

分类号

密级

硕士学位论文

题目：一种专家系统开发工具——EST 的实现
及其在发酵过程中的应用

英文并列题目：

THE REALIZATION OF EST, AN EXPERT SYSTEM DEVELOPMENT TOOL
AND ITS APPLICATION IN FERMENTATION PROCESS

研究生：陈伟 专业：工业自动化

研究方向：计算机在生化过程中的应用

导师：张遐、孙燕唐

学位授予日期：

1991 年 6 月 日

无锡轻工业学院

地址：无锡市青山湾

摘 要	1
ABSTRACT	2
前 言	3
第一章 人工智能、专家系统和专家系统开发工具	5
§ 1 人工智能及其发展	5
§ 2 专家系统及其发展	5
§ 3 专家系统开发工具	8
第二章 建立小型、多功能、通用的专家系统开发工具的设想	11
§ 1 建立EST 的基本思想	11
§ 2 问题和对策	14
第三章 专家系统开发工具EST 的实现	17
§ 1 接口的具体实现	17
§ 2 开发机的实现	21
§ 3 推理机的实现	24
§ 4 解释机的实现	28
§ 5 EST 的其他开发环境	28
第四章 EST 在发酵过程中的应用——FERMES	29
§ 1 FERMES的总体结构	29
§ 2 一级知识库的构成	31
§ 3 二级知识库和二级推理机的构成	31
§ 4 单元操作的定义	31
§ 5 单元操作与EST 的连接	33
结 论	35
参考资料	36
致 谢	38

摘 要

建立发酵过程的知识咨询系统或其他小型实用的专家系统时，需要一种合适有效的专家系统开发工具。本文先运用专家系统技术和Turbo Prolog人工智能语言，实现了一种具有小型、多功能、通用特点的专家系统开发工具EST，它包括动态窗口驱动，通用的知识表达方式，多功能的开发机，有带回溯和不带回溯的深度优先的正向搜索及深度优先的逆向搜索三种搜索算法的推理机以及解释机四部分，并提供了用EST建立实用的专家系统的多种连接方式。然后我们把EST应用于发酵过程，建立发酵过程知识咨询系统FERMES的总体结构，它具有分类性的咨询知识库，正向搜索的推理算法，和能完成咨询操作的智能数据库。并在发酵专家的帮助下，具体实现了FERMES中的五种咨询操作。实践证明EST在开发专家系统过程中的灵活性和有效性，同时也证实FERMES是可行和有前途的。

关键词：专家系统开发工具；接口；推理机；开发机；解释机；知识库；数据库；正向搜索；逆向搜索；深度优先搜索；发酵过程；知识咨询系统。

THE REALIZATION OF AN EXPERT SYSTEM DEVELOPMENT TOOL--EST AND ITS APPLICATION IN FERMENTATION PROCESS

ABSTRACT In order to build the knowledge consulting system for fermentation process and other expert systems, it is necessary to have a suitable and effective development tool. In this paper, a multiuse expert system development tool with small scale EST has been first realized. It contains a dynamic-windows driving, a universal expression prototype for knowledge, a multi-functional developer, a inferencing engine of three algorithm---depth first forward reasoning algorithms with or without backtracking and depth first backward reasoning algorithm; and explanations. Also several methods to build expert system with EST has been supplied. Than we used EST in fermentation process. The FERMES, a knowledge consulting system for fermentation process has been built. It is composed of a knowledge base with claasified consults, a forward reasoning inferencing algorithm and a intelligent database which can execute consulting operations. With the help of fermentation expert and technical books, five consulting operations in FERMES has been realized. It is resulted that EST is flexible and effective and also the building of FERMES is successful.

KEYWORD Expert system development tool; interface; inferencing engine; developer; explanation; knowledge base; database; forward reasoning; backward reasioning; depth first search; fermentation process; knowledge consulting system.

