Artificial Intelligence

Expert System

By,
Shailendra S. Aote
Dept. of CSE,
RCOEM, Nagpur

What is Expert System?

Computer software that:

Emulates human expert

Deals with small, well defined domains of expertise

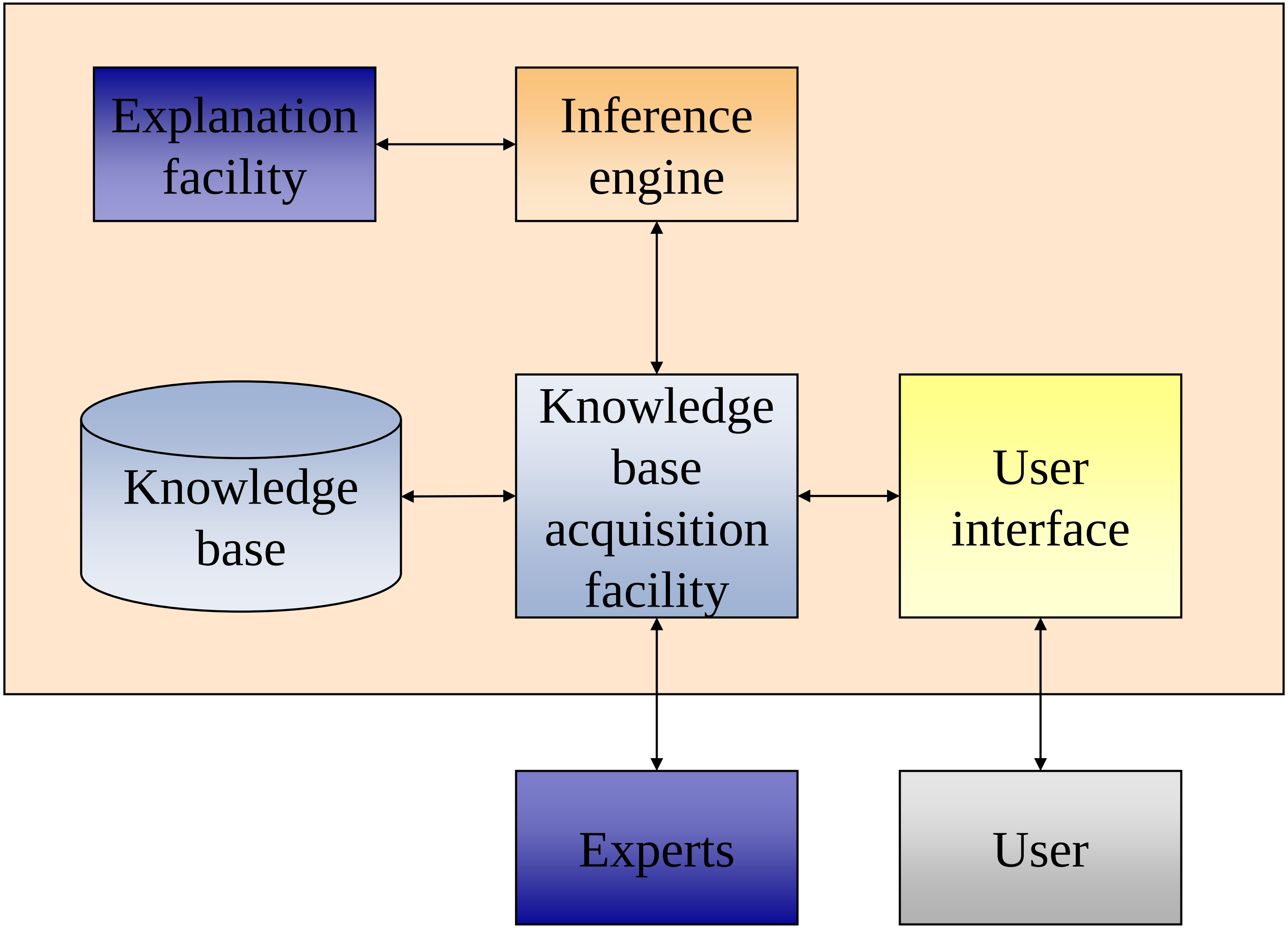
Is able to solve real-world problems

Is able to act as a cost-effective consultant

Can explains reasoning behind any solutions it finds

An Expert System is a computer program that simulates human intelligence and behavior in specific and limited domains

Architecture of an Expert System?



Architecture of an Expert System?

