

第五章 专家系统

主要内容:

§ 5.1 专家系统基本概念

§ 5.2 基于规则的专家系统

§ 5.3 基于框架的专家系统

§ 5.4 基于模型的专家系统

§ 5.5 基于Web的专家系统

专家系统

5.1 专家系统基本概念

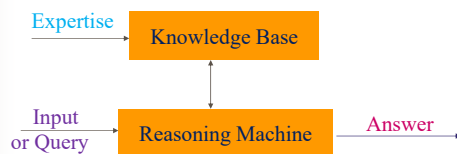
1. **专家系统**: 一个智能计算机程序系统, 内部含有某个领域专家水平的知识与经验, 能够利用人类专家的知识解决问题的经验来处理该领域的难题。

专家系统已具备这4个要素



专家系统结构

2. Structure Architecture of Expert System



Simplified block diagram of expert system

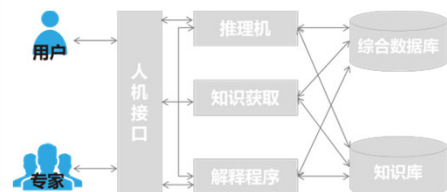
专家系统分类

3. 完整的专家系统主要包括 ES两个基本模块

- Knowledge Base
- Reasoning Machine
- User Interface
- Explainer

知识库 (Knowledge Base)

推理机 (Inference Engine)



知识库

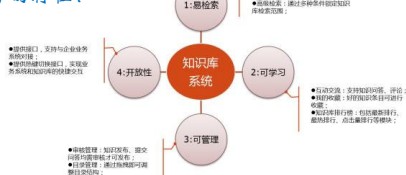
■ 知识库 (Knowledge Base)

存储某领域知识, 包括事实、可行操作与规则。为建立知识库, 要解决知识获取和知识表示。

知识获取: 从专家那里获得专门知识;

知识表示: 用计算机能理解的形式表达和存储知识。

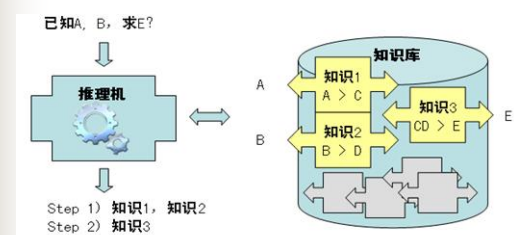
知识库具备的特性:



推理机

■ 推理机 (Reasoning machine)

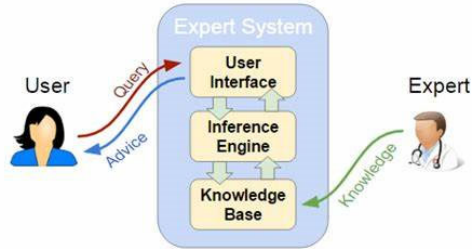
记忆所采用的规则和控制策略的程序, 能够根据知识进行推理和导出结论。



解释器

■ 解释器 (Explanator)

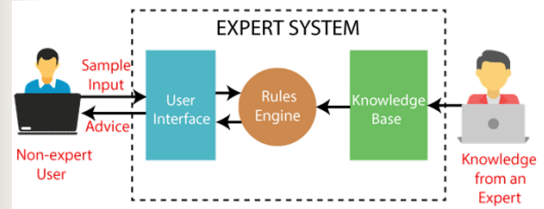
向用户解释专家系统的行为，包括解释推理结论的正确性以及系统输出其它候选解的原因。



接口

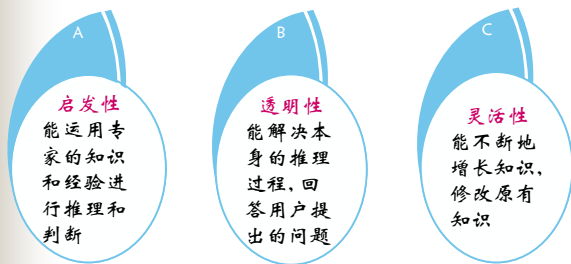
■ 接口 (Interface)