前 言

在课题提出的最初，计划建立一个关于发酵过程的专家系统。

随着发酵工艺及其生产实践的发展，发酵工业对其过程的自动化程度和发酵过程的优化提出了新的要求。要做到这点，除了在发酵工艺、设备等方面进行努力外，在计算机应用方面同样大有作为。计算机的介入，尤其是将人工智能的思想的引入，不仅可以实现对发酵过程的优化，而且还可以对发酵工艺等方面的知识和经验进行总结。

在发酵过程的模型化和控制方面，由于发酵过程机理的复杂性和参数检测手段的相对落后，发酵过程的模型化较之其他过程的模型化困难得多。鉴于这个特点，人们除了根据不同的原理，借助不同的手段，探索出了一系列发酵过程的模型结构，以及模型参数的间接估计方法，还将人工智能用于控制，在专家系统和智能控制方面进行探索。

在发酵的工艺和设备方面，许多人从事这项研究，并取得相当多的成果。但由于各自从事的领域不同，无论是手段、成果以及成果的推广，其广泛程度都受到影响，有必要用有效的手段，将这些成果和经验，以及其他规律性的知识进行系统的总结，并以简单易行的方式为使用者提供咨询。这使我们产生了建立一种发酵过程的专家系统的想法，用专家系统的技术，吸收专家和书本上的知识，产生较为系统和相对完整的知识库，运用于具体问题。这样的专家系统对发酵问题有专家水平的分析和解决手段，又可以由非专家使用。虽然，这个专家系统在建立之初，其知识水平受到这方面知识不够完整的限制，但正是由于专家系统有学习的特点，才使这样的专家系统的诞生成为有效、合理。

要建立上述的发酵过程的专家系统，使用一种合适的开发工具是至关重要的。离开这种开发工具和环境，即使有最完美的知识系统也不能正常工作。而这样的专家系统对其开发工具有其特有的要求：

1、无论是发酵研究人员或发酵生产部门，拥有的计算机多为PC机或工控机，这就要求专家系统开发工具不能过大。

2、由于发酵涉及的知识面广，而且知识系统尚在逐步完善中，它又要求开发工具能比较灵活，能适合多种操作的需要，如能添加、修改、删除规则，并能检查知识库，及时排错，能够解释推理过程，解释规则，并能记录静态和动态数据。

3、由于使用者多为发酵方面的用户，这就要求有良好的人-机界面。

在分析了现有的专家系统开发工具以后，我们发现，虽然它们多具有较强的知识表达能力和有效的推理策略，但要用于建立发酵过程的专家系统，仍需作相应的改进；且修改的幅度不小；需使用的软件工具不充足，为此，我们建立了这种小型而多功能的专家系统开发工具。

✓ 在建立过程中，我们还尝试将知识库中的知识表示作通用化处理；将知识库与专家系统的其他部分相对独立。这样，只要使用这种开发工具，根据知识的表达方式，产生一系列的产生式规则；建立起其他领域的知识库，同样也能建立起一种新的专家系统。

为在实践中检验这种开发工具，我们在建立发酵过程知识咨询的专家系统中应用了该工具。为了区别于其他的开发工具，称我们所建立的专家系统开发工具为EST(Expert System development Tool)。

我们在对EST的基本构成和具体实现作介绍后，还将EST应用于发酵领域中；建立起发酵过程知识咨询系统FERMES的简单模型；它具有分类性的知识咨询系统和带智能的咨询操作数据库，使咨询在这个智能数据库的自动搜索中最终完成。我们用FERMES在发酵工艺、发酵工厂设计和发酵过程模型化等方面作了试验，并获得了预期的结果。

建立EST要涉及人工智能领域中的许多概念和约定，为此，在本文的第一章中将作专门的介绍；在第二章中对建立EST作基本的设想，并从理论上解决面临的一些困难；在第三章中对EST的具体实现作较详细的介绍；在第四章中则对EST的具体应用——FERMES作介绍；结论在第五章给出。

第一章 人工智能, 专家系统和专家系统开发工具

§ 1 : 人工智能及其发展

人工智能是一门新兴的边缘性学科。自1956年问世以来,三十多年间,已经取得很大的进展;并日渐引起人们越来越多的重视。许多具有不同专业背景的科学家正在从人工智能这门年轻的学科中发现一些新的思想和新的方法。

人工智能,概括地说,是研究怎样让电脑模仿人脑从事推理、规划、设计和思考;学习等思维活动,解决迄今认为需由专家才能处理好的复杂问题。其研究领域包括语言处理;自动定理证明,智能数据检索系统,视觉系统,问题求解;人工智能方法和程序语言以及自动程序设计等,并建立了一些具有人工智能的计算机系统,能够完成求解微分方程;设计分析集成电路;合成人类自然语言;检索情报;诊断疾病以及控制太空飞行物和水下机器人等具有不同程序的人工智能的工作。

人工智能之所以具有强大的吸引力;与其说是由于它的成就,不如说是由于它的潜力。已经可以看到,人工智能将使计算机能够解决那些至今人们还不知如何解决的问题;从而大大地扩充其用途,它将带来计算机硬件和软件的革命;人工智能正向各个领域渗透,带来这些领域内知识的更新换代;智能计算机管理;智能计算机辅助设计;智能机器人等新的研究领域,由于人工智能的发展而不断出现;人工智能的发展还有助于人们进一步理解人类智能的机制。