Rules for a Credit Application

Mortgage application for a loan for $100,000 to $200,000

If there are no previous credits problems, and

If month net income is greater than 4x monthly loan payment, and

If down payment is 15% of total value of property, and

If net income of borrower is > $25,000, and

If employment is > 3 years at same company

Then accept the applications

Else check other credit rules

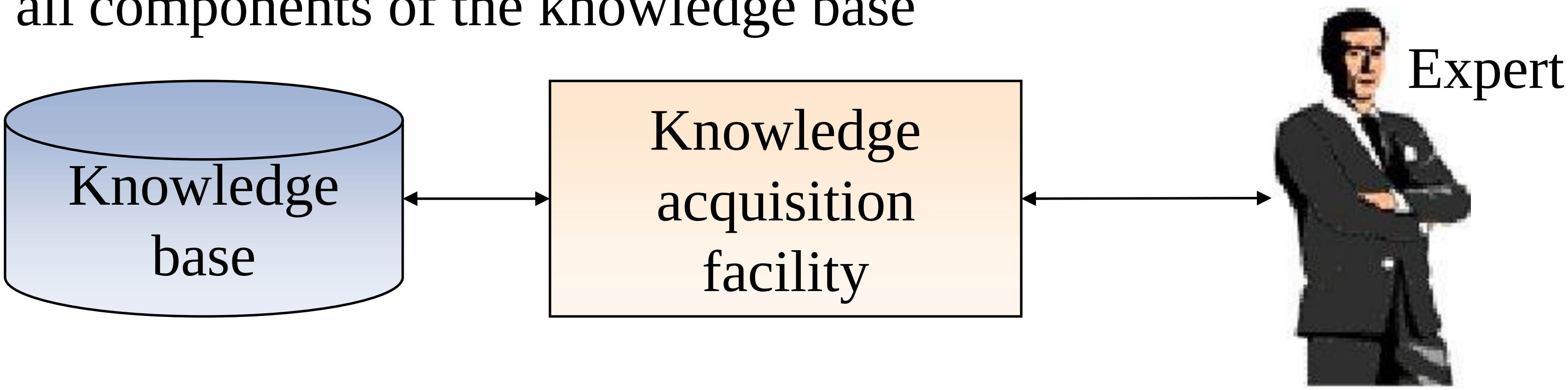
Architecture of an Expert System?

- Explanation facility

  A part of the expert system that allows a user or decision maker to understand how the expert system arrived at certain conclusions or results

- Knowledge acquisition facility

  Provides a convenient and efficient means of capturing and storing all components of the knowledge base

