

专家系统及其应用

(Expert Systems & Applications)

毛文吉

中国科学院自动化研究所

2016年12月

专家系统与知识工程

- **专家系统（ES）**

专家系统是一个智能程序，对那些需要专家知识才能解决的难度较大的特定领域问题提供专家水平的解答

- **ES 主要特征**

- ① 专门领域 ② 符号推理 ③ 解释 ④ 复杂性和困难性
- ⑤ 实用性

- **知识工程（KE）**

知识工程是 AI 学科的一个应用领域，研究如何运用 AI 原理和方法建造专家系统

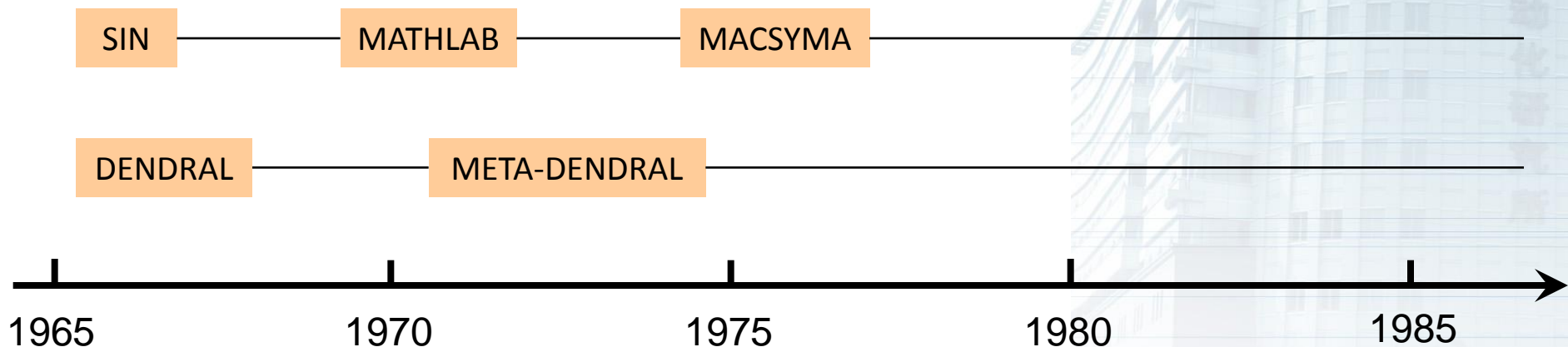
- **KE 研究目标：建造高性能的知识系统**

专家系统的发展（60年代末）

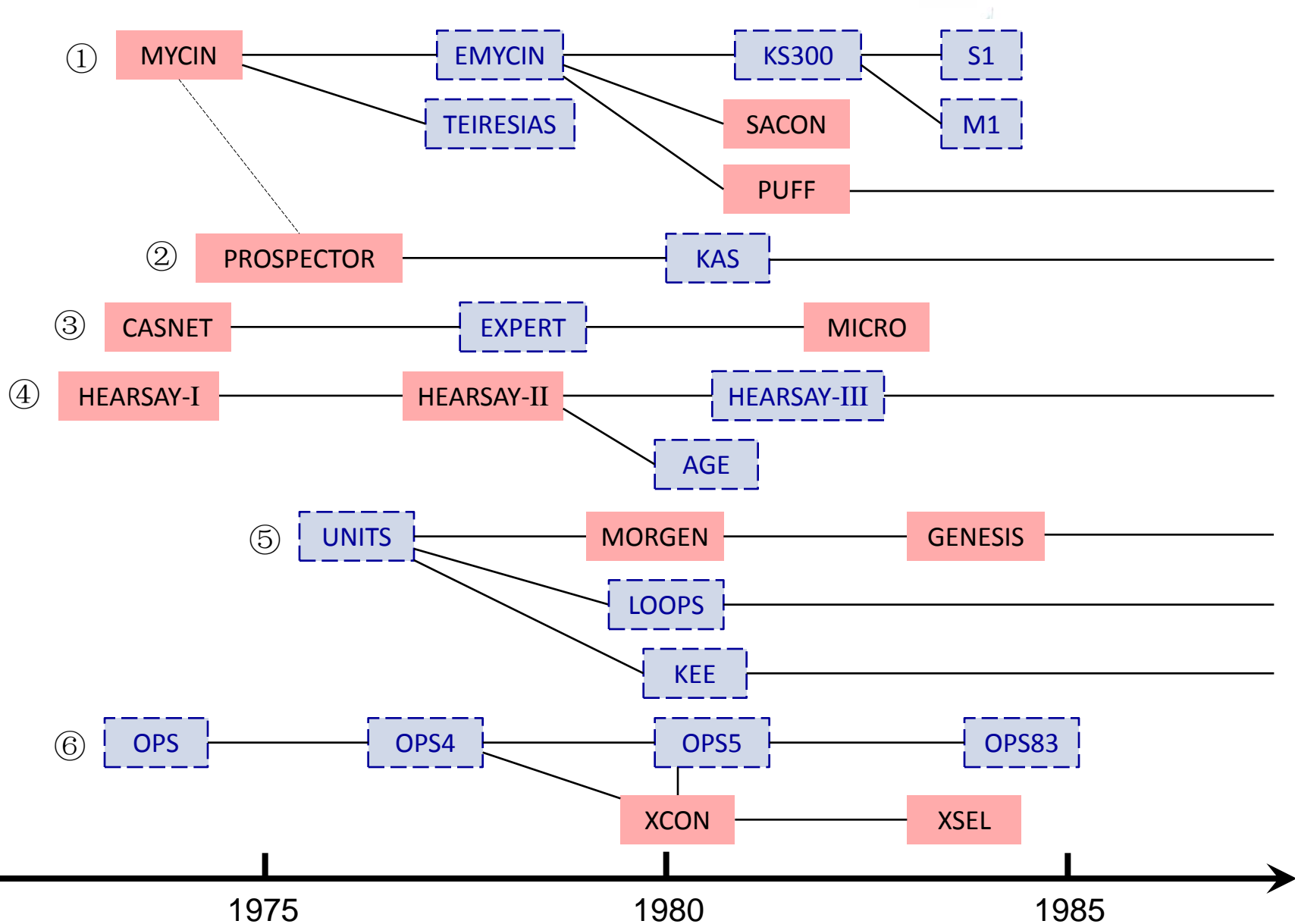
- 第一代专家系统

DENDRAL: 化学质谱分析系统（Stanford 大学 Feigenbaum 等）

MACSYMA: 符号积分与公式化简系统（MIT, Engleman 等）



专家系统的发展（70-80年代）



专家系统的发展（80年代末）

- 与第一代 ES 相比，以下技术方面做了改进：
 - （1）知识的形式化组织技术
 - （2）解释机制
 - （3）知识获取技术
 - （4）不确定推理技术
 - （5）系统的人机接口
 - （6）ES 的通用性研究
- 八十年代末，ES 走向实用化和商品化，发展趋势：
 - （1）ES 开发范围越来越广，开发工具发展较快
 - （2）AI 语言→通用编程语言，面向对象的思想逐渐被采用
 - （3）ES 实用化、商品化中的重要问题研究，对弱点进行改进

专家系统的发展（国内）

- 国内研制开发的专家系统：

施肥专家系统（中科院合肥智能所）

新构造找水专家系统（南京大学）

油气资源评价专家系统（吉林大学）

花布图案设计专家系统（浙江大学）

中医诊断专家系统（中科院软件所）

... ..