专家系统,作为人工智能的一个重要分支,在其研究中已经出现了成功和有效地应用人工智能技术的趋势。

§ 2 : 专家系统及其发展

一、专家系统及其功能特点

在人工智能的研究和应用中,有相当大一部分的工作要求具备许多专门知识来完成推理任务;这些知识并非人人都掌握,只有少数积累了专门知识的专家们才能完成这样的工作。我们称完成这类工作的程序为专家系统;而这种专家系统的处理过程则称为知识工程。

建立专家系统的目的有以下几点:

- ① 为了传播珍贵的专门知识。专家系统能汇集和复现这种专家的知识。
- ② 使专家的知识形式化,即把专家的知识进行系统的总结和表示。
- ③ 汇集各种来源的知识。

用户和专家系统进行“咨询对话”,就如同在与具有某个方面经验的专家进行对话:专家系统解释用户的问题,建议采取某种策略或进行某种实验;用户也可向专家系统提出询问,以求得有关的解答。

可见；要使专家系统能够正常有效地运行，它必须具备以下几个条件：

- ① 系统必须拥有知识，而且知识必须相对正确和完整。
- ② 系统必须具备某种推理能力，能够借助于推理，由已知求得未知。
- ③ 系统必须具备某种继续获得知识的功能，因为只有通过学习，才能舍弃原有的不够准确的知识，“学习”新的知识；更新知识系统，使之不断地完善；更具生命力。

因而，称专家系统为基于知识的系统（Knowledge Based System）。

二、专家系统的构成及其工作过程

专家系统与一般的应用软件之间的显著区别在于：一般的应用软件把问题的求解知识用程序语言编写在程序中，而专家系统则将其应用领域的问题求解知识单独分开，组成一个叫做知识库的实体。知识库的处理通过与知识库分开的、易于识别的控制策略进行，亦即，一般的应用软件将知识分为两级：数据级和程序级；而专家系统则将知识组织为三级：数据库级、知识库级和控制策略级。

故一个计算机的专家系统是一种知识信息的加工处理系统。一般地讲，它是由环境模型（又称工作数据库；短记忆；工作存储器等）；知识库（又称规则存储器；定理集；子句集等）和控制策略（又称推理机制；知识管理等）三部分组成。如图1.1所示。这三部分或多或少是分离的。图中单箭头表示控制信息，双箭头表示数据流。

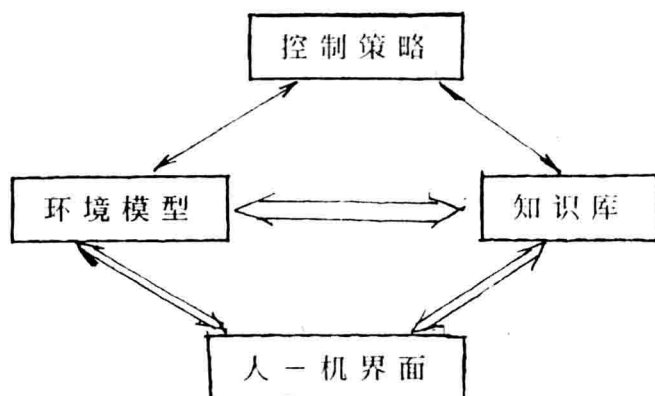


图1.1 专家系统的组成

环境模型是已经解决了的特定问题的说明性知识，以及当时需要求解问题的有关事件的状态，它包括了已知的事实以及工作过程中的动态生成的事实。知识库中存储着某领域的专业知识和经验，又称系统的目标知识。是否拥有大量的正确的知识是专家系统成功与否的关键。

控制策略是关于使用目标知识的知识，又称元知识 (meta-knowledge)。它根据既定的求解问题的知识，来决定应用知识库中的哪些知识。这里，控制策略是指推理方式。根据是否需要用概率信息来决定是精确推理还是非精确推理。此外，推理方式的不同还取决于其对搜索程度的依赖，它受下面两个重要因素的影响：

- ① 问题可行解的个数；
- ② 求问题的所有可行解，还是求第一个可行解，还是求最优解。

一个专家系统的工作过程，可以大致描述为：系统根据用户提出的目标，以环境模型为出发点，在控制策略的指导下，运用知识库中的有关知识，通过不断的搜索，实现要求的目标。所以，专家系统的工作过程实际上是问题求解的过程，是一个搜索的过程。