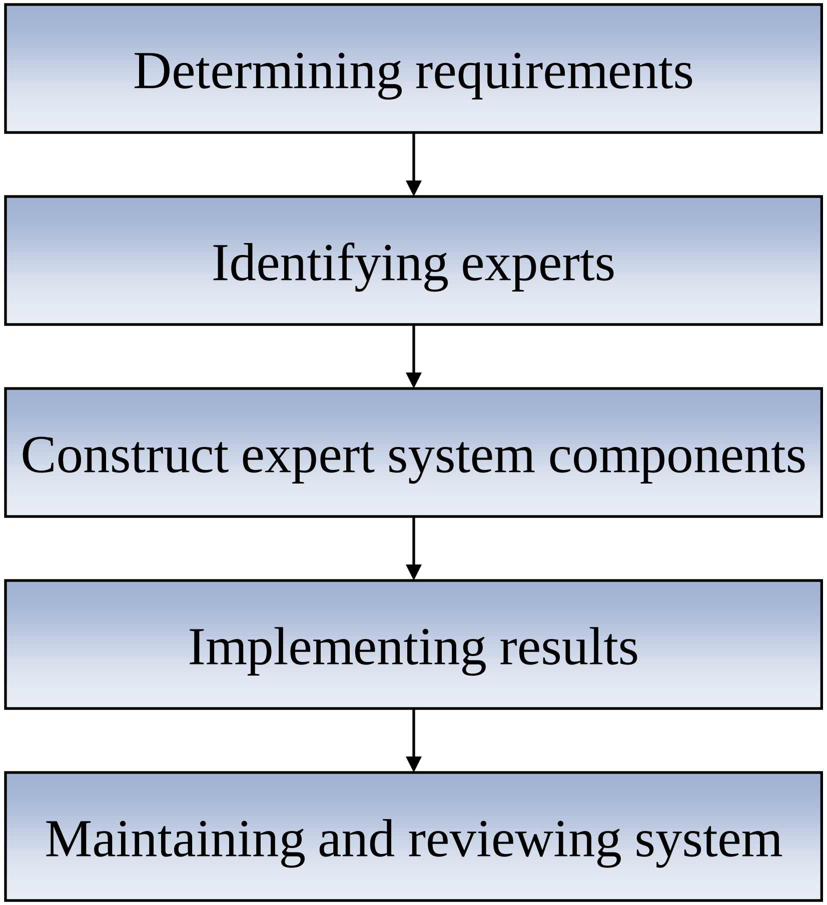


Knowledge base

Knowledge acquisition facility

Expert

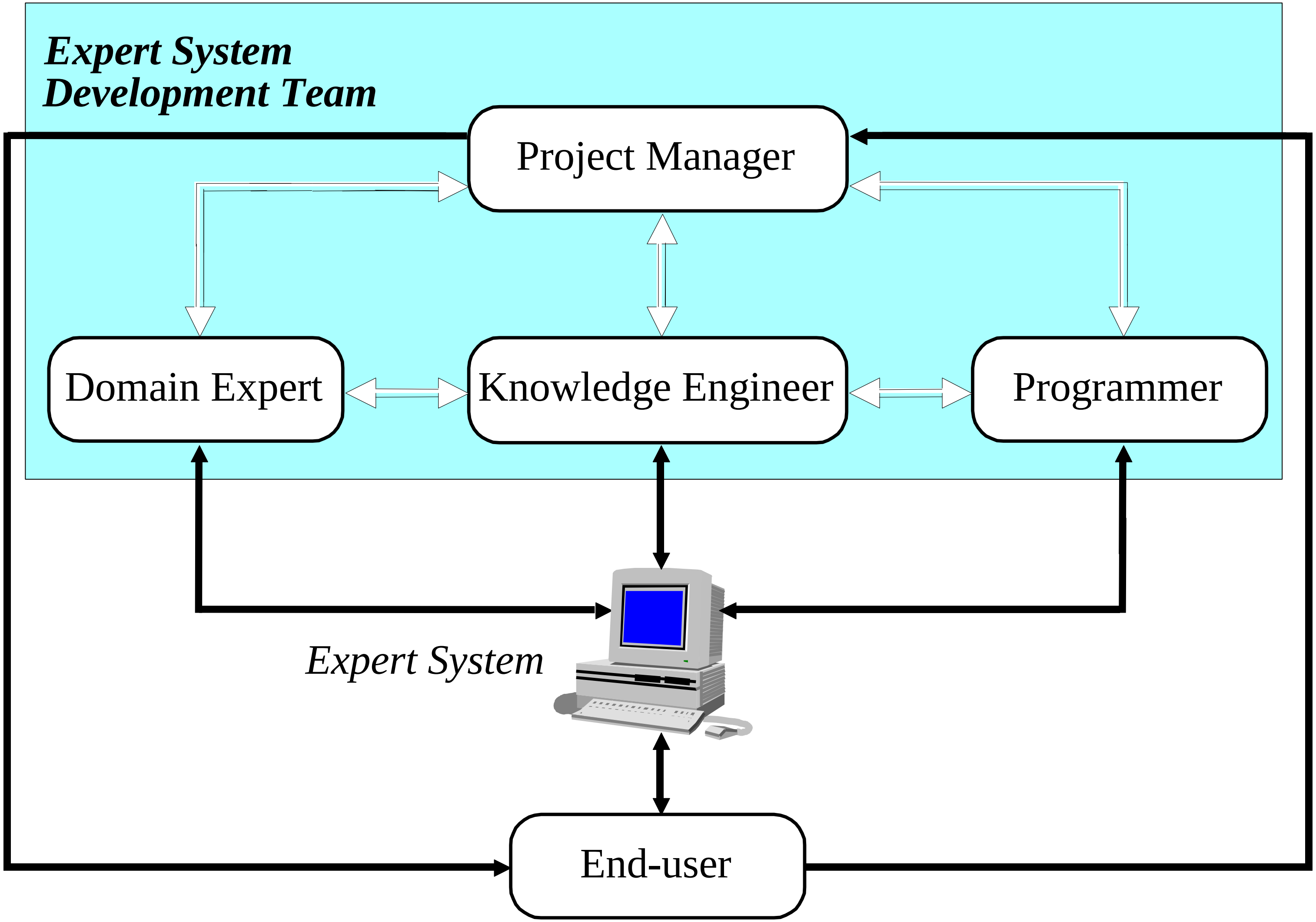
Expert Systems Development


Determining requirements
Identifying experts
Construct expert system components
Implementing results
Maintaining and reviewing system

Domain

- The area of knowledge addressed by the expert system.

The main players in the development team



The main players in the development team

❑The ***domain expert*** is a knowledgeable and skilled person capable of solving problems in a specific area or ***domain***.

