可能。所有这一切带给我们的将是更好的决策、更优的解决方案和更成功的运营。

对 EIS 的误解

对于 EIS 系统的误解是一个很严重的问题。EIS 对于组织来说既不是包治百病的灵丹妙药,也不是其他信息技术或计算机系统的替代品。事务处理系统 (TPS)、管理信息系统 (MIS)、个人决策支持系统 (DSS) 和 MDM 支持系统,这些系统在为现代组织的各个层级提供相关信息方面仍是至关重要的。EIS 与它们的不同之处,在于它的内部对于信息的需求以及将自身与外部信息资源相结合,为更宏观的目标服务。

由上面的论述可以得出,EIS 系统不应使管理者的角色变成一个计算机操作性的岗位。好的 EIS 设计对于那些非技术人员的用户来说应该具有直观、灵活、易操作的特点。因此,EIS 系统应该被高层管理者看成是一个随时随地都值得信赖的朋友,一个随时随地都可召唤的助手。

7.2　经理信息系统的发展历史

“经理信息系统”的概念是由美国麻省理工学院 (MIT) 在 20 世纪 70 年代后期提出的。最初的 EIS 系统是由几家私有企业开发的,它们认为使用 EIS 将会为自身带来巨大的竞争优势,当然作为先行者它们所面临的风险也很大。历史上,Rockart 和 Treacy 于 1982 年发表在 Harvard Business Review 上的一篇名为“The CEO Goes On-Line”的文章标志着 EIS 的诞生。文章中十分生动地将决策者的计算机描述成抵御专业计算机技术疯狂入侵的最后一座堡垒。当这种组织管理工具广泛传播的时候,有很多人持怀疑的态度,但很快怀疑论者就悄无声息了。到了 80 年代中期,几家 EIS 提供商 (如 Pilot Software 和 Comshare) 已经在大公司打开了销路,它们提供相对简单的 EIS 运行环境,其中包括简明的屏幕显示、灵活的接口设计、预设的电子新闻资源获取机制、加速数据输入机制和很多种实用的分析工具。MIT 的信息系统研究中心在 1985 年所做的一项调查表明,有将近 1/

3 的美国公司已经安装了或正在安装 EIS 系统。

经过 20 世纪 80 年代和 90 年代初期的成长,新的供货商和制造商的出现扩展了可用信息资源和分析技巧的范围,使得 EIS 可以在组织中经理之外的层级上得到应用。现代 EIS 中包含多种类型的数据,其中包括关键商业活动、研发成果、客户相关信息、财务活动以及支持必需的环境审查活动的外部数据。现今的 EIS 系统将面对比过去更广的用户群,而其应用能力早已超出一个典型的公司组织的界限。采用 EIS 系统并不是没有风险的,但开发和使用 EIS 获得成功的例子使我们相信,与收益相比冒风险是值得的。

要想完全理解和领会 EIS 的价值和能力,我们必须首先理解和领会用户的信息需求。7.3 节将会重点讨论 CEO 们的信息需求。

7.3 为什么高层决策者的差异如此之大

要回答这个问题,我们必须先定义什么是高层决策者以及他们都要进行哪些活动。经理 (executive) 这个词没有普遍的定义,在不同的组织之间它的含义也是有区别的。就像其他缺乏集中定义的概念一样,“经理”的概念应该由其普遍特性来定义。表 7-2 列出了一些与决策机构相关的特性。

表 7-2 经理的共有特点

- 他们管理着整个组织或是独立自主的子单位
- 他们从企业级的角度出发考虑问题
- 在组织中他们的控制范围最广
- 他们在战略级的水平上为将来考虑而非只为每一天打算
- 他们对制定决策负有责任
- 在某种意义上他们代表着企业与外部环境相互作用
- 他们的行为可以在财务、人事、商业等方面产生重大影响
- 他们必须广泛地关注公司内部及外部的重要事件

资料来源:Watson,Houdeshel 和 Rainer (1997)。

经理的信息需求

要想完全领会每一个决策支持系统的用户的信息需求,必须首先搞清楚工作的特性,以及与决策支持过程相关联的各项工作是如何进行的。在 EIS 系统的设计中,对每个目标用户的特点的了解是取得成功的关键。

从管理学第一天的课程开始,我们就知道了每一个管理者都有 5 个基本职能:计划、组织、人员调配、指导和控制。你也许会记得,虽然所有的管理者都会在这 5 个职能下处理一些事情,但是他们在每个方面所花的时间却并不是一样多的。所以区分组织中管理层内的各个不同管理角色的方法之一就是,通过他们使用各个功能的频率加以分别。高层决策工作的特点决定了与其他层级的管理者相比,决策者通常要花较多的时间在计划和控制方面。决策层所特有的工作包括运营控制、战略计划、谈判以及对混乱局势的控制。Rockart (1979),Jones 和 McLeod (1986) 分别对高层决策者专门职能的识别以及这种

行为出现的频率进行了研究。在研究期间（一段连续的时间），决策者们被要求在工作时小心行事。另外，他们还必须说明每次行动所使用的信息资源以及该次行动的重要性。研究结果表明决策者的行为可被划分为 5 个基本类型。图 7-1 给出了这 5 个类型的行为和它们出现的相对频率。

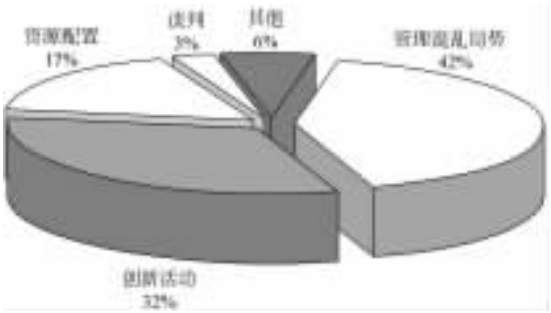


图 7-1　经理活动的频率

对混乱局势的管理

这里使用“混乱（disturbance）”这个词，可能会掩盖这个领域内事件的真实本质。当出乎预料的事件发生的时候，特别是那些可能对组织的财务体制产生重大影响的事件发生的时候，决策者必须保证做出迅速反应来调配资源。在混乱时期，解决目前棘手问题的决策是最重要的，相比起来决策者做出的与这一问题没什么直接关系的决策是次要的。此外，在混乱管理的初期可能需要对问题进行连续不断的关注，这种关注可能会持续几个星期甚至是几个月，直到混乱状况得到充分的控制。正如在图中看到的一样，决策者的大部分时间都花在混乱管理上了。

发生在 1996 年 5 月 11 日 ValuJet 公司 592 航班坠毁之后的一系列事件，是混乱管理行为的一个鲜明的例子。ValuJet 公司的总裁 Lewis Jordan 迅速被推到媒体的关注焦点之下，他将要面对的是 ValuJet 公司的记者招待会、公司内部和外部对事件进行的调查、公共关系处理以及为无法回避的司法程序的介入进行准备。联邦航空局（FAA）对公司的调查导致公司的日程表被打乱，Jordan 不得不决定将原定日程分割成两部分，以便公司的飞机接受更严格的检查。用 Jordan 的话来说，“空前规模的调整和媒体的关注”迅速集中到公司的航线问题上。Jordan 说为期四个星期的调查给 ValuJet 公司带来的影响将长达四年。

之后，Jordan 决定设立“安全官”的职位，并提高了雇员每个季度的奖金。Jordan 表示：“很明显我们将会放弃一部分利润以及运送更多乘客的机会，但是我已经强调过了我们把安全放在第一位。”最后，Jordan 预言道：“我们会平安渡过这段时期的。”事实上，他所采取的一系列措施成功地唤回了公众的信心，提高了员工的士气。

创新举措

这类行为通常被归入战略规划事务之内。决策者的很大一部分时间花费在选择、设计、实施改革措施以及那些计划进行改进的项目方面。在很多实例中，创新举措是被已经

发生的或是可预期的外界变化所引发的。决策者必须集中注意力，专注于千变万化的市场和全球大环境，充分利用那些可能为公司带来机遇的信息做出正确的预测。由于这种预测所带来风险的大小以及现在和将来要为项目投入的资源的多少都要依赖于外界环境的变化，决策者必须挤出很大一部分时间投入这一领域。

资源的配置

不管什么形式的组织，也不管在组织中的哪个层级上，高层决策者始终是资源配置问题的核心。在一个预定的分配计划的基础上，各个层级的管理者会被赋予管理和配置资源的权力，同时要受到一套约束机制（如预算和运营计划）的限制。真正配置这些资源的权力和为了得到资源而进行的一些幕后操作，通常是掌握在高层决策者手中的。组织中的资源绝不仅仅是金钱，还包括人、机器设备、车间、仓库用地以及其他所有有助于实现公司战略目标的实体。总之，企业对资源的需求远大于可得资源，因而高层决策者必须决定如何将企业保有的有限的资源存量进行分配。

协商与谈判

当公司内部或是外部发生纠纷的时候，决策者有责任以组织代言人的身份进行协商或谈判。显然，扮演这种角色意味着要去解决各种纠纷和冲突，以及在各个派别之间做大量的协商工作。组织内部的各职能部门之间经常因为各自的职权范围或是工艺流程产生纠纷，面对这种情况，决策者必须通过在那些受到影响的成员之间进行协商来促成纠纷的化解。外部纠纷通常发生在一些外部团体的长远利益受到公司的行为影响的时候，比如恶意并购、罢工、劳动纠纷、股东关系问题以及政府的调查等。虽然图 7-1 中所显示的决策者花费在这上面的时间并不是很多（只有 3%），但这里对于快速信息和即时决策支持的需求并不亚于处理混乱局面时的情况。有时甚至需要决策者同时在两个领域内做出反应。

1997 年，United Parcel Service（UPS）公司的卡车司机因为对卡车司机工会的选举不满而举行罢工。15 天的罢工伴随着不间断的协商谈判，给 UPS 公司带来了超过 5 亿美元的损失。在这段时期内，迅速准确的信息对于成功化解纠纷来说是至关重要的。此外，UPS 的决策者不得不将他们的精力同时用于谈判和管理混乱的局面。他们为解决罢工问题所花费的每一分钟时间，也同时是在为了恢复公司正常运营而努力。由于纠纷的不确定性，对于所有可能出现的情况，都要根据它对资源利用率、整体战略计划以及公司的未来等几个方面的影响程度，按顺序予以考虑。

经理信息的类型及来源

经理信息的类型

从前面的讨论中我们知道，决策的形式有很多种，而制定专门决策所需的信息必然带有与所解决问题相关的特征。不同类型信息所具有的特征不同，并且这些特征的相对重要性也因管理的不同层级而有所不同。表 7-3 列示了各种信息的特征，并且按管理者由

低到高的顺序对它们的相对重要程度进行了比较。

表 7-3　信息的特征在各个管理层之间的比较

信息的特征	低级管理层	高级管理层
精确性	高	低
及时性	高	低
适用范围	狭窄	广泛
时间范围	过去的和现在的	将来的
关联性	高	低
详细度	高	低
概要度	低	高
定位	内部	内部和外部
来源	书面	口述和图示
可量化性	高	低

资料来源：Watson，Houdeshel 和 Rainer (1997)。

认识到各种信息的特征，人们期望决策者所需的信息及其来源是可以惟一识别的。Rockart 在 1979 年对于高层决策者需要的各种信息进行了一系列的研究，得出了以下结论：

1. 成本会计系统 (cost accounting system) 与传统的财务会计系统 (financial accounting system) 相比起来，由于从运营的角度引入了详细的收入和费用计算功能，从而更能帮助经理追踪关键成功因素。
2. 外部获取的关于市场、顾客、供货商和竞争对手的信息，在制定战略时是非常有价值的。
3. 高层经理需要那些能在几种不同的计算机系统中传播并灌输于整个企业的信息。
4. 对于组织内部和外部的问题，高层经理既需要客观的也需要主观的评价。
5. 高层经理利用信息非常注重短时间内的效果。
6. 高层经理既需要短期信息又需要那些极度多变的信息。

经理信息的类型和来源

了解了经理独特的信息需求之后，我们需要在给定条件下测定和满足这些需求的机制。Rockart 定义了五种基本确认方法。表 7-4 按复杂度和详细度顺序列出了这 5 种方法。

表 7-4　确认经理信息需求的方法

- 副产品法
- 空方法
- 关键指标法
- 总体研究法
- 关键成功因素法

副产品法　副产品法 (by-product method) 是所有用以确认高层经理信息需求的方法

中最简单的一种。这种方法来源于传统的用于概括整理副产品信息的 TPS 系统和 MIS 系统。这种方法不适用于那些预先确定了价值范围的区域或是很易概括的历史数据。数据的采集主要来自网上收集和文本报告。

空方法 空方法 (null method) 假设经理无法从正常渠道得到想要的信息。理由如下：由于经理所需要的信息具有高度的即时性和动态性，所以一般信息系统所提供的报告都是过时的。与副产品法不同，空方法使用一种非常规的信息收集方式——由可信赖的信息来源收集口头的信息。这种方法由于 Peters 和 Waterman (1982) 的 In Search of Excellence 一书而闻名，在书中他们讨论了惠普公司的决策者们使用术语“曲线管理学”一词的哲学含义。空方法依赖于组织在非正式场合所自发产生的信息交流和发现。虽然这种方法肯定了主观信息的价值，但是它也忽略了更客观的信息 (如通过计算机信息系统获取的信息) 的价值。

关键指标法 关键指标法 (key indicator method) 建立于三个概念的基础之上：

1. 一个组织是否健康可以通过比较一组关键财务指标来确定。
2. 可以基于例外报告对组织加以管理。
3. 可以使用技术手段以图形的形式来更灵活地展示关键指标信息。

第一个概念需要对一些财务指标如内部财务比率、产量水平、收入和开支、市场份额等进行认定。这些指标一经认定，其相关信息马上被收集起来并被用于制定成功运营的决策。

第二个概念说明，企业运作的价值观和一些规范可以建立在如下的概念之上：“没有新闻就是最好的新闻”。高层经理只对那些能反映本质的信息感兴趣，它们还收集附加的信息作为基础来制定可以改变环境的决策。

第三个概念需要在图中对信息加以充分利用而非在表格中。使用这种方法与表格数据分析相比起来，经理可以更容易地由图中“看”出异常的情况并迅速做出反应。

总体研究法 总体研究法 (total study method) 从高层经理中提取一些样本，并收集关于他们的信息需求总量的信息。采访调查之后，系统开发人员将经理的需求与组织中计算机系统的现有信息处理能力进行比较。一旦发现缺口，开发者将会开发子系统进行弥补。这种方法虽然比前几种方法综合性更强，但它仍然很昂贵，并且不能很好地满足单个经理的需求。因而虽然看上去颇具吸引力，但并未显示出对于 EIS 设计的实用性。

关键成功因素法 关键成功因素法 (critical success factors method) 是 Rockart 找到的用以克服其他方法缺点的最有效的方法。无论在什么性质的组织中，也无论是属于什么范畴的行为，所谓“关键成功因素”都是存在的，若它们取得令人满意的成果，将可以保证整个组织的健康和竞争能力。Rockart 认为，关键成功因素就是企业要想取得成功，就必须正确运作那些事情。与关键指标法相似，这种方法需要对已经确认的关键成功因素的信息持续地加以收集，并将其提供给高层经理。这些信息将成为制定决策的基础。

寻找组织中的关键成功因素的最主要的方法，是对公司的高层经理进行有计划的采访。之后召开一系列旨在加强合作的商讨会，会上经理将会就公司的运营目标、关键成功因素与这些目标的联系、关键成功因素的度量方法等问题进行讨论。学过了第 6 章关于 MDM 结构的知识之后，你认为这种会议最适当的结构是怎样的呢？

在所有 5 种方法中，后 3 种可以方便地作为计算机 EIS 系统设计的基础。在后面的

章节中，我们将会关注典型 EIS 系统的组成部分，研究在它们的设计和实施阶段几个有效的应用环境。

7.4　经理信息系统的组成

很多组织至今仍致力于从“大铁柜”式的主机时代向更灵活、功能更强大的客户/服务器模式过渡。通常这种进化可以使得以计算机为基础的决策支持技术的开发和应用变得简便易行。几家公司以开发和使用 EIS 系统作为这种转变结束的标志。

客户/服务器结构允许对商业应用程序做即时的增强或修改，并且推动了各种各样的数据库系统在全球的推广使用。早先的几种 EIS 系统是为在高性能计算环境下应用而设计的，但现在市场上的产品都是基于客户/服务器平台的，可以在各种条件下使用（本地、移动、因特网等）。下面列出了客户/服务器结构的几个突出优点：

1. 地理上相对分散的数据（从主机到每一台 PC），不管数据的形式如何，都被集中到一个互连的计算机平台上，数据的多样性为决策者提供了更广泛的视角。
2. 降低了投资于新的硬件系统和厂房的成本。
3. 建立了一个更灵活的、可以针对组织动态需求进行调整和自适应的平台。
4. 使组织使用“即时”数据成为可能，决策者能够做出更迅速、更精明的决策，决策时间的缩短为组织带来了竞争优势。
5. 该种结构可以很容易地建立一个“战略级”的应用程序，用以挖掘组织数据中所隐含的价值，显然，信息已经成为竞争中的武器。

图 7-2 说明了一个典型的 EIS 系统在组织的客户/服务器环境下的实现。

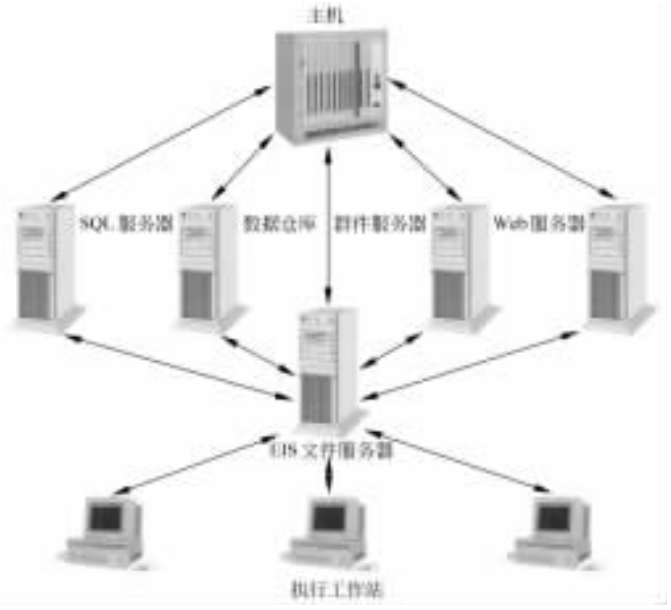


图 7-2　EIS 应用的典型的客户/服务器构架

硬件组成

除了现代客户/服务器环境所需要的设备之外，EIS 系统并不需要任何额外的硬件和外设。在客户/服务器模式下客户端所需的随机存取存储器、硬盘空间、可移动的存储介质、高速图形显示、数据访问接口、大型号显示器终端以及多媒体功能，在 EIS 系统中也同样是必需的。关键在于确保 EIS 的各个组件最优化使用，并且和组织的计算机资源保持协调，要适应现存的计算机系统和数据。在初期的培训阶段和后期的系统修正阶段，经理往往是最需要支持的用户。系统的设计必须与组织中那些能提供这种支持的资源和功能相匹配。否则，经理们将会感到不满，而 EIS 系统的优点也很难被发现。

软件组成

与 EIS 系统的硬件组成的“通用”特性形成鲜明对比，软件组成一般“专用性”较强，是为特殊用户的专门需求而设计的。尽管近年来向开放式模块化的开发环境发展的趋势使得“成品型 EIS 系统”的概念成为现实，很多供货商仍然在提供只具有典型 EIS 系统特点和功能的多用途 EIS 应用环境。这种多用途环境允许第三方把附加程序或组件集成进来，从而方便设计者针对特殊用户群的需求定制应用程序。最近一段时期，在 EIS 应用程序中使用可相互协作的群组部件的趋势正在显现出来，这在标准 EIS 功能中是没有的。Ventata 公司的 Groupsystems 和 Lotus Development 公司的 Lotus Notes 就是这样的产品。Groupsystems 可以完备 EIS 的设计，协助运行维护；Notes 既可作为现有 EIS 系统的部分整合工具，又可作为一个完整的 EIS 软件的开发平台。

只需一句话就可阐明我们的观点：EIS 的个性来源于其软件组成而非硬件。不管经理们有什么特殊的需求，所有的 EIS 系统都会包含一些常用的部件和功能。表 7-5 列出了一些以客户/服务器模式为基础的现代 EIS 的常用部件和功能。

表 7-5　现代 EIS 的常用部件和功能

- 数据访问、钻取、异常报告、趋势分析、特别查询/报告
- 多种用户界面
- 多样型的搜索引擎和算法
- 图形化的用户界面（GUI）及“导航”系统（如下拉列表、弹出菜单等）
- 多种输出通道（如屏幕显示、打印、幻灯等）
- 与商业职能的紧密结合
- 完整的决策支持系统功能
- 分布广泛的可访问外部数据库和信息库
- 数据管理功能
- 多维的数据挖掘和可视化
- 电子信息传递
- 日程安排
- 开放式结构
- 在线的相关帮助机制

续表

- 屏幕设计模板和向导
- 应用设计外壳 (解释程序)
- 多级存取控制安全体系
- 备份恢复功能
- 监视功能

当前的 EIS 技术

在 EIS 市场上可供设计师们选择的产品很多,从整体解决方案到高度个性化的系统。仔细研究各种各样的 EIS 系统会发现,如果从组件的角度看,它们是非常相似的。为了对现有产品进行分类,Dobrzeniecki 在 1994 年提出了一个三层功能分类法:

- 第一类:包含一整套由同一个供应商开发的应用程序的 EIS 产品。
- 第二类:供应商在以前开发的 DSS 系统的基础上扩展得到的 EIS 产品。
- 第三类:帮助客户组织将其现有资源捆绑、整合成一套 EIS 系统的产品。

从 EIS 的角度来看,应用软件应该从以下几个方面来定位经理的需求:

- 办公支持:提供电子邮件服务、公司内部和外部的产业新闻。支持办公自动化功能,如文字处理、日程安排、地址簿、待处理事务清单。
- 分析支持:非结构化系统问询支持;DSS 功能与帮助;对于趋势、关键指标、概述文档、异常报告的图形输出;关键词查询、钻取功能、解释文档和帮助系统。
- 个性化:允许对报告的形式、图表的类型和菜单的内容进行较灵活的修改。
- 图示功能:多种图形生成和显示类型的选择。
- 规划功能:项目管理、日程安排。
- 界面:友好的用户界面,容易学习和使用的数据导航形式。
- 实施:与组织中的计算机资源进行高效的整合;培训和全方位的技术支持;远程访问功能;数据安全功能。

EIS 商业市场每天都会因新的供货商和应用程序的产生而发生改变。在本章的最后部分我们概括地介绍了当今主要 EIS 产品所具有的功能,并通过例举几个流行的产品来对这些功能进行了简短的描述。

7.5　使用经理信息系统进行工作

从某种意义上讲,建立一个 EIS 系统与建立其他任何一种现代信息系统是非常相似的,都要使用结构化的开发方法,收集需求信息,制作原型,修改逻辑模型,实施成本分析,满足系统最终需求。从另一个角度看,EIS 系统的特征表明,若想成功地建立一套 EIS,我们仍有许多事情要去学习。对于一些开发人员来说,每一个 EIS 的项目都是全新的一个项目。为此他们不得不进入经理们的工作领域,在这里并不一定要依靠计算机才能取得成功,时间和耐心都是宝贵而有限的,问题常常是非结构化的,计算机的强大功能可能变得毫无用处。此外,建立 EIS 系统所需的物理要素,如定制硬件、数据搜寻和获取、与现有

计算机组织结构的整合,也都是这项富于挑战性的工作所特有的。暂且抛开一些政治性的因素(如中层管理者担心会受到上级的监视),我们很容易得出结论:建立一个成功的 EIS 系统将是分析员和开发人员所面临的最严峻的挑战之一。本节讨论了一些关于这项挑战的话题。

EIS 的开发框架

"开发一个成功的 EIS 系统需要怎样的过程",为数众多的研究者和从业人员把他们的精力投入到对这个问题的研究之中。Watson,Rainer 和 Koh 于 1991 年提出的"EIS 开发框架"为这项研究提供了一个标准。一个开发框架为开发系统提供了必要的术语、概念和指标,这有点像指导手册和地图。他们的框架主要分为 3 部分:

1. 结构规划:主要描述开发 EIS 系统所需的关键因素及其关系;
2. 开发过程:确认必要步骤的原动力(也就是说为什么这些步骤是必需的)以及它们之间的相互作用;
3. 用户交互:指导系统行动的界面,显示输出,为用户提供高效使用系统的工具。

要想详细地讨论这种框架,以上这些简单的论述是远远不够的,但是,我们仍将要简短地对三个组件以及它们对于一个成功的 EIS 系统的重要性进行探讨。图 7-3 表示了结构规划模块中的要素以及它们与 EIS 的关系。

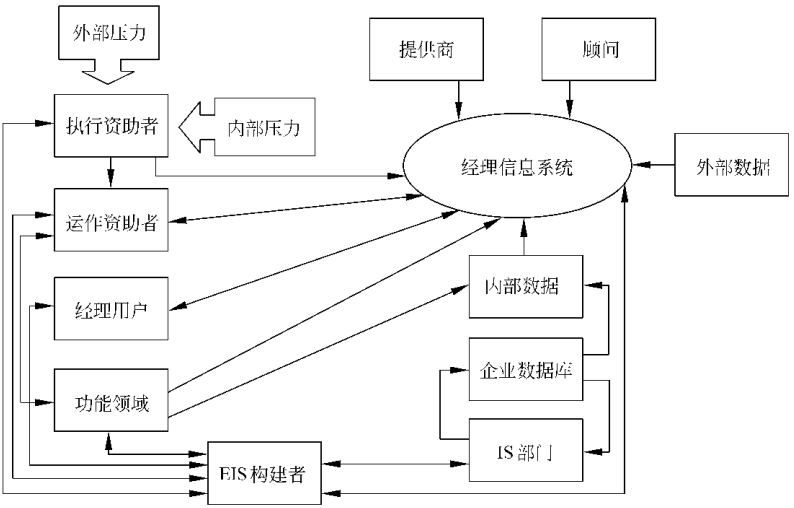


图 7-3 EIS 开发框架的结构化观点(Watson,Rainer 和 Koh,1991)

结构规划

在本模块中,焦点是与 EIS 有关的人和数据。发起人、用户群体、开发设计人员、组织中各个功能部门的人员以及与系统有关的供应商和专家都需要进行确认和定位;组织内部和外部的各种资源也要进行确认和定位。最后,系统开发的各种动力和阻力也都要被确认进来。结构规划为我们提供了一个模型,借助它我们可以更好地理解 EIS 开发各阶

段中各个要素之间的关系以及它们潜在的相互作用。

EIS 开发过程

开发过程模块以结构模型中的要素及其关系为基础，附加了时间的度量。在这个模块中，描述了各个活动和事件的先后顺序 对与时间、关键路径以及标志性事件相关的实际项目管理问题也进行了确认。图 7-4 描述了 EIS 开发过程的各个阶段。

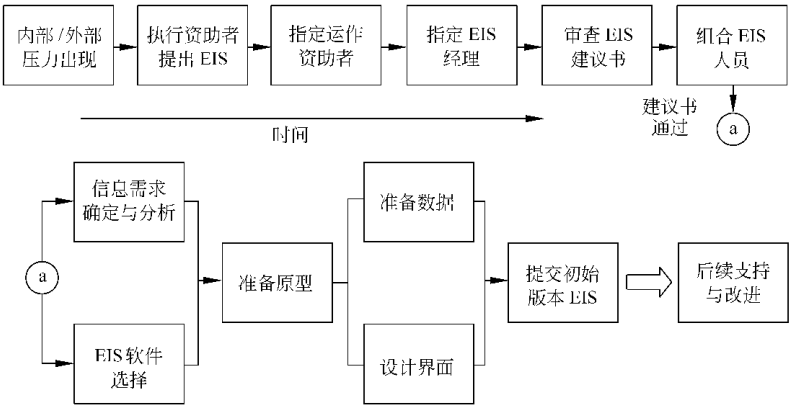


图 7-4　EIS 开发过程

与用户的交互

本模块是 EIS 的设计开发中与其他类似工具如 DSS 和 MDM 系统最相近的部分。交互系统包括一个指令集，其中含有各种命令和操作，这些命令可以指导用户如何使用系统。命令集中的各条语句可被看成是从经理到 EIS 系统的信息输入通道。

与指令集相对应，模块中还含有一个“显示（输出）语言集”，信息的输出及输出的形式都由它来管理。其中主要含有文本的、图形的和表格的格式，语音注释，以及音频视频属性等功能。

模块中的第三个要素就是知识库。第 1 章中曾详细介绍过，知识库就是经理所掌握的使用系统的知识和所有设计用来在使用中提供帮助的机制的集合。经验表明经理不喜欢读枯燥的文件。出于这个原因，要想高效地使用 EIS 系统，知识库就必须为经理提供一个风格一致的、与上下文紧密相关的帮助系统。

开发框架的设计意图在于为 EIS 系统的开发提供一个界定和指导。除了开发框架的几个要素之外，Watson，Singh 和 Holmes 在 1995 年给出了另外两个在设计和开发 EIS 系统中需要考虑的问题。首先，EIS 系统必须方便易用。作者指出：“由于经理用户的特点，系统不仅要设计的友好，更要设计成为‘用户的直觉（user intuitive）’，甚至是‘用户的诱惑（user seductive）’。”其次，EIS 系统必须有足够快的反应时间。一些设计者试图将系统反应时间的最慢基准定为 5 秒或更短，但 Watson，Singh 和 Holmes 提出，经理对反应时间的忍耐度在不同的查询操作之下是不同的，相对于简单的扫描工作来说，对那些较特殊的查询的忍耐度要高一些。作者引用了一位不知名的开发者的话来定义屏与屏之间切换可

被接受的反应时间,相当于“经理用来翻动一页《华尔街日报》的时间”。

需要注意的一些缺陷

EIS 系统的开发和使用为组织带来了巨大的收益,它改善企业竞争优势和绩效的潜能加速了它日益广泛的普及。但是,这并不是说它没有值得警惕的地方。这里,我们简要介绍 EIS 开发中的几个缺陷。

成本

Watson,Rainer 和 Koh 在 1991 年对使用 EIS 系统的组织进行了一项调查,发现平均开发成本为 365000 美元。对这一数字进行细分,结果是软件花费 128000 美元,硬件花费 129000 美元,人力成本 90000 美元,剩余的成本用于培训。调查中还发现,EIS 系统的平均年运营成本(包括运行升级和增强)超过 200000 美元,其中主要部分用于人员培训和附加培训。换句话说,从 EIS 中获得收益的代价是昂贵的,开发之后用于维护的费用很快就超过了原始开发成本。当做出在组织中采用 EIS 系统的决定的时候,必须认识到这意味着巨大的组织资源的投入。

技术上的缺陷

EIS 的本质在于,从经理的角度出发,利用相对分散的数据和信息做出决策,而这一切都是在一套简单易用的系统中完成的。完成这一功能所需的千万种数据资源以及蕴藏这些数据的各种各样的媒介,给 EIS 设计者制造了一个严峻的挑战。为了完成这一“不可思议的壮举”,开发者需要新的文件结构、数据组织形式和编程语言,还要谋求各种数据供应者和管理信息系统人员的帮助和支持。

除了设计者所面对的技术方面的问题,如何很好地将 EIS 系统与组织中现有的计算机基础设施结合起来也是一个挑战。前面已经提到过,许多组织至今仍处于从“大铁柜”式的环境向客户/服务器结构转变的阶段。这种情况必须在 EIS 设计的初级阶段就予以考虑。这种变革的趋势经常可以决定设计和运行 EIS 系统所使用的平台和软件。即便如此,EIS 系统必须做到对存储于旧式系统中的数据和存储于现代化环境中的数据二者都能访问。

组织上的缺陷

除了技术问题以外,我们还要认识到组织方面的问题。这些问题往往要比技术上的缺陷更复杂和更难以处理。Millet 和 Mawhinney 在 1992 年提出了 EIS 对于组织的三类负面影响:

1. 议事日程和时间取向的偏离。
2. 丧失了管理上的同步性。
3. 组织结构的不稳定。

对议事日程和时间的偏离　虽然 EIS 系统设计者的目标是为经理提供一个全面的信息和决策支持工具,实际上系统通常只能提供制定关键战略决策所需信息的一部分(即

使是很大一部分）。EIS 系统提供的是组织以及环境方面的可以度量的信息，而那些不具可度量性的问题常常要从 EIS 以外的渠道获得。同样的，EIS 系统只能占据经理议事日程的一部分。过分依赖 EIS，经理的注意力就会过多地集中到可以度量的信息上，而忽视了其他同样重要的信息。这会导致经理的议事日程过分倾向于一些肤浅的分析之上。

另一个使用 EIS 的缺陷来源于 EIS 自身。由于它能够为经理提供比平时更频繁更准确深入的相关信息，这很容易诱使经理将目光集中于短程的、低层次的决策上，而失去了对宏观决策的重视。EIS 系统可以促进使用小规模的管理方法来处理事件，这会扰乱中低层管理者的行为。当职员们得知高层经理可以通过“钻取”功能对他们的每一个行为进行监视的时候，他们的时间取向就会发生改变。低层管理者会将注意力转向时效很短的决策，以应付高层经理紧密的监视，同时也就失去了对长远问题的注意。

管理上的同步　这个问题是前面讨论的问题的延伸。组织中各种各样的管理步骤，不仅要被详细的描述、有效的配置，还要为了达到公司的目标而协调同步。实现这一目标的方法之一是建立一个周期性的报告体制，这种方法可以把各个管理层之间所共有的关键指标和关键成功因素信息协调起来。像许多 EIS 系统已经能够并且正在制作多种类型的周期性报告，它们也可以更广泛地被用来制作按需而定的专题报告。对专题报告过分的依赖，有时会打乱一个组织中为同步管理行为而设计的稳定有序的汇报体系。最重要的一点是，虽然专题报告是 EIS 系统中很有价值的工具，但它不能完全取代稳定的周期性汇报体系。

不稳定因素　常用的关于飞机的比喻可以更好地说明不稳定性的问题。驾驶一架飞机需要对三个物理量进行连续不断的修正：海拔高度、位移、方向。飞行员为了达到某个目的必须改变这三个量中的一个或几个；如果目的达到了，那么飞机就会飞的更平稳一些。如果飞行员对小股气流做出了过度的反应，飞机就会变得不稳定，需要额外的更集中的行动来弥补。这个比喻说明，“稳定性”的概念应该被看成是对一段时间以内的平均状态的描述。但是，保持稳定的关键并不是简单地影响平均水平，而是去影响那些基于尽可能微小的变化的基础之上的平均水平。从一个组织的角度讲，经理应定位于“做决策”而不是“做反应”。EIS 系统对经理的即时查询能做出快速的反应，使系统具有对运营环境的改变做出迅速反应的能力。但重要的是，这种快速的应答并不鼓励经理在做出频繁的或是重要的判断时出现过于迅捷的反应。经理这种过激的反应可能会破坏一个组织的稳定性，就像一个过激的飞行员破坏一架正常飞行的飞机的稳定性一样。

不允许失败

与其他开发工作一样，专门的 EIS 系统是有部分失败甚至是完全失败的可能的。无法开发出实用的系统，或是开发出的系统不可用或是不能带来需要的收益，都可称为 EIS 的失败。Corckett 于 1992 年提出 EIS 系统如果不能满足目标用户的需要，那么对于组织以及每个参与了这一工作的人来说将是一个严重的挫折。竞争优势和机遇将会丧失，失望的情绪蔓延，专家们肯定会受到负面影响，最可怕的是，组织将会对开发 EIS 失去兴趣，正如阿波罗 13 登月计划的飞行主管 Gene Kranz 说的那样：“失败不在可选范围之内。”表 7-6 列出了 Crockett 定义的许多可以导致 EIS 失败的因素。

表 7-6 导致 EIS 失败的因素

- 管理上的失败 (缺乏支持、失去兴趣、不愿培训)
- 政治性因素 (中层管理者的抵制、缺乏资源供给)
- 开发者的失误 (开发过于缓慢、需求分析不恰当、数据不真实完整)
- 技术性失误 (能力不足、不恰当的功能)
- 成本 (开发成本、培训成本、政治性成本)
- 时间 (开发时间、培训时间、维护时间)

很难预先说明如何防止 EIS 失败,当然,获得经理的广泛支持是必不可少的,培训以及参与者之间定期交流也是取得成功的关键因素。最后,在现代组织中成功地运行 EIS 系统所必需的非常重要的步骤,就是时刻警觉那些可能导致失败的因素,并将其最小化。

7.6 经理决策制定和经理信息系统的未来

未来的 EIS 技术和经理决策制定将会在几种趋势的作用下发生改变。有几种趋势现在就已显露出来,因此较容易预测,而其他的趋势则还没显露出来。尽管如此,我们还是要花时间看一看将来的 EIS 系统会是什么样子。

变革的趋势

计算机技术带给我们越来越多的舒适方便感

一种现存并将继续下去的趋势就是,经理们会从使用计算机中获得方便。在早期的 EIS 系统的开发过程中,一个很大的障碍就是经理由于对亲手使用计算机缺少经验而感到很麻烦。由于越来越注意到诸如 EIS 这样的信息系统对于组织的重要价值,针对它们的培训也越来越多,这个障碍现在大大减弱了。而且,经理的年龄构成正在越来越年轻化。新一代的领导者是从计算机时代成长起来的,他们对于能帮助自己更好地挖掘组织潜力的系统有更深刻的理解,并且能从中获得方便。随着对计算机的抵制被渐渐消除,新的更强大的 EIS 系统将会出现。

经理的职责范围的扩大

组织的结构将会越来越扁平化。这种趋势使得组织更紧凑、更灵活、反应更快,也会缩短员工与经理间的距离。裁员将会在工人和中层管理者之中进行,未来的经理将要直接面对更广泛的人员并对他们负责。我们可以设想 EIS 将会在这种组织扁平化的过程中发挥重要的作用。

组织仍将继续扩张。市场是全球化的,对于市场份额、自然资源、劳动力的争夺将会更加激烈,随之而来的将是在社会关注之下的市场规范化。经理的职责范围将会包含这种复杂性在内,而对于精确的、多样化的、快捷的信息的需求仍将继续增长。这些趋势使我们越来越依赖于 EIS 系统。

未来的 EIS 系统

应用程序和技术逐渐结合，向着一个覆盖全企业的大系统方向发展，这样的趋势预示着经理信息系统美好的未来。电信、信息系统、决策支持系统在技术上和理论上的发展，都将带来新 EIS 系统的新发展，并将促成新的更强大的 EIS 的诞生。Watson，Houdeshel 和 Rainer 认为，将来的 EIS 系统将会是一个经理智能系统（executive intelligence system）。

智能 EIS

提供给经理的海量信息是无法抗拒的，即便是使用 EIS，也要面对信息过载的危险。人工智能（AI）可以为经理提供某种形式的数据审查和过滤，以减少花费在寻找相关数据上的时间①。

语音输入功能促进了人工智能的发展，经理习惯于口头信息的传输，语音输入可以省却数据录入的时间和减少在这个过程中经理的一些失误。恰当运用语音输入和输出功能，可以增强 EIS 的灵活性以及对信息的理解与领悟能力。语音命令软件已经诞生，但还不成熟，目前的语音识别技术限制了语音输入界面的使用。但是在另一个方面，语音注释技术已经在减少长篇注释文档、增强理解力方面证明了自身的价值。随着语音识别技术上的难题被解决，智能语音控制 EIS 系统将会出现。

管理大量的语音存储需求是语音技术领域最难突破的障碍。物理上存储一个能识别各种语言、方言和其他语言习惯的系统需要海量的存储空间（不管是磁存储介质还是光存储介质），这已被证明是很困难的。而且，用以克服语音识别系统“依赖于使用者”这一缺陷的技术还不成熟，现在的系统需要用户通过一个漫长的练习过程来对其进行“训练”。为了系统能制作出用户的声音样本，用户必需拼读许多单词、词组、音节，并不停地重复它们。虽然这种方法的分辨率很高（Kurzweil Voice Plus system 可以达到 99.7% 的高分辨率），但它过于单调乏味和耗时，这两点恰恰与经理的特点相悖。几种独立的语音识别技术目前正在开发之中，如果能研制成功，那么可以迅速增强 EIS 系统的功能。

人工智能的另一个分支是专家系统（ES）。ES 可以在 EIS 的内部使用，主要用于帮助用户针对问题选择特定的模型。按照 ES 的运行规则，它可以提示用户哪一个模型最适合当前的问题。ES 和 EIS 都含有处理数据的组件，而它们的不同之处在于存储知识的方法。EIS 使用预先设定的模型和相关算法，而 ES 采用启发式的算法。我们将在第 8 章和第 9 章进一步探讨专家系统。

多媒体 EIS

出于对文件进行修复、分析、操纵和升级的需要，EIS 系统中要有一个针对数据操作的组件。多媒体数据库管理系统（MMDBMS）在结构中大大增加了用户可操纵的文本、声音、图片等数据类型。MMDBMS 不仅具有传统数据库管理系统的优势，还可包含声音、信息转换、图片翻转、目标缩放以及各种数据形式的结合。这种系统所存在的问题，特别是

① 对于人工智能技术我们将在第 8 ~ 10 章中详细介绍。

对于经理来说,就是操作界面过于复杂。随着系统功能的不断扩充,更多的应用程序将会被开发出来。把如此之多的应用程序整合到一套方便易用的系统中来以获得竞争优势,这种趋势将会越来越明显。

灵通 EIS

我们知道绝大部分决策需要外部信息的支持。这类决策的战略特性决定了它们对市场、环境、宏观经济形势、规则和技术上的变革以及竞争方面的数据的需求。现在我们对通过 EIS 获取外部数据已不再感兴趣,我们感兴趣的是 EIS 对这些数据的利用和组合能达到什么程度。

目前,世界上存在着数以万计的私人的或政府的数据库系统,随着新世纪的来临这一数字将会增长一倍。这些数据库中的信息都是可利用的,但目前用于高效的扫描、过滤、提取信息的工具还不完善。下一代 EIS 设计人员将要面临的挑战是如何将数据获取工具整合于应用程序之中,使得 EIS 可以从世界各地的数据库中系统地查找和组织信息为经理服务,并以一种便于理解和应用的形式将这些信息传送到 EIS 的桌面上来。

互联 EIS

当代因特网的广泛使用加强了公司与客户及供应商之间的电子互联。再加上以网络为中心的群件 (组合件) 技术 (如 Lotus Notes) 的使用,更加速了大规模的国际互联的趋势。电子商务的出现是一个巨大的进步,公司可以通过因特网面向现在的和未来潜在的顾客群对新产品进行宣传。虚拟市场正在形成,以因特网为基础的金融和商务活动已经成为现实。

随着高速宽带网的广泛使用,这场全球互联的风暴必然会波及经理的工作领域。未来的 EIS 会允许与企业有利害关系的各个群体进行访问,如关键部件的供应商、客户、股东、合作伙伴,这一点对于 EIS 系统处理和保持各种特殊关系来说非常重要。这种对资源和信息的共享可以促进各团体之间的联系,加强合作,打造一个通用的知识库。这一概念只是当代客户管理概念的一个简单扩充。未来的 EIS 系统仍将有人际关系管理的需求,但它将作为一种促进关系建立和维持的有价值的资源出现。

7.7 小　　结

虽然人们越来越注意到信息对于组织战略的成功执行所具有的重要意义,EIS 的概念仍处于初级阶段。随着新的决策支持和信息收集技术的出现,越来越多的充满活力的 EIS 将会被设计出来。抛开功能不谈,未来的 EIS 系统设计的目的与现在的 EIS 是相同的:为高层管理者提供他们所需要的信息,范围包含运营环境、内部管理、行业知识、突发事件、市场、顾客和供应商。将来的 EIS 系统将会规模更大、功能更强,而且会因为使用的数据资源范围不同而有所区别,但它们的目标不会改变。在信息时代,EIS 将会成为经理手中的常规武器。

复习题

1. 请为经理信息系统下一个定义。
2. 请写出 EIS 系统设计的两个要求。
3. 请描述 EIS 的“钻取”功能。
4. 组织中的 EIS 系统可以看成是其他信息技术或是计算机系统的替代品吗？请说明理由。
5. 写出经理的几种行为并加以简单说明。
6. 写出五种基本的经理信息测定方法并加以简单解释。
7. 建立组织中关键成功因素 (CSF) 的最主要的方法是什么？
8. 客户/服务器结构有哪些优点？
9. 请描述开发和使用 EIS 系统所需硬件环境的几个关键因素。
10. 写出当今 EIS 技术的几个类别并加以简单描述。
11. 描述 EIS 开发框架的组成部分。
12. 请比较用户交互系统的三大基本模块。
13. 写出 EIS 系统的几个缺陷并加以简单说明。
14. 请解释 EIS 系统可能对组织造成的三种主要的负面影响。
15. 怎样防止一个 EIS 应用项目的失败？
16. 请描述目前决策制定领域所面临的变革环境。
17. 简单介绍一下未来的 EIS 系统。

讨论题

1. 分析市场上的一种 EIS 产品，研究它的“钻取”功能并说明它是如何帮助经理的。
2. 先复习一下经理行为的类别以及每类行为所需要的信息，然后具体考察一个快餐业企业，如果是服务类行业又会怎样？
3. 研究一个组织中混乱管理的案例，在这个案例中 EIS 系统是如何帮助经理的？
4. 假设你和你的同伴是一家保险公司的高层经理。使用关键指标法和关键成功因素法来识别你所需的信息。
5. 复习一下 EIS 系统所具有的功能。具体分析一个市场上的 EIS 产品，说说它的功能。
6. 假设你和你的同伴要为组织设计一套 EIS 系统，使用开发框架来设计 EIS 系统。请指出可能导致这个项目失败的因素，你和你的团队怎样来防止失败呢？
7. 使用一种普通开发平台如 Excel 或者 Access 来为你的部门开发一个 EIS 原型。用其他的颜色标示出异常之处并使用模型来进行预测。做出一份报告说明你的开发步骤以及在开发过程中遇到的问题。

8. EIS 系统在大规模的组织中已不再是经理的专用品，它正在组织的各个管理层中迅速传播开来。讨论这种现象并为高层管理者提出促进这种传播的建议。
9. 分析一下在你日常工作中所使用的信息的类型，使用本章学过的关于信息类型的知识，描述你通常使用的信息的类型。在日常行为中你能分辨出这几种类型的信息吗？你能从“个人 EIS”中获得收益吗？

EIS 产品介绍

本部分中将会包含一些关于 DSS 产品的描述，但这绝不是一个完整的名单，因为 DSS 产品在不断被引入，被重新包装或是重新设计。它是对市场上各种各样产品的功能的一个回顾。

第一类软件

Pilot Decision Support Suite——Pilot Software 公司

Pilot Decision Support Suite 与其他的决策支持系统不同。有预见价值的数据挖掘、高效率的 OLAP、开发式即插即用的结构为它带来了很好的灵活性。Pilot Decision Support Suite 可以快速地投入使用，易于修改，随时随地都可以对分析进行支持，为我们提供了一个真正全面的解决方案。

Pilot Decision Support Suite 的应用开发环境——Pilot Designer，在快速数据获取和可视化方面是全行业的顶级产品。该设计器支持 windows95 的图形界面标准和 Vbscript 脚本语言，为定制用户应用程序、建设数据库、扩展分析范围提供了一个完美的空间。它的工具集可以帮助我们创建从简单的经理信息系统界面到复杂的 OLAP 应用程序的各种各样可视化的决策支持程序。使用 Pilot Designer，不管数据存储于何种介质中，组织中各个层级的决策者都将会有不同的数据获取途径。他们通过一种图形化的格式来获取这些信息，这种格式将表格、电子地图、文本、图片、音频和视频等各种方式整合在一起。

Commander——Comshare 公司

在 EIS 大型机系统的市场上，Comshare 和 Pilot Software 一样是领导者。IDC Research 所作的一次市场调查表明，Comshare 所获利润占整个 EIS 市场的 60%。Comshare 公司的 Commander 被认为是市场上最典型的基于 PC 的 EIS 系统。

Comshare 公司最新发布的“Commander Prism”电子表格软件，允许 Microsoft Windows 的用户对大规模的数据进行查看、分析和管理大量的数据资源，并建立详细的企业模型。Commander 系统的成功源于其几大特点。首先该系统与 Lotus 1-2-3 文件是兼容的（Lotus 1-2-3 是大多数决策者使用的用于存储数据的一种电子表格）。该系统的用户接口可以使用触摸屏、鼠标、键盘等多种形式。系统使用图画来显示财务和运营数据，在数据报告中使用多变的颜色来突出异常之处。Commander 还具有深层“钻取”功能，用户可以自己决定搜索路径，也可以选择预先设定的搜索路径。系统其他的特征还包括：可便捷地收取电

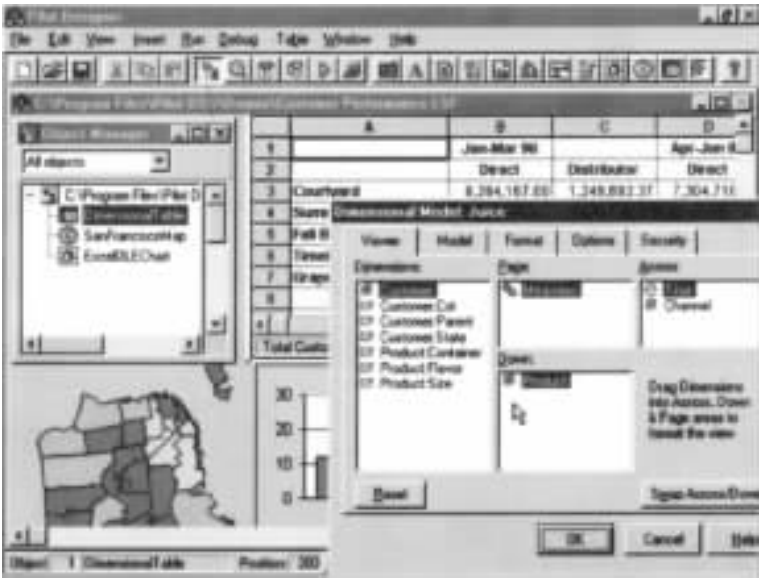


图 7-A1　Pilot Designer 应用开发环境

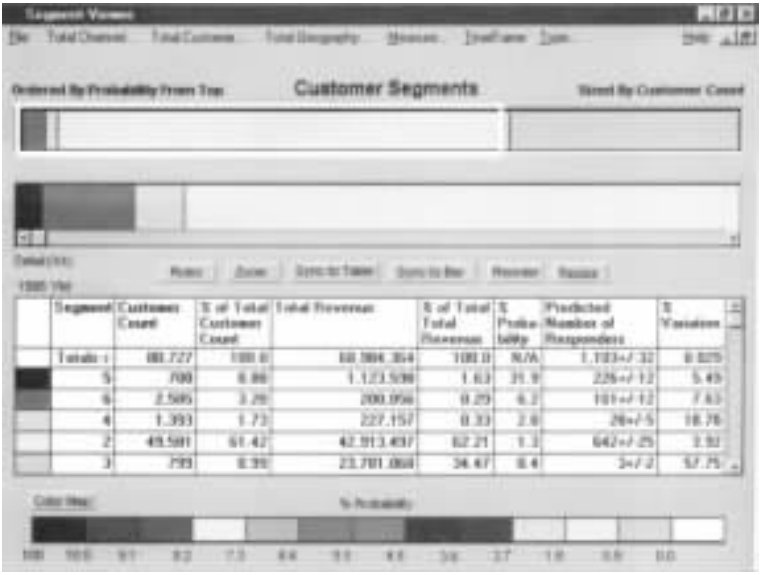


图 7-A2　Pilot 网络出版

子邮件，可以十分方便地与外界信息系统如在线新闻服务器、公告牌等进行互连。

除了 EIS 产品 Commander，Comshare 公司在市场上还提供决策支持系统。该公司的 DSS 产品一般称其为“W 系统（System W）”。W 系统是为管理会计应用如预算、财务合并、运营规划、建模、预测等等而设计的。Comshare 公司 1989 年市值 8500 万美元，这很大一部分得益于它的 EIS 和 DSS 产品。

图 7-A3 Comshare 公司的 Commander

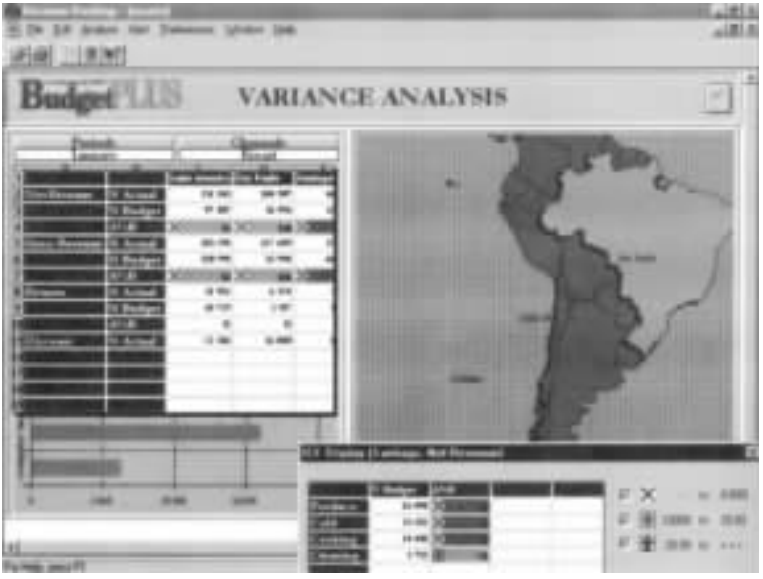


图 7-A4 Comshare 公司的 Commander

Destiny——IBM

IBM 公司提供了一种 EIS“客户设计”的服务，定制是为了让用户可以从其工作领域的信息系统中只接收和处理那些自己需要的信息。市场上提供这种服务的产品名为 Destiny，这款产品强化了单个主机的知识库系统，可以同时保存扫描图片和声音信息。知识库在这里充当着一个保存文本、图片、声音和未加工数据的信息仓库的角色。该系统的

另一个吸引人的特征是它的“任务简介手册”,对于系统中关键的地方都会提供一些帮助信息。Destiny 可以在 IBM 的各种硬件平台上开发以向顾客提供 EIS 帮助,它与 IBM 的办公系统产品是一体化的。用户接口可以是触摸屏或者鼠标。

DrillDown——Specialized Systems and Software 公司

DrillDown 系统是 Specialized Systems and Software 公司为配合 Ashton-Tate 公司的 dBase 系统的数据库文件工作而专门开发的。但它仍可以识别经过翻译的 Lotus 1-2-3文件而无需大量的编程工作。DrillDown 特别适合于那些使用 Lotus 1-2-3 作为数据库的公司,因为它可以成组甚至是全部地记录下共享数据,这种功能通常被称为“卷起”数据。举个例子来说,DrillDown 可以通过“卷起”一个简单的发货单记录文件来计算单个商品的总销售量。首先在标明各种产品销售量的列表中选定一种产品,在这里各个产品的销售量是按一定规则(假定是地区)进行分类的(这也正是用户下一步要进行选择的项目),接下来对具体地区进行了选择之后(比如选东部地区),我们就可以看到这种产品在该地区的每一个销售人员的销售量记录。

软件是通过建立一种搜寻模板才能如此迅速地找到数据,其特点是一经定义和保存,就可以被未经训练的用户重复使用,就算数据发生变化也不例外。DrillDown 是一个简单便宜的数据浏览器,主要是为帮助初学者分析数据库信息而设计的。该系统提供初级的绘图机制,并且不具备动态连接外部数据的功能。

Executive Information System 10.1——Marcam 公司

Executive Information System 是以 Trinzic 公司的产品 Forest and Trees 为基础的。它为决策者提供强大的运营数据信息的浏览功能,可以针对决策者不同的需求而提供专门的信息。Marcam 公司的 EIS 产品具有钻取、性能指标追踪、潜在问题预警机制等功能。Executive Information System 是 Marcam 公司的产品 PRISM Maintenance 的一个组成部分。

Hyperion——IMRS 公司

IMRS 公司主要为客户/服务器环境开发企业级的财务管理软件。它主要从事网络(即局域网 LANs)应用产品的设计,其目的在于使大量的用户能更快捷地访问中央存储式数据。Hyperion 系统是该公司的主要产品之一,是一款工作在 Microsoft Windows 环境下的财务信息管理软件。该公司的其他产品包括 Fastar 和 IMRS OnTrack。

Hyperion 在 Windows 的图形界面下具有公司报表功能,是一款客户/服务器软件。通过在 Windows 的用户界面中整合入高效的服务器数据库,Hyperion 系统可以提供最新的公司信息。使用这种方法,系统可以进行更简单的数据收集、数据分析、商务和财务报告等工作。

MasterCEO——Management Systems Associates 公司 (MSA)

Management Systems Associates 公司的 MasterCEO 系统,是为该公司的 Confidence Healthcare 信息系统设计的一套整合的 EIS 工具,可以提供病人关心的信息、应收款项、占

用率、门诊量以及其他一些统计信息。其文本和画图工具可以为高层管理机构清晰地展示公司的运营状态。这套 EIS 系统可以对那些经常在预算及行业标准中被使用的指标进行跟踪。

PRISM Maintenance——Marcam 公司

PRISM Maintenance 是 Marcam 公司的制造业应用软件家族的总称。PRISM 被设计用来帮助制造商对它们的日常维护作业进行监视和跟踪。这套软件特别适合那些连续加工的行业,如制药业、纸制品业以及类似的分批式的生产环境。PRISM 允许对整个生产过程和维护作业 (从用户发出订单到成品运输) 进行完整的跟踪。

Teamwork Enterprise Intelligence System II——Health Care Microsystems 公司

Teamwork EIS II 设计的目的是为企业中各个层级的管理者提供详细的信息以支持决策。传统的 EIS 系统只能满足 8 ~ 10 人的需要,而这款产品可以同时支持 100 ~ 200 人工作。Teamwork EIS II 不仅可以整合各种不同来源的数据,还可以作为全面质量管理 (TQM) 的一种工具。它可以针对单个企业的需求而专门设计,用户也可以在安装后做进一步的修改。Teamwork 系统与其他数据库系统有很好的兼容性,可以直接从与 SQL 兼容的数据库中获取数据。它适用于独立的 PC、局域网 (LAN),可以在 DEC 的硬件环境及各种 IBM 的设备平台上运行。

The Alternative View——Gerber Alley 公司

Gerber Alley 公司在 1991 年开发了名为 Precision Alternative 的医院信息系统 (HIS),The Alternative View 是其中的 EIS 组件,可以在 Windows 或者惠普公司的 New Wave 环境下使用。The Alternative View 既可以在独立的决策工作站工作,也可以在局域网 (LANs) 中工作。其包含的工具集允许用户根据自己的需要来设计定制格式。该产品惟一的缺陷就是限制了用户的开发灵活性。用户接口使用鼠标。

第二类软件

Executive Decisions——IBM

Executive Decisions 系统由 IBM 公司在 1989 年 5 月研制成功,并于同年 9 月首次发行。这款产品是为那些希望能便捷访问关键数据的高层决策者,特别是首席执行官设计的基于图形界面和触摸屏的系统。

Executive Decisions 是 IBM 公司的 Office-Vision 家族的一员,遵循 IBM 提出的“系统应用结构 (SAA)”。SAA 针对图形化用户界面给出了一个普通用户存取 (CUA) 标准以及一种能同时对局域网和 PC 中的信息进行访问的联合结构。该系统可以在如下几种基于 IBM 操作系统的平台上运行:MVS,VM,OS/2 EE。由于该产品在用户中受到好评,IBM 公司正计划让它能支持 AS/400 环境。Executive Decisions 系统运行需要使用 OS/2 EE 1.1 的 PS/2 模型 70 或 80,可以与使用 VM 操作系统的 IBM System/370 联机工作。

IBM Data Interpretation System——IBM

首次发行于 1989 年 9 月。该系统是一套允许管理者使用鼠标接口对数据进行访问、分析和处理的图形化的决策支持系统。目标用户是公司中财务、市场、企划部门的管理者。运行该系统需要 OS/2 EE 和四套使用 System/370 运行相关数据库的 PS/2。

Express/EIS——Information Resources 公司

Express/EIS 系统允许用户对非结构化的数据进行跟踪、识别、分析以及在组织内部进行信息交换。多样化的图形数据展示支持智能预警报告和钻取等功能，其中钻取功能使得用户可以在自己定义的标准下对深层数据进行操作。系统还为用户提供了自动识别和突出“数据新闻 (news in the data) ”以及针对用户需求定制报告的功能。此外，典型的经理信息系统还应具有大多数决策支持系统都有的分析功能，Express/EIS 系统也同样具有这种方便快捷的用途。

Express/EIS 使用具有统计和计算功能的第四代计算机语言、多维数据库以及数据字典。使用这些工具，用户可以更灵活地提出系统设计阶段没有涉及过的问题，从而进行“即时”分析，这种特征在其他执行信息系统中并不多见。系统还具有数据查询、报告撰写以及软件内部图形保存等功能。Express/EIS 使用 3270 终端仿真协议来访问 DB2 (IBM 产品) 中的主机数据库。

Forest & Trees——Channel Computing 公司

运行于 Windows 或 DOS 环境下，可以读取很多种类型的标准 PC 数据库、电子表格和结构化 ASCII 文件，还具有使用图标从 Lotus 1-2-3 文件以及其他来源中获取活动数据的功能。Forest & Trees 系统还可以产生 SQL 查询和动态数据交换 (DDE) 请求，使得开发者可以创建视图和复选框，让用户来选择数据链接和输出格式；“查询助手”功能可以帮助用户以非正规的方式来研究数据；该软件还具有钻取、针对不同条件使用不同颜色编码以及警报等功能。Forest & Trees 系统在特殊查询和识别跟踪关键数据条目等方面均优于同类软件。一旦 EIS 系统建立，Forest & Trees 的维护需求很低，因为该软件是从其他应用程序中摘录数据的。

Executive Eagle——Transitions Systems 公司

该系统可以整合来自于 Transition Systems 公司生产线以及其他数据资源的信息，是一套可以针对个人需求而定制的 PC 机系统。与这种定制功能有关的几个特征包括：“公司浏览器”——可以对特殊数据集进行快速浏览；“实况手册”——可以跟踪成功因素、图表、异常报告以及其他信息的一个文件夹，以避免将来为了进行对比而再次更新它们。建模和规划功能允许用户输入详细的对未来趋势的假设，并以图表的形式进行观察。异常报告功能还允许用户提出假设分析 (what-if) 。Executive Eagle 系统可以运行在 IBM 的兼容机上。

Smartview——Dun and Bradstreet Software 公司

Smartview 是在由混合案例分析、程序级的成本会计、客观生产率分析等组成的决策支持数据库驱动下产生的 EIS 产品。该系统是以工作站为基础的，用户接口采用触摸屏或鼠标。系统可以提供定制的信息视图，突出和整合各个关键成功因素，同样也有钻取功能。

第三类软件

EIS ToolKit—— Ferox Microsystems 公司

EIS ToolKit 系统是基于 Ferox Microsystems 公司的 EIS 产品 Encore Plus 而研制的。它是一款完全用户化的附加型模块，使得那些非专业人员也可以编写精密的 EIS 系统。使用 EIS ToolKit 开发的系统具有异常报告、钻取、实时预报和趋势分析等功能。该系统可以根据用户需求专门定制。

EIS Pak——Microsoft 公司

EIS Pak 工具集开发的目的在于加快使用 Microsoft Excel 以及其他主流应用程序开发 EIS 系统的进程。其中 EIS Builder 是一种附加开发工具，可以帮助开发人员迅速创建以 Excel 为基础的企业级应用程序而无需编写大量代码，它具有利用图形进行查询和分析的功能，还可以与外部数据库和应用程序交换数据。EIS Pak 使得简化过的高级图表分析和文本格式化等功能成为 Microsoft Windows 应用程序不可缺少的组成部分。这款产品被设计用来满足组织中部门级管理者的需求。其他的功能包括视频、范例程序、开发向导等。

Hospital EIS——Global Software 公司

这款基于 PC 的产品运行于 Windows 环境下，用户接口使用鼠标。该软件可以对战略方面、战术方面、运营方面、财务方面的数据进行访问和分析，还可以与 Global 公司的很多应用程序相结合，如电子表格、数据库程序、桌面打印以及一些外部资源。该系统中还结合了安全机制，限制了对预先设定的某些信息的访问量。这使得用户只能访问那些对他们有用的 EIS 数据。

IMRS OnTrack——IMRS 公司

IMRS 公司推出过几款财务管理软件，IMRS OnTrack 是一套可视化的信息获取软件，它是基于网络的商业信息管理软件包，通过网络平台向决策者、管理者和分析员传送全面即时的信息。OnTrack 向用户提供综合而简明的信息，数据以高度概括的形式呈现给用户，用户可以使用钻取功能来获取详细资料。其图形处理系统可以通过“卷起”功能使得用户可以查看那些屏幕无法全部显示的大报告。该系统通过复合文本和对数据、图形的注释来对其报告进行增强。

MAXPAR——Maximus 公司

该系统主要应用于医疗领域。它是一款基于 PC 的产品，使用 C 语言编写，在 UNIX 或 DOS 环境下运行，同样适用于局域网。MAXPAR 利用从主机或整个计算机系统的分部上下载的数据来生成和升级 EIS 内部数据库，通过获取公司的财务、生产、雇员的数据，使得决策者可以制定低成本高利用价值的计划和业务模型。

Media——Info Innov 公司

这是一款基于 Microsoft Windows 的 EIS 产品，Media 的界面围绕着关键数据条目进行建设，以便于提供强大的钻取功能。它以结构化的 ASCII 文件的形式从其他应用程序中获取数据，并将必要的数据编入一个专用数据库中。这种方法需要给予 EIS 比其他软件产品更高的维护支持。维护要求高，但钻取和数据分析功能更为强大。

Quantum——HBO 公司

Quantum 是对 HBO 公司及其子公司 Healthquest 以前生产的各种 EIS 应用程序的综合。这款产品在 IBM，DEC 和 HP 平台上具有快速安装和便携性。Quantum 工作于 IBM PS/2 或苹果机上，通过局域网与主机连接，用户接口采用鼠标、触摸屏或键盘皆可。该系统提供一套使用 Pilot Software 公司的 EIS 工具开发的包含 70 种管理报告程序的文件库，这些报告采用“时间序列法”来分析潜在趋势。系统使用一套相互关联的表格并将数据统一存储在主机中的单一数据库中，因而所有用户所看到的数据都是相同的。其强大的整合功能使得用户可以与基于不同硬件的系统相连接并从中获取资源。

EIS 软件：实施与开发

DSS Executive——MicroStrategy 公司

DSS Executive 系统是 MicroStrategy 公司研制的用于开发 EIS 程序的设计工具。它们的 DSS 开发工具——DSS Agent 所具备的所有高级报告功能在 DSS Executive 中都是通用的，因而它可以创建“任务简介手册”、“动态报纸”以及系列报告等工具。

Time Base——Pilot Software 公司

Time Base 是一个具有时间序列功能的多维数据管理分析系统，用户可以推测发展趋势、根据不同的时期的特点重置 EIS 的结果。Pilot Software 公司针对局域网开发的 EIS 产品是它的另三个产品（Lightship，Lightship Lens 和 Time Base）的集成。

PowerPlay——Cognos 公司

PowerPlay 2.0 时一款 EIS 产品。对于管理者而言它是一个数据浏览和演示工具。该系统适于个人使用，对于公司数据具有“钓取”功能。PowerPlay 可以在 Microsoft Windows 和惠普的 HP New Wave 环境下运行。该产品突出的特点是报告生成和钻取功能。外部

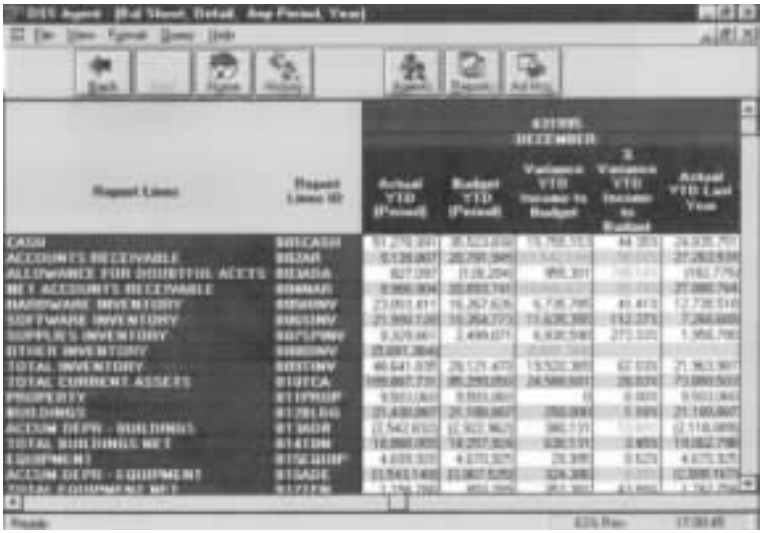


图 7-A5 Micro Strategy 公司的 DSS Executive

图 7-A6 Micro Strategy 公司的 DSS Executive

信息可以以结构化的 ASCII 文件的形式进行输入。这套 EIS 系统维护性很高,因为数据来自 ASCII 文件而且要被编入专用数据库中。与前面介绍过的 Media 相同,PowerPlay 也有利有蔽:它的维护度很高,但有非常强大的钻取和分析功能。

Excel——Micrcsoft 公司

Excel 是功能强大的电子表格软件。它的信息操作"bells and whistles"可以满足商务信息需求。Excel 具有统计数据分析、数据库开发、与其他软件之间的交互、图形输出等特点。IBM 公司表示他们支持将 Excel 电子表格引入其 AS/400 微型机,这将会提供一种在不同的计算机硬件平台之间交换数据的新方法。

Lotus 1-2-3——Lotus Development 公司

EIS 系统通过在其各种报告中使用来自 Lotus 1-2-3 的工作表单来完成与 Lotus 1-2-3 电子表格的结合。EIS 系统编译各种各样的数据，并允许用户在详细程度不同的数据中工作来寻找信息。EIS 把来自电子表格的数据和其他来源的数据整合到一起，并以一种简单明了的形式显示出来。通常 EIS 系统可以把 Lotus 1-2-3 作为其前台终端，一方面可以利用用户对 Lotus 1-2-3 的格式的熟悉，另一方面可以用来检查海量的商业数据。

International Data 公司在 1991 年所做的一项调查表明 在马萨诸塞州佛莱明顿市，接受调查的公司中有 65% 在它们的决策制定系统中使用了电子表格，不管是作为大系统的一部分还是作为独立的一套系统 在 1991 年 12% 的公司安装了 EIS 系统。

目前，很多公司正在致力于从诸如电子表格之类的普通产品中拼凑出一个执行信息系统。Lotus 1-2-3 与 Commander，Pilot Lightship，Forest & Trees，DrillDown 等系统，以及 Information Advisor EIS 软件公司的产品都是兼容的。

Lotus 1-2-3 不是数据库管理系统，因而如果把它当成 EIS 系统的数据仓库来使用，将不会很好地发挥功用。当工作表的容量达到一定限度的时候，速度会减慢下来并且变得笨拙和难以控制。一个较好的解决办法是将电子表格升级为数据库产品，如 SQL Server，dBase，Paradox 或者 Oracle。很多公司都正处于这个阶段，它们正在寻找一种可以顺利实现这种转换而不会影响它们将来开发完整的 EIS 系统的产品。

基于 Windows 的 Lotus 1-2-3 可以作为 EIS 系统的前台终端。使用 SmartPak（一款最新的 Lotus 1-2-3 的补丁产品），用户可以对 Lotus 的菜单进行修改，定制对话工具箱，建立一个使用电子表格的简单的 EIS 系统。

NotesWare EIS——NotesWare 公司

NotesWare 对于决策的支持主要是通过一套用 Lotus Notes 开发环境开发的企业信息系统来实现的。它将各种来源的信息集中起来进行分析，并以要求的形式做出报告。决策支持系统采用简单的点击界面，允许用户查看报告和分析标准，钻取详细数据或者建立个人查询。NotesWare EIS 的主要特征包括：强大的屏幕设计工具、菜单和子菜单、快速查询向导、直接访问 SQL/ODBC 数据库、全部或部分功能的操控向导，以及与电子表格（如 Lotus 1-2-3，Excel，Quattro Pro 和 OLAP 等）的集成、多维数据支持、钻取和切片（cut/slice）功能。

Performance Advisor——SRC Software 公司

这是一个高端的财务报告生成器，是从 Lotus 1-2-3 中调用的一系列菜单。Performance Advisor 系统增加了用以获取外部数据并加以分析和报告的菜单、功能和方法。该系统从主机或其他财务系统中获取数据来建立灵活性报告。

Information Advisor——SRC Software 公司

Information Advisor 系统与 Lotus 1-2-3 文件联合使用，作为 Performance Advisor 的前

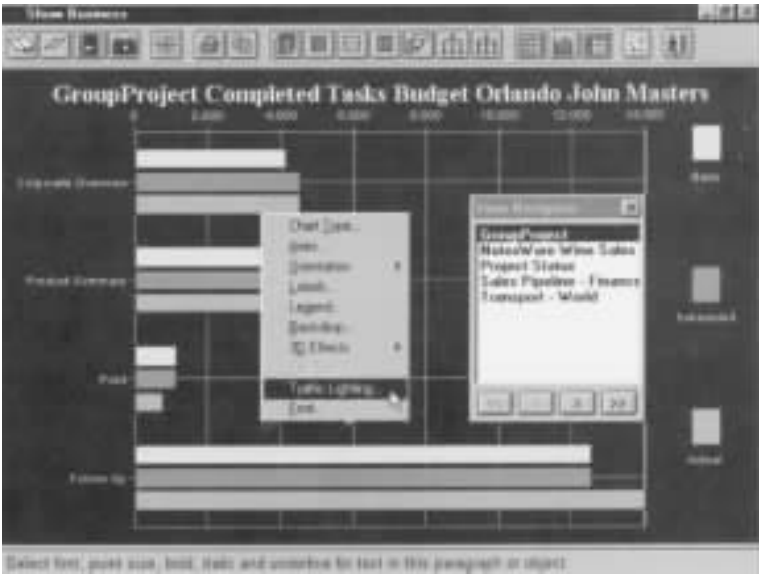


图 7-A7 NotesWare 窗口

台终端。这种附加式的特征提供了一种帮助初学者浏览现有报告的机制。

第 8 章

专家系统与人工智能

学习目标

- 专家系统与人工智能的概念
- 描述几种人们常用的推理过程
- 描述用来建立基于计算机的推理系统的有效方法
- 解释专家系统的概念和结构
- 理解建立一个专家系统要进行的预设计活动
- 学会如何评价一个专家系统

小案例

专家系统辅助发现欺诈

我们可以开发相应的专家系统来协助我们发现欺诈,这些系统可以通过危险信号来标示出可疑的地方。在欺诈者得手之前,调查人员可以投入更多的时间在这些可疑的项目上,这样就可以阻止欺诈行为。

包括银行、保险公司、美国政府在内的许多行业都利用专家系统来发现欺诈。美国政府的医疗保险账单欺诈、健康和财产保险索赔欺诈、信用卡欺诈和外汇交易欺诈是几种能被专家系统检测出来的常见的欺诈事务。

在电子商务日益膨胀的今天,大量的事务处理使得我们很难靠人力去仔细地检查所有的数据来防止欺诈,而绝大多数事务是正当的,需要我们及时处理的。实际上,这里的部分问题是顾客习惯于利用计算机提供的快速服务,因而期望更快的服务速度。

财产保险和意外灾害保险索赔

索赔调查员往往处在快速的处理索赔事务的压力之下。近年来,随着夸张或者伪造索赔事务的数量不断增加,索赔调查员的这项任务变得越来越难。根据行业估计,10%至 15%的索赔都是欺诈。绝大多数公司采用电话和计算机代替了人工的沟通去采集信息,这助长了欺诈问题的发展。现在,索赔调查员必须及时有效地去划分责任、评估损失并处理假的或者虚报的索赔。

对可疑的索赔,我们可以采取附加的调查,但这样可能会增加开销并拖延解决问题的时间 若不进行任何附加的调查的话,公司将支付虚假的索赔款项,并招致将来更多的伪造的索赔。在面对大量存在疑点的索赔的时候,即使是经验丰富的索赔调查员也会处于左右为难的境地。难点是要决定哪些索赔可以接受,而哪些索赔要拒绝。

更困难的是,确定一个虚假的或者夸张的索赔只能说是解决了问题的一半。为拒绝索赔,我们必须通过律师和陪审团这一关,这才是真正的挑战。这将是一场昂贵的、旷日持久的官司。如果官司输了,保险公司还将面临惩罚性的赔偿。