三、产生式系统

产生式系统 (production system) 中，领域的知识被分成两部分：凡是静态的知识，如事物、事件和它们之间的关系，以所谓事实来表示，而把推理和行为的过程以所谓产生式规则来表示。由于这类系统的知识库中主要存储的是规则，所以又称为基于规则的系统 (rule-based system)。

概括地说，产生式系统的基本要素是：

- ① 一组规则。每一规则有一个左部和一个右部。左部确定了该规则的可应用性，右部描述了应用该规则时要采取的行动。
- ② 一个或多个数据库，它们含有适于该特定任务的信息。数据库中有些部分可以是永久性的，有些部分则仅与求解当前问题有关，可用任何适当的方式构造数据库中的信息。
- ③ 一个控制策略。它规定了规则同数据库作比较的顺序，还规定了多条规则同时匹配某一数据时的解决办法。

图1.2给出了产生式系统的组成。

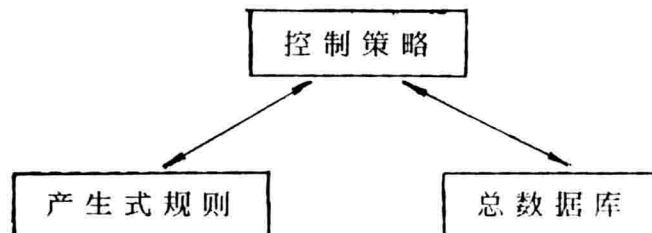


图1.2 产生式系统的组成

产生式系统除用作一种描述搜索的方法外，还具有其他的特点：

- ① 便于模仿智能行为中强数据驱动的特征。新的输入一旦进入数据库，

系统的行为就发生变化。

② 易于增加新的规则；考虑新的情况，同时并不破坏系统中别的部分。这一点十分重要，因为任何专家系统都不可能完美无缺。

§ 3 专家系统开发工具 (Expert System Development Tools)

一、专家系统开发工具及其发展

在专家系统研制的初期，即六、七十年代，研究者都致力于建立针对特定应用领域的专家系统，如医疗诊断、地质勘探等。然而，建立一个完整的专家系统并非易事，通常需要领域专家和知识工程师的数人月乃至数人年的通力合作。领域专家负责总结确定放入专家系统中的知识，并确定专家系统应达到的目标；因而需要具备高深的专业知识和丰富实践经验的待开发领域内的专家来完成；知识工程师负责从领域专家总结的知识和经验中抽取精华，设计知识结构；选用知识表达方法；根据不同的知识表达确立不同的知识存储方式和推理机制；并选用LISP、PROLOG等人工智能程序设计语言实现系统原型。几经试用和修改后，再提交最终用户使用。因而须由精通人工智能的计算机人员承担。最终用户可以不熟悉计算机和专业知识，但是要知道专家系统的用法，并用专家系统作更好、更合理的判断。照此，用LISP、PROLOG从头开发每一个系统，重复性工作太多；研制周期太长，对研制者，特别是知识工程师的素质要求太高，不易推广普及；难以满足实际应用的需要。

随着专家系统的应用潜力的不断增加，对开发专家系统的有效工具的需要日趋迫切，一批复杂程度不等，支撑计算机机型不同的专家系统开发工具也应运而生。人们称这种研制建造专家系统的辅助开发工具，即为建造某一类专家系统的通用软件工具，为专家系统开发工具。

二、专家系统开发工具的功能和构成

专家系统开发工具能够大大加快专家系统的开发过程，包括迅速原型化，逐步结构化和知识工程化。

迅速原型化使开发者们能够很快查明自己选用的工具和路线是否正确。逐步结构化使开发者们集中精力注意待解问题的实质，而不是它们的形式。知识工程化是在工具内部定义了知识表达方式，做好了开发机和推理机，从而缓和了领域专家不熟悉计算机、知识工程师不了解领域知识的困境。