- 国内研制的专家系统开发工具（九十年代初）：

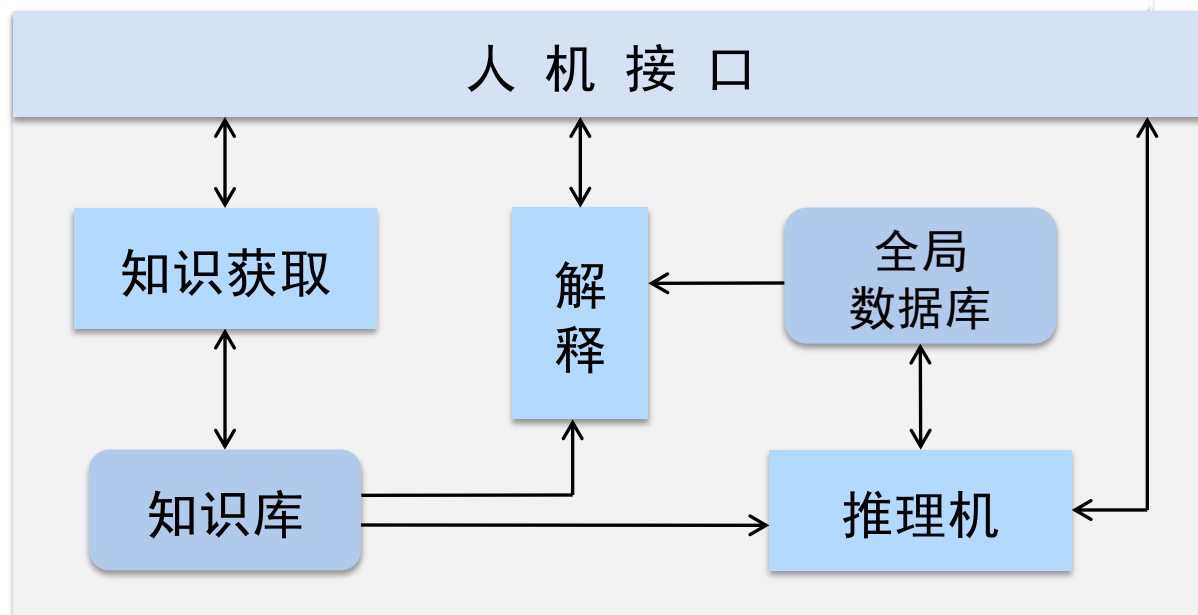
通用型集成式“天马”专家系统开发环境

（中科院、机电部15所、浙江大学、武汉大学等七家单位）

专家系统的分类

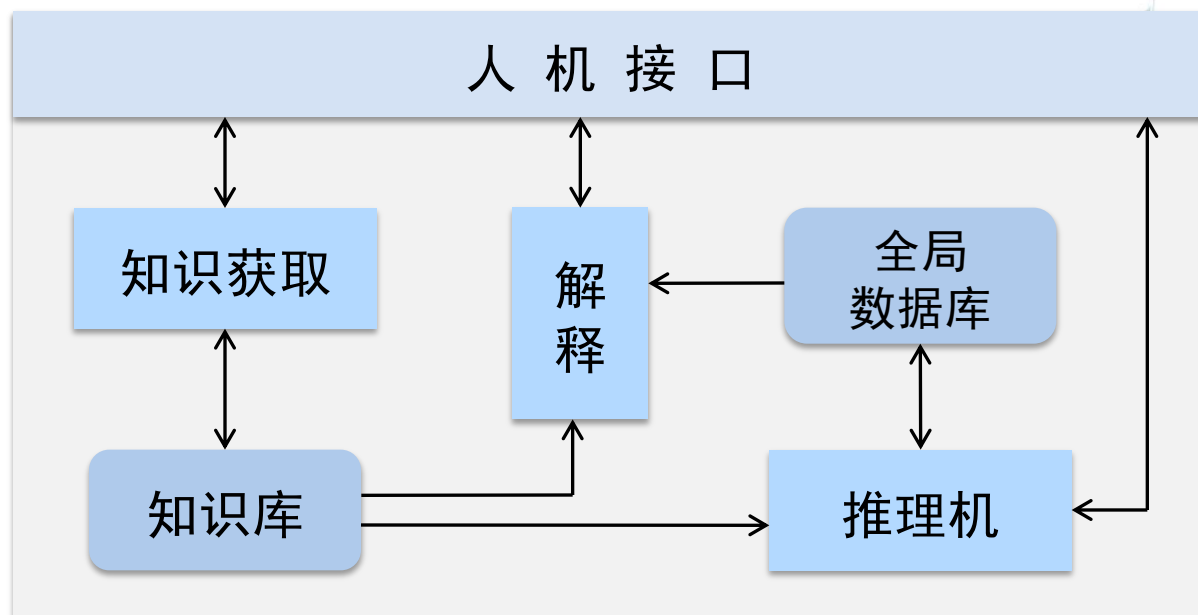
类型	处理的问题	应用
诊断型	根据观察现象诊断系统故障	医疗、电子、机械、软件等
解释型	根据已知数据进行解释分析	语音理解、图像分析、仪器数据分析、信号解释等
预测型	根据已知情况推测可能结果	气象、人口、市场、农作物收成、社会态势、安全预警等
设计型	给出满足约束条件的目标配置	VLSI、建筑、机械、服装、预算等
规划型	给出达到某一目标的动作序列	机器人、自动程序设计、交通运输、军事指挥等
监视型	将观察对象的行为与期望行为比较	核电站、航空、车辆、流行病、社会管理等
教学型	诊断、纠正学生行为	各种智能辅助教学、训练系统、演练环境等
控制型	自适应地管理对象的全部行为
排错型	对故障给出排除方法
修理型	对发生故障的对象进行处理

专家系统的基本结构



- **知识库：**按一定的表示形式存放领域知识（包括专家的经验知识）
- **推理机：**在一定的控制策略下根据数据库中的当前数据，选择知识库中的知识求解问题，将中间结果记录在数据库中，继续推理过程直到求得问题的解

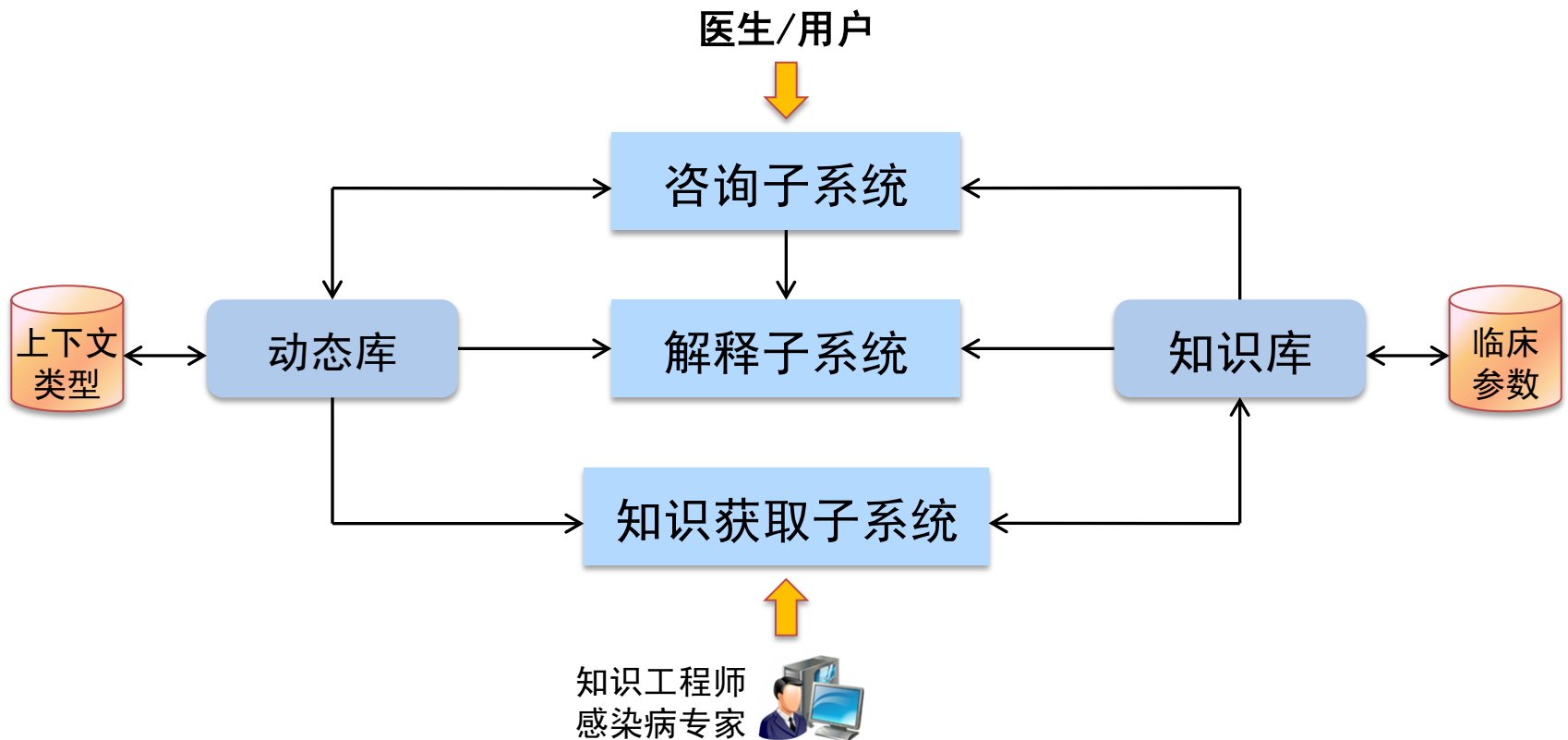
专家系统的基本结构



- **全局数据库**: 记录初始数据、推理的中间结果和最后结果
- **知识获取**: 从知识源提取知识，建立知识库；还包括对知识库进行增删、修改、检查（知识库管理）
- **解释**: 对求解过程进行说明，回答用户提出的问题
- **人机接口**: 人能接受的外部形式 \Leftrightarrow 系统可接受的内部形式

