



原著
译

〔美国〕
沈锦泉

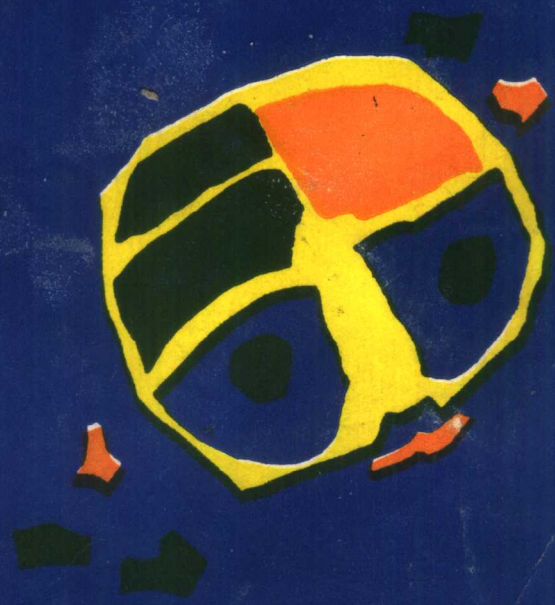
D·W·罗尔斯顿
袁天鑫

葛自良

吴修敬

人工智能与专家系统开发原理

上海交通大学出版社



人工智能与专家系统开发原理

【美】D·W·罗尔斯顿 著

沈锦泉 袁天鑫 葛自良 吴修敬 译

上海交通大学出版社

(沪)新登字 205 号

内 容 提 要

本书主要介绍专家系统的概念与方法,并对与此有关的人工智能技术也作了扼要的说明。全书分为三大部分:第一章到第五章给出了专家系统的综合介绍,比较详细地说明了专家系统中知识的作用、专家系统程序设计语言及其开发过程;第六章到第十一章讲述了专家系统的主要方面——知识获取、不确定性处理、系统体系结构、解释系统和消解原理等等,叙述范围十分广泛,并尽可能地提出了许多可供选择的观点;第十二章比较详细地介绍了一个专家系统范例。全书叙述通俗易懂。

本书可作从事人工智能和专家系统研究开发的科技人员的参考书,也可作大专院校的计算机和自动控制专业的本科生和研究生的教学参考书。

人工智能与专家系统开发原理

出版:上海交通大学出版社
(淮海中路1984弄19号)

发行:新华书店上海发行所

印刷:上海交通大学印刷厂

开本:787×1092(毫米)1/16

印张:12

字数:292000

版次:1991年5月 第一版

印次:1991年5月 第一次

印数:1—2000

科目:255—812

ISBN7—313—00907—0/TP·13

定价:3.15元

译 序

D·W·罗尔斯顿所著的《人工智能与专家系统开发原理》(*Principles of Artificial Intelligence and Expert Systems Development*)一书比较系统地介绍了人工智能和专家系统的概念与方法。其内容丰富,叙述方式新颖,读者虽然需具有计算机科学方面的基础知识,但它易读,易懂。书中所讲到的有关人工智能的基本原理、知识表示和获取、不确定性处理、解释系统、消解原理和系统体系结构等内容,包括了80年以来国际上的一些最新研究成果。它将有助于我国从事人工智能和专家系统开发的科技人员,使他们从本书中可学到一些方法或得到一些启发。因此我们将它翻译出来,介绍给我国的读者,以推动我国人工智能和专家系统开发的研究。

参加本书翻译工作的有沈锦泉、袁天鑫、葛自良、吴修微和张建中等副教授,方盈毅、王渝中、裘杰、钱允琪、付尧清、王健、邹强和孙巍等同志也参加了本书的部分翻译工作。在本书的翻译过程中还得到了张钟俊教授的大力支持。由于我们的水平有限,译文中可能会出现缺点与错误,请读者批评指正。

译者 1990.12 于上海交通大学

前 言

本书的目的是向计算机科学专业人员和学生介绍专家系统 (ESs) 的概念、技术以及基本的人工智能 (AI) 技术, 它通俗易懂。

本书的叙述风格是务实, 并且着重于发展。它适合于那些想要知道如何构造一个 ES 的开发, 而不适于那些试图回答 (大家感兴趣的) 计算机是否确实能思考问题的学者。在全书中, 主要讲述 AI 和 ES 开发的具体的基本原理, 以使读者能尽快地得到一个全面的理解。