此外，它还具有解释能力；它的解释程序可协助用户寻找推理路线，检查静态知识库；系统内具有各种高级的知识库编辑程序。

通常，专家系统开发工具由四部分组成：

1、推理机，根据一定的搜索方式、事实和知识库中的规则，进行搜索、

推断，得出结论，向专家系统的使用者作专业咨询。

2、开发机；它通过自然语言获得知识，生成知识库，修改已有的知识，从而完成知识的更新和学习。

3、解释机，必要时，它根据搜索过程中所涉及的事实和规则，向用户解释推理的依据和过程，从而使用户能根据理由的充足与否决定是否接受系统的咨询意见。

4、人-机界面；它为专家系统开发工具中上述三者与用户之间提供良好的交互环境，确保开发工具的正常工。

图 1、3 给出了专家系统开发工具的构成示意图。

图 1、4 提供了专家系统开发工具的工程化图。

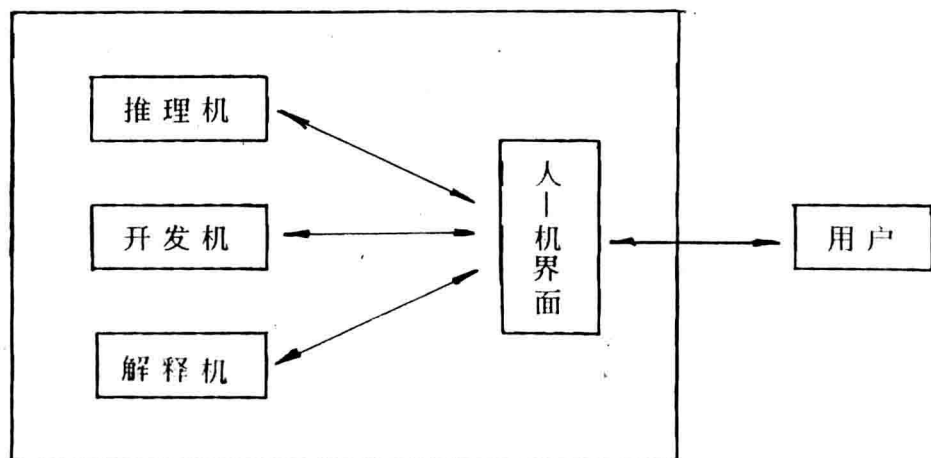


图 1.3 专家系统开发工具结构图

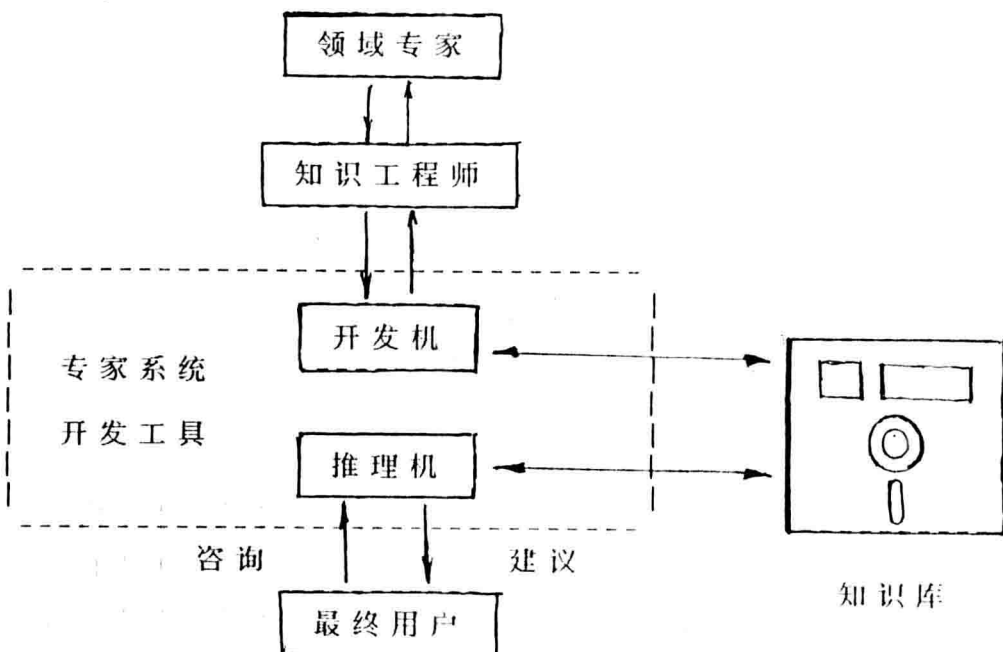


图 1.4 开发工程图

三、现有专家系统开发工具分析

目前流行的专家系统开发工具不少，其中有的是从已有专家系统中抽去领域知识，并作若干修改、扩充和一般化工作之后提炼而成的，适用于限定类型的专家系统的开发的骨架型专家系统开发工具。有的是不依赖于任何已有的专家系统，完全重新设计并提供更多灵活性的，适合于各种类型的专家系统的开发工具。有的则是由帮助获取和表达领域知识的模块，以及帮助设计和建造不同类型的专家系统的模块组成的辅助型专家系统开发工具。