一个专家系统实例：MYCIN

- 用于血液感染疾病的诊断和治疗
- 系统于 1975 年采用 INTERLISP 编写
- 知识库包含 200 多条规则，可以识别约 50 种病菌

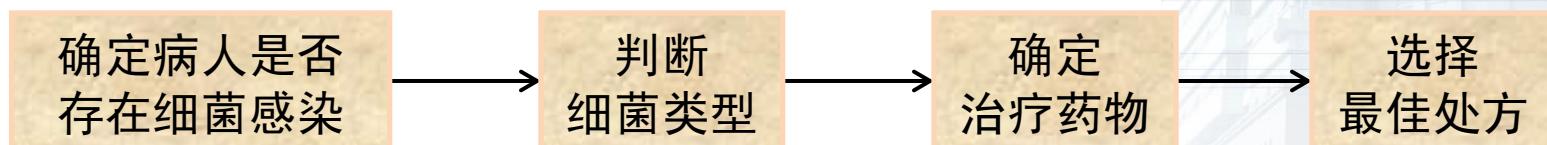


系统设计需求

- 专家对血液感染病的诊断步骤:

- (1) 确定病人有无需要治疗的细菌感染
- (2) 判断疾病可能由哪种细菌引起的
- (3) 确定对这种细菌有抑制作用的药物
- (4) 根据病人情况, 选择最佳治疗方案

- MYCIN 系统的咨询过程:

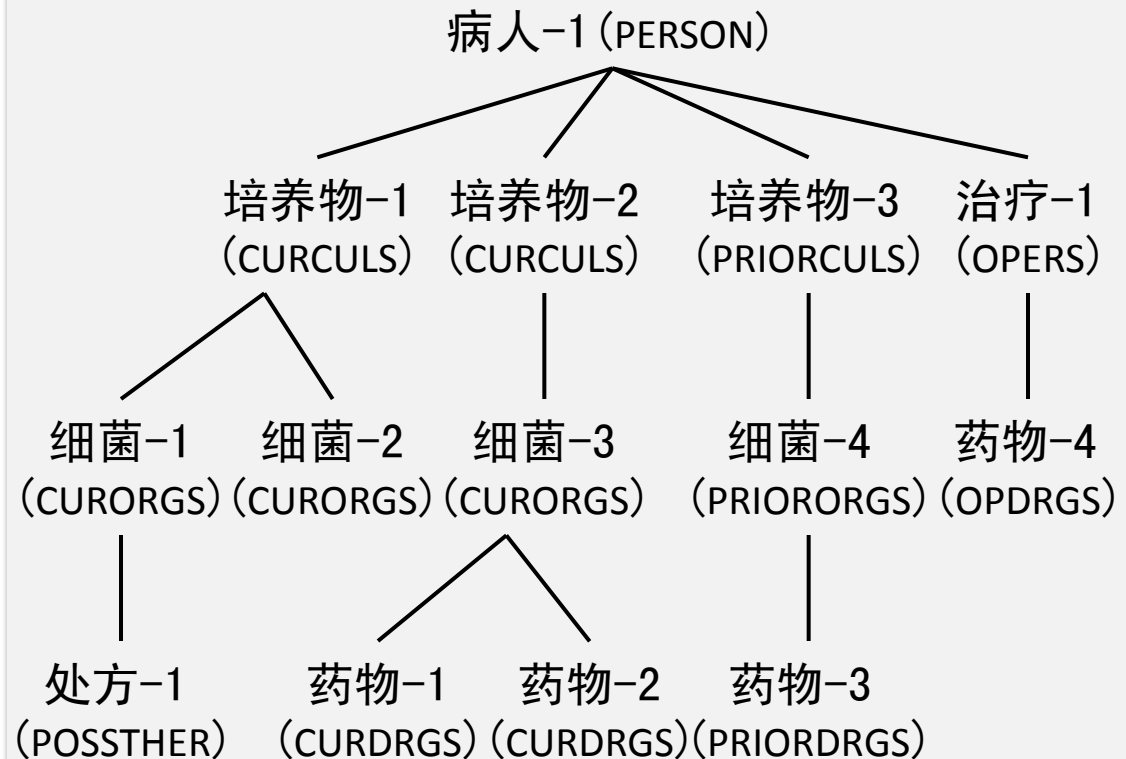


- 通过咨询系统询问病人情况, 将有关数据记录在动态库中
- 根据动态库中的数据, 应用知识库中的可用规则进行推理

数据表示

- 提供十种上下文类型，每种上下文包含一组属性/临床参数
- 采用上下文树描述当前诊断的病人情况
- 咨询过程逐步形成关于病人的完整上下文树

1. 病人 (PERSON)
2. 当前培养物 (CURCULS)
3. 先前培养物 (PRIORCULS)
4. 当前细菌 (CURORGS)
5. 先前细菌 (PRIORORGS)
6. 手术治疗 (OPERS)
7. 手术药物 (OPDRGS)
8. 当前药物 (CURDRGS)
9. 先前药物 (PRIORDRGS)
10. 处方 (POSSTHER)



知识表示

- 规则的前提和结论部分：采用对象-属性-值三元组形式
- 临床参数按对应的上下文分类，参数取单值、是非值或多值
- 每种临床参数用一组特性（特性表）来描述

MEMBEROF: 参数所属的参数组名称

VALUETYPE: 参数的取值类型（单值、是非值、多值）

EXPECT: 所期望的参数取值范围（布尔、数字、表元素、无限制）

... ..

LABDATA: 该特性为真时说明是原始参数，应由用户提供

LOOKAHEAD: 指明哪些规则的前提涉及该参数

UPDATED-BY: 指出从哪些规则的结论部分可得到该参数

... ..