专家系统可作为财产保险、意外灾害保险欺诈问题的解决方案的一部分。通过对索赔信息的严密检查,专家系统能鉴定并用危险信号标示明显疑点。在专家系统的协助下,调查员能采用专家系统自动质问可疑索赔得出的观点。

一个处理汽车人身伤害索赔的专家系统

位于马萨诸塞东部的尼德姆镇的 Correlation Research 公司有一个专家系统的原型模型,这个原型模型包含了他们与马萨诸塞州汽车保险公司管理公署一起获得的许多有用的数据。索赔欺诈的指标和保险公司为对抗欺诈者所采取的行动被作为关键变量收集起来。索赔调查员可以针对每个索赔项目回答 10 至 15 个问题,问题的答案将输入专家系统,而系统则通过这些答案给每个汽车人身伤害索赔标上一个“可疑分数”。这些分数可以用来鉴别索赔,并决定对各个索赔如何采取行动 :可能是立即赔偿,也可

能是重新索取一份医疗意见书,或是收集更多的信息。只要索赔调查员能得到一份身体检查报告或者收集更多的信息,索赔就能得到很快的处理。

这个专家系统仍然在不断地改进,但是它已经极大地提高了我们制定如何处理各个索赔决策的速度。随着大量的索赔被专家系统处理,储存索赔性质、分数结果、处理方案的数据库不断地成长起来。最终将形成专门评定欺诈分数和计算处理索赔成本的专家系统。每个索赔所需的核心数据将随着时间的过去而改变,这些数据将成为揭露虚假索赔的新手段。索赔调查员的工作效率和准确度提高了,而建立专家系统的成本则与原来收集信息的成本大致相抵。

公众关注

公众对保险公司的同情心不如 20 年前了。他们要求保险业控制欺诈行为,并且不允许保险公司通过提高保险费率把支付虚假索赔的成本转移给顾客。公众关注的结果就是,许多保险公司拥有了自己的索赔信息数据库支持的专家系统。国家保险犯罪委员会正在考虑建立一个专家系统,这个专家系统拥有一个巨大的国家数据库服务项目。将有 150 个保险公司、10 个州立保险欺诈公署和超过 20 个执法机构使用这个即时服务项目。

医疗账单欺诈

医疗账单欺诈和滥用医疗账单在某种程度上能用专家系统发现。专家系统拥有一个索赔审核软件,这个软件能像一个真正的专家一样去检查可疑的索赔项目。专家系统利用医疗诊断过程的规范来寻找欺诈。滥用这些规范将增加医生的医疗保险金的赔偿。1992 年,健康保险欺诈的金额总共有 840 亿美元,占当年美国健康保险支出的 1/10。

信用卡欺诈

银行业使用专家系统。1993 年,加拿大信托银行安装了 Trinzic 公司的基于知识的 Aion 开发系统,该系统用来监控信用卡交易模式。这个专家系统很成功,以至于加拿大信托银行不再使用万事达卡的报告。

在工作日,每隔两个小时都要检查一次数据 ;每晚也检查一次。按照交易是否属于欺诈行为的可能性,系统将给每个交易标示一个分数。比较可疑的分数将发给银行的反欺诈部门,以便采取更加严密的调查。采用这种方式,银行可以很及时地调查每笔可疑的交易。由于大部分的交易都是真实的,我们用手工检查出可疑的交易的可能性微乎其微。适时地处理问题是预防进一步的欺诈的关键。

外汇兑换交易欺诈

银行也用专家系统来监视外汇兑换交易。纽约化学银行已经设计了一个名为“检察员”的专家系统。它能连续地监测位于世界各地的交易所的外汇兑换交易。任何有问题的交易都会被标示出来,这样外汇兑换专家就能检查这些交易的细节了。

“检察员”为银行节省了大量的时间。而且由于纽约化学银行每天要处理超过 10 亿美元的外汇兑换业务,“检察员”成了银行成功的重要因素。一个欺诈的交易可能会

使银行损失上百万美元。

将来会有更多的欺诈吗？

专家系统在欺诈发掘领域建立了属于自己的位置。当所有的专家系统连接起来，以至于能监控所有的业务的时候，专家系统真正的潜力才能被人们意识到，才能发挥出来。通过这种方法，我们能轻易地发现横跨几家保险公司或者银行的欺诈交易。在交易业务中，专家系统扮演了业务检查员的角色。专家系统提供可疑点的警示，这能给专家们提供了进行人工调查可疑点的宝贵时间，而其他的业务也得到高效的处理。

8.1　专家的定义

丹麦物理学家 Niels Bohr 曾经把专家定义为“能解决狭小的领域出现的所有错误的人”。虽然这个定义可能有一点夸大，但是，他指出了专家是在特定的领域的观念。要想解决给定知识领域的所有问题，一个人必须经历所有可能会经历的事情。因而，只要我们给出足够的关于特定问题的信息，一个专家通常能制定一个近似的解决方案，这个方案也是非常可靠且有效的。尽管这些可能看上去更加依赖直觉而且解释不清楚，然而专家是如何如此高效地得到有效的解决方案的？这个问题仍然存在，而问题的答案就在逻辑领域。

逻辑推理证明是解决真假问题的一种有效方式。逻辑推理基于这样一个前提，那就是特定的知识领域的所有知识可与其他领域的所有知识以某种方式联系起来。在这个前提下，关于该领域的新情况的事实或者观察资料必须相符合。这种符合的必要条件是允许应用像演绎和推理等逻辑方式去得出问题的结论。专家们将特殊知识领域的信息结合起来，并利用这种方法去得出他们的结论。

考虑你所能想像的所有的状况，不管它们是否包括数据解释、优化、规划、分析或者社会行为，它们都可以有规律地表示出来。如果我们考虑用一系列的“如果—那么”描述问题，我们可以看到专家们如何展现他们的“神奇”。他们用现有的信息去获取新的信息，从而得到结论。通过应用信息匹配的概念，他们能建立一个有效的调查步骤去收集必需的信息，然后得出结论。我们先看看下面的例子：

已知信息：

- John 是 Sam 的儿子。
- John 是最大的孩子。
- Mary 是 Sam 的女儿。
- John 和 Mary 的母亲叫 Anna。
- Sam 和 Anna 已经结婚 50 年了。

推知的信息：

- 如果 John 是 Sam 的儿子，则 John 是个男孩。
- 如果 Mary 是 Sam 的女儿，则 Mary 是个女孩。

- 如果 John 和 Mary 的母亲是同一个人,则 John 和 Mary 是兄妹或姐弟关系。
- 如果 Sam 和 Anna 结婚了 50 年,则 John 和 Mary 是他们亲生的或者收养的孩子。

上面的例子应用了信息匹配的概念和规则去获取没有明确指出的信息。逻辑推理显示,如果已知信息是正确的,那么推知的信息必定是正确的。这样,我们能够收集到足够的信息去快速地制定正确的决策,因为我们知道哪些具体信息是我们制定决策时所必需的。我们可以很容易地想到,通过这些概念(信息匹配的概念和将问题看作一系列的规则的想法),我们可以开发一些扮演这些"专家"的计算机软件,这些软件可以在事务处理中灵活地应用。

专家系统与人工智能

专家系统(expert system,ES)通常描述的是一种计算机软件。软件中使用了一系列的规则,这些规则是从人类解决实际问题的经验中得来的。在解决特定知识领域的问题时,专家们运用到了信息匹配的概念,专家系统则模仿了基于这一概念的推理方法。即使是一个不是专家的人,他也可以通过专家系统来提高自己解决特定领域的复杂问题的能力,而他所需做的仅仅是与专家系统的交互式的对话。专家们也可以将专家系统看作一个知识渊博的助手。专家系统可以考虑用来弥补组织中专家资源的不足,这样可以增加解决问题的质量和持续性。

专家系统作为专家的功能主要是应用了人工智能(artificial intelligence,AI)领域的相关技术。人工智能主要研究人们如何思考、推理和学习,而且人工智能是横跨计算机学科和认知心理学的交叉学科。从这种角度理解,人工智能试图发现并发展一种可用的机制,这种机制使计算机能够模仿人们在解决特定问题时所使用的推理方法。Rich 和 Knight 在 1991 年给人工智能下过一个定义,他们认为人工智能是"如何使计算机能做到人所能做的事情的学科"。尽管这些人工智能机制并不完全等同于人类的思维机理,但是,它们能得到与人类决策者相似而且有用的推理结果。

在这一章里,我们将着重介绍专家系统和人工智能在制定决策和解决问题中的重要性。我们也将考察专家系统和人工智能的发展历史、构造和一些附属概念,这将有助于我们了解它们当今的发展状况、应用软件以及未来的发展潜力。在我们简要地集中介绍专家系统和人工智能的优点及其局限性之后,我们将为第 9 章打下基础,第九 9 章将详细地介绍专家系统中所应用的知识和方法。

专家系统和人工智能的发展历史

专家系统和人工智能的根源可以追溯到 20 世纪 50 年代中期,以及兰德公司卡耐基小组的工作。卡耐基理工学院(现在的卡耐基梅隆大学)的 Herbert Simon 和 Alan Newell 以及兰德公司的 J. C. Shaw 共同领导了一个开发基于计算机的推理系统的研究小组。他们的第一个系统就是 LT(Logic Theorist),这个系统以一套公理作为逻辑计算的知识基础,并结合反向推理与逻辑计算来解决一些简单问题。反向推理就是先分析原问题的状态,然后将原问题分解成一系列独立的子问题。分别地解决这些子问题,整个问题就解决

了。到 50 年代后期，这个研究小组启动了一个更具雄心的项目，开发一套 GPS（General Problem Solver，通用问题求解程序）系统。GPS 不同于 LT，它使用的是一般推理方法技术，能解决多种问题。GPS 能解决基本逻辑问题、国际象棋问题，甚至能解决中学代数问题。使用了解决问题的通用步骤是系统方案的一个重大突破。

直到 60 年代早期，专家系统的雏形才真正显露出来。麻省理工学院的 Minsky 和 McCarthy 开发了一种名为“LISP”的人工智能程序设计语言，它是开发人工智能系统的一种颇有影响力的语言。与此同时，斯坦福大学开发了一个名为“DENDRAL”的专家系统，它能够通过大量的光谱信息推测分子结构。这是专家系统开发技术高速发展的时期，也是规划今后 10 年建立产业化的专家系统所必需的知识的时期。

70 年代，专家系统在产业领域得到了广泛的应用。它们在医疗诊断、心理学、教育、学习、定理证明、竞赛模拟、制图及语音识别等各个领域都显示出了极大的作用。一些主要的现代化的大型机构都在开发和使用人工智能技术和专家系统。比如，数字设备公司开发了 XCON 和 XSEL 两个专家配置系统；通用电气公司开发了一套名为 CATS-1 的柴油机故障检修系统；施乐、IBM、通用汽车、德州仪器等公司也在使用现代化的专家系统。现在，全世界差不多有数千个专家系统在运行，而且一个繁荣的专家系统开发软件市场正在崛起。

8.2　人工智能的智慧

按照先前的说法，人工智能着重认识人们的推理方法和思维方式，并且将认识到的情况转换成机器语言，让计算机系统去完成与人们相同的推理和思考任务。为了更好地理解人工智能如何去完成转换任务，我们先要考虑人们是怎么推理和思考的。

人们是如何推理的

我们不需要针对这个题目对人类的推理过程进行一次彻底的讨论。认知心理学集中研究影响人们推理和反应的复杂过程和环境，它已经发展成了一个完整的、复杂的知识领域。虽然如此，我们还是能确定一些人们使用的简单的推理方法，这些推理方法能很容易地转化到专家系统和人工智能计算机领域。

归类法

与人类的推理相关的最普通的方法就是归类法（categorization）。当我们确定了一些足够重要的信息，并想把它们记录下来的时候，我们按照不同的性质或者标准把它们分类。各个类有不同的子类，信息将存储在子类的存储器里面，较低层子类的信息可以从高一层父类信息那里继承全部属性。图 8-1 举例说明了归类法的概念。

每一层的信息片断的属性都会详细地记录下来，依靠这些属性，我们可以运用分类法从更高层次信息的属性得到低层信息的属性。例如，如果小汽车是一种陆上交通工具，公共汽车也是陆上交通工具，那么，它们一定用相似的方式运送乘客，因为它们都继承了陆上交通工具的属性。同样，我们也可以从低层的信息的属性细节中抽象出高层信息的属

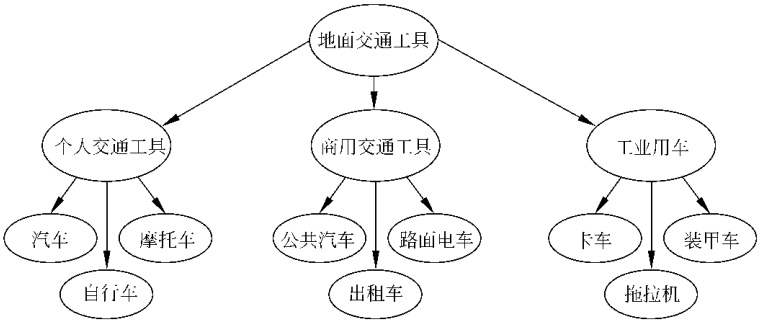


图 8-1 归类法的概念举例

性,例如,陆上的载客交通工具都有轮子,则小汽车也有轮子。分类法也应用了由类别之间的联系衍生出来的一些规则。

特定规则

人们推理的另一种方法,就是灵活应用已知规则。如果一个特殊的规则或者是一系列的规则是已知的,而且是正确的,那么我们可以结合这些规则,并运用现有的问题的信息推出结果。例如,如果我们知道路上遇到红灯必须停车,而且知道闯红灯会收到警察开出的罚单,那么我们可以推出,如果遇到红灯,我们应该停车(除非我们宁愿接受罚单)。另外一种推理形式就是我们所熟悉的法律。比如税法,现在有一笔特殊的开销,而且是正当的,这种开销在交税的时候是要扣除的。那么到底该申请扣除多少税金呢?由税法中的详细规定就可以得到答案。人们通常把一些规则联接成一个推理方法,运用这些方法可以得到可靠的结论。

启发式方法

我们在第 3 章讨论决策过程的时候曾经讨论过启发式方法这个话题。启发式方法是人们推理的另外一种方法。这种单凭经验的方法也有很大的用处。比如,如果菜谱里面有红牛肉,那么我们就要选择喝红酒 如果芝加哥公牛队是在主场作战,那他们很可能会赢。上面的两个例子都是启发式方法,都将以前获取的经验作为了判断的法则。尽管启发式方法缺乏正规性、严密性,但是,如果我们在调查和决策时使用那些非常有根据的启发式规则,我们将花费更少的时间来形成一套可靠的解决方案。

经验法

依靠经验的推理方法是归类法和启发式方法的结合体。我们可以如此看待经验法,也就是当人们考虑整个问题时,他们先将问题分类,然后试图与过去他们所经历的一些事情做比较。如果事件有足够多的特征或者场景与过去的经历匹配的话,我们就可以基于过去的处理方法得到必要的结论。这里有一个最普通的运用经验法推理的例子,那就是本科生和研究生教学中所用到的案例教学方法。商务案例叙述了一个人或多个人或组织在特殊环境中的行为。在整个课程中,除了详细地叙述事件的细节、制定的决策之外,这

些案例还要记录决策结果，并检验这些决策的正确性。人们也正是用这种方法进行推理的。如果当前的情况和以前的经历很相似，而且我知道以前我是怎么处理这个问题的，我就可以结合当前的情况得到一个相似的处理方法。这样的推理方式有一些假设，就是历史可以重复，而且当前的情况在大多数方面和过去很相似。如果这些假设条件都满足的话，经验法是一个非常有效的制定决策方法。

期望法

人们推理的最后一个方法是通过期望（expectation）推理。一旦我们经历一种问题或现象足够多的话，我们就开始期望问题以一定的方式或以可预期的情形出现。如果事情如我们期望的那样发展，我们就能推断一切状况良好 如果我们期望的情形没有出现，我们可以推断，一些事情改变了我们期望的状况。

我们每天与之共事的人都有自己独特的性格，而且这些特性是有规律地显露，以至于每天，我们都预期会看见这些特性。这也是一种期望的推理。我们若出乎预料地看到有些同事与预期状态不同，根据我们的观察，我们就会以“今天你不在状态啊”或“今天有什么烦心的事儿吗”做出响应。期望推理是模式识别的一种简单形式。我们期望有一个确定的模式，当这个模式与当前的情况有些不符合的时候，我们会采取一系列的调查活动来确定那些不同点。需要注意的是，虽然期望推理和经验法有些相似，但实际上是不同的，因为我们的期望是在一系列相似的条件下，通过反复的相互作用而产生的。显然，一些相似的情景可能有许多不同的结果，也会有许多不同的经验，而经验法不包括那些相似情景下有不同经验的情形。

就像你所看见的那样，人们在推理过程中所使用的每一个推理方法都是由一定的规则或模式构成的。这些规则和模式可以以自动的、机械的方式记录下来，这是人类的这些推理过程的特征。考虑了这些特征，计算机推理才顺利地发展起来。

计算机是如何推理的

人工智能系统设计者使用计算机建立推理系统，它们使用的推理模式与人类用的推理方法和推理机制几乎是一样的。一个“推理”计算机运用了一套自动模式，这些模式模仿了人类认识的一些主要推理方法。

基于规则的推理

人工智能推理系统所使用的规则在很大程度上和我们所用的是一样的。实际上，基于规则的推理（rule-based reasoning），是专家系统使用的最普通的推理方法。计算机就是通过这个过程，以输入值的形式获知问题域的特征。然后，运用知识库里的规则，有系统地改变问题域的状态，直到我们期望的状况出现。每个规则包含两个部分：⑴ 状态改变的操作、结果或结论；⑵ 操作的前提条件。规则以下面的 IF-THEN 的形式出现：

IF 前提或条件　　THEN 操作、结论

如果前提条件在逻辑上是正确的，那么后面的操作就可以执行，可以说这个规则可用。如果前提条件不满足，那么这个操作就会被忽视，而进行下一个规则判断。这个推理

过程将一直进行下去,直到问题域与期望的状况匹配,或者知识库里已经没有可用的规则。

许多类型的知识可以用一个正式的规则群来表示。比如,推论性知识 (inferential knowledge) 也可以添加到规则里面。这种类型的知识一般都有这样的特征 :只要满足一个或多个条件,我们就能由此得到结论。

IF	前提条件	IF	这条路正在维修中
THEN	结论	THEN	我们要小心行驶

能转换成规则群的第二种知识形式是程序性知识 (procedural knowledge)。这里,条件以事物的状态的形式表现出来 ;而操作这是状态满足时所采取的动作。

IF	状态	IF	车速大于 55 千米每小时
THEN	动作	THEN	开出超速罚单

程序性知识规则的动作部分可以用其他需要规则进一步证实的状态来代替。

IF	状态	IF	车速大于 55 千米每小时
THEN	新规则	THEN	检查路面状况
IF	状态	IF	这条路正在维修中
THEN	动作	THEN	双倍超速违规

第三种可转换的知识是陈述性知识 (declarative knowledge)。我们根据先行词和结果从句构造了这种表达形式。

IF	先行词	IF	构成有罪性超速
THEN	结果从句	THEN	驾驶员必须支付罚金

除了与人类的推理方式有些类似之外,人工智能系统中的基于规则的推理没有任何特别的优势,而且,基于规则的推理过分形式化。最终,随着知识库生成的规则的数量不断地增加,推理的效率将不断下降,除非运用一些标准结构 (比如规则群) 来改变这种状况。另外,也很难用 IF-THEN 的形式把目标对象的静态特征表示出来。人工智能系统必须通过运用类似于人类分类认知活动的机制,表示上述目标对象的静态特征。

框架

框架 (frame) 的概念最早是由 Marvin Minsky 于 1975 年提出来的。他指出,框架是语义网的知识表示形式的外延。在语义网中,用节点来表示概念和实体,用连接各节点的弧线来表示各节点之间的关系。图 8-2 举例说明了语义网的概念。

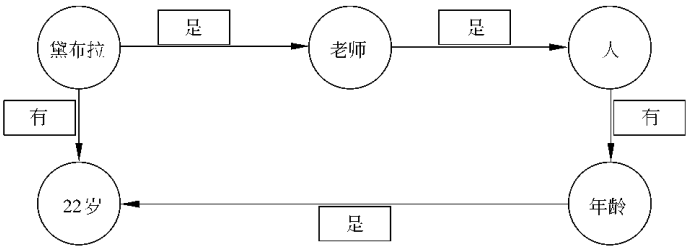


图 8-2 语义网的概念

语义网扩展到框架，建立了一个面向对象的视图。框架中，知识被分割成了一些具有独立属性的离散结构体。这些称为“槽（slot）”的离散结构体可以看作类似数据库的数据域或我们所说的类别。总起来说，槽可以用来表示框架中对象的类别。

> 框架是一种数据结构，用来表示某种状态。每个框架里面都存有不同种类的信息。其中，有些信息是关于如何使用这个框架的；有些信息是关于如何处理期望数据不符合要求等意外事情的。（Minsky，1975）

框架适合描述各个类型中典型的固定形式的状态、对象。每个类型都有一定的特征属性，该框架的各个成员都可以继承这些属性。图 8-3 中给出的是子框架继承可交换证券的属性的例子。

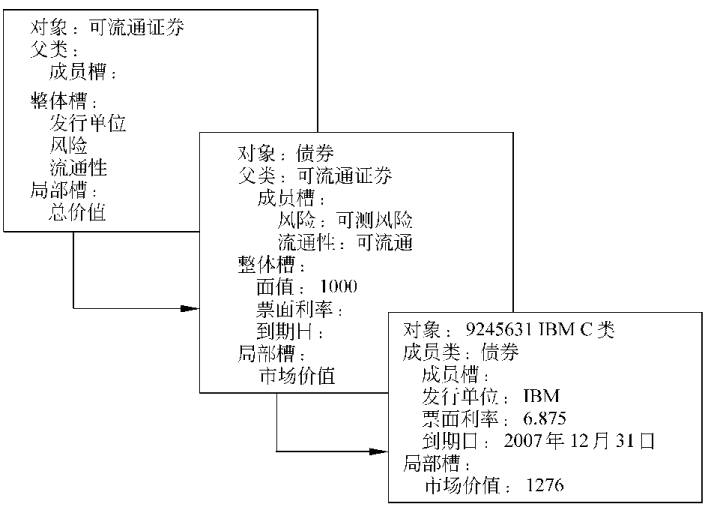


图 8-3　框架的继承性

另外，除了描述框架各方面属性的槽之外，还有其他的一些重要的组成部分。比如，每个槽里都含有一个或多个侧面，这些侧面可以进一步说明槽的值或特定程序。一些典型的侧面包含了基数（单值或多值的），数值类型（数字的、文本的、逻辑的，等等），以及取值范围。另外，框架体系中还有一个重要组成部分：缺省值侧面。如果有些属性值没有直接填入槽里，就用缺省值直接填入。例如，人身保险框架中，可能有一个“发行日期”，如果框架保存的时候，这个槽还是空的，当前日期就会作为缺省值填入槽中。有了缺省值侧面，系统就能保证有足够充分的信息来支持决策。

另外一种很重要的程序是附加过程。它是在槽建立、修改或者存取的时候，都会触发的一小段后台程序。一个框架体系中一般有下面三个附加过程程序：

1. If modified。一旦过程侧面所在的槽值被修改，就会触发侧面的后台程序。这个程序的功能一般是纪录槽值改变的日期、时间和修改槽值的用户的身份。
2. If deleted。当槽值被删除时，将触发这个程序。功能大致和上面的后台程序一样，另外还会启动提示用户注意的友好的信息框。
3. If necessary。当槽值被引用，但是本身还没有赋值时，系统自动启动附加程序计算

槽值。

框架的概念中还需要注意两个对象描述类型 类和实例。正如上面所讨论的那样,框架描述的对象包括可以从上层框架中继承的槽值。这里有两种构造框架的层次结构的方式 :子类连接 (IS-A) 和成员连接 (INSTANCE-OF) 。成员连接可以看成是分级机制 ;子类连接则可以看作分类机制。

根据一些概括性的特征,一些属性可以进一步地划分成更小的信息类型。如果属性能被该类的任何成员继承,那么该属性一定被存储在一个成员槽里。框架所描述的特殊对象的属性存储在整体槽里。最后,任何与特殊对象无关、且不能被继承的属性都存放在局部槽。

我们很容易发现,框架的概念和人类的分类推理方法很相似。由于使用规则推理,框架也有它的局限性。这里面最值得注意的是,框架是描述面向对象信息的表示法,而对存储依情景而定的自然信息没有多大的帮助。实例方面,基于人工智能的系统则使用了模拟人类经验推理方法的机制。

基于案例的推理

基于案例的推理 (case-based reasoning) 依赖于一个前提,就是运用相似性、经验性推理方法去学习并解决问题。这种思想类似于匹配,即用相似问题的解决方案解决现有的问题。基于案例的推理方法有两个主要步骤 :(1) 从案例库中找到与现有问题类似的案例, (2) 修改检索到的符合当前问题的案例的解决方案。图 8-4 是基于案例推理结构的推理流程图。

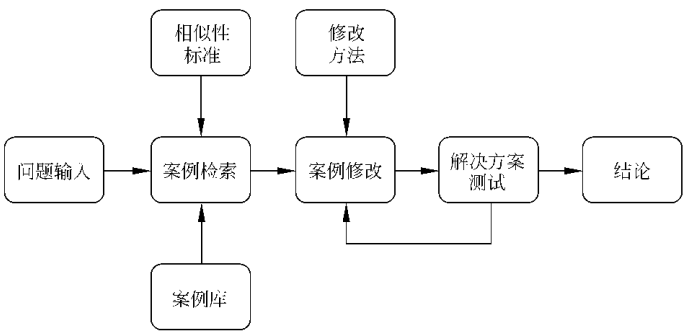


图 8-4 基于案例的推理流程

推理过程中最关键的步骤是从案例库中定位检索相应的案例。这要通过一个复杂的案例索引系统来完成,这个索引系统用来高效率地检索与当前问题相适应的案例。我们有一套相似性标准来衡量检索到的案例与当前问题的相似程度,这样,每个可能的案例都要与当前问题比较。一旦检索到相应的案例,系统将分析该案例包含的解决方案,并修改这个方案以适应新的情况。这个修改过程主要是要修改一些参数。案例中有许多参数都是关于旧问题的,要修改过来以符合当前的新情况。最后,新的解决方案要经过检验,如果检验成功,将产生新的案例并加入案例库 如果没有通过检验,要进行新一轮的修订,或者用一个新的案例库检索。

在与基于优先权推理结合使用时，基于案例的推理在认知领域有非常大的作用。基于案例的推理系统在医疗诊断、审计、司法和索赔结算等领域有强大的发展空间。

模式识别

模式识别（pattern recognition）是人工智能使用的主要机制，它与人类的期望法推理过程很相似。模式识别包括图像识别和语音识别。通过模式识别系统，计算机能够模仿人的某些智能行为。现在，模式识别系统已经具备了适应其所处的客观环境的能力。如果计算机的概念仅仅局限于键盘、鼠标或者光标移动球，那么计算机的感知能力就会限制在一定的范围内。可是，如果我们给计算机配置一些硬件、软件，使它能检测声音，感知事物的模式或形状，那么，模拟智能行为就不是可望不可及的事情了，我们也由此进入了一个新的研究领域。先进的模式识别系统能够识别人的声音、指纹甚至能通过一些图片或者录像片断确定一个人的身份。早期的模式匹配机制受到运算速度和可用资源的限制，而新的编程软件摆脱了这些资源的约束，提高了模式识别的可靠性和实用性。

Rete 算法　Rete 算法就最近发展起来的一种模式识别算法。它能不受任何资源限制地解决包含几千个工作存储区元素和规则的复杂的模式匹配问题，而且不会失真。这种聪明的算法充分利用了工作存储区的内容不会有太大改变的特性，每激活一个规则，工作存储区的内容都只会在上次激活的基础上作少许改变。Rete 算法能计算出在上一轮循环中没有匹配的规则、上一循环中匹配但在下一循环中不匹配的规则、以及上一轮中不匹配但在下一轮循环中很可能匹配的规则。有一种内部表示法能表示工作存储区的每一条规则的状态，通过运用这种内部表示法，以及上面的算法，Rete 算法不会在每次模式匹配循环中都检验所有的规则。而且，这种内部表示法是匹配循环的基础，Rete 算法要依靠这种方法来不断地重复匹配循环，直到模式匹配成功。

现在，模式识别系统在许多领域都有应用。比如，声控系统、语音识别、图片分析、模式匹配、信号验证系统等。人工智能的研究将对未来专家系统的发展产生深远的影响。

人工智能的其他形式

在现在广泛流行的基于人工智能的系统软件中，专家系统仍然处在前沿的地位。尽管如此，人工智能还有其他一些热门的研究方向。它们不像专家系统那样发展得很成熟，但是，它们也有潜力为智能系统的发展做出卓越的贡献。我们将简单地介绍人工智能系统的其他发展方向。

机器学习

机器学习（machine learning）包括基于人工智能的运行机制，这种机制让计算机学习通过实例和模拟去学习经验。机器学习要么改变了计算机的程式化行为，要么修改现有的按计算机程序执行的知识。机器学习有两个知名的研究领域，一个是神经网络（neural network），另一个是遗传算法（genetic algorithm）。

神经网络　神经网络是人工智能系统研究的一个分支，主要致力于用计算机机制模拟人脑的学习过程。人脑是由数千个神经网组成的，大量互相紧密连接起来的神经元组

成了神经网。神经元以电压的形式发送信号,一个神经元的输出信号可以作为上千个其他的神经元的输入信号。也就是说,每个神经元进行离散的计算,计算结果可以通过神经通道传递给上千个其他的神经元。这些信号要么抑制其他神经元的活动,要么激活其他的神经元的活动。这个过程导致了某些神经元的电压的积累。人类的神经系统就是通过这种反复的积累、释放电压的过程,学习并适应环境的变化。

和人脑的神经网络的运行方式差不多,基于计算机的神经网络也构造了许多称为"神经元"的计算单元,这些紧密相连的神经元将从一系列的事例中寻找特定的模式,并通过改变连接各个神经元权值来学习。在这种类似人脑的精确的输出模式中,这些权值会不断的自我修改。大量相关的实例将作为神经网络的训练样本,神经网络将判断由这些实例值得到的输出值,依据计算的输出值与实际值之间的差别做出反馈,并依靠反馈来修改权值。这就是基于计算机的神经网络的训练过程。

遗传算法 20 世纪 40 年代,John Holland 在麻省理工学院提出了遗传算法的概念。这个时期,遗传算法 (GA) 运用到了计算机系统的一套适应性程序中,这个系统是基于达尔文的自然选择和优胜劣汰的理论的。达尔文认为,为了获取更大的支配地位,物种必须主动地适应环境的改变。遵从这个理论,遗传算法依靠各种结合不断地"繁殖"自己,目的就是为了找到一个结合体,它能获得比自己的前辈更好的适应性。遗传算法有一个主要优势,就是在使用遗传算法的时候,我们不需要知道正确的答案和合理的解决方案,就能审查复杂的问题。遗传算法能和神经网络、专家系统结合使用来解决实际问题,现在已经能解决制作图表、设计、营销等领域的问题了。我们将在第 10 章详细地探究遗传算法、神经网络及其应用软件。

自动化程序设计

人工智能的这个研究领域主要包括研究程序的自动生成机制。运用这种机制,计算机可以自动生成一段程序来完成指定的任务。用这种方法,一个不是程序员的人,即使不知道怎么写程序,也能向系统描述任务的性质以及程序应具有的特性和行为。自动化程序设计 (automatic programming) 软件将这些描述作为输入信息,并生成合乎要求的程序。现在,自动化程序设计经常被用在管理领域,能生成一些用来自动生成决策支持报表的程序。

人工生命

人工生命 (artificial life) 是描述用人工智能研究"自然生命"的术语,它试图通过计算机系统再现生物现象。生命科学传统的分析方法是依靠"拆分"活着的生命体来研究其运行机制,与此相反,人工生命试图"装配"类似生命机理的系统。人工生命的潜力不仅在于生物现象理解理论的进步,而且在于把生物理论扩展到许多其他科学领域的实际应用中,比如计算机硬件软件设计、机器人技术、太空飞船设计、医药等其他的工程领域。

人工生命、自动化程序设计、神经网络等其他的一些基于人工智能的机制,一定会成为将来决策支持系统的非常重要的贡献者。当这些机制结合起来使用的时候,它们将成为各种强大而神奇的决策支持系统的基础。

8.3　专家系统的概念和结构

专家系统的基本结构与第 1 章曾简略提到的决策支持系统的遗传结构是一致的。知识库是对专家系统的特定问题域的详细描述，相对于用来解决横跨多领域问题的解题算法而言，它是一个特殊的、独立的部分。专家系统与决策支持系统的主要不同点是，专家系统受限于从应用专家那里获取的知识。在这一节，我们将着重介绍专家系统组成要素，它们详细描述了一个专家系统。图 8-5 描述了专家系统的基本结构。

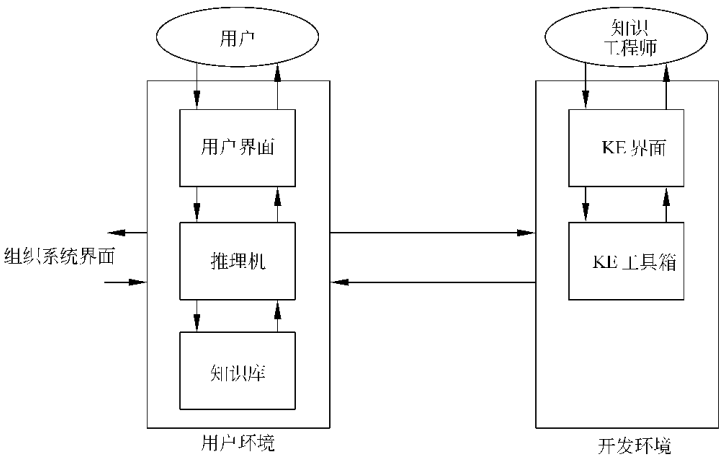


图 8-5　专家系统的一般结构

用户界面

现代化的专家系统软件生成程序的出现让我们注意到用户界面的相对重要性。一项关于用户输入、输出和控制程序代码的分析显示，超过 45% 的程序代码直接与专家系统的用户界面有关。另外，建立一个新的专家系统需要进行的前期分析和需求分析，其中绝大部分与人机对话界面的功能相关。简而言之，一个专家系统的成功与否，经常归结于它的界面的品质和功能。

用户界面设计的焦点主要在于人所关注的方面，比如系统使用简单、简洁、可靠性高、出错率低、减少疲劳和疲劳性伤害等等。当专家系统设计者优化人们的关注点时，会遇到存储能力和硬件设备限制的问题，他们则会采取一个平衡方案。Berry 和 Broadbend 于 1987 年得到了一些关于设计高质量的专家系统用户界面的结论。表 8-1 简略地叙述了这些结论的要点。

用户界面的设计应该尽可能地考虑交互方法的适应性。这些方法包括输入、控制和询问机制，用户可以根据自己的个人偏好选择不同样式的交互方法。表 8-2 列举了专家系统界面中应包含的一些有用的交互方法。

表 8-1 设计好的专家系统用户界面的要点

- 在用户界面的设计中,用户应该是同样重要或者有支配权的合作者
- 专家系统的对话框应该足够灵活,让用户主动提出信息并容许轻微的改变
- 必须有高质量的、详细的解释说明功能
- 自然语言界面并不适合使用在专家系统中
- 专家系统的对话框对图示应该适当地比对文字更为重视

表 8-2 专家系统界面的交互方法

- 触摸屏
- 用户自定义热键
- 功能键
- 鼠标
- 光标运动球
- 触摸垫板
- 光笔
- 语音触发控制
- 光标控制
- 菜单
- 图像映射

知识库

专家系统的知识库 (knowledge base) 包含的是特殊领域的知识,这些知识是从领域专家的设计实例中收集来的,包括领域专家在解决领域相关问题的过程中所使用的典型知识,这些领域问题有:对象描述和关联、解决问题的操作、约束问题、启发性知识和不确定性问题等。专家系统的成功与否依赖于它的知识库的完善和准确性的程度。如果知识库中存在不完善、不一致或者不准确的信息,无论专家系统的其他操作模块多么有效率,专家系统将只能快速地得到一个不完善或者不正确的解。当我们讨论专家系统知识库的内容的时候,我们一定要把数据和知识的概念区分开来。如果我们准备让一个职员或自动化过程去收集想得到的资料,这些资料就是刚才提到的数据。可是,如果我们指望一个专家来提供想要的资料,我们在这里谈论的就是知识。换句话说,描述事实的数据的集合就是数据库;专家的论据和启发性知识的收集就是知识库。相对于数据库,知识库包含了更加抽象的信息。从上面的讨论来看,知识库容纳了规则、框架、语义网、剧本、案例和模式匹配等信息,这些都是在专家系统知识领域解决问题的必要信息。

推理机

如果我们把知识库作为整套系统的大脑,那推理机 (inference engine,IE) 就是肌肉。专家系统的推理是在推理机里进行的,而且也在这里输入知识,并经过推理得到结论。推理机是基于规则和事实来执行演绎和推理的。另外,推理机也具有执行基于概率推理或模式匹配的模糊推理的能力。推理机的基本过程叫做一个控制循环,一个推理控制循环可以分成三步:(1) 用给定的事实匹配规则,(2) 选择下一个要执行的规则,然后执行第三步,(3) 执行规则,将推出的事实加入到工作存储器中。

推理机的基本工作原理是基于 modus ponens 演绎推理规则的,即:如果 A 是真的,

A 蕴含 B (A→B) 也是真的,那么 B 也是真的。考察下面的例子:

1. 当马克知道有地方卖衣服打折的时候,他总是会去买衣服。
2. 马克了解到商场里有打折卖的衣服。
3. 因此,马克会买衣服。

与演绎推理法相对的一种规则是 modus tollens,它规定:如果 A 蕴含 B (A→B) 是真的,而且 B 是假的,那么我们可以总结出 A 也是假的。例如

1. 如果是晴天,我们就去游泳。
2. 我们没有去游泳。
3. 因此,一定不是晴天。

一个推理机用两种基本的方法来实施演绎推理的两种规则,并得出正确的结论。这两种基本方法是:推理链和分解法。

推理链

推理链 (chaining) 是一种很简单的推理过程,大多数推理机用它来生成一个推理的链条。运用这种方法,规则库可以以回归的形式组织起来,这样由一条规则推出来的事实就可以作为下一个规则的前提条件。根据搜索方向的不同,推理机中有两种典型的推理链:正向推理和反向推理。

我们先来研究一下这两种推理过程是如何工作的。假定下面是知识库里的一些规则:

步骤	规则库	操作域
(1)	R1 IF A and B THEN D	A,B
(2)	R2 IF B THEN C	D
(3)	R3 IF C and D THEN E	C,D
(4)		E

在正向推理链中,推理机首先初始化工作空间的内容,然后通过一系列的推理循环,向目标结论推进。在操作域中有值为真的一些事实,在每一次循环过程中,推理机会在知识库中搜寻以这些事实为前提的规则 (步骤 1)。一旦找到了这个规则,则认为该规则的前提条件均已满足,规则可以执行,而且将推出的事实添加到操作域中 (步骤 2)。匹配—选择—执行的推理循环将重复执行,另外一个事实加入到了操作域中 (步骤 3)。经过匹配循环,当知识库中的备选规则都用尽了的时候,推理过程将终止,并且知识库中包含了希望得出的结论。由于过程蕴含了从数据到最终目标的运动,这种推理方法也被称为数据驱动。

第二种链式推理方法是反向推理。推理机从最终目标开始,反向推出一些为得到目标而必须满足的数据,这些数据都在操作域中,从而推理过程结束了。我们可以把反向推理看作一种假设检验:先假设结论是正确的,然后收集数据支持这个假设。我们用前面的

那个知识库：

步骤	规则库	操作域
(1)	R3 IF C and D THEN E	C,D
(2)	R1 IF A and B THEN D	A,B
(3)	R2 IF B THEN C	

用反向推理,推理机从目标值 E 开始,然后在知识库中收集支持目标的证据。推理机发现 R3 规则是以目标值 E 为结论,选中这个规则 (步骤 1)。R3 被选中之后,将它的前提条件 C 和 D 放入操作域 (步骤 2)。另一个推理循环将先寻找一个以 C 为结论的规则,然后寻找一个以 D 为结论的规则。通过规则 R1,推理机要证明 D 是正确的,那 A 和 B 也必须是正确的。查看操作域,推理机发现 A 和 B 都是事实 (步骤 3),同样可以证明 C 是正确的 (通过规则 R2)。由于这种方法从最终目标开始,通过寻找子目标的支持数据反向搜索,所以反向推理也被称为目标驱动的搜索。

正向推理或反向推理的选用取决于两个主要因素:专家推理的模式和效率。专家推理的模式是专家在特殊领域用来推理的方式,它限制了推理方法的使用。如果专家通常用问题分解的手法作为搜索方法,那反向推理将很适用。可是,如果有大量的目标要与输入值或者已知事实匹配,运行效率将成为主要问题,那么正向推理就比较适合了。不管使用哪种方法,无论怎样,正向推理或者反向推理都要通过知识库,根据一定的逻辑路径得到满意的结论。表 8-3 描述了两种方法在确定从一地坐飞机到另一地的最佳路径方面的用途。

表 8-3 正向推理和反向推理举例

- 情景：你希望坐飞机从华盛顿到圣地亚哥。不幸的是所有的直航飞机的座位都被预定一空,所以如果你还想飞到圣地亚哥,你就必须乘坐转接班机。
- 反向推理：你会检查到圣地亚哥的航班,看看起点都是哪些城市。然后,搜索到达这些城市的航班,像这样一直向前,直到你找到了华盛顿。
- 正向推理：你会检查华盛顿起飞的航班,找到它们的目的城市。然后,搜索从这些城市起飞的航班,像这样一直往前,直到你找到圣地亚哥。

黑板模型

黑板 (blackboard) 给专家系统用户提供了一个工作存储区,用来作为专家系统各种模块的操作域。它可以看成是计算中心的电子便签簿或记事本,在问题解决过程中,它可以用来存储中间假设和结果。通过黑板系统,专家系统能完成备选方案之间的沟通,或要求额外的输入。例如,急救室里,在对一个外伤病人的诊断中,专家系统的知识库中包含了与急诊室外伤处理相关的所有的规则和推理方法。主治医生运用这个专家系统,并将病人的症状和生命指标输入到专家系统中,这些初始信息将存放在黑板中。规则选择和假设检验之后,专家系统可能会建议进行一些补充的检验。这些信息需求也临时存放在

黑板中。当输入了补充检验的结果的时候，推理过程将一直进行下去，直到得到一个结论。一旦系统完成了工作，黑板存储区将被释放，准备下一次解决问题。

8.4　专家系统的设计与建立

专家系统的骨架系统

20 世纪 90 年代，人工智能研究的惟一挑战或许就是用于快速、低成本地建立专家系统的通用工具的发展。我们开发的专家系统必须有一套独立于问题领域的推理机制，并允许一个通用系统使用知识库中包含的所有特定问题的知识。在开发这种系统的过程中，我们遇到了挑战。我们称这些类系统为专家系统的骨架系统（expert system shell）。

EMYCIN 是最早的一套成功的骨架系统，它是一个面向特定诊断问题的通用问题解决程序。在同一时期，名为 OPS5 和 ROSIE 的两个骨架系统也成功面世。这两个骨架系统都用正向推理作为它们的主要搜索方法，并且允许开发人员建立一套描述推理流程的程序说明。用现在的标准看来，它们很笨拙。可是，这些骨架系统初步运用 LISP 或 PROLOG 设计专家系统，这是专家系统设计方面的一大进步。

现代的骨架系统一般包括两个主要模块：规则库生成器和推理机。开发人员可以运用模型来协助初始化知识库，以及作为后期推出结论的工具。近期，出现了完整的专家系统开发环境的概念。在这些系统中，有大量多样的知识管理工具，比如超文本、图解、电子公告牌等。这些工具与基本的基于规则方法结合起来建立专家系统，建立的这个系统超出了原有的传统专家系统的范围。经过每一次精炼，人工智能都会向真正独立于语境解题器的最终目标前进一步。

建立一个专家系统

应该确信，专家系统在管理软件领域的发展空间实质上是无穷大的。即使如此，专家系统的目标特性要求专家系统软件的领域和界限必须在系统设计之前确定清楚。解决这个问题的一种方法就是：确定专家系统要完成的任务的类型。表 8-4 给出了专家系统任务的大致类型。

表 8-4　专家系统任务类型

- 解释：用传感器信息推断情况的种类和含义
- 预测：根据给定情况的历史信息进行预测，并提出当前最可能的结果
- 诊断：根据观察结果和对信息的分析，推断错误和故障所在
- 计划编制：设计方案，并计划完成既定目标
- 设计：为满足特定需求而制定项目计划书
- 处方：提出补救方法和解决问题的方案
- 监控：将信号和观测资料与期望结果相比
- 控制：管理全面的系统行为
- 指导：判断、建议并引导用户行为

20 世纪 70 年代以来,人们设计了很多与智能计算机系统(比如专家系统)的设计、实施相关的程序和方法。Gerrity (1971) 提出,这些程序的关键问题是方法论的问题,而不是技术的问题。Getrrity 的设计方法的本质是在强调两种模型,一个是基于可确定系统范围的叙述性系统原型,另外一个是基于规定系统目标的规范性系统原型,然后试图减少两种原型之间的差异,直到一个满意实用的设计完成。

1978 年,Keen 和 Scott Morton 扩展了 Gerrity 的方法。他们提出,任何智能系统设计的第一步必须是社会思潮的转变和对专家系统的投入的确立。一旦实现了第一步,项目成功的关键就是专家系统问题的鉴定,并把握建立专家系统的机会。

很多地方都适合建立专家系统,专家系统可以覆盖整个组织的任何角落。管理者关键是要培养把握这种机会的能力。管理者不仅要寻找那些适合专家系统补充稀有的专家资源的领域,还要听取专家系统通过提供必要协助来扮演专家的任务的情况。简而言之,在任何人类专家任务繁重、或专家力量不足、或聘请专家花费昂贵、或很难请来专家的领域,都有建立专家系统的机会。表 8-5 列举了存在专家系统机会的几种状况。

表 8-5 存在专家系统机会的几种状况

- 需要对难处理的局面或分歧的分析 (审计、解决纠纷等等)
- 需要理解特定情况的本质
- 需要预测当前或未来事件的结果
- 需要管理一个特殊的活动或进程
- 需要提出一套解决方案或行动计划
- 需要评价事件或步骤的优先级

设计前活动

专家系统的设计实施的过程不同于决策支持系统或者其他的信息系统的设计实施。结合第 13 章和第 14 章介绍的决策支持系统的开发,我们同样要进行普通的分析、设计并实施各个模块等工作。然而,还有一些活动是要优先于普通设计活动执行的,这些是专家系统开发所特有的。

寻找专家 要成功地设计一个专家系统,最重要的可能是选择专家或专家群体。这些专家所提供的知识将输入到知识库中。理论上,没有专家就没有专家系统。我们要清楚地告诉专家,专家系统开发需要他们投入很多的时间。这一点,非常重要。专家们必须乐于参与专家系统开发的整个过程,并需要在早期开发过程中花费大量的时间和精力。另外一点也很重要,就是专家要充分地理解这个系统在组织的生产、业绩和质量方面的重要价值。同时,系统应该定位于增强专家的作用,而不是取代专家。如果一个专家从要建立的专家系统那里感受到了威胁,他就不会贡献有用的信息给专家系统。

开发团队 正如上面叙述的那样,专家系统的开发和其他信息系统的开发有许多共通的地方,但是,专家系统开发中还有一些特有的问题要处理。专家系统开发团队的选择

也非常重要,他们负责系统的开发和建立,必须非常熟悉专家系统这一领域。开发人员应该对要建立专家系统的知识领域也应该有一定的了解。在许多系统开发实例中,开发团队的一个或更多的成员在建立知识库的过程中,会扮演专家的角色。通常,在开发过程的早期,开发团队会分成更小的小组,小组会着重系统的特定方面的工作。开发团队的一些成员要建立规则库,另外一些则着重用户界面的工作,在专家系统设计过程中,还有一些人将扮演知识工程师这一特殊角色。知识工程师直接与专家一起工作,负责提炼和组织知识主体,这些知识主体必须包含在知识库中。知识聚集的过程不但复杂、单调,而且要求开发者有一些特殊技巧。我们将在下一章探究与知识获取和知识工程的方法相关的问题。

开发工具　在传统的信息系统设计工程的早期开发阶段,通常不需要选择软件开发工具或语言平台。最初的焦点应该是描述系统功能的逻辑模型的开发。开发语言是实体解决方案的一部分,主要解决我们如何实现系统的问题。无论如何,在专家系统设计实例中,较早的确定开发工具是非常重要的。开发过程中,我们要不断地重复系统原型开发、评审、精炼等工作,这一重复性质要求我们在系统设计前选择好开发工具。开发工具必须能完善开发团队的技能,而且要为问题域提供适当的功能。此外,理想的工具必须有很强的适应性,就是它不仅要支持当前专家系统开发项目,更要能支持许多其他专家系统开发项目。保持开发工具的通用性和适应性,开发团队就能利用开发工具提高自身的开发能力。

硬件选择　和大多数其他系统的开发不同,专家系统硬件平台的选择应该在设计前完成。选中的平台不仅支持最终的专家系统产品,并考虑到之后的升级,而且最理想莫过于能支持专家系统的开发工具。这样,系统原型将和最终产品完成相同的工作,并且开发后,硬件平台的改变不会招致系统功能或性能的任何改变。

8.5　专家系统的利益评估

专家系统的成功开发和实施将带来许多的利益。它们所获得的成功是与大量的努力分不开的,而且只有当用户在系统功能方面的期望得到满足之后,我们才能获得这些利益。这里有一些标准来帮助用户评价一个新的专家系统的质量和功能。表 8-6 概述了八个基本的评价标准。

假如系统完全符合表 8-6 提出的标准,而且被系统用户一致认同,那组织内部将产生巨大的运作收益。

表 8-6　专家系统的评价标准

• 系统响应性好,且易于使用	• 解释系统应该清晰且友好
• 系统设计和功能应该与当前的分析标准一致	• 知识库所包含的应该是公认的知识
• 系统能使不完备知识发挥作用	• 系统应该不受任何地理限制
• 用户能管理咨询程序	• 系统应当考虑与其他系统的兼容性

专家系统利益

增加制定决策的及时性

有时候,特定的关键信息的可得性通常与需要它的急迫性相反,也就是你越想得到的信息越可能得不到。虽然你有相对丰富的知识领域的专家资源,他们很可能和你在同一栋楼里,但是当你需要一个专家的时候,你很难顺利地找到他们。专家系统也可以帮助那些需要专家意见的知识领域,提供专家意见。与人类专家不同,专家系统可以实现一天运转 24 小时,它永远也不会疲劳、生病。此外,当多个管理人员遇到一些相关、但又截然不同的问题,都需要专家意见,专家系统则可依托其覆盖的广大知识领域,同时为这些管理人员提供服务。

提高组织内专家的生产率

由于知识领域的特殊性,组织对常驻专家的工作时间有大量的需求。因为对组织专家的需求没有区分优先级别,组织内部经常会出现下面的情况 :有一个非常紧迫的问题需要专家注意,而此时专家忙于解决组织其他部门并不是很紧迫的问题。专家系统的引入可以为那些不是很紧迫的问题提供有价值的处理意见。这样可以使常驻专家集中解决那些更有挑战的组织性问题,从而提高了常驻专家的生产率。此外,这减轻了专家日常工作的负担,使他们有时间从事那些与组织利益相关的更具创造力的活动。最后,专家系统具有复制的能力,能安装在遍布全球的不同的分支机构。这样,在决策活动中,所有的管理人员都可以从专家那里受益。这将是一种很划算的方法。

提高决策的一致性

与专家系统相关的另一个预期收获是决策的输入信息的一致性增加,以及由此带来的决策结果一致性的提高。专家系统不会像人类专家那样存在认知的偏差和局限性,因此,也就不会在记忆和注意细节信息时发生错误。同样,在对事态分析中,专家系统不会像人类专家那样掺杂任何形式的政治动机。尽管我们在分析时还是要考虑政治动机,但这些应该由决策者把握,并且要从引导最终决策的基本信息中剥离出来。

提高理解能力和解释能力

专家系统在解决问题的过程中有一条明确的推理路线。和人类专家采取的方式一样,专家系统将给用户提供对这一推理路线的解释。在分析出结论之后,这种功能允许决策者评价专家的思考过程,并对基本原理的正确性做出判断。时常,决策者使用这些解释来增强自己对专家思维过程的理解能力,理解他们如何从特定的初始事实中得到一个独特的结论。随着决策者对推理过程细节的理解能力越来越强,他们不仅提高了对手头的决策的理解能力,还提高了对整个领域知识的理解能力。

改善不确定性管理

在分析问题时,专家们必须有能力妥善处理不确定性问题。在制定决策的整个环境

中，不确定性问题是无处不在的，并且专家在其提出的解决问题的建议中，必须衡量出建议的不确定性（也就是可信度）。专家系统也能够处理结果的不确定性，这也是专家系统区别于其他传统的信息系统软件的特性。专家系统处理不确定性问题的方法是模糊逻辑。我们会在第 10 章集中讲到神经网络，同时我们会探究智能系统的模糊逻辑领域。

知识格式化

专家系统在特定的知识领域要使用一些推理方法和规则。组织可以启动知识代码化的方法，将这些方法和规则融入专家系统。正是这种获取知识的方法允许组织对解决问题的步骤进行详细地检查，并且有助于了如何解决组织问题。专家系统开发时获取的知识是长期有效的，即使原有的专家走了，新的管理者和新一代的专家同样可以使用专家系统，并从中获得利益。

专家系统存在的问题和限制

尽管专家系统已经广泛地被人们接受了，而且专家系统拥有使组织受益的潜力，但是专家系统还是有它公认的局限性，这些局限性亟待解决。表 8-7 列举了专家系统的设计、实施和使用过程中必须克服的一些难题。

表 8-7　专家系统的局限性

- 所需的知识并非总能得到
- 专家会利用常识。对常识编程并不现实
- 专家意见是很难提炼并转化成代码的
- 专家比专家系统在辨认知识领域外的问题时速度更快、更有效率
- 专家系统不能消除用户的感知局限
- 一个专家系统只具有解决狭小领域问题的功能
- 专家的表达能力也可能有限，表述的知识也不容易理解
- 人类专家能够自然地适应环境，而专家系统显然需要更新才能适应环境
- 与人类专家相比，专家系统缺乏灵感

除了表 8-7 中列出的问题之外，我们必须清楚，专家系统也并非是完全可靠的。专家系统与人类差不多，也经常犯错误。因为专家系统的规则库只不过是依靠专家的知识和推理方法组合起来的模型，也因为这些专家都会犯错误，所以专家系统必然可能得出不正确的结果。事实上，由于我们是在建造知识模型，而不是切实地反映知识的内涵，所以，专家系统比人类专家更容易出错。XCON 系统是由数字设备公司开发的配置复杂计算机系统的专家系统，可以用来协助销售代表投标，它被公认为产业界最成功的专家系统。XCON 经常被人作为功能强大的、成熟的专家系统的典范。尽管 XCON 很有名气，也很成熟，但是，它还不能完成给它配置的任务的 2%。这里，真正的问题是，一个组织能否很放心地将解决复杂问题的重任交给一个容易出错的计算机程序。随着技术的发展，我们期望专家系统的出错率会下降，推理方法也会得到改进。时间可以证明。

还有一个确定专家系统的局限性的方法。专家系统在完成作业时会遇到一些特殊的困难，我们分析每一个遇到特殊困难的作业，这样就可以了解专家系统的一些局限性。表 8-8 针对表 8-4 提出的作业类型，列出了与每个作业类型相关的局限。

表 8-8 专家系统普通作业类型存在的局限性

作 业	问 题
• 解释	信息可能丢失或存在噪音
	信息不准确
• 预测	必须考虑偶然性和不确定性
• 诊断	多种症状会混淆诊断
• 计划编制	复杂情景下会存在多种选择
• 设计	子计划中存在冲突约束并相互影响
• 处方	可能存在多种问题
• 监控	特定的语境中会有错误条件和不切实际的期望值
• 控制	经常需要常识性解释

专家系统的通用技术

自专家系统出现以来,系统数量和纵向市场应用的多样性方面都有了巨大的发展。在本章的最后,列出了一些很受欢迎的、成功的商用专家系统,并作了简要的介绍。尽管列表中的系统的覆盖面不是很广,但列出的系统在技术上都很具代表性,会使读者在了解整个智能系统的应用方面有更高的起点。

8.6 小 结

本章阐明了专家系统以及建立专家系统的主要方法——人工智能的概念。我们着重介绍了专家系统和人工智能这两个学科在解决问题和制定决策方面的重要性。

通过讨论人类的推理方法,我们找到了建立计算机推理系统的有效方法。并且,我们阐明了专家系统的概念和结构。我们也着重介绍了建立专家系统的设计前活动。从我们着重介绍的焦点问题中,我们增加了对专家系统局限性的了解,以及学会了如何评价一个专家系统。在下一章,我们将介绍如何获取专家知识,以及将这些知识转入专家系统的方法。

复习题

1. 详细说明下面每条术语的概念:
   专家系统
   人工智能
   侧面
   附加过程
   机器学习
   神经网络

专家系统骨架系统
黑板系统
人工生命

2. 什么是基于规则的推理？
3. 什么是框架？描述框架是如何运作的。
4. 什么是模式识别？
5. 列举并解释人类推理方法的不同类型。
6. 专家系统和决策支持系统的主要不同点是什么？
7. 描述专家系统的基本结构体系。
8. 专家系统的结构体系中，哪一部分经常对专家系统的成功或失败的贡献最大？
9. 解释在专家系统的设计中，为什么开发工具的选择要先于系统设计？
10. 专家系统能带来哪些利益？
11. 描述专家系统存在的问题和局限性。
12. 论述评价一个专家系统的标准。

讨论题

1. 选择一个运用专家系统技术来解决问题的系统，并描述这个系统所带来的利益和局限性。
2. 你认为，哪种类型的组织最需要用专家系统来协助决策制定和问题解决？专家系统将如何协助？
3. 你正要打算买一辆小轿车。使用基于规则的推理技术，描述你是如何做出决策的。
4. 识别神经网络和遗传算法的不同点，分别举出一些应用了两种机器学习方法的系统实例。
5. 分析专家系统应用软件中用户界面的设计方案。论述你认为这个设计是好或坏的理由。

专家系统技术介绍

DENDRAL（1965—1983）

DENDRAL 是最早的专家系统之一。DENDRAL 最初是用来研究科学推理的机制，以及有机化学领域知识的结构化的。DENDRAL 的另外一个研究重点是使用人工智能方法来增强对哲学的基本问题的理解，以及判定解释性假定的方法。经过众多的化学家、遗传学者以及计算机专家近十年的努力协作，DENDRAL 不仅成为了基于规则推理的专家系统的成功典范，而且是进行分子结构分析的重要工具，在理论和工业研究方面都可以应用。DENDRAL 可以从大量的光谱数据和其他的资源中搜寻示例和信息，这样就可以推知新的或未知的混合物的最可能分子结构。它对人类专家对有机化合物的分类提出了挑

战,并使得大量的文章在化学刊物上面发表。作为学术研究的最新版本的交互结构发生器,GENOA 已经得到了斯坦福大学的商业用途证书。

META-DENDRAL (1970—1976)

META-DENDRAL 是自动格式化新规则的归纳程序,这些新规则是 DENDRAL 用来说明未知化学混合物信息的。使用计划产生测试示例,并通过对现有规则的再发现以及提出新规则,META-DENDRAL 成功地对大量的光谱进行了格式化。尽管 META-DENDRAL不再是一个有效程序,它在学习和发现方面的思想将对新的研究领域有新的贡献。下面是 META-DENRAL 的思想:归纳法可以以启发式搜索的形式实现自动化;为了提高效率,搜索可以分成两步进行——粗搜索和详细搜索;学习必须能善于处理噪声和不完善的信息;同时学习多种概念的情况有时是不可避免的。

MYCIN (1972—1980)

MYCIN 是一个交互程序,它可以诊断某个传染性疾病,进行开药治疗,并能解释它的推理细节。在一个特定的测试中,MYCIN 性能和专业医师不相上下。另外,MYCIN 综合了一些重要的人工智能成果。MYCIN 扩展了知识库必须独立于推理机的观念,并且它基于规则的推理机是建立于反向推理 (目标驱动) 的控制策略上的。因为 MYCIN 定位于医生顾问,它拥有解释推理线路和知识的功能。为了适应制药业的快速发展,MYCIN 的知识库很容易扩张。并且由于医疗诊断存在一定程度的不确定性,MYCIN 的规则综合了一些确定因素来表明一个结论的重要程度 (比如可能性和风险)。尽管医生们在日常诊断中几乎不使用 MYCIN,但是,MYCIN 还是充分地影响了其他的人工智能研究。在 MYCIN 的基础上,像 TEIRESIAS,EMYCIN,PUFF,CENTAUR,VM,GUIDON 和 SACON 等下面要介绍的系统都开发出来了,还有没有介绍的 ONCOCIN 和 ROGET。

TEIRESIAS (1974—1977)

TEIRESIAS 是知识获取程序,它是用来协助领域专家精炼 MYCIN 的知识库的。TEIRESIAS 提出了超水平知识的概念,也就是:通过这种知识,一个程序不但可以直接使用它的知识,还可以检查知识,推理相关知识,引导知识的使用。TEIRESIAS 使诊断中应用的推理链更清楚,而且它可以协助医疗专家修改或者增加知识库。这里的大部分功能都综合到了 EMYCIN 的框架中。TEIRESIAS 在知识库调试程序中引入的适应性和易懂性,已经成为许多专家系统的设计模型。

EMYCIN (1974—1979)

EMYCIN (Essential MYCIN) 是 MYCIN 的核心推理机,并加上知识工程界面。EMYCIN 是一个领域独立的框架,在基于规则推理,大多是解决咨询问题 (例如,诊断或故障排除中遇到的问题) 的专家系统,经常会使用到 EMYCIN。EMYCIN 一直是专家系统软件的范例,它有利于建立专家系统,并且,医疗和非医疗等多种领域都使用过 EMYCIN (医疗领域的产品有 PUFF,非医疗领域的产品有 SACON)。EMYCIN 在美国和美国之外广泛传播,并且,它是美国德州仪器公司的软件——Personal Consultant 的基础。

PUFF (1977—1979)

PUFF 是第一个使用 EMYCIN 的系统。PUFF 应用在对肺病病人的肺部功能检查作分析的领域。它能诊断已发现的严重的肺病,并生成报告,存在病历中。太平洋医疗中心在使用 PUFF 系统,以协助分析肺部功能测试的日常工作。这个领域的规则库开发完成并调试成功后,PUFF 转移到了旧金山太平洋医疗中心的小型机上。已经有一种版本的 PUFF 获得了商业应用的许可。

CENTAUR (1977—1980)

CENTAUR 系统是用来试验的专家系统,它将基于规则的推理方法和基于框架的推理方法结合起来,并用其描述和使用关于药物和医疗诊断策略的知识。为了便于比较,CENTAUR 的应用领域和 PUFF 一样,都是做肺部功能检查分析的。结果是 CENTAUR 做得更好一些,这也证明了这种描述和控制方法的效果。

VM (1977—1981)

VM (呼吸器管理) 程序可以解释重点护理小组的大量的即时信息,并协助医生管理手术后病人需要的呼吸器。尽管 VM 是基于 MYCIN 的体系结构,但是它还是重新设计来支持对随时改变的事件的描述。因而,它能监控病人的病情进展,解释病人当前和过去的病情信息,并提出调整疗法的建议。VM 在旧金山太平洋医疗中心的外科重点护理小组进行了测试。这个程序的一些概念直接应用到了后来的呼吸管理设备中。

GUIDON (1977—1981)

GUIDON 也是一段试验程序,它旨在利用包含在基于 EMYCIN 的系统中的专家知识来服务于学生,教育学生。GUIDON 将一些独立的知识库综合起来,包括领域自身的知识库和教师教育的知识库。GUIDON 还组织一系列的会议,让学生与当前领域的知识对话。通过使用 MYCIN 的知识库作为所需教育的领域知识,GUIDON 研究了许多智能计算机辅助教育 (ICAI) 方面的问题,包括组织和计划对话的方法,生成教育素材,建立和修改一个学生已学知识的模型,解释专家的推理方法和过程。尽管 GUIDON 在许多方面都很成功,但它的诊断策略和一些暗含在 MYCIN 的规则中的医学知识都需要清楚地表述出来,这样可以便于学生的理解和记忆。由此,新的专家系统 NEOMYCIN 就被开发出来了。

SACON (1977—19778)

SACON 是测试 EMYCIN 的推理情景框架的工具。SACON 建议结构工程师使用一个大型的结构分析工具——MARC,SACON 也是许多咨询系统的原型。

MOLGEN (1975—1984)

MOLGEN 应用人工智能的方法研究分子生物学。它的初期工作集中在获取并描述那些设计和模拟化学实验中需要的知识。下面要介绍的 UNITS 系统也由此而生。第二

阶段的研究工作是描述遗传试验设计的特有方法,这里也导致了两个专家系统的产生。其中,一个系统是"骨骼平面图",它抽象出了一些试验方案的轮廓,这些试验方案能适应特殊的实验目标和实验环境。另一个系统是建立在有约束规划的基础上的,规划的决策在一个空间内制定,这个空间是由全局战略、独立领域的决策、有依赖领域的手工决策组成的,各个独立的步骤之间的交互作用或试验的子问题也构成了问题的限制空间。这两个系统合成了第三个系统——SPEX。MOLGEN-II 是 MOLGEN 的第二代版本,它研究遗传生物学的理论形成过程。

UNITS (1975—1981)

UNITS 是在 MOLGEN 项目开发过程中开发的基于框架的系统,它主要是知识描述、获取和处理的工具。UNITS 是给拥有一定计算机知识的领域专家使用的,它提供了一个界面,允许专家描述事实或启发式的知识。它既包含领域独立的模块,也包括特定领域的模块,这些模块包括修改描述过程性知识的英文规则。斯坦福大学已经特许 UNITS 可以用作商业用途。

AM (1974—1980)

AM 通过初等数学领域的发现来研究机器学习。通过应用由 243 条启发式规则组成的框架,AM 成功地提出新的数学概念,搜集相关信息,注意其中的规律性。这是 AM 工作的循环,完成了这个循环,它就通过定义新的概念来缩短与假设概念之间的距离。无论怎样,AM 并不能产生新的启发式方法。AM 的设计方案注定了它不可能成功;相关的发现新启发式方法的工作是由 EURISKO 来完成。

EURISKO (1978—1984)

EURISKO 是 AM 的后继产品,也是研究自动发现的。在发现过程中,它非常重视启发式方法以及它的表达方式,还有类推法的重要性。与程序、设计和模拟相关的几百种启发式方法引导 EURISKO 适应各种不同的领域。在每个领域,EURISKO 要执行三种不同层次的任务:解决领域水平内的问题;发明新的领域概念;合成新的功能足够强的启发式方法来协助解决领域问题。EURISKO 适用于:初等数学领域;发现 LISP 错误的程序设计领域;超大规模集成电路设计领域;石油外泄清理;其他的一些领域。

RLL (1978—1980)

RLL 是用来建立定制的表达语言的原型工具。RLL 是自我描述型的,也就是说,它用 RLL 的单元来描述自己。RLL 曾被用来作为 EURISKO 及其他系统的基础语言。

Contract Nets (1975—1979)

Contract Nets 的体系结构是早期的应用于计算机并行计算的一种体系结构。最近,Contract Nets 十分关注关于多处理器符号计算的文献。在 Contract Nets 的结构体系中,它用松散的双处理器结构来解决问题。处理器可以通过"协议"完成系统资源消耗低的作

业，如果作业失败，就将该作业分配给另外一个处理器。

CRYSALIS (1975—1983)

CRYSALIS 研究黑板模型在解释蛋白质晶体的 X 射线信息方面的能力。通过精炼蛋白质分子结构的描述，CRYSALIS 要拼凑出蛋白质分子的三维结构。尽管 CRYSALIS 的知识库只是为了解决这个问题的一小部分而设立的，分层次管理结构的黑板模型依然显示了解决如此复杂问题的强大能力。CRYSALIS 的成效正在合成到另一个系统 KSL 中，并对管理模型的发展做出了很大的贡献。

AGE (1976—1982)

AGE 项目要探索开发一个软件实验室，用来建立一些基于知识的应用程序。AGE-1 是一个知识工程工具，它用问题解决框架的黑板来建立应用程序。AGE-1 辅助程序的建立、调试和运行。AGE-1 已经在许多做学术研究的实验室使用，并在工业方面也有许多应用。

QUIST (1978—1981)

QUIST 将人工智能与传统的数据库技术结合在一个拥有大型相关数据库优化查询功能系统中。QUIST 用启发式方法来表达与数据库内容相关的语义知识，这样就可以推出查询条件的含义。然后，QUIST 重新构建一个最原始的等价查询，系统能利用这个查询在数据库中更高效地找到查询结果。最后，传统的查询优化工具将建立一个有效的检索序列。

GLisp (1982—1983)

GLisp 是一种程序设计语言，它可以修改对象以及对象的属性和行为。GLisp 编译器将这些程序转化为有效的 LISP 代码。有一个基于窗体的 GEV 数据检察员，它可以根据 GLisp 的描述来显示数据信息。与之一起，GLisp 编译器解除了对外部用户使用的限制。现在，得克萨斯大学在提供 GLisp。

XCON (1980 至今)

XCON 可能是最著名，也是最成功的专家系统。它是由美国数字设备公司开发的，主要是根据客户订单来管理复杂多样的产品配置方法。在 XCON 之前，数字设备公司的配置任务是由一些熟练的计算机专业人员手工完成的。且不管他们的技能，手工的配置方法经常会带来一些配置错误，也会造成一些订单不完善。自从使用了 XCON 系统之后，配置订单的正确率由以前的手工配置的 65% 提高到了现在的 98%。最初，根据评估，XCON 为数字设备公司节约了超过了 1500 万美元的成本；后来，每年为公司节约 4000 万美元。XCON 至今仍在运行，是专家系统历史上仍在使用的成功的系统之一。

第 9 章

知识工程与知识获取

学习目标

- 理解知识工程的概念以及它与传统的 IS 开发的不同之处
- 比较决策支持系统与专家系统
- 概括目前知识工程使用的工具与方法，预测这些工具与方法在未来的发展情况
- 理解知识的本质，并同数据和信息进行区分比较
- 从三种不同的角度看待知识 :表达法、生产法和状态法
- 了解知识的来源，能够对不同类型的知识进行分类
- 识别知识获取与管理的不同方法

小案例

Robotrader 与 Christine 的头脑

背景

20 世纪 90 年代初期，英国投资机构 Pareto Partners Ltd.的研究主任，开始寻找技术伙伴来帮助 Pareto 实现 170 亿美元资金的自动化管理。因为金融决策者与军事决策者都面临着信息的超载和精神上的压力，所以 Hughes 电子公司（一家从事导弹制造、机器人设计、间谍卫星制造等活动的公司）就成为了 Pareto 最合适的伙伴。

Robotrader 的开发

Christine Downton 作为一名经验丰富的分析家，曾经研究了过去 20 年的贸易市场，她于 1993 年加入了位于加利福尼亚州 Malibu 的 Hughes 研究实验室，并将她有关于世界债券市场的知识上传到了计算机中。Charles Dolan 是一名毕业于 UCLA 的计算机专业的博士，由 Hughes 公司派遣，负责获取 Downton 的经验。

Dolan 研究 AI 的方法是传统的符号逻辑与较新的联系理论的混合方法，这种方法产生于一个设想：智能行为产生于一种人工“神经网络”。Dolan 的观点是这两者都是很重要的——人脑的神经元网络中存在着一定的结构，这个结构就是符号的具体化。他试图在计算机上创造这样一个“知识空间”，主要基于一个符号化的结构。经过艰苦的努力，这个结构已经建立在他反应灵敏的“湿件”（wetware，即人脑）中了。

为了完成这件工作，Dolan 建立了一个 Hughes 称之为 MKAT（Modular Knowledge Acquisition Toolkit）的系统，用来进行人类专业技术的提取和编码。MKAT 以前被用作“知识工程师”的军事技术，例如坦克司令官如何策划对敌方目标的攻击。因为知识工程意味着交叉检验专家的思维过程，所以常常会揭露出一些名不副实的专家。Christine Downton 显然不是这样的人。尽管 Dolan 同 Hughes 技术队伍越来越精通知识工程方面的技术，但是 Dolan 还是花费了 18 个月的时间才得到这些过程的样本，并且采用了各种各样的工具模拟 Downton 经常描述的思维的复杂的训练过程。

结果，生成了具有 2000 条规则的“全球债券分配战略”。应用这些电子市场数据的反馈，系统吸收了大约 800 条的经济信息——譬如国家的公共部门与通货赤字、通货膨胀率、货币供应量等等。经过大量的复杂转化后，系统得出了一系列的建议性结论，例如卖掉丹麦马克而购买德国债券。这些信息被送到 Pareto 的真正的交易员手中，进行实际的交易。

Pareto 主要采用“定量的”的交易方法，从而通过使用简单的模型来模拟现实生活，进行交易与投资，而不是将精力放在理解这些理论和感知上。同样地，Pareto 很自然地转移到 AI 领域，而且 AI 也很容易地就适合于公司的具体业务。电子化交易系统提出一些建议，同 Pareto 的其他模型一样，交易员必须借此找出最好的市场价格。我们肯定地说，系统是在更复杂的层次上进行这些操作，但是实现的基本功能是相同的。

Robotrader

那么 Robotrader 表现如何呢？1997 年底，Robotrader 负责 2 亿美元资金的管理，而这些资金是极其多样化的投资组合，风险较低。系统得到了比债券市场基准收益高出大约 3％的收益——这正是一些大型的养老基金组织所追求的业绩表现。这个回报并不令人震惊。但是，Robotrader 不需要令人震惊，这些较低的风险水平是它可重新设计的参数中的一部分。它们都是程序自我驱动的。Downton 反对任何违反系统建议的尝试活动，尤其是市场处于不稳定状态时。

Downton 坦言说，人类是不能处理这些机器能够吸收的信息量的。正因为这样，她相信 AI 系统有朝一日能够从根本上减少金融行业中的中层人员的规模。人所起到的作用以及为此而索要的巨大利润额，将会被取消而全部变成自动化。互联网会加快这些过程，并能够将复杂的服务直接送到顾客手中。Downton 以及 Pareto - Hughes 队伍中的其他人相信他们的人工智能交易系统，Robotrader，就是迈向应用新技术的金融行业的重要的第一步。

——Yu-Ting“Caisy”Hung

Feigenbaum 发明了第一个专家系统，与其同时，他在斯坦福大学的同事创造了系统分析中的一个新角色：知识工程师（knowledge engineer，KE）：

> 知识工程师训练规则的产生技巧以及 AI 研究的工具，用来解决需要专家知识的困难的应用问题。如何获取、表示这些知识，以及正确应用这些知识来建立或解释因果链，是基于知识的系统设计中的重要问题……建立智能主体的技术既是编程技术的一部分，也是它的扩展部分。正是这种建立复杂的计算机程序的技术，才能表示、解释世界知识。（Feigenbaum1977，1015）

尽管知识工程的概念可以看作是系统分析的一种特殊形式，两者的目标却不是相同的。知识工程与系统分析都在寻求创造一个能够成功实现的信息系统的完全规范。知识工程的过程在核心方面不同于传统的 IS 开发：每个 ES 问题的背景是惟一的，它在寻找模拟人类专家的问题的解决方法。从这一点来看，我们明白了 KE 需要较高程度的领域知识同系统开发的广泛技术紧密结合起来。我们也明白了为什么 Feigenbaum 称知识工程为一种艺术。

Gaines（1995）指出了专家系统研究对信息技术做出的重大贡献。除了这些贡献之外，它也许是令人失望的，因为同过去的人工智能研究的许多经历一样，ES 研究也没能够指出人类知识过程的本质。到底什么是专业技术，为什么我们这么重视它，它是如何产生的，其内在的、社会性的、心理上的、逻辑过程是什么，它在人类物种的动力学中起到什么样的作用，等等。专业技术的观点集中于专业、事业结构、医疗保健系统、教育系统等方面。这些就是我们的社会运行的核心过程，也是人类社会中对人们进行评价的基础。

从信息技术的狭义观点来看，计算机与通讯系统可以在很小的程度上支持我们社会中的专家技术动力学，即使它们也不能有效地模拟或替代专家技术。在这样的技术发展

中，合理的专家技术模型是很有价值的，而且许多默认的专家技术模型就存在于发展的过程中。随着技术的发展，社会发生了变化，一个关于人类的完整的模型是很有必要的，它包括人类社会的一些过程以及一些产出。这样的模型会提供一些基本的方法，让我们理解我们到底是谁，我们从哪来，我们心理、文化、技术的相互影响，以及对于我们已了解的个人、社会、技术方面的影响。当以知识为基础的系统研究的目标从模拟人们的专业技术转移到通过企业模型模拟组织的专业技术的时候，这个长期的目标具有短期的重要性（Petrie，1992）。