本书是在假定读者通过学习和实践在计算机科学方面已具备了一定基础的情况下展开的, 但它并不假定读者在 AI 方面已经有所了解。因此, 前五章对 AI 的概述进行了介绍, 并集中在那些对建立 ES 特别重要的概念上。这一部分是想让读者了解构成一个 AI 系统的方法的实质, 并了解 AI 系统的组成部分。

第六章至第十一章讲述了 ES 领域内 AI 概念的应用, 这些章节中的每一章集中讨论 ES 开发的诸方面——知识获取和处理不确定性、系统体系结构、解释、消解 (作为形式推理的一个例子) 以及开发过程本身。书中讲述的范围是广泛的, 尽可能多地表达了可供选择的观点, 而不去评价任何一种方法, 目的是使读者熟悉组成 ES 的团体的各个流派。

第十二章详尽地讲述了实际的 ES 开发工具, 这个工具已非常成功地应用于日常的现实问题。其目的是想要阐述如何将前面各章中讲述的内容与现实的系统开发联系起来。在这里描述了理论的直接应用, 并说明了折衷方法的必要。

尽管本书是着重讨论 ES 开发过程的, 但并不打算把它作为一个具体的实践指导, 也不进行具体的编程语言讨论, 这样做既能避免本书的内容过多, 又能防止对一些重要概念的理解被相当一部分次要问题所混淆。另外, 由于语言和工具领域是在不断地更新和演变发展的, 今天有用的描述也许明天就过时了。因此, 本书中仅为那些尚未接触该领域的读者就 Lisp、Prolog 以及开发工具作一些简短的小结, 这些小结至少能让读者对这些领域有个大致了解。

虽然这本书主要是提供给计算机科学工作者阅读的, 但它作为大学中 AI/ES 课程的主要参考书或作为研究生课程的附加参考书也是非常有用的。书后的参考文献则能让读者在所需了解的各个领域作更进一步的探索。

除此以外, 本书也能为那些对 AI 和 ES 思想产生兴趣的读者尽快地进入这个领域提供方便, 同时它也是一本引导读者开始在这领域中认真探索的书籍。

D.W. 罗尔斯顿

目 录

第一章 专家系统导论	(1)
1.1 专家系统的定义	(1)
1.2 专家系统的发展史	(2)
1.3 专家解题方法的分析	(2)
1.4 专家系统中知识的作用	(3)
1.5 知识分析	(3)
1.6 专家系统的结构	(4)
1.7 专家系统中的编程语言	(7)
1.8 开发过程	(8)
1.9 专家系统的开发现状	(8)
第二章 人工智能问题求解的概念	(10)
2.1 双桶问题	(10)
2.2 人工智能技术的分析	(11)
2.3 成功的标准	(13)
2.4 人工智能中的问题求解	(13)
第三章 知识表示与形式逻辑	(20)
3.1 知识成分	(20)
3.2 表示的评价标准	(20)
3.3 表示的层次	(21)
3.4 知识表示模式	(21)
3.5 形式逻辑	(22)
3.6 采用形式逻辑的知识工程	(26)
3.7 推理过程	(28)
第四章 非形式化知识表达	(30)
4.1 语义网络	(30)
4.2 框架	(32)
4.3 剧本	(35)
4.4 产生式系统	(36)
第五章 问题求解策略	(45)
5.1 穷举搜索	(45)
5.2 大规模搜索空间	(45)
5.3 生成与测试	(45)
5.4 空间转换	(47)
5.5 规划	(48)