TRANS: 用于把该参数译成英语，便于进行解释

推理策略

- 采用反向推理、深度优先的策略
- 咨询过程的目标：对病人的治疗处方
- 首先使用目标规则 092：

<规则092>：（外部表示形式）

如果：（1）存在一种**需要治疗的病菌**

（2）已经考虑了其它**可能存在的附加病菌**，虽然它们还没有出现在当前培养物中

则：（1）根据病菌对药物的敏感情况，编制一个能有效抑制该病菌的**治疗方案清单**

（2）从已编制的清单中**选择最佳的处方**

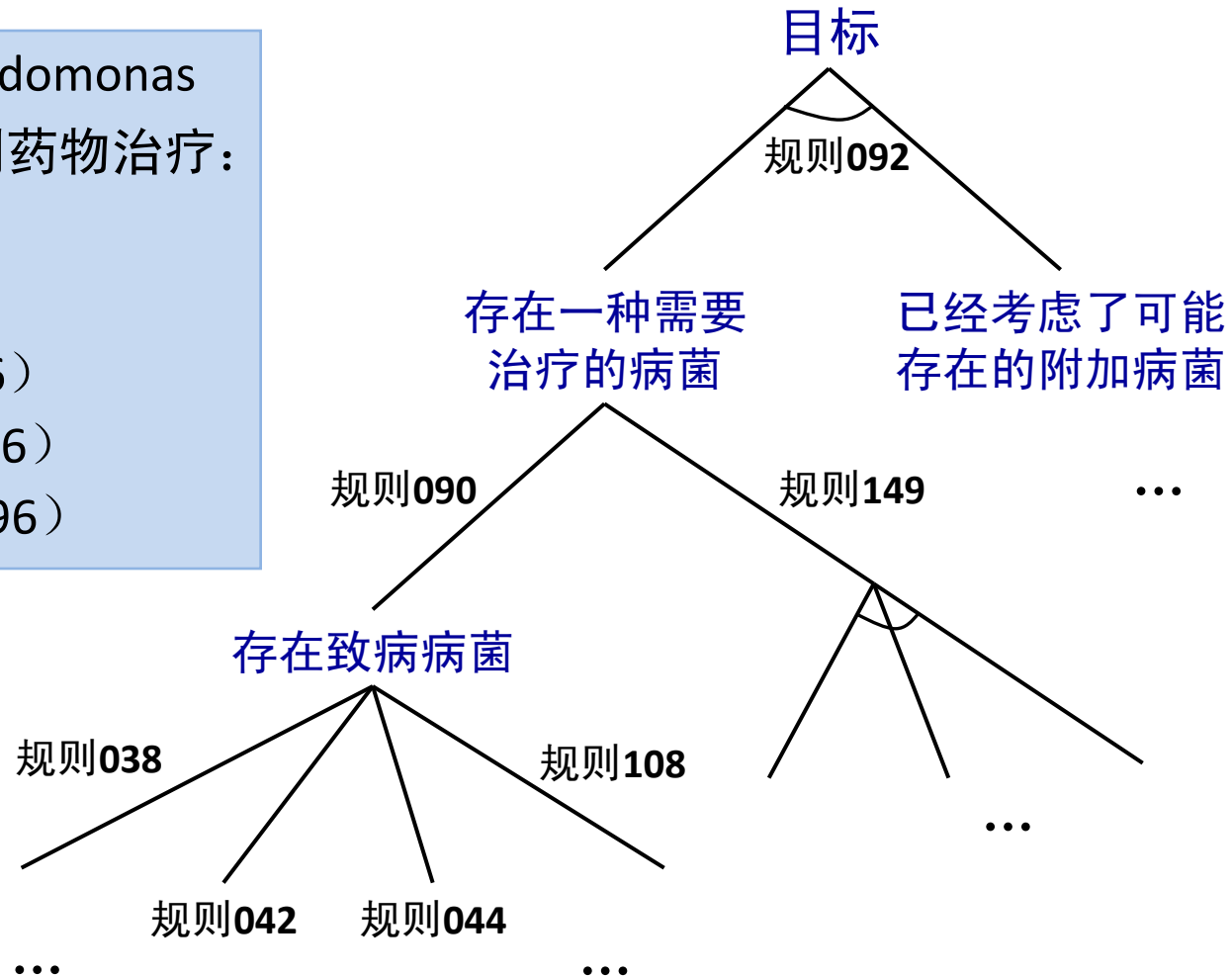
否则：病人不需要治疗

推理过程

- 若目标规则的前提成立，即存在“需要治疗的病菌”，则生成可能的“治疗方案清单”，例如：

IF 细菌类型是 Pseudomonas
THEN 建议选择下列药物治疗：

- colistin (0.98)
- polynyxin (0.96)
- gentamicin (0.96)
- carbenicillin (0.96)
- sulfisoxazole (0.96)



咨询过程

Q. What is the patient's name?

A. Fred Smith

Q. Sex?

A. Male

Q. Age?

A. 55

Q. Do you know the type of the infection?

A. Yes

Q. What is the type of the infection?

A. Blood infection

Q. Have you obtained a blood sample?

A. Yes

Q. What is the morphology of the organism in this sample?

A. Rod

Q. What is the gram stain of the sample?

A. Negative

Q. Is the organism aerobic or anaerobic?

A. Anaerobic

... ..

调用推理规则 293:

IF **gram stain** (革兰氏染色) is **negative** (阴性)
and **morphology** (形态) is **rod** (杆状)
and **aerobicity** is **anaerobic** (生长厌氧)

THEN this evidence suggests that the organism
is **E. Coli** (CF 0.65)

... ..

Q. The organism is probably **E. Coli** (CF 0.9)

It is also possible that it is **Salmonella** (CF 0.1)

解释及用户接口

- 提供两种提问类型：How 和 Why
 - How**: 提问系统是如何得出某个结论的
 - Why**: 提问系统为什么要向用户问某个问题
- 通过用户接口对知识库进行修改和扩充

Q. 有规则丢失吗？

A. Yes

Q. 你想给出它吗？

A. Rule 384

If: 1) The site of the culture is blood

2) ...

3) ...

4) ...

Then: 1) The organism is an E. Coli. (0.4)

Q. 它是关于细菌本性方面的新概念吗？

A. Yes

Q. 请给出 E.Coli 的全名？

A. **ESCHERICEA-COLI**

Q. 给出 **ESCHERICEA-COLI** 的所有同义词或缩写词？

A. E. Coli

Q. 给出 E.Coli 作为细菌本性的简短描述

A. **normal flora in the comer intestine, ...**

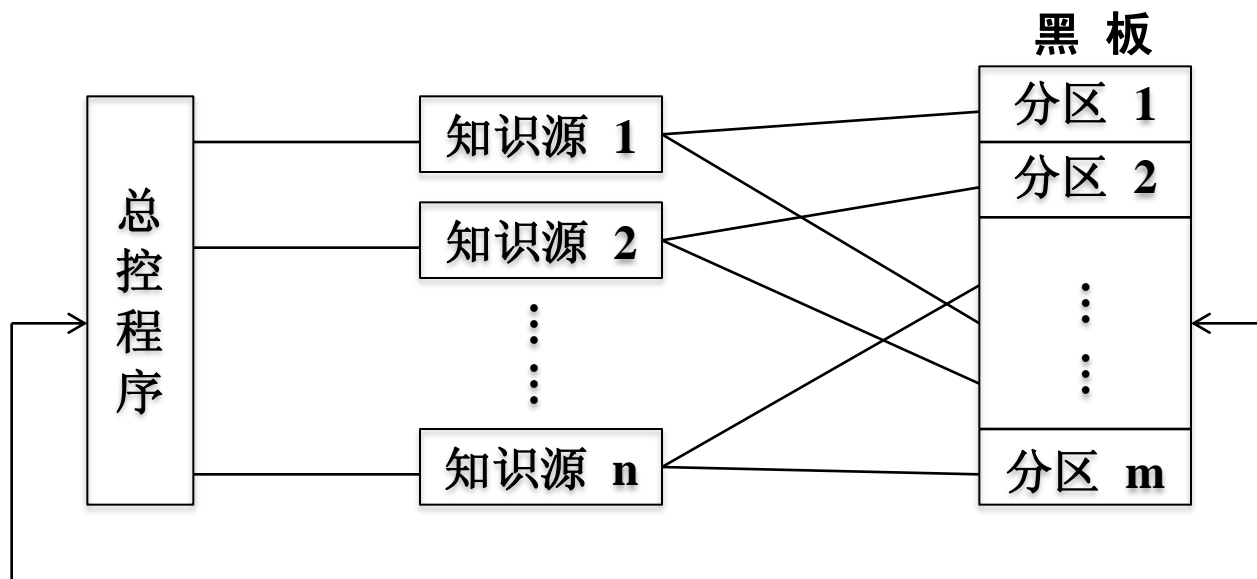
Q. E.Coli 的 aerobicity 是什么？回答形式：
<aerobicity> <certainty factor>