在这一章中，我们会探讨与专家系统的设计与开发相关的知识工程和获取的一些关键问题和元素。首先，我们必须从本质上了解知识的主要组成部分。

9.1　知识的概念

在这个数字化时代的开始阶段，我们不断受到关于知识与信息重要性的干扰，而这些知识与信息主要来源于全球竞争优势与经济繁荣等方面。Drucker（1993）认为知识是未来组织的一种关键资源：

> 知识社会相对于我们已知的任何社会都具有较强的竞争力——由于一个简单的原因，就是知识很容易被广泛地接受，所以没有理由得不到实行。世界上没有“贫穷的”国家，只有无知的国家。同样对于各种公司、行业、组织也是这样的。对于个人来说也是这样，没有贫穷只有无知。（p. 80）

尽管数据、信息、知识这些词语有着常识性的意义，但是通常它们之间的区别并不很明确，因此它们常常被错误地交换使用。如果我们要操纵知识，我们必须十分清楚我们正在操纵着什么。

知识的确切含义

我们可以在任何好的字典中查到人类思维和推理能力的三个主要构造性定义，但是为了更好地理解它们之间的关系，我们必须知道它们的演变过程。

人类从用语言交流到文字的产生经过了 1.2 亿年。进化的早期阶段是从计数的出现开始的。早在公元前 3100 年，进化的过程是从刻有标记的木棍和木板开始，到抽象的数字、有含义的文字、带音标的符号语言。文字一旦出现，就开始支配着人类社会以后的 4500 年。到了 15 世纪中期，欧洲拥有大约 30000 本书籍。到了 1500 年，印刷术发明还不到 50 年，就存在了大约 300 万本书。在这个“数据不断丰富的时代”①我们第一次发现了完全理解人类思维与推理构造之间区别的重要性。

这一章没有进行关于数据、信息、知识的彻底的鉴别，也没有罗列出所有的哲学观点和定义。然而，我们会在这个范围内建立我们的一些定义，围绕一些细微差别进行讨论。

---

① 我必须感谢来自多伦多 Ryerson 综合大学的 JacekKryt 先生创造了这个词组，并且提供给我们，来解释我们认知活动的当前阶段。

出于这个目标,我们会采取以下的定义:

1. 数据 (data):就是一些事实、测量结果以及观察现象,分为有上下文与没有上下文两种。没有上下文的数据的例子就是 60,62,66,72。带有上下文的相同的数据就是 Laura,Samantha,John,Kristinede 的身高。数据的正确性与有效性主要是由数据的精确性决定的。
2. 信息 (information):以某种方式组合在一起的有用的数据,或者与解决决策方面的问题相关的数据。评价信息的关键的标准就是它的有用性。信息应该被看作是一种事件,而不是一种实体。同样地,当决策者仔细认识、理解结构化的数据集的时候,就产生了信息。
3. 知识 (knowledge):本质、观点、规则、步骤以及信息的混合应用,并在一定前提下来指导实际活动和问题解决者的决策活动。从这一观点来看,知识就是思维作出的解释。是否成功地解释问题之间的相互关系就是评价知识准确性的关键因素。

从上面的定义,我们可以推断出另外两个观点。第一,信息是个人的。一个人的信息就是另一个人的数据。第二,知识必须随着决策环境的变化而变化,所以不是静态的。对于一个实例或一段历史时期起作用的知识也许对于其他情况是完全无用的。同样,ES 只有在它的知识库包括当前情况知识的时候才会起作用。在内战时期我们用当时的知识可以建立一个 ES 辅助医生治疗伤员,但是在今天复杂的医疗环境中这个 ES 就完全没有用武之地。

关于知识的一些观点

认识论作为哲学的分支,将知识描述为一种实体。尽管从认识论的观点来分析知识的综合本质已经大大超过了我们的研究范围,但是我们还是可以通过回顾与知识构造有关的基本概念来浅尝知识的本质。

对于我们来说,了解关于知识的不同的观点是有用的。对于知识,有三种主要的观点:(1) 表示法 (representation),(2) 产品法 (production),(3) 状态法 (states)。

知识作为一种表示法

知识作为具体事物的表示方法,这种看法是由 Newell (1982) 提出的。他认为如果系统包含一种知识的表示法,例如模型或规则集,关于"某种事物……,然后系统本身就具备了知识,也就是关于这个事物的知识" (p. 89)。值得说明的是,这种观点区分了知识本身与知识的表示法之间的关系。你正在阅读的书并不是知识,而只是知识的表达方式。如果你不懂英语,那么交流思想的符号和形式对于你来说就是没有用的,知识的表示方式也是没用的。也就是说,只有表达方式可以被人理解,真正有用,作为表示法的知识才真正有价值。

这种观点另一个重要的特点就是,区别了知识的表示法与应用知识解决问题时所需要的处理过程之间的不同之处。Newell 认为"我们不会很容易地看到知识,而是仅仅通过使用符号表达式的解释过程的结果来推想出知识的内容" (1982,89)。换句话说,一个系统,例如 DSS 或 ES,不仅需要必要的知识表示法来解决特殊的问题,还需要有能力在知识表达的框架中加工这些知识。也就是暗示,系统中各种形式的知识表示法需要一个完善的知识处理机制。

知识作为一种产品

持有这种观点的理论家把知识看作是一组仓库或库存，每条信息从一个仓库传输到另一个基于过程或外部的事件的仓库。这种生产的观点认为学习是知识生产的一种形式，知识被创造出来，同样也被人们所获取。这里，我们发现决策制定的过程中有很多的知识仓库，这些仓库用来储备知识，直到需要知识的时候，或者根据需要而产生出来进行应用。这种高度比喻性的观点，在解释一定数量的认知行为的能力方面很有限，但是在定义一些计算机系统的设计，来支持制定决策方面是很有用的。

知识作为一种状态

看待知识的第三种途径就是把知识看作一个六个层次的状态过程，这种方法由 van Lobhuizen (1986) 提出。表 9-1 按照发展的顺序列出了这六个状态。知识的每一个状态都可以产生一个不同的、更相关状态的知识。解决问题的活动，如收集、分析、评估、赋权、综合以及选择等等，在知识从一种适当的状态转换到一种较高的状态的过程中起到了重要的作用。更进一步，当知识的状态升级的时候，信息超载的机会就会减少，知识的质量就得到了提高。决策作为知识的最高状态，就意味着状态转换过程的结束，也就实现了转换过程的目标或目的。

表 9-1　知识的六种状态

1. 数据	2. 信息	3. 结构化信息
4. 洞察力	5. 判断	6. 决策

知识存在于什么地方

了解一下对决策者有用的知识的来源，对于知识的特点的讨论也许会有所帮助。知识同时存在内部和外部两种来源，它们对于产生成功的解决方案来说都是很必要的。进一步来说，从外部来源获取的知识比较活跃、积极。也就是说，知识可以通过决策者与环境之间的积极的通讯与信息传递来获取，或者通过简单的观察与缓和的收集手段来获取。

知识的内部来源是通过知识的派生活动来获取的。决策者必须利用当前问题存在的知识作为基础，通过推理、逻辑和排除的方法来产生新的知识。对于解决特殊问题时使用的混合来源来说，决策活动是基于时间、经济资源、与问题相关的认知等的限制。另外，决策者必须估计出内部或外部获取的知识的可靠性，并且在知识相关的成本和认知结果之间进行平衡。即使获取一条知识很容易，而且并不昂贵，但是我们必须通过更多的资源来得到这条知识，以保证它的可靠性。

知识的分类

在我们讨论 ES 中知识的获取和管理这个话题之前，我们必须集中讨论另外一个方面 知识的分类。从直觉上讲，不是所有的知识都是相同的，知识的分类对于组织知识的存储以及确定给定条件下合适的知识的类型来说是很有用的。为了满足这个需要，我们

必须采取 Holsapple 和 Whinston (1988) 提出的关于知识的分类法。在这个分类中,确定了知识的三种基本类型以及三个二级类型。表 9-2 列出了这个分类法中知识的类型以及一些范例。

表 9-2 知识类型的分类法

基本类型
- 描述性:数据,信息,过去、现在和未来形式的描述
- 过程性:如何做事情
- 推理性:编码的管理,规则,原则,诊断规则

二级类型
- 语言型:词汇,语法,知识的表示
- 同化型:允许的内容,保持循环,相关性过滤
- 说明型:交流的模式,画图,信息传递,语言知识的转化

分类法中知识的三个基本类型不仅起到基本输入和与解决问题相关的过程的作用,也与决策者认识需求和进行决策活动机会的能力相关。过程性知识用来分析描述型知识,推理性知识用来控制分析和确定输出结果的有用性和适应性。

知识的二级类型被看作决策制定的核心过程的外部影响。每一种二级类型为决策者提供了推理的额外渠道,也提供了一些知识的产生过程,这些知识能够"反馈"与知识基本类型相关的过程。我们可以说有效的决策者 (或者有效的 DSS) 必须能够管理知识的这六种类型。

9.2 专家系统的知识获取

知识工程的本质就是知识获取 (knowledge acquisition,KA) 的过程。在这里,知识是从专门领域的专家获得的,并且进行收集、组织、形式化,以满足将知识转化为计算机可识别的表达形式的目标。KA 过程是一个复杂的过程,需要完全理解将信息从一个人转移到另一个人的不同渠道和机制。已经存在大量的 KA 手段与方法,但是到目前为止,还没有一个完全的可概括的方法。在本书中,我们会探索不同的措施和过程,这些措施和过程与建立一个 KA 的说明性的框架的一般方法相关。图 9-1 提供了心理上、计算上的与知识工程相关的基本情况。

图的左上方是能够熟练执行某领域内任务的专家。在右上方是能够模拟在那个领域内任务的执行情况的计算机。从专家到计算机的箭头表示知识获取研究的目标就是支持需要的专业技术从专家传递到计算机。图的下层部分描述了 KE 采用的工具与心理、计算模型之间的相互影响,这些模型是用来表示专家头脑中的知识和过程的。

任务的建模

不管通过什么途径,大多数的方法都是从任务或问题域的建模开始,ES 就是为此而进行设计和建立。任务模型的基本功能就是完全定义学习任务,辅助挑选专家,设置用户

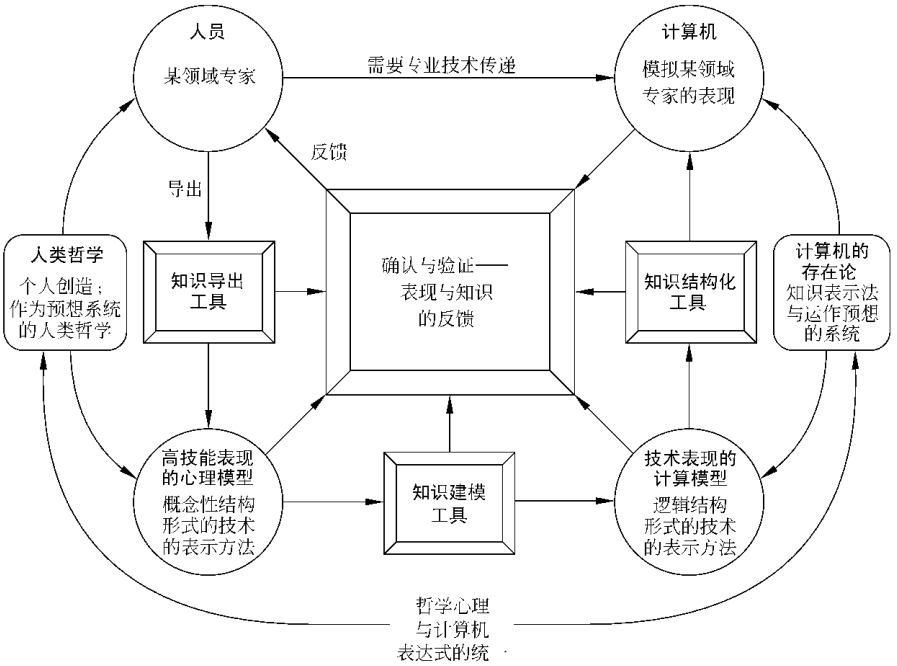


图 9-1　知识工程过程的概念模型

资料来源：Gaines，B. R. **和** Shaw，M. L. G. Knowledge Acquisition Tools Based on Personal Construct Psychology，© 1995 **年** Knowledge Science Institute **版权所有，经作者同意使用。**

特性和系统的性能标准，并实现专家反应协议分析的翻译框架的功能。任务模型（task model）可以被简单地看作是专家执行任务或解决问题的一般能力模型。初始的任务模型把任务或问题分解成一些子任务以及这些子任务之间控制和信息流动的说明。当 KE 应用其他的经验数据时，任务模型会从它的最初模型继续发展。

除了任务模型外，还需要建立一个概括性的领域描述。这个描述，通常从领域的一些相关的参考文献和教科书中综合出来，包括领域的词汇以及相关的理论和方法。另外，领域内用户的一般特征必须包含在这个描述中。用户的能力水平以及在任务和问题解决中的不同的作用必须规范化。最后，这个阶段必须进行发展 ES 的潜在成本和收益的评估。

性能的建模

性能建模（performance modeling）利用模型、假设、感知的分析技术从专家处提取真正解决问题的知识。这个过程发生的最初的手段是通过专家的语言表达，通常被称作协议报告（protocol reporting）。KE 从专家处提取知识，是通过一些语言表达，包括从非正式的访谈技术到高度结构化的协议安排等各种表达形式。获取了知识之后，就产生了知识模型，然后由专家来测试其准确性和适应性。KE 进行任何必要的修改，并且重复这个过程。直到从协议的角度来看问题空间中所有情况已经解决了，知识获取和性能模型过程才能进行下去。换句话说，只有在问题空间中对专家解决问题的行为的每一步进行了详

细描述和分类之后，KE 才能继续进行。我们在后面会进一步详细地讨论语言协议分析的过程。

知识获取的维度

Kim 与 Courtney (1988) 提出知识获取的方法应该分为两维：战略和战术。图 9-2 解释了各维的特征。正如你看到的，战略维度集中于驱动 KA 过程的具体手段，而战术维度关心提取专家知识的具体手段。

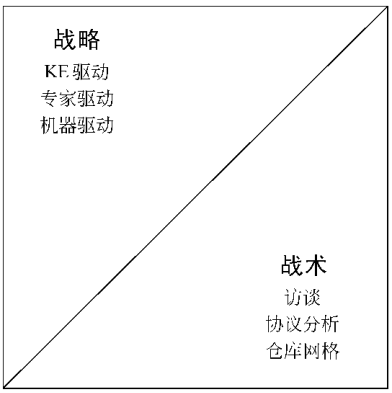


图 9-2 知识获取的维度

资料来源：Gaines，B. R. 和 Shaw，M. L. G. Knowledge Acquisition Tools Based on Personal Construct Psychology，© 1995 年 Knowledge Science Institute 版权所有，经作者同意使用。

KE 驱动

在战略维度中，知识工程师驱动的获取是最普通的方法。在这种方法中，KE 直接同专家相互作用，并且寻找方法将领域内的任务知识和性能知识模型化。好的通讯技术具有良好的耐性、持久性、移动性、合理的智能水平，对于这个过程的成功进行起到很大的作用。采用这种方法的基本的 KA 技术，包括访谈 (interviewing)、协议分析 (protocol analysis) 和仓库网格方法 (repertory grid method，RGM) 等。

专家驱动

在这种 KA 方法中，专家将他们自己的专业技术和知识进行编码，直接输入到计算机系统中。这种方法非常有效，专家起到两个作用：专家和知识工程师。尽管存在透明性减少的可能 (主要由于缺少关键要素的自治的 KE 净化机制)，但是如果专家能够通过独立的估计手段保证最终模型的性能，这种技术是会有帮助的。使用这种方法，可视化建模是建立领域模型的基本技术。这种技术使得专家将现实问题可视化，并且能够通过图形化的实体表示法熟练处理其中的要素。

机器驱动

AI 的这个领域集中于机器学习 (第 10 章会有比较深入的解释) 以及开发、提炼 KA 的机器驱动 (machine-driven) 方法。目前的技术正在尝试用感应推理机制从计算机接受的案例中提取知识。这些基于案例的解决方法，以及在解决问题过程中的不同属性，通过内部规则的计算机化的运算法则来获取。尽管还处在发展的早期阶段，但是这种方法很可能会对下一代的智能计算机解决问题产生重大的贡献。

知识获取技术

Hayes-Roth (1985) 认为 KA 发生在五个主要的阶段 (参看图 9-3)。每一个阶段想要

构造 KE 某部分的活动性能体系，使得每一阶段的输出可以作为下一阶段的输入。例如，识别阶段的活动中产生的要求就作为概念化阶段活动和过程的输入。

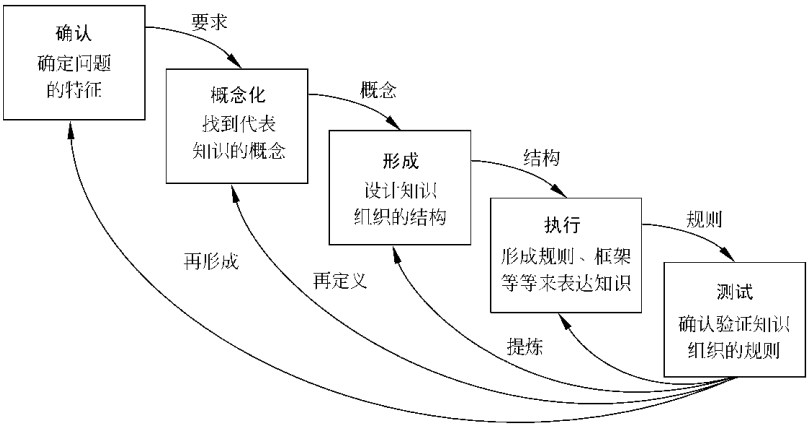


图 9-3　知识获取的五个基本阶段

图 9-3 清楚地描述了一般 KA 的过程。一旦过程完成了初始的循环，一系列的提炼、重定义和再形成就要求 KE 重新访问过程的不同阶段，直到所有的要求和知识的要素都确定下来，并且进行了正确的模型化。这个回归过程是一个冗长、乏味的过程，如果要取得成功就需要时刻关注 KE 各部分细节的运行情况。所以，这是保证获取整个知识领域的必要方法。

KA 的整个过程，知识工程师必须使用不同的提取工具来收集专家的知识。接下来，我们会讨论 KE 驱动方法最常见的三种技术。

访谈技术

访谈是所有 KA 技术中使用最广、最熟知的一种技术。尽管使用比较简单容易，但是访谈既是一种艺术也是一种技术，如果要获得有用的结果，也要求进行大量的准备工作。尽管已经存在各种各样的访谈技术，但是主要分为两个主要的类别：结构化和非结构化访谈。

非结构化访谈　这种访谈技术的类型是最正常的交流方式。KE 让专家进行没有特殊目标或目的的访谈，然后进行一系列的询问，这些询问最初是由专家对前面问题的反应所驱动的。KE 也许会选择准备一些话题，但不是进行正式的准备。非结构化访谈的本质就是不断地访谈、寻找方案。话题要么进行得最广，要么进行得最深入，然后 KE 使用探索性方法来指导访谈沿着最能满足特殊维度的细节要求的方向进行。这种形式的访谈在 KA 的最初阶段用得比较多，是知识领域内进行比较细节的分析所不可缺少的探索性阶段。

结构化访谈　与非结构化访谈相反，结构化访谈需要提前进行讨论话题的计划和组织、顺序安排、以及问题的构造等方面准备。如果正确地进行结构化访谈，就可以提取一些细节的观点输入到专家的知识库中，然后对一些案例解决方案的结果进行必要的过滤、

解释、判断,并且将它们转化成计算机可以识别的形式。与访谈结构化有关的步骤包括话题的挑选和组织、特殊问题集的生成、询问的方向、问题与询问方向之间关系的确定以及调查中不同案例的解决方案的确定等等。从这一方面来看,结构化的访谈不再是简单的交流,而是一种形式的询问,这样问题的内容就集中于特殊的信息和推理的路线。然而我们已经看到,专家被询问的问题语义的构造(问题形成的方式)也对过程的有效性以及回答的准确度有重要的影响(Marakas 和 Elam,1997)。

一般的访谈技术 不考虑访谈过程的结构化程度,还存在一些基本的技巧和技术。开始的时候,访谈的人需要以一些烦琐但很重要的要点开始访谈——比如接受访问的人的姓名和职务,访谈的日期,访谈的地点,以及访谈的主题等。同样,他尽量不问那些只需简单地回答"是"或"否"的问题,尤其是选择的访谈的方式是非结构化的时候。这样的问题被称作封闭式问题。除非问题的目标是让专家来确认已经得到事实的正确性,否则最好选择一些开放式的问题。开放式的问题通常会产生一个更自由的、具有"意识流"的气氛,来增加专家回答内容的细节信息。

在任何可能的时候,问题会通过避免双重问题的单独方式来表达:"这是你通常执行的步骤吗?采取这种方法是合适的吗?"总的来说,专家首先会趋向于集中回答最后一个问题,然后并不会给出混合询问的第一部分的完整答案。另外,询问者会尽量避免将问题构造成引导型或是假定型的。在前者中,像"你认为这是执行任务的最好途径吗"这样的问题将使专家要么作出简单的确认性回答,这样就限制了回答内容的细节内容,或者作出否定的判断性回答,从而使专家处于一种不必要的自卫性的境地。在后者中,像"假设你不具有这个层次上的知识,你是否能够执行这个任务"这样的问题会让专家有这样一种想法,就是说,自己很有可能与研究的任务不相关,所以只能得到较少有用的知识。最后,对于 KA 的任何活动,询问者需要花时间提前进行计划。表 9-3 包含了提前准备需要的一般要素。

表 9-3 提前准备访谈的基本检查清单

- 决定你需要知道哪些信息
- 询问自己为什么需要这条信息
- 决定包含这条信息的最好的访谈方法
- 考虑问题答案编码和分析的方法

基于访谈的 KA 的优点与局限性 尽管访谈始终是知识提取的最普通的方法,并且有许多明显的好处,但是我们必须同时认识到它的局限性。在 KA 的早期阶段,访谈是最有效的,并在此基础上建立领域知识的基本结构,以及在最后的阶段知识库的内容进行提炼和最终化。但是在获取性能知识的阶段,这个技术并不是很有效的。总的说来,通过访谈获取的信息本质上是综合的、不准确的、范围较小而且通常是矛盾的。由于缺少普通的、确定的技术结构,得到信息的分析通常是有问题的。最后,除了一些优点,访谈通常是花时间的而且是乏味的。虽然存在这些局限性,访谈始终还是 KA 过程中的一个重要的工具。

口头协议分析

大多数专家知识建模所需的知识形成于大量的认知过程中，并且在这些过程中他解决特殊的问题，执行特殊的任务。只有专家本人才知道这个信息的真实结构，而我们这些有一定认知能力和活动能力的科学家却无法理解这个结构。我们的问题在于我们不仅需要得到这些知识，还要按照计算机可以识别的方式进行知识的结构化。这种表达方式的不匹配 (representation mismatch) 就是 KE 面临的最困难的问题之一。

KA 的一种方法已经成功地减少了表达不匹配的一些困难，那就是协议分析的方法。协议是指一种记录或某种文件，用来记录专家在执行特定任务时信息过程的每一步以及决策制定的行为。在大多数案例中，在专家实际执行任务或者尝试将性能细节可视化的时候，协议就存储在磁带上。然后这些磁带由 KE 来进行转录和使用，进而建立一个规则集以及关于专家活动和行为的精确的性能模型。有几个聚集协议的方法，包括：回顾、反省、解释、协作。其中，协作的方法，或者“自言自语”，是最普遍和最实用的；所以，我们会集中讨论这个方法。

协作的协议要求专家在执行任务的同时自言自语或者描述他的想法。图 9-4 包含了 8 个任务的集合指令的列表范例，以及每项任务的执行过程中所包含的口头协议。

这种口头协议分析是由 Ericsson 和 Simon (1984) 首先提出的，它能够直接提取专家思想中的细节的过程信息，而且可以密切地反映正在发生的认知过程：

> 我们要求认知过程不由这些口头报告来修改，而是由这些任务导向的认知过程决定需要什么样的信息，需要描述什么样的信息……执行活动也许会缓慢下来，而且描述也是不完全的，但是……任务完成的过程和结构会保持基本不变。 (p. 215)

口头协议分析的局限性　对于任何 KA 技术，某些优点与局限性都与这个协议有关。首先，尽管口头协议提供了专家认知行为的密切的轨迹，但是它们不能提供建模必需的细节，因此就需要采用附加的技术。另外，尽可能地使协作协议的过程的设置与任务的自然过程的设置相近，这是很重要的。同样，要获取有用的结果，特定的任务条件必须与协作的环境相一致。表 9-4 列出了这些必要的任务条件。

表 9-4　成功协作协议的必需任务条件

- 使用的案例必须具有研究任务的代表性
- 每个任务必须有清楚定义的结论或完成点
- 在协议中任务可以被完善
- 所有的数据必须以相似结构展示给专家
- 测试案例必须在协议收集之前提供给专家，这样他就可以比较熟悉描述的过程

最后，要取得完整的协议，保证过程的不间断性是很重要的。只有描述完全后 KE 才会对专家进行询问。

1. First of all, write down the name of a subject or author of particular interest to you (a new name if this is the second time you are performing this task. Now try to a book on this topic that you would find of interest and briefly write down the name of it.
2. You have heard that Isaac Asimov has written a book on Halley's comet. Check if it is in the library and if you find it, write down the title and location code?
3. Now check if Isaac Asimov has any books in the library from his Foundation series of books. If so write down the number of items in this series that the library keeps?
4. You want to find the UDC code on hazardous substances so that you can browse through the shelves on that and similar topics. Try to find a book on this subject and write down the code.
5. You are looking for a book on pop art, preferably a dictionary. If you find one that matches the description, what is the publisher's name and year of publication?
6. You receive a telephone message to find a book by Sidney Siegell on non- parametric statistics. What is the ISBN number of this book?
7. For the book 'Fishes of the World', check and write down how many copies have not been lent out?
8. You have a friend interested in music, decor and dolls. You want to buy her an encyclopaedia on one of the topics but with not too many pages. Find a suitable book to buy for them?

Key:

S=user query	!=problem	=useful feature
@=assistance given	results of search= ?	achievement=50%

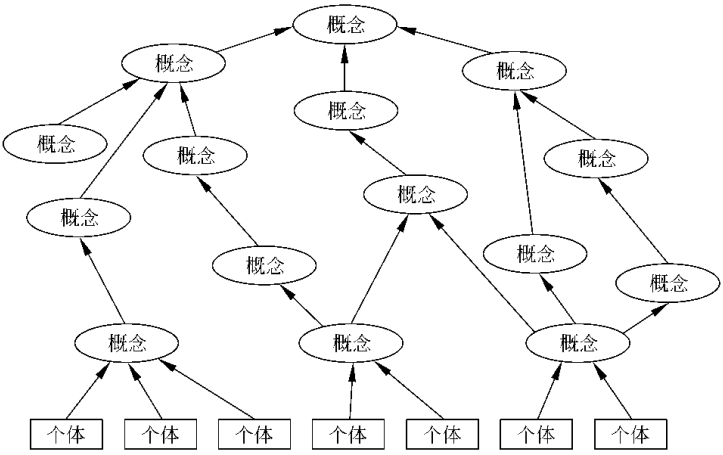
T1
Keyword=?
Shorter versions=many
! Some words in German.
? How to get to next page. Gave up 0%
T2
!How to get back to main menu.
! Help not helpful.
@Assistance given.
Looked at TIPS
_Suggested search by author.
! Wanted boolean to cut down
Scanned list.
Found details 100%
T3
Back to author list.
_Able to step through details.
_'v' to narrow search.
Found book and checked number of copies.
! Didn't expect 'u' to get loan details.
Found number copies of 100%
T4
! Selected subject no. option but did
not help to locate a UDC.
_ Got keyword list.
! Selected numbers but delimited with '+'.
_ Help useful.
! Pressed 'Enter' and lost search.
Restarted search, got list and found subject code 100%
T5
? Still not sure of option to get back.
? Wondered if can enter 2 keywords.
_ Entered one and found item 100%
T6
Entered incorrectly spelled name.
Couldn't find book.
! Entered 2 worded title but got long list based on one
Tried another word but gave up 0%
T7
Entered word in title.
? Confused by order alphabetical based strictly on first word.
! Pressed enter in error and exited search. Re-entered and
found number of copies. 100%
T8
? 's' didn't take to top level.
! Entered keyword but confused by context selection.
! Difficult to distinguish book types.
Entered word from title and got long list.
! Wanted to enter two words.
Found book, partially satisfied 50%

图 9-4 专家任务列表范例和协议

仓库网格方法

Kelly (1955) 提出了仓库网格方法，为 KE 的知识收集提供了另一个强大的技术。Kelly 把人类想像成为“个人科学家”，每个人都有他各自关于世界的个人模型。科学家进行知识的分类，然后把知识和感知进行归类，希望能够通过形成理论和检验假设来预测并控制事件。Kelly 把人类的思维定义为一种个人构造的理论 (personal construct theory)。每位个人科学家的世界模型由单独的个人构造组成，RGM 则作为一种有效的手段进行提取和记录。图 9-5 描述了 Kelly 的典型的仓库网格的概念化过程。

使用 RGM 时，向专家展示知识领域三个部分的目标。同时要求专家比较每个集合中的元素，识别出任何两个元素与第三个元素之间的一个或多个不同之处。每个连续集合就是用这种方法进行分析的，直到整个目标集合内部或之间的所有的目标集都区别完毕为止。我们用一个数值范围来确定后续分析的每个构造的等级和最后的聚集。一旦提



构造	元素 1	元素 2	元素 3
区别 1	约束 1, 1	约束 1, 2	约束 1, 3
区别 2	约束 2, 1	约束 2, 2	约束 2, 3
区别 3	约束 3, 1	约束 3, 2	约束 3, 3
区别 4	约束 4, 1	约束 4, 2	约束 4, 3
区别 5	约束 5, 1	约束 5, 2	约束 5, 3

图 9-5　仓库网格的典型结构

取和确定了这个表格，就可以作为专家个人领域模型的一个表达形式。

RGM 也可以用来推断多领域专家建立的观点之间的相似点。如图 9-5，如果一个人在网格中某处选择了一个特定的定义，以及在这个概念下需要确定的个体集，那么这个概念下的个体的衍生物的每个区别就形成了矩阵的各行；个体形成了矩阵的各列，某个区别相关的个体的约束就形成了矩阵中的数值。在一些简单地应用了仓库网格的例子中，这些约束用一些与其特征相应的取值来表示。

图 9-6 展示了地理学家关于空间画图技术的基本的网格。图中下方的元素就是一些画图技术，作为图表的列名。提取构造的目标列在了左侧，右侧的名称作为行名。在构造的维度之内的画图技术的等级形成了网格的主体。例如，“概率性画图”在“定量 - 定量与定性”的维度内处于等级 8，这就意味着主要是“定量的”。需要记住的是，坐标轴名字定义的谓词通过语义来表示它们自己是原始谓词的混合物。网格就是分析和提炼的开始点，并不是一个必要的概念性结构，而是发展概念性结构所需要的一个数据集。像多维和聚类分析等技术则用来进行表格的进一步分析，确定构造和元素之间的细微的相同点和不同点。图 9-6 的网格就是基于 Kelly 初始假设的一个典型的例子。最新的发展确定了元素之间和指定构造之间的结构化的联系，以及使得等级也可以扩展到更复杂约束范围内。最近，已经出现了一些计算机程序实现了知识收集活动的 RGM 的自动化应用。

仓库网格的优点和局限性　当 KE 面对高度结构化问题空间的知识建模任务时，RGM 是进行提取的较好的选择。需要诊断、分类、提取特性、建立关系的领域，是 RGM 比较合适的应用领域。然而，网格技术并不适合设计和规划问题，这些问题可以直接从问题的组成中得出（Klein 与 Methlie，1995），这是因为 RGM 不适合深层知识和执行知识的提取。

		1	2	3	4	5	6	7	8	9	10	11	12		
定性与定量	1	8	8	6	9	1	9	8	8	9	4	4	4	1	定量
局部的	2	9	9	6	3	5	1	9	9	1	4	4	4	2	全向
不考虑自相关	3	1	1	5	4	7	2	3	1	2	9	9	9	3	考虑自相关
不注重数据点	4	2	1	3	5	9	9	1	1	9	8	7	7	4	注重数据点
考虑多个变量	5	9	9	9	9	9	9	9	9	9	9	9	9	5	考虑一个变量
数学曲线拟合	6	4	4	8	9	9	1	2	4	9	5	8	8	6	不进行数学曲线拟合
非参数的	7	9	8	6	1	1	3	6	8	3	8	8	1	7	参数的
区间或比率数据	8	9	1	1	5	5	1	1	1	1	1	1	1	8	标称数据
需要周期性	9	6	6	9	9	9	9	1	6	9	9	9	9	9	不需要周期性
不符合趋势	10	9	9	1	1	8	1	7	9	1	6	1	1	10	数据符合趋势
大量计算负荷	11	7	6	7	8	9	4	4	5	3	1	2	3	11	无计算量
假定均质表面	12	1	1	4	3	8	9	1	1	9	7	6	6	12	假定非均质表面
对簇不太易受影响	13	8	8	6	9	3	4	7	8	5	1	2	2	13	影响簇
未结合地质模型	14	2	2	3	1	9	1	2	2	1	6	6	6	14	结合了地质模型
解释的	15	9	9	5	9	1	7	9	9	7	3	4	4	15	有代表性的
不太重要	16	4	5	2	1	9	1	1	6	7	8	9	9	16	很重要
不太有效	17	4	5	3	1	9	2	4	6	7	9	8	8	17	很有效
使用不广泛	18	3	8	7	5	9	3	3	4	6	2	2	1	18	广泛使用
		1	2	3	4	5	6	7	8	9	10	11	12		

1	概率映射	7	双傅立叶级数
2	趋势表面分析	8	可预言的表面
3	距离加权平均	9	三角测量
4	仿真映射	10	通用克利金
5	手工等值线	11	准时克利金
6	双三次样条	12	非参数克利金

图 9-6 空间绘图技术的仓库网格分析实例

多来源的知识获取

到现在为止,我们已经讨论了从一个专家获取知识的方法。尽管根据想像,由一个“领域专家”来构造知识库会比较有效,但是要完全获取知识或进行推理过程以及与所给问题相关的规则的建模,很有可能会使用多重的来源。当开发的 ES 中对专业技术没有进行很好的定义,或者发展方向是将专家资源扩散到那些任务主导的环境中,面对这些情况,使用多重的知识来源明显是很重要的。在这些情况中,KA 活动必须有多重资源来引导。

迄今为止描述的技术和过程在多专家工程中还是有用的,但是还是会有一些重要的不同点。值得注意的是,当使用单一专家的时候,不存在不可解决的模糊问题以及不同观点。当使用多个专家的时候,经常会有冲突的意见和必须解决的问题。在给定个人推理水平和典型知识领域中存在启发形式的时候,就会存在这些冲突。另外,因为专家也许会沿着不同的推理路线解决特定的问题,这种不同之处就产生了解决问题方法方面的冲突。为了最小化这种情况,已经发展了一些方法将知识从多个来源处进行集成。我们会简单地介绍两个比较相似的集成方法。

多数人的意见

这种方法使用一些制定群体决策时使用的技术和工具,当存在冲突观点时,在这些领

域内多个专家通常可以取得统一的意见。为每位专家提供其他人的判断和想法，每个人重新看待他们自己的看法，并且基于这个信息进行相应的修改。这个多数人意见的方法不能解决所有人的冲突，但是在大多数情况下还是会取得成功的。

Meta 分析

这种方法同“多数人的意见”方法相比采用了统计的、定量的方法，在专业技术涉及到像可能性估计这样的数值时比较实用。在这里，KE 使用专家的个人估计来定义一个概率分布，形成一个由个体元素组成的赋权集合。同样，数学统计方法也存在于多标准决策制定的领域中，这对解决多资源冲突的问题会起到一定作用。

9.3　确认和验证知识库

一旦 KE 获取了知识，就需要对知识的实用性和精确性进行评估。这个方面必须说明的两点是知识的确认和知识库建立的检验。在很多情况下，这两个要素被混合在了一起，通常也会同时在其他知识获取过程中发生。

确认 (validation) 和验证 (verification) 这两个名词通常被互换使用，这是不正确的。在评估的过程中确认集中于知识库的性能方面。换句话说，确认主要是检查我们是否已经建立了一个具有满意的能力和准确性的系统。相反，验证是检查我们建立的系统是否符合它的最初设置。验证回答了“我们是否建立了合适的系统”这个问题；确认则回答了“我们建立的系统是正确的吗”这个问题。虽然验证的过程只涉及到比较系统初始设定是否与最后完成的功能一致，知识确认的过程也涉及到了大量的验证和控制任务。表 9-5 列出了在 KA 最后阶段比较有用的许多不同的确认技术和测量手段。

表 9-5　知识库确认的测量方法和技术

- 准确性：系统如何能真实地反映现实世界？知识库中的知识的准确度是多少？
- 适应性：未来发展或变化的可能性
- 充分性：知识库中包含必要知识的比重
- 实用性：知识库中直觉、灵感与实用性之间的符合程度
- 广度：领域覆盖的程度
- 深度：知识详细的程度
- 表面的有效性：知识的可信度是多少？
- 一般性：知识库处理大范围的类似问题的能力
- 精确度：系统复制特殊系统参数的能力；知识库中建议的一致性与变量的覆盖程度
- 现实性：说明相关变量和关系；与现实的类似之处
- 可靠性：系统正确预测的频率
- 强壮性：模型结构结果的敏感性
- 灵敏性：知识库变化对输出质量的影响
- 技术性/操作性：假设、背景、约束和条件的好处
- 图灵检验：人类评估员判断给定的结论是由专家还是计算机完成的能力
- 有效性：正确解决问题的知识的足够程度 (在参数和关系方面)
- 正确性：知识库产生经验型正确预测的能力

资料来源：摘自 Marcot (1987)。

注意,逻辑上正确的知识库不一定是有效的,这一点非常重要。确认就是一种测量手段,检查知识库与知识领域模型的符合程度。从这一点来看,知识库的确认就是系统扩展的概念——就是系统反映现实世界的真实程度的一种表示方法。

有很多途径反驳知识库的有效性,经验证明主要有三种类型的缺陷:(1) 事实缺陷,语句没有明确地表示出与它相关的事实;(2) 推论确认,规则没有准确地表示出领域知识;(3) 控制缺陷,规则在前后关系上是正确的,但是存在结果上不理想的控制性行为。

9.4 小 结

为了很好地理解从专家获取知识的复杂性,并且将知识编码成为计算机基础的系统,需要对数据、信息和知识之间的区别有比较透彻的了解。如果一个 ES 要取得成功,知识工程师必须具有较高的技术水平,熟悉提取和组织专家知识的过程。

很明显,到现在为止,收集和确认具体专家知识的过程是一个复杂而又花费时间的活动。为了加深我们对专家知识的理解,在下一章中我们会将注意力从综合知识转移到“模糊”和近似知识上面。

复习题

1. 比较知识工程和传统信息系统的开发。
2. 区分知识、数据和信息之间的不同点,并且分别举例说明。
3. 在知识作为表达法的观点中,解决问题之前 DSS 或 ES 系统两个关键要求是什么?
4. 当知识作为一种产品时,知识的层次是怎样建立起来的?怎样进行改善?
5. 知识的六个阶段是什么,决策制定过程中它们是怎样转变到其他相关阶段的?
6. 信息收集过程中一些限制性因素是什么?
7. 比较知识的基本类型和二级类型,描述它们是如何联系在一起的。
8. 任务模型的主要功能是什么?
9. 定义一个“领域描述”(domain description)。
10. 从专家获取解决问题知识的主要工具是什么?
11. 知识获取和性能建模过程什么时候结束?
12. 比较知识获取的两个维度。
13. 比较 KE 驱动的 KA 和专家驱动的 KA。讨论每种方法的优缺点。
14. 同封闭式问题相比较开放式问题可以得到什么样的知识?
15. 协作协议分析的目的是什么?如何进行这个分析工作?
16. 知识收集的仓库网格法的优点和局限性是什么?
17. KA 协作和 Meta 分析技术是如何避免冲突的?

18. 在知识库的前提下定义确认和验证。
19. 获取有效知识的主要的三个挑战是什么？

讨论题

1. 按照你喜欢的规则采访一个专家。要求她识别出她按规律制定的决策，尝试记录下她得出结论时具体的思考过程和设想。你认为计算机可以学习思考和制定决策吗？
2. 结构化和非结构化访谈技术的优点和局限性是什么？描述一下采用各种技术的环境。
3. 描述一个领域的个人专业技术。你是如何获取这个领域的知识的？
4. 一些人认为专家系统总有一天会使得某领域内的专家知识成为普通技术。解释一下你是否同意这种说法。
5. 讨论一下，为什么说专家知识的获取和建模是非常困难和复杂的。
6. 大多数专家对于知识获取过程持有不同的感觉和反应。一些人因为被当作专家咨询和询问而感到沾沾自喜，然而其他人认为他们的决策制定过程太复杂了，所以不适于进行建模和模拟。作为一名专家，如果你要进行 KE 过程，你的观点是什么？

第 10 章

学 习 机

学习目标

- 理解什么样的问题需要机器学习系统来解决
- 理解使用模糊逻辑处理时，如何确定隶属函数，以及语义模糊性如何建模
- 理解模糊逻辑系统的优势和局限性
- 理解人工神经网络及其结构的基本概念和组成
- 理解神经计算的优势和局限性
- 理解遗传算法的基本组成和功能
- 根据不同种类的问题确定最合适的智能系统

小案例

大通银行信用积分系统

1985 年,大通银行 (Chase Manhattan Bank) 开始研究新的定量技术,来帮助高级信贷员预测公司贷款申请者的信誉度。大通成立了一个拥有 36 个成员的内部咨询机构,叫做"大通财务技术 (Chase Financial Technologies)",来指导模式分析网络模型的发展情况——这个模型是用于估算公司贷款风险的。

这样得到的模型,叫做 CreditView 系统,可以完成三年的预测,给一个公司赋予大通的风险等级:良好的、有争议的、较差的。除了一般预测之外,CreditView 详尽地列出了对于预测影响重大的项目、由专家系统产生的对这些项目的解释,以及若干比较报告。

CreditView 的模型是在大通财务技术的主机上运行的。客户端可以安装在每个用户的个人电脑上,通过电话线与主机交互。另外,传统的财务报表分析也可以用大通的财务报告系统来实现——那是一个独立的财务扩展分析软件包。财务报告系统也安装在用户的个人电脑上,并可以访问和显示公司的标准财务报表。

大通的分析系统成功的"秘密"在于其内含的模式分析技术,此项技术可用于建立混合神经网络,只要有足够的高质量历史数据。每个混合网络代表模式分析模块产生的一个单独的"模型"。PLCM (公众贷款公司模型) 是大通实现的第一个模型,来自于大通曾经给予贷款的大量的大型公开上市公司的历史数据。(大通的已有客户和潜在客户中,既有上市公司,也有私有的公司。)

使用模式分析模块分析,以产生预测模型的历史数据包括大量的数据单元。每个数据单元包括一个公司多达六年的连续财务数据,以及相应的行业标准数据 (六年中的最后一年称为"数据单元年"。) 数据单元的状态是公司的等级——G 代表良好,C 代表有争议的,X 代表较差的。系统使用这些数据构造一系列的申请者变量,这些变量有可能 (当然也有可能不) 预示着公司未来的财务状况。这些变量成为模式形成的基础。

A 模式本质上是对某特定财务变量或变量集的取值的声明。一个简单的模式可能具有以下形式:C1 < V1 < C2,其中 V1 是一个财务比率或变量,C1 和 C2 是常量。例如:1.75 < 酸性比率 < 2.00,这就是一个简单的模式。通常,模式是更为复杂的;它们包括几个这样的元素,并用"与 (and)"、"或 (or)"、"非 (not)"连接起来。下面的例子是一个小型的较复杂模式:

$$C1 < V1 < C2$$
$$V2 < C3$$
$$C4 < V3 < C5 \text{ and } C6 < V4 < C7$$
$$C8 < V5 < C9$$

这里所有的 C 都是常量,V 都是财务变量或者比率。申请者的变量被表示为成千上万的复杂模式,系统分析这些模式,生成一组最优变量和模式,它们组成了一个称为

“预报员”的模式网络。模式选择的标准如下：

- 分数：分数（可以从历史数据中观察到）衡量的是模式用于区分良好、有争议和较差的三个等级的能力——换言之，就是模式正确分类的能力。
- 复杂性：复杂性衡量的是模式的复杂程度（根据涉及变量的个数）、内含的简单模式数、以及能满足它的历史数据数量。
- 虚假性：衡量的是模式所得分数（不管它对预测的解释能力多强）仅仅是出于偶然的可能性。

这些统计量用于评估模式的预测能力，并保证它们的预测能力不是偶然产生的。对于每一模式和状态，都存在一种可能性（称为“精密度”）：一个对应于这个模式的数据单元可能具有相应的状态。这个系统使用一套具有专利权的网络平衡技术，来选择组成网络的模式，以最大化精密度，最小化偏差。

这套系统给大通带来的好处是显而易见的。大通现在可以确定一个债务人财务结构中的优势和弱势，并预测这些因素对于公司以后三年的财务健康状况的影响。这种“预测未来”的能力带来的成本节约乃是大通银行长盛不衰的关键因素之一。

资料来源：NIBS Pte, Ltd. (1996), www.singapore.com/products/nfga.

数字化计算机时代的降临，带来了一种普遍存在的误解——认为世界上任何时刻正在进行的信息处理工作大多数是由自动化的设备执行的。计算机的普及化让我们忘却了世界上最复杂、最强大的信息处理设备——人脑。让我们考虑一下控制论以及其他信息科学的基础性法则，我们就会发现，信息处理的实际本质植根于活生生的动物争取生存、适应环境的行为。从这种视角出发，今天的计算机能够处理的信息只能看作是信息处理的总量中极小的一部分而已。在这个基础上进一步，我们就可以认真地考虑发展模拟人类和其他生物的构造与运行原理的信息处理设备的了。

能够模拟人脑处理非结构化商务问题的数字化信息处理机，这是一个很吸引人的课题。相关的研究已经产生了许多新兴的跨学科领域：神经计算以及发展人工神经网络就是这些研究领域之一。神经计算涉及的问题是：使用数量巨大的简单处理单元相互连接构成网络，这些单元之间可以通过交换信号进行交互作用，共同决定网络的状态，从而进行信息处理。这种互联的信息处理过程，正是对人脑信息处理和学习方式的直接模拟。这一章中，我们将会介绍机器学习的概念，主要是讲解一些特殊的处理方法，例如神经计算和遗传算法，另外也会讨论把这些系统作为一个构成部分的众多应用系统和领域。

到现在为止，我们在一定程度上已经成功地理解了：我们如何对知识进行建模，把它转化成为传统观念中包含的规则和关系。不过，在我们着重分析机器学习之前，作为序言，我们还需要对人脑的实际工作方式建立一点认识。人类思考和推理的世界，是一个充斥着含糊、语义模糊以及模糊描述的世界。

10.1　模糊逻辑和语义模糊

模糊逻辑（fuzzy logic），相对于传统的从属关系设定和逻辑（源于古代希腊哲学，在人工智能研究初期得到应用，如 1985 年产生的 B 规则）的观念，是同领域中比较新的概念。作为一种推理方法，它允许不完全或者“模糊”地描述一条规则。将这个逻辑推理过程和数字处理机领域结合在一起，就产生了一类计算机应用程序，它们可以从自己的错误中“吸取教训”，并且能“理解”人脑中常常出现的多变的奇思怪想。

语义模糊

我们的语言充斥着模糊和不精确的概念。而且，有时候一条规则是鲜明清晰的，有时候却又很难用严密定义的特性清楚地描述（如果不是不可能的话）。举个例子，“Dan 很高”，或者“今天天气特别热”，这样的陈述，在日常交流中是非常常见的，也是复杂性可度量的推理过程的典型代表。然而，如果把这些说法转换成“更精确”的陈述，就往往会导致它们失去了语义上的丰富性和某些意义。例如，“目前的气温是华氏 97.65 度”，这样的陈述并没有明确地表示天气很热，而“现在的气温对于本地理区域在一年中这个时段的平均气温水平高出 1.1 的标准偏差”，这样的陈述同样存在许多语义上的困难。如果现在的气温比平均水平高出 1.099 标准偏差，那么今天算不算很热呢？要想确定今天是不是很热，我需要考虑我是在地球上的哪个地方吗？

关键在于，我们的语言已经进化到了这样的程度：我们可以通过近似的语义来传达自己的意思，而不需要精确的内容。我们很自如地使用带有分类性质的形容词，描述着聪明人、中等大小的汽车、高楼、强大的计算机系统，还有很多很多其他的说法。这种分类的方法给我们提供了一种对于很多一般名词的简单的表示方法，并且我们可以基于它们使用的上下文确定这种概括的真正含义。虽然对于人类，这方法很有用，但是这些基于上下文的意义没法直接翻译成通常的基于规则或者基于案例的编码方案，而这种方案是将知识导入专家系统知识库中所必需的。而模糊逻辑，则特别长于支持这样的情况下的推理，因为它更多地关注排定次序，而非准确的区分。

我们通常用来描述事物的分类方法并没有准确的边界。事实上，一个范畴往往包含一系列的值。例如，“技术高超”这个术语，就包含了一系列由上下文决定的意义。如果上下文讲的是一群小学生在打篮球，其中的一个学生投球比别人远一些，她就可以被称作“技术高超”。而另一种情况下，这个学生被放在篮球专业联赛中，“技术高超”这个词（虽然这个词和她曾有着直接联系）就不再适用了。这些范畴的“真相”如此多变，正如变光开关滑动时一盏灯亮度的改变。如果开关彻底关闭，那么这个范畴和原来就没有一点交集了；如果开关完全打开，范畴则与事实达成百分之百的一致；而这两种情况之间，无疑存在着无穷无尽的不同程度等级。

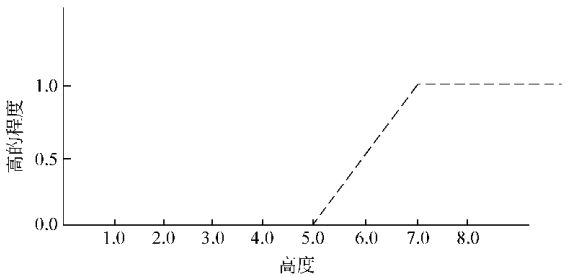
如果我们用这样的视角观察世界——任何事物都“在一定程度上”属于某个范畴或者集合，那么我们就可以理解模糊逻辑是怎样表示许多问题的上下文了。一条生产线的操作员不成文的规定可能是这样表述的：“当生产线接近其最大生产能力的时候，我就放

缓进料速度。”要把这条规定转化成为一系列精确规则，恐怕存在很大的问题，如果不是完全不可能的话。然而运用模糊逻辑，这样的规定就是完全可以接受的，而且也很容易适应。

模糊逻辑基础

模糊逻辑的基础是一个可以回溯到柏拉图时代的过程。它从根本上区别于亚里士多德等人早期对世界的概念化方法，因为它把事实定义为一个存在于（0.0，1.0）区间上的值，其中 0.0 代表彻底错误，而 1.0 代表完全正确。要注意的是，模糊逻辑并不意味着含糊的答案，与其说它是模糊的答案，还不如说它是随着一系列给定值而变化的精确答案。模糊逻辑可以处理任何精确程度的输入数据，并返回刚好同样精确度的结果。

与其搬出大量的数学证明，我们不如用一个简单易懂的例子来增进对模糊逻辑的理解。看看这个短句，“Dan 很高”。从这个句子中，我们可以推断出，存在一个包含所有“高个子”的子集，而 Dan 就包含在这个子集中。进一步，我们可以推断出，Dan 在“高个子”的子集中应当取到极值，因为他对于这个子集的从属关系是被副词“很”进一步修饰的。正式定义了模糊子集“高个子”（TALL_PEOPLE）之后，我们就可以回答“某某人高到什么程度”这样的问题了。为了做到这一点，我们需要对所有人的全集（ALL_PEOPLE）中的每个成员赋予一个它属于模糊子集“高个子”的从属度。最容易的做法就是创建一个隶属函数，自变量设为每个成员的身高。图表 10-1 中写出了这个隶属函数的表达式，并画出了派生的函数图形。



TALL(x) = {0, if height(x) < 5.0 ft.;

(height(x)–5.0) / 2.0, if 5.0 ft. ≤ height(x) ≤ 7.0 ft.;

1, if height(x) > 7.0 ft.}

图 10-1 “高个子”子集的隶属函数

有了派生的模糊子集“高个子”的隶属函数，我们就可以计算所有人的全集中任何成员“高的程度”的值了。表 10-1 给出了全集中取出的一组样本人员的“高的程度”值。从表中可以看出，每个样本人员都得到了一个“精确地”描述“高的程度”的数值。通过这个过程，我们就得到了一个描述高的程度的规则，这个规则是以“高度运算法则”的形式构造出来的，而这种法则可以包含在一个专家系统的知识库之中。通过建立这样一些法则构成的网络，我们就可以分析相当复杂的范畴的集合，也就可以进行基于给定问题背景的更高层次的推理。这就是专家系统的发展中模糊逻辑推理的核心本质，也就是我们这一

章的焦点问题 机器学习。

表 10-1　从高度函数中获取的样本值

人　名	身　高	高的程度
Tom Thumb	2 英尺 10 英寸	0.000
Mugsy Bogues	5 英尺 3 英寸	0.125
Albert Einstein	5 英尺 5 英寸	0.208
Thomas Edison	5 英尺 10 英寸	0.417
Michael Jordan	6 英尺 6 英寸	0.750
Kareem Abdul-Jabbar	7 英尺 2 英寸	1.000

但是,还要注意很重要的一点 隶属函数并不都像我们前面所举的例子那样简单。它们往往会涉及到多重复合子集、基于不同准则的从属度问题。例如,“高个子”子集的隶属函数可能不仅仅依赖于物理高度,还要考虑同年代的情况(对于他这个年龄,他算高的了)。事实上,某个从属关系函数涉及到选自大量不同准则和人群的元素,这并不是什么少见的事情。这种函数的复杂性是不容忽视的,要完全识别它们,还需要对一套精密复杂的数学概念有较深层次的理解。这种层次的理解还是留待那些建造这些系统、为管理阶层提供服务的计算机科学家们去完成吧。而在管理学的范畴内,重要的是这个观念 在智能计算机系统中,模糊性是可以编码的,人类交流中常见的语义模糊也可以建模,从而达到系统化。

模糊性与可能性

在我们结束模糊逻辑的概念之前,还需要澄清一点。模糊性(fuzziness)和可能性的概念经常混淆。两个概念都存在于相同的取值区间,乍一看取值的解释也差不多 0.0 代表错误(或者不属于),1.0 代表正确(完全属于)。然而,模糊性和可能性还是存在根本性的差异的。

为了完全理解这些差异,我们再来看一下这个陈述:“Dan 很高。”假设利用我们的高度运算法则,我们计算得到的这个陈述中“高的程度”的值是 0.80。因此,我们可以将这个陈述用符号表示如下:

mTALL_PEOPLE (Dan) = 0.80

其中 m 表示模糊子集 TALL_PEOPLE 上的隶属函数。

而对于一个可能性分析过程,我们应当把从属度值的概念解释为“Dan 有 80% 的可能性是高个子。”这两种陈述在语义上的差异是明显的。第一种陈述说明的是 Dan 个子高,也许不高;它只说明我们有 80% 的机会知道他应当属于哪个子集。相反地,模糊的陈述说明的是 Dan“高的程度”或多或少。换言之,可能性表示的是某事物具有某种属性的可能,而模糊逻辑处理的是某种属性在某事物身上体现的程度。对于前者,我们是在猜测一种属性存在与否,而对于后者,我们已经假定这种属性存在,并与集合中的所有其他成员比较这种属性的强度。使用可能性,我们只能知道个体是否属于某个集合,但使用模糊逻辑,我们还能了解到这个个体在某个集合连续的从属性中处于什么位置。

模糊逻辑的优势与局限性

关于规则最好的一点是：它允许用“联系”的形式（Dhar 与 Stein，1997 年），在普遍意义上简洁地声明关系。另一方面，它们不要求你能像数学模型那样精确。这一点十分有利，因为我们知道，当变量之间的交互作用具有非线性形式的时候，系统的复杂性会大大增加。公认的“模糊逻辑之父”Lotfi Zadeh 提出，“随着复杂性的增加，精确的表述失去了意义，而有意义的表述失去了精确性。”当特定知识范畴内的推理系统和处理过程越来越复杂的时候，模糊规则能够在精确性、简洁性、可实现性的多方需要之间做出一个诱人的折中和平衡。模糊逻辑能归纳出类别的概念，因为，从定义上看来，一个事物可以在某种程度上归属于任何集合，必要的话也可以属于多个复合集。从这种意义上说，模糊逻辑解决了“边界问题”就是取值为 1.1 使得规则生效，而取值为 1.099 则不满足规则的情况。最后的结果就是 在涉及到连续变化变量的情况下，模糊逻辑能产生比传统的基于规则的系统更为精确的结果。在此处，我们将研究应用模糊逻辑带来的主要好处，并指出它在应用中的几个限制条件，这些条件是我们所应当知道和理解的。

模糊系统的优势

模糊逻辑系统已经在许多应用软件中得到了使用，在商务和社会的许多方面，也都可以见到它的踪迹。它们在基于微处理器的设备和产品中得到了普遍深入的应用，而这种应用的价值也是毋庸置疑的。不考虑应用程序或环境的影响，模糊逻辑相对于传统的基于规则的逻辑，具有几条普遍适用的优点和好处。

矛盾的建模 模糊逻辑允许矛盾建模，并纳入知识库内。模糊逻辑可以是完全相互对立的，但仍然和谐共处。传统的逻辑中，相反的指令就会造成计算机无法执行，因而不能处理矛盾，相比之下，模糊逻辑则可以对这样的模糊性赋予一定的容忍度，从而达成妥协。例如，我们允许同时属于多个集合，一个 6 英尺高的人可能同时被赋予对“高个子”集合 0.80 的从属度，对“中等身高”集合 0.45 的从属度，以及对“矮个子”集合 0.01 的从属度。使用模糊逻辑，设计者可以完全控制一个给定的结论或者推理过程的精确性，他只要增加或者降低所需的规则数目，以及从属关系的灵敏度就可以做到这一点。

增进了系统的自主性 模糊系统的知识库中包含的规则可以彼此独立地运行。在一个传统的基于规则的系统中，如果一条单独的规则不完善，结果可能会导致从错误结论到系统完全无法求解等一系列的问题。而在模糊系统中，如果一条规则有误，则其他的规则就可以“补偿”这种错误。模糊系统确实改变了从前那种在系统灵敏度和鲁棒性之间做出牺牲、进行折中的局面。不同于传统的基于规则的系统，合理构造的模糊系统可以在系统鲁棒性提高的同时，在事实上提高灵敏度。模糊逻辑即使在整个系统完全改变的环境下，其规则也可以继续发挥作用。

模糊系统的局限性

除了优势之外，模糊推理对于知识工程和最终用户也都有特定的局限性。它的应用

并非总是强于传统的推理方法的,而且,存在多种问题解决策略的情况下,也需要根据给定问题的上下文,认真考虑这种方法是否适合。

系统验证的障碍　模糊推理可能成为系统稳定性和可靠性测试的障碍。在高度复杂的情况下,不可能知道是否应用了正确的规则。而且,当逻辑系统进行调整时,相应的从属关系也需要重新定义,而对于这种定义方法,并没有容易实现的固定的指导方针或规定可循。因此,设计者可能会发现,他们很难确定自己的行为是有助于提高系统性能,还是适得其反。针对这种局限性,我们使用了模拟方法来对一个模糊系统进行验证。多次模拟的结果可以用于分析在从属关系设定中的细密差别,及其对于系统输出结果的相关灵敏度。

模糊系统缺乏记忆性　基本的模糊推理技术不能从它们过去的错误中吸取教训,也就是说没有记性。因此,模糊逻辑还不能使系统效率达到最大化。目前为止,还不存在校验模糊系统正确性的精确的数学方法。进一步地说,如果给定的条件下,可以用模糊推理的方法进行建模,而且情况非常复杂,那么在表述上可能要同时应用多条规则。这种现象被称为模糊集饱和。意思就是,模糊集的推论太多,以至于后续的模糊集不堪重负。最终导致的结果就是:整个系统丧失了模糊规则提供的信息,而整个模糊域开始在平均值附近寻找平衡。怎样才能保证所有的规则和推论持续有效,怎样应付饱和问题?为了解决这些,人们致力于改进系统,使它能够从错误中吸取教训,并且能"学会忘记"在特定问题的上下文中不适用的信息。这样的系统是我们在本章中关注的第二个话题:神经网络。

10.2　人工神经网络

人工神经网络(artificial neural network,ANN)是在 20 世纪 40 年代由芝加哥大学首次提出的,作为一种模拟人脑认知学习过程的尝试。在过去十年中,ANN 以其建立复杂模型的强大能力在商务、金融等众多领域中得到了相当多的应用,然而人们仍不知道,这些有效的数据是怎样收集起来的。由于其强大的学习能力,与用传统的统计或数学方法相比,它们往往能够提供更好的解决方案。

ANN[①] 是很简单的基于计算机的程序,其基本功能是根据检验和误差建立问题解空间的模型。从概念上描述这个过程是很容易的:一个数据输入网络,ANN"猜测"出某种形式的返回值。如果猜对了,网络就停止运行;而如果猜错了,网络就进行自我分析,找出应当修正的内部参数,进行修正以提高预测质量。随着时间的推移,ANN 能够收敛于一个能够相当准确地描述过程的模型。

神经计算基础

为了更好地理解 ANN 从经验中"学习"的过程,我们首先需要理解作为人类神经过程和人工神经计算的基础的几条生物学法则。

---

① 本章中,"人工神经网络","神经网络","网络","网"等名词可以互相代替。

人脑

人类神经系统中最为基础的处理单元就是神经元。我们的神经系统是由神经细胞互联而成的网络构成的,神经细胞从不同的感官来收集信息,例如我们的眼睛、耳朵、皮肤、鼻子,这些感官分列在神经网络的不同位置。图 10-2 简单地表示了单个人类神经元的外观及其组成部分。

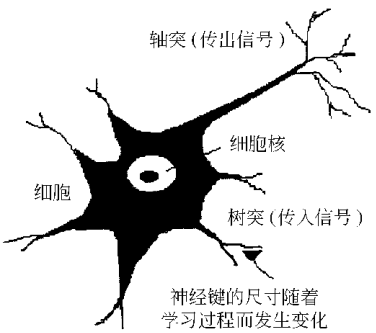


图 10-2 典型的人类神经细胞

如图所示,人类神经元包括核子和称为"树突"的连接器——它的作用是向神经元和轴突提供输入信号,而轴突将神经元的输出传送到神经网络的其他部分。神经突触形成的时候,就可以从一个细胞到另一个细胞传递信号。这是通过一个特殊的化学过程实现的,在这个过程中,神经传输质(某种特殊的化学物质)从结合丝一端释放出来,从而升高或者降低神经元内部的电势。如果一个神经元内部的电势达到了某种极限,就会有脉冲电流沿着树突发送出去,这个神经元称为"受到激发"。在神经元被激发的时候,产生的脉冲电流通常会沿着神经网络将信息传递到其他神经元。从概念上说,神经突触产生的放电规模越大,说明其中携带的信息越重要。

一个神经元接收到的信息可能起到以下两方面作用中的一种:要么"激活"细胞,要么"抑制"细胞。如果一个细胞被激活,那么它会成为激发态,并且立刻利用神经网络把信息继续传递到其他的神经元。而新细胞如果被抑制,就不会进入激发态,实际上阻碍了信息在网络上的流动。通过这个过程,每个神经元都对原始的输入数据进行了处理,并决定信息的重要程度是否足以让它通过并继续传递。

细胞之间的结合可以随着时间的推移和经验的积累而加强或者减弱。结合加强的结果就是"学习",而结合长期不使用会导致"忘记"。这个动态的过程造成了一系列改变:对于新刺激(stimuli)建立新的反应(response),对于现存刺激的修正以及对无用反应的拆除。

将大脑放在盒子里

根据对生物概念的模拟,人们在计算机的存储器中建立了 ANN,它包括一套节点互联的系统,这些节点称为神经节点,它们之间通过有权重的连接相互联系,就像人类神经的突触一样。ANN 被设计为多层结构,在层与层之间存在连接。图 10-3 表示的是一个简

单的 ANN。

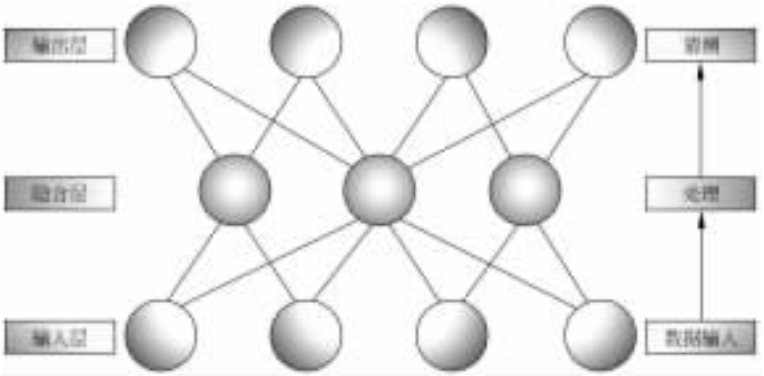


图 10-3　典型的神经网络结构

如图所示，ANN 包括三个基本层。接受数据的层叫做输入层（input layer），传播网络的最后结果的层叫做输出层（output layer）。中间层，也叫隐含层（hidden layer），才是对输入信号进行处理，确定输出信号种类的部分。这种节点之间分等级的系统动态作用，决定于对每一个节点输入的权重、以及相应节点的初始参数这两者的数学组合。每做完一次对网络的遍历，网络都会返回一个与上次输出的准确性有关的值，而所有的权重都要根据这个返回值进行调整。随着时间的推移，这些权重会被调整到足以精确地完成对输入数据的转换。在神经网络中，学习就是通过这些权重的调整实现的。

节点内部

图 10-4 表示了一个典型的节点的构成部分。最基本的构造包括一组带权重的输入连接、输入误差、状态函数、非线性转换函数，以及一个输出连接。下面我们将对这些基本组件和它们的功能进行详细的考察。

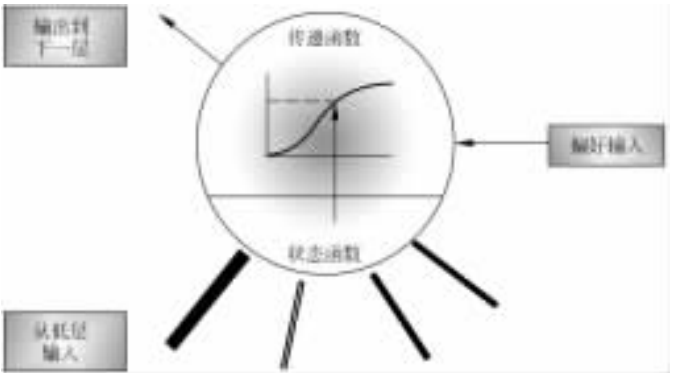


图 10-4　典型的神经节点构成部分

带权重的输入　如图所示，对节点的每一个输入都是与网络更低一层的连接。节点能够（也往往如此）接受多个输入，每一个输入都有相应的权重，即重要性。图中，我们用

来表示输入的线条粗细和样式均有差别,以显示其相应权重的不同。线条的粗细是与它的权重,即重要性,成比例的。而空心的线表示其权重为负值。一个输入的权重代表了它的重要性,表现为它对节点输出的贡献。因为在节点内部,所有的权重是要加在一起的,所以越重要的输入被赋予越大的权重,而相对不那么重要的输入就得到较小的权重。

误差输入 误差输入通常并不与网络连接,它被设为状态函数输入值为 1.0 的情况,其作用在于对节点的阈值进行个别修正,以便在学习过程之后实施对 ANN 的调整。在通常情况下,误差输入没有什么用,它的状态函数的值在遍历网络的过程中保持为一个常量。但是,一旦需要对某个节点进行修正的时候,那个节点的误差输入就可以被设为 -1.0到 1.0 之间的任何值,以找出应当赋予转换函数的恰当"误差"值。

状态函数 状态函数可以是任何形式、任何复杂程度的数学运算法则,但是通常情况下,它是一个累加函数。它的用途就是把节点各个输入的权重合并成为一个单值,以便传递给转换函数进行处理。由状态函数得到的值决定了加权输入对于转换函数的影响,并进一步影响节点的最终输出。图 10-5 表示了这一过程。

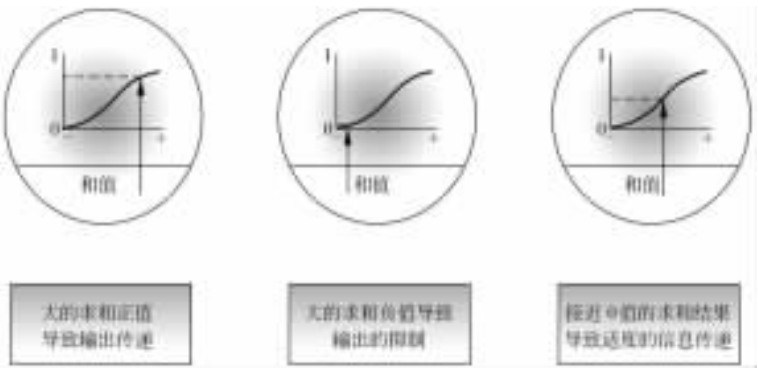


图 10-5 状态函数值和转换函数值之间的关系

转换函数 从它的名字就可以看出,转换函数的基本用途就是作为一个媒介,将加权后的信息变成输出继续传递。不过这个功能的实现方式是很特别的。见图 10-5,注意输出的值(最终是要观察阈值),它对于加权的输入并不成线性关系。转换函数可以比作是一个变光开关,控制着节点的"亮"和"暗"。

注意,图中我们选用了典型的逻辑函数(logistic function)的图形,来描述转换函数。逻辑函数的特点就是它的"S 型"形状。转换函数既可以是连续的(如图中所示),也可以是离散的(例如一系列值域为"开"或"关"的值),而且可以采取任何合适的数学形式。然而逻辑函数是最常使用的,另外常用的一种转换函数是基本放射型函数(radial basis function),更接近钟型的形状。在这里,关键问题是:转换函数必须在加权输入和合成输出之间加入一定的非线性因素。

训练人工神经网络

训练神经网络,使之能对特定的输入形式返回正确的输出响应,这个过程要用到重复样本及其反馈,就像训练人类一样。操作过程开始,先将网络中所有的连接权重设为小随

机数。这样,网络开始的时候就不会带有什么“记忆”和影响。然后,从已知输出的一组训练样本中取出一个数据样本,输入网络。网络对这个样本进行处理,然后返回一个对样本已提供的结果的“猜测值”。

可以想见,网络做出的第一次猜测多半是不对的。这是因为对于每一个输入连接的权重设置还不正确。但是,正如人类可以从自己的错误中吸取教训一样,ANN 也可以。一旦样本数据被处理过,ANN 就会比较它的计算结果和希望得到的正确反馈,并将比较结果记入“训练记录”。如果网络做出的计算结果和反馈匹配得很好,就不需要采取任何行动了。但是,如果它认为猜测值和反馈相距甚远,网络就要开始对特定节点的输入权重进行反复的微量修正,试图让它的输出与反馈取得一致。图 10-6 表示了这个反复修正的过程。

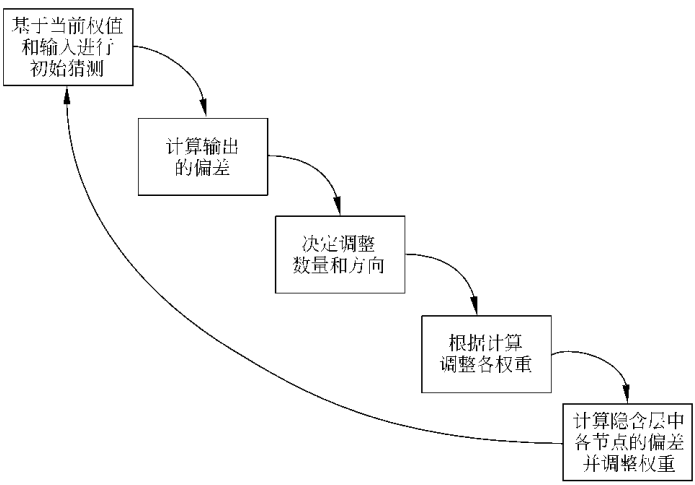


图 10-6　典型的神经网络训练流程

实际中,网络对每个输入权重确定适当修正值的过程是相当复杂的数学运算。然而本质上,它可以看作是一种形式的灵敏度分析:先确定错误的规模,然后调整一下,看看这个变化对错误的规模有多少影响(还有,是正面还是负面的影响)。这个过程要持续处理网络中的每个隐含层,直到所有的节点都经过了灵敏度分析,网络就可以进行下一次检验了。

送网络去学校——ANN 学习范例

神经网络寻找适当权重设置的实际过程称为它的“学习范例(learning paradigm)”。与人类获取知识和学习的方法相似,ANN 的学习范例可以大体分为两类:基本无需指导的和有指导的。

无需指导的学习范例

虽然我们建立了大量无需指导的学习范例,它们有各自的特色,针对不同的问题环境需要,但是它们都具有相同的基本特征和处理过程。因此,我们只讨论所有无需指导的学

习范例中共同拥有的特征。

无需指导的学习范例,它的本质就是“无需指导”,意思很简单,就是:我们提供给ANN输入数据和样本,但是不提供希望得到的反馈结果。无需指导的网络要想影响它的学习过程,需要从开始就根据它在数据样本中发现的相似性,来对它产生的训练记录进行“聚类”。网络继续处理数据样本,产生一条训练记录,然后对记录进行重新聚类,直到形成一组可定义的模式为止。每次成功的遍历之后,网络都能发现一个与输入数据匹配得最好地模式。随着时间的推移,网络不断改进对模式的识别,直到最后,训练集的输入和输出模式彼此能够很好的近似。

为了设想一个无需指导的学习过程,一个很好的办法是想象如下场景:你向一群陌生人分发一叠顺序随机的照片(照片上表现的是不同的场景),然后让他们把照片用某种方式分类。在不指定任何分类原则的情况下,每个人大概都会采用一个独特的过程,这个过程将是基于每张照片中某些共性的。一个人可能根据场景的不同进行分类(室内、室外、白天、晚上、风景、人物……),而另一个人可能根据照片中表现的情形分类(人的照片、机器的照片、自然景物的照片……),还有人可能根据照片的摄影风格特征分类(黑白、彩色、平滑、粗糙……)。虽然从某种角度来说,每一种分类都是合乎逻辑的,但在不同的分类方法下,照片与照片之间的关系会有差异。通过比较每种分组内部的关系,以及跨分组的关系,我们就可能发现从前的聚类中不能识别的关系。

ANN的无需指导的学习过程工作原理和刚才描述的场景很类似。网络并不知道应当发现什么,应当得出什么结论,因此,它需要自己发掘输入数据样本之间的关系,并利用这些关系创造出准确的输出。通过这样的过程,神经网络就可以根据数据的特定维度来“定制”它的学习了。在第12章中,我们将进一步关注这个关系发现过程的一种组织更严密的形式。

有指导的学习范例

与无需指导的学习过程相反,最普遍的有指导学习范例叫做后向传播(back propagation)。这种情况下,网络接收到输入样本,进行猜测,然后将猜测与希望得到的结果反馈相比较。从这种意义上,ANN是得到了反馈的“指导”,从而知道是哪里犯了错误,结果应当是怎样的。用以修正节点权重的运算法则是这样的:先计算出猜测和反馈之间的误差,然后把这个误差的度量值向后传播,经过网络的各层,直到抵达输入层。这种技术的名称“有指导的”,也就源自于这样的事实。后向运算法则的最普遍形式是使用误差平方求和的过程,产生一个误差总计的度量值。之后,对网络上其他权重进行进一步修正,把网络向产生正确输出的方向推进。这个修正过程基于一个混合数,包括了计算出来的误差度量值和网络训练者设定的修正增量。这个修正增量,或称“学习率(learning rate)”,是对于每个训练过程进行多大修正的规定。如果学习率定义的太大,网络的权重可能不断地“修正过头”,一直达不到要求的精度。反之,如果学习率增量定义的太小,网络就可能需要很长的时间才能达到要求的精度,甚至根本达不到。

在ANN的大部分内部工作中,数学计算的过程很复杂,因此,要想成功地构建一个可

运行的 ANN，需要有比较强的理解力。对于那些对微积分有一定天赋的人，我们在本章的附录中进一步描述了后向神经网络运算法则的数学基础。而对于其他人来说，理解这项技术使用的条件和方法，远比准确地知道它的具体工作方式重要的多。因此，我们下面把注意力集中到神经计算技术的应用中应当注意的几项优势和劣势。

神经计算的好处和局限性

神经计算带来的最明显的收益之一就是能够获得推论性的知识，这是其他所有基于知识的科学技术都不能提供的。除此之外，相对于传统的问题解决系统，ANN 还提供了其他几种优势。表 10-2 列出了这些好处。

表 10-2　神经计算带来的好处

- 避免了直接编程和复杂的“if. . . then. . . ”规则库
- 降低了对昂贵而且较难得到的专家资源的需要
- ANN 天然地适应性强，当输入改变时不需要更新
- 建少了对重新定义知识库的需要
- ANN 是动态的，随着使用不停地改进
- 能够处理有错误、不连续、甚至于不完全的数据
- 允许从特定的信息上下文中进行概括
- 便于从分散的数据集中创建数据提取
- 可以将“常识”纳入问题解决领域中来

ANN 的一项核心优势是它减少了对本领域专家直接输入的需要。在某些情况下，从某个领域的专家那里获取信息可能十分昂贵，不管是从经济的角度还是从所花费时间的角度。在另外的情况下，专家可能根本不存在。我们涉足的知识领域很分散，包括物理、生物以及社会科学，这种大胆的行径往往已经超越了一切领域专家的研究界限。在这样的情况下，ANN 为新的关系和知识的发现提供了一种宝贵的媒介，这超出了其他支持问题解决的技术的能力。

ANN 的另外一项独特的优势，在于 ANN 与生俱有的处理含有噪音和不完全数据集的能力。大多数情况下，高质量无杂质的数据集很难得到，即使不是根本得不到，因此，问题解决系统就面临着分析分布不均、不连续的数据的问题。ANN 的结构本质决定了它比之传统的统计分析和人工智能方法，更适合处理这样的数据。因为每一个神经节点都只处理问题的一小部分，而神经节点和层都是被设置为从不同的角度处理每一个小元素。因此 ANN 往往能够推断出数据“应当”是怎样的，然后将这个推断作为它目前的猜测的输入，从而实现对丢失或者损坏的数据的重构。当网络试图降低先前的训练阶段中误差的时候，这样的推断也可以同每一个输入的权重一起更改，如果必要的话。

同理，ANN 擅长考虑一个大而复杂的问题的每个小部分，这使得它具有一项独一无二的能力：当问题的复杂性上升的时候，它的规模也可以随之扩大，以解决问题。直觉上来看，更复杂的问题需要通过向网络增加更多的隐含层，从而提高网络应用较低层节点输

出(即允许更多节点参与交互)的能力,来解决问题。通过这种方法,许多复杂的、高度非线性的关系和处理过程都可以得到高效的建模。

然而,我们必须知道,这些神经计算的优势和好处都是有代价的。表 10-3 列出了在一定的环境下,与神经计算应用技术相联系的局限和缺点。

表 10-3 神经计算技术的局限性

- ANN 不会对它的推论做出解释
- ANN 的本质是一个"黑盒子",因此,在可说明性和可靠性方面存在一定的问题
- 重复的训练过程可能很消耗时间
- 熟练的机器学习分析员和设计者还是稀缺资源
- 神经计算技术常常会挑战现存硬件的极限
- ANN 要求对输出结果赋予一定的置信度

ANN 常常用于模式识别和非线性关系方面的复杂问题求解,效果很好,因为 ANN 具有发现数据集中微妙关系的独特能力。但是,很不幸地,这样的微妙关系被揭示出来之后,ANN 没有提供解释或者修正它们的技术。ANN 确实可以根据数据集固有的、而且是隐藏的特性对其中的元素进行有效的区分,但是这并不是说,蕴含在这些数据集中的特征意味着识别是建立在预期特征差异上。事实上,ANN 可以理解并区分噪声中包含的模式与数据。

军方有一个著名的例子。他们建立了一个神经网络,试图对包含特定军事装备和不包含这些装备的图片进行区分。在早期的训练阶段,对网络输入两种图片:一种图片包含战斗坦克,另一种图片不包含坦克。每一幅图片都被转换成计算机能够识别的格式,输入训练对象网络。经过相对较少次数的遍历,网络表现出了令人振奋的很高的识别率,能够区别包含坦克和不含坦克的图片。为了保证训练的成功,又一组数字化的图片被输入网络进行分析。然而训练阶段产生的兴奋很快转为失望,因为网络并不能成功地区分指定的元素。对训练集的检查很快找到了症结所在:包含坦克的图片都是晴天的时候拍摄的,而不含坦克的图片一般拍摄于多云天气。网络学会的是区分晴朗和多云的天气,而非识别战斗坦克的形状。试图改进现有方法的新途径每天都在产生,这大大促进了神经计算和机器学习技术的进化。下面,我们就要讨论对"常规的"神经计算技术的一项改进。

10.3 遗传算法和遗传进化网络

回想我们在第 3 章中讨论过的有约束的理性和优化:人类虽然希望得到最优解决方案,但是往往容易满足,而接受一个比最优解差一点的解。Simon 对此解释说:最优解固然值得称道,但难以获得,因此是一个有矛盾的目标;虽然如此,我们在日常问题解决中仍然不懈地追求最好的可能解决方案。我们知道,对于我们的问题是存在一个最优的、完美的解决方案的,只要我们能找到它。

最优化的概念

任何问题解决系统的目标就是找到一个满足问题背景中所有约束 (可认知的,或现实存在的) 的有效解决方案。这说明,对于给定的问题背景和约束集,存在一个可用事先确定的指标来衡量的最优解。虽然 Simon 关于优化和满意的观点从人类认知的角度来说是正确的,但是确实存在某些形势下,最优解不仅是我们所追求的,而且是可以达到的。管理科学领域一直都成功地利用了数学技术的发展,在给定的 (往往很复杂的) 约束集内取得最优的解决方案。这些技术在资源稀缺的情况下很有用,因为它可以通过最大程度地利用问题中存在的有限资源来达到最优。计算机的出现,结合这些数学优化模型方法的使用,大大拓展了我们找到最优解的能力。我们的问题是:我们熟练解决复杂问题的能力越来越强,而我们对寻求更复杂问题及其约束条件的最优解的欲望也越来越强。这个复杂性的循环决定了:我们需要不断地发明获得最优解决方案的新方法。

考虑第 3 章中推销员的旅行问题。回忆一下,问题之一就是确定不重复路过任何城市的独特路线条数。我们知道,要访问 10 个城市,就会有181440种不重复的路线[①]。假定我们有一台计算能力比较强的计算机,一秒钟可以估算 100 万条不重复路线,那么我们在大约 0.18 秒中可以完成对这个问题解空间的穷举遍历——考虑到效率提高和成本节约的回报,这个时间不算长。但是,如果我们对这个新实现的优化能力很感兴趣,想用它解决规模更大的问题,我们很快就会达到穷举遍历的极限。如果我们把城市数目增加到 25 个,那么不重复路线的可能数目就是 3.10×10^{23} 条。这没问题,我们有一台可以在设定的解约束条件下找出最优路线的计算机。但是,很不幸的,我们的这台每秒能估算 100 万条路线的计算机,要想解决 25 个城市的问题,运算时间长了一点儿。按照这个速度,它要完成穷举遍历并找到最优解,大约需要 100 亿年左右[②]。这表明,常规的优化方法是有局限性的。

继续回想推销员旅行问题,我们找到了一种启发式的方法,通过最小的付出得到一个比较好的结果。这样的技术已经进入了机器学习领域,而且其中的一种方法——遗传算法,正在赢得广泛的欢迎和应用。

遗传算法简介

与 ANN 的理论基础相似,遗传算法 (GA) 也是基于生物学理论的。ANN 的根基是神经系统科学,而 GA 则不同,它植根于达尔文"自然选择和适应"的进化论。

自然选择的理论提出了一些引人注目的论点:具有某些特征的个体生存能力比较强,因此会把这些特性遗传给它们的后代。遗传算法是一项非常简洁,但是功能强大的优化技术,它源自遗传学和自然选择理论。它最初是在 20 世纪 40 年代年由 MIT 的 John Holland 提出的,GA 的本质是一系列模拟"适者生存"概念的适应过程,具体说就是根据计算出的网络猜测值和要求的方案状态之间的差异,对计算法则进行不断的重组。GA 的力

---

① 公式为:0.5 (城市数目 - 1)!