❑ This person has the greatest expertise in a given domain.

❑This expertise is to be captured in the expert system.

❑Therefore, the expert must be able to communicate his or her knowledge, be willing to participate in the expert system development and commit a substantial amount of time to the project.

❑The domain expert is the most important player in the expert system development team.

The main players in the development team

- **The knowledge engineer** is someone who is capable of designing, building and testing an expert system.
- He or she interviews the domain expert to find out how a particular problem is solved.
- The knowledge engineer establishes what reasoning methods the expert uses to handle facts and rules and decides how to represent them in the expert system.
- The knowledge engineer then chooses some development software or an expert system shell, or looks at programming languages for encoding the knowledge.

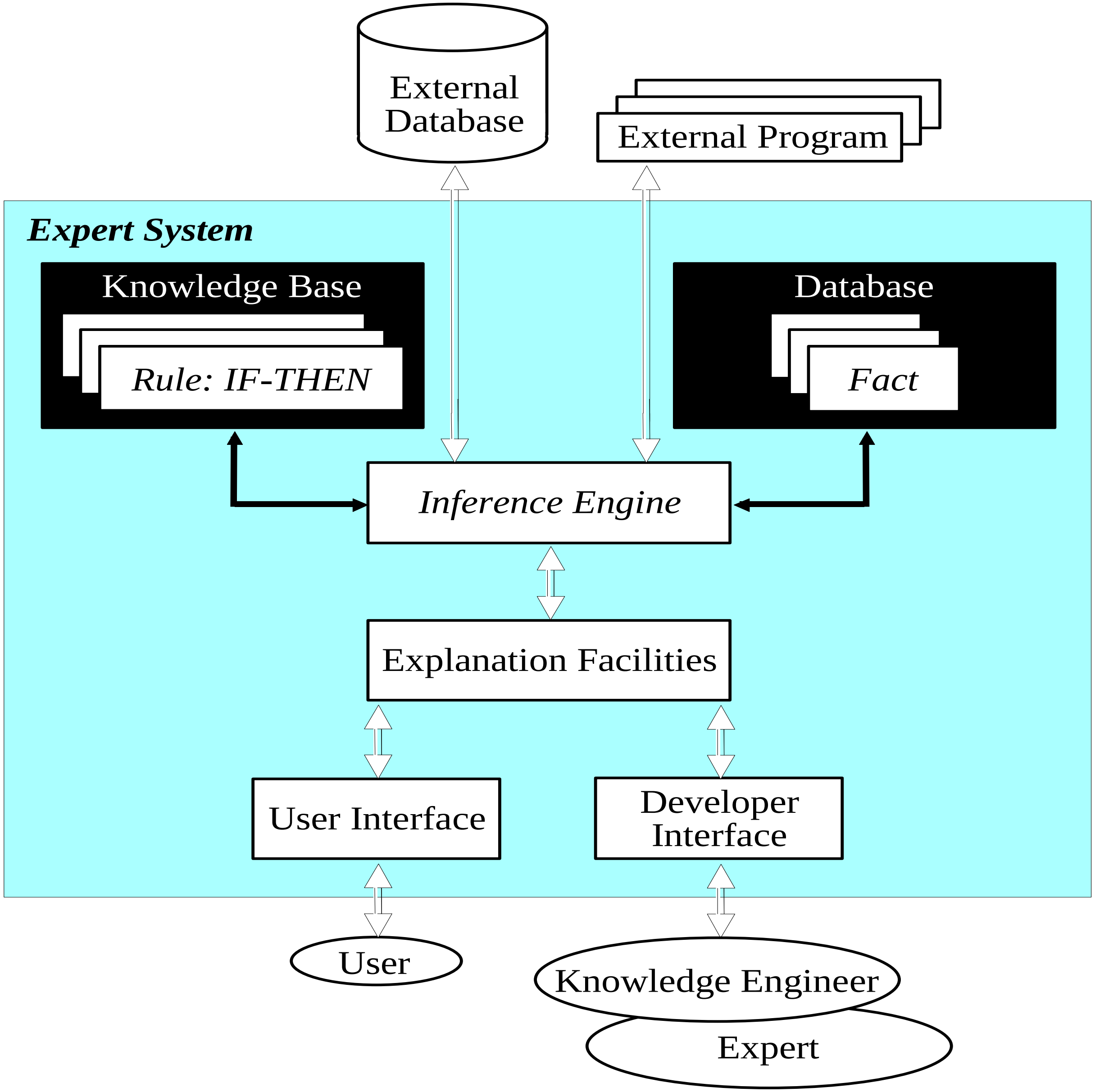
The main players in the development team

- **The programmer** is the person responsible for the actual programming, describing the domain knowledge in terms that a computer can understand.