5.6	最少约束原理及约束传播	(56)
5.7	分类模型	(57)
5.8	黑板模型	(58)
第六章	处理不确定性	(60)
6.1	基于部分信息的推理	(60)
6.2	非单调推理	(61)
6.3	正确性保持系统	(63)
6.4	基于概率的推理	(66)
6.5	可信度	(67)
6.6	模糊推理	(72)
第七章	解释工具	(75)
7.1	基本解释——重点在调试方面	(76)
7.2	因果模型	(79)
7.3	PRESS——集中于对状态的解释	(81)
7.4	CLEAR——集中于说明性的解释	(83)
7.5	XPLAIN——集中于合理性的解释	(85)
7.6	关于训练的解释	(87)
7.7	联机用户辅助系统	(89)
第八章	专家系统的开发过程	(91)
8.1	专家系统与软件工程	(91)
8.2	专家系统开发生命周期	(92)
8.3	问题选择	(93)
8.4	原型构造	(96)
8.5	形式化	(97)
8.6	实现	(100)
8.7	评价	(102)
8.8	长期发展	(102)
第九章	知识获取	(103)
9.1	专家——知识工程师 (KE) 的相互影响	(103)
9.2	认知过程	(104)
9.3	知识获取技术	(105)
9.4	样本问题实现	(105)
9.5	概念的系统阐述	(107)
9.6	知识表示的开发	(107)
9.7	关于核心问题的知识获取	(108)
9.8	多个专家的利用	(108)
9.9	无知识工程师的知识获取	(108)
第十章	专家系统工具	(110)
10.1	专家系统开发语言	(110)

10.2	专家系统外壳	(111)
10.3	小型工具	(112)
10.4	Lisp 机器	(113)
10.5	大型混合专家系统工具	(113)
10.6	基于 PC 的专家系统工具	(114)
10.7	知识获取工具	(115)
第十一章	基于形式逻辑的推理	(117)
11.1	用基本子句消解	(117)
11.2	合一	(120)
11.3	谓词逻辑中的消解	(121)
第十二章	多范例专家系统举例	(123)
12.1	用户接口	(123)
12.2	内部设计概述	(126)
12.3	故障排除核心	(127)
12.4	成分知识库	(134)
12.5	状态的表示	(134)
12.6	混合初始化处理	(135)
12.7	预兆维护	(137)
12.8	文件和媒体的恢复	(139)
12.9	解释系统	(139)
12.10	支持工具	(140)
12.11	结果分析	(140)
12.12	专家系统的未来方向	(141)
附录 A	Lisp 编程	(143)
A.1	Lisp 理论	(143)
A.2	函数型编程语言	(143)
A.3	变量赋值与函数定义	(144)
A.4	表处理	(146)
A.5	动态存储器管理	(148)
A.6	控制	(148)
A.7	特殊的 Lisp 函数	(151)
附录 B	Prolog 编程	(152)
B.1	理论概念	(152)
B.2	基本编程元素和语法	(152)
B.3	Prolog 中的推理	(155)
B.4	Prolog 中的显式控制	(159)
B.5	Prolog 程序举例	(161)
参考文献		(163)
索引		(171)

第一章 专家系统导论

本章对专家系统作最为基本的介绍，并引出一些基本概念，其中主要是许多术语。在以后的章节中，我们将对这些术语作更为详细的解释。如果读者对专家系统已有一定的了解，则可以从第二章读起。

专家系统 (ESs) 就是用来完成一系列非常复杂、在过去也许只能由极少数经过高级训练的人类专家才能完成的任务的系统。应用人工智能 (AI) 的技术，ESs 可以获取一些基本知识，这些知识足以使一般的人在处理复杂问题时如同专家一样。

专家系统与传统计算机应用的区别也许在于它有能力采用人类判断及直觉的反应过程来处理当今世界上一些富有挑战性的问题，而这也恰恰就是 ESs 最为使人感兴趣和有效的特点。

目前，ESs 正在许多领域中得到应用，如：医疗诊断、工程计划、天气预报、图纸设计、语言翻译、工业控制、状态监测和学校教育等领域。将来，随着直接支持 ES 执行的新的硬件结构的发展以及 AI 技术的成熟，我们有理由期待专家系统的进一步发展，而这些发展标志着专家系统在许多领域里的行为已越来越接近于人类专家的行为。

这种系统的发展将不仅能提供强有力的技术能力，而且将更进一步地完善我们自己对人类认识过程的理解。

1.1 专家系统的定义

一个专家系统其实质就是用来解决复杂问题的一种计算机应用，而这类问题如不通过专家系统来解决，那就要求有广博的人类专家知识。专家系统其实就是通过应用专门知识和推理来模仿人类的推理过程而进行工作的。