② 设每年 365.25 天:求解 3.10×10^{23} 条无重复路线大概需要大约 8.62×10^{13} 小时,大约 9830411719 年。

量在于:当种群中的某些个体进行配对之后,会产生后代,而这些后代中就很可能保留了它们双亲中需要的特性,甚至可能结合了双亲最优秀的特性。通过这种方法,种群总体的适应性将会一代代地提高,我们也就得到了更好的解决方案。相对于常规的无需指导的神经网络,GA 的最大优势是它可以对行为准则进行复杂的组合,从而克服复杂 ANN 发展过程中组合上的局限性。

自然选择的概念

为了更好地理解 GA 的基本原理,我们首先需要理解它的理论基础。自然界中所有生命体面临的首要问题就是生存。那些能拿出好的解决方案的个体能够延续生命,而应付不了问题的个体就此灭绝。达尔文关于物种进化的观点是:那些能够生存下来的生物之所以能做到这一点,是因为它们尽量让自己适应最适合自己生存的环境,并且远离那些对它们有害的环境因素。对环境的适应性的存在,就是遗传算法发展和应用的基础。

遗传算法的基本组成部分

GA 的最小单位叫做基因(gene),这就跟它在生物上的喻意相一致了。基因表示的是问题域中最小单位的信息,也可以看作是每个可能解的最小基石。比如说,如果问题背景是设计一个均衡的投资组合,基因就代表着某种证券要购买的股数。如果问题背景是旅行商问题,基因就代表一条路线上必须访问的一个城市。继续与生物概念类比,一系列的基因(代表一个问题可能解的一部分)就叫做染色体(chromosome)。图 10-7 列出了典型的 GA 中可能包含的染色体串的例子。

Well-balanced investment portfolio chromosome:

{100 300 1000 500 ... 200 1500}

The decoding function for this chromosome interprets this to mean:

Buy 100 shares of Stock A, 300 shares of Stock B, ... etc.

Traveling salesman city routing problem:

{Miami Newark Chicago Denver Phoenix ... Dallas}

The decoding function for this chromosome interprets this to mean:

Start in Miami, go to NewYork, go to Chicago, ..., go to Dallas, return to start}

图 10-7 遗传算法染色体串举例

染色体在计算机内存中以一串二进制的数字的形式而存在,可以被 GA"解码",来决定某个特定的染色体基因库的解决方案对于给定的问题是否合适。解码过程仅仅是告知 GA 染色体中的基因各代表什么。例如,对于推销员问题,解码器应当知道怎样把基因转变为沿线的城市。对于一定问题背景下 GA 的应用,解决方案的这种作为字符串的表现形式大概是惟一重要的约束。为了使 GA 有效地求解一个问题,问题的解决方案必须能表示成二进制数字字符串的形式。

图 10-8 画出了 GA 应用中的基本过程。

初始化 GA 创建的第一步就是生成一个初始染色体种群,也就是考察后得出的问

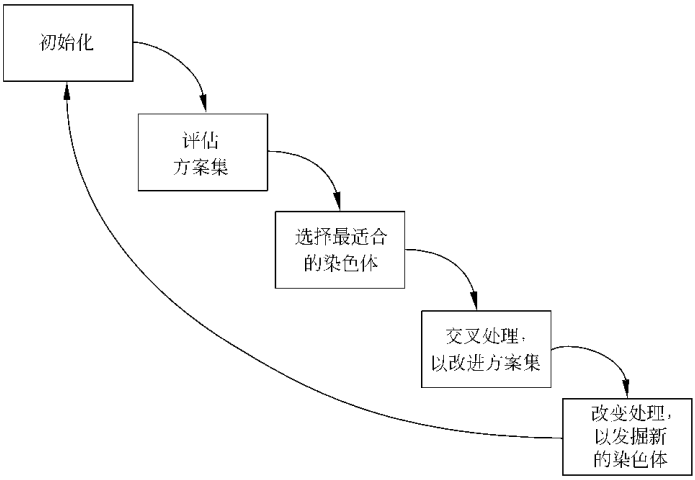


图 10-8　遗传算法解决问题的基本流程

题的可行解。这些初始基因结构的种群应当是这样建立的：它们代表着随机分布在解空间里的一些解。

进化　在进化阶段，每一个染色体都被解码函数解码，并且使用一个合适度函数来判断哪一个基因结构比较好，哪一个不够好。合适度函数相当于线性规划这样的优化技术里的目标函数。这个函数代表着对某些资源使用率最小化/最大化的某些约束或者要求。在推销员问题中，典型的合适度函数可以包括了对路上时间最短的要求、对燃料或旅行费用最少的要求，以及每个城市联系时间最长。

图 10-9 表示的是解码过程，举了一个二进制染色体的例子，表示了一条路线沿线的城市、它们到某个基点的距离，以及联系的时间。按照问题域来考虑，解码函数和合适度函数是惟一根据问题上下文来决定的 GA 的组成部分。因此，要想改变 GA 对解决方案的排列顺序，只要简单地改变合适度函数的性质就可以了。另外，同样的 GA 也可以被用于解决广泛的问题，只要简单地把解码函数和合适度函数改为新问题域的相应形式就可以了。

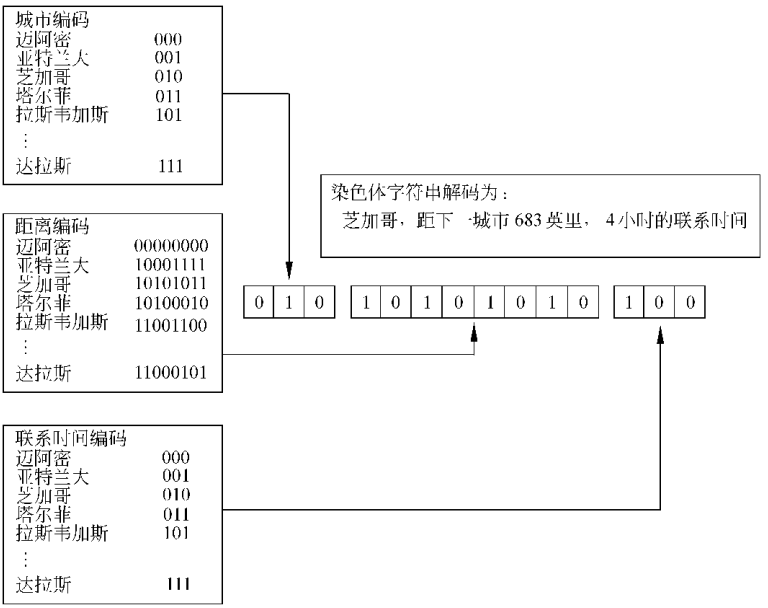


图 10-9　染色体编码和解码示例

选择　一旦初始的染色体池受到合适度函数的评估，GA 将重复地改进初始解决方案，试验新的解决方案，以求最合适的解决方案排在所有方案前面的可能性最大。基本上，GA 会从现存的染色体集开始试验，手段是对每个染色体中包含的基因进行联合和改进。目的则是产生新的染色体，并组成一代新的可行解，再进行评估。

在第一步——选择中，GA 挑选那些最合适的种类，把它们保存下来，并在解决方案染色体的种群中进行复制。比较差、不那么合适的染色体则被淘汰掉。这种进化过程的最普遍方法称为合适度比例选择。在最简单的形式中，这个方法确定每个染色体相对本代所有其他染色体的相对合适度。如果染色体 A 被认定比染色体 B 合适一倍，那么染色体 A 生存下来的概率就是染色体 B 的两倍。最合适的种类不断进化，弱势种类灭绝。选择过程清除了较差的解决方案，并将剩余的染色体传递到交叉过程。

交叉　在经过改进过程之后，交叉阶段要在两个被选定的染色体之间进行基因交换。图 10-10 表示了这一过程，其中一个染色体的一部分被传递到另一个染色体中，反之亦然。

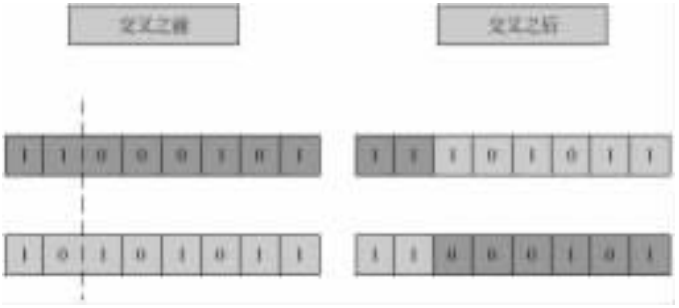


图 10-10　染色体交叉过程

交叉操作的目的在于允许 GA 创建新的染色体，新染色体不仅能继承积极的特性，同

时也降低了原先可行方案中相应的消极因素。这和生源论中“优化繁殖”的相应概念也是类似的。

但是,交叉阶段有一个重大的局限性,就是它只能重组在染色体种群中已经存在的基因信息。例如,我们的推销员问题,如果我们没能在原始种群中包含入一个城市或一种特性,那个值就再也不会成为任何解集的一部分,于是,也就不可能被交换到更好解决方案的染色体中。不过,GA 实际上有一种技术,可以进行新基因码的自发进化。

突变　改进过程中的最后一个过程就是突变。在突变阶段,基因现有设置的值可能随机地改变为另一个完全不同的值。图 10-11 表现了这一过程。

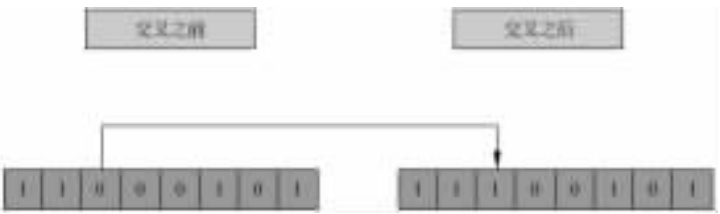


图 10-11　染色体突变过程

这一阶段发生的绝大多数突变,通常像自然界的情况一样,退化居多,进化较少。然而,偶尔也会发生一次向更好、更有利方向转化的突变。突变给了 GA 一个机会,使它能够创造出触及从前未知的问题解空间的染色体和信息基因,因而提高了发现最优解的可能性。但是与自然规则一致的是,突变发生的频率是非常低的,通常用以保证在解空间内任何点都能搜索到。

遗传算法的好处和局限性

抛开遗传算法的强大力量不提,在建立一个 GA 问题,或者说对一个 GA 问题进行编码的时候,还有一些应当注意的事项。确实存在几个参数,对最优解决方案的发现速度有着重要的影响。

在这一点上,最重要的参数之一就是种群规模。理想的初始解决方案染色体数量是在过程消耗时间和解决方案种群分散度之间进行平衡的结果。Wayner (1991) 已经证明了 对于大多数问题,种群大小为 10 似乎是最优的。少于 10 个染色体会在寻找可行解的过程中导致更多失败,而多于 10 个染色体则会增加处理时间,而且未必能找到更好的结果。

人们发现,使用 GA 解决复杂问题的另一个优势也可以归结为种群规模。对于一个解决方案来说,GA 收敛的速度比神经网络迅速的多,而且它们的处理时间也非常容易预测。既然 GA 总是沿着相同的步骤进行,那么总处理时间就取决于初始解决方案种群中染色体的数目,以及设定运行几代。用评估单个染色体所用的时间乘上染色体个数,然后再将这个值乘上代的数目,就能得到一个比较精确的解决方案寻找所需时间值。

另一个可能直接影响 GA 效率的参数就是交叉和突变的过程。突变必须谨慎地使用,以保证不会出现重复的解决方案。而交叉必须进行合理的设置,以保证适当数量的新结合体得到评估,而且不浪费分配给 GA 的资源。如果不进行交叉,那么评估中就不会出现新的结构。如果种群中的所有染色体都进行了杂交,那么好的基因也会和坏的基因一起消失。启发式的说法是 染色体种群的 90% 参与交叉过程,而突变的发生时间保持在总评估时间的 5% 以下,比较合适。

GA 对于解决复杂问题的最大优势在于,它们能够让自己覆盖广大的解空间,从中寻求最优解决方案。而且,由于所有的最初种群都是由可行解决方案组成的,GA 通常会得到一个基于初始种群的可能的最优解决方案。换言之,对某个优化问题应用 GA 方法时,肯定能得到一些可行解。很多情况下,期望得到的解决方案总是比生成初始染色体种群所用的那些要好。

回想一下我们对 Simon 的讨论以及他的观念——人类会满足于一个非最优解,只要它效果尚好,能满足他们对于成功结果的判断标准。生物体的自然过程和进化过程,也倾向于符合 Simon 的理论,而不是最优理论。很难想像,在人类种群的任何种族中,会出现哪怕一个同时具有托马斯·爱迪生的创意、戴安娜王妃的魅力和爱心、卡尔·萨根的智慧、迈克尔·乔丹的运动才能的个人。我们接受这个现实,我们的种族的进化并非最优的,而且我们也必须接受,用这种方法获得的解决方案也是非最优的。重要的问题是:物种的演化过程似乎是不断进化的,而且多数情况下,引起了物种内部生活质量的提高。同样地,每一代非最优解决方案,也都比前一代非最优解决方案有所进步,而且可以帮助我们增进对复杂问题的理解和处理能力。在这一点上,GA 在未来的管理人员面对的决策挑战中会起到重要的作用。

遗传算法的应用带来了一个重要的收益,那就是它们可以用于解决我们完全没有解决线索的问题。如果我们能够描述出一个优良解决方案的各个组成部分,并能给出 GA 能使用的合适度函数,我们就能得到一个或者几个用别的方法得不到的解决方案。这种"发明"解决方案的能力令 GA 胜过常规的专家系统——它们需要大量关于问题域的外部信息。换言之,GA 无需知道怎样解决一个问题,它只要能够识别一个"好"方案就行了。

GA 的最后一个优势是:它们解决问题的能力并不是来自于复杂的算法,而是来自于相对简单和容易理解的一些概念。这注定了 GA 比较容易被成员不具备很强技术背景的组织广泛接受。几乎每个人都上过生物课,"适者生存"的概念也在几乎所有的知识域中都有应用。大家都很熟悉遗传算法的背景概念,因此相对于其他的机器学习和人工智能技术,它们更容易得到接受和信任。

10.4 学习机的应用

学习机的应用领域

表 10-4 列出了机器学习系统(例如神经网络和遗传算法)已经得到成功应用的问题类型。虽然并不全面,但是列表的分散性已经证明了基于人工智能的系统能够应用的问题和领域的广泛性。

媒体报道过许多应用了 ANN 技术的建设项目。例如,日本钢铁公司建立了一个利用 ANN 技术的鼓风炉操作控制系统。这个系统中使用的神经网络装配了一些函数,使得它可以识别传感器数据和八种经验温度分布模式之间的关系,保证其适合于鼓风炉整体运行需要,同时还要识别并输出最能模拟系统的传感器数据输入的模式。神经网络学习得很快,而且学习后能达到高于 90% 的模式识别率。由于这套系统在试运行中表现特别优秀,日本钢铁公司已经打算把它引入除鼓风炉之外的其他业务中,包括故障诊断和其他的控制过程。

第二个例子是 Daiwa 证券公司和 NEC 公司的实验性工作,它们试图将神经网络技术应用于对股价图表模式的学习和识别,从而进行股价预测。NEC 已经完成了用于它的 EWS 4800 工作站的神经网络模拟软件,而且,它把股价图表模式学习限制于几十种主要

的股票，从而提高了这个软件的预测准确性。基于这些结果，Daiwa 计算机服务公司（DCS）——Daiwa 证券集团的一个附属信息处理机构，把 NEC 的系统移植到它的超型计算机上，并教它识别在东京证券交易所上市的 1134 个公司的股价图表模式。这样，DCS 将这套系统充分应用于它的股价预测行为。

另一个最近的例子是 Mitsubishi 电子的一个项目，它将神经网络技术与视觉技术相结合，建立了世界上的第一台视觉神经计算机系统，能够识别 26 个字母。这个系统包括一套将字母模式输出为视觉信号的发光二极管（LED）、光纤、显示字母模式的液晶显示器（LCD），以及接收光信号并读出字母的设备。这套系统达到了百分之百的字母识别率，甚至输入的手写字母有点变形也能够处理。

表 10-4　机器学习系统的典型应用

- 预测每年中不同时间和不同条件下对职员的需求数
  布鲁克林联合燃气公司就根据一年中的时间、预测的温度以及星期几来预测处理要求服务的电话的职员数目
- 预测一个工作申请者最适合什么类型的工作
- 预测哪些顾客会付账单
- 识别特殊的交易模式
  Ivan Boesky，一个违规操作的公司，就是这样被发现的
- 预测化学混合物的属性
- 诊断疾病
  已经存在一个神经网络，在诊断嗅觉紊乱方面胜过了专家系统
- 预测股票市场、期货市场或类似的市场
- 标记装配线上有缺陷的部件
- 利用过程中多处传感器的输入，调节工业过程
- 识别疾病（例如听力丧失）、生物体以及细胞（例如癌细胞和非癌细胞）
- 根据进入焚化炉的垃圾预测污染情况
- 预测销售和/或成本
- 预测一个公司的公司债券利率
- 对房地产估价
- 预测运动项目结果（例如赛马）
- 预测太阳黑子活动
- 预测患了某些疾病的病人的生存时间
- 寻找核密封装置中可能支持不住高压的裂纹
- 测试啤酒质量
  Anheuse-Busch 能确定其竞争对手的啤酒蒸气中的有机成分，准确率高达 96%
- 预测监狱同室者中，哪一个会受益于减刑方案

第四个例子是应用了神经网络系统的设备检测系统，目前还处在建设阶段，是由日本石油公司和 CSK 研究中心合作启动的。这个项目吸引了相当多的注意力，因为这是将神经网络系统应用于设备检测的第一次尝试。最初，这个项目的目的是用于对油泵设备进行振动分析。日本石油公司仅在它位于横滨（神奈川辖区）的 Negishi 石油精炼厂就有 5000 多个油泵，而且需要大量的熟练人员来维护这些油泵。公司决定将神经网络技术应用于对油泵的诊断操作，主要是为了节约劳力。

ANN 成功应用的另一个领域是债券评级。债券评级指的是债券被赋予标识的一个过程,这个标识用以区分债券发行者付清债券利息和票面价值的能力。举个例子来说,标准普尔公司可能对于投资级的债券给出不同的级别,从 AAA (偿付可能性极大) 到 BBB (经济衰退时可能无法偿付) 。这里的问题是,并没有一种必须遵守的规则来决定这些级别。评级机构必须考虑很广泛的因素,才能决定赋予发行者怎样的级别。其中有些因素,例如销售额、资产、负债之类的,可以很明确地定义,而其他因素,例如是否愿意偿付,就很模糊了。所以,给出精确的问题定义是不可能的。

机器学习系统常常应用的其他领域还包括信用积分和目标市场营销。信用积分用以根据已知的个人信息,鉴别信用卡申请人。这些信息通常包括工资、支票账户数目以及从前的信用历史。大银行和其他信贷者每年因为坏账损失几百万美元。对于哪个账户可能不被偿付的预测准确性只要提高一点,就能为大信贷者带来每年几十万美元的节约。为了解决这个问题,主要的银行和金融公司都积极寻求新技术和系统,来帮助进行信用预测。

机器学习的应用也可以帮助解决广告中一个长期存在的问题。多年以来,广告机构和其他公司一直试图区分目标市场,并进行针对性的销售。例如,一个出售人寿保险的公司在每月的信用卡账单中附加一份广告。它可先将一小部分广告发给顾客,并记录下来哪种类型的人会做出应答。一旦公司有了应答者的资料,它就能建立一个预测模型,分析潜在顾客。这样,一个人寿保险公司就能挑选出 100 万个比较可能购买人寿保险的信用卡持有者,并只把广告发给他们,而不是没有选择地将广告发给所有的信用卡持有者。成功的目标市场营销可以为大公司节约每年几十万美元的费用。

大通银行使用的基于统计的混合型神经网络,如我们在本章开头的小案例中讨论的,是美国人工智能技术应用中规模最大、最成功的之一。它在银行的战略计划中产生了一个关键成功因素 降低对上市公司或私人持有公司的贷款损失。大通对多数公司的业务包括了对它们信誉评估。大通每年借出 3 亿美元,早就开始寻找改善贷款评估体系的工具。这种评估制度降低了大通银行的风险,并得以寻求新的业务机会。对于银行来说,财务重组是一项很有前景的业务机会。

最近,许多研究和成功案例在各种国际研讨会和专业学术报纸、杂志上得以报道。在本章的末尾,我们给出一个典型的多目标神经网络应用的例子——NeuroForecaster。

机器学习系统的前景

当我们对神经计算及其他机器学习领域进行思考时,常常想到的一个焦点问题就是:我们到底能造出一个多么"聪明"的机器。就单方面的能力而言,我们已经拥有了在某方面超出人类的 ANN。发展能够反映复杂的人类行为 (例如作曲、话题广泛的交谈) 的大规模神经网络,在可预测的将来还不可能。

建立这样的系统的主要障碍在于获取有效的训练数据。人脑中信号传输的速度大约是 100 米/秒,而普通铜导线的传输速度至少是它的 100 万倍。虽然理论上计算机是可能达到人脑思考和学习速度的 100 万倍的,但是要做到这一点,系统必须能够以同样的速度获取数据信息。瓶颈是识别的局限性,这是人类要创造训练集,并将它们输入网络造成

的。同时,为了满足自己的需要,我们必须研制能学会支持更复杂活动的机器,例如医疗诊断、飞机驾驶、向外星球发送探测器及其他能提高我们生活质量的行为。

10.5　小　　结

合情合理地,我们对人脑的工作过程以及在计算机系统中模拟这一过程很感兴趣。然而,在我们这一追求过程中,人脑能够轻易处理的不清晰和非特指现象成为建立真正仿人机器的主要障碍。虽然如此,模糊系统、遗传算法和神经计算系统的出现,还是让我们向目标迈进了一步。

复习题

1. 模糊逻辑是怎样处理分类问题的?这与典型的分类过程有何不同?
2. 描述一下隶属函数是怎样依赖于复合集的内容的。
3. 解释模糊性和可能性之间的差别。
4. 列举并简单描述模糊系统的优势和局限性。
5. 什么是模糊集饱和?
6. 描述人工神经网络的基本构成。每一层的用途是什么?
7. 转换函数的用途是什么?
8. ANN 中的反馈是怎样起作用的?
9. 简单描述并比较有指导和无指导的 ANN。
10. 遗传算法相对于 ANN 的最重要优势是什么?
11. 描述遗传算法应用中的基因和染色体。
12. 描述 GA 处理过程中的步骤 初始化、评估、选择、交叉、突变。
13. 总结遗传算法的好处。

讨论题

1. 模糊逻辑能对表面上相互排斥和矛盾的复合集的内容进行建模,这种能力有用吗?什么情况下它是有用的?
2. 举例说明本章中描述的神经节点的典型组成结构。
3. 人工神经网络能很好地处理噪声或不完全数据。某些情况下,它比其他的技术抽取数据中模式的性能更好,描述这样的情况。
4. 突变是怎样提高遗传算法达到满意解的能力的?突变是必需的吗?

5. 描述染色体数量对遗传算法性能的影响。

NeuroForecaster 与 GENETIC

NeuroForecaster 是一个在 Windows 环境下运行、具有友好的用户界面的先进商业预测工具。它封装了一些最先进的技术,包括神经网络、遗传算法、模糊计算以及非线性动态规划。它有如下特长:

1. 时间序列预测 (例如,股票、外汇市场预测,GDP 预测)
2. 分类 (例如,选股,债券评级,信用评估,资产估价)
3. 指标分析
4. 识别有用输入指标

用 NeuroForecaster,你可以建立任何规模的神经网络——惟一的约束就是你的计算机的存储限制。它也是分析神经网络体系 (隐含节点、隐含层数目,转换函数) 对问题求解有效性的优秀工具。它可以接受数据输入属性、模式、代码、技术指标或者基本指标,并且可以把它们结合起来,创建单变量或者多变量模型。像其他的数据工具一样,它不能处理描述性的信息,例如新闻、流言或者财政政策,除非这样的信息伴有用其他指标反映的数据信息,或者可以量化。

NeuroForecaster 是一个通用的预测工具,具备适应性学习能力。你可以将它植入一个自动交易系统,只要你用交易策略很好地训练它生成买/卖信号。

GENETICA Net Builder 是另一个神经计算应用程序,NeuroForecaster3.0 或以上版本都能支持它。它能搜索输入数据的最优组合和能预测的范围 (例如,要作出预测还需要多少步),而且能通过进化和基因搜索,自动生成并优化预测器结构和控制变量。

GENETICA 与 NeuroForecaster 的训练过程能够无缝连接——它能决定什么时候暂停训练,清除性能较差的神经网络,并生成下一代网络,继承停止运行的双亲已生成的知识。

强大而与众不同的特征

- 能读取多种格式的文件,例如 MetaStock,CSI,Computrac,FutureSource,ASCII,以及 VisuaData,这些会显示在它内置的数据表中。
- GENETICA Net Builder,它可以搜索最优输入数据组合和预测范围,并建立最好的预测神经网络。
- 重新调整范围分析和 Hurst 指数,能揭示隐藏的市场周期,并检验可预测性。
- 相关性分析,计算出相关系数,用来分析输入指标的重要性。
- 权重和 AEI (累计误差量) 柱状图,指导学习过程。

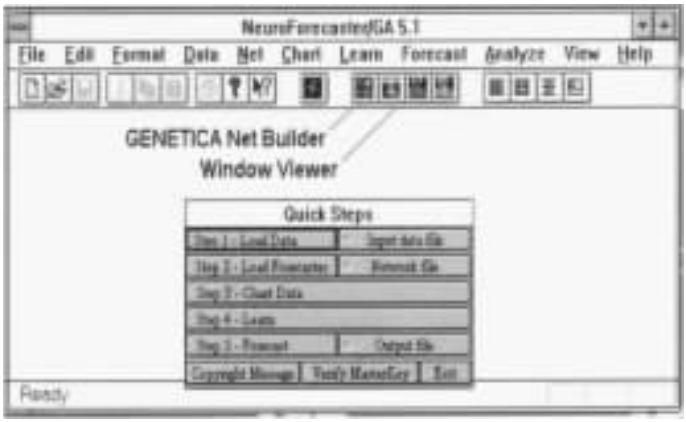


图 10-A1　NeuroForecaster **主界面和快捷按钮**

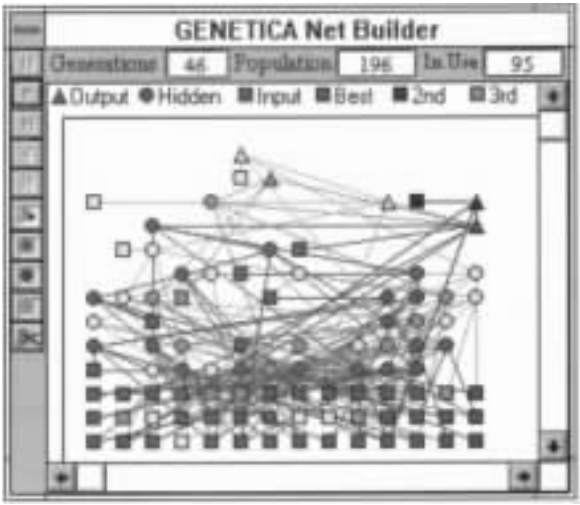


图 10-A2　**进化中的神经网络种群**

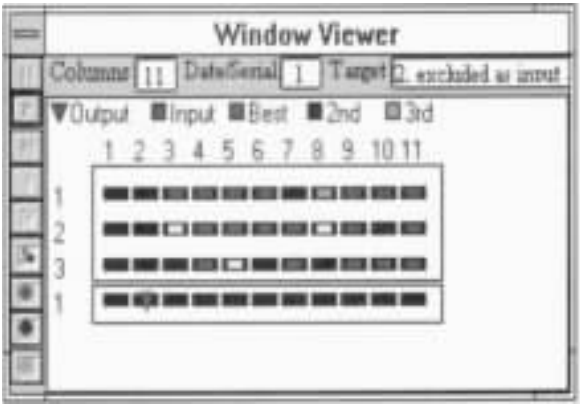


图 10-A3　GA **找到的输入组合和预测范围**

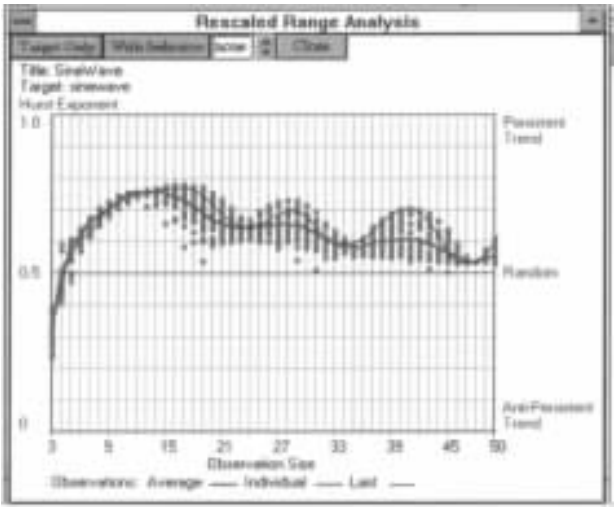


图 10-A4　**为确定隐藏周期的重新调整的范围分析**

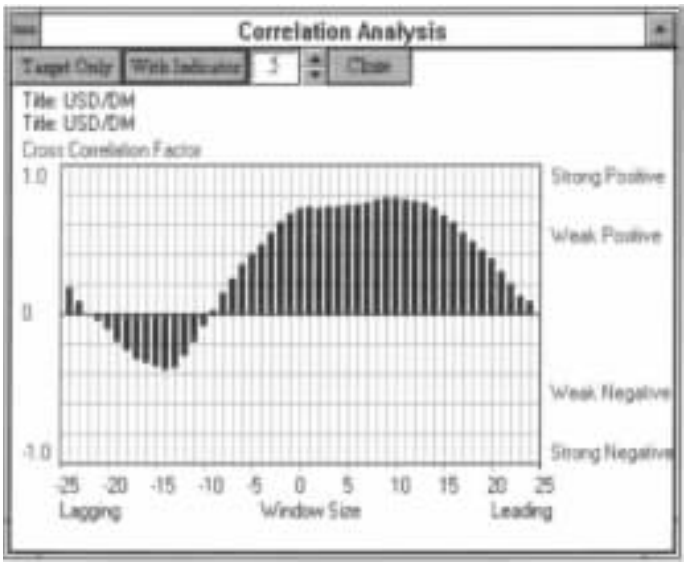


图 10-A5　**确定关键因素的相关性分析**

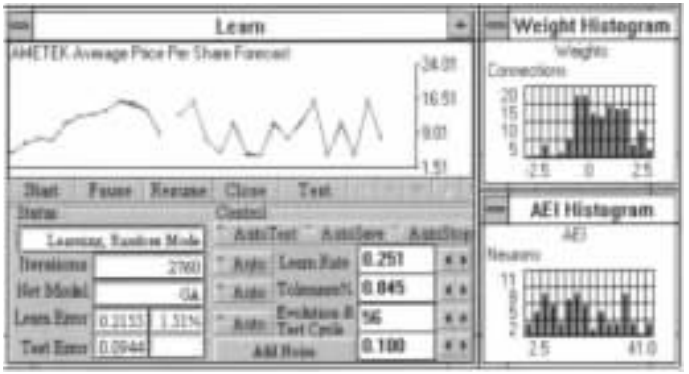


图 10-A6　**带有柱状图的学习过程**

图 10-A7　**具有用户友好界面的主界面**

后向传播算法

后向传播的基本目标是使得神经网络的输出与训练数据集合中的真正解的误差最小。①

设定一个计算使用的转换函数,我们就可以把给定一个节点的权重调整为目前节点的权重,公式如下:

$$\omega_{j(t+1)} = \omega_{j,t} + (\lambda)(\varepsilon\omega_{ij})(n_i)$$

其中 λ 是学习因子,下标 t 表示网络经过调整的次数,$\varepsilon\bar{\omega}_{ij}$是节点 n_j 对权重 $\bar{\omega}_{ij}$的改变的灵敏度。

回想一下,对于给定节点,总的输入描述如下:

$$s_j = \sum n_i \bar{\omega}_{ij}$$

其中 s_j 代表对于此节点的全部输入,n_i 代表前一层第 i 个节点的输出值,$\bar{\omega}_{ij}$代表第当前节点和前一层之间的输入联系的权重。

这种加和的结果被传递给一个非线性的转换函数,从而产生第 j 个节点的最终输出结果 n_j。典型的转换函数形式是指数函数,形式如下:

$$n_j = \frac{1}{1 + e^{-sj}}$$

有了这些方程,我们就可以确定 ANN 按照以下步骤遍历一次之后的总误差了:

- 计算 (RMS) 输出层的误差 E:

$$E = \frac{1}{2}\sum_{\text{output}} (n_j - d_j)^2$$

其中 d_j 是输出节点理论上的输出值。

- 计算每一个输出节点的误差值,并确定误差对节点输出值变化的灵敏度:

$$\varepsilon o_j = \frac{\partial E}{\partial n_j},$$

$$\varepsilon o_j = (n_j - d_j).$$

- 确定输入输出改变之间的关系:

$$\varepsilon s_i = \frac{\partial E}{\partial n_j},$$

$$\varepsilon s_j = \frac{\partial E}{\partial s_j}\frac{dn_j}{ds_j},$$

$$\varepsilon s_j = \varepsilon o_j n_j (1 - n_j).$$

- 然后,我们计算对每个权重的调整量,$\bar{\omega}_{ij}$,从当前层的下一层指定节点 n_i 到当前节点 nj 的权重:

$$\varepsilon\bar{\omega}_j = \frac{\partial E}{\partial \bar{\omega}_{ij}},$$

① 本部分内容引自 Dhar 和 Stein,1997。

$$\varepsilon\bar{\omega}_{ij} = \frac{\partial E}{\partial s_j}\frac{ds_j}{d\bar{\omega}_{ij}},$$

$$\varepsilon\omega_{ij} = \varepsilon s_j n_i.$$

- 我们对隐含层的节点进行这一操作，只要将所有的输入相加，就得到低一层隐含节点的误差。变量 εh 代替了前面方程中的 εo，以区分输出层误差和隐含层误差：

$$\varepsilon h_i = \frac{\partial E}{\partial n_i},$$

$$\varepsilon h_i = \sum_j \frac{\partial E}{\partial s_i}\frac{\partial s_j}{\partial n_i},$$

$$\varepsilon h_i = \sum_j \varepsilon s_j \bar{\omega}_{ij}.$$

通过这个过程，我们可以将递归地将误差向后传递，经过整个 ANN，调整所有的权重，以最小化网络的整体误差。

第 11 章

数 据 仓 库

学习目标

- 解释数据仓库的目的和它的特性
- 解释操作型数据存储、数据集市和数据仓库之间的区别
- 简要描述数据仓库体系结构中的各个相互关联的元素
- 描述元数据在数据仓库中的作用
- 分析实施数据仓库时的各种挑战
- 描述不同的数据仓库技术以及数据仓库的未来发展

小案例

Capital One 公司

由于目前海量定制化（mass customization）的需求，信用卡行业已经从用同一种产品服务于所有的客户转变成为了为不同的顾客提供上千种产品，从而满足各个细分市场顾客的需求。Capital One 公司，美国十大信用卡提供商之一，是在信用卡行业已经成为一个真正的信息产业的前提下成立的，并且它的海量定制化已经成为其增长和获取利润的主要因素。

Capital One 使用信息技术，实施科学的测试方法，来定位赢利性的业务。一个战略是从数据仓库中挖掘有用的数据而得到的。他们把其目标市场分段，并且建立检测单元。他们利用检测单元对其目标市场的信息进行挖掘，并把结果进行记录。通过整体的经济评估，可以判断此战略的潜在赢利能力。评估的结果可能继续进行更大规模的测试，包括修正以及重复，或者把这个战略抛弃。不管得出的结果是什么，从这些检测中得到的数据都将作为数据仓库的一部分存储起来，为评估以后的战略所使用。

数据容量

支持 Capital One 公司的市场运作所需要的数据量是很大的。分割到什么程度取决于所有数据的层次。并且，一个有经验的预测模型，又需要很多底层的、细节的数据和方法。现在，他们的数据仓库中包括将近 900 万条活动账户数据、4000 种产品类型、以及 20000 组试验数据，包括过去八年内的原始数据，总的数据量超过了 1 万亿字节。

数据的复杂分割可能需要很多步骤，每一步都需要建立一个临时文件，这个文件在执行以后的步骤时是需要被存储的。由于每年有许多专家进行上千种的测试，为了支持这些分析工作，需要超过 500G 的磁盘空间。分析者在分析时，建立起很多临时的表，这些表包括上百万的记录数据，由于数据分割的复杂性，表格需要进行定期的大量的更新。

数据结构

Capital One 公司很少对同一个问题进行两次以上的分析，因此它缺乏统一的数据入口，使得他们不可能使用标准的决策支持系统体系（比如星形方案），它要求使用者把他们的问题限制在一个预定的有限维空间当中。面对大量的要进行分析的数据资源，在目前的技术下，有限的维数不可能解决问题。

大多数数据的相互关联性差，粒度低，很少经过预处理。为了保持数据结构足够的灵活性，分析者们必须忍受某些查询的较长的周期以及更加复杂的查询语法。

数据仓库产品

目前市场上存在的大多数数据仓库工具都不能支持海量定制化环境。在数据仓库的数据获取方面，大多数工具不能够处理复杂的、完全不同的、海量的数据资源。随着市场上提供的工具不断地提出更多的假想环境，这种状况将会有所完善。但是现在，像 Capital One 公司这样的企业，还是更偏好于自己开发。

数据仓库的潜能

寻找一个能够适应 Capital One 公司独特文化的 IT 专家是一项持续的挑战。一个成功的数据仓库的主要特征包括技术与商业运作的完美平衡、团队作业的能力以及强大的灵活性。对这些类型的需求好像是没有止境的，因此，使得新的数据仓库的性能不断提高。

使用海量定制化的概念，数据仓库远远不限于传统的其他 IT 项目。在存储高质量数据的数据仓库的支持下，Capital One 使用海量定制技术和科学试验的方法，在八年内把他们的客户信息从 100 万扩展到了 800 万。

海量定制是数据仓库的主要用途，使得它成为一个业务过程的完整部分。就好像人的肌肉，数据仓库在积极使用时会增强力量。每一次新的测试和新的产品，都会往数据仓库中增加新的有价值的信息，使得分析者能够通过这些或成功或失败的历史数据得到帮助。

在 20 世纪 80 年代，全球的商业和工业都在疯狂地进行自动化转换。如果要发展，就必须进行计算机化。在这方面，办公室成为系统分析和软件工程的先锋，而一般的工厂还处在十年前的状态。这种迅速向自动化商业模式靠拢的过程，为组织提供了发展的机会，并且使企业认识到这样做的好处——增加利润和减少成本。虽然在这段时间内，很多团体企业都在改善它们的商业模式，但是对整个企业的决策支持系统都还处在操作型和职能型的层次上。这样的信息系统每天、每周以及每月都会及时迅速地产生出各种基本报表。这些周期性产生的数据太旧，太细节化，太多，太傻。从某些方面看，使用这些报表也许会比不使用这些信息产生更多的消极作用。与此同时，全球范围内的市场变化非常迅速。在 80 年代末，很多组织都很清楚地认识到只有拥有了分析、计划，并对市场环境的变化迅速相适应的能力，才能够在 90 年代生存。为此，各个层次的管理者都需要获得更多更好的信息。

除了这些对更多信息的需求，组织每天都或多或少地产生出关于企业各个方面的成千上万的数据——包括大量的客户信息、产品信息、业务信息，以及其他一些需要在事务层次上需要组织的没有经过格式化处理的信息。过去，这些数据都被“锁在”许多计算机系统中，被喻为“数据监狱”。

建立一个获取、处理以及在企业内部存储数据的机制，使得决策支持成为可能。数据仓库 (data warehouse，DW) 的定义就是信息技术用来满足这一需求的部分。这是一个非常简单的概念，随着时间的推移，它发展成在全球市场上一个组织成功和保持稳定的主要因素。数据仓库的概念的本质是认识到，用于自动商务处理的功能性系统的特性和使用模式与决策支持系统是根本不同，但是，它们之间还是存在联系的。数据仓库提供了一个整合由原来不完整的信息系统提供的数据的工具，一个操作型数据仓库组织并存储所有的可以获得的数据，用来进行对历史数据的分析。数据仓库的目的是重整大量的企业内部以及外部的数据，在不改变信息的情况下使数据趋于一致。

11.1 存储、仓库和集市

数据仓库是一个新兴的概念，一些定义着重强调数据，另一些概念着重强调人、软件、工具和业务过程。大家公认的数据仓库之父 W. H. Inmon，从可衡量的属性出发，提出了数据仓库的一个非常清晰和有用的概念，如下所示：

> 数据仓库是大量的、面向主题的数据库的集合，具有支持决策的功能，数据仓库中每一个数据单元都是稳定的，与一定的时间相关联。（Inmon 1992a，5）

Inmon 的对数据仓库的定义包括两个潜在的假设：(1) 物理上，数据仓库与其他操作型系统是分离的；(2) 数据仓库中存储的既有集合的数据又有极微的数据，与在线处理过程中的数据相互独立。

数据仓库需要有一个独立的环境。在很多情况下，很多方面，譬如决策支持和分析，在操作型环境中开发的系统是不完全的，主要是由于在这些环境中产生的数据类型、数量和数据质量都不适于进行历史分析。数据仓库中的数据必须是一致的、综合的、定义完全的，最重要的是要有时间标记。另外，内外环境的大量数据和方法的结合，也需要独立的数据仓库系统。表 11-1 对操作型数据与数据仓库的特性进行了比较：

表 11-1 操作型数据存储和数据仓库的特性

特 性	操作型数据存储	数 据 仓 库
如何建立	一次只有一个应用或者功能领域	多重的主题领域
使用者的要求	在逻辑设计之前就具体定义	经常是模糊的和冲突的
支持的领域	日常业务运作	管理活动中的决策支持
访问类型	在一个查询中检索相对少的几条记录	扫描大量的数据集，来检索一个或者多个查询的结果
访问频率	对少量的数据访问频率较高	对大量的数据访问频率较低
数据容量	与日常运作交易的数据量相当	比每天要处理的数据量大得多
保存时间	满足日常运作的需要	保存的时间不确定，需要满足历史报告、比较和分析的需要
数据更新时间	实时，以分为单位	一般都是一些静态的时间点
数据有效性	可能需要较高和及时的可用性	及时可用性不是非常重要
分析的单元	小的、可管理的、事务处理层次上的单元	大的、不可预测的可变单元
设计重点	高性能，有限的灵活性	高灵活性，高性能

资料来源：Bischoff 和 Alexander (1997)。

“数据的仓库”可以非常方便地扩展为一个完整的数据仓库环境。由于本章主要讲述数据仓库本身，我们必须简短地区别数据仓库环境的三个额外组成部分：数据存储 (data store)，数据集市 (data mart) 和元数据 (metadata)。

数据存储

操作型数据存储 (operational data store,ODS) 是数据仓库环境中最基本的组成部分,它的主要功能是每天存储各种应用程序的数据。在数据仓库环境中,为数据仓库提供必需的原始数据。

ODS 中的数据组织形式是面向对象的、易变的、近期的,主要针对于顾客、产品、订单、政策等等。通常 ODS 的数据来源于一个或多个遗留系统,这些数据库管理系统与那些进行数据过滤、变形和整合的系统相联接。虽然 ODS 中的数据非常适合于一个或者多个遗留系统的分析,但是与其他不相关的系统就不容易结合了。因此,许多数据存储 (内部和外部的) 为了能够用于分析,都必须进一步整合到数据仓库中。

数据集市

虽然把不同的但有联系的数据集中起来是一个很有吸引力的概念,但是,在整个组织内进行集中的成本是十分昂贵的。对整合的特殊类型的数据的需求只存在于公司某些特定的业务单元。每个单元都有区别于其他单元的特殊需求。数据集市就是适用于各个公司的小型的、低成本的数据仓库。

数据集市经常被看做是开发数据仓库的一种方法。它的规模比较小,直接向一个独立的数据使用者提供数据比向整个组织的一个特定部门提供数据要容易得多。虽然,"微型集市"的概念既流行又有用,但是也存在缺点:它不能够从整个企业的范围内进行规划,从而使得数据集市成为一个个信息孤岛,其他的部门无法使用这些数据。假设一个数据集市是在整个企业的层次上进行构筑的,它可以提供低成本的数据存储,随着一段时间的发展,规模不断扩大,它有可能发展成为整个的数据仓库环境。

元数据

数据仓库环境的另一个部分就是元数据。遗留系统一般不能保存关于数据特性的记录,比如说,存储了什么样的数据,数据的存储位置,数据来源以及如何获得这些数据等等。元数据是关于数据的简单数据——也就是说,是关于数据仓库的信息,而不是数据仓库内存储的信息。我们将在 11.3 节中对元数据进行详细的介绍。

数据仓库环境

图 11-1 描述了组织内的数据流,以及数据仓库环境中,各部分相对其他信息系统的地位。

如图所示,组织的遗留系统和系统外部相关数据存储是数据仓库和数据集市的主要数据来源。数据在不同数据存储中传输的时候,进行一个清理和转化的过程,从而统一整合到数据仓库中。同时,系统还收集元数据,并与数据仓库的数据相关联,这样潜在的用户能够知道数据仓库中数据的来源和特性。最后,从数据仓库或数据集市中产生一个或多个个人数据仓库,应用于独立的分析。在下面的章节中,我们将进一步探讨与数据仓库环境相关的过程和体系。

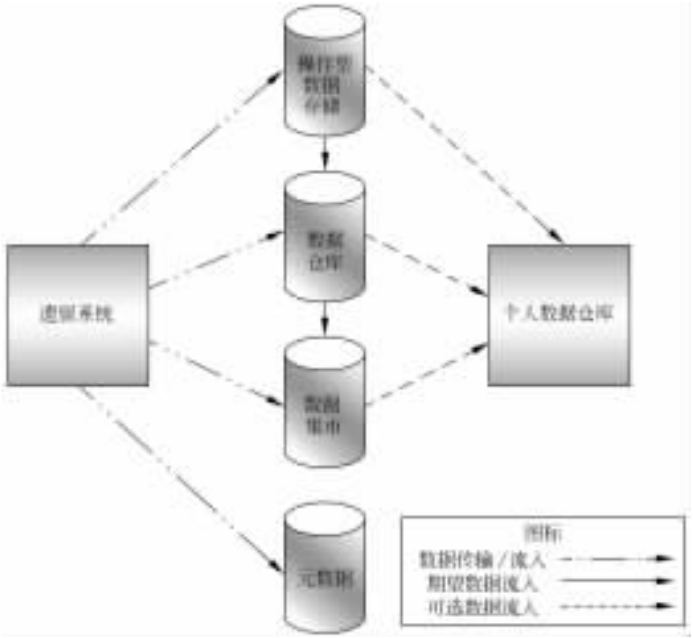


图 11-1　组织的数据流和数据存储组件

数据仓库的特性

到现在为止,我们已经解释了为什么数据仓库是决策支持中的一个重要革新。我们定义的数据仓库是：

- 面向主题的
- 数据整合的
- 与时间相关的
- 稳定的

表 11-2 概括了数据仓库的基本特征,并对每一个特征进行了简要说明。

表 11-2　数据仓库的特性

• 面向主题：数据以用户需要的方式进行组织
• 数据整合：所有的名称和单位都进行了统一
• 不可变性：数据以只读方式存储,不随时间变化
• 时间变量：数据不是当前的数据,而是时间序列数据
• 综合的：操作型数据映射为了制定决策可以使用的格式
• 海量的：时间序列数据集一般数据量很大
• 元数据：关于数据存储的数据
• 数据源：数据来自内部和外部的未经过整合的操作型系统

面向主题

数据仓库的第一个特性就是面向组织的主题。这与公司的遗留系统中面向功能的各

种应用程序是有明显不同的。图 11-2 描述了两者的不同。

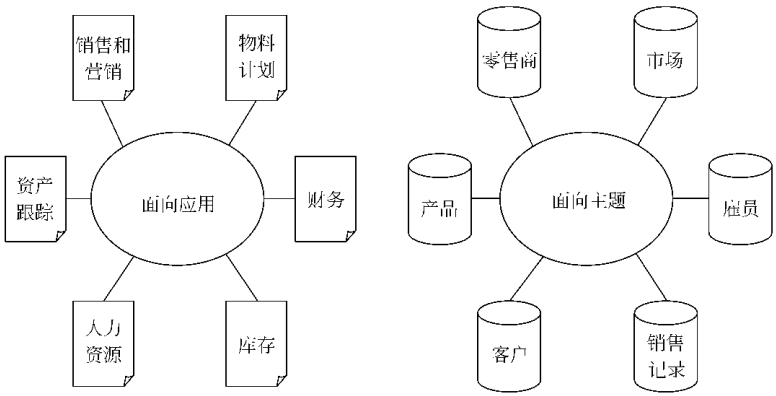


图 11-2　面向应用与面向主题的比较

如图所示，面向应用的系统一般是围绕库存、人力资源等过程和功能设计的。这些系统要求的数据是完全针对那些特定的功能的。与此相反，数据仓库中存储的数据是面向决策支持的，主要是围绕公司有关的主要目标，例如顾客和零售商等。

这种独特的面向主题，使得数据仓库与一般的应用程序存在很多不同点。比如说，在设计应用程序的时候必须同时考虑应用程序的设计和数据库的设计。而数据仓库就只需考虑数据建模以及数据库的设计，而无需顾及过程的设计。

数据仓库还有一个独特的特性就是：数据之间是相互联系的。数据表是建立在商业规则的基础上的。例如“每一项订货都必须有客户参与”这个规则，它建立了订货信息表与客户信息表之间的联系。对客户信息的一次查询也许会与订货信息表中很多项记录有关。同样，对订货信息表的查询也会涉及到顾客信息表的一项记录。数据仓库中的数据是具有时间跨度的，很多商业规则被表示为数据仓库中两个表或多个表之间的联系（Inmon，1992a）。

数据整合

Inmon（1992b）提到的数据仓库最重要的特性是：其中所存储的数据都是整合的。这种整合是通过惯例命名、度量属性、精确度和一般集合体的一致性表现出来的。图 11-3 描述了操作型应用程序和数据仓库中数据整合性的不同。

一致命名和度量属性　设计应用程序的时候，对变量的命名是自由的，不论是在数据字典中还是在屏幕显示上。这种自由性使得应用程序的数据在整个组织范围内缺乏一致性。举个例子，一个数据元素可以在应用程序中表示成任何类型。在一个程序中，该变量可以叫做 M 和 F，在另一个程序中，它可以被定义为 1 和 0，1 和 2，还有可能是 0 和 1，在下一个程序中，又可能表示为一个 x 和一个 y。只要在应用程序中能够分清这些变量，并且程序的使用者能够理解它们的含义，就不会出现比较严重的问题。但是，一旦那些与应用程序相联系的数据库装载入数据仓库时，采用什么命名方式就必须进行确定了，需要使用命名方式的转换程序。这样，只要数据的表示方式一致，在数据仓库中就可以随便使用

这些数据而不出问题了。为了保证这种一致性,每一个数据源必须经过一个转换过程,从而使得数据处于一致的状态。

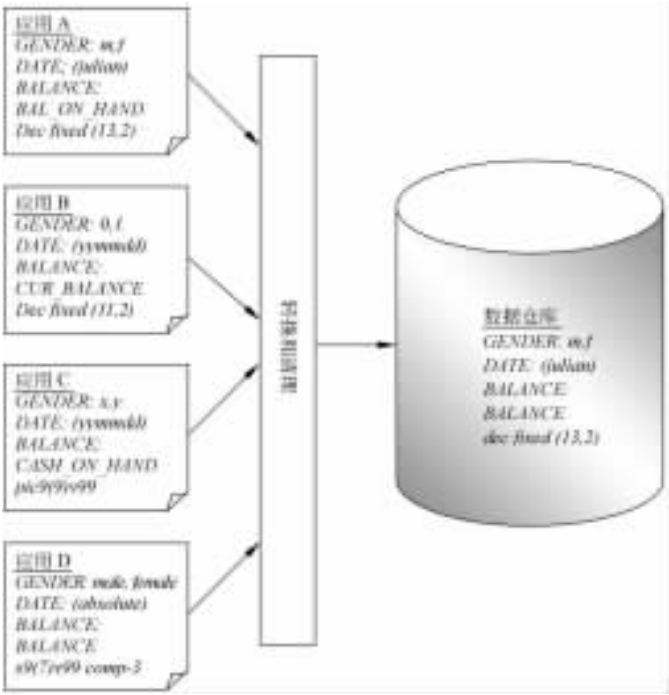


图 11-3 ODS 数据的整合与转化

数据整合的另一个结果是对于不同数据库中相似的数据建立统一的单位。不仅仅要对装入数据仓库中的数据进行统一单位,而且还要对最终数据统一单位。在一个数据库中,一个"雇员任期"可能用周表示,也可能用月表示,或者用年,十年表示。像这样的名称相同的数据元素,在装入数据仓库中时必须进行单位的统一。在统一单位的同时,还需建立一个标准,允许决策者和 DSS 使用数据仓库。某些情况下,很容易表示特定的数据,例如温度,只有三种表示单位:华氏、摄氏和开氏。选择哪个取决于数据的设计。在另一些情况下,数据的表示方法则很复杂,这时,或许有必要把数据转化成不标准的单位的数,满足数据跨度的灵活性。在雇员任期的例子中,我们把单位用天来表示,因为这样,所有的数据在输入和输出时都比较容易转换。对这一概念进行扩展,我们不存储任期了,而是改为存储雇员首次上班的时间。采用这种方法,不论需要的量纲是什么,我们都可以很容易地计算并转化出来。

不论如何,数据必须是以一种整合的、规范的形式存储在数据仓库中。在数据仓库中 DSS 能够针对数据进行分析,比针对确定性的问题的分析要重要得多。

时间变量

在一个操作型应用系统中,很多数据库中的数据都要求是精确的。当查询信用卡余额时,用户要求得到的数据是当前的,系统一般不会说用户的余额在三周前是 459.00 美

元。但是,在数据仓库领域,数据都假定为精确的,但不需是现在的。一般来说,数据是在装载的那一时刻是精确的。因此,数据仓库中的数据是与时间相关的。

数据仓库中数据的时间变量有不同的表示方法。一般地,数据仓库中数据的时间跨度比较长——5 年到 10 年。相比较来说,应用系统中的时间跨度就小的多——当前的或 60 天到 90 天内的。这是应用系统中大量的事务处理过程所要求的。但是在数据仓库环境中,一般需要分析一段长时期内的数据,从而寻找关系或进行预测。

数据仓库中另一个显示时间变量的地方就是记录的主键。每一个主键必须或显式或隐式地包含时间变量(天、周、月、年等)。在显式表示中,时间标记必须附加在主键上(例如 CUST-ORDER + 0992,主键的后半部分表示月和年)。在另一些情况下,时间标记以隐式方式显示。例如一个总是月底或季度末转载的数据,其名称以那个时间段命名。不管用什么方法,数据仓库中的每一条记录都必须包含时间标记,以供将来进行分析。

数据仓库中数据时间变量的另一个方面是,数据一旦被记录,将不可更改和变化。在这方面,数据仓库中的数据可以看作是一系列连续的相片,那些不精确的相片会被精确地覆盖。但是一旦精确的相片被确定下来,则不可更改。

不可变性

为了保证数据仓库中的数据不可改变和更新,在应用系统中的插入、删除和更新操作在数据仓库中是完全不存在的。在数据仓库中,只有两种数据操作方法:数据装载(data loading)和数据访问(data access)。

这种数据操作的简单性,使得数据仓库与应用系统有着显著的不同。在应用系统的设计中,必须以数据完备性、更新异常为重点,因为应用程序的编程必须要求数据高度的完备性。在数据仓库中则不必考虑这些,因为不会有数据更新。数据仓库的最优数据访问的设计方式,是不能用于应用系统设计的。回想一下,在数据库设计中为了使得数据冗余最小,数据库需要符合一定的范式要求①。在这方面,三范式消除了记录间的数据依赖关系,在应用设计中,设计者无需存储所有可能的记录,如果需要某些数据,可以根据所存数据很容易地计算出来。但是在数据仓库中,设计者发现存储很多操作数据中所没有的计算结果和概括信息却是非常有用的。例如,通过在操作型数据库中合并一些日常的销售数据,得出每周或者每月的销售数据,并存储在数据仓库中是非常有用的。

由于数据仓库环境中处理比较简单,因此所需要的技术也比较简单。在操作型系统领域,支持事务处理、数据恢复、回滚、检测、修正死锁的技术是非常复杂的。在数据仓库中,这些都是不需要的。

数据冗余问题

数据仓库环境中需要考虑的,比如数据范式要求等问题要少得多,而这在操作型系统领域中,对于减少数据冗余却是十分重要的。因此,这意味着数据仓库中以及数据仓库与操作型系统之间存在大量的数据冗余现象。Inmon (1992a) 指出,他不同意以上的说法。

① 如果读者想了解这方面的知识,可以查阅任何有关数据库设计方面的图书。

操作型应用数据库中的数据装入数据仓库中时,只有很少的数据冗余产生。下面几点支持这个观点。

首先,考虑到从操作型数据库中装载入数据仓库的数据都必须经过过滤和转换。由于这个过滤的过程,大量的操作型数据库中的数据不能装入数据仓库。只有那些 DSS 和 EIS 需要的数据才能装入数据仓库。当然,数据仓库中存贮着很多操作型数据库中没有的综合性数据。

再考虑数据仓库和操作型数据库的时间轴的根本的不同。数据库中的数据都是比较新的,而数据仓库中的一般是较早的。从时间轴的方面看,操作型数据库和数据仓库环境之间的交迭和冗余是很少的。

最后,再回想一下在操作型数据库中的数据在装入数据仓库时常需要进行转化,由于这项转换功能,大部分存入数据仓库中的数据已不同于原操作型数据库中的数据 (至少在数据完整性方面)。

根据以上事实,Inmon 提出这两种不同环境之间的数据冗余是很少的,只有不到 1%。

11.2 数据仓库的体系

数据仓库体系 (data warehouse architecture,DWA) 是数据仓库中所有的数据结构、通信、处理过程和表达的总和,用于企业内部最终用户的计算。图 11-4 描述了组成 DWA 的各个相互联系的元素。

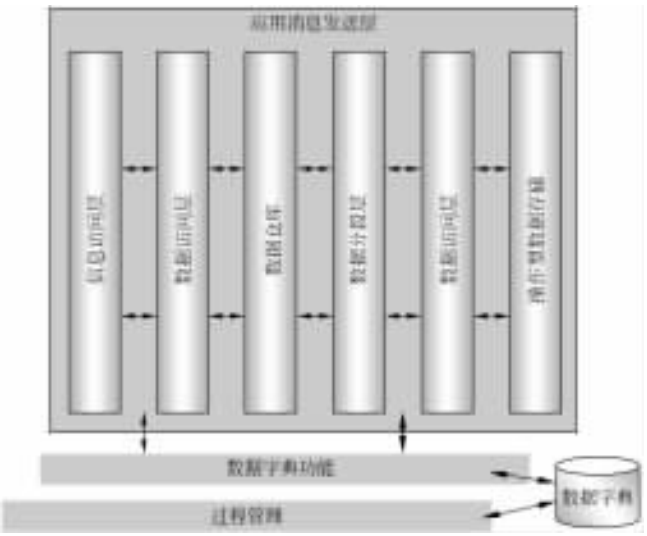


图 11-4 数据仓库体系的组成

操作和外部数据库层

操作和外部数据库层 (operational and external database layer) 表示数据仓库的数据源。这一层主要包含操作型事务处理系统和外部备用数据库。数据仓库的目的就是把锁在操

作型数据库中的信息释放出来,与其他地方的信息进行混和。从这一层看到的数据仓库的另一个目的是减少对操作型数据库的影响。换句话说,这个必要的软件层使得用户不必考虑访问数据库的操作型应用的执行过程。

很多大的组织都从外部数据库中获取数据,Web 和 Internet 使得企业能够容易而又经济地获得和使用外部数据。例如人口统计的、经济计量的、竞争的和消费趋势的数据都可以很容易地在公众网上获得。这些数据与 ODS 用同样的方式,进行抽取和转换后,被装载入数据仓库。

信息访问层

DWA 的信息访问层 (information access layer) 是直接与最终用户打交道的一层。它是一种工具,最终用户在日常工作中可以用来提取和分析数据仓库中的数据。这一层包含显示和打印分析和表达中的报表、电子表格、图形和图表的软件和硬件。信息访问层可以用来设计使用数据仓库数据的决策支持系统,从而支持组织中各种类别的决策制定过程。

数据访问层

如图 11-4 所示,数据访问层 (data access layer) 是连接操作型信息访问层与数据仓库本身的一种接口,或是说"中间人"。这层包括数据仓库中所涉及的不同的数据库,为数据仓库用户访问数据提供方便。数据访问层不仅在硬件上包含了大量的数据库和文件系统,而且还包含了各种制造商和网络协议。数据仓库成功的关键之一就是能够给用户提供通用的数据访问。也就是说,至少在理论上,最终用户无需考虑在什么地方、用什么软件,都能够获得所需的企业中的任何数据,这就是数据访问层的功能。

元数据层

为了实现通用的数据访问,绝对有必要保留一定形式的数据字典或者元数据仓库的信息。11.1 节中我们已经简要地讨论了,元数据是关于数据仓库中存储的数据的数据。例如,元数据中包括了数据存储的位置、使用的规则以及数据的来源等信息。

过程管理层

过程管理层 (process management layer) 主要着重于调度数据仓库的建立以及元数据的维护所必需的各种任务。这一层可以看做为了保持数据仓库是时新的而必须进行的过程的调度员或者高级控制员。这层的主要功能包括:定期从操作型数据存储中下载数据、总结操作型数据、访问和下载外部数据源,以及更新元数据等等。

应用消息发送层

应用消息发送层 (application messaging layer) 用于在企业的计算机网络中传递信息。它也被看做一个"中件 (middleware)",但是它不仅仅包括网络协议和请求路由的功能。举例来说,它可以使得操作和信息的应用与数据的格式相隔离,从而形成特定的数据格式

与分析工具需要的特殊数据格式之间的无缝的接口。这一层还可以用来收集事务处理信息,然后在特定的时间传送到特定的位置。在这个意义上,应用消息发送层可以看做是数据仓库底层的传输系统。

物理数据仓库层

物理数据仓库层 (physical data warehouse layer) 是组织内用于决策支持的数据实际存储的地方。在某些情况下,我们可以简单地认为数据仓库就是虚拟的和本地的数据。这是因为,在有些情况下,数据仓库可能没有实际存储通过它访问的数据。

数据分段运输层

DWA 的最后一部分是数据分段运输层 (data staging layer)。数据分段运输层 (有时指拷贝或复制管理) 包括选择、编辑、小结、合并以及从操作型和/或外部数据库中装载数据仓库和信息访问数据的所有过程。

数据分段运输层通常包括复杂的程序。但是越来越多的数据仓库工具使得该层的复杂性越来越小。数据分段运输层还包括数据质量分析程序以及其他一些诸如过滤器的程序,来识别操作型数据中的模式和结构。

数据仓库的分类

如前面所提到的,数据仓库虽然可以看做各种分析和决策活动的数据源,它可以不是访问的数据的物理位置。现在建立数据仓库的技术有很多,有三种基础的结构:虚拟 (或点对点) 的数据仓库、中央数据仓库和分布式数据仓库。

在介绍每一种结构之前,需要说明的是这三种方式都不是数据仓库最好的结构。每一种方式只适合特定的需求。数据仓库的设计一般都包括所有这三种方式。

虚拟的数据仓库

虚拟 (virtual) 或点对点 (point-to-point) 数据仓库允许最终用户使用数据访问层上的工具直接访问操作型数据库中的数据。这种方式不需存储和保留数据,使得数据冗余最小,同时提高了灵活性,但是这种方法会产生大量的不可预期的查询,从而降低了操作型应用系统的性能。

当最终用户很多,而且对操作型数据需求不确定、请求的频率较低的情况下,虚拟数据仓库经常可以用做初始的战略。虚拟数据仓库使得最终用户访问他们想要的操作型数据的成本比较低。

中央数据仓库

中央数据仓库 (central data warehouse) 是很多人第一次听说数据仓库概念时最先想到的。中央数据仓库用一个单一的物理数据库来存储所有的各种功能领域的数据。这种方法多用于有大量的已经连到中心计算机或网络上的用户,他们对信息数据往往有相同的需求。中央数据仓库要存储任何时期、多个操作型应用的数据。

中央数据仓库是真正存在的数据仓库。存放在数据仓库中的数据在物理上保存在一个地方,从这个地方访问,并且经常装载和维护。这是三种方式里面最普遍的一种。它也成了数据仓库实施的实际标准,因为它提供了大量的创建和操纵工具。

分布式数据仓库

分布式数据仓库 (distributed data warehouse) 就像它的名字一样,其各个组成部分分布在不同的物理数据库中。现在,越来越多的组织将决策的权力下放到组织的较低层次上,因此反过来将决策所需的数据推放到局域网或者本地计算机中,以满足本地决策者的需要。许多旧的数据仓库实施都采用了分布式的方法,因为最初创建多个小规模的数据仓库比建立一个完整的大的数据仓库要容易得多。然而,现代数据仓库实施和管理应用的出现,使得对多重或者分布式数据仓库的需求减少了。

11.3　数据的数据——元数据

这正是所需要的——更多的数据

元数据的真正困难是,这个名称使人想起一些高技术概念,并且这个概念需要很大的精力来讨论和理解。随着数据仓库作为一个主要的决策支持来源,元数据和它们所描述的企业数据一样,已经成了一个重要的组织资源。

元数据的标准定义是“关于数据的数据”。虽然这个定义说出了元数据的最主要特征,但是它还不能完整地概括出元数据的作用。比如说,我们如何通过关联数据来增加数据的意义?为了回答这个问题,我们必须首先明白元数据建立的主要概念。

抽象的概念

为了理解元数据,我们必须注意另一个相对简单的一般语义概念,即“抽象链”。比如说,沙发是一个普通的家具实体,任何一个沙发都具有很多性质:它是由木头、布、钢和皮革制成的,它有两个扶手和一个靠背,它有自己的形态和样式,它有垫子等等。我们可以从中抽象出单词“沙发”。换句话说,当我们使用“沙发”这个词,我们就可以想像出各种不同的沙发。事实上,我们可以描述所有的沙发。把各种各样的沙发用一个很简单的词“沙发”来表示。这就是抽象链的第一步。我们继续往后进行,我们把“沙发”这样的所有单词又可以用“单词”这个简单的概念表示。抽象链一层一层地进行概括是没有尽头的。

软件只是抽象链中一个简单的链接。现在你读到的“沙发”这个词,如果把它存储在计算机中,就是硬盘表面氧化物的一个磁通量变化或者是光盘表面上的小坑。例如如下一条源代码语句:

```
LET CHECK_BALANCE = CHECK_BALANCE + CREDIT - DEBIT
```

这个源代码语句,就是一个银行事务的抽象,它又是一个银行的抽象,一个行业的抽象,一种经济的抽象……

元数据就是数据的抽象。它们是高层次的数据,为我们提供对低层次数据的简明的

抽象。世界充满了元数据——请看以下的例子：

- 作者、题目、ISBN 号、出版社
- 小说的标题
- 数据仓库中所有数据元素的商业定义
- 地图
- CASE 工具仓库中的数据流程图

数据的键

元数据是把原始数据转化为知识的主要元素。例如，元数据对于某一字段的定义告诉我们这部分字节流是顾客订单号，是位图图形的一部分，或者是我们刚刚建立的字处理文档的名称。一旦元数据出现混乱，则数据仓库中的数据将失去意义。订单号不能与我们的订单相匹配，我们建立的文档找不到了，甚至图形也不能用正确的方式浏览了。虽然这些只是由于元数据混乱所造成的，但是我们往往不易察觉到其中的不同。

可以把元数据看做一种“夹子”，利用它来处理原始数据（APT，1996）。没有了元数据，数据就失去了意义，我们不知道它们在哪里，占有多大的容量。没有了数据的定义，我们如何去查询数据库？我们能找到的只是一系列的 0 和 1。在早期的商业计算机中，每一个应用程序建立并操作自己的数据文件，由于这些文件的元数据是位于应用程序的数据定义之中的，其他的应用程序无法对这些文件进行访问。当数据库出现后，它所带来的巨大的进步就是元数据存储在数据库的目录中，而不是在单个的程序中。

活动中的元数据

假定在同一时刻我们查询组织的数据仓库时，找到以下三种数据集：

1. 615397 8350621 885214 0051023 6481921
2. 一个小组 9/11/96 的报告指出，机械工具在亚洲的市场 1995 年增长了 33%
3. 领先的体育用品营销公司：IMG 45%，SportStars 33%，Legends Inc 16%

我们从这些数据中得到了什么信息呢？

在第一个例子中，答案是“没有”。这些数字可以是部门或地区的销售数据，可能是欧洲城市的人口，可能是一组样本的细胞数目。这些数字甚至可以代表一系列计算机的机器码。这些数据也可以是关系数据库中数据表的数据。有两个方法可以确定这些数据的意义：通过上下文或通过元数据。通过上下文，这些数据是我们已知意义的给定表的查询结果。根据元数据，我们查询描述这些数据的元数据，元数据便可告诉我们表的名称或者更多的信息。

第二个例子看起来易懂一些，这是一句话，自己对自己进行了描述。但是，有一点：日期的显示出现了歧义。9/11/96 究竟是指 1996 年 9 月 11 日（美式），还是指 1996 年 11 月 9 日（英式）呢？在这个例子中，元数据指出所显示的数据格式为“月/日/年”，从而消除了歧义。

第三个例子包含了一些元数据：我们知道数据代表着领先的体育用品营销公司。我们不知道百分比代表着什么，数据是哪个时期的，是如何收集起来的，甚至不知道信息的

来源。没有足够的元数据,本例中的数据就毫无用处。

这些例子表明了元数据在数据仓库中的重要性。这里,数据分析者和执行者正在寻找有用的元素和关系。日常的应用程序对他们来说是没用的:他们需要进入数据中,需要了解这些数据的结构和意义。虽然时刻表比较重要,但是火车上的乘客不需要地图。司机试图穿过其他国家到达一个陌生的村庄,如果没有详细的地图,是很不明智的(APT,1996)。一个没有元数据的数据仓库就好像一个装满文件的橱柜,但是没有文件夹和标签。想像要怎么样才能从中找到有用的东西呀!