- **The project manager** is the leader of the expert system development team, responsible for keeping the project on track. He or she makes sure that all deliverables and milestones are met, interacts with the expert, knowledge engineer, programmer and end-user.

- **The end-user**, often called just the user, is a person who uses the expert system when it is developed.

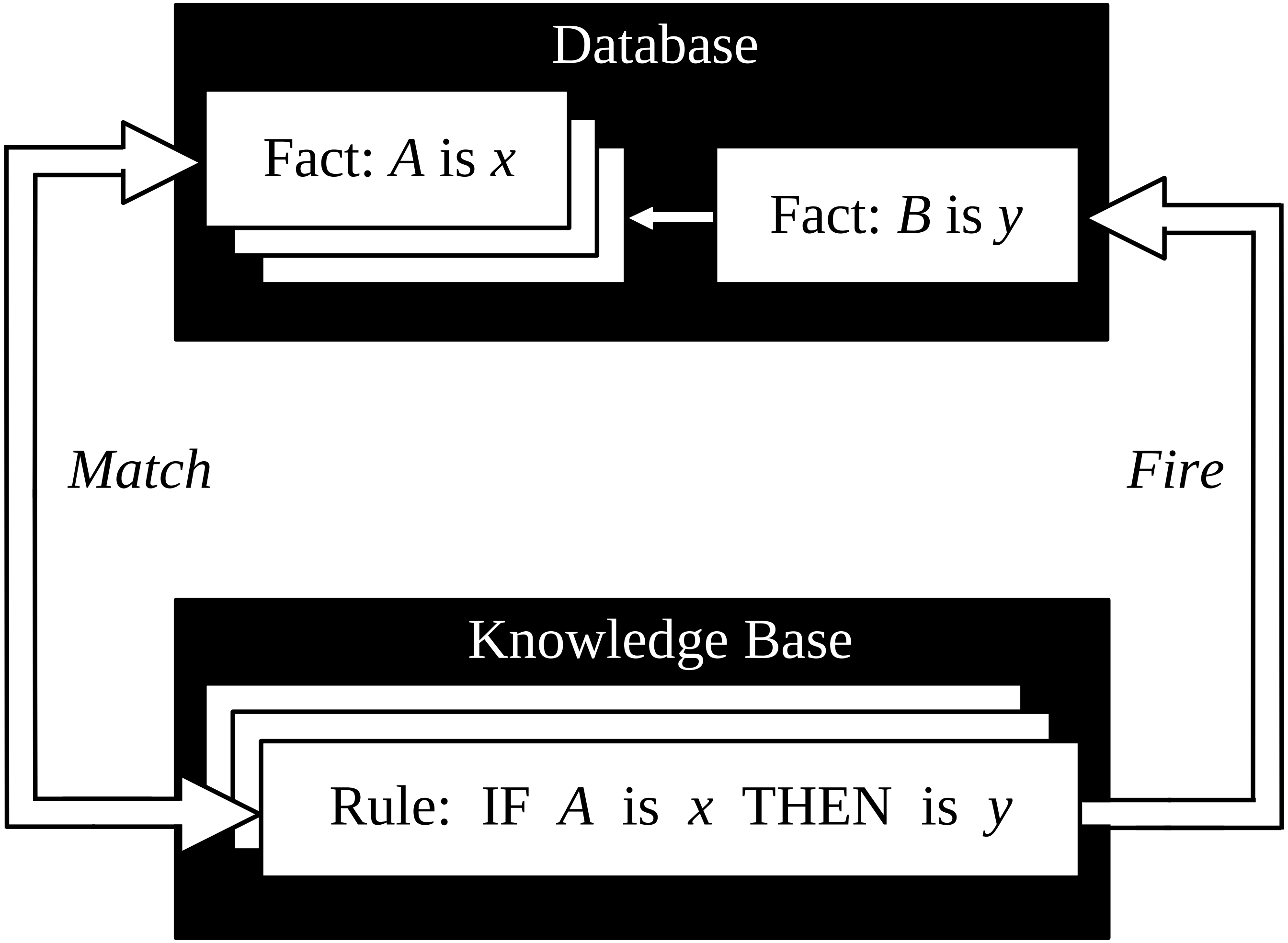
Complete structure of a rule-based expert system



Forward chaining and backward chaining

- In a rule-based expert system, the domain knowledge is represented by a set of IF-THEN production rules and data is represented by a set of facts about the current situation. The inference engine compares each rule stored in the knowledge base with facts contained in the database. When the IF (condition) part of the rule matches a fact, the rule is **fired** and its THEN (action) part is executed.
- The matching of the rule IF parts to the facts produces **inference chains**. The inference chain indicates how an expert system applies the rules to reach a conclusion.