一个理想化的专家系统按其内在特点可包括如下诸方面：

- 来自感兴趣领域的广博的专门知识
- 搜索技术的应用
- 支持启发式分析
- 具有从现有知识推理出新知识的能力
- 符号处理
- 解释本身推理过程的能力

表 1.1 早期的专家系统

系 统	日期	研制者	主 题
Dendral	1965	Stanford	推断化学结构的信息
Macsyma	1965	MIT	完成复杂数学分析
Hearsay	1965	Carnegie-Mellon	子集语言的自然语言解释
Age	1973	Stanford	专家系统生成工具
Mycin	1972	Stanford	血液疾病诊断
Teiresias	1972	Stanford	知识转换工具
Prospector	1972	Stanford Res.Inst.	矿物勘探及鉴别工具
Rosie	1978	Rand	专家系统构造工具
OPS 5	1974	Carnegie-Mellon	专家系统构造工具
R 1	1978	Carnegie-Mellon	DEC计算机设备的结构布局
Caduceus	1975	Univ. of Pittsburgh	内科医疗诊断工具

1.2 专家系统的发展史

当前出现的专家系统技术,是50年代后期广泛开展的AI技术研究的自然产物。有关的研究起源于支持符号推理的语言,如程序语言IPL就是第一个符号、列表处理语言,它广泛用于早期的AI实现中。Lisp语言,1958年由约翰·麦卡锡开发出来后,已成为当今世界上人工智能领域中最为流行的语言之一。

真正具体研究专家系统是从60年代中期开始的。在1965年至1970年间有几个系统相继问世,而其中大多数系统的应用范围很有限,还只涉及到游戏或高度学术性、理想化的一些课题。表1.1列出了几个特别有意义的专家系统,它们奠定了专家系统的大部分技术的历史基础。虽然专家系统的发展历史至今仍十分短暂,但是目前已有许多专家系统被不同的用户用于各种应用领域中[布坎南(Buchanan),1986;沃特曼(Waterman),1986]。

1.3 专家解题方法的分析

有一些人已概括了他们对专家行为的看法,指出专家是一种“带着公文包、离家50英里(1英里等于1609.344m)以上”的人。但是如果我们要模仿人类专家处理一个复杂问题的过程,首先必须对该过程有一个清晰的了解。

作为一个例子,让我们来考虑一个飞行员对指示电路故障的报警灯的反应过程。为了讨论方便起见,我们假定飞行员必须作出的决定是,执行紧急着落、关闭一台发动机或完全不顾报警。另外,假定发生以下事件:

飞行员迅速作出以下规划:

- 试图验证报警的正确性;
- 如果报警是错误的话,就在飞行日记中做下记录,继续正常飞行,不顾报警;
- 如果报警是正确的话,且仅有一台发动机遇到故障,则关闭这台发动机;

• 如果报警是正确的话,且不止一台发动机发生故障,则执行紧急着陆。

飞行员通过验证报警的正确性来开始执行该方案。验证过程包括观察飞机的总体情况,检查飞行设备和要求机组人员报告不正常的情况。通过检查飞行设备,飞行员注意到姿态指示器显示飞机在作水平飞行,而垂直速度指示器指示出飞机正以 4000 英尺/分(1 英尺等于 30.48cm)的速度下降。飞行员知道是由于电气故障而导致了指示器上的相互矛盾的读数,同时还观察到机舱灯闪烁不停,因此他得出了报警是正确的结论。

然后,飞行员试图确定上述问题是否可与一个发动机的电气系统相分离,并且,他想通过机载计算机执行一系列的诊断测试来作出这种判断。测试的结果并不令人满意。有五种测试指出仅仅有一台发动机出现故障,而另一种测试则指出总体系统发生了障碍。这样,飞行员就得同时考虑飞行状态下这些测试的总可靠性,因为带有毛病的电气系统也驱动着计算机来执行一系列的测试。

基于与其在不可靠的条件下继续飞行还不如在可控制的情况下尝试紧急着落来得更好的这样一条基本原则,最后,飞行员决定开始紧急着落。

很明显,飞行员必须在确保几个约束条件(如旅客安全)下力图逐渐接近目标(即避免旅客不舒服)。另外,很重要的一点,就是飞行员必须非常迅速地完成任务。

1.4 专家系统中知识的作用

在上面的例子中,飞行员在其整个规划过程中集中显示出两种不同形式的专家知识。第一,他了解逐步逼近一个问题解答的基本过程。这是许多领域中相当普遍的专家知识形式。同样的基本解题技术也有可能被一个船长应用于类似的事件之中。早期的 AI 主要集中于详细研究这方面的专家行为。第二,飞行员所采用的是具体领域的知识。这种知识贯穿于飞行员的整个反应过程中。