一致性——避免重复的记录

不可靠的或丢失的元数据,会导致出现这种情况:一个商业单位说利润增长了 15%,而另一个商业单位却说利润降低了 10%。每一个功能领域都用自己的数据,把应用与过程相连。这种"算法差异"经常导致企业的决策受很多人为的或是不确定因素的影响。

与 ODS 为特定的商业单位或部门带来的利润不同,数据仓库的建立是为了对组织内部的所有领域进行分析,从而获得好处的。如果需要元数据(并且有了元数据)那么数据仓库的作用就是提供大量的用户所获得的信息的数据支持。在这种情况下,数据仓库可以看做一个图书馆馆长,如果你需要它帮助你找到你想要的东西,它就会指导你,如果你没什么目标,你也可以在你的权限范围内自由地浏览。

在数据仓库环境中,不论是数据还是元数据都是不可更新的,决策者们不断地寻找新的方式,对企业内外各个角落的信息进行发掘和比较。他也可以请求载入新的数据集,这些数据必须是操作层应用产生的或是外部数据源中导入的。数据仓库中数据的每一次增加,元数据都会进行扩展。

确保数据不重复

数据仓库的管理员不仅必须管理数据的更新和录入,还需确保数据没有重复。如果一个公司的各个部门的利润相加得到的结果远远大于公司的总利润,这是一件很可笑的事情。为了描述数据仓库中大量的数据元素,元数据必须组织为精确的、前后对照的方式。数据仓库的开发者和维护者面临的最大的任务就是保证数据的完整性和精确性。

11.4　访问数据——元数据的抽取

一般元数据问题

Inmon (1992b) 指出,不论数据仓库中查询的性质有多么奇特,在元数据中存储着的一般信息对决策者来说都是十分重要的。表 11-3 列出了使用数据仓库中的数据时,一些非常重要的基本信息。

虽然数据仓库中的元数据是对历史数据进行分析时使用的,但是也不能说它与操作型数据库的元数据完全不同。这也就是说数据仓库的元数据必须具有一定的说明,让用户知道仓库中数据收集的一般情况的信息。当收集的方式和环境改变的时候,元数据说

明必须一起改变。在这种情况下,我们可以认为元数据说明是元数据的元数据。

表 11-3 与数据仓库使用相关的一般元数据问题

- 数据仓库中存了什么表、属性和键?
- 每一个数据集合的来源是什么?
- 在装载入库时使用的是什么转换逻辑?
- 元数据如何随时间变化?
- 数据的别名是什么以及数据之间的关系如何?
- 技术和业务过程的关联是什么?
- 数据重载的频率是多少?
- 数据仓库中共有多少数据元素?

元数据的组成

在适当的时间收集合适的元数据是一个成功的数据仓库所必需的。虽然某项特殊的元数据对一个数据仓库有用而对其他的数据仓库没用,但是在典型的数据仓库中还是存在几种一般的元数据的。

转换映射

转换映射 (transformation mapping) 元数据,记录了数据是如何从操作型数据库或外部资源中输入数据仓库的。当这些数据被转换为数据仓库中的结构时,这些转换将被记录或保留在元数据中。随着新的数据仓库工具的研发,这项功能已经融入数据转换过程中了。表 11-4 列出了必须存在的几种类型的转换映射元数据:

表 11-4 典型的映射元数据

- 辨别原始来源
- 简单的属性到属性的映射
- 属性转换
- 物理特性转换
- 索引表转换
- 命名转换
- 键值转换
- 默认值
- 从多重来源选择的逻辑
- 算法的变化

提取记录和关联记录

每当分析过去的信息时,必须保留繁琐的更新记录。通常,决策者必须通过查看提取记录 (extraction history),来构建一个基于时间的报表,因为为了在适当的数据上应用正确的规则,所有业务规则的变化都需要进行确认。如果在 1994 年重新划分销售区域,那么决策者应该知道,1994 年前后的销售报告是不存在可比性的。

除了有关规则变化的元数据外,数据仓库中还需要保持有关各种关联 (relationship) 的信息。数据仓库中数据关联的表示不同于应用数据库。必须保留关系表、约束、基数、上下文描述以及归属记录。这些信息和变化的记录在决策者针对多重集合的数据进行分

析时是十分有价值的。

综合算法

一个典型的数据仓库不仅包括底层的数据，还要包括各种不同层次的综合性数据。这些应用于细节数据的综合算法（summarization algorithm），对于决策者分析或者解释综合数据的意义是非常重要的。对于一个给定的分析内容，这一元数据可以告诉决策者他们需要哪一层次上的综合数据，从而节省时间。

数据的归属

操作型数据库往往是属于组织中某一特定的单位或部门的。而在数据仓库环境中，所有的数据按照统一的格式进行存储，并且为所有人共享。因此，有必要区别出每个数据集合的创造者，由他们来进行数据的查询和校正。区别操作型系统中数据的“主人”和数据仓库中数据的“管理员”是十分有用的。数据仓库中的管理员负责数据的收集、综合以及传播，因此他们只能成为数据的管理员，而 ODS 数据源的管理者对事务层的数据的精确性负责，他们才是数据的真正“主人”。

数据仓库的访问形式

记录数据仓库访问形式的数据通常也是很有用的。了解什么表被访问、访问的频率和访问者，可以提醒数据仓库的管理员完善和简化最终用户进行的查询操作。访问量少的数据可以被存储到相对便宜的介质上去。对访问量大的数据，提高访问速度的方法有很多。而且记录历史查询也是一项很有用的资源，为了重新进行同样的查询，决策者可以方便地访问过去查询的结果，并选择与需求最接近的查询。

其他的通用元数据

除了上述几种主要类别的元数据外，还有一些不太通用的，却同样重要的元数据。这些元数据经常是与上下文相关的或者取决于组织的。但是，一旦需要进行分析，这些元数据和上面的普通元数据是同样重要的。

这些元数据中有一个重要的成员是“逻辑商业数据模型（logical business data model）”，逻辑模型定义了数据所在特定的商业环境中的实体、关系和规则。建立数据仓库而不包括数据模型资源是十分危险的，而且容易出错。如果这些数据模型是可用的，那么，元数据就描述出了物理设计到数据模型之间的映射，这样就允许重新解决模糊问题和不确定性问题。

另一种类型的元数据可以使得数据仓库对用户更为友好，即记录数据别名（alias）的元数据。在数据仓库中，各地区的单位都能查询数据表，别名的使用使得查询的构造和结果解释更加易懂，别名还可以使不同的业务部门能够自由地用自己定义的名称来指代相同的数据。慎重地跟踪数据的别名是数据仓库中一个重要的部分。

开发状态（development status）元数据是另一种对于数据仓库使用者十分重要的资

源。通常,数据仓库中不同的部分会处于不同的开发和完善阶段。这些状态信息用来向决策者提供信息。例如,根据一个特定的查询建立一个表,这个表可以分为设计阶段的、测试阶段的和非活动阶段的。

最后,数据容量元数据(volumetric metadata)告诉用户他们操作的数据有多少。这样他们就可以得出他们查询需要的时间和资源代价。容量元数据通常包括的信息有:总记录数、表格增长速度、使用特性、目录和字节说明等。

11.5 数据仓库的实施

访问组织内的数据集的方法并不是一种新兴的事物,在 20 世纪 70 年代就出现了信息中心(information center),这是一个"注定要遭厄运的"概念。它不仅要求有高性能的计算机,而且对软件、硬件和人力等资源也有很高要求。80 年代又推出了使用扩展关系模型对数据进行重新驱动。同时,伴随而来的是更加复杂的程序和不方便性。到了 90 年代,数据仓库的出现解决了这些问题。我们把企业中所有的数据全部联系在一起,从而避免了过去的那些难题。IT 管理者们以及组织内数据仓库的维护者们必须不仅知道需要做什么,还要知道如何来做。

Denis Kozar (1997) 是大通银行企业信息系统的副总裁。他总结出了数据仓库实施中的 7 个致命的错误。这些错误中的每一个都会导致数据仓库的失败。表 11-5 列出了数据仓库这 7 种致命的错误。

表 11-5 数据仓库实施中的 7 种致命错误

1. "建立之后,它们自己会来"
2. 数据仓库体系框架的遗漏
3. 低估了记录所有假设和潜在冲突的重要性
4. 滥用方法和工具
5. 错误的数据仓库生命周期
6. 没有考虑解决数据冲突
7. 没有记录第一个数据仓库项目中出现的错误

错误一:"建立之后,它们自己会来"

Kozar 指出 7 个错误中的第一个就是没有明白数据仓库建立的意义。在建立数据仓库时,认为改善既定的商业目标比数据结构更重要是错误的。一个成功的数据仓库的设计需要考虑到整个企业的需求,用这些需求来指导设计、建模,并贯穿项目的始终。数据仓库并不是简单地建立在某个人特定的需要上。

错误二:遗漏体系框架

成功的数据仓库最重要的一个因素是开发和保留有意义的体系框架。这种框架是作为建构和使用不同的数据仓库部分的蓝图。比如期望的最终用户数量、数据的容量和差

异、期望的数据更新周期等,这些都必须在数据仓库设计时考虑到。

错误三:低估了记录假设和冲突的重要性

在数据仓库建模的过程中,必然包括一些假定和潜在的数据冲突。这些都必须尽早地记录在文档中,以保证它们反映在最终产品中。在建立数据仓库的需求分析阶段,明确这些问题是十分重要的:数据仓库中应该存储的数据量为多大?数据的粒度是怎样的?需要多长时间更新数据?数据仓库在哪一个层次上开发和运行?明确地回答这些问题,对于建立一个成功的数据仓库是十分重要的。

错误四:使用不适当的工具

数据结构的建立和设计的方法有很多,它与操作型应用系统的建立有很大的不同。因此,数据仓库项目中所用到的工具也与传统的应用程序的工具有很大的不同。

数据仓库的工具一般分为四种:(1) 分析工具 (analysis tool);(2) 开发工具 (development tool);(3) 实施工具 (implementation tool);(4) 传输工具 (delivery tool)。每一类工具都适用于某些特定的与数据仓库开发相关的一些设计活动。

分析工具　这类工具用于帮助识别数据的需求、数据仓库的主要数据源,以及数据仓库中数据模型的构造。现代的 CASE 工具属于典型的分析工具。另一个分析工具是代码扫描仪,它们可以扫描文件和数据库定义的源代码以及数据使用标识。这些信息确定了源 ODS 中的数据需求,从而帮助构建数据仓库的初始数据模型。

开发工具　这种工具用于数据过滤、代码生成、数据整合以及在数据仓库中装载数据。这类工具还是数据仓库中元数据的主要生成器。

实施工具　这一类别主要包括一些获取数据的工具,用来收集、处理、清理、复制和合并数据仓库中的数据。此外,这一类中的信息存储工具可以用来装载外部数据源的综合性数据。

传输工具　这类工具用于数据的转化、追寻数据来源以及报告最终的传输平台。这类工具包括查询、报表和建立数据索引的专门工具,使得最终用户能够知道有什么数据真正地存储在数据仓库中。

错误五:错误的生命周期

这个错误是指数据仓库的开发者没有认识到数据仓库的生命周期 (DWLC) 和传统的系统开发的生命周期 (SDLC) 的方法论的不同。虽然两者比较相似,但是,在一个重要的方面两者是不同的:DWLC 是永远没有止境的。一个数据仓库的生命周期是连续的,一直活动着的,需要反复地通过数据管理来研究数据仓库的需求。通常,数据仓库的一个阶段的结束就是另一个阶段的开始,它不断地需要新的数据,增加用户组和新的数据源。如果数据仓库想保持支持决策的可行性,那么数据仓库的开发者一定要认识到这一点。

错误六:没有考虑解决数据冲突

之所以建立一个新的数据仓库,通常就是因为组织内决策需要高质量的数据,这正是

建立数据仓库的目的。但是,在实际运行的数据仓库中这却是很难实现的。在建立数据仓库及其应用的时候,人们和组织总是很自然地想到保护自己的数据。结果,必须进行大量的冗长分析来决定组织内最佳的数据源。一旦这些系统进行了整合,数据的冲突就会表现出来,名称的不一致、文件的格式和大小、取值范围的差异必须解决。这个过程中可能要与数据的主人协调工作,以对源数据将来规划内的和规划外的变化达成共识。没有足够的时间和资源来消除数据冲突,将阻碍数据仓库的建立,从而导致组织的停滞,直接影响到项目的完成情况。

错误七:没有从错误中吸取教训

DWLC 的一个特点是:先前建立的数据仓库对后面的数据仓库是有影响的。因此,认真总结以前数据仓库的失误将直接提高后面所有数据仓库的质量。通过不断地从过去的经历中总结失败的教训,才能够建立一个健全的、完善的数据仓库。

11.6 数据仓库技术

在过去还没有建立数据仓库的时候,每个人都梦想有那么一天向 Wile E. Coyote 的最好的供应商 Acme 订货,在几秒钟后,就会收到一个大木箱,上面印着"Acme 数据仓库",里面装着所有我们需要的东西,甚至包括电池。

这一天可能已经到来了。但是目前的数据仓库产品还是"Acme"层次上的。现在,企业的选择很少,但是能够从很多厂家购买到各种各样复杂的硬件和软件。零售商、分析家和专家们并不直接把注意力集中到数据仓库技术上,他们就像挥舞着红布的斗牛士,对顾客说:"一个数据仓库是一项建筑,而不是一个产品",或者"它是一个过程而不是一个静止的东西",甚至说,"它有 90% 是专门的知识,只有 10% 是技术"。这些修辞的意思很简单:祝你能有好运气把这些东西整合在一起。

为了建立有用的数据仓库,在结构、过程、专家知识以及其他方面,开发者必须拥有非常丰富的资源。但是光有这些资源,还不能保证建立一个成功的数据仓库环境。数据仓库的投资必须有一个严密的评估过程,来评价领先数据仓库提供商提供的数据仓库工具的优缺点。

目前没有一个数据仓库厂家能够提供端对端(end-to-end)的数据仓库解决方案。但是 SAS,IBM,Software AG,Information Builders 和 Platinum 已经开始朝这方向努力了,但是热度还远远不够。比如,在下面的例子中,主要数据仓库供应商 IBM 就面临了一种这样的挑战。

IBM 起初的产品是 Visual Warehouse(可视化数据仓库),如果在 OS/2 系统下运行,可以很好地整合,但是在其他的操作系统平台上,例如 windows NT 和 Novell 等,它的灵活性就很差。而且,Visual Warehouse 还不能管理局域网之外的数据库。更大的、更健全的数据库建立在 DB/2 并行处理的基础上,如 DB/2 MVS 和新的 AS/400 并行数据库,也是 IBM 开发的数据仓库产品。尽管 IBM 宣称他们会不断地解决这些难题,但是他们不愿意提供任何时间进度安排。换句话说,即使是数据仓库供应商,也在协调各部分工作中遇到

了很多麻烦。

有一个好消息是，目前市场上存在很多数据仓库厂商，为企业提供了更多的选择，它们可以选择适合自己的数据仓库开发所需的工具。还有一个坏消息是，环境的多变性以及昂贵的成本使得小公司很难在这样激烈竞争的市场上立足，并且，每天各个厂商的地位都在发生变化。表 11-6 列出了目前数据仓库的提供商及其产品（截止到本文所写时间）。

表 11-6　数据仓库厂商及其产品

厂　商	产　品
Actuate Software	Report Server，Reporting System，Web Agent
Andyne Computing	GWL，PaBLO，Txet Tool
Angoss Software	KnowledgeSEEKER
Aonix	Nomad
Applix	TM1 Server，TM1 Client，TM1 Perspective，TM1 Show Business
AppSource	Wired for OLAP
Arbor Software	Essbase，Essbase Web Gateway
Attar Software	SpertRule Profiler
Belmont Research	CrossGraphs
Brio Technology	BusinessObjects，BusinessMiner
Carleton	Passport
Cognos	Impromptu，PowerPlay，Scenario
CorVu	Intergrated Business Interlligence Suite（IBIS）
Computer Associates	CA-LDM，CA-OpenIngres，Visual Express
Concentric Data Systems	Arpeggio
Data Distilleries B. V.	Data Surveyor
Data Junction	Data Junction，Cambio
Data Management Technologies	Time Machine，RQA-Remote Query Accelerator
DataMind	DataMind
Digital Equipment	Alpha Warehouse
Dimensional Insight	CrossTarget
Enterprise Solutions	InfoCat
European Management Systems	Eureka
Evolutionary Technologies International	ETI-Extract
Harbor Software	HarborLight
Hewlett-packard	Intelligent Warehouse
Holistic Systems	OLAP，Spider-Man
Hyperion	Data propagator，DB2 Database Server，Enterprise Copy
IBM	Data Propagator，DB2 Database Server，Enterprise Copy Manager，Data Hub for OS/2，Data Hub for Unix，FlowMark，GataGuide，Applicaton System，Visualizer family，Intelligent Decision Server，Query Management Facility，Intelligent Miner
Informatica	PowerMart
Information Advantage	DecisionSuite，WebOLAP

Information Builders	EDA/Copy Manager，Focus Fusion，Focus Six EIS Edition

续表

厂 商	产 品
Information Discovery	Data Mining Suite
Informix Software	Online Dynamic Server-Unix，Online Dynamic Server-WindowsNT，New Era ViewPoint Online Extended Parallel Serever，MetaCube，MetaCube Warehouse Manager
InfoSAGE	DECISIVE
Innovative Group	Innovative-Warehouse
Integral Solutions Limited	Glementine
Intersolv	DataDirect Explorer，SmartData
Intrepid Systems	DecisionMaster
IQ Software	Acumate ES
Liant Software	Fiscal Executive Dashboard
Logic Works	Universal Directory
Mayflower Software	Sentinel，Information Catalog
Mercantile Software System	IRE Marketing Warehouse
MathSoft	Axum
Microsoft	Microsoft SQL Server
NCR	Teradata
NewGeneration Software	NGS Managed Query Environment
Oberon Software	Oracle8，Discoverer/2000，Oracle Express Server
Pilot Software	Decision Support Suite，Command Center
Pine Cone Systems	Data Content Tracker
Platinum Technology	InfoRefiner，Info Transport，Fast Load，Data Shopper，InfoReports，Object Administrator，Query Analyzer，Report Facility（PRF），InfoBeacon，Forest & Trees
Postalsoft	Postalsoft Library products
Powersoft	InfoMaker
Praxis International	OmniLoader
Precise Software Solutions	Inspect/SQL
Prism Solutions	Prism Warehouse Manager，Prism Change Manager，Prism Directory Manager
Progress Software	EnQuiry
QDB Solutions	QDB/Analyze
Red Brick Systems	Red Brick Warehouse5. 0，Red Brick Data Mine Option，Red Brick Data Mine Builder，Red Brick Enterprise Control& Coordination
Reduct Systems	DataLogic/R
ReGenisys	Extract :R
Sagent	Sagent Data mart
SAS Institute	SAS Data Warehouse，Warehouse Administrator，SAS System，SAS/MDDB
Seagate Software IMG	Crysta Reports，Crystal Info
SelectStar	StarTrieve

续表

厂　商	产　品
ShowCase	STRATEGY
Siemens-Pyramid	Smart Warehouse
Silicon Graphics	MineSet
Smart Corporation	Smart DB Workbench
Software AG	Intelligon，Passport，SourcePoint，Esperant，ADABAS
Softworks	meta VISION
Spalding Software	DataImport
Speedware	Esperant，EasyReporter，Media
SPSS	SPSS
Sterling Software	Vision：Journey
Sybase	Sybase SQL Server 11，Sybase IQ，Sybase MPP
Syware	DataSync
Tandem	Tandem NonStop
Thinking Machines	Darwin
Torrent Systems	Orchestrate
Trillium Software	Trillium Software System
Visual Numerics	PV-Wave
Vmark Software	UniVerse，DataStage
Wincite Systems	WINCITE
WizSoft	WizWhy

11.7　数据仓库的未来

数据仓库成为一个组织决策支持基础设施的成熟部分是一个必然的趋势。它将融合在组织的结构中，通过数据通信网络把数据从一个地方传到另一个地方。换句话说，一旦目前访问企业数据的相关问题得到了解决，将会有越来越多的公司找到新的途径来使用这些数据。这些新的焦点将会带来新的挑战。这一节将介绍数据仓库未来要面临的几种挑战。

规章的约束

从大量的不相关的数据中分析并提取信息的能力，使得必须产生一些保护某些数据不被其他对象访问的要求。随着数据的访问更加容易，这种防止隐私泄漏的要求也不断提高，这就需要建立一些规则，在进行大量有用分析的同时，保护个人的隐私。

存储非结构化的数据

通常，一般的数据仓库只局限于存储结构化的数据，形式一般为记录、域以及数据库。非结构化的数据，例如多媒体文件、图形、图像、声音、视频文件，在组织中已越来越重要了。对这些文件的存储、整合和访问要求有扩展的数据仓库结构和接口。在未来的数据

仓库环境中，用户可能会寻找不同产品之间的联系，数据仓库不仅要存储结构化的数据，而且还需要能够扫描和分析图像、音频和视频文件来促进这种关系的建立。要实现这个层次上的使用和功能，数据仓库应用和工具厂商面临着大量的技术上的和实现上的挑战。如果未来的数据仓库包括了这些数据和技术，如何管理非结构化的数据的存储和恢复、如何找到特定的数据记录等这些事务都将被解决。

万维网（WWW）

随着目前越来越多的信息的相互关联，WWW 无疑对数据仓库的建立有着重要的影响。网络使得访问和转换大量的相关数据更为容易和经济。这使得 Internet 和 Web 成了把外部数据库和数据仓库整合起来的理想工具。这样一来，数据的一致性、精确性和数据质量问题就需要注意和解决。这样可能会出现第三类企业，它们的主要目标就是评估外部数据源的一致性和质量。这种质量评估可以决定外部数据源载入数据仓库时的价值。同样，这种质量评估可以决定访问这些数据需要的价格，数据质量越高，价格也越高。

11.8　小　　结

随着势不可挡的自动化过程的实现，各个商业和工业企业组织，不论大小，每天都产生出大量的数据。最大的困难就是把存在于各个独立信息系统中的数据加以集成。随着数据仓库的出现，产生了从未有过的数据驱动以及对数据之间联系的挖掘。这个概念是服务于重新整合各种来自内部和外部的信息系统的。在下一章中，我们将在理解数据仓库的基础上，主要探讨对数据中所隐藏的资源信息进行挖掘。

复习题

1. 定义下列术语：
   数据存储
   数据集市
   元数据
   面向主题
   数据整合
   时间变量
   数据不变性
   抽象链
   转化映射
2. 数据仓库为工厂各种层次的管理者提供了哪些独特的好处？
3. 什么是数据仓库？它比传统的信息收集技术好在哪里？

4. 描述数据仓库环境。
5. 数据仓库的特性有哪些？
6. 列出并解释数据仓库体系的不同层次。
7. 什么是元数据？为什么元数据在数据仓库中如此重要？
8. 在建立数据仓库时，7 种致命错误是什么？

讨论题

1. WWW 包括大量的关于数据仓库的信息，甚至包括数据仓库本身。从数据仓库的展望中看 WWW，描述数据仓库的各个部分，考虑一个组织如何利用 WWW 作为一个有用的数据仓库。
2. 数据仓库的支持者说，数据仓库的概念能够运用到任何行业或知识领域。考虑几个利用数据仓库完善信息管理的工具，你能够想出一些不能运用数据仓库的行业吗？
3. 在你身边找一个建立数据仓库的组织，如果可以的话，与数据仓库的管理者交流一下，在设计和实施数据仓库时面临的困难，他是如何克服的？这些困难现在还存在吗？
4. 元数据无处不在，找到一个与学校、工作或者家里的信息相关的数据库，尽可能地找出所有的元数据。

第 12 章

数据挖掘和数据可视化

学习目标

- 理解数据挖掘（DM）的概念
- 认识决策支持活动从确认到发现的转变
- 理解联机分析处理（OLAP）的概念和规则
- 学习两种多维数据分析的方法——多维 OLAP（MOLAP）和关系 OLAP（ROLAP），并探索两种方法的不同适用条件
- 熟悉四种主要的用于挖掘数据的处理算法和规则：分类、关联、序列和聚类
- 了解当前的数据挖掘技术：统计分析、神经网络、遗传算法、模糊逻辑、决策树
- 通过实例学习知识发现的一般过程
- 了解当前数据挖掘面临的限制和挑战
- 了解数据可视化的历史以及它是如何协助决策活动的
- 认识数据可视化技术的典型应用
- 回顾几种当前的“siftware”技术

小案例

NBA 通过数据挖掘取得飞跃

每个关注职业篮球赛的人都知道，达拉斯小牛队还是不成熟的，而且像很多 NBA 球队一样，小牛队也在培养新球员、制作有吸引力的球赛节目上费尽心思。他们不可能短期内就夺得 NBA 冠军。副教练 Bob Salmi 很清楚这一点。

但是他也清楚迈克尔·乔丹从芝加哥公牛队退役的时候，至少有 10 个 NBA 球队会实力相当——他希望也包括小牛队。在这样一个水平相当的比赛场上，一个教练怎样培养球队的竞争优势呢？

一个办法是利用信息——具体地说，就是由 Advanced Scout 提供的信息。Advanced Scout 是 IBM 为 NBA 提供的一套用于赛后分析的数据挖掘应用系统，是在职业体育赛事中典型的技术应用。

Salmi 使用 Advanced Scout 来挖掘训练数据中的模式，比如成功的球员组合。Advanced Scout 与记录 NBA 球赛的数字录像保持同步，这些录像都根据统一的时钟标注了时间，并把所有统计数据（比如，抢篮板和得分）以表格形式记录下来。一旦出现看似成功的模式，教练就切换到数字录像机查看相关的附加因素：他们用了什么打法，是如何进攻和防守的，还有哪些球员参与等等。

这种采用数据挖掘的赛后分析比起人工的老方法（用纸笔匆匆记录统计数据，还要不停的倒播录像）节省了不少时间。那个过程对猜测和教练的直觉的依赖性比较强，常常要花整夜的时间。相反，Advanced Scout 允许在几个小时之内给出分析，以便指导下一场的比赛。

Advanced Scout 采用了基于C++的叫做属性聚焦（attribute focusing）的数据挖掘技术，寻找有意义模式和统计数据间的关联。属性聚焦的关键特征在于它是为外行人（或者领域专家）而不是专业的数据分析师而设计的。这种技术强调非技术性用户，这是它与数据挖掘领域中的应用（如目录管理、购物篮分析、序列分析）所采用的神经网络、决策树、回归分析等方法的区别之所在。这些应用的软件工具大多十分复杂，更适合分析师专用。

现在，NBA 根据每个季度的数据进行挖掘。如果教练有了两个季度的经验，他们就想把挖掘范围扩大到历史统计数据上，去回顾近几年的情况。教练不仅可以在赛后分析中用到这些结果，在与球员签约谈判时也可以用上。

这在 1997 年 NBA 奥兰多魔术队和迈阿密热队的季后赛中得到了清楚的显示。和 Bob Salmi 一样，奥兰多魔术队的助理教练 Tom Sterner 从一开始就使用 Advanced Scout。热队在五场激战后最终获胜。但是，“如果不是利用 Advanced Scout 进行数据挖掘，热队可能在第三局就把我们淘汰了”。

“在迈阿密的前两场比赛中，我们打得十分狼狈——第一场比分是 99：64，第二场是 104：87，”Sterner 回忆说，“第二场下来，我们进行了分析，认真观看了录像，研究发生了什么问题。Advanced Scout 给我们提供了一些有趣的数据。”

第一，Advanced Scout 指出第二场中，奥兰多的 Anfernee “Penny” Hardaway 和 Brian Shaw 的配合是 - 17 分（篮球教练通常采用加分减分的系统来评价球员或球员组合的表现，用球员在场时的本队得分减去对方的得分）。它还指出，Shaw 和 point guard Darrell Armstrong（一位比赛时间并不长的候补球员）的配合是 +6 分。而且，Armstrong 和 Hardaway 的配合得了 +14 分。

根据 Advanced Scout 的分析，Sterner 决定在第三场比赛中让 Armstrong 上场多一点。结果很让人吃惊。“第三场比赛距中场还有六七分钟的时候，我们又落后了 20 分。我们让 Amstrong 上场，这样，中场时我们将比分拉平至 42：42。比赛最后，我们以 88：75 的比分取胜。Penny Hardaway 得了 42 分，Armstrong 得了 21 分。最重要得是，我们没有在这局比赛中被淘汰。”

Sterner 说，“教练需要领会信息再做球队的调整——他的洞察力是一种艺术。为什么是芝加哥的 Phil Jackson 获胜？迈克尔·乔丹当然起了一定作用，但是每支队伍的人力因素对全队的胜利是相当重要的。”

“如果你比对手具有更高的天赋，你就能取胜。当双方天资相当时，胜败的关键就看为比赛做准备时采用的技术了。”

“球迷对我们的球队充满了信心，我们出乎任何人的预料，又多打了两场主场赛。”Sterney 说，“（门票、电视广告……）这些有上千万的价值。这简直是正面效应在滚雪球。”

æskibm@ vnet.ibm.com

12.1　一幅图值一千句话

这一部分的题目是句老俗话了，是说我们观察事物比阅读或者谈论它得到的信息更多。乍听起来，这可能是正确的。但是，如果它按照字面意思确实正确，那么我就可以用 350 张左右的图片把这本书的意思传达给你了（如果是恰当的图片的话）。

这句俗语不是通用的，但在给定的一些情况下的确可行，这样说才是合理的。有些时候只用图片传情达意很困难，用文字语言更方便；也有些情况下，我们用文字语言来查询，以便发现那些用图形显示效果最好的新东西。有些时候，图像确实是传达某些信息的惟一的有效方式。

在这一章里，我们将探讨数据挖掘和数据可视化的问题。在认识了数据仓库之后，我们具备了新的能力，能够提出原先无法提出的问题，能够在数据中发现原先不能发现的联系。

什么是数据挖掘

简单地定义，数据挖掘（data mining，DM）是发现数据中新的、隐藏的或不可预知的模

式的活动。利用数据仓库中包含的信息,数据挖掘可以回答决策者原先根本没有想过的问题。

数据挖掘技术的一个常见的同义词是知识数据发现(knowledge data discovery,KDD)。这个概念形象地描述了有关从聚集数据中发现有用的知识的所有活动和过程。采用一系列技术,包括统计分析、神经和模糊逻辑、多维分析、数据可视化和智能主体,KDD 可以发现十分有用的、可增长知识的模式,可以在广泛的知识领域中建立行为或结果的预测模型。

比如,使用 KDD 技术,我们知道习惯用左手的妇女一般购买右手使用的高尔夫手套。这个模式把顾客的属性与产品的销售联系起来。还有,每次道琼斯指数跌落三天之后,AT&T 的股票价格就会上浮至少 2%。每个人都可以想出这条信息的几个作用(我知道我能)。

另一个应用中的 KDD 的例子是顾客消费习惯的研究:买尿布的男士也买啤酒;买水下呼吸面罩的顾客要去澳洲度假;买了脱脂牛奶的顾客也会买麦片面包。最后,本章开头的小案例清楚地解释了数据挖掘如何在 NBA 比赛中发挥作用的。以上只是对 KDD 技术的应用方式的一个小抽样。众多的潜在的方式需要使用广泛的 KDD 方法和技术来发掘。

确认与发现

早期的决策支持活动主要基于"确认(verification)"的概念。这样,关系数据库对组织好的问题给出动态的答案。确认的关键在于,要求决策者事先具备丰富的领域知识,发现了一个值得思考的关系后通过查询来进行确认。这种"确认式"的查询在赌博或者娱乐场行业中的应用比较普遍。赌场老板利用记录顾客特征的数据库对顾客进行分类。他可以输入新顾客的已知的特征,来鉴定顾客的类别。赌场老板需要管理数据库中蕴含的大量的已知关系,但是能够对一个新客户分类的技术还是很有用的。

当赌博业采用更复杂的计算机技术时,"确认"发展成了另外一个新概念:"发现(discovery)"。通过采用 siftware(这是一种专门设计的软件,用于在数据中发现新的、原先没有分类的模式),赌场老板可以识别顾客消费习惯的新模式和类别,可以更有效地制定目标,为特定的顾客群提供服务。

意识到 KDD 的应用对决策支持的巨大潜能,并不是困难的事情。不管采用什么技术,如果我们要有效地运用 KDD 技术,首先就要理解它的基础。因此,我们探索数据挖掘要从联机分析处理(OLAP)开始。

12.2 联机分析处理

1993 年,公认的关系数据库之父 E. F. Codd 提出了联机分析处理(on-line analytical processing,OLAP)的概念。Codd 指出,鉴于数据视图的维度问题,用于事务处理的传统的关系数据库的能力已经极为有限。他还指出,操作数据和操作数据库不足以回答管理者提出的各种问题(在 Codd 博士提出此观点之前很多年,DSS 和 EIS 研究组织就持有这

样的观点)。多维数据库使组织中的决策者能够从多种维度自由地使用企业数据模型。Codd 提出了多维数据库的发展和使用的 12 条规则,表 12-1 总结了 OLAP 的这些规则。

表 12-1　Codd 关于 OLAP 的 12 条规则

1. 多维视图	7. 动态的稀疏矩阵处理
2. 对用户是透明的	8. 支持多用户
3. 可访问的	9. 跨维操作
4. 一致的报告	10. 直观的数据操作
5. 客户机/服务器结构 (C/S 结构)	11. 灵活的报告
6. 一般维度	12. 无限的维度和聚合水平

到目前为止,并没有实现严格遵守所有规则的技术。事实上,学术界一直有争议,说 12 条规则不能同时实现 (Gray,1997)。再后来,出现了 OLAP 这个术语,代表决策者对汇总的企业数据进行多维分析的各种软件技术。同时,也出现了两种分析的特殊方式:(1) 多维 OLAP (multidimensional on-line analytical processing,MOLAP) 和 (2) 关系 OLAP (relational on-line analytical processing,OLAP)。

多维 OLAP —— MOLAP

从多维角度分析数据时,观察比描述更容易一些。图 12-1 表示了如何从时间、产品和市场地区等维来分析销售数据。

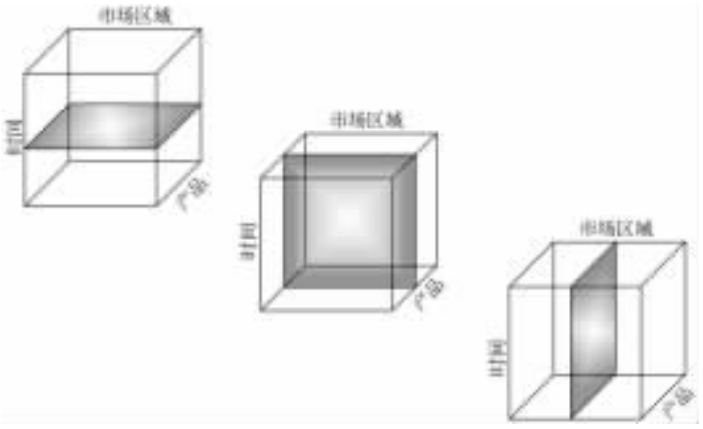


图 12-1　数据的层次结构

如图所示,数据看起来好像存放在一个三维矩阵 (或立方体) 中,立方体每一边代表一个单独的维。三维构成的小单元中存储着用户感兴趣的数据,每一维可以包括一个或多个类别。比如,市场地区维包括东、西、南、北、中,产品维可以包括每一种经销的产品,时间维可以由月、季、年或者指定的财务时期代表特定的销售周期。

三维分析在有些情况下十分有用,但大多数多维分析已经远远超出了三个维度。实

际上,MOLAP 用 n 维方体[①]来组织和分析数据。从概念上,可以把 MOLAP 想像成一个普通的电子表格,它有两个延伸:(1) 支持多个维度;(2) 支持多个合作的用户。

在一个超立方体中,数据存储在多维矩阵中,矩阵的每一个小单元表示所有维的一个交叉点。采用这种方法,可以同时分析任何数目的维度,可以创建数据的任何维度的视图。但是,这种设计使超立方体的许多小单元可能没有值。举个例子说,一个超立方体通过产品、地区、时间、销售数量、实际完成计划销售的比率来分析销售系统,不难想到,在任何一个时间段内,不是所有的产品都在所有的商店或地区出售。一般地,超立方体的维数增加时,可能的空单元也随之增加。这种现象称为稀疏性。由于要为所有的单元格而不仅仅是带值的单元格分配存储空间,大大提高了对 MOLAP 超立方体的存储要求。但是,采用了特殊物理存储技术 (如数据压缩、复杂索引机制) 的新产品悄然兴起,减少了稀疏性的不利影响,也保证了对数据立方体单元的快速访问。

伸缩性是当前 MOLAP 面临的限制之一。虽然它便于处理总结性的数据,却不太适用于大量的细节数据处理。当前的 MOLAP 只适用于 50G 以下的数据仓库。为了解决数据容量受限以及伸缩性的问题,数据库领域又一次采用关系模型,来替换 MOLAP。

关系 OLAP——ROLAP

在 ROLAP 中,多维数据库服务器被大型的关系数据库服务器代替。这种超级关系数据库包括细节的、总结性的数据,允许对数据集做"下钻 (drill down)"操作。如果数据库中没有所需要的数据总结,ROLAP 的客户端工具将动态地生成总结。使用 ROLAP 要在伸缩性和性能之间权衡。一方面,ROLAP 有强大的管理工具,开设了 SQL 接口,可以开发出方便的、有一定伸缩性的工具。另一方面,ROLAP 需要大量的关系表来存储大规模的数据和维度关系。在进行表连接和索引处理的时候需要强大的处理能力,这方面 MOLAP 的性能就不尽如人意了[②]。

一种用来减小 ROLAP 相关的费用的处理办法是将表组合成星型模式。图 12-2 解释了这种数据组织方式。

星型模式的中间是中央事实表 (fact table),这个表包括了所有维交叉点上的数字度量。如果维度的任何组合都没有值,那么就不存储零值。中央事实表中的事实代表了在不丢失含义的情况下进行聚合的、而且需要从两维以上来描述的数据 (Inmon, Welch 和 Glassey, 1997)。中央事实表的目的是理顺访问事实表中的事实的流程。

中央事实表四周是维度表 (dimension table),这些表对事实表中的数据进行描述或分类。中央事实表的键是所有维度表的键构成的复合键。中央事实表中,每个维度表的键的属性域的组合都对应着一行。采用这种方式,可以对大型的数据集进行多维分析,不像

---

① 立方体用来描述数据占有的 n 维空间。一旦超过三维,就不再是几何意义上的立方体了。一个大于三维的数据立方体通常称为超立方体。

② 表连接是一项基本的数据库操作。通过各表的一列或多列连接两个或更多的表的行 (或记录)。索引处理包括查找表的创建、维护和使用。查找表把索引文件中某属性的值与文件中的源记录联系起来。这就允许在主键以外的属性上建立查找。这些操作都需要强大的处理能力,尤其是面向大型数据库的应用。

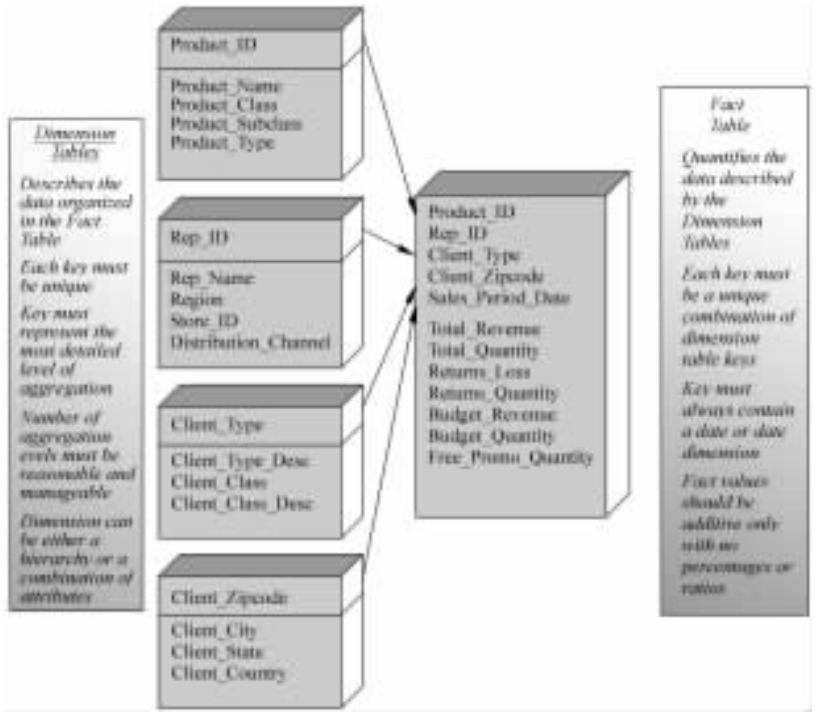


图 12-2　典型的星型数据组织

MOLAP 的实施中要受到 50G 数据量的局限。ROLAP 结构最适合那些既对总结性数据又对细节数据进行动态访问的系统，而 MOLAP 在使用总结性数据或预先处理的数据时有比较好的性能。

12.3　用于挖掘数据的技术

随着数据挖掘飞速流行，挖掘数据仓库的新技术正在快速成型。很多新技术是前边提到的那些方法的提炼，也有一些主要是革新的方法。但是，由于缺少一种开发的标准，数据挖掘的革新也仅限于特殊的开发平台，不能以全面、普遍的应用来推动此项技术的发展。我们暂且不管具体的技术和挖掘方法，着重理解当前应用的挖掘技术的基本类别。忽略具体的技术，数据挖掘方法可以按功能或应用来进行分类。这样，有四种主要的处理算法和规则：(1) 分类；(2) 关联；(3) 序列；(4) 聚类。

分类

分类 (classification) 是说一个项目或事件是否属于特定的数据子集或类别，分类算法就是挖掘和定义这些规则的过程。这种技术可能是解决各种商业问题时适用性最强的一种。分类有两个子过程：(1) 建立模型；(2) 预测类型。

举个例子，假如我们想利用数据仓库的数据挖掘出顾客的购买模式。分类模型要建

立顾客属性（如年龄、性别、收入等）和产品购买情况（如豪华轿车、音乐会入场券、衣服等）的对应关系。如果找到一组适当的预测属性，模型可以预测一群顾客中有哪些在下个月可能会购买。另外，还可以利用数据记录对顾客数据库做选择查询，从而有针对性地促销，或者制定邮寄列表。

分类还包括其他的问题。比如“哪些家庭可能对即将到来的直邮促销活动有所反应”；“在下次股票市场整顿后，投资组合中哪支股票被看好”；“现在的哪些保险索赔最可能是骗局”。通过建立和提炼商业问题的预测模型，数据挖掘分类方法可以为类似的问题提供有用而且相当准确的答案。

一般地说，分类方法的模型中包括普通的 IF-THEN 规则①。因为我们是寻找一个类别的可能的成员，所以在确定一个特定规则时容许出现三种情况，或说三种子类别：

1. 确切规则（exact rule）。不允许出现例外情况——就是说，每个 IF 对应一个确切的 THEN。这种方式确定了成员的最大可能的类别：100% 的可能。
2. 强规则（strong rule）。允许出现例外，但是对出现例外的范围有所限制。这种方式确定了高概率下的成员类别：90% ~100% 的可能性。
3. 弱规则（probalistic rule）。条件概率 P（THEN | IF）与概率 P（THEN）相关。这种方法确定可衡量的可能性下成员类别：X% 的可能性。

关联

关联（association）分析技术是从操作数据库的所有细节或事务中抽取频繁出现的模式。这种方式促进了关联规则的发展，关联规则总结了一组事件或条目与其他事件或条目的相互联系。关联算法下的规则经常这样表述，如“包含 A、B、C 项的记录中有 83% 的记录也包括 D、E 项。”其中的百分比表示规则的可信程度。关联在规则两边可以有任意多个条目。

使用关联分析方法的一个普通例子是购物篮分析。通过网络连接，零售商可以采用销售点的系统（比如在食品店常见的价格扫描设备）产生的数据进行挖掘。经过分析购物者篮子中的产品，并使用关联规则算法对大量篮子进行比较，就可以发现特定产品之间的密切关系了。关联方法的结果可能是“29% 的购买 X 组合的顾客，买了一套大玻璃杯”，或者“68% 的购买饮料的顾客购买了饼干”。这类信息可以帮助设计商店布局，制定促销内容或者货架摆放（比如在卖饮料的货架旁边放置 10 英尺高的饼干做展示）。一些商品的相关性（如酸奶和脱脂奶，啤酒和奶酪）可能相对直观一些，但是采用关联分析的数据挖掘可以发现更隐蔽但很有用的联系，比如买尿布的男士也买啤酒（这是真的）。

序列

序列（sequence）或时序分析方法常用于与时间相关的事件分析上，比如对感兴趣的利率波动或股票业绩的预测，它们都是基于一系列的已知事件。通过这种分析，可以发现各种隐藏的趋势，对未来有很高的预期价值。

---

① 见第 8 章，基于规则的推理。

直邮中可以看到普遍的序列分析应用。它常出现在特定顾客或顾客群的相关分析中。使用这样的信息,可以根据已知的顾客购买情况有针对性地邮寄特定商品的目录。比如有一个购买序列,父母在租影碟看电影的两周内,会给孩子购买与电影相关的促销玩具。这个序列暗示了在进行此类玩具促销时,应针对由影碟租赁所得出的顾客名单。另一个序列模式是顾客购买一台微波炉之前,往往会决定买一套东西。通过掌握顾客购买模式,尤其是使用信用卡的交易模式,我们就能很有针对性地制作直邮列表,进行促销和制定市场战略,集中精力说服那些最有可能的顾客进行购买。

另外一个技术点是如何确定相似序列。在一个典型的序列分析中,输出结果是按时间顺序排列的时序模式 ;但在相似序列分析中,目标是找到一群时序排列的序列。一个最普通的例子,大型零售商采用相似序列分析不相关部门中相似的销售情况。利用这样的信息,可以发现盈利性更好的促销、顾客流或商场布局。相似序列技术也在平衡投资组合中有所应用,用以识别有相似价格波动的股票证券。

聚类

有很多情况下,要定义数据类的参数十分困难,甚至是不可能的事情。这些时候,可以用聚类 (cluster) 方法进行划分。就某个标准或者某些标准来说,每个集合中的数据元素具有一定的相似性。聚类是由于相似或者相近而聚在一起的,例如,聚类方法可以用于我的信用卡购物数据。可以发现,商务金卡上的膳食支出典型地是工作日消费,有一个大于 250 美元的平均值。然而,用于个人白金卡上的膳食支出明显地发生在周末,有一个平均值,为 175 美元。有 65% 的时候还会要一瓶酒。

聚类处理一般用于特定的事件,比如对用户注销信用卡的研究。经过分析这类用户的特征,可以发现一些对信用卡发行商很有用的规则,以减少今后的注销信用卡的情况。

数据挖掘技术

现在,你可能已经意识到数据挖掘的潜力所在,它可以为原先的问题提供新的答案,能够通过发现而产生新的知识和理解。采用数据挖掘技术,提出和回答的问题的类型都不再受限制,回答问题的方法也多种多样,并且不断增加。与数据挖掘有很多可用的技术一样,建立挖掘模型同样有许多技术。这些技术大多在其他章节会进行更详细的讨论,为了保持连贯,这里对它们进行简单的介绍。

统计分析

尽管真正把握统计分析 (statistical analysis) 的要点需要特殊的本领,但是这种方法是数据挖掘技术中最成熟的一种,理解上也最为简单。如果 69% 的信用卡顾客购买 X 的同时也购买 Y,而 Y 从不单独卖出,那么很容易建立模型对 Y 的销售情况进行预测,这个预测有 69% 的准确度。当然,我们更有兴趣的是研究产品 X 的销售情况。

在这里,传统的统计分析方法和数据挖掘方法产生了分歧。传统的统计建模技术如回归分析对建立描述预测数据点的线性模型比较有利,但是现实中的复杂数据模式一般

是非线性的。如果数据没有以线性模型很好地描述出来,或者数据集包括很多孤立点[①],传统的统计模型经常带来负面影响。虽然这些已经超出了本书要论述的范围,但是提到数据挖掘也需要统计技术就足够了。统计技术可以适应非线性、多孤立点、非数字数据的环境,这些在数据仓库环境下都是很常见的。

神经网络、智能算法和模糊逻辑

如第 10 章所述,神经网络试图反映人类大脑的工作机理,通过建立有学习能力的数学结构进行模式识别。这样,建立非线性的预测模型,研究变量组合以及不同的组合是如何影响数据集的,就成为可能了。机器学习技术,比如遗传算法和模糊逻辑,可以从复杂的不精确的数据中挖掘含义,析取模式,探测趋势,这些都是人类和传统的自动分析技术无法做到的。因为有了这种能力,神经计算和机器学习技术在数据挖掘中有了广泛应用,也解决了相当多复杂的商业问题。

决策树

回忆第 5 章讨论的问题构造工具,决策树是一种概念上相对简单的数学方法,每个事件或决策都对后续事件产生影响。比如,一个简单决策树要对从事户内、户外活动进行决策,如果在初始选择时选择了户内,那么下一步的决策就是“楼上/楼下”,而不是“露天/遮蔽处”。随着数据集不断划分成独立的小组,就建成了一个预测模型。决策树经常在数据挖掘的应用中帮助进行数据仓库中项或者事件的分类。

知识发现的过程

尽管挖掘数据仓库的规则并不存在,对每个查询必须单独考虑,但是,仍然可以确定一些方针。表 12-2 描述了数据挖掘的一般过程。

表 12-2 知识发现的一般过程

- 选择研究主题:对商业问题有一定的理解
- 识别目标数据集:识别与研究有关的数据
- 清理并预处理数据:识别遗漏数据、数据噪音等
- 建立数据模型:使用挖掘软件建立数学模型,解释不同输入对输出的影响
- 挖掘数据:寻找有意义的模式(不是所有模式都是有意义的)
- 解释和提炼:回顾初步挖掘的结果(如果有必要,要对模型进行提炼),以便对已识别的有趣的模式有更多理解
- 预测:使用提炼过的模型,对输出未知的一系列输入进行输出预测
- 共享模型:一旦确认,将模型提供给其他数据仓库用户

要想加深对数据挖掘过程的理解,需要看一下它是怎样实际运行的。虽然我们不能在这本书中做到这一点,但是我们可以看一些典型的、数据挖掘过程中产生的查询的实例。

① 孤立点是数据集中异常偏离均值的数据点。有些时候它们代表极低可能性的数据点。而有些时候,可能暗示当前模型没有涉及的更复杂的关系。

例如，建模的 SQL 表述如下所示：

```
CREATE MODEL promo_list
Income character input, age integer input... respond character output
```

注意，CREATE MODEL 与标准的 SQL 语言中 CREATE TABLE 很相似。数据挖掘软件设计人员这样做是有目的的。新建的模型可以作为一个表或者视图出现在后边的 SQL 处理中。上边的 CREATE MODEL 使不同的输入与目的输出相关，比如收入水平（高、中、低）和年龄与直邮响应比率（响应为 yes 或者 no）相关。这只是一个简单的例子。对要考虑的输入并没有数量和类型上的限制。一个有代表性的实际应用涉及几十个到一百或更多的属性。

运用模型对一个数据集进行处理时，只需要将数据插入模型，标准的 SQL 命令如下：

```
INSERT INTO mail_list SELECT income, age, respond
FROM client_list,
WHERE mail = "Q2_Southeast"
```

这个过程自动地建立模型表的附加视图，用来理解关系和预测未来收入（mail_list_UNDERSTAND 和 mail_list_PREDICT）。一旦建立了模型，它就代表了对公司直邮战略做出反应的客户的大致情况。这是决策者拥有的最好信息。如果要查看模型产生的信息，可以用 SQL 语句对表进行操作：

```
SELECT * FROM mail_list_UNDERSTAND
  WHERE input_column_name = "income" and
  input_column_value = "high" and
  output_column_name = "respond" and
  and output_column_value = "yes"
ORDER BY importance, conjunctionid
```

向迈阿密的新的地区促销提供直邮列表时，可以用 SQL 访问 PREDICT 表中反应预测为 yes 的记录：

```
SELECT name FROM client_list, mail_list_PREDICT
WHERE city = "Miami" AND respond = "yes"
```

模型和相关的视图一旦建立，就可以当做数据库中的表，可以进行适当的查看、连接。模型表不仅对于创建者是可见的，对任何数据仓库访问者、可以产生 SQL 的用户也是可见的。其他用户可以把此模型当作模板使用，并做轻微的改动。实际过程比这个例子要困难得多，但是基本的步骤是相同的，采用的 SQL 语句也是相同的。

12.4　当前数据挖掘面临的限制和挑战

需要明确，尽管数据挖掘对于商业问题具有很强的潜能和价值，它还是一种新兴技术，还远不发达。当前，贸易数据挖掘产品的大多数开发商的规模都相对较小。他们的产品一般卖给更有实力的大型软件开发商或者将产品特许给他们，集成在当前的软件产品

中。于是,数据挖掘产品的发展受到了严重的限制。这一节将对几点挑战进行阐述。

遗漏信息的识别

当前数据挖掘已经不是新鲜概念,传统的数据库已经被新的友好的数据仓库替代,而数据仓库的设计者又面临着 ODS 向数据仓库统一形式的转换问题。虽然这种转换十分彻底,技术上也比较先进,但是不能察觉原始数据集是否包含了进行有效挖掘所必需的元素。比如,ODS 中包含的数据一般不包含领域 (field) 知识。或者说,不是所有的特定应用领域的知识都在数据中有所体现。从过去的讨论中可以知道,一般地,ODS 中的数据只限于与数据库相关的操作应用需要的数据。即使与应用相关的查询可以成功地执行,也有很多一般的查询不能正常执行。数据挖掘应用需要具有记录数据的功能。这样在把数据加载到数据仓库之前,可以先检查属性是否充足。例如,如果决策者想通过病人的数据库研究疟疾,那么所有加载进数据仓库的病人数据集必须包括红细胞数量。没有这些数据,不可能做出有效的诊断。

数据噪音和空缺值

事实上,所有的操作数据库都受到不同程度的"污染"。一些数据属性依靠主观衡量或者判断,这些都会引起错误,出现一些不正确的项类别。当属性的值或者数据分类出现错误的时候,我们说这是数据噪音。挖掘应用不能忽视数据噪音的问题,它对挖掘出的规则的整体准确性有不良影响。当前的数据挖掘系统只是用统计技术处理噪音,这要求了解噪音的来源。将来的系统可能包含更复杂的机制来处理空缺值和噪音数据,如空缺值的推测,空缺值均值推测的贝叶斯技术和其他一些数据推测方法。

大型数据库和高维度

操作数据库本身就很大,是动态的 随着数据增加、修改或删除,它们的内容也不断变化。由于这种变化,从数据集得到的规则也越来越不准确。由于要求数据是及时的,因而要求数据维度不断增加,以不断地发现知识。或者说,为了避免挖掘到的模式只反应一个时期的特征,数据模型必须根据一个不断变化的时间敏感数据进行更新。这就产生了一个具有相当广度和深度的问题空间,对搜索的计算能力的要求大大提高。未来的应用一定要解决这个螺旋式的问题,在不丢失重要属性的前提下把问题空间分解为可管理的小块。

12.5 数据可视化——"看到"数据

数据可视化 (data visualization) 是数字型数据转换成有意义的图像的过程。源数据可能有若干不同的来源,包括卫星图像、海下声速测量、勘测或者计算机模拟等。由于数据量极大,信息复杂,而且包含隐藏的模式,要解释这些数据是很难的事情。人的大脑可以处理很大规模的视觉信息,立即辨识众多不同的自然对象。数据可视化技术正是有效利用了人类视觉系统的优点,将物理属性与数据联系起来,辅助进行复杂数据集的分析。反

射及其他照明效果、颜色、阴影的方向和大小、物体间的相对尺寸和距离、速度、曲率、透明度等，这些是数据可视化常用技术的一小部分而已。

比如，表 12-3 是一些现实生活中的天气传感器的数据实例。正如你所看到的，我们根本没有通过看这些数据得到有用信息的本领。

表 12-3　天气传感器的数据举例

偏移	X	Y	Z	矢量
0.000000	- 17.833000	6.661000	0.011000	- 0.118000
0.000000	- 17.773001	6.683000	0.015000	- 0.122000
0.000000	- 17.673000	6.718000	0.009000	- 0.143000
0.000000	- 17.563999	6.757000	0.008000	- 0.163000
0.000000	- 17.461000	6.802000	0.011000	- 0.165000
0.000000	- 17.375000	6.865000	0.012000	- 0.149000
0.000000	- 17.297001	6.940000	0.009000	- 0.127000
0.000000	- 17.239000	7.016000	0.010000	- 0.114000
0.000000	- 17.200001	7.091000	0.018000	- 0.116000
0.000000	- 17.170000	7.167000	0.030000	- 0.133000
0.000000	- 17.169001	7.256000	0.050000	- 0.148000
0.000000	- 17.187000	7.345000	0.064000	- 0.152000
0.000000	- 17.226000	7.429000	0.074000	- 0.153000
0.000000	- 17.290001	7.509000	0.088000	- 0.152000
0.000000	- 17.375999	7.583000	0.106000	- 0.146000
0.000000	- 17.488001	7.645000	0.112000	- 0.145000
0.000000	- 17.620001	7.692000	0.118000	- 0.156000
0.000000	- 17.768000	7.724000	0.126000	- 0.158000
0.000000	- 17.931000	7.737000	0.133000	- 0.143000
0.000000	- 18.106001	7.736000	0.128000	- 0.134000
0.000000	- 18.284000	7.719000	0.121000	- 0.132000
0.000000	- 18.450001	7.688000	0.121000	- 0.129000

使用数据可视化技术建立传感器数据的三维模型，得到图 12-3 所示的形状。阴影代表暴风雨的相对剧烈程度，最黑的部分代表最猛烈的暴风雨①。

根据源数据建立多维结构和模型，是决策支持技术的重要革新之一。在这一部分，我们继续讨论多维分析（12-2 节开始的）和 n 维数据可视化。

一点历史

早期的数据可视化的概念源于统计和科学学科。早期的工作包括多维或多元数据集的二维分析。这些分析使用一系列的统计图像和图表，来反映跨维数据的类似性质。

① 通常的数据可视化的图像经过染色，颜色（而不是阴影）可看做是数据值的指示器。在本书中，我们以阴影的灰度来区别数据值。

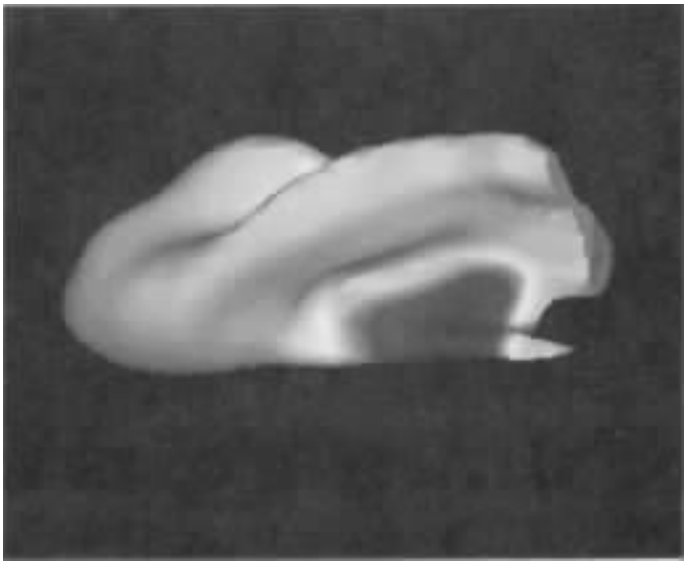


图 12-3 数据可视化:天气传感器的数据预示着暴风雨

数据可视化的许多进步应当归功于伊利诺斯州大学的 NCSA (National Center for Supercomputing Applications)。多维数据分析的一个实际应用是由 NCSA 负责的洛杉矶的烟雾项目。NCSA 在洛杉矶盆地的三维地图上添加了来自烟雾数据的计算机模拟动画,这样就可以指出污染的主要影响因素,并准确预报整个盆地范围内的烟雾情况及其运动变化。

数据可视化的一些最新发展当数 Xerox PARC 在虚拟现实领域的研究。三维可视化程序使用户可以在大型数据集中"驰骋",从不同角度观察数据,对代表数据的物体进行检查和重置。这些创新的数据分析方法在军事和工业的模拟中得到了实际应用。计算机可以演示都市坦克的虚拟大战,或者模拟港口的大型油轮的停泊。于是,我们可以从经验中学习,而不必担心真实训练中由于差错而引起的杂乱现场。

人类视觉感知和数据可视化

如果你现在走出门,你的视觉会立即接触到各种各样的物体——汽车、楼房、人、树、狗——所有这些都有一个连贯、有意义的组织体系。你的大脑通过边界、运动和距离对这些物体进行处理,并组织成多维的整体,以便回忆起它们的特征。这些看似无用的处理过程是连续的,而且是潜意识的。

数据可视化的功能强大不仅是因为人类的感知由视觉皮层控制,还有一个原因:将物体转化为信息的过程简直太迅速了。使用数据可视化,可以显示出大量的信息,加速数据中隐藏模式的辨别。你很容易理解图 12-3 中的图像,而理解表 12-3 却十分困难。有这样的一个对比,你应该意识到数据可视化作为分析工具的价值所在了。

还是那句老俗语,"一副图值一千句话",为了对数据可视化的价值有更好的理解,我们再看几个例子,以视觉形式表现的复杂数据集。

图 12-4 中是一个私有的全球计算机网络的动态数据集的模型，显示了其连通性、相关吞吐量和使用情况。连接节点的线条中，颜色最浅的代表吞吐量的最低值，最深的代表最高值。每个地面节点的和卫星节点的相对垂直高度代表当前活动的处理数目，或者使用该节点的用户数。这个图像是一个网络活动的统计表示，这些数据很容易转换为网络状态的动画，使分析人员看到随时间变化的使用模式，也能够指出节点的使用高峰或者超载情况。

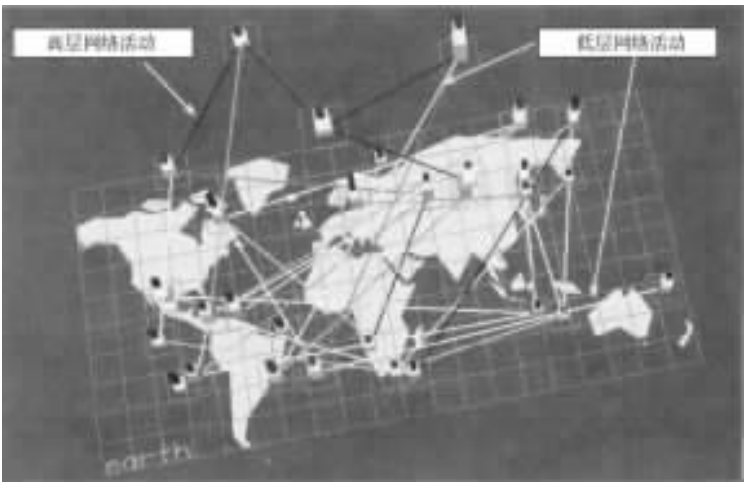


图 12-4　全球私有计算机网络活动的可视化数据

图 12-5 是交互式的统计视图，用于控制天然气管道。地图和管道网络图显示出每个压缩站，每个站点的总流量用站点的高度表示。虽然灰度表示并不容易分辨，但最显著的站点在本案例中用特别显著的高度表示（在彩图中以大红色表示），可以引起注意。操作者可以进行下钻操作来仔细研究站点状况——指向站点就会弹出附带数据的有关文本、表格、图表或详细的图形，比如站点中每个气轮机的情况。

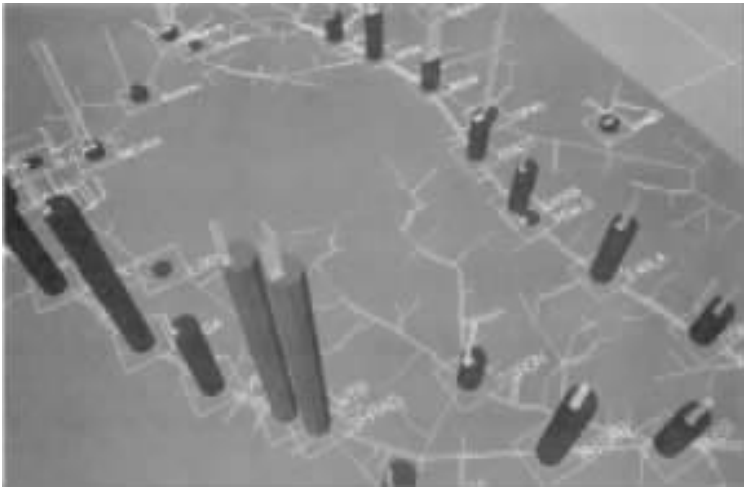


图 12-5　天然气管道分析

图 12-6 显示了三维度表示可以给最最普通的打印报告带来什么好处。管理者不再面对提交上来的曲线和对投资组合的过时的影响，也不用再等着对不同情况的分析报告了，他们可以直接与曲线进行交互的“What-if”分析。

图 12-6 风险分析报告

最让人讨厌的事莫过于在体育赛事或者电影的直播快要到关键时刻，尤其正值精彩部分时接到调查公司的电话。但是你肯定明白，这些电话调查是很重要的，开销也很大。但是通常情况下，这些调查人感兴趣的话题并不是你所感兴趣的。可是调查人只能随机地拨打成千上万的电话直至得到足够的调查结果。如果投票系统采用了数据可视化技术，人们就可以参与他们感兴趣的调查。这样想像一下吧，在电视或者广播里给调查问题做广告，并公布一个 1-800 电话号码。感兴趣的观众或听众可以拨打电话，然后按 1、2、3 进行投票，表示他们选择“是、否、其他”。数据库与投票系统相关联，通过辨认打电话的

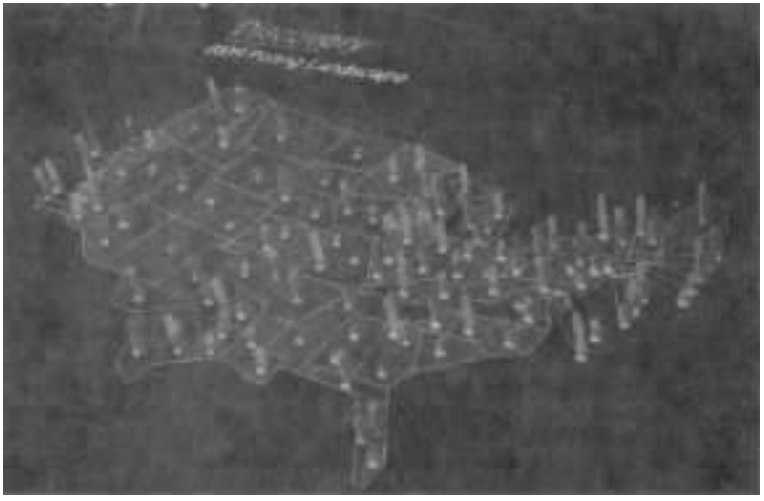


图 12-7 交互式的电话投票发现

人来保证一个人只有一次投票权利。数据可视化应用把收集的观点数据转换成来电地区的三维地图，用不同的带阴影的垂直条来描述每个选择出现的频率（如图 12-7 所示）。用户点击感兴趣的地区查看具体情况，而且可以不断点击，直到看到每个区的投票结果。

地理信息系统

地理信息系统（geographic information system，GIS）是一种特定目的的数字数据库，其中普通的空间定位系统作为主要的参考方式。全面的 GIS 需要有：⑴ 数据输入，来源有地图、航空图像、卫星、勘测等；⑵ 数据存储、检索和查询；⑶ 数据转换、分析和建模，包括空间统计；⑷ 数据报告，比如地图报告和计划。

很多人都对 GIS 下过定义，这些定义强调了 GIS 的不同方面。一些定义没有提到 GIS 的真正威力（整合信息和辅助决策的能力），但是所有的都包括了它的本质特征：空间查询和数据分析。

在本书中，我们对 GIS 做如下定义：GIS 是一个硬件、软件和程序的系统，用于支持空间数据的捕获、管理、控制、分析、建模和显示，来解决计划和管理中的复杂问题。

或者说，GIS 是一个具有专门的空间数据功能的数据库系统，也是处理这些数据的一套流程。

空间数据

空间是可以以地图形式存储的元素。这些元素惟一地对应地球表面的一个位置。空间数据（spatial data）有三个基本组成：点、线和型。

- 点（point）：点是二维或三维空间中的独立的位置。例如，美国地图上的代表城市的点。
- 线（line）：线是独立的树结构或网络结构的元素。例如，河流或道路系统。
- 型（polygon）：型可以是独立的、相连的或嵌套的。例如，地图上的国界线或海岸线。

属性数据

除了空间数据，GIS 系统还要能够处理属性数据（attribute data）。属性数据，简单地说，就是地图上看到的对空间数据的描述。比如，宾夕法尼亚州的匹兹堡地图会有一些属性数据描述海拔、土地使用、边界信息。这些信息一般是以表格形式保存的，当做一个普通的数据库来处理。如果地图上有变化（比如，城市的区域划分），可以修改属性数据，变化也就在地图上反映出来了。

GIS 经常被说成是“20 世纪 90 年代的电子数据表格”。在 80 年代，计算机的电子数据表格改变了人们组织和使用信息的方式，GIS 今天也在继续这样的工作，而且是更强烈地改变。GIS 明确了空间的特征和模式，因而促进了有限资源的合理使用。

举一些 GIS 适用的决策情况的例子。

- 在这里建一条大路有多大意义？
- 法定的地区界线设在这里还是那里？

- 我们把现存机场扩建还是重建新的？
- 这里十年内有足够的学校供孩子们上学吗？
- 我们应当保护受威胁的环境的哪一部分？
- 废旧设备对当地人的健康状况会有什么影响？
- 在哪里重新培养新的自然生物从环境上讲才是明智的？
- 如果出现各种可能的灾害，我们有足够的能力为当地和周围的人们提供服务吗？

数据可视化技术的应用

我们生活的时代充满了严酷的竞争，因而需要更多更快的信息。数据可视化可以帮助我们，它似乎涉及了很多的应用领域。现在，银行业中数据可视化用于信用评分和风险分析，政府将它用于欺诈分析和毒品侦缉，各种行业都有效地利用了可用的数据可视化技术和应用。

表 12-4 简单列举了典型的数据可视化技术的商业应用。

表 12-4 数据可视化技术的典型应用

零售业	远程通讯
• 客户产品交叉销售分析	• 呼叫中心管理
• 电子银行管理	• 网络运行管理
• 信用风险	• 服务政策分析
• 产品定价策略	• 负荷模型分析
政府	运输
• 预算分析	• 收益分析
• 资源管理	• 资产利用
• 经济分析	• 车队管理
• 欺诈侦测和毒品贸易分析	资本市场
保险	• 风险评价
• 资产负债管理	• 衍生品交易
• 保险精算的建模	• 机构销售系统
• 工作流管理和分析	• 零售营销
保健医药	资产管理
• 索赔分析	• 投资组合分析
• 病人行为分析	• 资产分配
• 治疗分析	• 投资最优化

12.6 SIFTWARE 技术

数据可视化产品市场上，开发商有进有退，但是有一些开发商的确占据了一部分市场，而且有了一定的顾客忠诚度。下边的部分将简要介绍几个行业领先者的情况和特点。

Red Brick

Red Brick 的数据挖掘技术和产品已经有所作为，其中两个客户是 H. E. B. 和 Hewlett-Packard（惠普）。

H. E. B. 是位于得克萨斯州圣安东尼奥的一家公司，销售额约为 45 亿美元，拥有 225 家连锁店和 50000 名职员。它的需求比较简单，同时有 Red Brick 的数据库和 Hewlett-Packard 公司的服务器支持，因而在品种管理方面（从设计到亮相不到 9 个月的时间）做得很好。

以前，市场信息部门直接从用户那里获取特殊需求的信息，编写程序来析取信息，约一周之后将信息反馈给用户。这对于大多数商业决策来说不够及时，有些时候也不是顾客当初真正的需要。

1990 年，组织进行了品种管理的改造。品种经理作为管理种类的“CEO”，负责该品种产品的赢利和亏损，制定购买哪些产品的最后决策，以及产品的货架布局等。同时他也负责指明哪些连锁店出售哪些产品。虽然 H. E. B. 的连锁店仅在得克萨斯州范围内，但是它的市场十分多样化，一些接近墨西哥的连锁店有 98% 的西班牙裔顾客，而达拉斯连锁店可能只有 2% 的西班牙裔顾客。不考虑具体连锁店的决策品种经理的区别主要在于销售规划和市场营销的策略上。

三年里，品种经理提高了谈判能力、技术能力和合作技巧，越来越需要更为及时的决策支持信息。截至 1993 年 9 月，公司完成了用户调查和需求分析。1994 年 3 月，最终系统交付使用，一直无故障运转。公司按星期、按条目（257000 个通用产品编码，即 UPC）、按连锁店维护了两年的数据，约有 4 亿条详细记录。总结文件按时间和整个公司进行维护，这是个优点。

系统目标是使所有查询能够在 4 秒钟内得到响应，一些涉及长期、大量条目的趋势报告在 30 至 40 秒钟内得到响应。系统特意设计得很简单，不只面向技术性的用户。系统具体到可以由用户指定时间、地点和产品的程序。

H. E. B. 意识到，品种经理采用基于事实的决策方法，决定哪些产品分给哪些连锁店，分配多少，还有合适的产品搭配。过去的品种经理一般都是从连锁店采购员中直接提拔的，而现在都是从财务、人力资源等其他运作部门调来的。这种办法之所以可行，是因为系统为缺乏产品知识的人们提供了同等时间的经验。

Red Brick 技术的另一个市场应用是 Hewlett-Packard，全球硬件系统头号供应商。HP 以生产高质量产品著称，为了维护声誉，它需要提供产品销售和售后期的服务和支持。

HP 的全球消费者支持组织（Worldwide Customer Support Organization，WCSO）负责向它的硬件、软件客户提供支持服务。几年里，WCSO 使用财务、会计、产品和服务合同的数据仓库来支持决策活动。WCSO 信息管理系统负责开发和支持这个数据仓库。

直到 1994 年，WCSO 信息管理采用基于两个 HP3000/Allbase 系统和一个 IBM DB2 的数据仓库支持商业信息查询。这是首次尝试通过收集、集成和存储与客户支持相关的数据来支持决策。WCSO 的用户越来越依赖于数据仓库，他们开始要求有更好的性能、附加的数据，而且要更及时地获取数据。

这个数据仓库结构越来越不能满足用户日益增长的需求。用户想要更快地获得信息,但是加载和查询的负担随着数据量的增加也不断加重。WCSO 决定再建一个数据仓库,希望显著提高加载和查询的性能,提高有效性,在不牺牲性能的前提下支持大量数据。

HP 选择了 Red Brick 的软件和 HP9000,项目首先把当前的三个数据库合并成一个称为 Discovery 的数据仓库。这样减小了规模,节省了大量费用,提高了管理和支持数据仓库的资源使用效率。今天,Discovery 支持大约 250 个地区的营销、财务和管理业务,遍布美国、欧洲和亚太地区。用户把查询结果输入桌面的报告生成器,把信息加载到工作表中,或将数据输入行政信息系统。用户满意度明显上升,这些要归功于 Discovery 的显著提高的性能和改造后的商业观点。

Oracle

建在 SP2 上的 Oracle 为大型的数据挖掘提供了强大的功能和良好的性能。数据散布在多重 SP2 处理器上,作为单纯的图像,提供对大型数据的快速访问。Oracle Parallel Query 允许多个用户同时提交复杂查询。每个复杂查询可以被分解,然后由几个处理器同时进行处理。执行时间可以从整个晚上缩减到几分钟,这样组织决策就更加迅速了。

Oracle 提供了可以帮助用户创建、管理和使用数据仓库的产品。Oracle 有一套连通的产品,提供对许多流行的数据库的透明访问,这样,用户可以将原先应用的数据转移到 SP2 上的数据仓库。

它的一些技术举例如下:

- 佛罗里达州迈阿密的 John Alden 保险公司采用了 SP2 上的 Oracle Pavallel Query 来挖掘医疗保健信息,使得一般商业查询的响应时间有了数量级水平上的改善。
- ShopKo 是威斯康星州的大型经销商,在中西部和东北部有 128 个连锁店,收入达到 20 亿美元。ShopKo 选择了 SP2 来满足数据挖掘和关键任务销售应用的需要。
- Pacific Bell 和 U. S. West 是两大通信供应商,1995 年经介绍采用 Oracle 的数据仓库,要提高跟踪客户和识别新的服务需求的能力。Pacific Bell 的数据仓库提供了一套总结和压缩的信息作为决策支持系统的基础。第一个系统用于分析产品获利性,当前正在开发类似的营销、资本投资和采购的决策支持系统,还有两个附加的财务系统。
- U. S. West 已经开发了一个数据仓库系统,从三个运营公司里抽取数据来分析内部代码。在一个 9 CPU 均衡多重处理系统中运行 Oracle7 Release 7. 2,可以支持 20 个经理和营销专家的使用。下一阶段,数据仓库系统将供 400 多个服务代表访问,最终将扩展到了 45000 个。

Informix

Informix 是数据挖掘领域的主要竞争者之一,有一些成功的例子。比如,美国西北部头号杂货零售商、年收入达 12 亿美元的 Associated Grocers,将它过去的主机环境换成了基于 Informix 数据库技术的三层客户机/服务器结构。新系统使订单周期缩短了一半,降

低了存货费用,使公司能够以更低的成本为 350 个分店提供更广的产品选择范围。

1991 年,Asscoiated Grocers 开始由基于主机的信息系统转向开放系统。起初,公司使用 IBM RS/6000 硬件,1991 年后一直采用 HP 和 NCR。在评价关系数据库管理系统时,Associated Grocers 列举了它的需求,包括教育/训练、可升级性、技术支持、可靠的用户参考和未来产品的方向。

决定选择 Informix 作为全公司数据库标准后,Associated Grocers 把系统其他的部分也用三层模型装配起来。第一层采用 Microsoft Windows 和 Visual Basic 开发客户表达层和图形界面。第二层采用 HP (惠普) 硬件和 Open Environment 公司的软件包 OEC Developer Package,使用 Micro Focus COBOL 软件。第三层,也就是数据层,是 Informix 的在线数据库。

Associated Grocer 的基于 Informix 的系统为食品仓库提供了实时的存货信息。过去,进货都是人工进行,同时把相关的产品信息输入财务系统。现在相反,新的系统在接货码头就识别了产品。手持的无线频率设备可以将产品信息立即扫描进 Informix 数据库。产品被指定了存放位置,过期日由系统负责。签订单时,过期日比较早的产品首先出货。

食品仓库系统的一个扩展是 Postbill 系统,将存货的实体和财务记账区分开来。以前,直到财务系统刷新 (一般地在夜间进行) 后产品才允许出售。新的 Informix 系统允许即时的产品销售和分配,即使是刚刚进货的产品。

还有一个基于 Informix 的应用使得 Associated Grocers 能够经济地出售特色产品——周转慢的,按月订货而不按天订货的产品。Associated Grocers 不再以很高的成本来存储这些物品,而是与外部的专业仓库合作,及时提供客户需要的产品。独立的商店只是简单地从 Associated Grocers 订货。订单进入 Associated Grocers 的付款系统,被传送到专业的仓库,然后专业仓库将产品送至 Associated Grocers。特色产品也装在 Associated Grocers 的运送卡车上,同商店的其他订货一起进行配送。

Sybase

最近,Meta Group 公司进行了一项对数据仓库的兴趣和有关活动的调查。调查结果显示,70% 以上的被调查公司有数据仓库的项目正在预算之中,或者已经进行,平均的花费是 300 万美元,平均的建设期是 6 到 18 个月。

传统的数据仓库应用从操作系统中提取基本的商业数据,加以修改或进行转换以确保准确性和明确性 ,并以产品改造、客户规划或者“隐蔽网”(sneaker net) 的方式将数据转移到新的分析数据库系统。如果商业周期比较简单而且相对稳定,这些提取、编辑、下载、查询的系统是可以接受的,但是情况并不是这样。新的数据和数据结构增加了,现存的数据被改动了,甚至要添加上整个的数据库。

Sybase 数据仓库 WORKS 围绕数据仓库存储的四个重要功能而设计:

- 把来自多种数据源的数据合并起来
- 把数据转换为对企业而言是连续的和可以理解的
- 把数据分散到用户需要的地方
- 为用户提供快速的数据访问

Sybase 数据仓库 WORKS 的 Alliance Program 为组织建立和配置数据仓库提供了一个完全的、开放的、集成的解决方案。程序涉及对数据仓库发展的技术需求的全部范围，包括数据转换、数据分散、交互的数据访问。合作人已经承诺采用 WORKS 结构和 API，并且在营销和销售程序上与 Sybase 加强合作。

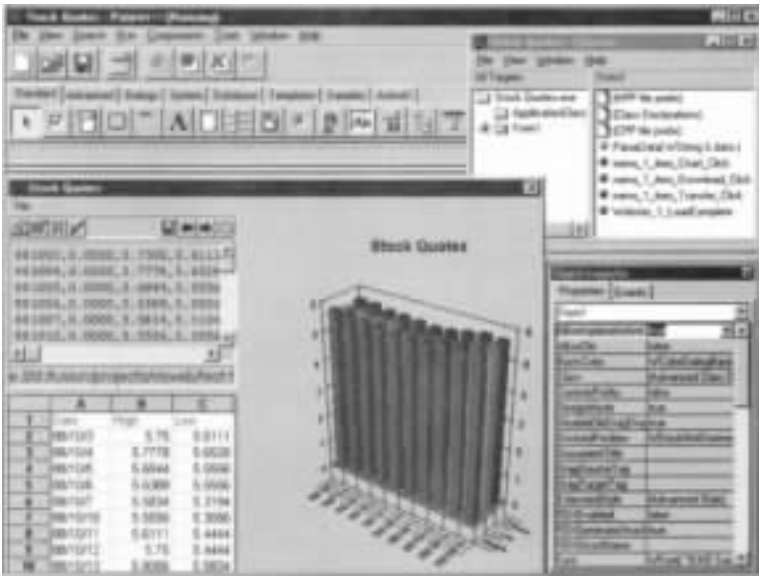


图 12-8 典型的 Sybase 数据仓库开发界面

Silicon Graphics

数据分析的进步在数据仓库存储方面实现了突破，现在将在数据挖掘新的解决方案方面继续有所作为。Silicon Graphics 开发了复杂的三维视图工具，加上数据挖掘软件，就可以找到传统的 SQL 技术所不能发现的数据模式和趋势，让用户注意到这些宝贵的信息，这必将有重要的意义。

使用 3D 穿梭 (fly-through) 技术，系统用户可以浏览顾客购买和销售渠道速度的信息，以把握趋势、观察模式。用户可以与数据直接交互，在模型中使用可视化计算进行“What-if”情景分析。这样，不用让本来就很忙的系统人员帮助分析，用户在这个过程中可以节省几天的时间，甚至几个月。

IBM

IBM 开发了一些决策支持工具，给用户一个强大而且易操作的数据仓库界面。IBM 信息仓库解决方案 (IBM Information Warehouse Solutions) 曾许诺提供开放的系统，允许用户选择决策支持工具来满足个人需要。

IBM 宣布与经由挑选的客户合作进行 Customer Partnership 项目，来验证数据挖掘技术的适用性。这将为客户提供 IBM 强大的数据挖掘技术，用来分析数据，寻求关键模式和关联。例如，Visa 和 IBM 在 1995 年签订协约，表达了合作的意愿。这将改变 Visa 及其

合作银行在全球范围内交换信息的方式。目标是加速信息的及时传递,改善关键的决策支持工具,尤其是全球机构的财务系统。

IBM Visualizer 提供了一个强大、综合的、即用的开发工具,可以支持最终用户的众多需求,包括查询、书写报表、数据分析、图表制作、商业计划和多媒体数据库。Visualizer 是一个基于工作站、面向对象的产品,插入附加功能(如上边提到的)十分容易。除了 DB2,Visualizer 还可以访问 Oracle 和 Sybase 的数据库。

IBM 还有其他一些基于工作平台、操作环境和数据库的决策支持产品。比如,IBM Application System (AS) 提供了客户端/服务器结构,还有面向 MVS 和 VM 的最广泛的决策支持功能。AS 能够访问许多不同的数据源,因而可以对这些环境下的决策进行支持。IBM Query Management Facility (QMF) 提供了 MVS、VM 和 CICS 环境下的查询、报告和图表功能。Data Interpretation System (DIS) 是一系列面向对象的工具,使用户可以存取、分析和显示而不需要技术帮助。它是基于局域网的客户端/服务器结构,可以对 IBM 和非 IBM 的关系数据库,以及 MVS、VM 环境下的特殊数据库进行存取。这些产品和其他一些产品都可以从 IBM 直接购买,在功能和性能方面都有很大的选择余地。

12.7　小　　结

数据可视化已经成为决策支持和组织战略中的重要组成部分。即将到来的信息时代,企业对组织和使用信息和知识的需求不断增加,发现信息中的隐藏结构和模式将越来越为关键。通过这项技术,我们当前对信息概念的理解("我们需要知道的东西")将最终转变为"只有这样做才可能知道的东西"。

复习题

1. 给数据挖掘下一个定义。
2. 描述决策支持行为从确认到发现的转变。
3. 联机分析处理 (OLAP) 是什么?
4. 解释多维 OLAP (MOLAP) 的概念。
5. 解释关系 OLAP (ROLAP) 的工作原理。
6. 比较 OLAP 的两种方法。它们有哪些不同之处?
7. 解释数据挖掘的分类技术。
8. 关联技术是如何应用到数据挖掘中的?
9. 列举并简要描述数据挖掘技术。
10. 列举并简要介绍知识发现的搜索过程。
11. 数据挖掘当前面临哪些限制和挑战?
12. 什么是数据可视化?