又称界面，它使系统与用户进行对话，使用户能够输入数据、提出问题和了解推理过程及推理结果。系统通过接口，要求用户回答提问，回答用户提出的问题



专家系统的特点

4. 专家系统的特点

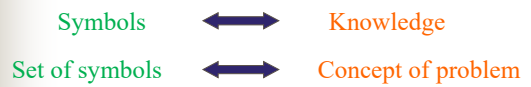


专家系统的特点

4. 专家系统的特点

■ 符号操作

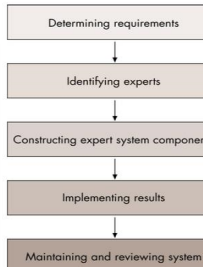
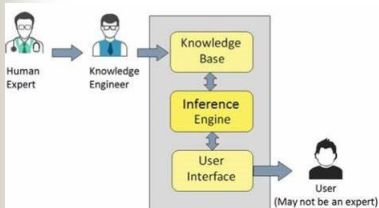
与常规程序进行数据处理和数字计算不同，专家系统强调符号处理和符号运算，用符号表示知识，用符号集合表示问题的概念。



一般步骤

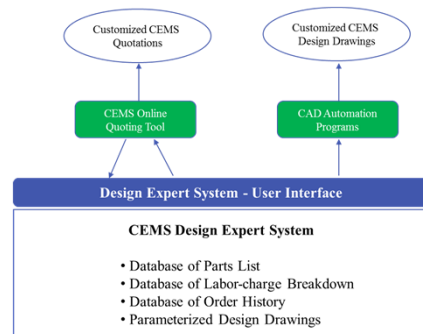
5. 建立专家系统的一般步骤:

- 设计初始知识库
- 原型机 (Prototype) 的开发与实验
- 知识库的改进与归纳



设计软件

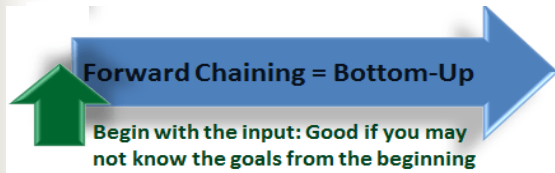
6. 专家系统设计软件



7. 专家系统工作方式

链接 (Chaining)

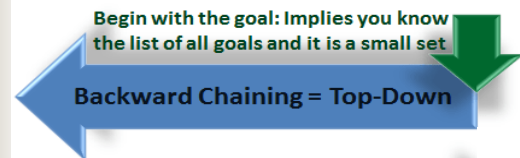
前向链接: 从可用的数据开始, 并使用推理规则来归纳更多的数据, 达到所需的目标。



7. 专家系统工作方式

链接 (Chaining)

反向链接: 从一个目标列表开始, 反向工作, 看看是否有数据可以让他得出这些目标中的任何一个。



8. 专家系统模型分类



基于规则的专家系统



基于框架的专家系统

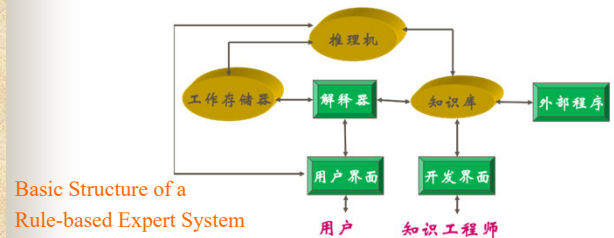


基于模型的专家系统

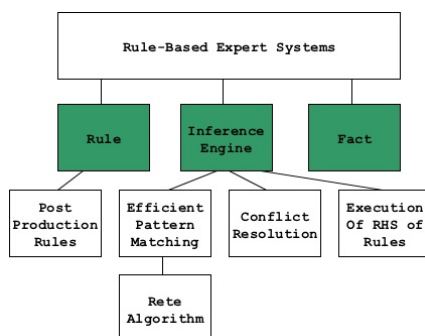
5.2 基于规则的专家系统

1. 基于规则ES的定义

根据专家的经验, 将其归纳成规则, 通过启发式经验知识进行推理, 具有明确的前提, 得到确定的结果。



Architecture of Rule-based Expert System



Expert System Components

■ Inference engine

Forward or Backward-chaining
Conflict resolution algorithms

■ Rule-based

IF-THEN rules

■ Database

Current state on which IF-THEN rules are applied.