具体领域知识的作用可以通过对比飞行员观察仪器的读数,和非飞行人员对本航班客舱的典型反应来进一步说明。大多数非飞行人员,即使他们非常聪明,也都只能把仪器、测量装置及控制器失灵看作故障的范围。

一个船长,他对飞机上仪器的了解程度并不会比一个没有专家水平的人好多少,尽管如此,他仍将花相当一部分时间从事海上的专家活动,这其中有许多同飞行员的活动是类似的。

由此可得到一个设计专家系统的最为基本的原理,那就是早在 1947 年由弗朗西斯·培根所写的“知识就是力量”。

原理 1.1: 一个专家的能力来源于广泛领域的专门知识,而不是对一般专家行为的了解。

当原理 1.1 被承认,并在 70 年代后期开始影响人工智能研究时,它对 ESs 的冲击便迅速加快。现在,它已作为一个基本原理被广泛地接受了。

1.5 知识分析

一个专家执行任务的能力来源于好多种知识,它们大致可分为如下几类:

- 事实。表达有关领域中某些真理的陈述。例如：牛奶是白色的；

- 粘性陶瓷瓦片，如果不到 2 个月长的时间，将由于粘性不够而不能凝固，一架波音 747 客机在三台发动机都工作的情况下可安全飞行。

- 过程规则。这是一种精心定义的、始终不变的规则，它描述了一系列基本事件以及与该事件领域有关的各种关系。例如：

在上高速公路之前，总是要观察一下交通情况；

如果姿态指示器指示水平飞行的话，则垂直速度指示器读数应为零。

- 启发式规则。当始终不变的过程规则不适用时，就可采用由经验得出的一般规则，这些规则只是大致性的，它们通常需要专家几年的积累才能得到。例如：

如果一根链条看上去是好的，但不能松开，那么就去掉链条上的张力；

与其在故障不明的情况下继续飞行，还不如在有控制情况下进行紧急着陆。

启发式方法的出现对 ESs 的功能及灵活性的提高起了很大作用，并且逐渐使 ESs 从传统的软件中区分出来了。

除了以上这些特殊形式的知识以外，一个专家还要有关于研究领域的一个一般的概念模型和找到问题答案的总体构想。这些“全程视图”形成了专家应用具体知识的基本结构。

1.6 专家系统的结构

ESs 之所以采用许多特殊的系统结构，主要是因为对于给定的应用，一种结构有可能比另一种结构更为有效的缘故。当前，广泛的研究工作正在推动着 ES 结构的各个方面的研究，其中许多方面尚待进一步讨论。

不管各种专家系统的结构有多大差别，它们中的大多数还是有一些共同点的。在图 1.1 中画出了一个较为典型的典型专家系统结构。下面各小节将对图 1.1 中的各个部分作详细说明。