这些专家系统开发工具在组织结构、推理方式、适用领域和支撑机型等方面各具特点，用它们开发专家系统要比用高级语言容易得多。但目前的开发对多数领域专家来说仍很困难；理想的专家系统开发工具，应能使领域专家在没有知识工程师帮助的情况下建造专家系统，这也是人工智能的研究者们为之努力的目标。

第二章 建立小型、多功能、通用的专家系统开发工具的基本设想

在这章中，我们将就EST 这种小型、多功能、通用的专家系统开发工具的基本构成进行规划，这种规划包括了整个系统的结构，各部分的组成和推理机的搜索策略。而这一切的依据是该专家系统开发工具应具有什么样的显著特点；它要解决的关键问题有哪些，以及如何去解决。

§ 1：建立EST的基本思想

一、总体结构的设定

前面已经提到，一个专家系统开发工具主要由四部分组成，即推理机、开发机、解释机和人一机界面，同时，它又要求知识库中的知识表达必须要以它能够接受的方式进行，这就意味着开发机事实上还必须包括关于知识表达方法的约定，这实际上是规定产生式规则的格式，EST 也不例外。图 2：1 给出了EST 的总体结构，用户通过接口与推理机进行搜索过程中的信息交互，利用开发机建立知识库或修改规则，利用解释机解答用户的咨询。

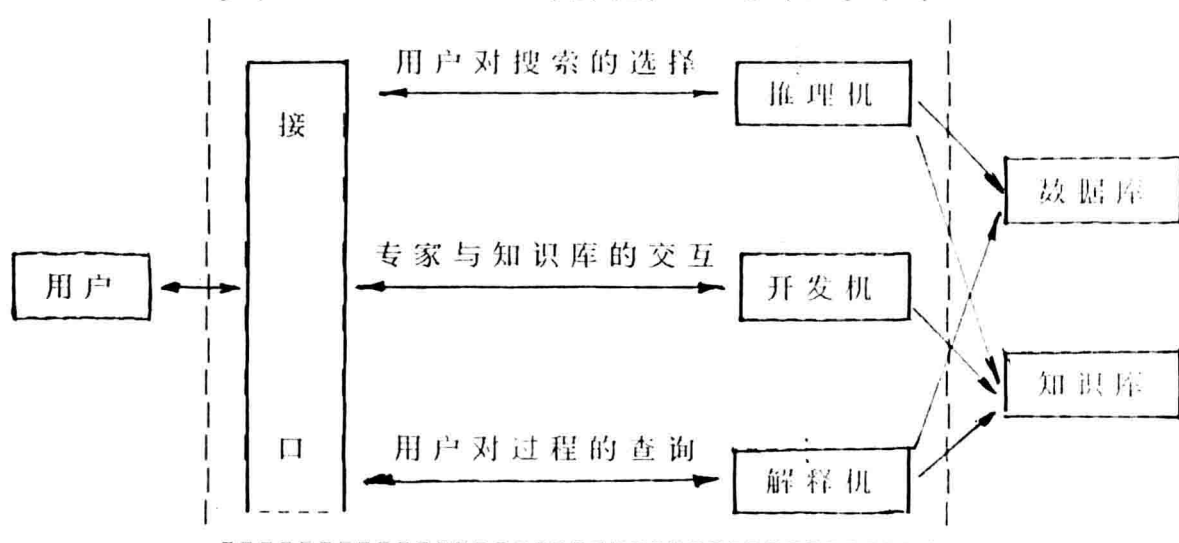


图 2.1 EST 的总体结构图

二、开发机及知识的表达

EST 中开发机的职能是完成对完整专家系统的开发，换句话说，即建立起合理、有效、系统能够接受的知识库，并能对知识库进行增加、修改和删除等操作。开发机在能够进行正常操作之前，必须接受对以规则表示的知识的表达方式的约定，即规则的格式的选择和确定。

知识库中的知识常用产生式规则来表示；这是因为这种方式直观而且可扩充。在确定了规则的形式后，开发机还须能够进行知识库的建造和修改，或者说是知识的学习和更新。这实际上是一系列对知识库的操作，这些操作是：