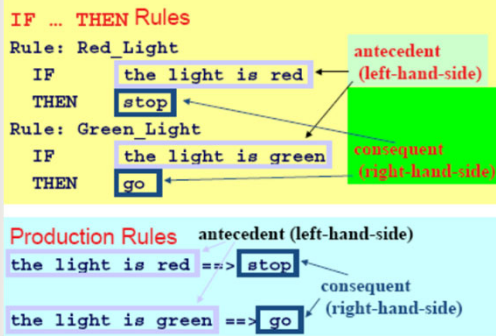
■ Explanation facilities

An important advantage rule-based expert systems hold over other types of AI

IF	the 'traffic light' is green
THEN	the action is go
IF	the 'traffic light' is red
THEN	the action is stop

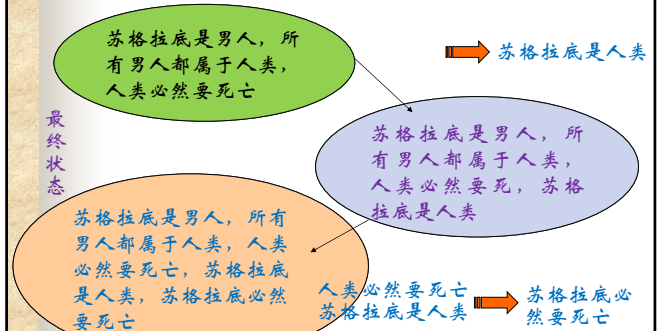
Example

Example:



Example

Example:



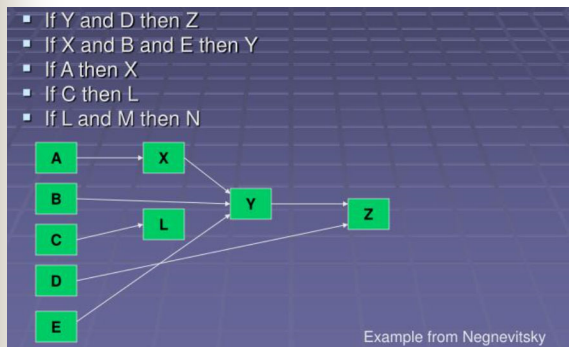
Inference Engines

Inference Engines

- **Forward-chaining**
 - Submit current data to all rules
 - Rules make conclusions, which in turn, generate new data.
 - "Inference Chains" result from initial data and the data generated conclusions.
- **Backward-chaining**
 - Try to prove a conclusion by working backwards from ways to prove it.

Forward Chaining

Forward Chaining Example (A, B, E, and D are given)



Backward Chaining

Backward Chaining

- **Prolog uses backward chaining**
 - Work backward from the goal
 - Check rules that can provided the desired goal.

Chaining

Forward or Backward Chaining

- What do experts use?
- Are we trying to prove a particular hypothesis?
Backward chaining
- Are we trying to find all possible conclusions?
Forward chaining
- What does the **rule set** look like?
 Could be either one or a combination of both

Conflict

Conflict Resolution

- What happens when two rules provide conflicting conclusions?
 - If it has feather **then** it is a bird
 - If it can't fly **then** it is not a bird
 - What if has feathers, but can't fly?