Inference engine cycles via a match-fire procedure



Database

Fact: *A* is *x*

Fact: *B* is *y*

Match

Fire

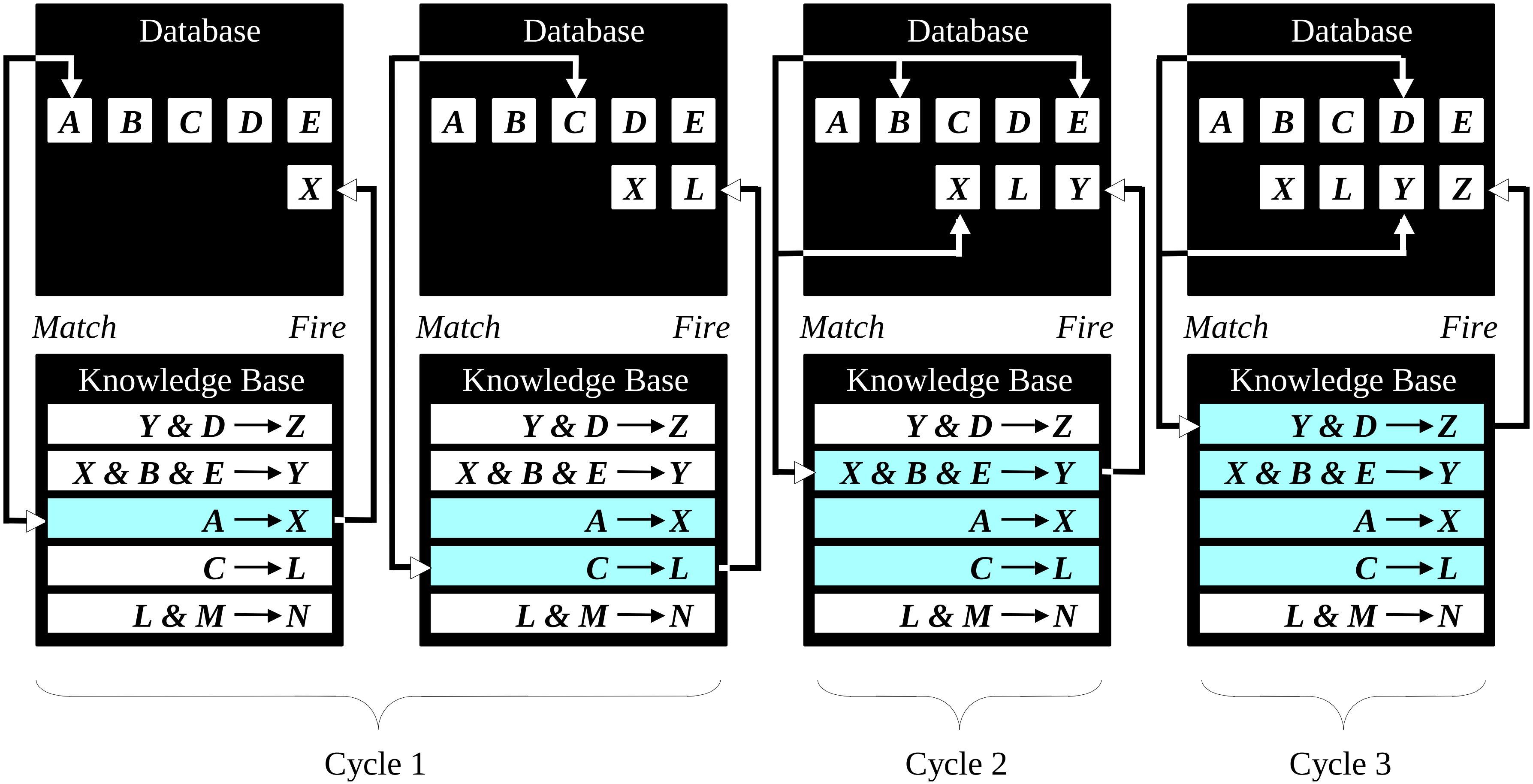
Knowledge Base

Rule: IF *A* is *x* THEN is *y*

Forward chaining

- Forward chaining is the data-driven reasoning. The reasoning starts from the known data and proceeds forward with that data. Each time only the topmost rule is executed. When fired, the rule adds a new fact in the database. Any rule can be executed only once. The match-fire cycle stops when no further rules can be fired.