A. **anaerobic** 0.8

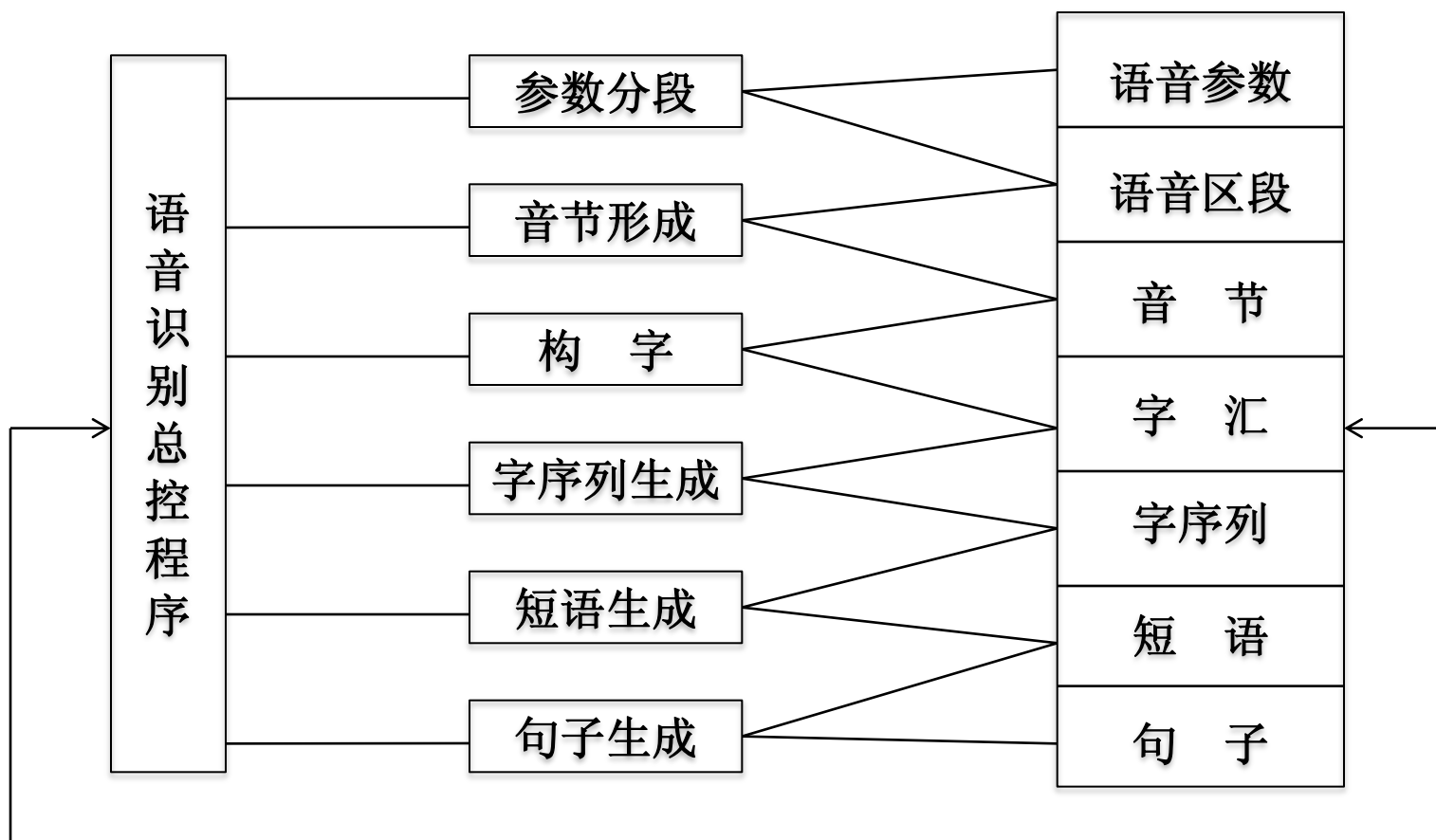
aerobic 0.2

黑板结构

- 复杂的专家系统，需要协调多个知识源进行问题求解
- 将领域知识划分成若干个相互独立的知识模块——知识源
- 黑板实际上是全局数据库，按信息的抽象程度分层；各知识源通过黑板交换信息，共享数据和结果

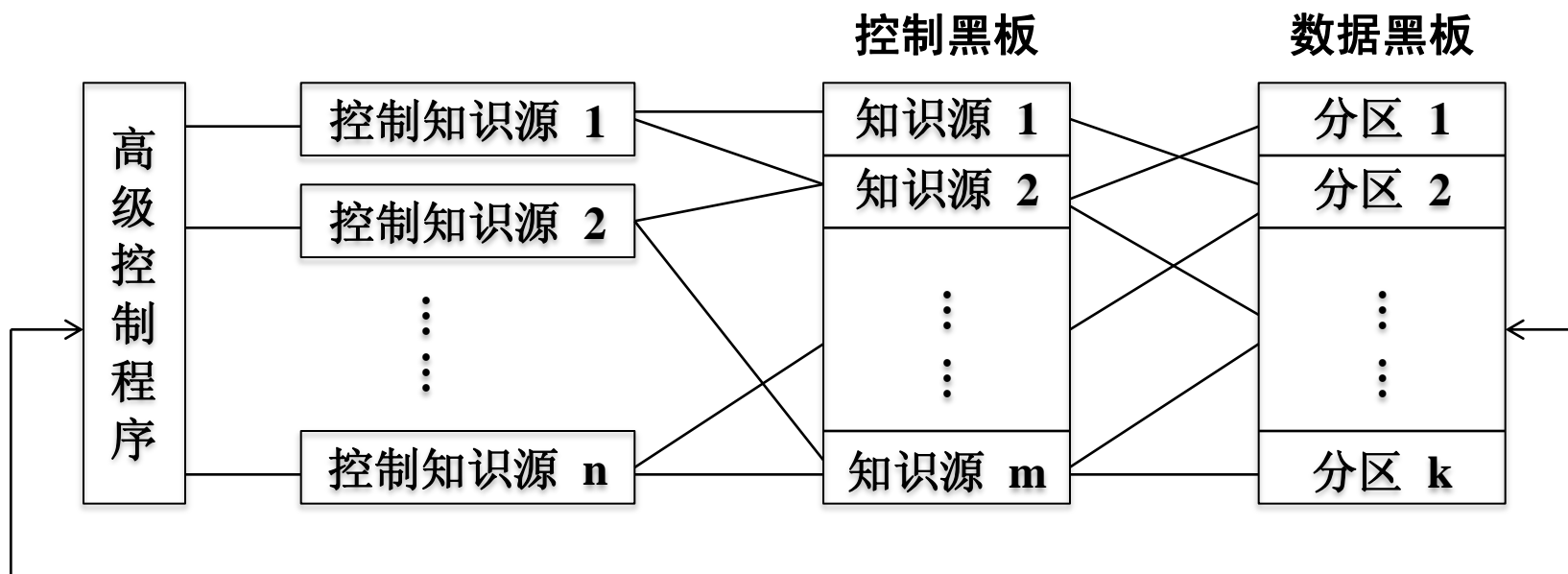


HEARSAY-II 系统的黑板结构:



多黑板结构

- 总控程序由多个分量组成，这些分量称为控制知识源
- 建立一块新的黑板——控制黑板，控制知识源通过这块黑板读取和修改领域知识源
- 高级总控程序驱动控制知识源，控制知识源驱动领域知识源