Conflict

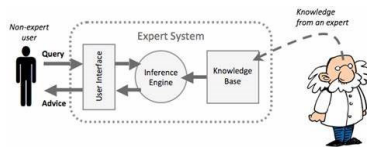
Conflict Resolution Methods

- Use rule-order as an implied priority
 - The first rule to provide an answer is used
- Assign a priority to each rule, the rule with the higher priority is sustained
- **Longest Matching Strategy** uses the rule with the most specific information
 - If it cannot fly and has feathers then it is a bird
- Certainty-based conflict resolution

Disadvantages

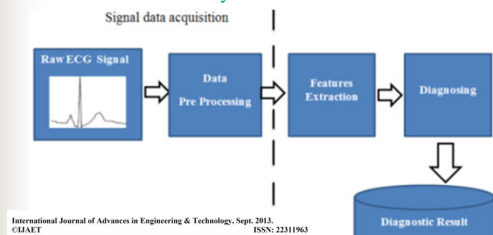
Disadvantages of Rule-based Expert Systems

Inability to learn. In general, it do not have an ability to learn from the experience. Unlike a human expert, who knows when to “**break the rules**”, an expert system can't automatically modify its knowledge base, or adjust existing rules or add new ones. The knowledge engineer is still responsible for revising and maintaining the system.



A Rule-Based Expert System for Automated ECG Diagnosis

General Diagram of Electrocardiogram Analysis



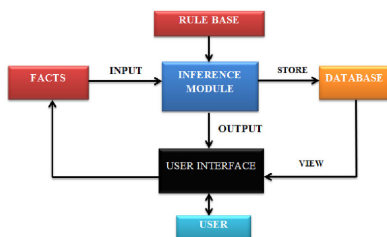
International Journal of Advances in Engineering & Technology, Sept. 2013, ISSN: 22311963

A RULE-BASED EXPERT SYSTEM FOR AUTOMATED ECG DIAGNOSIS

Muzhir Shaban Al-Ani and Aliaf Ayal Rawi
University of Al-Anbar, College of Computer, Anbar, Iraq

Example

Architecture of the rule based expert system



Emulates the ECG interpretation skills of an expert cardiologist for introducing way of automating the diagnosis of cardiac disorders

Example



Normal ECG signal



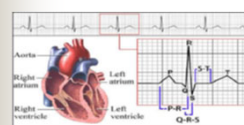
Sinus Bradycardia ECG Signal



Sinus Tachycardia ECG Signal



Sinus Arrhythmia



举例：下面是一个描述“教师”的框架：

框架名：<教师>

类属：<知识分子>

工作：范围：(教学，科研)

槽名：缺省：教学

性别：(男，女)

学历：(中师，高师)

类型：(<小学教师>，<中学教师>，<大学教师>)

“<知识分子> 又是一个框架名”

“缺省”就是侧面名，其后是侧面值

槽名的右面是值

Example: 下面是一个描述“大学教师”的框架：

框架名：<大学教师>

类属：<教师>

学历：(学士，硕士，博士)

专业：<学科专业>

职称：(助教，讲师，副教授，教授)

外语：语种：范围：(英，法，日，俄，德，...)

缺省：英

水平：(优，良，中，差)，



Frame-based ES

3. General Design Methods of Frame-based ES

- 设计步骤与基于规则的专家系统相似, 差别在于如如何看待和使用知识。
- 设计时，把整个问题和每件事想像为编织起来的事物。
- 在辨识事物之后，寻找把这些事物组织起来的方法
- 对于任何类型的专家系统，其设计是个高度交互的过程。

Buy Smart

基于框架的专家系统实例：Buy Smart

基于框架系统的开发主要包含的步骤：

1. 指定问题并定义系统的范围。
2. 定义类及其属性，3. 定义实例。
4. 设计显示。
5. 定义WHEN CHANGED和WHEN NEEDED方法及守护程序。
6. 定义规则。
7. 评估并扩展系统。

步骤1:

指定问题并定义系统的范围

先收集本地区的一些待售房产的信息。要确定相关的细节，例如房产类型、位置、卧室和浴室的数量，当然还有价格。应该提供关于房产的简短描述和精美照片。

步骤1: (续)

接下来就要列出能想到的所有查询:

- 你最多想在房产上花费多少钱?
- 你最喜欢什么类型的房产?
- 你喜欢生活在哪个地区?
- 你想要几个卧室?
- 你想要几个浴室?