讨论题

1. 理解决策支持行为从确认到发现的转变，举例说明这种转变。说明这种转变为什么是可能的。
2. 分析市场上应用了 OLAP 概念的数据挖掘产品。描述它使用了什么方法和技术，是怎样进行挖掘的。
3. 分析市场上的数据挖掘产品，找到它的优点和局限性。
4. 数据可视化应当如何辅助决策？
5. 比较 12.6 节讨论的几种 siftware 技术。它们的特征是什么？它们有什么差别？

第 13 章

设计与构建决策支持系统

学习目标

- 掌握决策支持系统的两种基本开发策略
- 理解不同的决策支持系统分析与设计方法
- 了解决策支持系统的开发过程 (DDP)
- 比较传统的系统开发生命周期 (SDLC) 方法与决策支持系统开发过程 (DDP) 的不同
- 熟悉原型开发方法的过程
- 掌握原型开发的两种基本方法 :丢弃型原型开发和反复型原型开发
- 了解原型开发的优缺点
- 掌握决策支持系统开发人员需要的所有技术
- 理解最终用户计算的概念
- 了解最终用户 DSS 开发的优势和风险
- 了解选择决策支持系统开发工具的准则

小案例

快速原型开发 Allkauf 信息系统

1994 年夏天，德国零售连锁店 Allkauf 的管理层决定建立一个整个公司范围内的经理信息系统（EIS）。Allkauf 公司的常务董事 Erhard Jusche 说："过去我们靠经验和直觉干得非常成功。现在我们想通过现代化的技术，为当前和将来的管理层创建一个易于访问的信息库。"了解客户的需求是 Allkauf 的首要任务，它包括准确地知道产品是在哪一家店以及什么时候销售出去的，公司新的 EIS 应该能在五秒钟之内提供出这种信息。

Allkauf 拥有并经营着 75 家连锁超市，每一家都占地至少 6.5 万平方英尺，存放着 8000 多种不同的商品。构建 EIS 需要收集并集成大量的信息，公司庞大的规模使这项任务变得非常艰巨。此外，新系统的关键需求之一是能够迅速响应。由此可见，系统的规模、复杂程度和需求都清楚地表明了只有最先进的技术才能实现，这些技术包括客户机/服务器结构、多用户工具、Windows 操作系统、多维模型工具以及其他实现 EIS 所有功能的工具。当 Allkauf 走进市场，寻找能够实现系统所有需要功能的销售商时，它却很快发现这样的销售商非常缺乏。只有两家能够满足新系统的严格需要，Allkauf 最终选择了 Comshare。

Allkauf 信息系统（AIS）计划实现三个方面的功能 标准报告（standard reporting）、异常报告（exception reporting）以及特别分析（ad hoc analysis）。其中，特别分析又被认为是最重要的，因为 Allkauf 希望系统所有的用户都能够根据自身需要分析任何数据。

AIS 是用按部就班的进化型原型开发方法开发的。两名 Allkauf 的分析人员只用一个月就开发出了最初的系统原型。它的一个模块通过直接访问公司 OLAP 服务器上的数据动态地生成报告以监控销售情况，数据可以通过图表、分类和不同颜色来表示，用户可以得到更详细的数据。另一个系统的基本需求——查询的响应时间必须在五秒钟以内——也得到了满足。

系统的第二个模块（短期收益报告）在 1995 年 3 月完成并投入运行。第三个模块包括连锁店的所有项目的全部统计信息，最初计划由中心采购部门的 30 个成员使用。这项应用的最大特点是在一个维度上就有将近200000个成员，只有 Comshare 的新产品 Essbase 4.0 能够满足这种需求。但这个版本按计划在 1996 年 3 月正式发布，在此期间，Allkauf 在 Essbase 现有的版本上开发出了三个不大完善的应用系统。

使用进化型原型开发方法能够开发出用户可以迅速做出反馈的原型。这样，开发人员可以根据这些反馈改善系统，以更好地满足用户需求。Allkauf 的分析人员认为更加结构化的开发方法，如典型的生命周期法，不能从本质上提高系统的质量，而且更重要的是，它可能延误这三个基本模块的投入使用。但是需要注意到，Allkauf 所有主要的事务处理信息系统都是采用高度结构化的开发方法开发的。

通过使用 AIS，Allkauf 可以提高业务处理的效率，而且在业务分析中节约了宝贵的时间。Allkauf 的管理层深信，他们选择了最恰当的时间使用 EIS，并由此获得了竞争优势。

我们正走向决策制定和决策支持的世界。在这里，就像上面那个小例子所表明的那样，我们必须将注意转移到技术本身。关于 DSS 的一个最具有挑战性的问题就是它的设计与实施。这一章将讲述构建一个现代 DSS 的复杂性与设计问题，第 14 章将讨论 DSS 的实现中的各种问题，以及如何将它融入到组织中去。

通过以往的讨论，已经了解了多种类型的 DSS，每一种都有它们自己的开发工具、假设与基本模型。虽然我们可以把普通的 DSS 描述成一些一般组件的集合（见第 1 章），但最终产品在和其他系统（无论是多么相似的系统）比较时，必然具有一些特点和特殊的功能。因为没有最完善的 DSS，所以也没有设计和构建 DSS 的最好的途径。一些 DSS 的项目只需要几天就可以完成，而另一些系统却可能要花几年时间来设计并实现。尽管如此，那些系统还不能从真正意义上结束，而是一直处于动态地调整与完善状态中。

近年来，关于 DSS 开发方法的研究取得了很大进展（Arinza，1991；Saxena，1992），已经确定了 30 多种不同的 DSS 设计与构建方法。我们不能介绍每一种已经被确定、且正在被使用的方法，但是我们将讨论各种体现了 DSS 设计领域的精髓和复杂性的 DSS 开发策略。最重要的是本章和下一章讨论的前提：不管 DSS 和组织的其他应用系统在外表上多么相似，它们每一个的设计和实现过程都是绝对不同的。为了说明这一点，我们首先对 DSS 开发的基本策略做一个概述。

13.1　决策支持系统的分析与设计策略

DSS 的基本开发策略

在所有的 DSS 开发方法中，可以确定两种普通的策略：(1) 编制一个用户定制化的 DSS，(2) 采用 DSS 生成器。每一种方法都有其独特之处，而选择哪一种方法通常是根据组织机构的设置和问题的环境决定的。实际上，一个复杂的 DSS 很有可能在项目的不同设计阶段同时采用这两种方法。下面我们将简单地介绍这两种策略，而将重点放在每一种方法所采用的不同开发工具上，也就是本章后半部分将详细介绍的内容。

编制一个用户定制化的 DSS

这种策略或者采用一种通用编程语言（GPL），如 PASCAL 语言和 COBOL 语言；或者采用第四代编程语言（4GL），如 Delphi 和 Visual C++。虽然早期的 DSS 都是采用 GPL 设计的，但是大多数当今的系统都是通过结合 4GL 和 Java 程序模型开发的。它们提高了程序员的效率，减少了开发时间和工作量。虽然出现了许多专门支持 DSS 开发过程的应用程序，但是一个大型组织的 DSS 应该和目前存在的所有不同的操作系统都有接口。所以，为了支持统一的接口，并连接 DSS 和组织的其他应用系统，通常采用商业化的编程语言产品。

采用 DSS 生成器

DSS 生成器 (DSS generator) 是一种应用系统,使用它能够在 DSS 的设计与实施过程中少编数千条指令或程序。最常见的一种 DSS 生成器就是电子数据表格,如 Excel,Lotus 1-2-3 和 Quattro Pro。有大量的程序和组织的 DSS 应用系统,是以 DSS 生成器作为开发基础实现的。如果用一种电子表格作为最基本的 DSS 生成器,可以使用许多商业化的附加项辅助开发多种 DSS 分析机制。比如@RISK 就是一种复杂的用于评估可能性的工具。更加复杂的生成器有 Micro-Strategy 的 DSS Architect (见图 13-1),和传统的电子表格相比,它提高了设计人员的灵活度。虽然和直接使用编程语言开发相比,使用 DSS 生成器开发效率要高得多,但是和使用 4GL 以及其他策略相比,DSS 生成器限制了开发的灵活性和能够达到的复杂程度。

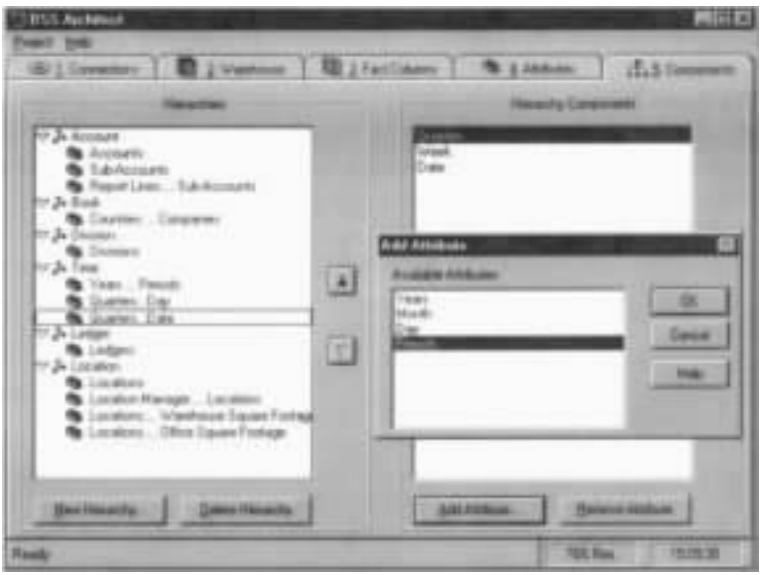


图 13-1 Micro-Strategy 的 DSS Architect

专用 DSS 生成器的出现是这种方法近期的一个进步。这些系统可以辅助开发高度结构化的专用 DSS。专用 DSS 生成器的例子包括实现复杂统计功能的 SAS,以及用于金融分析的 Commander FDC。

DSS 的分析与设计过程

有大量的分析与设计方法可以用于 DSS 的开发过程。我们将对每一个普通的步骤都做适当详尽程度的阐述,并且尽量给出各种不同方法之间的比较,从而说明它们作为 DSS 开发工具集中的一员所做出的贡献以及不足之处。

系统开发生命周期

系统开发生命周期 (system development life cycle,SDLC) 法体现了一种关于系统设计

与实施的普遍存在的观点。它将整个过程描绘成一系列递归的阶段，每一个阶段都有自己需要的输入、处理和输出。图 13-2 展现了 SDLC 方法的不同的阶段以及递归的特点。

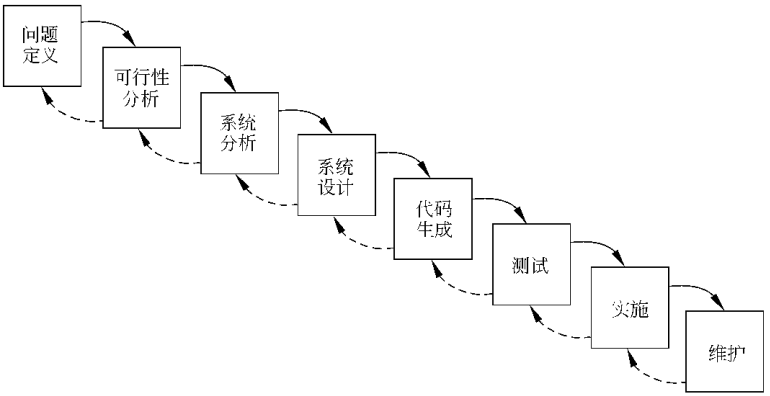


图 13-2　传统的系统开发生命周期

在问题定义 (problem definition) 阶段，系统分析人员评价组织中问题的性质，并得出一份描述这些问题和组织环境的文档。一旦完成了这些工作，就可以进入可行性分析 (feasibility analysis) 阶段，评价能否通过将若干个应用程序集成到信息系统中，有效地解决上一个阶段总结出的问题。这个阶段的工作包括将当前的实际系统转化成一系列的逻辑模型，从而从独立于系统实现的角度观察隐含的过程。这个阶段要综合考虑当前可用的技术、人力和财力资源、以及政治和法律上的约束，做出最初的可行性分析。

在系统分析 (system analysis) 阶段，系统分析员要关注对新系统的特殊需求的决定、收集资料和文档化。在这些需求的基础上，设计 (design) 阶段的工作包括为系统的每一个组成流程、数据项、以及它们之间的交互关系建立详细的模型。对最后的逻辑模型还要进行调整，从而突出对新系统的特殊需求，并解决旧系统的不足之处。在编程 (programming) 阶段，通过开发或直接获取为满足用户需求所必需的软件和硬件，将新系统的逻辑模型转换成物理系统。接下来是测试 (testing) 阶段。最终得到的系统被投入一系列测试性的运行，以确认它能否满足需要。系统实施 (implementation) 后还要进行最后的安装，维护 (maintenance) 阶段也就紧跟着开始了。为了提高系统的性能，同时也增加它的功能，还要对运行中的系统作各种各样的改善。

ROMC 分析

SDLC 方法的一种替代方案就是 ROMC 分析 (Sprague 和 Carlson，1982)。这种方法侧重于对开发过程中的表述 (R)、操作 (O)、记忆辅助 (M) 和控制 (C) 的分析和理解。

使用这种方法设计 DSS，系统分析人员要做出各种可行的、形象准确的表述 (representation)，以方便用户和系统沟通。表述的例子包括图形、图表、表格、目录、存货报表、输入表单和菜单等。

在操作 (operation) 分析中，系统分析人员主要为 DSS 确定那些必需的动作，从而推动系统中各种表述的生成和传递。这些动作包括对包含在 DSS 中的各种相关知识进行解

释、生成并打包。

记忆辅助 (memory aid) 是一些组件,它们为使用各种确定的表述和操作提供支持。记忆辅助的例子包括数据库、存储工作空间或黑板系统、以及内置的触发器,它提醒 DSS 使用者运行某种和当前任务相关的操作。

最后,控制 (control) 是帮助 DSS 从各种表述、操作和记忆辅助综合出一种特别的决策过程的机制。控制的例子包括帮助用户向 DSS 提交专门的请求或查询的机制,辅助用户改变 DSS 以适应自己独特的解决问题的方式或趋势的功能,以及通过例子指导用户学习使用 DSS 的特殊功能的模块,而不是让用户自己反复尝试。

功能分类分析

设计 DSS 的另一种方法是功能分类分析 (functional category analysis) (Banning, 1979)。这种方法的关键之处在于从一个总目录中确定一些特殊功能,它们对于开发一个特别的 DSS 是必需的。总目录包括所有可行的功能,如选择、聚合、估计、模拟、等化和优化。表 13-1 列出了不同的功能分类,并对每一种都做了简单描述。

表 13-1 功能分类分析

功能分类	执行的动作
选择	在知识库中查找需要说明的知识,或者作为新知识引出过程的输入
聚合	得到或引出一些统计信息,如平均数、总数、频率和分组
估计	得到或引出对模型参数的估计
模拟	得到或引出关于期望输出的知识,以及组织环境内的特殊行为所导致的后果
均衡	得到或引出关于维持平衡和问题一致性的必要条件的知识
优化	引出关于在限制的约束集合内,哪一个参数集能够最大化或者最小化一个给定的衡量标准的知识

通过使用功能分类分析,提出 DSS 的关键功能的组织能够做出更有效的安排,从而允许 DSS 的设计人员进行更加有针对性的详细分析。

DSS 的开发过程——DDP

虽然设计和构建组织的 DSS 与设计和构建组织的事务处理系统有许多相似之处,但标志着 DSS 的开发目的的半结构化问题和非结构化问题具有一些独特的性质,这指明了 DSS 的设计和构建也必须采用一种独特的方法。在 DSS 设计的初始阶段,遇到问题的经理们通常还不清楚需要哪些专门的信息,这些信息并不容易被识别。为了促进收集需要的 DSS 的功能,这个阶段要把重点放在原型开发上。在这一部分,我们确定系统开发的一个通用过程,再修正它使其适应 DSS 设计人员的特殊需求。因为每一个 DSS 的设计项目都是独特的,所以不一定会完成所有确定的活动。一种简化的、仅包含 DSS 设计的关键步骤的方法可能适合一个简单的 DSS 项目,而一个特别复杂的项目却要经过所有列出的步骤,完成所有的活动。我们的目的是讨论 DSS 开发过程 (DSS development process, DDP) 的一个通用的活动集合和开发阶段,它代表了 DSS 的典型设计方式。图 13-3 阐明

了这个过程包含的阶段。

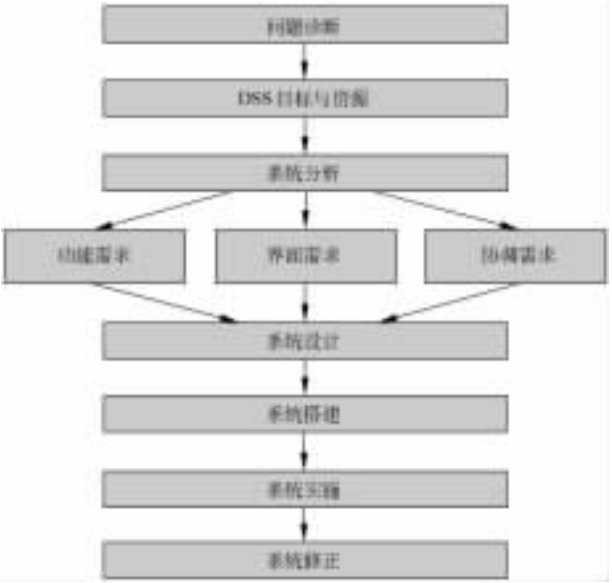


图 13-3　通用的 DSS 开发过程

DSS 开发的第一步是按规则定义和识别需要决策支持的问题。回顾第 3 章可以知道，问题的存在都是有症状的，可以通过这些症状识别出潜在的问题，也为决策支持提供了机会。

可以从很多方面发现决策支持的机会。DSS 除了支持解决识别出的组织中的问题，还可以帮助挖掘新的知识，提高公司的竞争地位，或提高其到达某个细分市场的能力。使用在第 11 和 12 章讨论过的数据挖掘和可视化技术，经理能够识别出一个单独的市场区域的购买模式，利用它可以提高公司的市场份额。另一方面，使用 DSS 可以帮助组织及时地发现问题，否则它们将不被觉察到，如资金挪用和偷窃行为。这些例子都说明了经理们必须保持警觉，寻找在组织的日常活动中使用决策支持的机会。

确定 DSS 的目标和资源

一旦确定决策支持的机会，就必须描述专门的目标和 DSS 所支持的关键决策。此外，还必须确定可用的资源，包括硬件、软件、当前技术和可用的知识。在这个阶段，设计人员要仔细识别 DSS 全部的目标。这些目标用来阐述清楚所规划的 DSS 的功能，关系到要管理的知识的种种类型、决策者可用知识的多少、以及系统的性能和特点。此外，还必须确定为 DSS 的使用者提供什么种类的支持。设计人员在这儿要决定 DSS 在处理确定的问题中扮演什么样的角色。表 13-2 列出了 DSS 所能提供的支持的种类。

在确定规划 DSS 的目标的同时，还必须确定如何衡量这些目标是否成功达到。这就需要建立专门的、有针对性的性能标准。例如考虑以下的目标集：

- 提高组织内决策的质量
- 至少将某种原材料的质量提高 12 个百分点

◆ 至少将某种分析的常规决策时间压缩 10 个百分点
◆ 增加产品 X 的市场份额

表 13-2 DSS 提供的支持种类

• 确定决策的需要	• 以建议、分析或评价的形式提供帮助
• 确定需要注意的具体问题	• 提高用户的创造力、想象力和洞察力
• 解决或帮助解决问题	• 促进多决策者之间的交互和沟通
• 帮助弥补人类决策者的知识限制	

尽管每一个列出的目标都对公司有明显的价值,但它们中只有两个能被精确地评定。第一个和最后一个目标的描述使其不能被客观地评价。先说明第一个例子,提高组织内决策的质量的是一个很好的目标,但若不首先确定评价决策质量的标准,就不能判断它是否达到。甚至即使决策质量通过 DSS 只得到了细微的提高,也可以说目标达到了。在这个例子中,如果组织为 DSS 的开发投资 100 万美元,提高的决策质量使组织的净利润增加了 10 个百分点,或许这才能成为目标达到的充分证据。试着向你的老板推销这个概念。

问题同样存在于列出的四个目标中的最后一个上。产品 X 的市场份额的任何增加都能满足这个目标。而第二和第三目标的叙述方式则考虑了如何评价目标是否达到,并不仅仅是评价目标的价值。虽然有些目标很难量化,或者在表面上是无形的,但是 DSS 开发项目的最终成功还是建立在确定清楚的目标以及评估这些目标时所采用的方法上。

系统分析

DSS 开发过程的这个阶段是为了详细地确定对 DSS 的需求。Holaspple, Park 和 Whinston 在 1993 年提出,对 DSS 的需求可分三个基本种类:(1) 功能需求 (functional requirement);(2) 界面需求 (interface requirement);(3) 协调需求 (coordination requirement)。

功能需求 DSS 这个类别的需求主要是存储、获取以及生成对解决问题有用的知识。其示例包括用一个专门的 DSS 存储大量的产品销售信息,提取某些特定的信息或信息的某些特征,并从这些信息中分析和估计可能会对销售产生影响的因素。

界面需求 可以证明,DSS 的性能是与它的界面质量和功能紧密相关的。对界面的需求主要是指 DSS 的交互功能。在这个阶段,设计人员必须确定各种对 DSS 的用户可行的交互渠道和方式,以及相应的条件。菜单、报表结构、命令界面和输出格式都要在开发过程的这个阶段确定。此外,开发人员还必须确定决策者对 DSS 所作查询的不同种类,以及最适合的 DSS 响应方式。

协调需求 除了完成规划的 DSS 的具体功能和交互机制,设计人员还必须确定系统的协调需求。这包括安排和决策过程相关的事件的时间,促进获得相关的信息,以及集成 DSS 包含的各种模型工具。协调需求的一个例子是,一个动作必须在另一个事件发生后才能初始化 (例如上一个会计期间的销售计划必须更新为实际数据后,才能制定下一个计

划)。

系统设计

DSS 开发过程的系统设计 (system design) 阶段要确定专门的物理组件、结构和开发平台。这个阶段最基本的一项活动是选择一套构建 DSS 的开发工具。一旦确定了开发工具,就要充分利用这些工具的优点和强项,同时在设计过程中也要避免平台已知的缺陷。我们将在本章的后面详细介绍各种可以用来设计 DSS 的工具和平台。

系统构建

在系统构建 (system construction) 阶段,设计人员采用反复型原型开发方法,在测试和用户反馈的基础上不断对系统进行细微的改善。这个改善的过程是持续整个阶段的,结果通常是最初的设计发生了显著的变化。对原型的每一次修改都促使 DSS 的用户提出没有反映在设计中的需求和愿望。此外,原型开发还可以逐步确定连接 DSS 和其他现存的组织应用系统的各种需要。我们将在本节的后面详细讨论和原型开发相关的活动。现在只需要知道设计阶段通常占用了开发 DSS 的大部分时间和精力。

系统实施

系统实施 (system implementation) 阶段的目标是测试、评估并配置一个文档齐全且功能完善的 DSS。这个阶段包括确定并评价 DSS 到底在多大程度上满足了用户需求,这一点非常重要。正如在设计阶段,为了完全挖掘并实现 DSS 的潜能,达到决策者确定的目标,开发人员必须不断修改系统。在这个阶段还要训练 DSS 的用户群,使他们了解系统的功能和结构。对这些用户群的训练还要持续到以后的系统维护和改善过程中。通过这一系列的开发活动,组织终于有了一个配置齐全,功能完善,能满足各种特殊需求,适应组织问题环境的决策支持系统。此后,过程的最后一步,逐步改善 (incremental adaptation) 就可以开始了。

逐步改善

DSS 开发过程的最后一步是没有终点的活动。在使用 DSS 所获得的经验的基础上必须不断重复前面的每一步,以提高 DSS 的能力。组织的 DSS 的运行生命结束后,又产生新的需求,以及为满足这些需求开发出的新功能。此外,环境中技术的变化和适应这些变化的愿望,导致开发过程新一轮的开始,这将对当前设计做出大的修改。可以看到,DSS 的开发是一个进化的、有生命的过程,它可以,而且必须产生一个对组织决策高度有效的支持机制。

SDLC 与 DSS 开发过程

我们已经确定多种常用的 DSS 设计与开发策略,既有高度阶段化的结构化策略,也有高度反复性的非结构化策略。为了进一步理解一个独特的 DSS 的设计开发过程的需要,我们必须详细介绍两种极端的开发策略,SDLC 方法和 DDP。

SDLC 方法已经发展了几十年,它依靠的是系统分析人员和开发人员的经验。这个过程有序化和结构化的特点作为一个关键因素,使它能够提供通过编程解决问题的工具,同时尽量减小人类解决问题时内在的偏差和认知的局限性。系统开发的生命周期方法是自顶向下的设计思想的代表。这说明了首先要确定系统确切的需求和特点,然后在接下来的系统设计和构建过程中满足它们。熟练的系统分析人员很快会指出,SDLC 方法只是实际过程的一个理想概括,通常为了推动特殊系统或组件的开发,必须反复采用自顶向下的设计方法。但是它的优点在于简单,以及在设计过程中利用必需结构的能力。时间证明 SDLC 方法是开发基于计算机的信息系统最简单有效的方法。但是,它的这个优点也导致它不能充分有效地设计与构建组织的 DSS。

不充分的根本原因在于一个 DSS 的目标,就是为半结构化问题提供支持。使用 SDLC 方法有一个前提假设,即在设计阶段开始以前就充分了解并识别了需要解决的问题的结构和环境。大多数恰当地采用 SDLC 方法的系统项目,大部分的问题环境、管理偏好、决策过程和外界条件是容易识别且稳定的。虽然 SDLC 方法的初期活动是和系统分析人员收集知识紧密相关的,但仍然要假设这些知识容易获得,而且能够按照系统需求分类。这种假设在 DSS 的设计和构建领域是站不住脚的。因为对 DSS 的设计人员来说,是缺乏对上述因素的了解的。通常,问题环境、管理偏好、决策过程和外界条件对用户来说也是未知的,而且在开发过程中会发生大的变化。这和生命周期方法的假设相矛盾:

> 一段时间后,最初按非结构化和半结构化规划的问题开始导致 DSS 项目发生改变,用户和 DSS 设计人员都会有这样的想法。当传统的生命周期设计方法在最初的问题定义上被提前终止,因而规则被"冻结"时,这一切就发生了。最后的 DSS 系统设计,仍然会在整个开发期间发生变化。

接下来的 DSS 设计需要一个独特的过程,允许设计方法在 DSS 的开发过程中发生大的、必要的变化。这种方法主要是使用原型开发方法进行各种活动。

原型开发方法

原型(prototyping)开发方法是一种常见的,而且越来越流行的系统开发方法。虽然对于如何正确地使用原型开发方法,是否将其作为生命周期法的一部分,大家的意见还不一致,但是在 DSS 的设计和构建中使用反复型或者进化型原型开发,似乎是一种合适有效的方法。这种方法可以命中 DSS 的用户需求这个移动的目标。虽然不存在原型开发方法的通用规定,但是对于开发一个要进行反复修改的原型,许多研究者(Courbon,Grajew 和 Tolovi,1980;Henderson,Ingraham,1982)都为任务的顺序提供了有效的描述。

原型开发方法的前几个阶段类似于典型的 SDLC 方法。主要差别最初发生在开始收集需求和原型的第一轮开发上。一旦有了第一个原型,反复的过程就开始了,不断对原型做小的修改,直到存在一个能够准确地反映 DSS 的用户需求和愿望的稳定的系统。这个过程需要系统分析人员和用户之间进行非常充分的交流,合作关系比采用基于 SDLC 的设计方法要紧密得多。整个项目过程中,责任的反复转移标志着这种合作关系。当对某一个原型进行修改时,系统分析人员背负着巨大的责任,而用户则没有直接关联。当修改

完成,等待检验时,DSS 用户则背负着决定系统是好是坏、需要添加或删除什么功能的责任。和 SDLC 方法中用户被动地参与相反,DSS 用户在 DSS 开发过程的评价活动中是一个内在的、主动的部分。此外,双方还要共同识别并确定在反复修改的过程中 DSS 需要的输入数据。一旦完成,原型开发方法就为极少部分的决策者或问题解决者提供了高度专业化的应用系统。它只对非常狭窄的范围内的问题有用。

丢弃型原型开发和反复型原型开发

原型法的开发过程衍生出两种基本的原型开发方法。丢弃型原型 (throwaway prototype) 开发只是为了做出示例,在它成为示范工具后马上抛弃。和生命周期法相比,它的优点在于用户能很快得到一个具有基本功能的应用系统,同时节约相当数量的开发费用。在减少了对于一个能运行的版本来说必要的项目时间后,应用系统的价值可以在项目早期就展现给高层经理和财力支持者。这个便利之处可以帮助保存必要的长期支持途径,这对 DSS 项目的成功完成非常关键。更重要的是,开发费用和用 SDLC 方法设计相比显著减少了,至少在项目初期是这样。这是因为在原型开发初期,就能方便地确定应用系统的最初成功。如果,不管什么原因,开发人员不能通过当前可用的技术和开发工具设计并实现 DSS,将很容易重新考虑这个计划,并在设计阶段的初期就进行修改。

丢弃型系统的替换方案是采用反复型原型 (iterative prototype) 开发方法 (也被称作进化型原型开发)。反复型原型不是在展示后马上被抛弃,而将被重新开发,并且不断地改进,直到它完全满足 DSS 用户的需求,然后将它和组织的现存系统进行必要的集成。换句话说,原型"进化"成最终的系统产品。和丢弃型原型开发不同,反复型原型的设计非常复杂和艰难。但是两种类型都具有相同的基本开发过程。最初,用户和系统分析人员通常不清楚,甚至一点不知道系统的需求,但最终了解了,文档化了,而且体现在能够运行的 DSS 中。

原型开发方法的优点和局限

正如前面指出的,采用原型开发方法设计和构建 DSS 似乎比更加传统的生命周期方法更有效。采用这种设计方法可以清楚地识别出许多优点。和 SDLC 方法相比,开发时间的显著减少和开发费用的降低,使得原型开发方法在当前这个快速发展的管理环境中成为具有吸引力的方案。此外,用户能做出关于系统功能的及时响应和反馈,这个特点通常使得 DSS 能够获得高层管理人员更多的支持,因为他们很容易发现这个项目带给公司的利益。最后,原型开发过程反复的特点至少从理论上促进了用户对系统和它的所有功能的理解。

原型开发方法的优势也导致了它的局限。虽然公认 SDLC 方法比纯粹的原型开发要慢而且要更加细致,但是更加细致也使得对整个开发文档的细节能够给予更多的注意,对系统的优势和相应的花费也能有更深刻的理解。此外,原型开发方法的过程使得系统维护可能比基于 SDLC 方法开发的相应系统要困难。因为采用原型开发方法时,用户和系统分析人员都关注于 DSS 的功能和使用的难易性,而不是像维护的难易性、详细的文档以及是否符合规范的设计标准之类的附属问题。这些缺陷并不是不能克服的。通过对准

确及时的文档保持高度警觉和仔细观察,就能够有效地克服,而且近年来还可以采用专门帮助 DSS 设计的开发工具。

13.2 决策支持系统开发人员

决策支持系统的开发人员和他们建造的系统一样,是多种多样的。从他们具有的能力和背景来看,既有经验丰富的专业系统分析人员,也有第一次开发的新手。即使所谓的受欢迎的专业人士,也是一个相当大的范围,既包括专业的 DSS 应用系统开发设计师,也包括那些虽然具有丰富的开发组织系统的经验,但没有任何设计开发 DSS 系统经验的人。不同于那些在计算机科学和商业信息系统领域都受过正规训练的专业 DSS 开发人员,没经验的开发人员通常是组织的管理决策者,他们从以前发生的问题或周期性发生的问题中觉察到需要基于计算机的支持。这些人基本上没有受过正规复杂的系统分析与开发的训练,但经历一次系统开发的过程可能会非常有益。没经验的开发人员和受过正规训练且经验丰富的分析人员相比,具有一个非常重要的优势:他们更清楚地知道需要 DSS 做什么,只要具有恰当的工具集,可以获得开发人员和用户是同一人的协和优势。

必需的能力

无论 DSS 的开发人员是否有经验,也不管他们在组织中扮演着什么角色,他们都需要具有一些关键的能力,才能在努力后成功。正如我们所看到的,关于这些能力的获取,专业的开发人员比新手有优势。例如上面讨论的,新手一般确切地知道对系统期望的功能,而专业人员则必须努力和 DSS 用户沟通才能获得这方面的了解。相反,和没有受过这方面的正规训练的新手相比,专业人员更有能力轻松地应付复杂的工具选择。我们将在以下部分简要地介绍并讨论开发人员应当具有的能力,它们对于设计一个成功的 DSS 是必需的。

理解问题领域的知识

对于任何组织的应用系统,尤其是决策支持系统的开发来说,最重要的一点就是要透彻地理解问题领域的知识。在某些情况下,必须达到这个领域的专家水平,而在另一些情况下,具有对问题领域的大概了解就足够了。然而 DSS 以问题为中心的特点,需要开发人员和用户都非常熟悉重要问题的具体特点。DSS 开发人员不仅需要具有开发基于计算机的应用系统的知识,也需要学习关于问题的细节知识。对于一个用于病人初期护理的外科中心的决策支持系统,一个具有医学方面的专业知识的 DSS 开发人员将非常有价值;但是对于一个消费品制造商开发的用于产品分配的 DSS,这个开发人员可能就不是那么有作用了。

理解具体的用户需求

设计人员不但需要了解问题的环境,他们还必须理解 DSS 用户的需求,包括规划的系统功能,以及用于用户和系统交互的特殊界面需求。设计人员要想具有这种能力,就要

使 DSS 用户尽早地参与系统的分析和设计。

掌握可行的开发技术

结合 DSS 开发人员和用户的能力与知识，可以决定 DSS 开发工具和平台的最终选择。现在存在大量的技术为 DSS 的开发提供基础。DSS 开发人员对这些技术越熟悉、知识越丰富，就越有可能选择最恰当的一种。例如，如果要建造一个提供“what-if”分析的 DSS，电子表格可能是最恰当的平台，然而对于一个帮助在复杂的操作流程中确定最正确的行动的 DSS，规范的管理平台则更适合。还有一些情况需要系统围绕着以案例为基础的推理机制开发。开发人员对复杂的技术越是熟悉，正确地选择开发工具的可能性就越大，成功地开发 DSS 的可能性也就越大。

获得恰当的知识

即使是经验最丰富、技术最熟练的 DSS 开发人员，他也必须能够恰当地识别、确定和表达用于在特定的问题环境下做出决策的相关知识，DSS 正是为这些特定的问题而开发的。在许多例子中，DSS 分析人员和用户都需要参与关于识别和确定相关知识的活动，但是精确表达这些知识的重任主要还是由开发者承担。如果 DSS 的用户想确认知识已经被准确地表达，只能通过仔细地检查 DSS 的每一个原型。这项必需的能力是和透彻理解问题领域的知识紧密关联的。一个开发人员如果对特定的问题领域不熟悉，就不可能了解那些以特定的程序或模型技术形式存在的关于 DSS 的知识。在这样的情况下，开发人员不是像“重新发明车轮”一样浪费时间和资源，就是最终设计出一个低效率的系统。

最终用户开发 DSS

DSS 开发新手包括的范围非常广泛，这些人积极地参与到被称为最终用户计算 (end-user computing) 的活动中。最终用户是指那些不是常规约定的组织 IS 部门或执行相关功能的部门的人员。最终用户计算的范围包括以上人员中的那些在日常业务和工作中使用电子邮件的人、定期使用字处理程序编写文档的人、以及设计和实现基于计算机的信息系统 (例如 DSS) 的开发人员。那些构建基于计算机的系统的个人是 DSS 的最终用户开发人员，他们或者是为了解决组织的问题，或者是为了帮助研究决策活动。

那些在组织中扮演各种各样的角色，具有各种层次的计算机水平的人构成了最终用户开发人员的群体。最终用户可以在组织的各个层次上找到，既包括工厂车间，也包括高级管理的位置；最终用户还可以在各个不同的功能领域找到，从财务部门到市场部门，再到物资计划部门，都有他们的踪影。此外，他们开发和使用 DSS 的经验和能力水平也参差不齐，既有第一次使用的用户，也有部门的计算机主管。最终用户开发的应用种类和选择的开发方法的规范水平也是各种各样的。

由于典型的最终用户开发人员缺乏系统开发方法的正规训练，所以大多数最终用户开发的应用系统显得相当不规范。虽然这种不规范不会在小型独立的 DSS 开发项目中造成一系列的问题，但是当应用于大型复杂的 DSS 开发时，它会导致一系列的不良后果。没有规范的设计方法的严格规定，复杂的 DSS 开发项目会深受缺乏指导、系统目标分散，

以及系统需求和说明不精确等缺陷的困扰。

最终用户开发 DSS 的优势

大大减少了应用系统的交付时间是最终用户开发 DSS 最明显的优势。假设最终用户具有成功完成项目所必需的能力,那么就可以避免因信息系统部门参与开发所造成的时间延迟。DSS 项目可以在一夜之间全面展开,视设计的复杂程度在从几个小时到几个星期不等的时间内完成。正如所知道的,DSS 的价值仅仅在于它支持一个成功的决策过程的能力。DSS 越快地被投入使用,组织也就越快地认识到它所带来的价值。

除去一个浪费时间的步骤是最终用户开发的另一个显著优势,这个步骤就是收集并规范化最终用户对系统的详细说明。用户通常不能通过言语将他们的需求表达到和他们自己所想的一样清楚。在最终用户开发中,分析人员和用户是同一个人。这一点使得所想的需求能快速地转化成工作原型。最终用户可能不擅长解释他们的需要,但是当他们想到时,他们自己肯定清楚。

项目的设计阶段结束以后,遇到的实现中的问题减少了,这是最终用户开发的第三个优势。为了理解复杂的系统,用户需要和分析人员进行沟通,这期间常常会发生问题,而相同的问题在实现过程中还会出现。在项目的需求收集阶段,识别并规范化用户的需求对分析人员来说是一个困难而且耗时间的任务,在项目实现阶段,他们还需要教最终用户如何操作新系统。在最终用户开发的情况下,用户在一开始就迅速地熟悉了 DSS 的操作,从开发到使用只需要花最少的时间和精力。

我们已经认定的前三个优势带来的结果,是最终用户开发的最后一个优势,也就是降低了开发费用。因为减少了开发的时间和占用的组织资源,最终用户开发为组织节约了巨大的花费,这增加了组织的价值。此外,更快的展示速度使得组织能够迅速认识到运行的 DSS 的优势。所有这一切使得和传统的生命周期方法相比,最终用户开发的概念成为一种具有吸引力的选择。

最终用户开发的风险

最终用户开发的内在风险和它的优势同时存在,所以要想成功地实现 DSS 项目,必须细心地管理。最值得担忧的是最终的产品缺乏必要的质量,不能成为可靠、有效的决策支持工具。最终用户经常忽略或回避传统的控制机制和测试程序,而这些却是正规的、结构化的开发方法的固有内容。这种不严格的开发过程经常造成各种严重缺陷,有时还会酿成大祸。电子表格中可以插入的一些像公式一样简单的东西,如果不能精确地计算总数,在测试阶段又没有被察觉,那么在算一个很大的数时,如果用户完全依靠 DSS 提供的结果做决策,可能会造成几百万美元的损失。另一些情况下,在 DSS 的开发过程中没有被检查出来的功能缺陷,却有可能在关键的决策活动中暴露出来。

最终用户开发的另一个常见风险是缺乏质量可靠的说明文档。专业的开发人员经常抱怨写说明文档的乏味,但是他们认识到了这个步骤的重要性,以及精确详尽的说明文档的必要性。最终用户也许知道精确详尽的说明文档非常重要,但是他们很少做出实际行动。最终用户只关注应用系统的实现,但却时常事后才想到应该写说明文档 (如果能想

起)。当系统的开发人员和用户是同一个人时,没有详细的说明文档不是大问题。然而有时候,一个最初由最终用户开发的应用系统,可能发展成为另一个应用系统的组成部分,该应用系统也是为某个功能领域的决策过程提供支持。一旦开发者离开了这个组织,他也带走了他所实现的系统的相关知识。几年后,如果必须对还可靠的 DSS 做一些修改,就会突然发现组织中没有人能够解释清楚源代码,因此很难做出必要的修改。这时,组织就面临着要不就不做希望的改善和修正,要不就从零开始,重新开发一个具有组织需要功能的新的 DSS (而且希望有好的说明文档)。

因为最终用户不熟悉如何恰当地控制应用系统的安全,导致系统缺乏必需的安全措施,这是第三个值得注意的方面。在大部分情况下,这一点只是导致组织数据的完整性受到一点威胁,因为典型的最终用户开发的 DSS 仅仅是一个由组织的一小部分成员使用的应用系统。然而在某些情况下,最终用户开发的应用系统需要从组织的一个甚至多个数据仓库中取得数据。这时,最终用户开发的应用系统不仅成为通往组织的私有数据的直接渠道,而且有些时候还通往其他运行中的应用系统。如果最终用户开发的应用系统成为一个正规地设计和开发的系统的一部分,那么一旦没有达到相同的安全控制水平,最终用户就留下了一道潜在的、没有上锁的门。外人通过这道门可以获得组织的私有财产。因此,一定要保护好组织的数据和信息。

如果有效地控制我们讨论过的这些 (以及其他一些没有被讨论的) 风险,就可以实现所期望的最终用户开发带来的优势。一个没有多少经验的开发人员,即使他缺乏正规的训练,仍然可以透彻地理解系统设计的重要因素,因此可以具有专业的开发态度,这增加了他成功地开发并实现 DSS 的可能性。

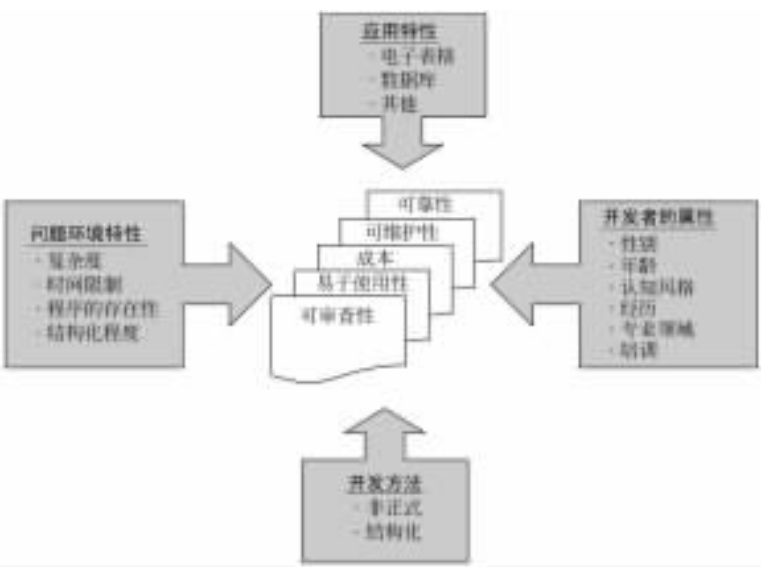


图 13-4　影响最终用户开发应用系统的风险和输出性能的因素
(根据 Janvrin 和 Morrison 1996 年的工作改编,348)

学术界和应用领域都在研究如何控制最终用户开发的风险。建立了开发框架后,恰

当地控制应用系统的安全（Walls 和 Turban，1992）、通过技术减少总体风险（Alavi 和 Weiss，1986）以及在电子表格中准确地找出并减少错误（Galletta 等，1996）成为关注的焦点。图 13-4 给出了一个模型，它包含所有影响最终用户开发应用系统的风险，可能对组织制定指导最终用户开发活动的准则有一定帮助。

最近，一些组织因为开发内部的教育系统，涉及到最终用户开发的风险。信息中心（IC）的概念也是在这时被提出的，它专门为最终用户设计和开发基于计算机的应用系统提供支持。IC 部门通常和组织的 MIS 部门一起帮助最终用户选择项目和恰当的开发工具。此外，IC 部门还可以为最终用户开发的项目提供技术支持。因此，组织信息中心的存在是支持最终用户计算和应用系统开发的标志。

13.3 决策支持系统的开发工具

随着越来越多的现代化组织实现 DSS，出现了很多新的、功能强大的 DSS 开发工具。一些专门的决策支持技术工具，如电子表格管理工具和文本管理工具，在商用计算机的桌面上已经十分常见。开发工具的技术也非常多样化，具有电子表格、文本和图形技术中的两种甚至更多种设计能力的开发工具越来越成熟。最终出现了高复杂度的 DSS 开发工具，如 DSS 生成器，促进了同样复杂的 DSS 应用系统的发展。我们将在这一节研究各种各样的工具，以及它们的分类方法。

开发工具分类

关于 DSS 的设计和开发工具，有各种各样的分类方案和框架。有一些方法按开发工具采用的技术分类，另一些则考虑在 DSS 的开发过程中工具所扮演的特殊角色，还有一些关注于开发工具生成的接口类型。如果对各种可行的 DSS 开发工具有透彻的了解，经验丰富的专家和最终用户开发人员都有更大的把握为当前任务选择最恰当的开发平台。

Sprague 和 Carlson 在 1982 年提出了一种最常见易懂的 DSS 开发工具的归纳分类方法。图 13-5 阐明了这种层次分类方法。

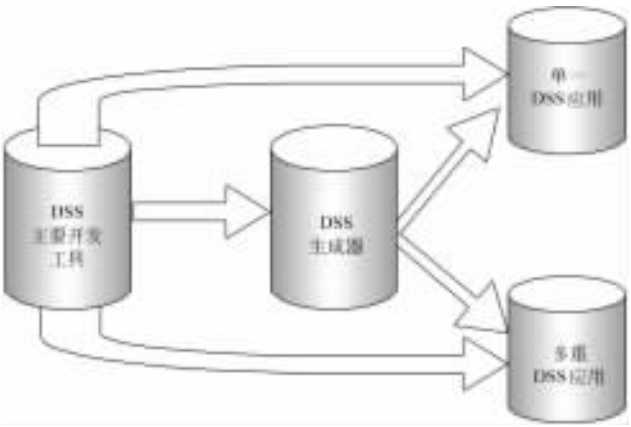


图 13-5 DSS 开发工具的分类

这种方法按照开发工具所采用技术的三个层次分类：(1) DSS 的基本开发工具，(2) DSS 生成器，(3) 专门的 DSS 应用系统。

DSS 的基本开发工具

就像它的名字所指出的，DSS 的基本开发工具采用 DSS 开发中最底层的技术。在这一种类中，我们可以看到编程语言、代码和文本编辑器、图形开发程序和数据库查询机制，它既可以被 DSS 生成器采用，也可以被专门的 DSS 应用系统采用。限于本书篇幅，我们不能更详细地讨论这些工具，但是有很多著作讲述了 DSS 应用系统的编程方法（比如 Lotfi 和 Pegels，1996；Sprague 和 Carlson，1982）。

DSS 生成器

根据 Sprague 和 Carlson 的分类方法，DSS 生成器是“一套帮助迅速方便地建立专用 DSS 的硬件和软件”(1982，10)。可以找到大量的各种类型的 DSS 生成器，很多 DSS 生成器都已经商业化了，从在大部分商用计算机的桌面上都可以找到的电子表格程序包（微软的 Excel），到更加复杂的应用系统如 MicroStrategy 的 DSS Agent。如果不考虑它们的复杂度，所有的 DSS 生成器都具有集成和多样化的功能，包括建立决策模型、设计报告、生成显示图形和基本的数据管理能力。

和 DSS 的基本开发工具相比，DSS 生成器最主要的优点就是方便。如果使用 DSS 生成器，工具间的集成、多样化工具的获取、数据的输入输出、工具间非标准化的命令格式等问题就都消除了。DSS 生成器的各种功能紧密结合，使得开发人员可以将注意力集中在设计过程上，而不是在得到两种能够一起传输数据的工具上。

DSS 生成器的名字并不非常准确，这一点需要注意。在大部分情况下，DSS 生成器并不能“生成”任何东西，而仅仅是一套工具和功能的集合，辅助设计和实现 DSS。实际上，开发人员才是真正的“DSS 生成器”。

专门的 DSS 应用系统

DSS 技术的第三层是专门的 DSS 应用系统。在某个特定的知识领域内，在很多情况下会出现决策者面临的非结构化问题。在这些情况下，可以使用一个商业 DSS 应用系统，而不必自行开发一个。在医药领域使用的 DSS 是这种类型问题的一个例子。虽然一个医药专家面临的每一种特定情况都是独一无二的，但是 DSS 丰富的医药知识基础可以应付大部分情况。因此为了获取必要的决策支持，采用一个商业上可行的 DSS 辅助诊断和治疗各种疾病，和采用 DSS 生成器自行开发一个 DSS 相比，是一种更加方便、经济且可靠的方法。

再次参考图 13-5，我们可以看到三种类型之间的关系并不是从本质上严格分等级的。DSS 工具可以用来构建一个 DSS 生成器，DSS 生成器又可以用来开发单个的 DSS，也可能用来开发许多不同的 DSS 应用系统。相同的工具也可以直接用来开发一个专门的 DSS 应用系统。同样，组织也可以通过一个商业化的 DSS 生成器开发各种专门的 DSS 应用系统。决策者特定的需求和问题的性质决定了选择哪一种顺序。

选择开发工具的标准

关于是选择一套基本的 DSS 开发工具还是获取一个现有的 DSS 生成器的讨论，本身就是一个复杂的决策，而且通常包含了大量的研究和分析。所以在做出是从零开始开发还是购买一个商业化的 DSS 生成器的决策时，需要考虑许多标准。表 13-3 是一个包含了各种标准的简略列表。这些标准对于选择一个 DSS 生成器十分重要。

表 13-3 DSS 生成器的选择标准

- 数据管理功能
- 模型管理功能
- 用户界面的性能
- 兼容性和通用性
- 可行的硬件平台
- 费用
- 销售商提供支持的质量和可得性

每一个种类都需要在决定购买 DSS 生成器之前回答许多问题。数据管理系统是否允许生成个人数据库？用户界面生成器是否为大量不同的输入输出设备提供了支持？什么是正在进行的维护？一个特定的销售商升级它的应用系统需要多少费用？这个应用系统需要哪些技术支持？记住，决策支持的领域复杂多变，而且特殊的需求通常是不可预测的。DSS 生成器必须为用户取得经济利益和保证长期的有效性提供最大程度的开发支持。

13.4 小 结

设计和开发 DSS 是一项艰巨的任务，但如果完成得好，可以为组织带来巨大的利益。选择开发工具和开发方法，以及收集用户需求，都是成功开发 DSS 的前期工作，而设计和构建 DSS 还仅仅是战役的一半。即使系统完成了，如果还没有挖掘出并利用它的所有功能，就还需要进一步地完全实现，并将它们集成到组织中。我们将在下一章关注 DSS 的实施问题。

复习题

1. DSS 的分析与设计有哪两种基本策略？
2. 解释专用 DSS 生成器。
3. 列出并简单介绍系统开发生命周期 (SDLC) 方法的步骤。
4. 描述功能分类分析方法。它最显著的优点是什么？
5. 在 DSS 开发过程的系统分析阶段中，需要识别哪几种需求？
6. 描述 DSS 开发过程中，系统设计与构建阶段最主要的任务。
7. 什么是传统的 SDLC 方法的优点和缺陷？

8. 什么是原型开发方法？
9. 列出并简单介绍原型开发的两种方法，比较它们的优点和缺陷。
10. 从用户角色和用户参与程度方面比较 SDLC 和 DDP。
11. 列出并简单介绍原型开发方法的优点和缺陷。
12. DSS 开发人员需要掌握哪些必需的能力？
13. 解释最终用户计算的概念。
14. 识别最终用户开发 DSS 的优势和风险。
15. 为 DSS 开发工具分类。
16. 你如何评价 DSS 开发工具的选择？

讨论题

1. 使用 ROMC 分析方法来分析一个 DSS 开发项目。
2. 为什么需要一个不同于传统的 SDLC 的 DSS 开发方法？
3. 讨论量化 DSS 目标的重要性。
4. 比较 DDP 和 SDLC，讨论这两种系统开发方法的不同之处。
5. 讨论组织中 DSS 开发者的差异性，以及这种差异性的影响。
6. 观察你所熟悉的一个组织中的决策制定过程。假设你正在为这个过程开发一个决策支持系统，列出你在选择开发工具时需要考虑的因素。

第 14 章

决策支持系统的实施与集成

学习目标

- 理解实施的本质是引入变化
- 学习变化的几种理论模型
- 根据项目的发起者,确定实施的六种模式
- 理解系统评估的体制:软件的综合质量,个人对于成功在技术上、组织上的衡量
- 学习怎样度量用户对于 DSS 的满意程度
- 理解一般 DSS 成功衡量体制中的四种衡量方法:系统业绩、任务业绩、商业机会、进化方面
- 理解 DSS 实施项目中的风险因素
- 学习怎样阐述可能的实施策略,以对付已经确定的风险因素
- 理解集成的重要性
- 学习与全面 DSS 集成相关的概念
- 认识那些会抵制新系统带来的变化的因素

小案例

3Com ASSIST 用户服务站点

3Com 公司遭到竞争对手的妒忌。公司自 1979 年成立以来，到 1996 年销售额已经急剧增长到 23 亿美元。作为网络数据行业中最大、也是发展最快的几家公司之一，3Com 为遍布全球的 32 个国家超过 3200 万的用户提供服务。

公司面临的最大挑战是，怎样为不断扩大的用户群提供世界水平的任何时间、任何地点的全球服务，而这正是公司的特色。短短几年间，公司的用户服务人员数量增加了两倍还多，这使得公司断定，增加用户服务人员数量并不是一个可行的长远用户服务策略。

“我们不能够再以现在这样的速度雇人了，否则，用户服务组织就会变得比整个公司还大。”Bob Noakes，电子商务营销经理这样说，“为了保持我们的产品支持水平，我们需要给予用户同样令人满意的其他途径去解决问题，而不需要用户打电话。”

加快问题解决速度

为了创造更多的产品支持选择，3Com 公司决定加强万维网（World Wide Web）所提供的用户服务。网络遍及全球，而且全天候的特性使得它成为提高用户服务的独一无二的工具。3Com 是 Internet 上的关键“人物”，可以用它的 Internet 访问解决方案将数以百万计的人联系起来。3Com 相信，它还有很多可做的事来加强网络服务。公司最初的网络技术支持没有信息和用户，只是依靠研究机构自己发现解决方法。

为了使站点智能化，3Com 采用了 Inference 公司的 CBR 技术。通过将现有的支持信息与 CBR 结合，3Com 创造了一个交互式问题解决工具，并且有热键给予用户按部就班的指导，以解决问题。任何人，只要有网络浏览器，不管其使用什么样的操作平台，都可以获得服务。

最小化开发和最大化收益

3Com ASSIST 使用 Inference 公司的 CasePointAE 网络服务器（Windows NT 版本），CBR Express，CasePoint，CBR Tester，以及 CBR 基于 Windows 的搜索引擎。Inference 的 CBR 软件使用的方法与医生的医学诊断相类似。一系列阐述同样问题的案例细节以及公认的解决方法都输入到计算机系统中。当访客询问时，软件引导用户，通过一系列的问题及其答案来确定解决方案。

将知识引入工作

3Com 以其网卡和网络接口设备出众的质量和可靠性而闻名。EtherLink III 适配器是 3Com 产量最大的产品，占公司交易总额的 40％左右。

公司已经销售了超过 3200 万的 EtherLink III 适配器，尽管只有其中的一小部分需要服务，绝对数字仍然很大，而且还在增加。因为此产品的用户服务电话数量最多，Noakes 预计把它纳入 3Com ASSIST，这将对公司的停止电话策略做出实质上的贡献。

3Com Impact 外置数字调制解调器是一个复杂程度不高的新产品，初期每批发货都有很多的服务电话。“用户对于调制解调器的 ISDN 技术并不太熟悉，我们也考虑到了没有经验的用户。”Noakes 说，“3Com ASSIST 是减少每批发货的电话数量的理想途径。”

快速解决

“通过将解决问题的方法直接送到客户的手上，我们希望提倡停止电话策略。”Noakes 说，“同样重要的是，我们希望将我们从用户的质疑中得到的信息变成自助。”

3Com 希望一半人通过使用 3Com ASSIST 获得产品支持，以解决他们的网络问题。用户可以不用等待，自己解决问题。

本年在开通了两条新生产线之后，3Com 计划扩大 1998 年的 3Com ASSIST，以包括更多的产品。“目的之一是在用户打电话时更好地收集信息，”Noakes 说。随着 3Com ASSIST 的发展，公司致力于将 CBR 与其他的方法集成以获取信息。“既然 CBR 是适应 OLE 的，我们希望将多媒体、其他数据库、在线文件管理和营销信息与系统进行集成。”

要了解 3Com 公司怎样将 Inference 的 CBR 技术集成到 3Com ASSIST 的，请访问 http://infodeli.3com.com/wcp/profile1.htm。

注：Inference，CasePoint，CasePoint Search Engine，CasePoint WebServer，CBR Express、CBR Express Tester和 CBR Generator 是 Inference 公司的商标。

14.1 决策支持系统的实施

孤注一掷

现在我们已经明白 DSS 设计和构造对特殊方法的需求了。同样重要的是实施 DSS 和将它集成到组织中的复杂流程。这或许是整个 DSS 设计流程中最重要的步骤，因为没有成功的实施，建立的 DSS 将没有用处，或者会被忽略掉。打个比方，实施流程是“孤注一掷”的。

19 世纪作家 Washington Irving 曾经说过：

> 变化中包含着解脱，即使它将变得更加糟糕。就像我在旅途中的驿站马车中发现的，换一个位置后另一个人在新位置上也会受伤，但却是一种安慰。”

事实上，实施就是引入变化。虽然它常常作为“事件”被提及（像“当我们实施系统时……”，“实施之后……”），但是，将实施作为集中于把 DSS 成功引入组织环境中的一整套活动，是更为准确的。这样说来，传统的 SDLC 模型中实施阶段所占位置就有些令人误解了。在这章的结尾，你将看到实施确实是一个戏剧性的活动，它实际上贯穿于整个开发阶段，并且在我们称之为项目实施阶段的过程中完善自己。DSS 开发流程中的其他步骤

都以一套活动的开始作为标志，与之相反，实施阶段以活动的结束作为开始标志。此外，以 Irving 乘坐驿站马车时的观察，成功的 DSS 实施可能需要组织内的变动更加频繁，这时，对某些人或许是痛苦的，对其他人却是解脱。

将变化引入组织

丰富而广博的文化中总是存在着对变化这一主题的讨论——变化的特征，如何实现它的步骤以及衡量其是否成功的方法。在这种文化中，更为普遍的是识别变化通过怎样的流程引入，在很多情况下，变化被接受和取得成功比变化本身的性质更为重要。尽管很多变化的模型已经复杂化了，但有两个描述变化流程的非常简练的方法可以在《Lewin-Schein Theory of Change》和《Kolb-Frohman Change Model》中找到。我们将简要探讨一下每个方法的构成，以更好地明白变化的概念。

Lewin-Schein 的变化理论

根据 Lewin-Schein 的变化理论（Schein，1956），变化的流程发生在三个基本的步骤中：

1. 解冻：开始意识到需要变化，并有一个能接受变化的环境。
2. 运动：压力大小和方向发生变化，这个变化通过开发新的方法以及学习新的看法与行为，决定了对于变化的初始需要。
3. 再冻结：希望的变化已经发生，可维持的、稳定的新平衡建立。

可以看到，Lewin-Schein 变化理论同时描述了结构性和可变性 结构性表现在对三个暗示了事件具体顺序的步骤的描述，而这些事件是成功实施的必要条件 ;可变性表现在每个步骤中的活动与变化的特性、变化的数量以及变化发生的环境有着巨大的联系。Zand 和 Sorensen (1975) 将这个变化理论应用于一个关于大样本项目的研究中，而这个大样本与各种各样的管理科学方法在组织中的实施有关。研究者发现，成功处理流程中的每个步骤出现的问题，与整个给定项目的成功有很大的关系。表 14-1 中列出了 Zand 和 Sorensen 在变化模型的每个阶段发现的关键问题。

Kolb-Frohman 模型

与 Lewin-Schein 模型相比，Kolb 和 Frohman (1970) 提供的模型更为精确。这个变化模型包含了 7 个步骤，被认为比 Lewin-Schein 模型更加标准化。Kolb-Frohman 模型的本质是假定成功的实施取决于用户与分析员之间明确的行为模式。Ginzberg (1975，1976) 经验主义的研究将这个变化模型应用于开发一系列假定，这些假定认为，7 个阶段中所解决问题的成功程度与整个实施流程的成功有着很大的关系。它的研究结果认为，符合标准化 Kolb-Frohman 流程的项目比那些明显背离模型的项目显示出了更大的成功。表 14-2 列出了模型的 7 个阶段并且提供了每个阶段活动的简要说明。

回顾这两个已经被认可的变化理论是为了当变化发生在典型的 DSS 实施阶段时，进一步理解其流程。模型并不是用来约定俗成的，而是用来描述一个框架的，这个框架可以指导实施流程。

表 14-1 Lewin-Schein 的变化模型中各个阶段中有利和不利的压力

变化的阶段	有利的压力	不利的压力
解冻	• 高层管理者认为这个问题对公司是重要的 • 高层管理者参与 • 部门经理认识到变化的必要 • 高层与部门经理在讨论中都很坦率 • 部门经理愿意修改他们初始的一些假定	• 部门经理不能清楚地表述出他们的问题 • 高层管理者认为整个问题太庞大 • 部门经理感到项目对自己有威胁 • 一些部门经理憎恶这个项目 • 部门经理在分析中缺少自信 • 部门经理认为自己有能力单独操作这个项目
运动	• 部门经理与分析员共同收集数据 • 所有相关数据都可以拿到,并可能加以利用 • 设计出新的替代方案 • 部门经理审查并评估所有的替代方案 • 将所有的方案报告高层管理者 • 高层管理者参与解决方案的开发 • 所有的解决问题的建议依次加以改进	• 分析员在培训部门经理时常常没有效果 • 部门经理不参加新解决方案的开发 • 部门经理常常不能完全明白分析员提出的解决方案 • 分析员认为项目结束过快
再冻结	• 部门经理尝试解决方案 • 利用频率显示了新方案的优势 • 在初期采纳使用时,分析员发起积极的反馈 • 最后的解决方案在取得初始成功之后被广泛接受 • 部门经理对新解决方案表示满意	• 在解决方案开始使用之后,分析员并不努力支持新的管理行为 • 在解决方案开始使用之后,分析员并不努力重建稳定性 • 结果难以量化或衡量

资料来源:Theory of Change & Effective Use of Management Science,作者:Zand,Dale E.,Sorenson 和 Richard E.,Administrative Science Querterly,Vol. 20,#4。经许可引用。

表 14-2 Kolb-Frohman 的系统发展变化标准模型

调查	用户和设计者调查对方的需求和能力,以判断是否匹配。选择组织对项目的适当着手点。
启动	用户和设计者开始对目标的最初陈述。项目责任确定。用户和设计者建立相互信任的关系,并以"合约"指导项目。
诊断	用户和设计者收集数据以推敲和明确问题的定义以及目标,为寻求解决方案做准备。用户和设计者评定可用资源(包括约束),以判断进一步的努力是否可行。
计划	用户和设计者定义详尽操作目标并检查达到这些目标的替代方案。检查已经提出的解决方案对组织所有部分的影响。用户和设计者考虑解决方案对组织的影响,制定行动计划。
行动	用户和设计者将"最好"的选择付诸实施。作为有效使用系统的必要条件,培训在组织中所有涉及到的部分展开。
评价	用户和设计者评估目标(在诊断和计划阶段有详细描述)达到情况。用户和设计者决定是否将工作推进一步(发展)或者停止现行的工作(终止)。
终止	用户和设计者确保新系统的"所有权"和有效控制掌握在使用和维护系统的人手中,同时确保必要的新行为模式成为用户常规行为中的稳定部分。

资料来源:An Organizational Development Approach to Consulting,作者:Kolb 和 Frohman,Sloan Managment Review,

实施的模式

Alter (1980) 认为 DSS 开发背后的推动力可以归结为以下几条：(1) 用户促进 (user stimulus)；(2) 管理促进 (management stimulus)；(3) 创始人促进 (entrepreneurial stimulus)。

前两条的名称暗示，DSS 项目的促进因素可以由用户感知的需求和管理层感知的需求产生。第三条假定一个或者几个来自组织内部或者外部的人，在开发 DSS 时，目的是努力将组织推销出去。在建立这 3 种发起人的基础上，Alter 确定了 6 个 DSS 实施的普遍模式，根据用户发起程度，委托开发使用程度以及用户设计和开发参与程度等不同而有所不同。每个方案的命名暗示了其潜在的性质。图 14-1 描述了 Alter 的 6 个普遍实施模式。

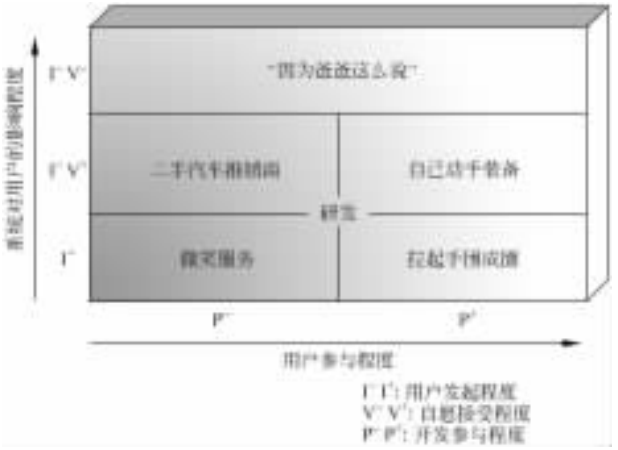


图 14-1　6 种普遍的实施模式

资料来源：Alter，S. L. 1980，Decision Support Systems：Current Practices and Continuing Challenges. Reading，MA：Addision-Wesley.