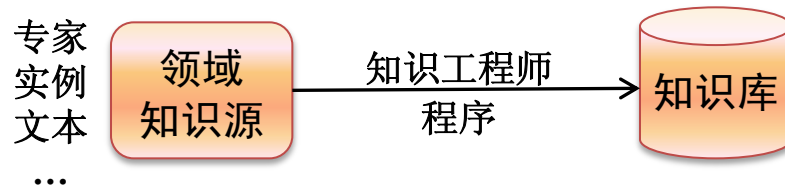
知识获取

- 从领域知识源提取知识并将其生成知识库的过程
- 知识获取贯穿于整个 ES 的建造过程中
- 消耗人力、物力、时间最多，一般占开发总工作量一半以上



知识获取方法

- 传统知识获取方法



- (1) 手工方式

知识工程师观察、记录领域专家解决问题的过程，手工获取知识

- (2) 智能编辑程序

领域专家通过与编辑程序交互，直接将知识输入到知识库

- (3) 辅助获取技术

知识诱导、场记分析方法、分类表格技术等

- (4) 归纳学习程序

采用归纳学习的方法，从一批领域实例自动生成规则库

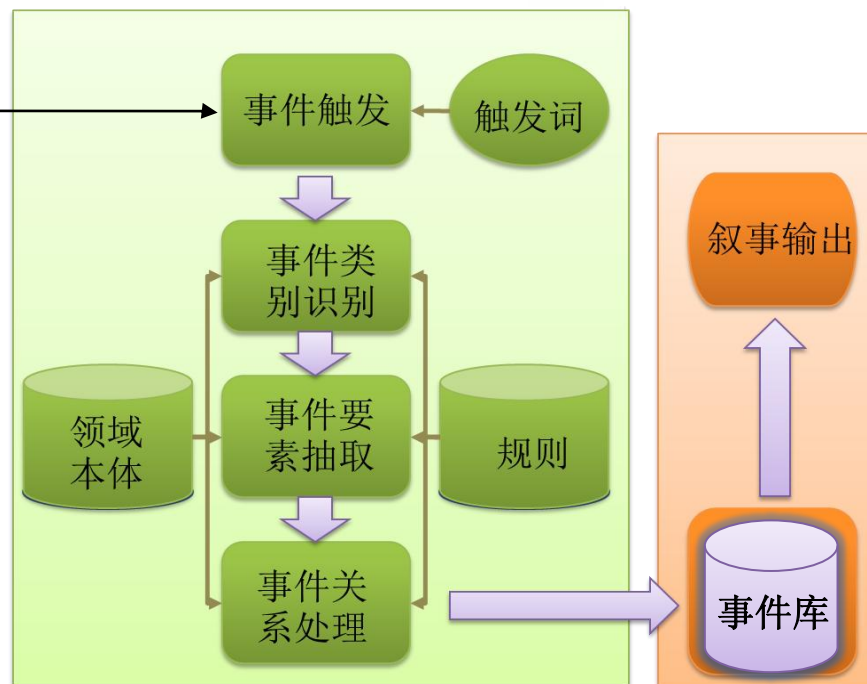
优点：高效性，并且归纳出的知识能够说明所有的实例

弱点：自动归纳出的规则往往复杂难以理解，从而使 ES 丧失透明性

面向互联网的文本知识提取

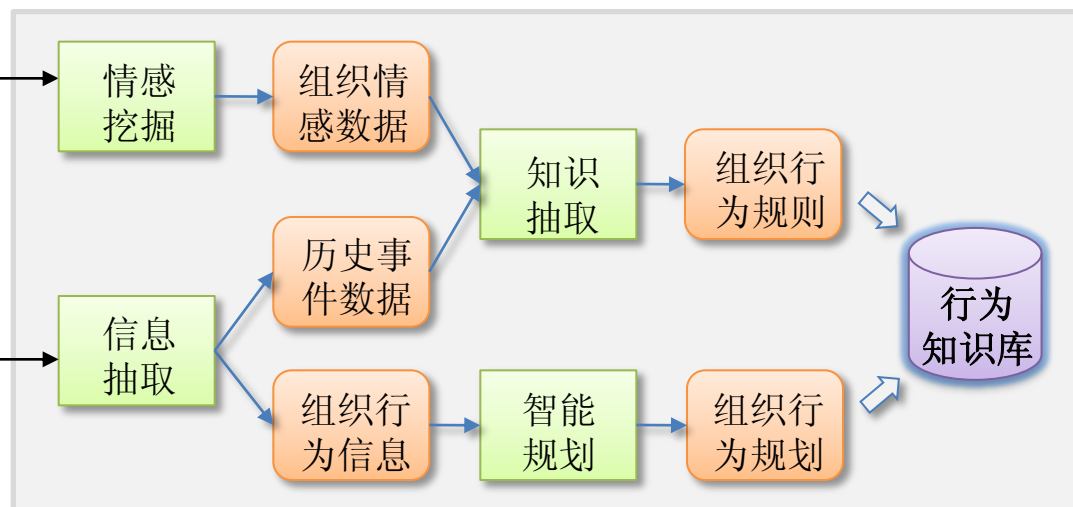
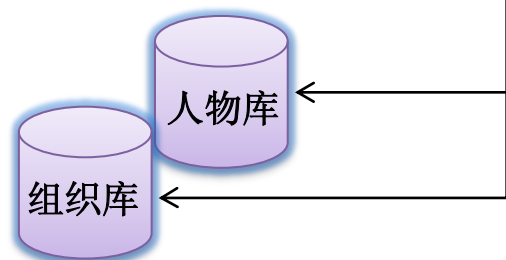
- 实体/概念抽取
- 实体间关系抽取

事件要素提取



人物、组织及属性提取

行为知识提取



知识求精

矛盾: $\begin{cases} R1: \text{IF } A_1 \wedge \dots \wedge A_n \text{ THEN } B \\ R2: \text{IF } A_1 \wedge \dots \wedge A_n \text{ THEN } \sim B \end{cases}$ 删除矛盾规则

$\begin{cases} R1: \text{IF } A_1 \wedge \dots \wedge A_n \text{ THEN } B \text{ (CF}_1\text{)} \\ R2: \text{IF } A_1 \wedge \dots \wedge A_n \text{ THEN } B \text{ (CF}_2\text{)} \end{cases}$ 删除矛盾规则

冗余: $\begin{cases} R1: \text{IF } A \text{ THEN } B \\ R2: \text{IF } B \text{ THEN } C \\ R3: \text{IF } A \text{ THEN } C \end{cases}$ 删除冗余规则

$\begin{cases} R1: \text{IF } A_1 \text{ THEN } B \\ R2: \text{IF } A_1 \wedge A_2 \text{ THEN } B \end{cases}$ 删除冗余规则

$\begin{cases} R1: \text{IF } A_1 \wedge A_2 \text{ THEN } B \\ R2: \text{IF } A_1 \wedge \sim A_2 \text{ THEN } B \end{cases}$ 合并两条规则

环路: $\begin{cases} R1: \text{IF } A \text{ THEN } B \\ R2: \text{IF } B \text{ THEN } C \\ R3: \text{IF } C \text{ THEN } A \end{cases}$ 修改或删除其中一条规则, 破坏形成环路的条件

专家系统开发工具

- 专家系统开发工具分为两大类：

生成工具： 用于生成系统的工具，主要是帮助知识工程师建造ES的推理机和知识库结构

- 骨架工具系统
- 知识工程语言
- 组合工具

辅助工具： 用于改善系统的工具，主要是帮助知识工程师获取和表示领域知识以及增强系统的辅助功能

- 知识获取工具、知识库管理与维护工具
- 解释工具、用户接口
- 验证工具等

骨架工具系统 (ES Shell)

- 最早采用、最方便的一种 ES 工具
- 实际上是一个空的专家系统，有完整的知识表示、推理机制和知识库结构，但不含任何领域知识
- 如：MYCIN→EMYCIN，PROSPECTOR→KAS，CASNET→EXPERT

优点：

- ① 知识表示形式和推理程序都是现成的，只需把新的领域知识装入知识库，因而可以快速生成新的 ES
如：EMYCIN→PUFF，SACON，CLOT，HEADMED 等
- ② 可以直接利用系统环境和辅助功能

局限性：

- ① 缺乏灵活性，知识表示形式和推理机制都已固定
- ② 领域局限性，内部结构往往带有原领域的烙印

知识工程语言

- 为建造知识系统专门设计、研制的语言
- 以一、两种较成熟的知识表示方法为基础，提供固定的知识表示形式和推理机制
- 如：OPS系列，OPS5(基于规则、正向推理)，OPS83(面向过程)M1(汉化)，S1(基于规则、框架、面向过程、反向推理、黑板控制)LOOPS(面向对象)，CLIPS(基于规则、面向过程、正向推理)

优点：① 知识表示形式、推理机制和内部结构并不针对具体领域，通用性、灵活性较强

② 与 AI 语言相比，知识粒度较大

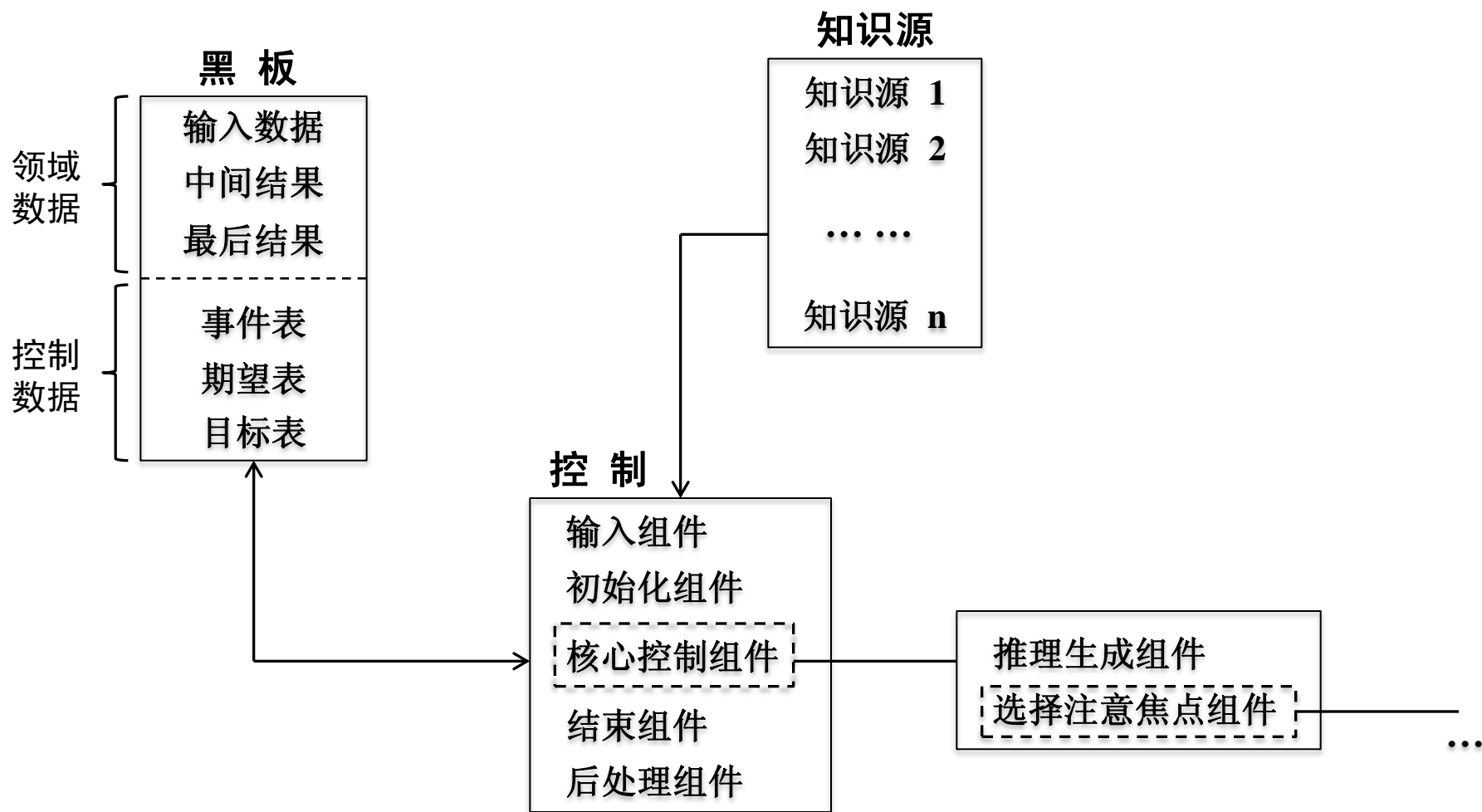
局限性：使用比较困难，用户不易掌握

组合(构造)工具



- 综合骨架工具系统和知识工程语言的优点：既提供骨架结构，又提供改变这种结构的手段
- 为用户提供一组预先定义和编程好的模块（称为组件），用户使用不同的模块组合可以构造出不同类型的 ES
- 对于对组件组装没有足够经验的用户，组合工具提供预先组装好的模块组合（骨架系统）
- 如：HEARSAY→AGE，MYCIN→TEIRESIAS (知识获取子系统)