用户

专家系统的用户可以是如下几种类型中的任何一种：

- 测试者。这种用户试图证实系统行为的真实性。

- 指导者。这种用户给系统提供附加知识或修改系统中的已有知识。

- 学习者。这种用户试图通过从系统中获取经过精心组织和提炼的知识来迅速提高个人专业知识。

- 客户。这种用户将系统的专门知识应用到特殊的实际任务中去。

上述那些用户的区分与传统软件系统中只有唯一一个用户（客户）的比较典型区分是不

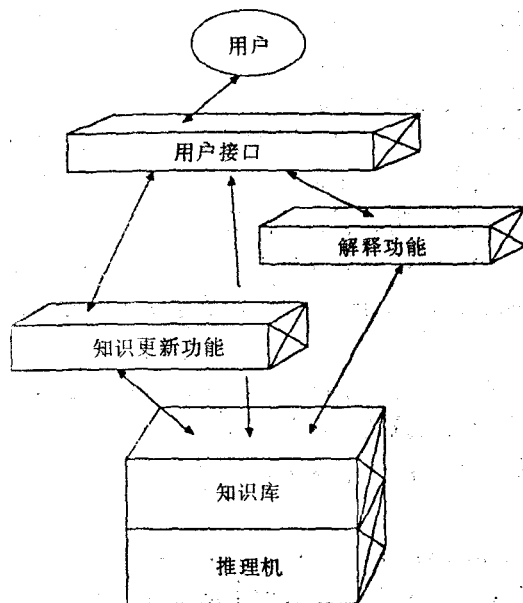


图 1.1 典型的专家系统结构

相同的。

用户接口工具

用户接口工具必须能够从用户那里接受信息，并应该将它翻译成系统其余部分能够接受的形式，或接受系统的信息并把它转换成用户能理解的形式。

一种理想、方便的工具由自然语言处理系统构成，这种系统能接受并返回其形式基本上与人类专家所能接受或提供的形式相同的信息。但是，今天还没有任何一种系统能复现自然语言，不过也有几种结果已经引起人们的注目，这些结果是通过应用受限语言子集而得到的。

ESs 的用户接口工具通常应设计得能够识别用户的操作方式、用户专门知识的水平以及具体事务的自然特征。尽管自然语言对话还不完全可能，但是，因为我们试图想用专家系统代替人类的行为，因此，借助于该系统来进行对话应该是很自然的。

知识的存贮和生成系统

知识的存贮和生成系统由一个知识库和一个推理机构成，它是 ES 的心脏部分，它的功能就是在系统中可靠地存放专家知识，并从存贮器中恢复知识以及需要时推理出新知识。

知识库

知识库是可用于系统的知识元（即基本事实、过程规则和启发式规则）的一个仓库。正像原理 1.1 所描述的那样，在知识库中贮存的知识确立了一个专家系统能够发挥专家作用的能力。

通常，知识是以事实和规则方式存放的，但是，也有许多用来存放信息的特殊结构，而且这种知识表达方式的设计影响到推理机的设计、知识更新过程、解释过程和系统的总体效率。

原理 1.2： 知识表达方式的选择是 ES 设计的最关键决策之一。

知识工程 知识工程是获取特定的领域知识并将它送到知识库中去的过程。图 1.2 是详细说明这个过程的典型例子。虽然知识可以由一系列信息源（所包含的文件和所存在的计算机信息系统）获得，但其中的大多数还是必须由人类专家提供。一般情况下，由专家所提供的知识是面向所讨论问题的领域的。

一个知识工程师（KE），其任务就是从领域专家处获取知识并将它传送到知识库中去。因为 ES 要求根据系统的知识表示习惯将知识存放到知识库中去，所以 KE 还必须将知识表示方法的转换作为其传送过程的一部分。

为了获得必要的知识，KE 首先必须对有关领域有一个总体了解，在自己的头脑中形成一个有关领域的专门词汇和术语的词典，并对一些关键概念有一个基本理解。其次，他们必须从专家提供的信息中提取出简明的知识。

在专家系统的开发过程中，最为困难的方面常常是其知识获取功能，这主要是由于知识获取过程往往要求领域专家和 KE 之间有广泛的人际交流，

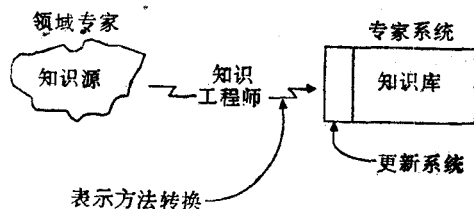


图 1.2 知识工程

因此，会遇到与此相关的一些问题。

由此得出结论，知识获取过程还并未真正得到了解，也并未很好定义。如果将 ES 开发的本身看成一个专家领域的活，那末与知识获取过程有关的知识则可看成是启发式的。

推理机 从 ESs 的本质来看，它必须能够灵活地处理各种情况，而对各种情况作出响应的能力有赖于从现有知识推理出新知识的能力。作为一个推理的简单例子，让我们考虑以下两个基本事实：

1. 所有动物都呼吸氧气；
2. 所有狗都是动物。

从这两个事实可推得一个新的事实：所有狗都呼吸氧气。为了对给定的情况作出反应，ES 必须应用合适的知识。例如，在以前描述的飞机例子中，飞行员必须采用波音 747 客机在三台发动机工作情况下能安全飞行这个事实。而所谓应用合适的知识就是指所需的知识既可以是已被放在知识库中现存的知识，也可以是由现存知识推得的新知识。