"拉起手围成圈"

依照对实施的论述和实践，这是最理想和最常用到的场景。假定所有因素不变，用户在 DSS 开发项目中较高水平的发起和参与程度，是与较高的成功几率相联系的。在这个场景中，假定预先没有定义明显的解决方案，用户随时预备为问题定义和问题解决而努力协作。

"微笑服务"

这里，用户只是简单关注于购买产品而不是购买服务。产品的形式和内容相对较为清晰和确定。这个场景的主要目标是使 DSS 可以根据时间和提出的预算约束的不同而采取不同的对策。与"拉起手围成圈"相似，用户发起程度非常高。但是，与前一个场景形成对比的是，用户对开发的参与程度相对较低。

“自己动手装备”

如图 14-1 所示,在这个实施场景中,用户的发起程度比较低,但是自愿采用程度相对较高。这个场景努力将较低的发起水平,较低的参与情况,转化成有利的情况。既然这样,用户就必须相信 DSS 对于公司的成功是必要的,必须鼓励参与它的开发。

“二手汽车推销员”

这个普遍的场景以顾问或者推销员企图将对 DSS 的需求强加给用户为特征。这种情况要为组织获取明显利益是不常见的,因为许多成功的创新常常由它们应用的场合之外开始。但是,正是这个场景,因为是外部发起项目,所以实施有相当大的风险。

“因为爸爸这样说”

第五个实施情况在高层管理者委托开发和使用 DSS 的公司中也是较为普遍的。尽管这种情况非常普遍,这个场景被认为是将决策支持技术引入组织中的最无效的方法。没有足够的用户发起和支持,DSS 将永远不会圆满地实现制度化,以使人们充分认识它的价值。

“研发”

Alter 将第六个也是最后一个场景放在其他所有场景组成的长方形中间,并且认为通过任何其他五个场景,内部发起研发努力可以整合到日常的实施中。

有些活动被认为是成功实施项目的积极步骤,而六种类型的 DSS 实施情况定义使我们可以洞察这些活动的类型。在下一节中,我们将在这一观察的基础上,关注那些可以应用到特殊场合和发展的需要的 DSS 实施战略框架的开发。

14.2 系统评估

看见成功的时候确认它

实施流程中暗含着成功的目标。如果实施成功,则表明所做的努力是值得的,同时也是充分的。如果是另一方面,即认为实施失败,则极有可能是竹篮打水一场空,而什么地方出错了,就成为参与者关注的焦点。虽然成功的目标是不明显的,衡量成功的方法更不清楚。只是精确地弄清是什么包含成功的 DSS 设计和实施?当设计者和用户看见成功的时候怎样确认它?这一节中,我们讨论用来衡量实施成功的方法,以及几种判断成功的指标,这些指标是从对各种各样的实施的研究中得到的。

系统进化的框架

成功多少是个难以定义的概念。一个人衡量成功的标准可能刚刚达到另一个人期望的可以接受的水平。尽管不存在衡量 DSS 实施是否成功的一套普遍标准,但如果开发一套局部标准是可能的话,就必须这样做,否则定义项目的成功程度就很困难了。已经提出了许多衡量成功的方法,每个都带着自己独特的指标。

软件综合质量

Boehm 等 (1978) 提出了衡量成功的框架,关注的是软件的特征和质量。表 14-3 列出了框架中的元素,并且提供了每个特定方法的简要说明。

表 14-3　软件质量的特征

可移植性	相对平台和设备的独立程度:软件可以在各种各样的硬件配置中执行
可靠性	完备度:软件的所有部件齐全,每个都充分发展
	精确度:软件输出要足够精确使预期的用户满意
	一致度:软件注释、术语、符号在内部保持一致,内容可以追溯到需求
效率	设备有效度:软件实现目标,而不浪费资源
	可达度:软件设计促进有选择的使用部件
运行工程学	沟通度:软件易于读进标准的输入,并且提供不但形式和内容便于吸收,而且有用的输出
可测试性	结构化程度:软件拥有组织明确模式的独立部分
易懂性	自我描述程度:软件包含足够的信息,为用户定义目标、假定、约束、输入、输出、部件和状态
	简明度:阅读文档和代码,就可以了解软件和部件功能
易于修改	可扩充度:软件可以方便地容纳更多的数据存储和部件计算功能

资料来源:Characteristics of Software Quality,© 1978 版权所有,经 Barry Boehm 许可引用。

Dickson 和 Powers (1973) 以明确量化的形式提出了更加常用的一套指标:

- 项目实际运行时间与预计时间的比率
- 管理者对系统的态度
- 开发项目实际的成本与预算成本的比率
- 管理者的满意程度
- 项目对公司计算机运作的影响

仔细考虑由 Dickson 和 Powers 提出的指标后,可以看到,以他们看来,成功与效率而不是效能联系更为紧密 (回想第 3 章)。但是如果项目的目标已经确认,即使效率可能是衡量成功的重要因素,它也必须从属于效能。

个人对成功的衡量

评价 DSS 设计和实施的成功与否的另一个方法,是调查系统用户关于系统使用各个方面的态度。一个普遍方法是关注系统的预期用户群中实际使用系统的程度。调查诸如在给定期间内的查询数量或者在特定活动中系统的访问量,都可以作为衡量成功的方法。尽管这些方法最初有吸引力 (主要是由于这些数据易于获取),但它们没有考虑系统是否由组织强制使用或者用户是否还有可选择的其他决策支持机制这样的问题。如果使用是强制的,那么测量系统的访问量并不象征系统的成功。此外,如果对一个特定问题环境,不存在替代的决策支持机制,那么系统访问量可能表示"总比没有好",而不是真正的用户选择。为了避免直接量度系统使用带来的问题,但仍然尝试调查用户对给定系统的态度,用户满意度的概念由此产生。

将用户满意度作为成功衡量因素,其背后的逻辑是如果用户感到 DSS 在决策制定流程中是有效的,他们将认为 DSS 是不错的。系统显得越有效,用户对系统越满意。Ives, Olson 和 Baroudi (1983) 开发了简单的 40 条标准衡量用户满意度,这些标准综合了这个领域若干名研究者的工作。虽然这是一个普遍的度量方法,看起来在不同的项目中都是可靠的,它在 DSS 开发环境中的使用却是有问题的。DSS 用户倾向于根据近期 DSS 使用经历而不是系统的整个使用经历来回答调查问卷。典型的组织应用中,一个人的经历与其他人类似,并具有概括性;与之相反,与 DSS 的交互绝不是可以预见的。正因为如此,用户满意度可能对于特定 DSS 随时间有着很大变化。

Davis (1989) 提出了衡量用户满意度的另一个方法。它用 Likert 标度[①]来衡量对两种与成功有关的概念的态度:(1) 可感知的有效性;(2) 可感知的易用性。系统在其使用领域为活动带来的效用和价值在多大程度上被用户所感知,前一个概念关注用户对于这一点的态度。后面一个概念关注系统用户感知工具好用的程度。可以用一个简单的例子来描述这两个概念的区别。假设你正面对一项任务,将一棵大树从一块非常贫瘠的土地上移走。你可能会考虑的一个工具是推土机,另一个是链锯。推土机在可感知的"有效性"程度上面可能会胜过链锯,因为它可以用一个简单的动作将树推倒,并且不费吹灰之力将其挪走。但是因为各种原因,链锯可能会在"易用性"上胜过推土机。熟悉工具的使用方法、工具易于得到、环境约束以及其他类似的原因都会使人们感觉一个工具是易于使用的,尽管觉得它并不太有效。使用 Davis 的方法衡量 DSS 成功的关键一点是感觉到系统很有效和易于使用。表 14-4 列出两个概念中包含的几个要素,因为它们与 DSS 开发和实施有关。

表 14-4 Davis 成功衡量法的特征

可感知的有效性	可感知的易用性
• 提高用户生产率	• 用户明白怎样与系统交互
• 提高选择的效能	• 增加系统的整体弹性
• 提高选择的性能	• 使学习速率更快
• 使制定决策更轻松	• 增加系统的熟练应用的可能性

成功的技术标准

衡量 DSS 成功的第三个方法是判断系统是否达到了开始的设想。在这点上,可以利用系统技术功能性来判断系统是否成功。关注系统技术方面的一个优势是,在绝大多数方面,它们很容易被衡量和理解。举个例子,假设存在着用户对 DSS 的需求,那么系统在多大程度上达到了初始的目标,例如提供有用的建议,可能是可以接受的 DSS 成功的衡量方法。这个领域另一个成功衡量方法可能是符合用户需求信息的 DSS 特征的数量。还有可能的衡量方法是在解决特定问题时提供给用户的充分的建议和说明,或者更可能

① Likert 标度项目由简单的陈述后面跟随着常用的"五点"选择组成,"五点"中每一点的描述都指出这些选择之间的距离是相等的。例如 非常同意,同意,无所谓,不同意,非常不同意。

是，没有建议时向 DSS 提交的事例的数量。在专家系统是 DSS 的组成部分的情况下，当现实中的专家和 DSS 都面对着一个特定问题时，用户可能会比较两者提供的建议，来衡量 DSS 技术上的成功。依靠问题前后关系的本质，成功的技术衡量可能足以决定 DSS 实施项目的整体成功。

成功的组织衡量

最后一个成功衡量方法关注系统达到或超越组织需求与期望的程度。举个例子，一个组织使用特定 DSS 的回报，或者是降低成本，或者是提高收入，或者两者同时进行。这种直接回报或许是衡量 DSS 实施成功的一个有用标准。直接收益和直接费用的比率也经常用来决定组织成功问题。

Meador，Guyote 和 Keen (1984) 提出一套组织适合的环境，这样，就可以确定 DSS 是否成功。表 14-5 列出了这些 DSS 成功指标。

表 14-5　判断成功 DSS 的指标

- 改进决策者思考问题的方式
- 适合组织的计划方式
- 适合组织内制定决策的行政方式
- 产生可执行的替代方案和选择
- 相对于开发成本，系统被认为具有成本有效性和价值
- 期望将它用在一段适当的时间

衡量 DSS 的成功

关于什么构成成功以及什么是判断成功程度的标准的问题，有一点是清楚的：所有可以阐述成功的方法对于 DSS 实施的某些方面都是重要的。我们可以将衡量成功的不同方法组合成为一种普遍的 DSS 评估框架。Klein 和 Methlie (1995) 提出了评估 DSS 的结构，包括 4 种衡量方法，如表 14-6 所示：

表 14-6　DSS 成功评价体系

系统业绩	商业机会
• 效率和响应时间	• 开发、运作和维护成本
• 数据输入	• 增加收入和降低成本所带来的收益
• 输出格式	• 对更好的组织服务、竞争优势和培训的价值
• 硬件	进化方面
• 利用率	• 柔韧程度，变化能力
• 人机界面	• 开发工具的综合功能性
任务业绩	
• 决策的时间、可选方案、分析、质量以及参与者	
• 用户对信任、满意、有效、理解的认识	

资料来源：Klein 和 Methlie (1995)。

系统业绩

Klein 和 Methlie 提出的第一种方法关注计算机系统的质量。诸如系统响应时间、利用率、使用时间、可靠性以及系统支持质量等等指标，都予以确认和衡量。通过使用包括直接观察、事件日志、结构调查或感觉等不同方法，可以获得这些指标。前面已经提及，DSS 表现的综合质量与用户接受系统的程度有着很大的关系。

任务业绩

这种评价方法关注的是微观层次上的业绩，并且处理特定环境下的、与 DSS 的功能相关的一些问题。理想地，DSS 用来提供能够提高决策结果的支持。不幸的是，典型半结构化和非结构化决策的不确定度，使得在表现评估中，注重结果质量多少有些站不住脚。这是因为决策一旦制定，其结果常常不能由决策制定者来控制。我们认为好的决策并不一定产生好的结果。此外，好的结果可能偶然产生，甚至面对的是低质量决策。问题在于我们假定了决策制定质量和现实中结果的质量及期望之间存在相关性。同样的，通过关注我们可以控制的（决策质量）来代替我们不能控制的（结果质量），我们衡量任务业绩。根据决策流程花费时间、评估的替代方案的数量、信息寻找跨度，可以评估决策质量。另外，涉及信任和信心、满意度、理解程度等定性的指标也可以包括在这种方法之中。

商业机会

Klein 和 Methlie 第三种方法衡量 DSS 对某些组织因素的影响。DSS 开发和实施通常需要组织财务和时间资源的有效投入。投入的效果可以用来判断 DSS 在这个领域里是否成功。这种方法在本质上是定量的，并且包括关注收入增长、成本下降、培训效率提高、竞争优势提高、成本变化等衡量指标。

进化方面

在问题环境中，DSS 适应变化的能力是衡量业绩的第四种方法的关注焦点。在为用户决策行为和问题求解任务提供支持的同时，用户的决策制定行为、问题特征以及决策的环境都会发生变化。DSS 适应这些动态情况的能力常常是初始选择的开发工具的功能，也是用户在数据与模型上做必要改变的能力。除去这种衡量方法的定性本质，感觉到的弹性和适应性可以说明 DSS 的综合质量的一个方面。

DSS 实施项目中的风险

回忆第 5 章，风险是来源于概率估计中的概念。一个人常常并不知道在项目过程中负面事件发生的准确概率，也不能保证所有可能的消极结果都被确定和评估过。所以，坏消息就是，我们不能完全定义可能的风险的范围和它们发生的准确概率。但是，好消息是，所有潜在风险的确定和任何概率性质的估计，对于 DSS 实施质量和项目成功有着直接贡献。为了在任何环境中完成风险评估，只需要简单知道风险是可能的，并且试着去定义暗示失败的可能性增长的环境。

Alter (1980) 鉴别了 DSS 实施流程中的不确定度。表 14-7 列出了风险因子、风险因子对项目的影响、风险因子出现的典型方式中每一项。

表 14-7　DSS 实施风险因素

风险因子	问　题	典型环境	结　果
用户不存在或者勉强使用	缺少使用系统的承诺	系统不是由潜在用户发起，他们也没有参与开发	废弃；不平衡的使用；缺少效果
多种用户或者实施者	交流问题；无力将很多人的兴趣结合	在多人之间进行协调	不平衡的使用
没有用户、操作员、维护人员	没有人可以使用或修改系统	最坏情况：系统是某个已经离开了的人的工具，或者系统发起人在系统安装之前离开	使用减少或者系统消失
无力说明目的或者使用模式	系统设计者和支持者对系统过分乐观	假定非计算机专业人员能领会到如何使用系统	废弃
无力预报和减轻冲突	没有得到收益，缺少工作或者变换工作模式的动机	系统没有给予“饲养”人员好处 组织程序中被迫的变化	“为什么烦恼”综合症状；害怕和/或烦恼
没有或缺少支持	资金需求；不合作的人的阻碍	缺少运行系统的预算；缺少有效使用系统的管理行为	系统停用
缺少类似系统的经验	不熟悉导致错误	开发一个创新系统的目的是，引起实质的变化而不仅仅是实现操作上的自动化	技术问题；解决方法和问题不适合；错误使用或者不使用系统
技术问题和成本效果	维护和改进系统的成本	主张条件：在潜在的提高前后，都没有适当的方法去估计系统的价值	系统不适合需求；或者无力地用下去，或者废弃

资料来源：Alter, S. L. 1980. Decision Support Systems: Current Practices and Continuing Challenges. Reading, MA: Addison-Wesley.

DSS 实施策略

既然表 14-7 列出的因素进行了全部风险分析，下一步就是阐明可能的策略，以对付已确定的风险及其导致的后果。

像任何组织应用开发项目一样，实施策略选择是基于一系列的考虑的。在选择合适的 DSS 实施策略情况下，选择将依赖于资源需求和可用性、成功完成项目的可能性，以及与特定策略有关的特定环境约束。Alter (1980) 确定了 4 种普遍的 DSS 实施策略。表14-8 总结了 4 种策略并列出了其适合的环境。

框架中包含的各种策略并不是普遍有效的，注意到这一点非常重要。举个例子，尽管避免变化的策略将新 DSS 的实施困难降到了最低，而如果新 DSS 的主要目标是鼓励和促进变化，这个策略就是不可行的。换句话说，实施策略的选择，是系统的设定目标以及系统实施环境的特征二者的函数。

表 14-8 DSS 实施策略

实施策略	典型环境或目的	遇到的缺陷
1. 将项目分成易于管理的小块	降低产生不工作的笨重系统的风险。	
使用原型	努力的成功以相关的未经检验的概念为转移 ;在提交完全版本之前检验这些概念。	在日常使用中,对原型系统的反应(试验性的安装) 可能与对最终系统的反应不同。
使用进化方法	实现者尝试缩短自己与用户之间、意图与结果之间的反馈环节。	要求用户忍受持续的变化,这些可能是令人烦恼的。
开发一系列工具	通过提供数据库以及小型的可以产生、修改和丢弃的模型,以适应特别的分析需要。	有限的适用性 ;维护极少使用的数据需支付费用。
2. 保持解决方案简洁性	鼓励用户使用,避免吓走用户。	虽然一般来说是有益的,但会导致歪曲、误解和误用。
简单	对本来就简单的系统这一点并非关键。对于其他系统或情形,在简单方法和复杂方法之间进行选择是可能的。	一些商业问题本身不简单,坚持简单的解决方法可能会导致回避了真实的问题。
隐藏复杂性	系统作为“黑箱”出现,回答问题时使用用户看不见的程序。	非专家使用“黑箱”可能会导致结果的误用,因为非专家对潜在的模型和假定存在着误解。
避免变化	在将现有实践自动化或者开发新方法之间进行选择时,应该选前面一个。	新系统可能没有实际影响。对意欲鼓励变化的努力是不合适的。
3. 开发满意的支持基础	用户管理支持基础的一个或者多个部件不见了。	危险是 :“支持获得”策略将在对其他方面没有足够注意的情况下应用。
获得用户的参与	系统不是由用户发起的。开始时使用模式并不明显。	在多种用户情况下,将所有人包括进来并将其兴趣组合是非常困难的。使用成熟的模型,降低用户在模型表述和阐明中参与的可能性。
获得用户许可	系统在没有用户参与的情况下开发。系统将通过管理手段对用户施加影响。	在没有显示系统能够帮助用户的情况下,获得许可非常困难。
获得管理层支持	这样做,是为了获得项目持续的资金。为了使管理层强制人员遵照系统或使用系统。	管理层的热情可能感染不了用户,导致敷衍的使用或者停用。
推销系统	一些潜在用户没有参与系统开发也没有使用系统。组织没有最大限度地使用系统。	常常是不成功的,除非可以令人信服地展示出系统真正的优势。
4. 适合用户需要,将系统制度化	在正在发展的应用中,系统有很多用户。	这个标题下的策略多少有些矛盾,强调一点就有可能排斥了其他。
提供培训	系统不是由所有潜在用户共同合作设计的。	估计所需培训的类型和强度时,时常出现困难。开始的培训计划经常需要切实的革新和精心制作。

续表

实 施 策 略	典型环境或目的	遇到的缺陷
提供持续的协助	系统由中间人而不是决策制定者使用。系统在处理机器具体事务的中介的帮助之下使用。	如果系统由中间人使用,决策制定者可能并不明白细节上的分析。
坚持强制使用	系统是制定计划时综合和协调的媒介。系统旨在促进个人工作。	对计划真正的使用与不认真的屈从之间的区别。强迫用户用特定思维模式思考是困难的。
允许自愿使用	允许自愿使用,避免产生对强行推销的抵制。	一般来说是低效的,除非系统遇到真正的需求或者使用者非常理性。
依靠传播和显示	希望热衷者向他们的同事演示系统的益处。	或许这是一个策略,但同样也是缺乏积极行动的托辞。
系统适合用户的实际能力	用户使用分析技术的能力和爱好是不同的。	怎样去做还不清楚。实际上,系统看起来是建立在用户的需求而不是他们的能力之上。

资料来源:Alter, S. L. 1980. Decision Support Systems: Current Practices and Continuing Challenges. Reading, MA: Addison-Wesley.

在下一节中,我们将注意力从 DSS 的大致实施转移到更为具体但是同样重要的系统集成上来。不能将 DSS 有效的集成到组织现有的应用基础设施中,DSS 要成为组织日常决策流程中制度化的组成部分,可能性就非常小。

14.3　集成的重要性

决定综合的组织……

在某些案例中,DSS 的开发和实施可能是以这样的方式:它不需要为其他的组织应用提供特殊的入口或者界面,只是简单地作为独立决策支持工具来运作。随着专家系统、经理信息系统和数据挖掘应用的逐渐普及,单独的 DSS 将有可能消失。与开发组织 DSS 有关的成本显示了有权使用技术的决策者最可能的一致性。此外,对复合的和特别的信息资源的需求,要求现代的 DSS 以某种方式与现有的大多数应用和数据仓库联系起来。DSS 连通或者叫做集成到现有的组织计算机基础设施中,是本节讨论的主题。

将 DSS 集成到组织中只是意味着,新的应用与现有的计算机基础设施以展现在用户面前的无缝连接体系的方式结合在一起。中介而不是单独的硬件、软件、通信和数据访问的表现,对用户感知系统的易用性和有效性有重要的作用,而这两者对于 DSS 成功实施非常关键。

另外一个与集成有关的重要收益来自于多方面的协同作用,而这个协同作用产生于同时使用多种决策支持和开发工具促进模型和数据在不同应用之间运动。当提供给决策制定者多种集成工具和支持机构时,他关注于在问题环境中使用每个工具达到特定的目标。决策制定者可以关注于解决对于组织非常重要的问题,而不是必须将精力从手头上

的问题转移出去,以完成一些与应用相关的目标,例如将数据从一个 DSS 转移到下一个,或者使用其他应用中包含的工具在本应用中修改模型。

集成的类型

DSS 集成的两种类型可以确认为:(1) 功能集成;(2) 物理集成。

在功能集成中,DSS 不同的决策支持功能被集成,并与现有的基础设施连接起来。这个连接包括提供通用的菜单访问,交互的和内部的应用转移数据和工具,以及在相同的客户端和工作站的普遍应用界面。这样,单独或多种用户可以随心所欲使用所有可用的组织决策支持机构和资源。

物理集成包括新 DSS 相关的硬件、软件以及数据交换特征与现有的物理系统在构造上的捆扎。在很多情况下,物理集成在 DSS 设计之前甚至就存在了,因为与现有基础设施集成的需求,在某些情况下决定了开发中使用哪种 DSS 开发工具,以及采用那种软件平台。

物理和功能性的集成活动都经常是技术性的工作,并且在每个组织环境中是不同的。正因为如此,任何细节的讨论都超出了这章的范围,最好留给投身于机械内部世界的计算机科学。以我们的目的,认识到将 DSS 与现有的计算机应用库集成起来的重要性,以及进一步认识实现这个必要目标的复杂性,就足够了。

DSS 全面集成的模型

关于将 DSS,ES 和 EIS 系统与现有计算机结构集成,前人已经提出了很多模型。每个模型都有自己的方法。比如说,一些是问题驱动的,一些是应用驱动的,还有一些是功能驱动的。此外,每个提出的模型中都包含了自己的一套术语和名称,诸如管理支持系统(MSS)、专家支持系统(ESS)、智能决策支持系统(IDSS)。除去所使用的方法,这些提出的集成模型的共同点是,全面的连通性以及可以访问所有的组织交流渠道与决策支持机构。图 14-2 描述了组织中决策支持的一般全面集成的概念。

抵制和其他一些完全正常的反应

实际中的很多基于计算机的信息系统(包括 DSS),在实施和集成过程中面对的最普遍的问题是某些个人或者团体抵制新系统带来的变化。通常用户抵制新系统和变化是正常的,而且在某些情况下是健康的人类反应,认识这一点是非常重要的。

变化倾向于破坏人们的参考框架,特别是如果它看起来暗示过去的经验教训不真实,或者可能不可靠。技术变化是最普遍的,也是组织成员最怕面对的。当这种变化被视为危机或者是对紧急事件的反应时,抵制变化常常是最显而易见的。在新系统快要建成的情况下,对于那些不能被完全理解的或看起来需要很费力才能掌握的部分,用户可能会以一些方式进行抵制。他们可能简单地否认变化。也可能歪曲新系统的信息,这样他们就能使自己和别人相信新系统对状况没有真正的影响。在其他情况下,他们可能既不隐藏也不公开,不去使用新系统,这样来排除不希望有的变化的来源(Marakas 和 Hornik,1996)。但是,我们不能夸大有效集成以减少对改变的抵制的重要性。不管新系统对达到组织目标多么重要,很多因素都有可能导致对变化的抵制:

- **自身利益：当用户通过有效使用旧的系统已经取得了地位、特权或者自尊，他们常常会将新系统视为威胁。如果计划在特定的问题环境中进行服务以降低加班费，用户们可能就会害怕这一点对他们的银行账户余额的影响了。**
- **惧怕未知量：用户对于自己学习新技能的能力，对新系统的适应能力，或者接受新规则的能力都没有把握。**
- **良心的抗拒或不同的感觉：用户可能真的认为新系统或者不适当或者会低效。他们可能会从不同的观点来看待这个情况，还可能有自己或者组织的追求目标，而这个追求目标与新系统设计者是相反的。**
- **怀疑：用户可能不相信新系统，或不信任开发项目的成员。**
- **保守主义：组织或组织内部的团体可能自然地反对变化。感觉到一切都是良好的，与顾客失去联系，缺少对更好的做事方式的发现，或者决策制定的迟钝，这些都会导致上面的反对情况。**

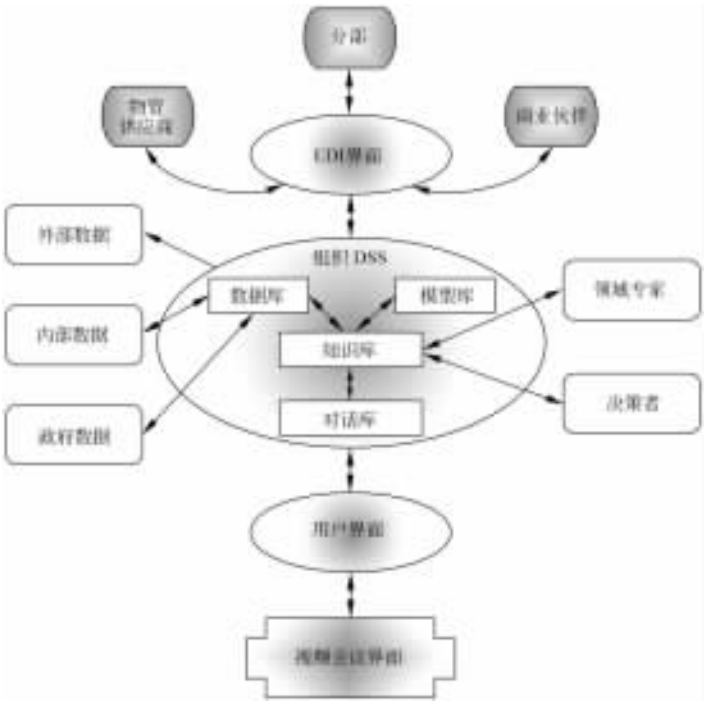


图 14-2　一般的全面集成 DSS (Min 和 Eom，1994)

表 14-9 列出了组织内新 DSS 应用的潜在反对者。

表 14-9　DSS 实施的潜在反对者

- **经理：一些经理担心自动化操作会使其丧失工作或失去重要地位。**
- **用户：一些用户抵制任何形式的计算机使用，另一些可能在适应用户—DSS 界面上遇到了困难。**
- **IS 人员：一些人担心会失去重要性或失业。**
- **非专家：一些人可能害怕在依赖决策支持技术的环境里，其无能会被发现。**
- **协会：抵制变化常常是增加协会成员数量的有效工具。**

这里重要的一点是,如果我们可以理解对变化可能的抵制的根源,我们就能制定计划,并在它成为主要障碍之前将其克服。

14.4 小 结

在本章中,我们完成了从第 1 章开始的循环。组织的 DSS 的实施,如果不超过,也与应用本身的设计同样重要。组织必须认识提供足够的资源给实施和集成流程以确保 DSS 项目能够成功和起到作用,这是非常重要的。在余下的两章中,我们着眼于现代组织中与决策支持相关的两个附加内容:决策制定中创造力的作用和 DSS 技术的未来。

复习题

1. 描述系统实施和变化之间的关系。
2. 描述 Lewin-Schein 的变化理论及其特征。
3. Kolb-Frohman 模型中变化流程的七个步骤是什么?
4. DSS 开发的初始推动力的可能来源是什么?
5. 简要阐述 Alter 用来划分实施模式的基本概念。这个分类有何价值?
6. 列出并简要阐述六种 DSS 实施模式。
7. 列出并简要阐述系统评价的框架。
8. 衡量用户满意度的不同方法是什么?
9. 系统业绩评价和任务业绩评价之间有何不同?
10. 描述本章介绍的衡量 DSS 成功的普遍框架。
11. 确定 DSS 实施项目中的风险因素。
12. 列出并简要阐述集成的类型。
13. 对于变化的抵制是自然的吗?为什么?

讨论题

1. 观察你熟悉的组织中的 DSS 实施,确定其实施类型。
2. 使用 DSS 成功衡量普遍框架衡量你所熟悉的组织中 DSS 实施项目是否成功。
3. 讨论明确表述和选择实施策略所需要的考虑事项。
4. 讨论集成的重要性。
5. 学习系统集成的一个成功案例,描述其使用的集成类型,并讨论对新系统带来的变化,抵制产生的影响。

第 15 章

创新性决策制定和问题解决

学习目标

- 学习创新能力理论的三个方面：心理分析、行为和过程
- 理解思考方式的五个基本分类：逻辑型、横向型、关键型、对立型和团体思考型
- 领会为什么对于决策制定者来说，将直觉和创新能力引入决策过程之中是很重要的
- 熟悉创新性问题解决技术的范畴以及它们的基本概念
- 学习智能主体（IA）的基本概念
- 熟悉智能软件主体的特性
- 学习智能主体能够解决的问题类型
- 熟悉将来的智能软件主体应用

小案例

捕捉新的想法

从 1964 年头一次参加加利福尼亚大学洛杉矶分校的一个写作班起，Marsh Fisher 就被创新能力迷住了。使他很惊愕的是，他在班上缺乏像其他同学那样的即兴发挥能力，不能像他们那样快速地想出好主意来。然而，当他开车回家时，他经常会想出一些有趣的主意来，他为自己为什么不能在班上想出这些同样的主意而感到疑惑。当 Fisher 追索这种制造有创造性联系的难以琢磨的能力时，他意识到他所能想出的那些有创新的主意主要受到他的记忆能力以及由他的生活经历所产生的各种联想的限制。

一般来说，人的大脑识别信息的能力强于回忆它们的能力。这是为什么做多选择题的测验要比做写短文的测验要容易的原因之一。Fisher 相信他在回忆和做出丰富联想能力上的局限是一种可以在计算机软件的帮助下克服的障碍。因此，当他于 1977 年以 21 世纪房地产公司的创始人身份引退时，他便将他的全部时间投入到一种软件包的开发之中，这种软件包可以通过扩展人们有限的记忆能力、帮助他们制造联想以及询问试探性的问题来帮助人们产生主意。

通过几年来对创新过程的研究，Fisher 总结道："创新能力大概就像做二加二的加法所需的技巧一样不可思议，两者都可以教授和学习，谁练得越多，谁就越精通。"

根据他的研究成果开发出来的软件包名叫 IdeaFisher，它是一个 25 兆的联合数据库（压缩后为 7 兆），被设计用来激励事实上任何主题的思想，并且可以为用户定制，包括特定行业或者产品的条目和链接。IdeaFisher 包括两个以一种独特方式链接起来的数据库，通过它们来执行这些任务。第一个数据库 Qbank 包含了将近 6000 个问题，这些问题被组织起来用于阐述问题，修改方案，以及评估解决方案。第二个数据库 IdeaBank 包含了超过 7 万个思想词汇以及将近 80 万个相关链接。它在某种意义上类似于一个巨大的自有关联的辞典。

IdeaFisher 软件的心理基础信赖于联想、记忆释放以及使用类推和暗喻的原理，思考者交互和使用思想词汇来刺激他们自己记忆库中的主意和联想。新的联系和联想被刺激产生，无数的主意就可以在一个非常短的时期内产生出来。

IdeaFisher 依靠模仿人的大脑对经验的组织和记忆，通过形成联想来将单词和概念联接在一起，从而加速了创新的过程。这些主意的联想在大脑中对所考虑的主题形成了不同的图像，从而刺激了新的观点和主意的产生。进一步来看，这些联想可以用于产生新的产品或服务的方案，新的市场或者对现有产品或服务新的用法，新的广告或促销活动，有特色的事件或程序，培训计划，演讲，故事或剧本，甚至公司或者产品名字等。

使用 IdeaFisher 的一个典型例子来自 Jim Shenk，他是位于加州的 CEC 设备合同和项目管理部门的管理者，他用这个软件来推动一项战略计划的处理。"我们需要为部门做一些战略性的思考，于是我们使用 IdeaFisher 作为这一过程的向导。它做的头一件事是问一些浅显以至于你一般不会去问的、实际上你不知道该怎么回答的问题，诸如

‘我们在什么行业中’及‘为什么顾客买我们的产品’等。正如前面提到的，IdeaFisher 提供了将近 6000 个问题来帮助刺激思考。”

“我们问了自己大量的问题，这些问题的答案把我们带到了一个机会面前。我们太过于依赖航天工业了，这个机会帮助我们建立了该行业之外的一个市场——在商业航空领域中的市场。”

“我借助这个软件建立了一个目标声明以理解我们的商业业务。那些问题对于思考是非常有帮助的。我还使用了战略计划模块，它有专门为计划而组织的大量有趣的问题，演讲计划模块也是这样。”(可用的还有一个商务和授权建议模块。)

重要的是要记住 IdeaFisher 不会为你创造任何东西。它不像一个电子表格或者数据库那样能够生成什么东西。IdeaFisher 只是帮助刺激你的思考，并保持你对那些绝大多数人倾向于认为是太浅显，以至于不必去思考的那些问题的重要性。在实际中，我们无法回答那些问题中的许多问题，因为我们实际上从来没有想过它们……而当我们经过充分考虑能够回答它们时，我们会发现它们将把我们带向以前我们也许从未发现过的方向。

“IdeaFisher 是一个能使你发现 100 个新主意的好办法……甚至即使你拒绝其中的 99 个，那又怎么样？如果你能够获得一个有用的主意或者遇到了一个新的机遇，那么你所经历的整个过程就是很有价值的了。”Shenk 说道。

Clayton Lee 是位于休斯顿的一种名叫“Obiter”的免冲撞的蹦床踏车的发明者，他使用 IdeaFisher 来降低附加于 Orbiter 之上的单位成本……从每单位 2000 美元降低到大概每单位 100 美元。在成本上的大量节约使他得以开辟另一个新的市场，Lee 说：“IdeaFisher 使我们将思考更多地回到市场上。它有助于开发我们的大脑——我们似乎已经忘了怎么去用它了。思考不再是个可有可无的技巧了。”

Ron Sargent 是一位熟练的工程师，他为 GM，Porta Potti 以及迪斯尼世界 EPCOT 中心等设计了它们第一个巡游控制系统。他说：“我们几乎把 IdeaFisher 当作一个广泛的平台，一种为创新而开发的 AutoCAD。”

IdeaFisher 是一种强大的思考工具，应用于广泛的领域之中。除了前面的例子外，IdeaFisher 还被用来书写训诫、设计织锦、开发大学课程、创建产品名称、以及为果园种植业发明树木的振动机修正器等。它甚至在美国副总统戈尔重组政府的项目中发挥了一些作用，它被商业部用在一系列重要的团体会议中，把会上的抱怨和指责转换成有用的建议。

尽管在人工智能和基于计算机的决策支持机制上有了进步，但我们在对关于人类决策制定过程所做的某些基本的假设还仍然仅仅是假设。例如，在决策模型的内容上，我们经常面对这样一个假设：所有行动的可选方向都是已知的或者是易于确定的。然而，在绝大多数非结构化问题中，这种假设显然是站不住脚的。我们知道如果要了解一个可接受的选择，我们就必须产生许多新的思想。通常这些新思想的产生是发生在问题解决的正常过程之中的。换句话说，有一些我们知道的东西正在发生，而我们却不能完全理解它

们在过程或结果中的角色。这其中之一就是在决策制定中创新能力的角色。在本章中，我们将探究创新能力的概念，并确认在管理决策制定中对它所扮演角色的已知部分。通过这种调研，我们将能够开始理解，为什么在决策支持系统领域中对创新能力的学习正变得越来越重要。

15.1 什么是创新能力

对创新能力的一种理解方式是我们和别人看到的同样的东西，但是却能对这些东西想出一些与别人不一样的东西来。创新能力是一种概念，对它自身的定义几乎和由它所产生的新思想一样多。从哲人到俗人，每个人似乎都有他自己对于创新能力的定义。通常这些定义都会是："我并不是真的知道它是什么，但当我看见它时，我就知道了它。"然而，不管是定义的起源还是定义的中心，看起来都赞同创新能力包括了产生新颖和有用的思想，以及对问题挑战解决办法的能力。

创新能力包括将我们独特的天赋、才能以及想像力，转换成某种新的有用的外部实体的能力。通常创新能力包括将多个单目标项目结合成一个新的服务于完全不同的目标的项目。当 NASA 意识到需要为太空飞船的热护套开发一种低阻力的覆盖材料，以保护宇航员在重返地球大气层时不会被烧坏，别人也想出了把这同样的覆盖材料作为解决鸡蛋粘锅的一个好办法，这种材料就是特氟纶。当阿波罗 13 号的宇航员 James Lovell 和他的船员们试图将他们自己和破损的太空飞船从月球开回地球时，他们面临着氧气几乎耗尽的危险。创新能力帮助他们和他们的团队使用与所面临危险无关的登月车中的部件，为登月指挥舱做了一个二氧化碳过滤器，使他们能够循环使用他们的氧气。当啤酒压榨机和模具冲压机于 17 世纪发展起来时，它们各自的发明者看到了它们在制造啤酒和制造金属铸件上的用途。而另一方面，一个名为 Gutenberg 的人则把这同样的东西作为制造出一台能够大批量印刷手稿稿件的印刷机所必需的部件。只要想像一下如果没有发明印刷机，我们手里拿着的只能是一本手抄的卷轴时，就可以感到他这么做对于我们来说也是多么的幸运。

创新能力是用新产生的思路来做旧的事情，以及想出办法去做那些尚未做过的事情的能力，而这些新的思路和方法对于获得在世界市场中生存和成功所非常需要的，对取得竞争优势也是绝对必要的。因为 DSS 用于支持决策制定过程，而这个过程中一个基本必需的因素是新思路和方法创新的产生，因此将这两个概念联系起来是很合适的。

创新能力理论

有关创新思想的起源是心理学家们几十年来所进行的争论之源。在这场争论中出现了大量的用以解释为什么有些人看起来比别人更具有创新能力的理论。

心理分析的观点认为，创新能力是一种超前于意识的精神活动。换句话说，创新的思考发生在我们潜意识的思想之中，并且同样的无法为我们的思想意识所直接访问。行为科学的观点则认为，创新能力不过是对在我们自身环境中的刺激的自然反应，而某些刺激的组合会比别的刺激的组合更能引发创新。而第三个观点——面向过程的观点认为，创

新能力是思考过程的一个属性,它可以通过指导和练习来获得和提高。最近的研究指出,个人的创新能力并不像发明能力或者独立能力等等那样属于明显的个人特性,因此,它是可以学习并能够提高的。这意味着创新思考的潜力或多或少地存在于每一个人之中(Marakas 和 Elam,1997)。这导致商业产品流入了 DSS 的领域之中,目的在于培养和增强企业中各层次的决策制定者和组织问题解决者的创新能力。

普遍认为一个人要想"教授"创新能力,他就必须打破会对新思路的产生造成障碍的硬性思维。在这方面,对创新能力的增强是一个改变的过程。通过引出和促进不同的思考方式,创新能力可以得到增强,新的思路也得以产生。

不同的思考方式

当我们考虑一个特殊问题的背景时,我们引导思路的方法,会对问题可选方案的产生或者从可选集中找出一个合适的方案产生重大的影响。采用一般的分类方案可以将思维方式定义为五种基本的类别:(1) 逻辑型,(2) 横向型,(3) 关键型,(4) 对立型,(5) 团体思考型。

逻辑型思维

决策制定者思考问题的一种方式建立在他的经验和分析能力之上。逻辑型思维(logical thinking)也许是最普通的、最广泛应用的、以及最经常被推荐来解决问题的思考方法。这种思考方法源于管理科学的建模和量化思想,而且是设计和开发基于计算机的决策支持系统所需的一个不可缺少的组成部分。

横向型思维

横向型思维(lateral thinking)最初是由英国内科及心理医生 Edward de Bono (1970) 定义的,它提供了一种脱离传统的"直线"型思考的思维方式。该思维的前提是:人脑是天生地依照某种特定模式的思想和行动来处理和存储信息的。正因为这种逻辑性的分类处理,大脑在做任何改变这些公认的思考方式的尝试时就会受到阻碍。横向型思维通过引入不连续性来打破这些传统的思考方式。表 15-1 对传统的直线型思维和横向型思维过程的特性进行了比较。

有三种行为可以促进横向型思维过程:(1) 意识 (awareness),(2) 抉择 (alternative),(3) 刺激 (provocation)。

意识　这个类别中的行动倾向于对当前思想的重新定义和阐明。在任何新的思想可以产生或者任何旧的思想被拒绝前,当前思想的范围必须得到完整地定义和理解。诸如对问题相关内容中主导思想的直接定义、对问题空间的边界的观察,以及对问题典型假设的定义等行为,对于促进横向思考所需的意识来说都是必需的。这里需要着重强调的是,意识行为仅仅是定义当前的思想而不是对它们进行评估。对思想的评估过程在其之后进行。

表 15-1 直线型思维和横向型思维的特性比较

直线型思维过程	横向型思维过程
评判各种关系时考虑其稳定性及对这些关系绝对的定义	考虑变化的和移动的事物,试图找到新的方式来看待事物
在过程的每一步试图通过“是”或“否”的判断来找到“正确”的答案	避免寻求绝对的“对”或“错”的答案,将注意力集中在寻找差异上
分析思想的成功的可能性	分析思想对产生新的思想有多大的作用
通过逻辑上一步接一步的处理来保证过程的连续性	通过鼓励非逻辑的跳跃性思维来将非连续性引入过程中
拒绝任何不认为是直接相关的信息,并有挑选性地选择在产生新思想中需要考虑的内容	欢迎机会的突临,任何事物都视为相关的
首要着重于明显的部分,并倾向于用已经建立的思考方式来进行处理	避开明显的东西,着重事物的处理过程
确保至少可以找到一个可接受方案	提高找到新的或有创新的解决方案的可能性,但是避免做出任何保证

抉择 在这个行为的第二个分类中做出有意识的努力,以产生尽可能多的不同方法来看待问题。这种努力的目的是为决策制定者提供尽可能广泛的解决问题的方法和思路,而不考虑哪种是最好的。为了使之容易,可以用任何数量的假设来产生一个抉择,而不必对那些假设做进一步的评价。这种灵活的问题分析形式允许决策制定者免受传统的假设判定的边界束缚,并把注意力集中在抉择产生的“如果我们能这么做会怎么样”的部分上。使用诸如旋转注意力(从一个不同角度来看待问题)、建立选择定额(强制生成给定数目的选择)、以及问题区分(将问题分成随机的部分而不考虑其逻辑边界)等技术,决策制定者就可开始生成新方法来做旧事情的过程。

刺激 刺激的行为是为了直接帮助新的想法的产生。刺激的最主要的目的是为了在思维过程中引入非连续性,这是通过强制改变我们看问题的方式来实现的。这种改变包括问题内部的环境和外部激发问题的环境。内部的改变可以通过把问题反过来看待而实现。例如:假设我们面临一个使用停车计时器来减少某个区域站台附近停车的情况,反过来看待该问题环境,我们需要从鼓励停车的角度进行分析。从这个角度来看,我们或许该建议为每个城郊购物的人提供在购物时可以免费每周停车一天的优惠票。这样解决将增加城郊购物者在销售点附近停车的数量,从而相应减少了在日常交通站台附近停车的数量。在横向的思维方式中,我们进行反向思维倒不是说其正确性有多么重要,而更看重一个简单的事实,就是我们看问题的观点的改变。

另一个在内部进行改变的方法是采用夸张。与反向思维不同,这里改变的方向是增大问题特征和界限,而不是简单地反转它们。例如,假设我们的问题是控制我们的生产设备制造的若干产品的质量,就可以夸张地认为我们所有的产品都有问题。从这个视角来看待问题,我们或许会发现新的革新性方法来完全重新设计生产设备,从而提高所有产品的质量。

外部环境的改变也可以用来激发横向思维。最通常的方法就是使用类比。通过将问题引入到一个崭新的环境中,并为其开发和优化解决方法,然后将这些方法转回到原来的

问题环境中，通常就可以发现新的有创意的可行方案。比如下面的例子。

假设我们的问题是找到一个在处理废物的过程中减少用水的方法。当前的流程只是简单地将可溶解的废物置于一个水容器中，然后定期将其冲入下水道之类的地方。水推动废物在通道中流动，到达一个更永久性的收集点。问题是因为水和废物混合在一起，该永久收集点必须足够大来容纳那些废物和用来冲走废物的水。我们需要一个类比环境来从另一个角度看待这个问题。

对这个问题的一个类比可以是，伐木企业如何利用河流来将它们砍下的木头传输到一个合适的运输位置。工人们只是简单地将砍下的树推到河中，然后让河流把木头送到装车码头，然后运送到工厂去。木头在收集装车时，用来传输的工具（即河流中的水）将被留下以被其他的自然过程重复使用，比如下雨和浇灌。能够这样做的原因是因为木头是不溶于水的，因此就决不会和水混杂在一起。这种非溶解性就使得对物件可以进行传输而传输的工具在物件到达目的地时容易收回。如果我们对固体废物如此处理将如何呢？通过使用一种不会溶解废物的液体来进行传输，比如油等，我们就可以将废物传送到收集点，然后回收油，并将其重复利用，这样就无需既存储废物又要存储油了。

关键型思维

关键型思维（critical thinking）方式认为在问题的环境中存在的某些元素是对任何可能的解决方案来说最关键的元素。在此基础上，通过仅仅关注于关键因素，能很快地建立一个解决方案，迅速对问题产生重大的影响。在关键型思维基础上最通常的解决问题的方法称为帕累托规则（通常也指的是 80—20 的原则）。图 15-1 就说明了一个产品的部件和各自成本的假设的帕累托分布。

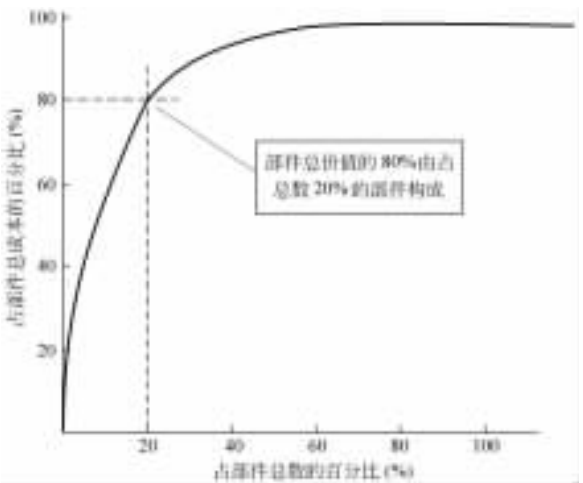


图 15-1　制造某产品所需部件的帕累托分布

正如我们从图中可以看到的，产品的 80% 的成本由其制造部件的 20% 构成。如果我们希望控制这件产品的总成本，则只需要重点控制这 20% 的关键部件就可以控制 80% 的成本了。因此我们可以设计一个系统连续地监视这 20% 的部件，而通过例外报告或者控

制的方式来对付剩下的 80% 的部件。关键型思维方式将任何的问题分为重要的和不重要的两个部分。

对立型思维

对立型思维 (opposite thinking) 是一种问题解决方法,其中决策者采用其他人的观点而非自己的观点。这是一种"穿上别人的鞋子"的方式。使用这种方法,一个决策者通常可以洞察为何特定的问题会首先出现。考虑如下问题:

Lynne 可能是 Trident 软件公司最有价值的软件开发人员。似乎在任何大的工作碰到无法解决的难题的时候,她都能进行解救,并为客户提供成本效益最好的解决方案。公司最后获得了一个已经对其努力了数月的重要客户的关注,其中大部分功劳就是 Lynne 曾经为该客户的两个最大的竞争对手做过工程。该客户希望在下一周的最后两天安排计划会议,并且因为一些政府事务而不能对该安排进行什么调整。Lynne 坚持认为她那两天不能参加会议,当要求说出原因时,她说下一周是她父亲逝世一周年纪念,她需要去印第安纳陪伴母亲。Lynne 知道该会议的重要性,但是仍强调她那两天必须要陪伴母亲。如果 Trident 拿到该业务,就必须要参加下周的会议而且必须要 Lynne 出席。你有什么办法吗?

看待该问题的一个方法就是从 Lynne 的角度而非 Trident 公司的角度来看。Lynne 认为她的母亲在那期间必须要她陪伴才能受到安慰。从逻辑上来讲,Lynne 并不一定要去印第安纳,而是要和母亲在一起。从她的角度看,答案就变得清楚了。为什么不将 Lynne 的母亲接过来而不是由 Lynne 去看她的母亲?当被问及此时,Lynne 看到了这个双赢的局面并同意了。对立思维通常能够产生新的看问题的角度,从而产生新的解决方案。

团体思维

因为团体思维 (groupthink) 已经在第 6 章进行了详细的说明,我们这里就不再多加讨论了。而在此简单地提及,是因为尽管它存在消极因素,它毕竟还是提供了另一种可行的思维方式,而且可以用来有效地达成某些活动中的团组参与,诸如思维生成和头脑风暴 (本章稍后讨论)。

直觉

尽管和创新性不太相同,直觉也通常被认为是创新性决策制定和问题解决中的一个重要元素。管理人员利用其产生的直觉通常可以更快地对给定情况作出反应,并能同时应用经验和判断。考虑如下例子:

> 在法国的马赛,一条渔船被怀疑进行毒品走私。非常不幸的是,当海军巡逻队登上这条渔船时,他们没有找到任何毒品。当海军巡逻队准备离开时,一个官员注意到该船的水泥压舱物不是放在通常放置的中部,而是放在前部。当检查压舱物的时候,巡逻员发现它是中空的,里面藏着很多海洛因。这位海军官员的直觉帮助解决了这个问题。(Rowe 和 Boulgarides 1994, 172)

在 DSS 的设计实践中,必须建立一种机制,允许决策者将直觉应用到决策过程中,并作为包括创新性在内的其他问题解决方法的补充。

15.2　创新性问题解决技术

回顾有关创新性问题解决方法的论著,可以很快发现已有很多不同技术,应用于需要解决问题的环境中。VanGundy 的《结构化问题解决技术》(1998) 中就详细描述了各种问题环境中的上百种蕴涵创新性的技术方法。这对于 DSS 设计人员来说是好坏参半的:好的一面是在需要加强或者促进创新思维的特定环境下,很可能存在特定的解决方法;坏的一面是由于 DSS 设计和应用本质上是支持半结构化的问题,从而使得判断使用哪种创新性问题解决方法技术就成了一个主要问题。值得庆幸的是,这些在电子化决策支持机制中适用的创新性问题解决技术可以很容易地分为三个方面,即:(1) 自由联想;(2) 结构化关联;(3) 群组技术。

自由联想技术

这一类的技术全部着眼于一个单一的目的:发散的思维及其产生。在指导自由联想 (free association) 技术的几个原则中,最重要的一项可能就是推延判断。发散思维的产生有赖于将分析和判断完全地保持暂停。如果每次思想的产生都必须进行完全的分析和判断,那么可以考虑的可选方案将大大减少,能够产生新的创造性的解决方案的机会也就减少了。

指导该类技术的其他原则还包括如下假设:数量是质量之父;越狂热越好;联合就是提高。第一个原则简单地假设在推延判断的条件下,将产生更多的思想,按照概率原则也就有更多的好思想存在其中。第二个原则说明发散思维需要一点冒险。保守和谨慎的思维方式通常不能产生突破性的思想。第三个原则提倡不仅要生成广泛发散的思想,还要将这些发散的思想进行联合并得到一个新的更好的结果。某份杂志在 1912 年的一个为 Oldsmobile 的广告中就引用了 Olds 先生的一句话,“我不相信能够制造出比这更好的车。”在那时这听起来真是一个美妙绝伦的梦想。而对我们来说,幸运的是,1912 年以后,一些人成功地进行思想的联合,并将 Olds 先生 1912 年所预言的车进行了相当的改进。

头脑风暴

在 DSS 领域应用最为广泛的自由联想技术可能就是头脑风暴 (brainstorming) 了。该方法已被证明是非常有效的,无论对于个人还是小组都能够快速有效地产生一个长长的主意列表。头脑风暴技术在上述原则的基础上,主要着眼于数量和多样性,并对判断、评判和分析进行推延,直到能够完成主意的列表。尽管该过程最初通常导致可能方案的数量过于庞大,但是进一步地将这个列表精简到一个易于控制的程度并不需要多大力气。头脑风暴可以应用于个人,也可以应用于团组,既可以应用于非结构化的问题,也可以应用于结构化的问题。在多方参与的环境下,头脑风暴可以刺激思想搭乘现象,即一个想法可以导致另一个想法的出现,继而又导致另外的想法,这样不断重复,最后可以得到更加

精炼和完善的思想。最重要的是，头脑风暴可以很容易地集成到 DSS 中，作为一部分向导过程或者一个独立的模块，这可以按照用户需求来决定。在 DSS 环境中，还可以将诸如主意列表的合并、排序和精简等附加功能集成到系统中，以辅助头脑风暴的进行。

很多用以辅助思维产生和头脑风暴功能的商业化的产品已经开始出现。其中最为成熟的一个就是 IdeaFisher（参见本章开始的小案例），该产品包含一个关联英语词典，能够快速方便地产生数以千计的交叉关联和词汇联想，从而帮助想法的生成。图 15-2 中例示了 IdeaFisher 中的几个典型界面。

图 15-2 IdeaFisher 的界面

另一个非常成功的电子化头脑风暴产品是 Ventana 公司的 GroupSystems 的一个功能部分。作为一个市场化的 GDSS 产品（参见第 6 章），该产品中所包含的头脑风暴工具据说是目前最为复杂的自由联想工具之一。图 15-3 显示了 GroupSystems 的几个典型界面。

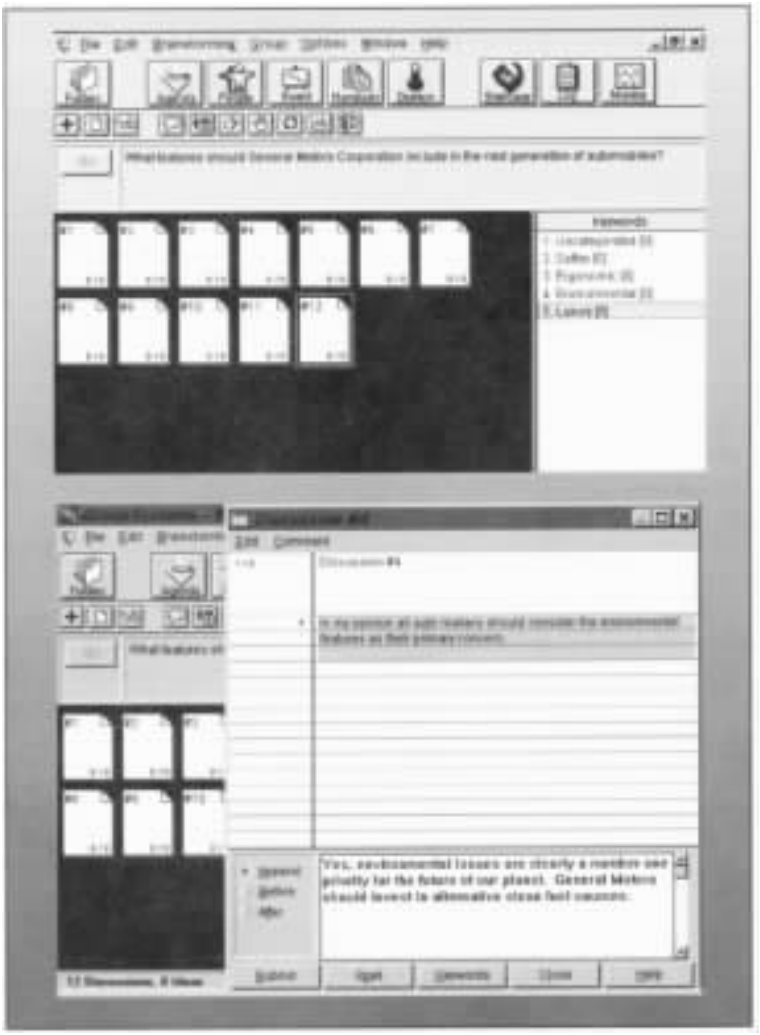


图 15-3　GroupSystems 的界面

结构化关联

在结构化关联 (structured relationship) 这类创新性问题解决技术中，着重点在于通过如下过程来生成想法，即将两个或更多的对象、产品、想法或者概念强制合并来生成一个新的对象或者想法。尽管通常情况下，相互有联系的元素会比无联系的元素所产生的想法更具有实用性，但是这种想法往往太过平凡和没有特色。

Osborn 的思维清单

在该类中有一项传统技术使用清单来记录解决问题的各种潜在的观点和创新性来源。在这方面最著名的例子就是 Osborn 的 73 个刺激思维的问题 (1973 年)。这些问题要求用户将决策内容转换到多种新的视角和形态上，从而强制性地重塑其中的元素。表

15-2 即列出了一个完整的 Osborn 的转换问题的清单。

表 15-2 Osborn 的 73 个刺激思维的问题

- 试着应用在其他方面。有哪些其他方面可以应用？如果进行修改又有哪些应用？
- 匹配。还有哪些与此相似的？由此产生的其他的想法又有哪些？过去是否有与此类似的？我可以从中复制些什么？我到底应该效仿谁？
- 修改。有无新的变化？是颜色、运动、声音、气味、形式、形状的变化？还有其他的变化吗？
- 扩大。应该增加什么？更多时间？更加频繁？更健壮？更高？更长？更密？其他的值？附加的成分？复制？相乘？还是夸大？
- 缩小。缩小什么？是使之更小？还是浓缩？或者缩影？或是更低？更短？更轻？甚至忽略？还是流线化？拆分？或者低估？
- 替换。有谁可替换？有哪些可替换？其他成分？还是其他原料？或是其他流程？其他力量？其他地点？其他方式？甚至其他语调？
- 重排。交换组件？其他模式？其他设计？其他顺序？变换原因和效果？调整步调？修改进度？
- 反转。变换正与反？对立面如何？向后转变？上下颠倒如何？变换角色如何？或者换双鞋子？翻转桌子？试试另外一面？
- 合并。试试混合、合成、分类或者加总如何？合并单元？合并目的？合并请求？或者合并想法？

资料来源：Osborn，A. O.，(1963)，Applied Imagination. Reprinted with permission from the publisher，The Creative Education Foundation Press，1050 Union Road，Buffalo，NY14224.

形态性强制关联

Koberg 和 Bagnall 提出了另一种强制性关联技术，即强调问题设计的形态学（研究生命变化的科学）特征方面的技术。该技术要求用户写下一个问题的各种特征，并为每种特征列出尽可能多的可行方法，进而考虑这些可行方法的所有可能结合和置换。这种分析最好通过一个矩阵来进行，这样也很容易采纳到 DSS 中。表 15-3 就列出了一个有关毒品走私问题的形态分析。

表 15-3 毒品走私的形态分析表

问题组成	可 替 换 项	
类型	A. 可卡因	B. 海洛因
	C. 安非他明	D. 大麻
来源	E. 泰国	F. 当地来源
	G. 土耳其	H. 美国
进口方法	I. 空中	J. 海路
	K. 信使	L. 本地生产
分销方法	M. 大规模，组织良好	N. 业余
	O. 小规模，组织分散	
赞助者	P. 有组织的犯罪团伙	Q. 三人组
	R. 政治性恐怖组织	S. 个人

可能的形态关联包括：

- 美国 (H) 当地产 (L) 的可卡因 (A) 通过有组织的犯罪团伙 (P) 大规模有组织地分销 (M)
- 土耳其 (G) 的海洛因 (B) 通过空中运入 (I)，并通过个人 (S) 小规模的分散性组织进行分销 (O)

NASA 在它的很多 DSS 中就采用了形态性强制关联模块，来支持确定空间探索活动的科学目的。诸如“有关……的潮汐变形的测量”就和“木星内部”相关联从而确定了一个可能的科学目的。

这种强制性关联技术的真正价值不在于其能够找到所有可能关联的能力，而是在于它能够产生一个所有可能关联存在的框架，并能够选择性地列示以决定最合适的候选项。

层次分析方法

要把形成决策的所有不同因素进行抽象形成概念通常是很困难的。因此，遗漏一个或多个因素，或者包含了决策中不重要的因素，是很常见的错误。另外，由于要对决策因素进行排序需要很强的感知能力，所以要记录下所有先前的排序结果非常困难，因此这往往导致在优先性排序判断的前后不一致性。为了解决这种问题，Thomas Satty 20 世纪 80 年代在宾夕法尼亚大学时建立了一种叫做层次分析的方法（analytic hierarchy process，AHP）。

AHP 方法是一种基于数学的方法，可以广泛应用在决策制定方面，其中有如下两个关键要素：

1. 组成决策的各种变量的数据值。
2. 决策者对这些变量的判断。

AHP 允许决策者组织和评估所选目标的相对重要度和各种可行方案的相对重要度，从而为决策过程提供支持。AHP 使用一种方法来测度判断的一致性，如果存在非一致性问题就对这些判断进行重新评估。最后，该方法还提供了一种进行一致性决策的理论基础。

要决定最好的总体性行为，AHP 需要如下步骤：

1. 将决策问题结构化为一个层次模型。
2. 对所有条件和可行方案进行两两的比较。

首先，如前所述，每个 AHP 决策都是从一个层次结构的模型开始的。模型的通常结构由如下四部分组成：

1. 目标：即决策目标。
2. 准则：有时也称为条件，即构成最后目标的因素。
3. 子准则：即最低层次的准则，构成它们上层准则节点的所有因素。
4. 可行方案：可选的不同解决方案或者选择。

将一个决策分解为这种格式，使得决策者可以集中精力于复杂问题中的每一个部分。

一旦建立了决策模型，下一步就是通过两两比较来评估各个因素。两两比较的过程就是进行任意因素关于上一层因素的相对重要度、偏好或者可能性等的比较。两两比较通常可以是从上到下的进行，并包括跨节点和节点内部两种方式。比如下面例子：

1. 关于最终目标来说，准则一是否比准则二更优越，更重要，或是更可行？
2. 关于最终目标和准则一来说，子准则一是否比子准则二更优越，更重要，或是更可行？

因为我们很少能在比较判断的时候总是保持一致性,所以 AHP 假设所有判断都是非一致性的,并试图通过一个非一致性比率公式来测量这种非一致性,其中 1.0 被认为是完全一致的,小于 0.10 的值则被认为是可容一致的,而如果该值大于 0.10,则需要重新检查上面的判断,以图提高一致性。

在记录下所有的偏好并保持了足够的一致性之后,将导入模型中各子准则相对应的数据。这个过程被称为合成。

利用如下两个要素——(1) 一致性偏好和 (2) 子准则的数据——AHP 方法通过合并整个模型的每一层的优先级生成节点的整体权重。通过合成,我们得到可行方案的总体优先级,从而为决策者提供了足以判断哪个方案是最好的选择的信息。

下面演示了一个 AHP 层次决策的例子:目标是决定是否接受一位申请者读 MBA。一开始,该决策被分解为子因素,这些影响决策的主要因素包括申请者的学历、职业和个人能力证明。这些因素也就是决策准则。在其中的每一个准则下,还有子准则,即组成各个准则的子因素。比如,申请者的学历证明包括本科时总体的 GPA、主要科目的 GPA,GMAT 成绩,本科学习院校的质量等等。最后,可选结果是一系列可能选择:在本例中也就是所有的申请者。每一个可选结果或者说申请者,都要为每一个子条件提供数据,从而进行合成。通过进行完全的两两比较,并对可选结果在此基础上进行排序,AHP 方法能够帮助决策者得到一个最后排名结果,它能够反映每一个子准则对最终决策也就是申请者选择的相对贡献度。

团组技术

正如其名称所指明的,该类创新性技术着眼于加强和鼓励多方参与解决问题的环境中的创新性。这些技术中的一个关键要素是如何提高团组内的交流。它基于的假设是:参与者在被激励进行常规的、自由的交流时,自然地会进行相互协调,从而产生创新性的思想。一旦交流开始,该过程就会自己发展并产生创新性想法。

记名式的团组献策技术

Delbecq, Van de Ven 和 Gustafson (1975) 年成功地开发了一种被广泛使用的团组交流机制,即记名式的团组献策技术 (nominal group technique, NGT)。这项技术建立在多名参与者环境下的头脑风暴概念之上,可以使用多种媒体,包括书面的、语言的和电子的等来促进交流。

该技术名字本身也就说明了它的运作过程。使用该技术的团组仅仅是名义组成的 (该技术也是因此得名)。NGT 的目的是为了消除团组行为的社会上和心理上的动态性,这种动态性将会抑制 MDM 环境下个人的创造性和参与精神。当团组使用这种技术时,要避免发生诸如个别人滔滔不绝,其他人侧耳聆听,而很少有人真正会就讨论的问题认真思考之类的常见问题。每个人都应当更富有创造性,每个人也都有机会参与进来。这将有助于克服那些小的团组在生成想法、计划程序和解决问题时往往遇到的一些问题。表 15-4 列出了 NGT 的每个步骤。

表 15-4　团 组 技 术

将想法写下来
- 提供时间进行思考
- 提供一个创新性的背景
- 提供一个思考重点并使思维不受干扰
- 鼓励每个成员搜索想法
- 避免发生竞争和地位差别
- 避免需要达成一致的压力
- 避免进行评估和总结
- 避免想法的两极化

循环用图表记录和列示想法
- 建立同等共享和参与的结构
- 鼓励问题鉴定
- 鼓励每个成员基于其他成员的想法来建立自己的想法
- 将想法非个人化
- 容许相互冲突的想法
- 不断通过听和看这些想法来集中注意力
- 提供持久的书面材料

讨论和解释图表上的想法
- 每个想法都和其他想法同等重要
- 每个想法都用同样的处理时间
- 解释所有想法

就优先级进行初步投票
- 提供重要问题的焦点所在
- 构造选择的平等性
- 允许一次试验投票
- 避免不成熟的决策
- 避免重要人物的影响

讨论初步投票结果
- 澄清错误认识
- 鼓励少数意见
- 推动对想法的批判——针对列出的想法而非针对人
- 为最后决策做准备

就优先级进行最后投票
- 每个成员形成各自独立的判断
- 进行总结
- 促进成就感
- 激励在计划和解决问题的将来阶段的参与
- 提供一个所生成的想法的书面记录

Delphi 法

回忆第 6 章中我们曾简单涉及到的所谓 Delphi 法的 MDM 技术。这项技术最初在兰德公司 (20 世纪 50—60 年代一个著名的智囊团) 得以发展,用以得出有关前苏联对美国各地实现特定的毁灭程度需要多少颗原子弹这一问题的专家意见。它还在很多研究小组

以及很多商务组织和私有企业中的 DSS 中得到了应用，包括加拿大贝尔公司，Honeywell，AT&T，Owens-Corning，General Dynamics，TRW，Skandia 保险公司，Weyerhaeuser，Imperial Chemical 等等。

Delphi 技术不同于 NGT 的一个关键因素就是所有参与者都是匿名的，而且在大多数情况下，他们并非同处一地。Delphi 中负责问卷的人通常为团组中的成员提供精心准备的书面问卷，偶尔也会使用口头询问。这些问题通常需要小组成员进行主观估计、评价和预测。

然后对第一轮的问卷进行分析，通常是进行统计分析（比如 均值、方差、中值、四分值、值域等）。这次分析的结果和回答者的初始估计作为反馈提交给原来的那些参与者，然后要求他们在知道了团组整体反应的基础上重新进行估计并做适当的修改。

上述过程可能要进行数次，直到达到一定的收敛（即方差减少）。

研究显示 Delphi 法形成的一致意见要比在语言形式的团组交流中得到的意见更加有效，也更加准确。这是因为在通常的面对面团组交流中，某些个人或者特权类型的人意见往往会占统治地位，这些意见并不一定就是最正确的。而这种无需言语的静静思考下的 Delphi 方法，将达成比嘈杂的会议环境下更好的结果。而且，Delphi 法通常对小组专家时间的利用更加高效，也更容易集成到 DSS 中。

Delphi 技术的使用有一个前提假设，就是在已知团组反应的基础上，那些对原始意见并不太自信的人，往往会比那些比较确信自己的估计的人更倾向于改变自己的估计。

15.3 决策支持中的智能主体

尽管创新性仍旧是人类认知研究中最后的阵地之一，但是其中新近兴起的智能主体（intelligent agent，IA）技术却已迅速地跻身于包括 DSS 在内的与计算机相关的所有领域，它可以辅助人们向计算机部署任务。IA 技术确实是未来 DSS 和问题解决方案的一缕曙光，智能主体能够执行很多之前必须由人类进行的决策支持任务。它们可以帮助完成诸如信息的查找和过滤、信息的视图定制以及很多自动操作。在下面的部分，我们将总结一下主体技术的特征，智能主体的一种分类结构，认识一些当前能够集成到新的或者现存的 DSS 应用中的智能主体技术。

到底什么是智能主体

在本书中我们已经看到将来组织中的决策制定和问题解决的复杂性不断增加，同时速度也在加快。但是，易用性却没能跟上这些大量可用的高级功能的增长步伐。结果就是 DSS 的应用变得越来越难于开始，甚至一些熟练用户都会知难而退。很多应该可用的信息及其处理仍旧无法使用。

在生活中，当我们发现自己处于一个时间和行为超越自己的环境时，我们往往寻求一种帮助，希望有人来处理我们能够做却不愿做的事情。在 DSS 的世界里，智能主体就实现了这种帮助的角色。IA 允许用户将他们的工作委托给主体软件。在重复性任务的自动化方面，在帮助用户记住关键日期或者事件方面，或者智能地总结复杂数据方面，智能

主体非常有用。更重要的是,就像人类一样,智能主体能够向用户学习,甚至对用户就特定的行为过程提出建议。

因为涉及很多边缘技术,人们对该概念的定义因各自的视角不同而不同。智能主体的世界已经萌发出了大量的早期定义和概念,每一个都对 IA 是什么或者应该是什么这些定义有其特有的说法。表 15-5 总结了当前一些智能主体的定义,以及它们的来源和基本假设。

表 15-5　对智能主体软件的当前定义

主体名称	主体定义
MuBot 主体	代表了两个相直交的概念:主体的自动执行能力和主体执行面向域的推理的能力。
AIMA 主体	任何可以通过感应器感知其环境并通过受动器在该环境上进行反应的事物。
Maes 主体	自动主体是计算机系统,它存在于一些复杂的动态环境中,并能在其中自动动作,由此实现一系列设计目标或者任务。
KidSim 主体	一种实现特殊目的的持久的软件实体。“持久”将主体从其他子程中区分出来,主体有它们自己的主意,它们知道如何完成任务,它们有自己的计划。“特殊目的”将它们和完整的多功能应用程序相区分,通常主体都比较小。
Hayes-Roth 主体	智能主体持续地执行如下三种功能:感知环境中的动态条件;采取行动影响环境中的条件;进行推理解释感知,解决问题,得出推论,并判断所要采取的行为。
IBM 主体	智能主体是软件实体,它代表一个用户或者其他程序执行一系列的有一定程度独立性或者自动性的操作,并由此获取一些知识或者对于用户目标与期望的表述。
Wooldridge-Jennings 主体	一个硬件或者更通常的说一个基于软件的计算机系统,具有如下的属性: • 自动性。主体在没有直接的人工或其他干预下运作,并对自己行为和内部状态有某种控制。 • 社交能力。主体要与其他主体(可能是人类)进行交互,这可以通过一种与主体交流的语言来进行。 • 反应能力。主体要能感知其所处环境(该环境可以是现实世界,一个通过图形化用户界面进行操作的用户,一系列的其他主体,因特网,或者上述各种的综合体),并能够很快做出反应以改变其中发生的事情。 • 主动性。主体不是简单地对其所处环境做出反应,它们要能够主动地采取面向目标的行为。
SodaBot 主体	软件主体是一种程序,它参与对话,谈判并协调信息的传递。
Foner 主体	在交流出现误解时,主体是合作的、自动的、值得信赖的。
Brustoloni 主体	自动主体是现实世界中具有自动能力的进行有目的的行为的系统。
FAQ 主体	主体具有自动性、面向目标性、协作性、灵活性、自启动性、持续性、特质性、交流性、适应性和活动性。

表 15-5 的各种定义中有两个隐含的共性:(1) 主体是指一个在做或者能做什么的东西;(2) 主体在得到另一个人允许的情况下代替该人行动。在这两个共性的基础上,Franklin 和 Graesser (1996) 提出了一个正式的智能主体软件的定义,体现了以前一些定义的本质内容,如下文所述:

一个自治的主体是置于一个环境的一部分中的系统,它能够感知所处环境,自动进行动作,并且随着时间推移,能够调整其行为,以期将来更好地实现其功能。

该定义涵盖足够广泛,避免了很多其他定义中的限制约束,同时提供了一个有益的界限,在其中可以继续探索智能主体软件的应用领域。在 DSS 中,我们认为 IA 的概念应该着眼于定义一个特定的机制来分析那些半结构化问题的系统,但是同时要有足够的可扩充性,在必须的情况下,允许对该技术采取创造性的、革新性的使用。在这种意义下,并非所有的程序都是一个主体。智能主体软件是一种计算机程序,但是一个计算机程序必须拥有上述几个特征才能被划分为主体。

智能主体软件的特征

表 15-6 列示了智能主体软件的各种特征,根据这些特征,我们就可以将其同别的基于计算机的程序区分开来。

表 15-6 智能主体软件的一般特征

自动性	协作能力	光荣降级
个性化	风险和信任共存	人格化
会谈能力	领域性	活动性

自动性

一个主体必须为其用户提供一定的自动功能。换句话说,一旦开始运行,它就要控制自己的行为。这样做的好处在于,当你委托主体做某些工作时,你一定希望它能够按照你的要求独立地进行工作,而不管外界发生什么。更高一点说,一个主体软件必须能够具有一定程度的并发执行和优先机制或者独立行为能力,以使其用户更加受益。从这方面看,IA 并不同于其他计算机程序,它们只是在直接操作下才进行反应,它们的功能不会理会其运行状态。

一个主体必须对相关事物作出反应。也就是一个 IA 必须能够感知它所处环境的变化并能及时作出反应。主体的这个特性也是委托处理和自动化的核心。就像你对你的助手所吩咐的,“当 x 发生时,做 y”,一个主体就总是等待 x 的发生。最后,为了达成用户的需要,所有的主体必须是持续运行的,甚至是用户不在了也要继续。

个性化

一个 IA 所有的着眼点在于帮助一个用户完成一项任务,比如信息收集或者数据分析,这比用户自己做要好一些。因为每一个决策者都有自己的特质,而且每个问题也都有其不同于其他的环境,所以一个 IA 必须能够就手中的问题可以进行教育,学会如何处理。个性化的特征在学习主体身上得以体现,它能够获取必需的信息,并且部分来自于对其用户的行为的观察。

会谈能力和协作能力

要保证一个 IA 能够和其用户共享日程计划，并能按照所期望的方式完成任务，进行某种形式的会谈或者双向反馈就成为必需的。这种会谈使得双方表达自己的意图，并能够获知对方的意图，通过这种反馈，就某事达成类似于合同的一致意见，以说明如何完成工作，由谁完成，何时完成。

除了和用户进行会谈或者反馈之外，IA 还需要唤醒一个或者更多的 IA 来帮助它完成一项任务。从这种意义上来说，一个 IA 必须具有在需要时同其他相关的 IA 相互作用或者进行交流的能力。进一步说，一些 IA 应该是“社会性”的。也就是说，它们相互作用或者同其他主体交流。这种交互可以是以个体的方式，也可通过某些通用的语言标准。

风险、信任和领域特性

一个智能主体的概念也就暗含了委托的概念。如果我们不信任要委托其帮助我们完成任务的那个机构，那我们就必须亲自完成这项任务。即便是我们对于这个主体有相当程度的信任，在把责任交给一个外部的主体的时候，我们还是面临了主体会犯错误的风险。使用智能主体软件，就要求在放弃人为控制的风险和我们对于科学技术的信任之间建立平衡。这个决策还要求使用者掌握合理的思维模式，不仅仅是知道主体会做什么（也是就我们对于它的信任度），还有前后关系和利益领域的问题（也就是智能主体犯错误时所牵涉的风险程度）。