- AGE 提供两种预先定义的骨架：① 基于规则、反向推理方式
② 针对多知识源、多求解方法的黑板结构



专家系统开发环境

- 提供给 ES 开发者的是一套工具，即一个开发工具包
- 一个较完整的集成化的 ES 开发工具包，称为专家系统开发环境

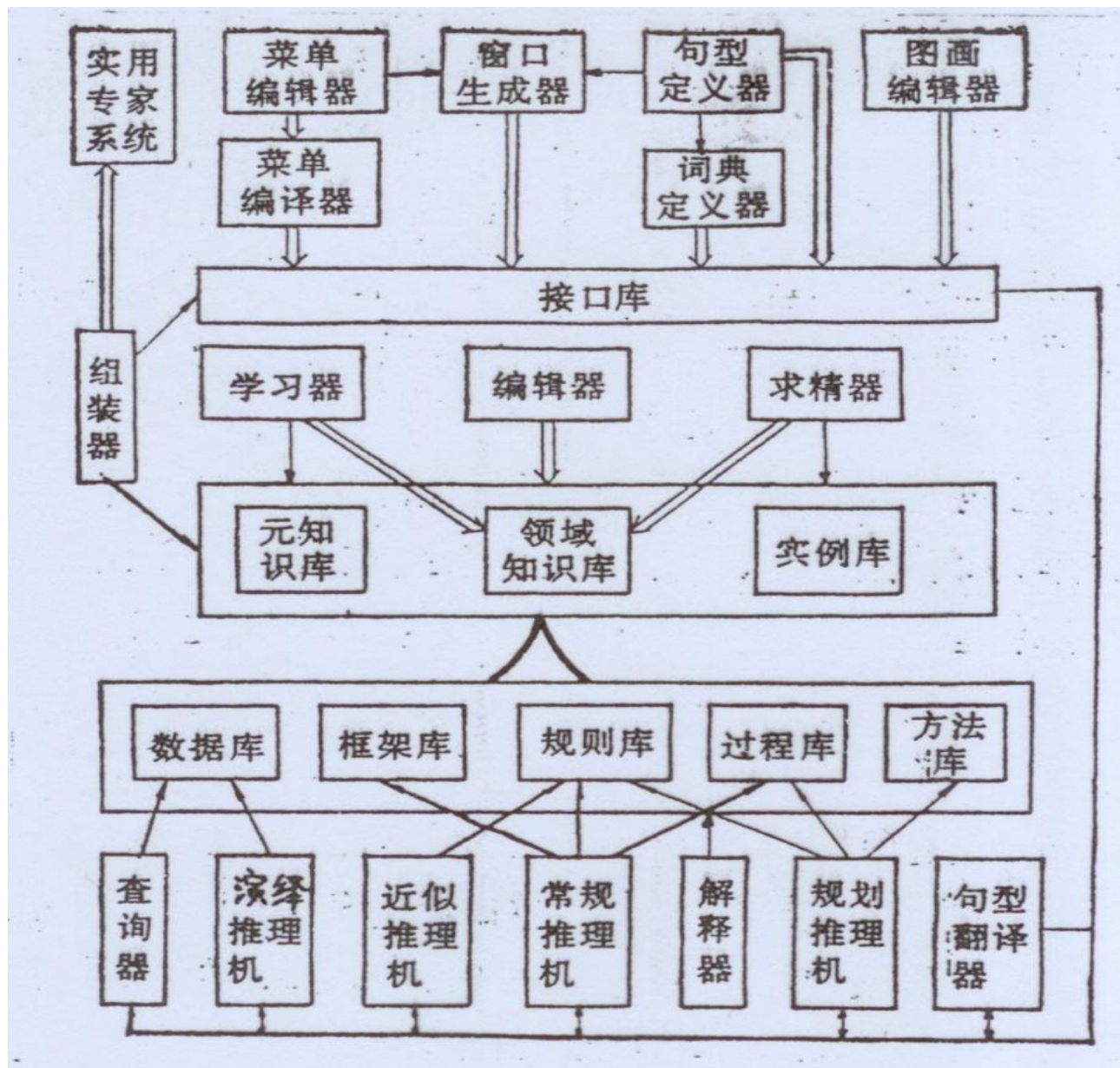
知识表示：一个好的工具至少提供两、三种以上的表示方法
(多知识表示往往需要多推理机)

推理机制：对不同问题类型采用不同的问题求解方法，提供多种
机制的推理机，以及多种不确定推理方法

知识获取工具：编辑、学习、求精

接口工具：提供多种接口工具

通用型集成式“天马”专家系统开发环境：



专家系统的弱点

- **脆弱性**

专业知识过于狭窄，在解决本专业问题时有高超的水平；一旦稍微偏离原领域，性能急剧下降

解决方法：增加知识的宽度，使系统具有常识和一般知识

- **知识获取的困难性**

知识获取作为建造知识系统的瓶颈，是一个长期存在的关键研究难点

针对知识本身的广袤性、模糊性、不确定性，以建造高性能知识系统为目标，结合多领域的研究成果，加强以知识为核心的智能研究

Toward Representing General Knowledge

- 以 D. Lenat 关于 Cyc 系统 (www.cyc.com) 的研发为代表
- 基于本体论知识表示观：表示是对自然世界的近似描述，注重的不是“其语言形式，而是其内容”，但所注重的“内容”不是某些特定领域的专家知识，而是自然界中那些具有普遍意义的“一般知识 (General Knowledge)”
- 寻求并建立一个具有常识知识性质并可为大多数领域使用的一般性知识库

如：

- The typical bird has 1 beak, 1 heart, lots of feathers,...
- Hearts are internal organs; feathers are external protrusions
- You can usually see people's noses, but not their hearts
- You cannot remember events that have not happened yet