原理 1.3：合适知识的搜索以及由它推得新知识的过程是专家系统执行过程的关键部分。

当然，将“所有的狗都呼吸氧气”这个事实直接贮存在知识库中，而不是在执行过程中由知识元推得，这也是完全可以的。

包含在知识元操作中最为困难的问题之一是，事实上即使少数几个个别的成分（即知识元）也能组合成大量独特的组合知识，因此从大量的知识成分中迅速地组合出各种独特知识的可能性就变得极其庞大。这个问题就是大家熟知的所谓知识的组合爆炸问题。为了解决这个问题，大多数 ES 实际上是依靠使用编译知识——通过几年实践后由“后台方式”产生的高层次的知识——而并不是直接操作知识元。

推理机是从知识库中找出知识，并推出新知识的软件系统，它的推理过程是用于推得所要求知识的一种搜索策略。虽然用于 ES 的推理过程有许多不同的范例，但是其中大多数是基于以下两个基本概念中的一个：逆向链，这是一个自上而下的推理过程，即由所要求的目标开始，向后推理至所要求的条件；正向链，这是一个自下向上的推理过程，即由所知道的条件出发，向前推理至所要求的目标。

原理 1.4：从知识的组合爆炸来考虑，推理范例的选择，将极大地影响 ES 的总体性能。

知识更新

按推测，当我们使用专家系统时，知识库应该是其相应领域知识的精确反映，然而，许多复杂领域中的知识是经常要扩充和改变的，因此知识库也必须作相应的修改。可利用知识更新工具来作出这样的修改，而具体修改过程可以采用以下介绍的三种方式中的任何一种：

第一种方式是人工更新知识。在这种情况下，知识的更新由知识工程师完成。知识工程师解释由领域专家提供的信息，并通过一个有限的知识更新系统来修改知识库。

第二种方式体现了 ESs 的技术发展水平，它由领域专家直接参与修正知识而不要知识工程师作为媒介。在这种情况下，知识更新系统必须相当完善。

第三种方式是机器学习。这种专家系统能自动产生新知识，而这些新知识是基于过去经验中的一些普遍性结论而产生的。实际上，该系统是经验的学习系统。因此，理想的叫法应

是修正系统。这种学习过程还处于理论阶段，但它却是许多研究工作的主要课题。学习能力是智能的一个重要组成部分，充分地提供这种能力将可极大地提高 ES 的功能。

原理 1.5：一个理想的 ES，其推理机决不需要修改。

理想情况下，知识系统的所有进一步提高都可通过扩展知识库来实现，然而，要保证知识库同推理机是完全独立的这种可能性极小。

解释系统

当一位专家面临一个复杂问题时，除了简单得到一个结论以外，他还需具有解释该问题的能力，从而在某种程度上，可推理出结论。因而 ES 也应设计得使其能提供与专家相似的能力，而这也是传统计算机系统通常无法达到的一个重要功能。

典型的解释程序包括推理过程步骤的识别以及对每一步骤合理性的解释。实质上，提供这种信息交换的能力，就是自然语言处理问题的子问题。同时，解释系统还必须按照知识库的表示结构来存取处理过程中用到的知识记录，并把它们转换为用户乐于接受的形式。

原理 1.6：一个专家系统的可信程度依赖于该专家系统解释其自身推理过程的能力。

一个人类专家也许能够根据其听众专门知识的水平来制定其解释推理过程的形式。例如，一个飞行员可以对乘客解释说：“飞机将由于技术原因而提前降落”。而另一个飞行员的等效解释可以是：“我看到 T-37（警告灯）亮了，它好像是机舱左舷螺旋桨（发动机）的警告灯，我无法关闭它，因此只好降落。”

为了提供各种不同层次的解释，解释系统必须能识别用户专业知识的水平，并要知道怎样制定解释才能为用户所接受。在许多现有的系统中解释工具仅局限于列出执行过程中所使用的一些规则而已。

1.7 专家系统中的编程语言

通常，专家系统编程的焦点是推理和启发式搜索的问题，它在很大程度上依赖于符号：字符串（即“名字”）处理，它们可自由地用于所讨论的领域中任何可能成分的表示。

编程语言 Lisp 和 Prolog 是迄今为止在 ES 开发中最常用的语言，不过，即使是比较常规的语言——特别是 C 语言——也逐渐地开始采用。从概念上看，Lisp 语言是一种函数语言，该语言中的每一条语句都是函数的一种描述。而 Prolog 语言则是一种逻辑语言，该语言中的每条语句都是形式逻辑语法的一种表示。