① 将已确立的知识库装入，构成一个较完整的专家系统。这个操作包括判断这个选定的知识库是否存在，在确信没有错误的情况下将知识库装入。

② 应用户要求显示指定的规则内容。当然，这种显示并非简单地写出规则本身，而是用通俗的语言重新表达。

③ 增加规则。增加规则要考虑到新规则对旧规则的影响。

④ 修改规则。因为任何知识库都需不断地更新，因而修改规则和增加规则一样是一种不可缺少的操作。修改的过程实际上是删除旧规则，增加新规则的过程。

⑤ 删除规则。删除规则要考虑对规则优先次序的影响。

⑥ 知识库的存储。知识库可由用户指定名字，以文件方式存入；也可以缺省方式存入系统默认的文件中。通常这个默认文件名就是装入的知识库文件名。

⑦ 搜索过程中产生的动态数据库的存储。这在某些场合也会被要求。我们可以用类似于知识库存入的方式进行，即存入用户指定的文件中。

三、推理机和搜索策略

无论是对专家系统，还是对开发工具来说，推理机都是其核心部分。推理机及其搜索效率直接决定着工作的成败；而推理机最根本的部分是控制策略的选择和实现。

在建造知识库时，把可能出现的每个事实当作一个结点，则所有的事实就构成一个事实空间。一条产生式规则是建立在两个事实之间的一种关系，所有规则就在事实空间中建立一有向关系网或关系树；关系网中的结点可有一个以上的父结点，而关系树中的结点只能有一个父结点。问题的求解，即搜索的过程，就是找到一条连接已知结点到目标结点的途径，这实际上是一个图的搜索问题。搜索算法的好坏，对问题求解的效率有很大影响。

根据搜索的方向，可分为正向搜索和逆向搜索两种算法，相应的推理分为正向推理和逆向推理。正向搜索从已知状态出发，以已知状态为树根结点，找

出与左部结点匹配的所有规则，用其右部建立新构型，从而生成一层树。取出上层树中的每一个结点，再生成下一层树，如此继续，直到匹配到目标状态的构型生成或规则用尽为止。逆向搜索则是在事实空间中，由所有可能的事实构成的事实树或事实关系网，从目标状态出发，找出右部匹配根结点的所有规则来，略去不匹配的那些分枝，从而进入上一层树，取出生成的每个结点，再进入上一层树，如此继续，直到匹配到已知状态的构型生成或规则用完为止。

根据搜索顺序选择的不同，搜索又可分为深度优先搜索和广度优先搜索。深度优先搜索是在事实空间中从某一分支搜索，直到产生某一解为止，或在到达预定的某一深度后再回去搜索别的分支。而广度优先搜索则是在事实空间中从某一层搜索，直到产生某一解为止，或在搜索完这一层之后再向下一层继续搜索。通常对于有明显层次性的知识，采用深度优先搜索，我们的选择也是这样。

要在状态空间的盲目搜索中找到一个解，所需扩展的结点数目可能很大，因为这些结点的扩展次序完全是随意的，而且没有应用已解决的问题的任何特性，因而除非问题很简单，一般都要占用很多时间和空间，这将会导致组合爆炸，这是应该尽力避免的，为有效求解，通常需平衡灵活性和系统性这两个不同的要求来构造一种控制策略，这时，可以借助具体问题的信息来简化搜索，这种信息叫做启发式信息，相应地，这种搜索叫做启发式搜索。其用途有二：

- ① 决定要扩展的下个结点，避免搜索中的盲目性；
- ② 在扩展结点过程中，决定要生成的可能的后继结点，以免生成全部结点；
- ③ 决定某些应从搜索树中抛弃的结点。

鉴于此，我们选用启发式的深度优先的正向和逆向搜索算法作为EST 的搜索策略，在同一推理机中用两种不同的搜索策略，是为了适应不同的知识系统的需要，使EST 具有通用性。只要在使用EST 时选择合适的搜索策略，推理将是合理和有效的。

四、解释机

如果说开发机是用户用于对专家系统内部输入的工具，那么解释机则完全是专家系统应用户的要求对外输出信息的工具。解释机主要完成以下几部分的工作：

- ① 对规则作解释；手段是借助于开发机中规则的显示。
- ② 在搜索结束以后，可以回答用户HOW 类的问题，即解释推理机搜索的路径。对正向推理来说，是从事实空间有向树上已知结点出发，到目标结点之间一系列的正向枝的端点结点的序列；对逆向推理来说，是从事实关系网中设定