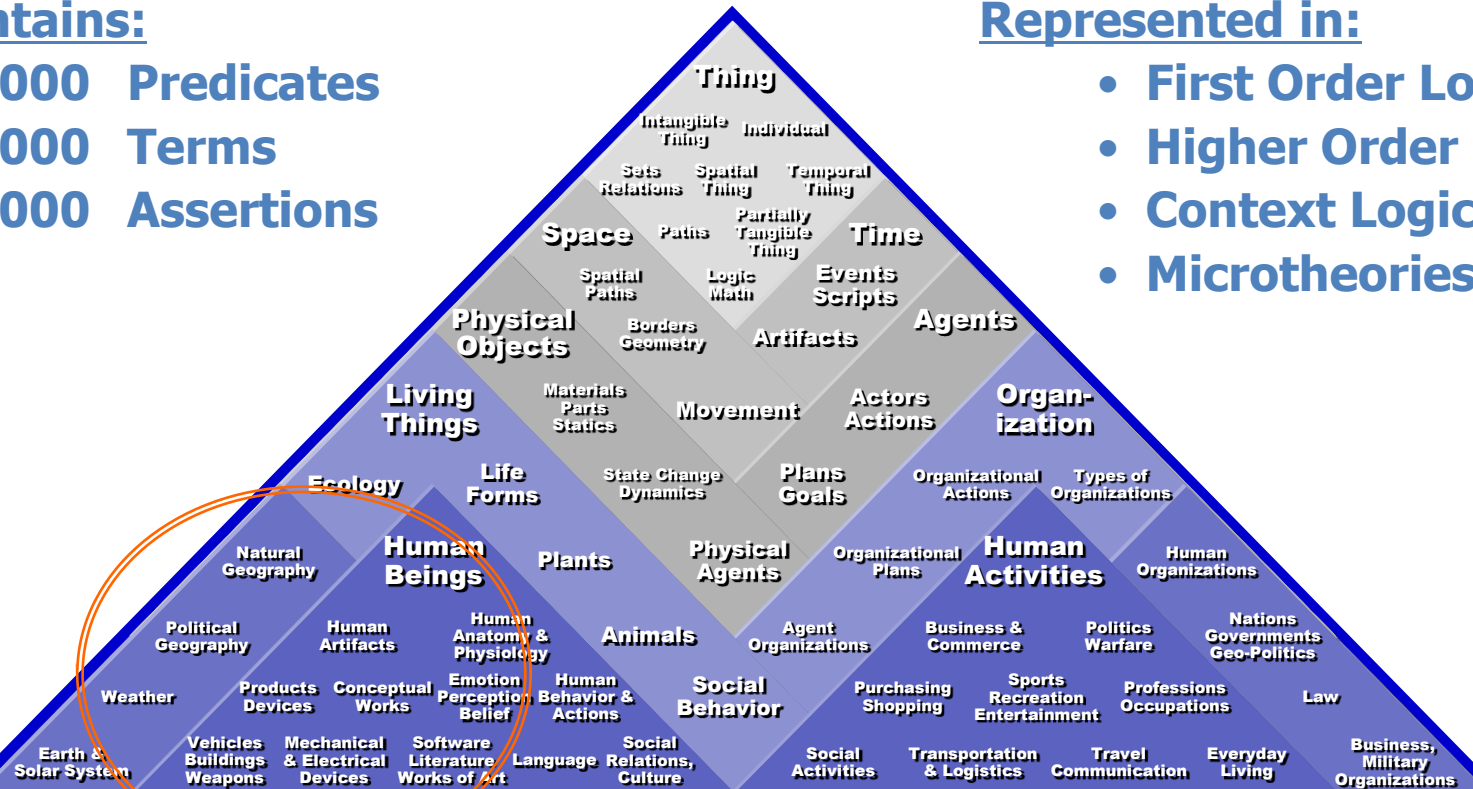
Cyc Knowledge Base (1984–now)

Cyc contains:

17,000 Predicates
500,000 Terms
7,000,000 Assertions

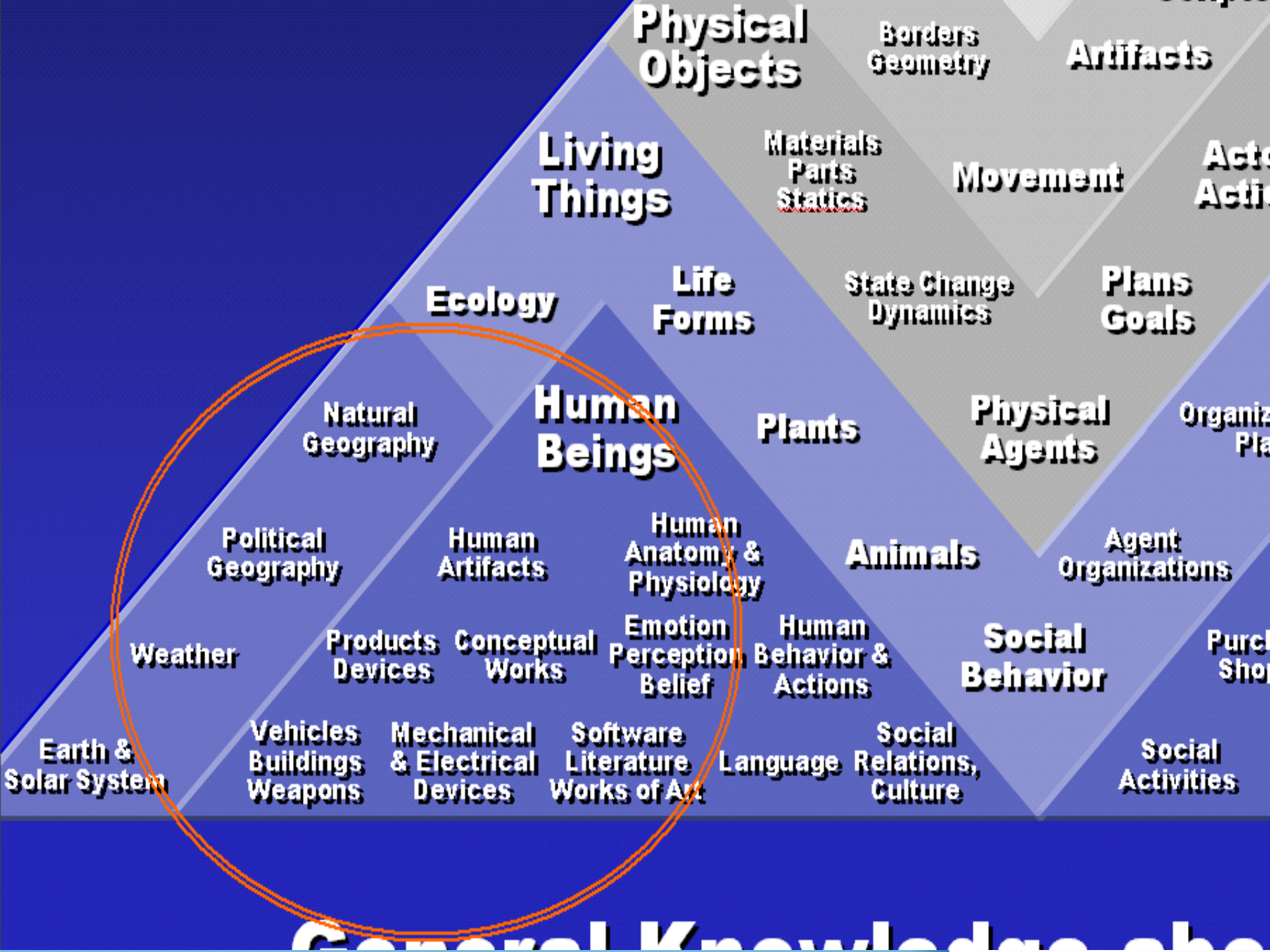
Represented in:

- First Order Logic
- Higher Order Logic
- Context Logic
- Microtheories



General Knowledge about Various Domains

Specific data, facts, and observations



Formalizing Commonsense Psychology

- 针对 30 个代表性的常识心理学研究领域开展的大规模常识知识形式化和逻辑推理研究

- | | |
|---------------------------|------------------------------|
| 1. Knowledge management | 16. Plan elements |
| 2. Similarity comparisons | 17. Planning modalities |
| 3. Memory retrieval | 18. Planning goals |
| 4. Emotions | 19. Plan construction |
| 5. Explanations | 20. Plan adaptation |
| 6. Envisioning (Thinking) | 21. Design |
| 7. Execution envisionment | 22. Making decisions |
| 8. Causes of failure | 23. Scheduling |
| 9. Managing expectations | 24. Monitoring |
| 10. Other agent reasoning | 25. Execution modalities |
| 11. Threat detection | 26. Execution control |
| 12. Goals | 27. Repetitive execution |
| 13. Goal themes | 28. Plan following |
| 14. Goal management | 29. Observation of execution |
| 15. Plans | 30. Body interaction |

Example: Goals and Planning

Causal Knowledge:

$(\forall e_1, x)[p'(e_1, x) \rightarrow (\exists e_2)[q'(e_2, x) \& \text{cause}(e_1, e_2)]]$
or, p causes q

$(\forall e_1, x)[p'(e_1, x) \rightarrow (\exists e_2)[q'(e_2, x) \& \text{enable}(e_1, e_2)]]$
or, p enables q

where $\text{enable}(e_1, e_2) \leftrightarrow \text{cause}(\sim e_1, \sim e_2)$

Planning Axioms:

$(\forall a, e_1, e_2)[\text{goal}(a, e_2) \& \text{cause}(e_1, e_2) \& \text{etc} \rightarrow \text{goal}(a, e_1)]$

↖ If we want the effect,
we want the cause, sometimes.

$(\forall a, e_1, e_2)[\text{goal}(a, e_2) \& \text{enable}(e_1, e_2) \rightarrow \text{goal}(a, e_1)]$

subgoal(a, e₁, e₂)

Means-end reasoning

↖ If we want the effect,
we want the enabling conditions.

专家系统小结

1. 专家系统及其发展概况
2. ES 基本结构和组成、黑板结构
3. ES 典型实例：MYCIN 系统
4. 知识获取及知识求精
5. 专家系统开发工具



谢谢大家！

Thank
You!!

wenji.mao@ia.ac.cn