Forward chaining

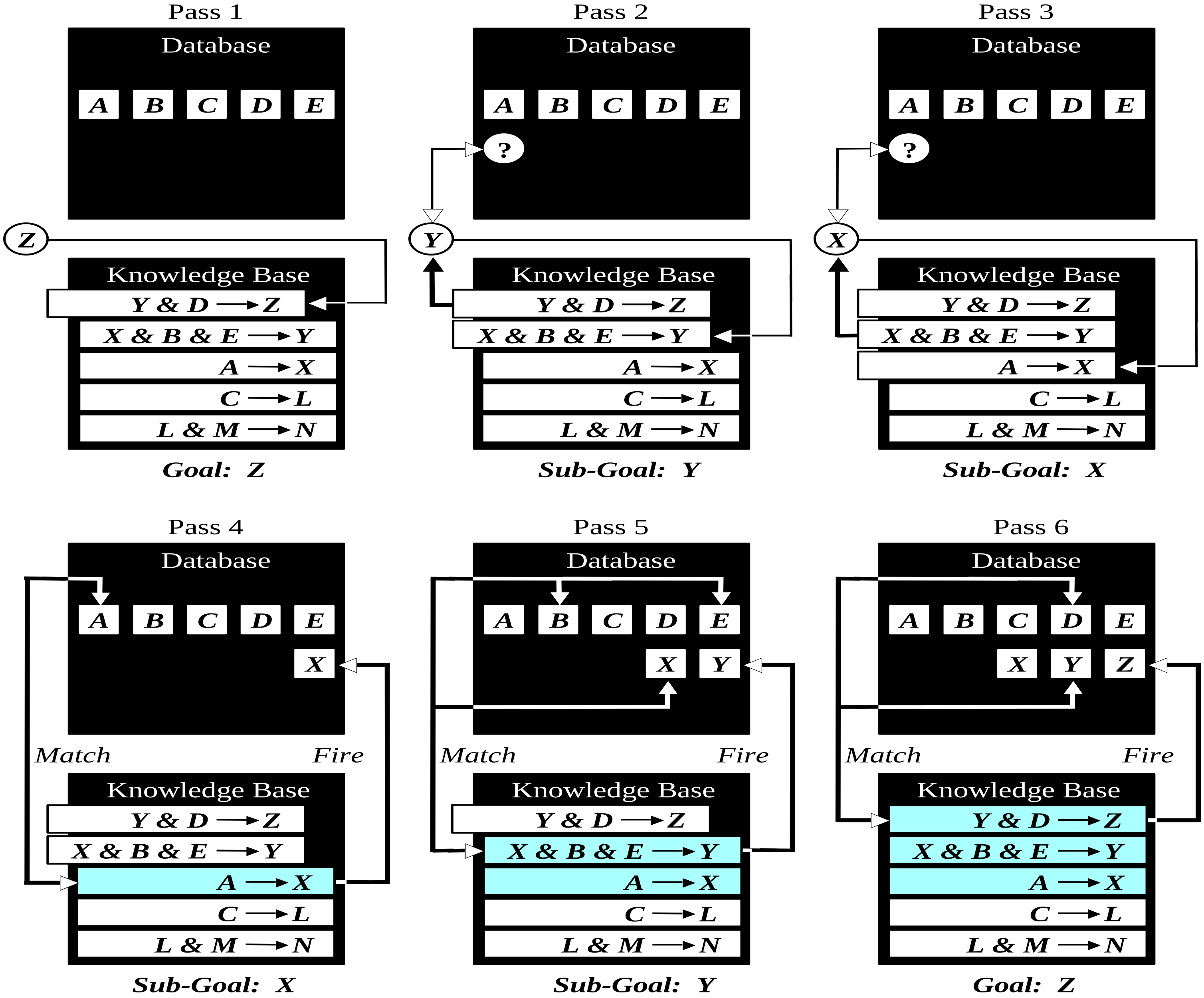


- However, in forward chaining, many rules may be executed that have nothing to do with the established goal.
- Therefore, if our goal is to infer only one particular fact, the forward chaining inference technique would not be efficient.

Backward chaining

- Backward chaining is the **goal-driven reasoning**.
- In backward chaining, an expert system has the goal (a *hypothetical solution*) and the inference engine attempts to find the evidence to prove it.
- First, the knowledge base is searched to find rules that might have the desired solution. Such rules must have the goal in their THEN (action) parts.
- If such a rule is found and its IF (condition) part matches data in the database, then the rule is fired and the goal is proved. However, this is rarely the case.