的目标状态结点到已知状态结点之间一系列的反向枝的端点结点的序列。从这些结点序列的顺序显示，用户得以知道推理机是如何搜索的，这也就对HOW问题作出了答复。

③ 在搜索结束之后，可以为用户解答WHY类的问题，也就是解释推理机搜索算法的依据，显然这种依据是一系列规则的集合，无论是正向推理还是逆向推理，推理机依据这些规则进行搜索。这些有序规则的显示，使用户明白推理机搜索的理由，这就对用户WHY类的问题作出了回答。

五、人—机界面——接口

接口是用户（包括领域专家）与开发工具进行信息交互的媒介，它完成以下几项任务：

- ① 完成整个工具的驱动。
- ② 领域专家和知识库的接口，提供增加、修改和删除规则的交互环境。
- ③ 用户与推理机的接口，通过它，用户回答对启发式信息的提问，并随时跟踪搜索过程。
- ④ 用户与解释机的接口。

这些任务，要求接口具有驱动能力强，使用灵活，路径明确，出入自如等特点，为此，我们选择简明而直观的动态菜单驱动的方法，和与之配套的一系列驱动谓词来完成这个接口。

§ 2：问题和对策

一、不同搜索策略的并存和统一

为使EST的推理机功能加强，我们选择正向和逆向两种不同的搜索策略，其中正向搜索还分为带回溯和不带回溯两种，问题是怎样使这三种算法并存且能相互独立，对此，我们用三个独立的谓词来驱动，并且保证三种搜索的中间过程相对独立。这样，虽存在于同一个程序中，三个独立的谓词就相当于三个相互独立的函数，并存而互不干扰。

三种搜索算法并存，并不意味着需要三种不同的产生式规则来表达各自的知识体系。一般地讲，适合正向搜索和逆向搜索的知识体系是有所不同的。为使不同的知识体系对应的推理算法能在规则意义上得到统一，EST必须解决知识的通用表达方式的问题。

二、知识的通用表达式

为尝试能否做到这点，我们先作如下的分析：

一般的产生式规则可以表示为“IF 前件 THEN 后件”的形式，我们称

之为表达式 1。这里的前件是一个可以通过数据库中的数据来确定其真伪的谓词，后件则具有两重任务，一个是实施对数据库的修改，另一个是它本身还可以是一个谓词。一条规则的实施意味着其后件谓词一定为真。后件是个谓词，对于采用逆向搜索的推理机特别有用。在某些场合，规则还带有操作部分，这样会使规则的意义更加明确。这时规则的形式就变成“IF 前件 THEN 后件 ACTION 操作”；我们称此为表达式 2。

为了使规则的形式具有一般性，即不仅能适用于推理时带有操作的知识体系；通常这种知识多与逆向推理相联系，同时还能适用于推理时不带操作的知识体系；我们把表达式 2 中的 ACTION 操作同样视为一个谓词。显然，这个谓词和前件谓词在某种程度上有共性；即一旦前件谓词为真那么为使后件谓词为真；操作谓词必须也为真；反之，若前件谓词失败，规则不起作用，操作谓词也不能为真；即为考察后件谓词是否能为真，须依次考察前件谓词和操作谓词的真伪；我们不妨把这两部分进行合并，规则表达式 2 变成“IF 前件 and 操作 THEN 后件”的形式，称为表达式 3。而且我们还可以将两者都视作前件谓词部分；而只在不同的推理机制中加以区别；即在不带操作的推理中，这部分只起前件谓词的作用；在带操作的推理中，这部分则起两种作用。这样，表达式 1 和表达式 2 统一地可用表达式 3 表示。

表达式 3 统一了不同推理机制对规则形式的要求，因而更具普遍性。我们选择表达式 3 的形式作为开发机中对规则形式的约定，即对知识的通用表达方式。

三、修改知识库带来的问题

上节中已经提到；无论是对知识库的规则作增加、删除还是修改，都会遇到的一个问题是；这些操作对知识库的优先级的影响。为了查询的方便，知识库中的规则是依次排列的。这种次序通常还具有影响搜索次序；或者说是知识库中规则优先级的作用。为此，对规则的操作会带来对规则重新排序的问题。

对知识库中的规则的优先级最简单的处理是；将匹配同一结点的不同规则根据优先级由高到低依次排列。在增加规则时，一般不是任意地插入规则队列中；而是加在其尾部；在删除规则时，若该规则不在队尾；删除后；其后的规则向前移一位。

增加规则时；还必须考虑新规则与旧规则的关系。首先，不能重复已有的规则；其次，不允许两条规则互为前后件，因为它会带来搜索的死循环；最后还要纠正规则输入过程中可能出现的误操作