符号处理在 ESs 中是十分重要的，这是因为知识库中的知识元以及知识元之间的联系是通过采用符号表示贮存起来的。因此，如果 ES 的编程语言能自由处理“一类事情”而不必考虑它们的组成的话，则是十分有用的。

Lisp 语言、Prolog 语言及其他一些可选用语言的相对比较问题一直是争论的热点。虽然这些争论的基础有时可能是传统的、地区性的而非技术性的，但是有一点是可以肯定的，即对于一定的应用来说，某种语言会比其他语言来得合适。此外，随着软件开发工具的建立，用得最普遍的语言（即 Lisp）得到了强大的支持。

随着 ES 开发速度的提高，随着新的硬件结构的开发，有可能流行新的符号语言。大量的研究工作都是朝着开发一种能将函数语言（如 Lisp 语言）和逻辑语言（如 Prolog 语言）

的特点结合起来的语言的方向在努力,当这种语言逐渐进入日常使用时,我们就有理由期望更为完善的开发工具发展起来。

1.8 开发过程

ES 的开发过程由类似于标准软件工程的生命周期部分的几个基本阶段组成,这些阶段有问题识别、原型构造、形式化、实现、评价以及长期发展等等。

无一例外,所有 ES 开发的第一件事情就是确定所提出的问题是否合适,是否有必要采用 ES 来解决。如果所考虑的问题可以直接用定义和算法来描述的话,则它可能开发传统的软件来求解更好,而如果不能直接定义或需要进一步的人工裁决(如裁决一次技术比赛)的话,则采用专家系统又可能太复杂了。

选定了一个合适的问题之后,我们就要构造一个小的原型,以便帮助我们理解复杂问题并估计出为了得到其完全解而所要做的工作。下一步开发过程就是正式形式化问题的表述和设计完整的 ES。接下去的一步便是进行实现,它主要包括连续循环地获取知识、知识库的更新以及测试等。

紧随在实现之后的是评价阶段,它是用来估价该系统接近于专家行为程度的。在评价及出版、发行之后,ES 便进入了长期发展阶段,在这个阶段中,系统继续得到完善(根据使用的经验),并不断地得到修正,以适应领域知识的改变。

1.9 专家系统的开发现状

目前,ESs 可分成以下三个基本大类[戴维斯(Davis), 1985]:

1. 助手型:它是一个小型*知识库系统,可用来完成经济上有价值,但技术上十分有限的专家任务的子集。许多“助手型”的系统都是以 PC 为基础的;
2. 同事型:它是一个中型知识库系统,可用来执行一个专家任务的有效子集。“同事型”系统既可在 PCs 上实现,又可在较大的机器(如特殊的工作站和常用的大型计算机)上实现;
3. 专家型:它是一个大型的知识库系统,在一个给定的领域中,性能接近于人类专家水平,通过使用复杂的开发工具,这类专家系统通常可在功能很强的机器上实现。

最近几年,“专家系统”一词应用得相当普遍。上述三类系统之所以都被称为专家系统,是因为它们都把专家作为其知识来源,并都采用知识库技术来实现其功能的缘故。大多数现有系统(和某些最有经济价值的系统)都属于同事型和助手型,实际上,目前几乎还没有专家系统能在复杂领域中用来替代人类专家。

在这本书中所描述的概念和功能对以上所有类型的系统都是适用的,虽然其中有一些概念和功能在助手型和同事型系统中可以不考虑。尽管现有的 ESs 已十分有效且作用也很大,但是我们仍然必须认识到,在当前的技术状态下其能力非常有限。一些典型的系统有以下限制:

- 从很少的人类专家那里获取知识;

* 有关其规模问题将在后面的章中讨论

- 只能应用于有限的特殊领域或一小部分领域中；
- 应用领域必须很少需要在时间或空间上进行推理；
- 任务并不希望采用大量普遍的或一看就知道的知识；
- 为了完成某个任务所需要的知识是合适而完善的、正确的和固定不变的。

虽然存在着这样的限制，但仍有许多领域适合应用 ES。本书的其余章节讲述了构成 ESs 基础的 AI 概念，然后提供逐步深入的 ESs 内部操作和实现的描述。