Backward chaining



Pass 1

Database

A B C D E

Z

Knowledge Base

Y & D → Z

X & B & E → Y

A → X

C → L

L & M → N

Goal: Z

Pass 2

Database

A B C D E

?

Y

Knowledge Base

Y & D → Z

X & B & E → Y

A → X

C → L

L & M → N

Sub-Goal: Y

Pass 3

Database

A B C D E

?

X

Knowledge Base

Y & D → Z

X & B & E → Y

A → X

C → L

L & M → N

Sub-Goal: X

Pass 4

Database

A B C D E

X

Match

Fire

Knowledge Base

Y & D → Z

X & B & E → Y

A → X

C → L

L & M → N

Sub-Goal: X

Pass 5

Database

A B C D E

X Y

Match

Fire

Knowledge Base

Y & D → Z

X & B & E → Y

A → X

C → L

L & M → N

Sub-Goal: Y

Pass 6

Database

A B C D E

X Y Z

Match

Fire

Knowledge Base

Y & D → Z

X & B & E → Y

A → X

C → L

L & M → N

Goal: Z

How do we choose between forward and backward chaining?

- If an expert first needs to gather some information and then tries to infer from it whatever can be inferred, choose the forward chaining inference engine.
- However, if your expert begins with a hypothetical solution and then attempts to find facts to prove it, choose the backward chaining inference engine.

How do we choose between forward and backward chaining?

- If an expert first needs to gather some information and then tries to infer from it whatever can be inferred, choose the forward chaining inference engine.
- However, if your expert begins with a hypothetical solution and then attempts to find facts to prove it, choose the backward chaining inference engine.

Applications

- *Classification* - identify an object based on stated characteristics
- *Diagnosis Systems* - infer malfunction or disease from observable data
- *Monitoring* - compare data from a continually observed system to prescribe behavior
- *Process Control* - control a physical process based on monitoring
- *Design* - configure a system according to specifications
- *Scheduling & Planning* - develop or modify a plan of action
- *Generation of Options* - generate alternative solutions to a problem

Applications

- MYCIN was a rule-based expert system for the diagnosis of infectious blood diseases. It also provided a doctor with remedial advice in a convenient, user-friendly manner.
- MYCIN's knowledge consisted of about 450 rules derived from human knowledge in a narrow domain through extensive interviewing of experts.
- The knowledge incorporated in the form of rules was clearly separated from the reasoning mechanism. The system developer could easily manipulate knowledge in the system by inserting or deleting some rules.

MYCIN:
Medical system for diagnosing blood disorders. First used in 1979

Applications

■**PROSPECTOR** was an expert system for mineral exploration developed by the Stanford Research Institute. Nine experts contributed their knowledge and expertise. PROSPECTOR used a combined structure that incorporated rules and a semantic network. PROSPECTOR had over 1000 rules.

■The user, an exploration geologist, was asked to input the characteristics of a suspected deposit: the geological setting, structures, kinds of rocks and minerals. PROSPECTOR compared these characteristics with models of ore deposits and made an assessment of the suspected mineral deposit. It could also explain the steps it used to reach the conclusion.

PROSPECTOR:
Used by geologists to identify sites
for drilling or mining

Applications

■**DENDRAL** was developed at Stanford University to determine the molecular structure of Martian soil, based on the mass spectral data provided by a mass spectrometer. The project was supported by NASA. Edward Feigenbaum, Bruce Buchanan (a computer scientist) and Joshua Lederberg (a Nobel prize winner in genetics) formed a team.

■There was no scientific algorithm for mapping the mass spectrum into its molecular structure. Feigenbaum's job was to incorporate the expertise of Lederberg into a computer program to make it perform at a human expert level. Such programs were later called ***expert systems***.

DENDRAL:
Used to identify the structure of chemical compounds. First used in 1965

Applications

LITHIAN:
Gives advice to archaeologists examining stone tools

DESIGN ADVISOR:
Gives advice to designers of processor chips

PUFF:
Medical system for diagnosis of respiratory conditions

Benefits of Expert Systems

- ❑Increased Output and Productivity
- ❑Decreased Decision Making Time
- ❑Increased Process(es) and Product Quality
- ❑Flexibility
- ❑Easier Equipment Operation
- ❑Knowledge Transfer to Remote Locations
- ❑Provide Training
- ❑Operation in Hazardous Environments, where people can not go

Problems and Limitations of Expert Systems

❑Knowledge is not always readily available

❑Expertise can be hard to extract from humans

❑Each expert’s approach may be different, yet correct

❑Hard, even for a highly skilled expert, to work under time pressure

❑Users of expert systems have natural cognitive limits

❑ES work well only in a narrow domain of knowledge