了解 IA 存在和运作的领域，是 IA 应用成功的一个重要元素。如果该领域是一个本地化的决策模拟，则与应用主体相关的信用程度并不需要很高，与此相关的风险程度也相应的非常低。如果，与此相反，主体要应用在一个责任很高的领域，比如监测原子反应堆的控制棒的位置并在某种特定情况发生时通报其用户，其中对 IA 的信用程度就必须非常高，而使用主体的风险也就很重要了。在这些情况下，IA 必须同附加的控制机制结合使用，这些控制机制不仅要对 IA 进行支持，还要同时监测表现。

光荣的降级和协作

这个特征是将 IA 区别于一个简单的计算机程序的关键要素。这一概念同上面谈到的风险、信用和领域概念紧密相关。在主体和其用户之间的交流存在困难的时候（甚至在一方或者另一方没有认识到这种困难的时候），或者主体应用的领域并不适合（一方或双方实体可能都脱离了其存在要素而可能没有认识到），主体应该能够完成赋予它的部分任务，而不能简单地说无法运作。在适当使用 IA 的情况下，能够完成部分任务总比根本不能运作要产生较好的结果。

人性化

IA 的这个特征引起了很大争议。人性化是使一个非人类实体具有类似人类的某些特征。这在很多通常情况下都是显而易见的，比如说用阴性代词来描述一艘船（“她是一

艘不错的船”）或者说计算机是相互作用的社会实体（“我的系统感染了病毒并传染了整个实验室”）。拿 IA 来说，人性化是一个很常见的，但并非关键性的特征。一些主体努力使别人相信，它们表现为一个用户可见的或者可听到的实体，甚至某些具有感情或者个性特征。软件主体的设计通常表现出类似人类的趋势，尤其在可能与主体的主人之外的用户进行交互的情况下。一个在 IA 的设计方面体现人性化的例子就是“化身”的概念。

化身定义为一个有关人生的原则、态度、视角的体现或者形体表现。在虚拟现实环境（VRE）中，你的化身就是你的签名、你的商标和代表你的物理实体的符号。在虚拟现实环境中工作的 IA 用户可能有着上百个化身：一些是异质的，一些是象形的，一些是标新立异的，但是通常能够持久存续的化身是那些具有可识别的流行文化特征的——垃圾邮件罐，Winnie 的蔑视之声，或者森林泰山等。随着化身变得更加普通和复杂，它们有更多的能力来执行操作——模拟会谈、展示动态生成的行为——我们将越来越多地依赖于在线识别的可视化技巧。这里对设计提出的挑战是相当大的：要有战略性、可视化、概念化和精致性。

活动性

一些主体是活动的，在需要时，能够从一台机器到另一台机器，跨越不同的系统结构和系统平台。活动的主体能够走近它所要处理的数据，而且，这样可以不受网络延迟的影响。可转移的主体是能够在一个异构的网络中从机器到机器进行移植的程序。该程序选择何时向何处移植。它能够在任意点将其挂起执行，向另外机器转移，并在新的机器上继续执行。主体能够在每一台机器上执行任何复杂处理，以确保信息能够抵达其目的接受者。图 15-4 就图示了一个携带邮件信息的活动主体的例子。

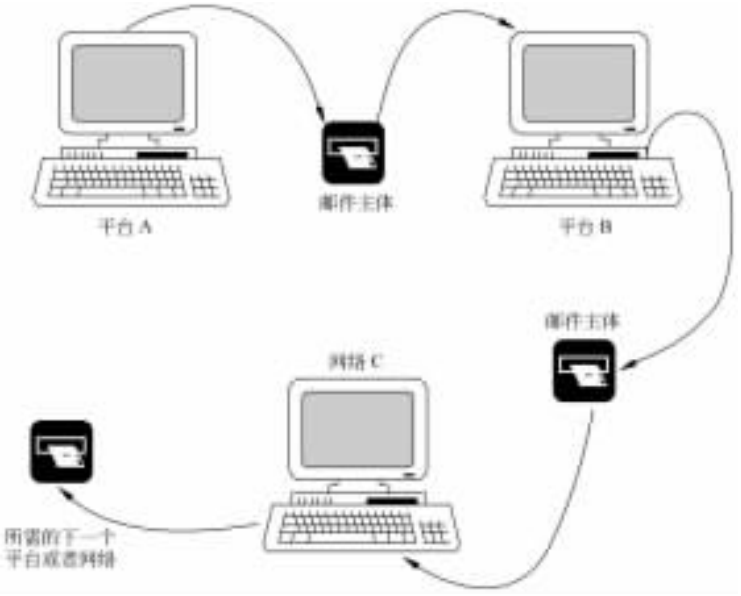


图 15-4　可移动的邮件主体的图示

可移动的主体比传统的客户机/服务器模型有如下一些优势：

- 效率。可移动的主体消耗更少的网络资源，因为它们将计算转移到数据处，而非将数据转移到计算处。
- 容错性。可移动的主体不需要计算机间的持续不断的连接。
- 方便示例。可移动的主体将交流通道隐藏，而不是计算的位置。
- 定制。可移动的主体允许客户端和服务器端通过相互规划来提高系统功能。

智能主体的分类

智能主体软件可以根据几个属性和特征子集进行分类。所有的主体都有活动性、自动性、面向目标性和现实连续性，其他的特征则可以用来进行分类，这种分类可以按照层次方式，也可以按照门类方式。

智能主体的层次分类

Franklin 和 Graesser (1996) 提出一种类似“生物学”的 IA 的分类结构，利用类似树形结构，“生物”位于根部，各种物种处于叶子节点。图 15-5 就展示了这种层次分类结构。

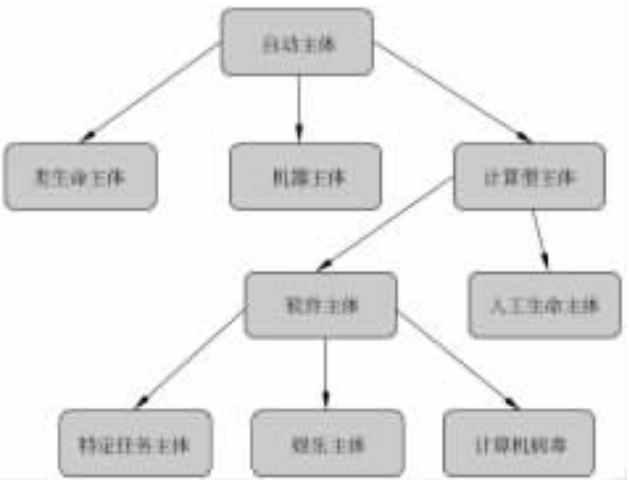


图 15-5　Franklin 和 Graesser 对智能主体的层级分类

智能主体的门类分类

IA 的另外一种分类方式是组织对个人的主体 (也指公共的或者共享的还是私人的主体)，其不同点是基于主体执行一个商务过程或者任务的目的是为一个组织应用的利益，还是为一个特定用户的利益。个人主体在当今的 DSS 环境中比较流行，但是组织主体的应用正在不断增加。日渐扩张的协作技术，比如 Lotus Notes，便使用了组织主体，来完成诸如邮件分类和过滤或者文档工作流管理等任务，这种技术的需求越来越大。

Brustoloni (1991) 提出了一种 IA 的分类方法，将主体分为三类：规则主体、计划主体和适应主体。表 15-7 提供了这种分类方法下每一类主体的简单描述。

表 15-7 Brustoloni 的软件主体的分类法

分 类	描 述
规则主体	对每个感应器的输入作出反应，并总能知道该如何处理。这种主体无需进行计划和学习。
计划主体	使用基于案例的方式或者随机算法进行计划。这种主体也无需学习。
适应主体	这种主体能够在执行计划的同时进行学习。

Lee，Cheung，Kao (1997) 等还提出了另一种主体分类方法，根据它们的智能级别和能力进行分类。表 15-8 列示了他们的这种分类方法。

表 15-8 Lee，Cheung，Kao 根据智能级别进行软件主体分类的方法

智能级别	描 述
级别 0	主体在直接命令下为用户提取文档。用户必须确切地指明所需文档及其存储位置。比如：网络浏览器。
级别 1	主体采取一种用户启动的搜索行为。主体能够在索引信息上匹配用户提供的关键字。比如：Internet 搜索主体 (例如 Yahoo，Alta Vista 等等)。
级别 2	主体为其用户维护档案。主体能够监测各种信息来源，并在找到相关信息或者发现新信息时通知用户。比如：网络监视器。
级别 3	主体能够学习并从用户制定的档案中进行推理，以辅助生成一个查询或者目标搜索。比如：DiffAgent。

智能主体能够解决的问题类型

为了增加我们对 IA 技术的理解，我们可以检查一下智能主体能够帮助我们解决的各种实际问题。这一部分所要讨论的问题并非全部，但是很能说明在 IA 应用中典型的问题。

考虑如下背景：Corinne 在信息处理上遇到了困难。她是一个快速成长的小的行业咨询公司的研究人员，这个工作非常激动人心也具有很高的挑战性。她经常需要收集有关公司客户的各种各样的广泛信息，并且要时刻密切关注与公司目前工作项目相关的新的进展。尽管 Corinne 并不厌烦去猎取项目相关的各种信息，她还是经常发现自己陷入数据的陷阱，不得不辛苦地去寻找她所谓的“信息库”。另一个更大的问题是她不得不时刻关注着重要的事件。她知道一定有更好的办法去完成这样的工作，也许她需要一些帮助了。

Corinne 可以应用几个 IA 来帮助完成其日常工作，从而获得巨大的裨益。她日常的那些辛苦地寻找信息库的工作，当信息数量不多时可以持续地进行，但是当数据量不断增大，新的数据源源不断地到达时，这件任务就很快变得难以为继了。智能主体能够用来帮助解决这种问题，它通过按照 Corinne 的喜好定制化信息，从而节约她处理和存储信息的时间。

那么如何解决她需要时刻关注新事件的问题？这些事件可能是现实世界中的事件，比如竞争宣告或者紧急状况，也可以是计算机事件，比如新版本的文档准备就绪。不管事件的本质如何，智能主体能够通过持续地监视各种信息源来帮助解决这种问题，它检查那

些标志着新事件的关键字来实现，其中这些关键字是 Corinne 的某个客户所关心的或者是与公司的项目有关的。当一个有意义的事件发生时，主体可以通知 Corinne 到某个特定地址下载相应的文档，或者安排同某些人进行会谈，或者其他什么。

IA 在当前的组织环境中得到了广泛的应用。设想在一个客户帮助窗口，IA 能够利用客户问题中的基本参数到数据库中提取相关的技术文档，然后交给客户支持人员使用。其关键的商业价值在于，委托智能主体来处理这些以及很多其他的问题，你便能够适时地得到适当的信息，而无需完全由自己去处理或者重写基本的应用。

智能软件主体的前景

在未来的十年，智能软件主体很可能会成为 DSS 领域最为重要的进步之一。预期在 2000 年时，基本上所有重要的基于计算机的应用都会有某种形态的主体使用（Janca 1996）。这个推论有如下佐证：

1. 桌面应用程序变得越来越功能多样。因此，用户仅能够掌握其中的很小一部分功能。IA 能够屏蔽新的应用程序的复杂性，并帮助用户完成他想做的事情。
2. 信息的来源和信息容量正在飞速增长。IA 能够帮助进行数据挖掘，还能帮助找到最有价值的数据宝库。
3. 在现存决策支持中，从用户的鼻尖到计算机屏幕之间存在着一个带宽的限制。越大的计算带宽就意味着更多的数据更快地涌现给用户，但是用户仍旧只有有限的认知处理能力以及有限的工作时间。IA 能够 24 小时工作并能帮助管理数据流，它能够对数据进行过滤从而只提交用户认为重要的那些信息。
4. 客户端计算机和服务器已经变得足够强大以支持多个 IA 的应用，使得应用和处理更加容易。
5. 因特网和 WWW 应用的快速发展产生了一个更加复杂的计算环境。这个新的环境通常称为“网络计算”，即指我们正在从一个简单连接的世界（如：终端到主机或者客户端到服务器）转移到一个更加复杂连接的世界，这里有多重的服务器和服务相互交叠就像一个高速公路网络。在这个新环境下，没有什么不可得到的，但却每一分每一秒都在发生着变化，决策者不得不解决如何找到所求这样一个问题。IA 则能够从这样一个看似乱糟糟的随机世界中创建一个相互联系的整体。
6. 学术和商业界对 IA 研究都超越了那种纯理论的范畴，如今试验性的系统已经应用在了因特网上。
7. 大的客户应用系统，诸如 Lotus Notes，便在很大程度上应用了主体，这已经超越了试验的界限。

在本书写作时，已经有超过 50 家的商业供应商提供应用主体的软件和服务。他们的产品涵盖了几乎所有的应用领域，包括因特网应用。当前的挑战是如何使得为任何应用添加智能主体变得容易，不管它们是新开发的抑或已经存在的。重要的一点是，IA 已经不是一个将来的概念，它们已经在今天广泛使用以简化我们的应用程序的运行，并且其用处也在快速增长。

15.4 小　　结

创造性、直觉性和革新性是我们需要完全理解的决策制定过程中的因素。而更重要的是，我们要认识到它们对有效的解决方案的贡献，并将其应用于我们的决策支持机制中来强化这种贡献。通过研究，我们知道人类的创新性能够由基于计算机的系统进行支持，而智能主体的使用则不仅扩展了我们的视野，而且把我们从众多任务中解脱出来，从而可以将更多时间用在解决方案的创新性上，也使得直觉性能够透入到流程中。

复习题

1. 什么是创新性？试着从不同的角度进行解释。
2. 解释逻辑思维的基本概念。
3. 描述横向思维的基础。
4. 列出并简单描述推动横向思维的三个主要行为。
5. 采用什么方法可以刺激横向思维中的变化？
6. 列出自由联想技术的主要原则。
7. 简单描述问题解决中结构化关联技术的主要概念。
8. 描述问题解决中记名式的团组献策技术的主要假设。使用该技术有什么好处？
9. 什么是智能主体？
10. 找出并讨论智能主体的两个主要特征。
11. 什么是学习主体？
12. 解释为什么 IA 的光荣降级非常重要。
13. 可移动主体的对于传统的客户机/服务器模型的优势是什么？
14. 列出并简单描述 IA 分类的方法。

讨论题

1. 讨论创新性和直觉在决策制定中的重要性。
2. 研究一个可以应用关键思维方法解决问题的案例（比如帕累托规则）。描述它如何应用在问题解决的过程中。
3. 在一个你所熟悉的组织中找一个发生的问题。简单描述该问题。使用头脑风暴的技术来找出该问题的解决方法。
4. 使用结构化关联技术来解决上一题中找到的问题。

5. 分析一个市场上的智能主体软件。概括其功能。讨论它所能解决的问题。
6. 讨论为什么说智能主体技术的用户要维持 IA 使用的风险和用户对 IA 的信任之间的平衡。
7. 在 WWW 上有很多关于智能主体的信息。找到一些讨论 IA 使用的网址。提示：Firefly，IBM 或者 BargainFinder。

第 16 章

21 世纪的决策支持

学习目标

- 了解决策支持系统的未来发展
- 了解专家系统和人工智能系统的未来发展
- 了解经理信息系统的未来发展

未来计算机的重量不会超过 1.5 吨。

——“对科学和创新上持续进展的预测”,《大众机械》,1949 年

我把这个国家走了一遍并且和最优秀的人交谈过,我可以向你保证数据处理不过是一种时尚,人们对它的热衷不会超过一年。

——Prentice Hall 出版公司商业书籍主管编辑,1957 年

我们正处于我们旅程的结尾(至少对于现在来说是这样)。不过这仅仅意味着,你只是对现有的决策支持技术的应用有了充分的了解,并能够在你的管理活动中有效地利用它们。然而旅程仍将继续,因为甚至就在你阅读本文的时候,新的、具有创新意义的决策支持系统技术仍在不断出现,并且需要管理者们接受、消化和吸收到他们的世界中去。这是我们需要处理的首要问题之一。

在管理者中有一种常见现象,即他们不太愿意直接面对组织所面临的主要的、长期的挑战——这种挑战源自新出现的技术自身强大而令人敬畏的属性。事实已经证明,对未来技术的漠然处之将会付出高昂的代价。从长期来看,新技术的作用将会上升并占据主导地位。这种创新将会导致新颖的产品和业务过程的产生,有时甚至会给整个行业大换血或者产生出全新的企业模式来。跟上持续前进的新技术发展步伐,是决策制定者们在新的世纪中将要面对的一项具有里程碑性质的任务。如果组织想要从新的决策支持技术所提供的强大能力中获益,那么组织管理者们就需要抓住新技术出现所带来的机遇,并同时注意防范随之而来的某些负面作用。

最后一章的目的是对未来的决策制定以及与决策支持系统相关的技术作一个展望。请相信这是一个艰难的任务。只需回顾我们近来在好莱坞电影上看到的大量的带有未来风格的技术,就可以理解这项任务的庞杂。某些未来模式的原型已经逐渐地具体化,但未来的主体模式还有待研究领域的先驱者们和发明家们去追寻。这正是由 Scientific American的总编辑 John Rennie 所描述的一个基本问题的结果,即“大多数技术预言……是过分单纯化的,因此,也是不现实的。”他进一步指出,外部事物,诸如市场力量、政府政策、社会经济条件、变化无常的风格时尚、甚至于人类社会自身的反复无常,都会对一项创新的技术是否能够开花结果并且进入你我的日常生活产生影响。这个重要而富有洞察力的观点不应被那些决策制定者们遗忘掉。

在本章中,作者的意图并不是要去预测那些不可预测的东西,而是要着重于观察那些在决策支持技术方面出现的趋向和创新,并探究这些创新给我们在思考和解决问题上可能带来的变化。需要着重指出,本章不会对未来的决策支持系统进行详尽无遗的浏览,而是要试图引导你们每一个人去探究你们在下一个千年中确实要经历到的那些可能性。

16.1　我们在哪里?我们已经去过哪里?

我想计算机的世界市场的容量为 5 台。

——汤姆斯·沃特森(Thomas Watson),IBM 董事长,1943 年

640K 的容量对于任何人都应该足够了。

——比尔·盖茨对个人计算机内存容量的评论,1981 年

如果不对事物现有状态有一些基础性的认识,那么对未来的观察也不会取得成功。因此,我们对未来的预测首先从对那些决策支持系统已经有效集成并成功应用于现代组织的多个领域的观察开始。

决策支持系统已经成功地用于解决各种各样的问题。决策支持系统在其中发挥重要作用的应用领域包括城市交通管理、保健及医疗诊断、环境规划、农林业等等。下面的章节仅对决策支持系统应用的大量且重要领域中的很少一部分做进一步的描述。

环境决策制定及影响评估

环境决策支持系统 (EDSS) 已经开发并可用于评估自然资源开发的影响。还有几个可用的商业系统已经用于环境监测和影响评估。一个主要的应用是在水资源管理领域上。自 20 世纪 80 年代末以来,用于支持水库决策分析的决策支持系统已经为多个组织 (包括政府的和商业的) 所开发,支持水资源传输维持与计划的工具也已投入使用。可以辅助决策制定者们对复杂灌溉系统做计划的决策支持系统也已得到开发。

农业

开发灌溉系统的首要目的是为了在缺乏自然水利灌溉的地方提高农作物产量。然而,灌溉管理也仅是为决策支持系统技术所支持的与农业有关的应用之一。着重于全面提高农业生产进程的决策支持系统,连同辅助庄稼和农业牲畜的疾病诊断和治疗决策的决策支持系统都已经被开发出来。这些系统的使用对世界粮食资源做出了重大贡献,并在第三世界的农业发展中扮演了重要角色。

林业

已经开发了几种可用于辅助与林业以及与自然资源相关的决策制定的决策支持系统。例如,一套名为 TEAMS 的决策支持系统被用来作为辅助森林管理人员开发特定地点处理日程表的战术计划系统。再生林问题以及与森林管理相关的财政问题也得到特定领域的决策支持系统的辅助。最近,已开发出了一个嵌入了线性规划模型的决策支持系统,用来评估被政府私有化的新西兰森林资源。

制造业

在工业和制造型企业中做出恰当的投资决定是一件困难的事情。决策支持系统已经成为工业投资决策领域普遍存在的元素。更进一步,在行业内广泛采用了分析和计划技术——如 MRP (物料需求计划) 和 MRP II (制造资源计划) ——的环境中,决策支持系统也能发挥作用。数控机器工具仅仅是基于计算机的决策支持和人工智能系统在制造业中得到成功应用的一个领域。

医学

第 8 章中曾提到决策支持系统和专家系统技术,例如 MYCIN 和 DENDRAL,已经在医学专业中发挥了多年的重要作用。最近,用于支持医院服务计划,如病人接收和职员预测

等的决策支持系统被开发来,以适应这些服务的重要性及由于成本增长所带来的管理需求。当前大量的医疗决策制定得到了决策支持系统技术的支持。例如,一个用于辅助食品的适当营养平衡开发的决策支持系统已经得到了市场商业化。医学专业领域运用到决策支持系统的其他用途包括热带疾病诊断、传染病模拟以及药效交互影响的设想等。

组织支持

我们可以找到许多决策支持系统在支持组织常见的决策制定行为上的应用。在这些应用中的特例包括辅助评估国际投资风险的决策支持系统,为与资产和债务管理有关的决策提供向导的决策支持系统,以及参与组织投资战略和资产预算决策开发的决策支持系统。一般认为,使用决策支持系统可以通过增强组织决策制定者们对相关财务数据形象化的能力,从而提高决策制定者们的决策能力。

这里对现有决策支持系统应用领域的列举并不完全。然而这个列举反映了决策支持系统技术在实际应用中的规模和重要性。下一节将开始对未来决策支持系统的探讨,再下一节将着重于 IT 相关技术的未来:专家系统及经理信息系统①。

16.2 决策支持系统的未来

但是……它有什么好处吗?

——IBM 高级计算机系统部的工程师对微型芯片的疑问,1968 年

这种名叫"电话"的东西毛病太多,无法认真地把它作为一种通信工具来看待,这种设备对我们一点价值都没有。

——西盟 (Western Union) 的内部备忘录,1876 年

未来决策支持系统的一个最终任务,就是要应用信息技术去增强信息工人们在新的"信息时代"中处理复杂性和结构各异的种种问题时的生产效率。Alvin Toffler (1970) 警告说这场即将到来的信息洪水是如此严峻,以至人们将会变得瘫痪并且无法进行选择。信息上溢是一个真实的问题:据估计一个典型现代组织所收集数据的数量每年都在翻倍。如果一个典型的知识型工人只能有效地处理这其中 5% 的数据,那么在决策制定领域提供支持的需要就显得是从未这么明确了。在这点上,明天的决策支持系统设计者和使用者们将面临大量的挑战。

决策支持系统集成

正如第 14 章中所讨论的,决策支持系统的使用中面对的一个主要挑战是一种集成——必须有一个贯穿整个组织的通用对话界面,以使用户们能够方便和有效地访问所有可用的信息资源。决策支持系统和各类可操作数据存储之间的连接窗口应该是透明的,并应允许各种结构和平台的集成,以使各种分离的资源可存储于其上。在这点上,应该使图形用户界面 (GUI) 标准化,以使所有的应用和数据兼容并易于访问。这种类型的

① 本节所提出的许多预言建立在 Kersten 和 Meister (1993) 及 Dobrzenieki (1994) 的工作基础之上。

标准化也是全世界都愿意接受微软的 Windows 及其相关产品的主要原因。然而,在决策支持系统开发过程中的一个主要问题,是缺乏适当的 GUI 来将各种结构和平台集成在一起。从典型的决策制定者角度来看,GUI 可以是一个决策支持系统在实施到工作环境中时究竟是成功还是失败的决定性因素。

在这点上,信息表示的概念必须被决策支持系统设计者和使用者们重视。会需要什么形式的信息表示来面对明天的问题和决策呢?我们是否仍将依靠传统的图形图表等报告格式呢?我们是否需要新的图形类型,例如支持多维信息表示的动态图形模型和报告呢?

将专家系统和其他人工智能应用软件的部件与组织型决策支持系统集成,已经成为近期的一个主要焦点。知识库成为一种数据库或者模型的形式,接口引擎可看成知识库管理系统,专家系统的语言系统现在也已经有效地成为决策支持系统的对话系统的一部分。未来的挑战之一是要更紧密地将决策支持系统和组织应用相集成,从而更进一步地提高决策支持系统和决策制定者的"智商"水平。

决策支持系统的连通性

连通性正在成为下一代决策支持系统极其重要的一个属性,尤其在相互之间的互联通信上。尽管我们当前通过局域网、广域网以及网关连接桌面客户的能力是朝这方面迈出的重要的第一步,然而接下来面临的挑战显然是要将通信协议、通信渠道、带宽以及数据传输率标准化,以便利不断增长的对大型数据集、图形数据库、数字图像以及视频进行交互的需要。

对于组织来说,系统能够连接到其他网络的能力是很重要的,因为这样可以为组织的资源提供遍及全球的不同来源。然而这方面的主要缺点在于其潜在的安全性问题,这个问题主要是由于通过不安全的网关开放网络联接而造成的。减小这种潜在安全问题的一个办法是将通信协议标准化,使得这些通信协议之间的差异最小化。

决策支持系统的文档处理能力

除了处理组织的和外部的数据之外,增强决策支持系统在访问和管理组织文档资源方面的能力,对于未来决策支持系统的力量和效率来说是必要的。新的搜索和构建技术,例如概念检索、超文本、以及多媒体等正在研究领域和商业领域中努力开发着,目的在于创建有效的使能技术。可能这种努力的一个最伟大表现,是最近出现并正在萌芽成长阶段的群件产品,如 Lotus Notes。世界正在飞快地进入互联时代,而这方面努力的结果也很好地预示了下一代决策支持工具的出现。

对知识恰当的正确表达将使系统能够更好地模仿专家,并且这种能力将会转变成具有竞争性的赢利——尤其当专家系统在边际利润非常高的关键领域中使用时。使用快速查询算法在广泛和不同的来源中查询访问文档,也有助于提高组织的竞争力。确保通讯手段及时升级以使决策支持系统能够与因特网和 WWW 连接,将使得组织能够跟上在决策制定上出现的各种新技术和创新,并因此保持它们在正在出现的电子化市场中的竞争优势。

决策支持环境中的虚拟现实

虚拟现实 (VR) 技术可以使个体在模拟的环境中活灵活现。这个计算机图形的产物将越来越依赖于日益复杂的硬件和软件程序,并使得模拟出来的环境变得更加“真实”。VR 所展示的环境涵盖了从在大型屏幕上观看的戏剧类型到电脑显示器所显示的内容到装备了立体视屏和头戴式电话的头盔等多个方面。VR 对人的刺激是多感官的:视觉、听觉和触觉是最常见的类型。这种技术有利于观察者们理解前后关联的元素中的相互关系,并提供了一种手段,使得观者可以学会有效地对有用的数据做出反应。

影响范围

有人预言 VR 将在诸如制造业、教育及培训、药物干预、军事准备以及娱乐等多个不同的领域产生戏剧化的影响。例如,使用计算机辅助设计 (CAD) 的工程师将能够对他们的设计进行交互以便对设计的缺陷做出早期检测,而这些缺陷在现有技术下不等到原型构建完成是无法明显看出的。VR 对教育的影响将是非常重大的。人们在学习中更倾向于实验性的练习而不是死记硬背。VR 提供了这样的工具。许多练习都可以通过 VR 有效地进行。基于 VR 的研究和开发的潜在的高经济回报鼓舞了企业家们的行动,诸如波音和 AT&T 等的巨头已经为此投资了数百万美元。

关注领域

VR 环境如此诱人以至会使观察者们沉迷其中。而一个负面的效果是某些使用者可能因此无法区别现实和虚拟现实,从而导致一些可能会对自身或者别人造成危险的行为。类似的,VR 还可以用来操纵那些使用者。1991 年由美国副总统戈尔主持的国会听证会的总结认为,美国在 VR 上的投资还不充分。VR 的发展早已被称作“计算机显然的命运”,但是如何控制和操纵它们的问题可能会导致对计算机空间道德法规的需求。

16.3　专家系统和人工智能系统的未来

所有能发明的都已经发明了。

——Charles H. Duell,美国专利局局长,1899 年

Goddard 教授不知道行为和反应之间的关系,也不知道应该用某些比空洞的言谈更好的东西来回应社会。他显然不是专家,并且看起来缺乏所有中学生日常应有的基本知识。

——纽约 Times 杂志关于 Robert Goddard 具有革命性的火箭研究工作的社论,1921 年

关于专家系统的未来有两个首要问题必须指出:(1) 未来人工智能对智能数据库的需求;(2) 知识管理。

人工智能对智能数据库系统的需求

专家系统在决策制定领域的进步,将需要非常重视可以包含和管理人工智能代理的数据库系统的发展。智能数据库系统 (IDS) 的概念可以很好地通过对一个典型的保健系统的类比来阐述。在这个领域里,保健的提供者,如医生、护士以及药剂师可认为类似于软件智能代理,每个人都拥有独立的和共享的知识以及推理能力,并被认为将会给病人提供最好的保健服务。然而,他们所能付出的关护程度,却经常依赖于他们相互交互与合作过程中所遇到的权威、角色、责任等多种因素所构成的复杂关系的影响。换句话说,为这些医护工作者所拥有的知识只能有效地适用于保健环境自身定义的规章和制度之中。

将同样的逻辑应用在专家系统和智能软件代理领域,两个或更多个代理相互合作以达到一个共同目标的能力引发了一个"互用性"问题。IDS 的出现将为互用性提供系统支持 (管理与访问共享的和私有的目标)。此类系统的开发将包括集成带有数据库数据模型的人工智能知识表达和推理,并经处理生成超目标模型。那时,在这个领域中的问题或者说任务,将是通过代理利用可得资源来协同执行以决定最佳解决方案。

尽管专家系统的智能软件代理概念很有吸引力,然而要在多个领域中做这项工作仍然要求系统拥有各种类的知识,并且其推理能力也要比现有的系统更强大。知识和推理能力将根据所需的知识和推理的属性和数量,系统的代理中的分布和共享程度,以及获取、增加和学习与领域相关知识的方式及其准确性和完整性等来进行特性化。对大量不同的持久、共享、分布的知识及相关推理的数据类型进行表达、建模、控制和管理,显然对智能数据库系统提出了重大挑战。

对这个挑战的一项有前途的回应是当前正在开发之中的高性能存储系统 (HPSS),HPSS 是一个软件,为超大型存储环境提供了分级存储管理和服务。这项技术将在那些对总存储容量、文件大小、数据速度、对象存储数量以及用户数量等有要求的环境,如那些有当前或未来规模需求的大型规则集中发挥特殊的作用。HPSS 的一个中心技术目标,就是要使大型文件在存储设备和并行或簇集计算机中的移动,在速度上要比现在的商用存储系统软件产品快许多倍,并比现有的系统工作起来更可靠、更易于管理。HPSS 是在起领先地位的美国政府超级计算机实验室和工业界合作努力的结果,它着重于面对非常现实、非常紧迫的高端存储需求。

知识管理

由于预期未来专家系统在规模和有用性两方面都会扩大,有关对其维护和管理的问题渐渐开始提出。问题的中心在于,如何提升收集和获取专家知识的方法,并确保知识库包含最当前的可用信息。当前正着重于对几个可能为专家系统扩展和获取信息的方法进行研究。第一个方法涉及到使用在传统程序中用到的软件工程来维持知识库。第二种方法把对规则的支持嵌入一个有效系统以维持大型的关系数据库。然而,当需要在几百万条数据中计算上百万条的规则时,就会引发比例缩放的问题。为了克服这里出现的比例缩放问题,设计系统时必须使规则支持机制紧密地耦合在 DBMS 中,使得规则能够有效地在大型应用、集成控制、参考完整性、转换约束及保护中使用。

第三种可能方法是使用便于描述的规则说明,使专家系统实施者可拥有一些抽象的标准格式,这些标准可以细化和定义来自于知识库的回答类型。这个方法的问题在于,需要找到建立好的便于描述并且程序上有效的知识表达。最后还有一个着重于知识的深度和宽度的方法。专家系统的深度知识库的出现,增加了解决复杂的搜索问题的可能性。一个拥有广泛(全面)知识并结合了大量各种特殊概念和情况的系统,可以很容易地为多重问题的概括和类推提供支持。这样的专家系统在合适的配置步骤和使用之下,应该能够从事于不同的项目领域,理解类似于英语的程序指令,并可能学会增强其自身的问题解决能力。

知识获取和表达上的进步

为了和专家系统的智能部分连接,还必须要注意到未来的知识获取和表达。从第 8 章和第 9 章中我们知道,为专家行为建模是一件复杂且经常相当困难的任务。不是所有的知识都能够通过制造规则来获取 知识也可以通过因果信息或者数学关系而存在。21 世纪专家系统知识获取的最终设计目标,是使系统允许专家直接将他们的知识编码到计算机中去,到那时就可以在知识获取阶段中取消掉知识工程师这个角色。然而,要在最近的将来实现这个目标是困难的,因为当前对知识工程师的理解认为其所发挥的作用更多的是一门艺术而不是科学。现阶段的知识转换主要是询问专家,使其反思决策制定的过程,这种知识的传输过程依然需要基于感觉、触觉、记忆以及感受等手段将专家的经验进行系统化和整理。我们目前还不能够建立一个足以处理这些复杂概念的系统,但可能很快就可以做到这一点。

然而,在近期的未来,对靠知识工程师来将专家知识编码成机器语言的过程的重视程度,依然会不断提高,因为当前的处理手段仍然是在尽可能地模仿专家的经验。在这点上我们需要回答很多问题,如 :专家和专家系统直接交互是否比通过知识工程师进行的交互更加有效?我们能否获取并表达能够在许多不同问题领域都能够适用的知识?多个来源的知识能否被获取并合并到一个专家系统中,使得它们能够在相关问题领域上与别的专家系统工具交互?最后,我们能否使用诸如机器学习这样的技术使知识获取过程自动化呢?

16.4　经理信息系统的未来

迄今为止经理信息系统的进展是依靠持续成功地从大型机系统中移植出来。这使得执行者及高级管理者们没有必要再去学习不同的计算机操作系统,此外还彻底降低了 EIS 的实施成本。

这种计算机系统占地面积的"小型化"并不意味着对未来的 EIS 不会去满足处理能力上的更多需求。未来的经理信息系统将基于超级个人计算机,这种计算机将是动态组织的,并将大量各类现有和未来的软件应用程序包集成在一个有用的 EIS 中。此外,未来的 EIS 将不仅仅提供一个支持高级管理者的系统,而且还要满足中层管理人员的信息需求。在这个意思上,明天的 EIS 也许称为组织信息系统(OIS)更为恰当。

几种新出现的功能在未来的 OIS 中将会非常常见。最近新增加的一个是将一个执行

可视化信息访问及分析（VIAA）的模块包含入内。VIAA 模块通过结合了文字、数字数据、图形数据和图像的可视化屏幕，提供了对内部组织和外部信息的访问功能。可视化屏幕为使用者提供了对数据快速方便的访问，提高了他们的决策制定能力。

一个包含了 VIAA 模块的 OIS 可以看成是第二代的经理信息系统，它利用了近来在个人计算机或工作站上显著增长的信息处理速度和能力。不过，典型的 EIS 和 OIS 在几个方面存在着差别：

- 最初的 EIS 是为高级经理设计的，而 OIS 则面向组织中各个层次的应用。
- 即使是现在的 EIS 也是典型地运行在封闭的、有所有权限制的系统上，这种系统对用户的输出依赖于薄弱的组织系统集成部件。例如，整个系统可能有很优秀的图形处理能力并且基于强大的第四代语言，但却被联接到一个或多个遗留数据库系统上。由于制造者们开始提供将数据从较老的遗留应用软件和数据库上转换过来的解决方案，用户将能够把那些来自不同软件供应商的产品“一把抓”，来建立一个完整的 OIS 解决方案。这种方法将允许基于独立终端用户或者组织的需要将“最优品种”的系统部件集成到一个客户化的产品中去。
- 许多老的 EIS 系统仍然只允许用户访问预处理过的数据。OIS 将使用户能够立即访问实时数据，不论它在哪里。基于诸如动态数据交换（DDE）之类当前标准的数据变换上的进步，使得 OIS 应用软件能够在数据或图像发生变化时自动做出反应。这将使得组织中所有层次的使用者能够快速地访问到绝大部分最新可用的数据。

许多 EIS 软件提供商已经将 VIAA 包含在他们下一代产品中。最新开发的 EIS 软件响应了它们用户的需要，使系统能够为一个公司中许多不同的管理层次访问到。为了增加他们的用户基础，软件开发商已经开始意识到访问已存储数据的能力和将其他软件中的最佳特色包含到自身软件上的重要性。随着新产品的开发和用户基础扩展到一个公司中的多个管理层次，EIS 和其他计算机辅助分析工具，如决策支持系统等的界限将会变得模糊和窄小。

将所有组织应用软件和技术集成的趋势，使经理信息系统的未来变得非常光明。在计算机系统和电信领域技术和理论上的长足进步，预示着实现具备真正功能的 OIS 系统为期不远了。

16.5 对未来决策支持系统技术的一些最后思考

于是我们到 Atari 公司去说：“嗨，我们有能令人吃惊的好东西，它甚至用到了你们的一些部件，你们能不能给我们投资呢？或者我们把它给你，我们只是想做它，付我们薪水就行了。”但他们说“不”。于是我们又去了惠普公司，但他们说：“嗨，我们不需要你，你甚至还没有大学毕业呢。”

——Steve Jobs，苹果计算机公司创始人

我们不喜欢他们的声音，况且吉他音乐就要过时了。

——Decca 录音公司拒绝甲克虫乐队，1962 年

依赖技术的行为,如决策制定,如果不同步地去观察其所依赖的技术的未来发展,是很难对其自身的未来做出预想的。最后一节将关注几个关于未来计算技术及它们与决策支持的关系的问题。

谁将带领我们进入未来

1996 年 1 月,一个为顶尖的 IT 供应商举办的高层会议召开。150 多名应邀出席者都是来自 IBM,微软,Compaq,Intel,AT&T,Sony 及 NEC 等公司在管理、市场、销售领域中的高级经理。

在会议上,与会者们使用一种类似于"立即投票"技术的群件,为他们在关于各种行业问题上的观点进行投票。答案最具吸引力的两个问题如下:

1. 你认为谁是 2000 年 IT 行业中最具影响力的供应商?
2. 你认为谁将是 2005 年最具影响力的供应商?

毫无疑问,大多数与会者认为 2000 年 IT 行业中最具影响力的供应商是微软公司。毕竟,那时离世纪之交只剩三年半的时间。出席者普遍地认为:不论微软如何对由于因特网的出现而引发的难以计数的全球变化作出响应,由于它现在的强劲市场地位所带来的剩余效应,仍然会对很大部分市场产生影响。然而,最有意思的答案是与会者们对关于 2005 年问题的普遍回应:得票最高者是"一个尚未存在的供应商"。

如果对第二个问题的回应似乎是出乎预料之外的,那么的确是如此。自从个人计算机出现以后,计算机行业中谁能占据领导地位的动力原则已经变得不稳定了。现在无处不在的微软(可以说是今天计算机行业中最有影响力的一员)在 PC 机出现以前并不存在,而且,今天美国计算机供应商排名前 20 位中的 7 位在 PC 机出现以前也是没有的。如果因特网像现在一样持续地影响着全球市场,在这种情况下首席供应商的更换是很有可能发生的。

当你更仔细地观察,就会发现这些参与会议的高层人士对这些问题的回应几乎可以说是极具洞察力的。投票的结果说明,有相当多的计算机行业领导者们已经清楚地看到由因特网带来的这些变化,并感受到了他们的地位在即将到来的增长和变化着的时代中很可能受到冲击。

硅谷革命

今天,技术变革背后的一个主要驱动力量,是由于半导体工业研究和开发所带来的处理器运行速度和处理能力的快速进步。当前最先进的硅芯片比一个人指甲还要小,但可容纳一幅完整、详细的美国交通地图,包括每一个州、每一条街道、甚至每一条小巷。而且,这个硅芯片拥有在电子"高速公路"上以百万兆的速度运行的能力。

今天的微处理器比以往制造的更加强劲,下一年将要生产出来的微处理器在能力方面更是无疑会大大压倒现在的微处理器。硅谷技术力量每 18 个月就会翻倍。这个观察结论首先由 Intel 的合伙创始人戈登·摩尔于 30 年前提出,现在已经被广泛接受并称为"摩尔定律"。基于这个定律,我们可以有把握地预计到 2004 年硅谷生产的芯片可以包含超过 10 亿个晶体管。一块拥有这样容量的芯片就能够满足 42 个中央办公电话交换机

的工作需要！

1992 年，半导体行业协会资助开发和出版了半导体技术的“国家道路地图”。这个道路地图的实质是一系列以三年为间隔的项目，每一代项目中的设备都成功地实现小型化并且更加强劲，每一代记忆设备在密度上要比上一代大四倍左右。与今天 RAM 存储芯片 16MB 的存储容量和 CPU 300MHZ 的运行速度相比，估计到了 2007 年，典型的 RAM 存储芯片将包含 16GB 的存储容量，典型的 CPU 芯片则可以运行在 1000MHZ 的速度之上。

带宽几乎要免费了

与硅谷技术在存储容量和处理能力上同步前进的，是各种通信媒介容量上的迅速增长，这些媒介包括玻璃光纤、铜线以及无线通信系统等。富士通公司及其他地方的科学家已经证实，能够在一根直径只有人的头发大小的玻璃导线中以每秒万亿比特的速度传输数据。按照这个速度，从《纽约时代周刊》第一期发行起的各期中的所有文字可以在不到一秒的时间内传送到一个新的地方。

随着带宽的增加，发送信息的成本也成比例地下降。一些人认为未来的通信费用将会便宜到可以忽略不计。一些考虑到未来的团体已经摆出了一种激烈的姿态来确保它们参与到这场通信革命中来。例如，肯塔基州格拉斯哥的所有居民可以以 200 万比特的速度访问因特网，而每月的平均费用为 11.45 美元。这项服务由格拉斯哥的电力公司提供，该公司是当地的一家市政企业，其服务已从提供电力扩展到提供有线电视和宽带数字通信服务。美国的电力公司已经安装了很多光纤缆线，所以如果它们选择进入该服务领域，它们就有能力成为全美第二大的电信服务提供者。

网络能力也增强了

计算机和带宽的先进技术结合起来给数字化飓风注入了能源：使因特网得以迅速发展。因特网是网络的网络——一个从底层构建起来的动态的、巨大的通信系统。该网络的所有参加者都遵循一个简单的协议集，该协议集定义了数据如何格式化以及如何从一个地方路由到另一个地方。遵循这些简单规则的结果是，因特网具有展示几乎不可想像的复杂行为的能力，包括它可以不断增长而不会被自己的重量压垮的能力。现在因特网每年在规模上都要翻番。家庭、学校、企业、图书馆以及博物馆都已连接到网络上，每个新增加的连接都增加了整个网络的价值。这种增值性首先由以太网的发明者 Bob Metcalfe 表述，他观察认为网络的能量以网络用户数的平方数增长着。这个表述现在被称为 Metcalfe 定律，它和摩尔定律一起组成了我们正在经历的通信革命的基础。

万维网（WWW）对决策制定的影响可能是意义深远的。不像规模每年扩大一倍的互联网，WWW 的规模几乎每 90 天就增加一倍。1996 年，美国邮政服务投递了超过 1850 亿封一级邮件。而在同一年因特网上传发的电子邮件则超过了 1 万亿。万维网以甚至在几年前都无法预想到的方式使信息的发布和访问变得如此民主化。正如 FCC 主席 Reed Hundt 所说：“通信时代已经和自印刷机发明以来的信息历史上最伟大的革命连接在一起了。”一些人认为工业革命使生产率增长了 50 倍。自微处理器发明以来的 25 年中，计算机的运算能力提高了 1000 多倍，这几乎等于每年进行一次工业革命！而且现在还在

进行！

梦想正在变成现实

技术专家们总是喜欢展望未来并梦想着明天将会发生的事情。决策制定领域的变化速度，和其相关技术以及环绕它的世界的变化速度一样的快，而且，因为这种关系，我们可以观察技术变化的速度，并用它作为在预测下一个世纪问题如何解决的一个基础。那些曾经的梦想将成为未来清楚的现实。让我们展望在 2000 年及以后可能会影响到决策制定的一些预期中的技术发展吧。

到 2000 年，预计国际象棋的冠军将会是计算机。国际象棋这些年来成为了对人类未来的推理和策略谋划能力的一个衡量。虽然 IBM 最近在本届国际象棋比赛中对加里·卡斯帕罗夫取得了有限的胜利之后，宣称不再想继续开发会下国际象棋的计算机，但是近期在决策支持运算法则上的进展将极有可能导致一场计算机化的国际象棋比赛。

到 2005 年，可翻译电话机将会出现，它将允许全球任何地方的两个人进行通话，即使他们的语言不同也没有问题。同样的核心技术也将用于创造实用的语音—文本转换器，以将语音转换成可视的文字显示。电话将可以通过一种智能电话应答器进行自动答复，这种应答器能够简短地和拨入方通话以确定呼叫的性质和优先级。

在下个世纪的头十年中，计算机将主宰着教育界。课件将具有足够的智能来理解和纠正任一个特定学生的概念化模型中的错误。多媒体技术将允许学生与所模拟的系统以及他们正在学习的过去历史上的名人进行交互。同样的智能技术将可用来创造人类生物化学模拟器，这将有助于设计和测试先进的药品和药剂。在 2020 年和 2070 年之间的某个时候，将会有一台计算机通过图灵测试，这将标志着机器拥有人类级智能的时代的到来[①]。

16.6　小　　结

本章意在刺激读者们的胃口而不仅仅是满足它。我们不会再进入信息时代——因为我们已经在信息时代之中了。未来的知识管理和决策代表了伴随着这些冒险而来的新的视野和挑战。21 世纪管理的目标将从对企业生产率的关注转向对企业知识生产率的关注。Drucker 认为 21 世纪中组织的关键资源是知识，确实如此。世界的企业组织将在知识竞争的运动场上竞争，胜利的管理者将是那些能够驾驭技术的力量来为需要解决的复杂问题提供支持的人。挑战是属于你的。

---

① 图灵测试由 Alan M. Turing (1912 - 1954) 发明，它作为测试计算机显示的类人智能的程度。一个讯问者通过一个终端与一个人和一台机器连接，因此，他们相互无法看到。讯问者的任务是通过向双方提问题来确定哪个是机器哪个是人。如果在一定的时间内讯问者无法做出区分 (Turing 建议是五分钟，但一般认为与精确时间无关)，那么机器将被认为是有智能的。

附　录

决策风格测试[①]

决策风格测试说明：

1. 请使用（只能使用）如下数字回答各个问题：
   8　如果问题非常符合您的实际情况
   4　如果问题基本符合您的实际情况
   2　如果问题稍微符合您的实际情况
   1　如果问题最不符合您的实际情况
2. 请将上述数字填写到各个问题下面的方框里。
3. 在每一道题中，只能使用数字 8、4、2、1 来作回答。
4. 比如，回答某题的答案可以为：8 1 4 2。
5. 注意，在一道问题中，一个数字只能使用一次。
6. 在回答问题的过程中，设想一下在各个特定的环境中，您一般会采取什么样的行动。
7. 在回答问题的时候，请根据您的第一直觉进行选择。
8. 回答问题没有时间的限制，也不存在正确或者错误的答案。您甚至可以改变您的主意。
9. 您的回答反映了您对问题的感受，表示您倾向于采用的行动，并不表示您认为应该做的正确的事情。

	Ⅰ	Ⅱ	Ⅲ	Ⅳ
1. 我的首要目标是：	拥有一定的社会地位	成为我的工作领域中的知名人物	让我的工作得到认可	获得工作中的安全感
	□	□	□	□
2. 我喜欢的工作是：	技术的，事先规定好的	具有一定的变动性	允许独立行动的	与人有关的工作
	□	□	□	□

① **资料来源：**Decision Style Inventory III and Scoring Information adapted from Rowe, A. J. and Boulgarides, J. D., (1994), Managerial Decision Making, Englewood Cliffs, NJ: Prentice-Hall. Used by permission, © 1985.

续表

	Ⅰ	Ⅱ	Ⅲ	Ⅳ
3. 我希望我的手下：	效率高,速度快	能力极强	具有责任心	乐于接受别人的建议
	□	□	□	□
4. 在我的工作中,我追求：	可行的结果	最好的解决方案	新方法、新点子	良好的工作环境
	□	□	□	□
5. 我最善于与他人沟通的途径是：	面对面沟通方式	书面沟通方式	通过小组讨论沟通	在正式会议中沟通
	□	□	□	□
6. 在我的计划中,我强调：	目前的问题	实现目标	未来的目标	发展员工的事业
	□	□	□	□
7. 在面临有待解决的问题时,我：	我依赖于已经经过证实的方法	进行仔细的分析	寻找具有创新性的方法	仅仅根据我自己的感觉
	□	□	□	□
8. 利用信息时,我倾向于：	具体的事实	精确的、完整的数据	广泛考虑多种观点	限于那些易于理解的数据
	□	□	□	□
9. 在我不确定该怎么做时,我通常：	根据直觉采取行动	寻找事实依据	寻找可能的折中方案	在制定决策之前一直等待
	□	□	□	□
10. 如果可能,我会避免：	长久的争论	半途而废的工作	使用数字或者公式	与他人发生冲突
	□	□	□	□
11. 我尤其擅长于：	牢记数据和事实	解决困难的问题	洞察多种可能方案	与他人打交道
	□	□	□	□
12. 时间非常重要时,我通常：	快速决策并做出反应	遵循计划和优先考虑事项	拒不受到压力	寻找向导和支持
	□	□	□	□
13. 在社会交际场所,我通常：	与他人交谈	思考该说些什么话	观察发生的事情	倾听别人的谈话
	□	□	□	□
14. 我善于记住：	别人的姓名	我们相遇的地点	别人的长相	别人的个性特点
	□	□	□	□
15. 我的工作给了我：	影响别人的权力	有挑战性的任务	实现我的个人目标	受到群体的接受和认可
	□	□	□	□
16. 与我相处得好的人是：	精力旺盛的和雄心勃勃的	自信的	头脑开放的	有礼貌的,值得信赖的
	□	□	□	□

续表

	Ⅰ	Ⅱ	Ⅲ	Ⅳ
17. 面临压力时，我通常：	变得很焦虑	集中精力思考问题	变得很沮丧	变得很健忘
	□	□	□	□
18. 别人认为我：	有进取心	有修养	具有想像力	能给予他人帮助
	□	□	□	□
19. 我的决策通常是：	现实的和直接的	有系统性的，或者抽象的	广泛的和灵活的	对别人的需求非常敏感的
	□	□	□	□
20. 我不喜欢：	失去控制	无趣的工作	遵循规则	受到抵制
	□	□	□	□

决策风格测试的评分标准：

1. 将每一列（Ⅰ、Ⅱ、Ⅲ和Ⅳ）的数字加总。
2. 计算上述四个数值的总和，其结果应该为 300。如果您的总和不等于 300，请检查计算是否正确，答题是否符合前面的说明。
3. 将您的得分放到下面相应的方框（Ⅰ、Ⅱ、Ⅲ和Ⅳ）中。
4. 参考第 2 章，具体了解四个像限中决策风格的特性。

分析型Ⅱ	概念型Ⅲ
指令型Ⅰ	行为型Ⅳ

参考文献

第1章

1. Alter，S. L. 1980. Decision Support Systems：Current Practices and Continuing Challenges. Reading，MA；Addison-Wesley.
2. Anthony，R. N. 1965. Planning and Control Systems：A Framework for Analysis. Boston：Harvard University Graduate School of Business Administration.
3. Donovan，J. J，and S. E. Madnick. 1977. Institutional and Ad Hoc Decision Support Systems and Their Effective Use. MIT Working Paper No. C15R-27，Cambridge，MA.
4. Gorry，G. A.，and M. S. Scott Morton. 1989. “A Framework for Management Information Systems.” Sloan Management Review 13 (1)：49～62.
5. Little，J. D. 1970. “Models and Managers：The Concept of a Decision Calculus.” Management Science 16 (8)：B466～85.
6. Silver，M. S. 1991. “Decisional Guidance for Computer-Based Decision Support.” MIS Quarterly 15 (1)：105～22.
7. Simon，H. A. 1960. The New Science of Management Decision. New York：Harper & Row.

第2章

1. Barnard，C. T. 1938. The Functions of the Executive. Cambridge，MA：Harvard University Press.
2. Miller，G. A. 1956. “The Magical Number Seven，Plus or Minus Two：Some Limits on Our Capability for Processing Information.” Psychological Review 63 (2)：81～97.
3. Myers，I. B. 1962. Manual for the Myers-Briggs Type Indicator. Princeton，NJ：Educational Testing Services.
4. Rowe，A. J.，and J. D. Boulgarides. 1994. Managerial Decision Making. Englewood Cliffs，NJ：Prentice Hall.

第3章

1. Adams，J. L. 1979. Conceptual Blockbusting：A Guide to Better Ideas. 2d ed. Stanford，CA：Stanford Alumni Association.
2. Beach，L. R.，and T. R. Mitchell. 1978. “A Contingency Model for the Selection of Decision Strategies.” academy of Management Review (July)：439～50.
3. Clemen，R. T. 1991. Making Hard Decisions：An Introduction to Decision Analysis. Belmont，CA：Duxhury Press.
4. Clough，D. J. 1984. Decisions in Public and Private Sectors. Englewood Cliffs，NJ：Prentice Hall.
5. Delbecq，A. L. 1967. “The Management of Decision-Making within the Firm：Three Strategies for Three Types of Decision-Making.” Academy of Management Journal.
6. Etzioni，A. 1967. “Mixed-Scanning：A ‘Third’ Approach to Decision Making.” Public Administration Review 27 (5)：385～92.

7. Geoffrion, A. M., and T J. Van Roy. 1979. "Caution: Common Sense Planning Methods Can Be Hazardous to Your Corporate Health." Sloan Management Review 20 (4): 31.
8. Harrison, E. F. 1995. The Managerial Decision-Making Process. Boston: Houghton Mifflin Co.
9. Katona, G. 1953. "Rational Behavior and Economic Behavior." Psychological Review 60: 313.
10. Keen, P. G., and M. S. Scott Morton. 1978. Decision Support Systems: An Organizational Perspective. Reading, MA: Addison-Wesley.
11. Lindblom, C. E. 1959. "The Science of 'Muddling Through.'" Public Administration Review 19 (spring): 113 ~ 27.
12. Miller, G. A. 1956. "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capability for Processing Information." Psychology Review 63 (2): 81 ~ 97.
13. Mintzberg, H. 1973. The Nature of Managerial Work. Englewood Cliffs, NJ: Prentice Hall.
14. Saaty, R. W. 1987. "The Analytic Hierarchy Process—What It Is and How It Is Used." Mathematical Modeling 9 (3 ~ 5): 161 ~ 76.
15. Simon, H. A. 1960. The New Science of Management Decision. New York: Harper & Row.
16. Sprague, R. H., Jr., and E. D. Carlson. 1982. Building Effective Decision Support Systems. Engtewood Cliffs, NJ: Prentice Hall.
17. Thompson, J. D. 1967. Organizations in Action. New York: McGraw Hill.
18. Tversky, A. 1972. "Elimination by Aspects: A Theory of Choice." Psychological Review 79: 281 ~ 99.
19. ——. 1973. "Availability: A Heuristic for Judging Frequency and Probability." Cognitive Psychology 5: 207 ~ 32.
20. Tversky, A., and D. Kahneman. 1974. "Judgment under Uncertainty: Heuristics and Biases." Science 185: 1124 ~ 31.
21. Zanakis, S. H., and J. R. Evans. 1981. "Heuristic Optimization: Why, When, and How to Use It." Interfaces 11 (5): 84 ~ 92.

第 4 章

1. Bowers, D. G. and S. L. Seashore. 1966. "Predicting Organizational Effectiveness with Four Factor Theory of Leadership." Administrative Science Quarterly 11 (September): 238 ~ 63.
2. Culnan, M. J., and B. Gutek. 1989. Why Organizations Collect and Store Information. In Information Systems and Decision Processes, edited by E. A. Stohr and B. R. Konsynski. Los Alamitos, CA: IEEE Computer Society Press.
3. Deal, T. E., and A. A. Kennedy. 1982. Corporate Cultures. Reading, MA: Addison-Wesley.
4. Gore. W. J. 1964. Administrative Decision Making: A Heuristic Model. New York: Wiley.
5. Hamson, E. F. 1995. The Managerial Decision-Making Process. Boston: Houghton Mifflin Co.
6. Mintzberg, H. 1985. "The Organization as a Political Arena." Journal of Management Studies 21 (2): 133 ~ 55.
7. Pascale, R. T., and A. G. Athos. 1981. The Art of Japanese Management. New York: Simon and Schuster.
8. Patz, A. E., and A. J. Rowe. 1977, Management Control and Decision System. New York: Wiley.
9. Peter, L. J., and R. Hull. 1969. The Peter Principle. New York: Morrow.
10. Peters. T. J., and R. H. Waterman. Jr. 1982. In Search of Excellence. New York: Harper & Row.

11. Robbins, S. P. 1990. Organization Theory: Structure, Design, and Applications. 3d ed. Englewood Cliffs, NJ: Prentice Hall.
12. Schein, E. H. 1985. Organizational Culture and Leadership. San Francisco: Jossey-Bass.
13. Stohr, E. D., and B. R. Konsynski. eds. 1992. Information Systems and Decision Processes. Los Alamitos. CA: IEEE Computer Society Press.
14. Swanson, E. B.. and R. W. Zmud. 1989. ODSS Concepts and Architecture. In Information Systems and Decision Processes, edited by E. A. Stohr and B. R. Konsynski. Los Alamitos, CA: IEEE Computer Society Press.
15. Zateznick, A. 1970. "Power and Politics in Organizational Life." Harvard Business Review (May—June): 101 ~ 09.

第5章

1. Beyth-Marom, R. 1982. "How Probable is Probable? A Numerical Translation of Verbal Probability Expressions." Journal of Forecasting 1: 257 ~ 69.
2. Clemen, R. T. 1991. Making Hard Decisions: An Introduction to Decision Analysis. Belmont, CA: Duxbury Press.
3. Davis, M. W. 1988. Applied Decision Support. Englewood Cliffs, NJ: Prentice Hall.
4. De Finetti, B. 1964. La Prevision: Ses Lois Logiques, Ses Sources Subjectives. Translated by H. E. Kyburg. In Studies in Subjective Probability, edited by H. E. Kyburg, Jr., and H. E. Smokier (New York: Wiley). First Published in Annales de l'Institut Henri Poincare 7 (1937): 1 ~ 68.
5. Golub, A. L. 1997. Decision Analysis: An Integrated Approach. New York: Wiley.
6. Howard. R. A. 1988. "Decision Analysis: Practice and Promise." Management Science 34 (6): 679 ~ 96.
7. Howard, R. A., and J. E. Matheson. 1968. An introduction to Decision Analysis. In Readings on the Principles and Applications of Decision Analysis. Stanford, CA: Decision Analysis Group, SRI International.
8. Savage, L. J. 1954. The Foundations of Statistics. New York: Wiley.
9. Watson, S. R., and D. M. Buede. 1987. Decision Synthesis: The Principles and Practice of Decision Analysis. Cambridge: Cambridge University Press.

第6章

1. Bavelas, A. 1948. "A Mathematical Model for Group Structures." Applied Anthropology 7: 16 ~ 30.
2. Coleman. D., and R. Khanna. 1995. Groupware: Technology and Applications. Upper Saddle River, NJ: Prentice Hall.
3. DeSanctis, G., and R. B. Gallupe. 1987. "A Foundation for the Study of Group Decision Support Systems." Management Science 33 (5): 589 ~ 609.
4. Ellis. C. A., S. J. Gibbs, and G. L. Rein. 1991. "Groupware: Some Issues and Experiences." Communications of the ACM 34 (1): 38 ~ 59.
5. Fellers. J, W. 1987. "Skills and Techniques for Knowledge Acquisition." Indiana University Working Paper Series.
6. George. J. F. 1991. "The Conceptualization and Development of Organizational Decision Support Systems." Journal of Management Information Systems 8 (3): 109 ~ 25.
7. Gray. P. 1981. The SMU Decision Room Project. In Transactions of the First International Conference on

Decision Support Systems, edited by D. Young and P. G. W. Keen. Providence. RI: the Institute of Management Sciences.
8. Hamilton. J., S. Baker, and B. Vlasic. 1996. "The New Workplace." Business Week, 29 April.
9. Harrison, E. F. 1986. Policy, Strategy, and Managerial Action. Boston: Houghton Mifflin Co.
10. ——. 1995. The Managerial Decision-Making Process. Boston: Houghlon Mifflin Co.
11. Holsapple. C. W. 1991. "Decision Support in Multiparticipant Decision Making." Journal of Computer Information Systems (summer): 37 ~ 45.
12. Homans, G. C. 1950. The Human Group. New York: Harcourt, Brace & Co.
13. Huber, G. P., J. S. Valacich, and L. M. Jessup. 1993. A Theory of the Effects of Group Support Systems on an Organization's Nature and Decisions. In Group Decision Support Systems: New Perspectives, edited by L. M. Jessup and J. S. Valacich. New York: Macmillan.
14. Janis, I. L. 1982. Groupthink. 2d ed. Boston: Houg. hton Mifflin Co.
15. Kraemer. K. F., and J. L. King. 1988. "Computer-Based Systems for Cooperative Work and Group Decision Making." ACM Computing Surveys 20 (2): 115 ~ 46.
16. Kunz. W., and H. Rittel. 1970. Issues as Elements of Information Systems. Working Paper No. 131. University of California-Berkeley. Center for Planning and Development Research.
17. Leavitt. H. 1951. "Some Effects of Certain Communication Patterns on Group Performance." Journal of Abnormal and Social Psychology 46.
18. Lindstone, H., and M. Turoff. 1975. The Delphi Method: Technology and Applications. Reading, MA: Addison-Wesley.
19. Marakas. G. M., and J. T. Wu. 1997. A Taxonomy of Multiparticipant Decision Structures. University of Maryland Working Paper Series in Information Systems.
20. Nunamaker, J, F., Jr., A. R. Dennis, J. S. Valacich, D. R. Vogel, and J. F. George. 1993. Group Support Systems Research: Experience from the Lab and the Field. In Group Decision Support Systems: New Perspectives, edited by L. Jessup and J. Valacich. New York: Macmillan.
21. Shull et al. 1970. Organizational Decision Making.
22. Van de Ven, A. H., and A. L. Delbecq. 1971. "Nominal Versus Interacting Group Processes for Committee Decision Making." Academy of Management Journal 14: 129 ~ 39.
23. Vogei, D. R., J. F. Nunamaker, Jr., W. B. Martz, Jr., R. Grohowski, and C. McGoff. 1989. "Electronic Meeting Systems Experience at IBM." Journal of Management Information Systems 6 (3): 25 ~ 43.
24. Vroom, V. H., and P. W. Yetton. 1973. Leadership Behavior on Standardized Cases. Technical Report No. 3. New Haven, CT: Yale University Press.

第 7 章

1. Crockett, F. 1992. "Revitalizing Executive Information Systems." Sloan Management Review 38 (summer): 17 ~ 29.
2. Dobrzeniecki, A. 1994. Executive Information Systems. IIT Research Institute White Paper Series.
3. Jones, J. W., and R. McLeod. 1986. "The Structure of Executive Information Systems: An Exploratory Analysis." Decision Sciences 17 (2): 220 ~ 49.
4. Millet, I., and C. H. Mawhinney. 1992. "Executive Information Systems—A Critical Perspective." Infor-

mation and Management 23 (1): 83 ~ 93.
5. Peters T., and R. Waterman. 1982. In Search of Excellence. New York: Harper & Row.
6. Rockart, J. F. 1979. "Chief Executives Define Their Own Data Needs." Harvard Business Review (March-April).
7. Rockart, J. E., and M. E. Treacy. 1982. "The CEO Goes On-Line." Harvard Business Review 60 (1): 82 ~ 87.
8. Watson, H. J., G. Houdeshel, and R. K. Rainer, Jr. 1997. Building Executive Information Systems and Other Decision Support Applications. New York: Wiley.
9. Watson, H. J., R. K. Rainer, and C. Koh. 1991. "Executive Information Systems: A Framework for Development and a Survey of Current Practices." MIS Quarterly 15 (1): 13 ~ 31.
10. Watson, H. J., S. Singh, and D. Holmes. 1995. "Development Practices for Executive Information Systems: Findings of a Field Study." Decision Support Systems 14 (2): 171 ~ 85.

第 8 章

1. Berry, D. C., and D. E. Broadbent. 1987. "Expert Systems and the Man-Machine Interface Part Two: The User Interface." Expert Systems 4 (1): 18 ~ 28.
2. Gerrity. T. P., Jr. 1971. "The Design of Man-Machine Decision Systems: An Application to Portfolio Management." Sloan Management Review 12 (2): 59 ~ 75.
3. Hayes-Roth, F., D. A. Waterman, and D. Lenat, eds. 1983. Building Expert Systems. Reading, MA: Addison-Wesley.
4. Keen, P. G., and M. S. Scott Morton. 1978. Decision Support Systems; An Organizational Perspective. Reading, MA: Addison-Wesley.
5. Klein, M. R., and L. B. Methlie. 1995. Knowledge-Based Decision Support Systems. 2d ed. New York: Wiley.
6. Liebowitz, J., 1990. The Dynamics of Decision Support Systems and Expert Systems. Chicago: Dryden Press.
7. Minsky, M. 1975. A Framework for Representing Knowledge. In The Psychology of Computer Vision, edited by P. Winston. New York: McGraw-Hill.
8. Rich, E., and K. Knight. 1991. Artificial Intelligence. New York: McGraw-Hill.

第 9 章

1. Drucker, P. F. 1993. Post-Capitalist Society. New York: Harper Business.
2. Ericsson, K. A., and H. A. Simon. 1984. Protocol Analysis: Verbal Reports as Data. Cambridge, MA: MIT Press.
3. Feigenbaum, E. 1977. "The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering." In Proceedings of the Fifth International Joint Conference on Artificial Intelligence, pp. 1014 ~ 29. Cambridge. MA: MIT Press.
4. Gaines, B. R. 1995. The Collective Stance in Modeling Expertise in Individuals and Organizations. Knowledge Science Institute University of Calgary. <http://ksi.cpsc.ucalgary.ca/articles/Collective/> (accessed January 1998).
5. Gaines, B. R., and M. L. G. Shaw. 1995. Knowledge Acquisition Tools Based on Personal Construct Psy-

chology. Knowledge Science Institute. University of Calgary. < http：//ksi. cpsc. ucalgary. ca/articles/KBS/KER/ > (accessed January 1998).
6. Hayes-Roth, B. 1985. "A Blackboard Architecture for Control." Artificial Intelligence 26：251 ~ 321.
7. Holsapple, C. W., and A. B. Whinston. 1988. The Information Jungle. Homewood, IL：Dow Jones-Irwin.
8. Kelly, G. 1955. The Psychology of Personal Constructs. New York：Norton.
9. Kim, J., and J. F. Courtney. 1988. "A Survey of Knowledge Acquisition Techniques and Their Relevance to Managerial Problem Domains." Decision Support Systems 4 (October)：269 ~ 85.
10. Klein, M. R,, and L. B. Methlie. 1995. Knowledge-Based Decision Support Systems. 2d ed. New York：Wiley.
11. Marakas, G. M., and J. J. Elam. 1997. "Creativity Enhancement in Problem Solving：Through Software or Process?" Management Science 43 (8)：1136 ~ 46.
12. Marcot, B. 1987. "Testing Your Knowledge Base." AI Expert (August).
13. Newell A. 1982. "The Knowledge Level." Artificial Intelligence 18：87 ~ 127.
14. Petrie, C., ed. 1992. Enterprise Integration Modeling. Cambridge, MA：MIT Press.
15. van Lohuizen, C. W. W. 1986. "Knowledge Management and Policy making." Knowledge Creation, Diffusion, Utilization 8 (1).

第 10 章

1. Brule, J. F. 1985. Fuzzy Systems—A Tutorial. < http：//www. austinlinks. com/Fuzzy/tutorial. html > (accessed January 1998).
2. Dhar, V., and R. Stein. 1997. Intelligent Decision Support Methods. Upper Saddle River, NJ：Prentice Hall.
3. NIBS Pte, Ltd. 1996. Neuro Forecaster Genetica Software Package. < http：//www. singapore. com/products/nfga > (accessed November 1997).
4. Wayner, P. 1991. "Genetic Algorithms." Byte, January, 361 ~ 64.

第 11 章

1. APT Data Group. 1996. Briefing Paper：What is Metadata? < http：//www. computerwire. com/bullelinsuk/212e_la6. htm > (accessed October 1997).
2. Bischoff, J., and T. Alexander. 1997. Data Warehouse：Practical Advice from the Experts. Upper Saddle River, NJ：Prentice Hall.
3. Inmon. W. H. 1992a. Building the Data Warehouse. New York：Wiley.
4. ——. W. H. 1992b. "EIS and the Data Warehouse." Database Programming and Design (November).
5. Kelly, S. 1994. Data Warehousing：The Route to Mass Customization. New York：Wiley.
6. Kozar, D. 1997. The Seven Deadly Sins. In Data Warehouse：Practical Advice from the Experts, by J. Bischoff and T. Alexander. Upper Saddle River, NJ：Prentice Hall.

第 12 章

1. Codd, E. F., S. B. Codd, and T. S. Clynch. 1993. Beyond Decision Support. Computerworld, 26 July.
2. Gray, P. 1997. The New DSS：Data Warehouses, OLAP, MDD, and KDD. In Proceedings of the Americas

Conference on Information Systems, edited by J. M. Carey, pp. 917 ~ 19. Phoenix, AZ: Association for Information Systems.

3. Inmon, W. H., J. D. Welch, and K. L. Glassey. 1997. Managing the Data Warehouse. New York: Wiley.

第 13 章

1. Alavi, M., and I. Weiss. 1986. "Managing the Risks Associated with End-User Computing." Journal of MIS (winter).
2. Arinza, B. 1991. "A Contingency Model of DSS Development Methodology." Journal of MIS (summer).
3. Blanning, R. W. 1979. "The Functions of a Decision Support System." Information and Management 2 (September): 71 ~ 96.
4. Courbon, J. C., J. Grajew, and J. Tolovi, Jr. 1980. "Design and Implementation of Decision Support Systems." 14.
5. Galletta, D. F., K. S. Hartzel, S. E. Johnson, J. L. Joseph, and S. Rustagi. 1996. "Spreadsheet Presentation and Error Detection: An Experimental Study." Journal of MIS 13 (3): 45 ~ 63.
6. Henderson, J. C., and R. S. Ingraham. 1982. Prototyping for DSS: A Critical Appraisal. In Decision Support Systems, edited by M. Ginzberg et al. New York: North-Holland.
7. Holsapple, C. W., S. Park, and A. B. Whinston. 1993. Framework for DSS Interface Development. In Recent Developments in Decision Support Systems, edited by C. Holsapple and A. Whinston. Berlin: Springer-Verlag.
8. Janvrin, D., and J. Morrison. 1996. Factors Influencing Risks and Outcomes in End-User Development. In Proceedings of the Twenty-ninth Annual HICSS Conference, vol. 2. Los Alamitos, CA: IEEE Computer Society Press.
9. Lotfi, V. and C. Pegels. 1996. Decision Support Systems for Operations Management and Management Science. 3d ed. Homewood, IL: Irwin.
10. Moore, J. H., and M. G. Chang. 1983. Meta-Design Considerations in Building DSS. In Building Decision Support Systems, edited by J. L. Bennett. Reading, MA: Addison-Wesley.
11. Saxena, K. B. C. 1992. DSS Development Methodologies: A Comparative Review. In Proceedings of the Twenty-fifth Annual Hawaii International Conference on System Sciences. Los Alamitos, CA: IEEE Computer Society Press.
12. Sprague, R. H., Jr., and E. C. Carlson. 1982. Building Effective Decision Support Systems. Englewood Cliffs, NJ: Prentice Hall.
13. Walls, J., and E. Turban. 1992. "Selecting Controls for Computer-Based Information Systems." Accounting Management Information Systems (October).

第 14 章

1. Alter, S. L. 1980. Decision Support Systems: Current Practices and Continuing Challenges. Reading, MA: Addison-Wesley.
2. Boehm, B. W., J. R. Brown, H. Kaspar, M. Lipow, G. McLeod, and M. Merritt. 1978. Characteristics of Software Quality. Amsterdam: North-Holland.
3. Davis, F. 1989. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Tech-

nology. ” MIS Quarterly, 13 (3) : 3i9 ~ 42.
4. Dickson, G. , and R. Powers. 1973. MIS Project Management : Myths, Opinions, and Realities. In Information System Administration, edited by F. W. McFarlan et al. New York : Holt, Rinehart & Winston.
5. Ginzberg, M. J. 1975. A Process Approach to Management Science Implementation. Ph. D. diss. Massachusetts Institute of Technology.
6. ——. 1976. A Study of the Implementation Process. Paper presented at Implementation II : An International Conference on the Implementation of Management Science in Social Organizations, University of Pittsburgh.
7. Ives, B. , M. H. Olson, and J. J. Baroudi. 1983. “The Measurement of User Information Satisfaction. ” Communications of the ACM 26 (10) : 785 ~ 93.
8. Klein, M. R. , and L. B. Methtie. 1995. Knowledge-Based Decision Support Systems. 2d ed. New York : Wiley.
9. Kolb, D. A. , and A. L. Frohman. 1970. “An Organization Development Approach to Consulting. ” Sloan Management Review 12 (4) : 51 ~ 65.
10. Marakas, G. M. , and S. Hornik. 1996. “Passive Resistance Misuse : Overt Support and Covert Recalcitrance in IS Implementation. ” European Journal of Information Systems 5 : 208 ~ 19.
11. Meador, C. L. , M. J. Guyote, and P. G. W. Keen. 1984. “Setting Priorities for DSS Development. ” MIS Quarterly 8 (2) .
12. Min. H. , and S. B. Eom. 1994. “An Integrated Decision Support System For Global Logistics. ” International Journal of Physical Distribution and Logistics Management 24 (1) : 3 ~ 23.
13. Schein, E. H. 1956. “The Chinese Indoctrination Program for Prisoners of War. ” Psychiatry 19 : 149 ~ 72.
14. Zand, D. E. , and R. E. Sorenson. 1975. “Theory of Change and the Effective Use of Management Science. ” Administrative Science Quarterly 20 (4) : 532 ~ 95.

第 15 章

1. Brustoloni, J. C. 1991. Autonomous Agents : Characterization and Requirements. Carnegie-Mellon Technical Report CMU-CS-91-204. Pittsburgh : Carnegie Mellon University.
2. de Bono, E. 1970. Lateral Thinking : Creativity Step by Step. New York : Harper & Row.
3. Delbecq, A. L. , A. H. Van de Ven, and D. H. Gustafson. Group Techniques for Program Planning. Glenview, IL : Scott, Foresman.
4. Franklin S. , and A. Graesser. 1996. Is It an Agent, or Just a Program ?: A Taxonomy for Autonomous Agents. In Proceedings of the Third International Workshop on Agent Theories, Architectures, and languages. Springer-Veriag.
5. Janca, P. 1996. Intelligent Agents : Technology and Application. Norwell, MA : GiGa Information Group.
6. Koberg, D. , and J. Bagnall. 1976. The Universal Traveler. Los Altos, CA : William Kaufmann.
7. Lee, J. K. W. , D. W. Cheung, and B. Kao. 1997. Intelligent Agents for Matching Information Providers and Consumers of the World Wide Web. In Proceedings of the Thirtieth Hawaii International Conference on Systems Sciences. Los Alamitos, CA : IEEE Computer Society Press.
8. Marakas, G. M. , and J. J. Elam. 1997. “Creativity Enhancement in Problem Solving : Through Software or Process ?” Management Science 43 (8) : 1136 ~ 46.
9. Osborn, A. F. 1963. Applied Imagination. 3d ed. New York : Scribner.

10. Rowe，A. J.，and J. D. Boulgarides. 1994. Managerial Decision Making. Englewood Cliffs，NJ：Prentice Hall.
11. VanGundy，A. B. Techniques of Structured Problem Solving. 2d ed. New York：Van Nostrand Reinhold.

第 16 章

1. Dobrzeniecki，A. 1994. Executive Information Systems. IIT Research Institute White Paper Series. <http：//mtiac. hq. iitri. com/MTIAC/pubs/eis/eis. txt>.
2. Hundt，R. 1997. “The Internet：From Here to Ubiquity.” IEEE Computer 30 (10)：122～24.
3. Kersten，G. E.，and D. B. Meister. 1993. Decision Support Systems：An Overview of Research Directions and Applications. <http：//www. business. carleton. ca/～gregory/papers/idrc_report_93/>.
4. Toffler，A. 1970. Future Shock. New York：Random House.