步骤2:

定义类及其属性

- 从通用的或概念上的类开始。例如, 分析房产的概念并描述对大多数房产都适用的通用特征。可以透过位置、价格、类型、卧室和浴室的数量、结构、图片和描述等特点来介绍每一个房源。
- 还需要列出联系方法, 例如房子的地址或电话。

Property 类

CLASS: Property
[Str] Area:
[Str] Suburb:
[N] Price:
[Str] Type:
[N] Bedrooms:
[N] Bathrooms:
[Str] Construction:
[Str] Phone:
[Str] Pictfile:
[Str] Textfile:
[N] Instance Number:

步骤3:

定义实例

- 一旦确定了Property 类框架, 就可以很容易地透过使用储存在数据库中的数据建立它的实例。
- 对于大多数的基于框架的专家系统而言, 需要告诉系统我们想要建立一个新实例。

步骤3: (续)

例如, 可以使用下面的代码来建立在Level5 Object 中的Property 类的新实例:

MAKE Property

```
WITH Area := area OF dB3 HOUSE 1
WITH Suburb := suburb OF dB3 HOUSE 1
WITH Price := price OF dB3 HOUSE 1
WITH Type := type OF dB3 HOUSE 1
WITH Bedrooms := bedrooms OF dB3 HOUSE 1
WITH Bathrooms := bathrooms OF dB3 HOUSE 1
WITH Construction := construct OF dB3 HOUSE 1
WITH Phone := phone OF dB3 HOUSE 1
WITH Pictfile := pictfile OF dB3 HOUSE 1
WITH Textfile := textfile OF dB3 HOUSE 1
WITH Instance Number := Current Instance Number
```

Property 的实例



weatherboard house in brisbane



beautiful brick house, bruges...

INSTANCE: Property 1
Class: Property
[Str] Area: Central Suburbs
[Str] Suburb: New Town
[N] Price: 164000
[Str] Type: House
[N] Bedrooms: 3
[N] Bathrooms: 1
[Str] Construction: Weatherboard
[Str] Phone: (03) 6226 4212
[Str] Pictfile: house01.bmp
[Str] Textfile: house01.txt
[N] Instance Number: 1

INSTANCE: Property 2
Class: Property
[Str] Area: Central Suburbs
[Str] Suburb: Tarroona
[N] Price: 150000
[Str] Type: House
[N] Bedrooms: 3
[N] Bathrooms: 1
[Str] Construction: Brick
[Str] Phone: (03) 6226 1416
[Str] Pictfile: house02.bmp
[Str] Textfile: house02.txt
[N] Instance Number: 2

步骤4:

设计显示

需要在每个应用开始时给出 应用程序标题显示，告诉用户一些通用的信息。显示界面可以包含应用程序标题、问题的通用描述、有代表性的图片和版权信息。

应用程序标题显示



步骤4: (续)

可以想到的第二个显示界面是 查询显示。这个界面应该允许使用者通过回答专家系统提出的问题显示我们的要求。

查询显示

Suburb	Bedrooms	Maximum Price
All Suburbs	Any number of bedrooms	No maximum
Central Suburbs	One bedroom	\$50,000
Eastern Shore	Two bedrooms	\$100,000
Northern Suburbs	Three bedrooms	\$150,000
Southern Suburbs	Four bedrooms or more	\$200,000
		\$250,000
		\$300,000
		\$350,000
		\$400,000
		\$500,000
		\$1,000,000
		\$2,000,000

Property Type	Bathrooms
All property types	Any number of bathrooms
House	One bathroom
Unit	Two bathrooms
Townhouse	Three bathrooms or more

步骤4: (续)

最后应该设计 房产信息显示 界面。该显示界面可以提供一系列合适的房产，并可以移到列表中后一个、前一个、第一个和最后一个房产，还能看到房产的图片和描述。

房产信息显示

Suburb	Price	Type
New Town	164000	House
Taroona	150000	House
North Hobart	127000	House
Taroona	110000	House
Lenah Valley	145000	House
South Hobart	140000	House
South Hobart	115000	House