

Introduction to Expert Systems

Lecture Outline

- knowledge representation.
- Expert Systems (Why, what, when?)
 - Rule-based systems
 - Knowledge base
 - Inference Engine
 - Other components.
- Other Types of Expert Systems
- Advantages and disadvantage.

knowledge representation using rules

- Rules are straight forward, they are given as **IF-THEN** expressions:
If (Condition) **then** (Conclusion)
- It can be represented as **condition-action** expressions:

If (Condition) **then** (Action)
- Rules are effectively propositional

Examples of simple rules:

IF someone at the kitchen for more than 5 min **AND** the cooker is on **THEN** they prepare food/drink

IF the temperature is high **AND** temperature gauge is working fine **THEN** stop the engine

If (Condition) then (Conclusion)

- Conditions  antecedent, independent variable
- Conclusion or action  consequent, outcome, or dependent variable

If (Antecedent)

- Antecedent or condition are a series of tests.
- Every antecedent acts like a node in the decision tree.
- The series of tests are logically ANDed together.
- Rules can have different logical expressions like > < == not.

THEN

- **Consequence:**
 - Conclusion, or set of conclusions.
 - Action(s).
 - The probability or the certainty in given consequence.
- Consequences can conflict when their **Antecedent** are satisfied.

Reason for Consequents

- Explaining the relationship:
 - IF the season is summer AND the forecast is sunny THEN it will not rain today.
- Making recommendations:
 - IF the car is not starting AND tank is full AND the battery is good THEN you need a mechanic.
- Propose actions:
- IF the car is dead AND the tank is full THEN check the battery

- Making Predictions:
- IF the kitchen is occupied for X-time AND the cooker is on for Y-time AND Fridge opened and closed AND Steam is detected THEN the activity is preparing meal.

Summary of Rules

	First part	Second part
Name	Antecedent Situation Information Observation IF	Conclusion Consequence Advice Action Prediction THEN
Nature	Condition	Resolution, decision
Size	Can have more than one IF	Only one THEN
Statement	ANDs	All condition must be true
	ORs	One true is enough

Logic relation

- Conditions like $>$, $<$, $==$, not can be used. They are called conditional relations.
- Variables can be compared with constant for example (temperature < 38).
 - IF the car value $< \text{£}3000$ AND the NCP > 10 years AND had no accident in the last 5 years THEN the annual premium is $\text{£}500$.
- Searching for relations between many attributes can be very expensive.

Therefor an extra intermediate variable usually binary is used:
is the car value $< \text{£}3000 \Rightarrow$ true or false

Less common Rules form

- Inclusion of ELSE:
 - If you attend the lectures AND do the coursework THEN you will pass the module ELSE I'm not sure.
- Conclusion before the IF condition
 - Your chance of getting the Loan is high, IF your credit record is good.
- Action that trigger another rule(s):
 - IF you attend the lectures AND you do the coursework THEN you likely will pass the module ELSE contact the lecturer

The execution order of rules

- Two ways of executing a set of rule:
 - Ordered set of rules when the order is important for interpretation.
 - Unordered set of rules, order may cause a conflict.

Default rule (binary class):

We only define rules when the instant is belong to a class is true , if it is false we activate the default rule. In this case the order is not important

If the room is occupied THEN switch the light on otherwise switch it off

Rules with exceptions

This can be used to handle the special cases. Exceptions make the rules easy to be updated when we need to incorporate new data, or new domain knowledge.

If the car value < £3000 AND the NCP > 10 years AND had an accident in the last 5 years THEN the annual premium is £1500.

Modified rule with exception:

If the car value < £3000 AND the NCP > 10 years AND had an accident in the last 5 years THEN the annual premium is £1500. EXCEPT if the accident was not their fault THEN the annual premium is £700.

Disadvantage of using exceptions

- Can cause overfitting in the expert system.
- Need constant update
- Do not work well with noisy data.

Rules as source of knowledge

- Each rule represents some amount of expertise.
- Individual rules are selected alternatively to each other (OR).
- The set of rules work collectively together when they fed to the inference engine. the system to make a choice, decision, or prediction.
- Rules can be considered as a way of modelling the cognitive behaviour of human experts.
- Rules represent the experience of the human experts.

Rule inference:

- Forward chaining (data driven):
 - It starts with facts, then find the relevant rules, apply one of them.
- Backward chaining (goal oriented):
 - It starts with selecting the rules that decompose the goal, then subsequent goals are solved to prove the original goal.

Rule Matching

- Matching is computationally expensive step.
- Matching usually leads to more than one applicable rule. Which will likely lead to a conflict.
- Matching is finding the relevant rules to the situation that cover the required knowledge.
- Rules are triggered based on the type of the inference or the chaining.
 - Forward chaining: The rule antecedents are matched to facts in fact base.
 - Backward chaining: The rule consequent is matched to a fact. The aim is to prove that the consequent of the rule is true.

Advantage of using rules

- Simple and easy to understand.
- Easy to use to map human's experience to knowledge system.
- Straightforward to implement and easy to debug.

Disadvantages

- Simplifying the rules can lead to a very complex implementation. It also can be computationally very expensive when searching and matching the rules.
- It is not easy to modelled the complex using rules. Therefore, rules are not the best choice when the problem is complex.
- Rules require huge memory space.
- Using large set of rules can be problematic. It become difficult to debug, maintain and understand the rules.

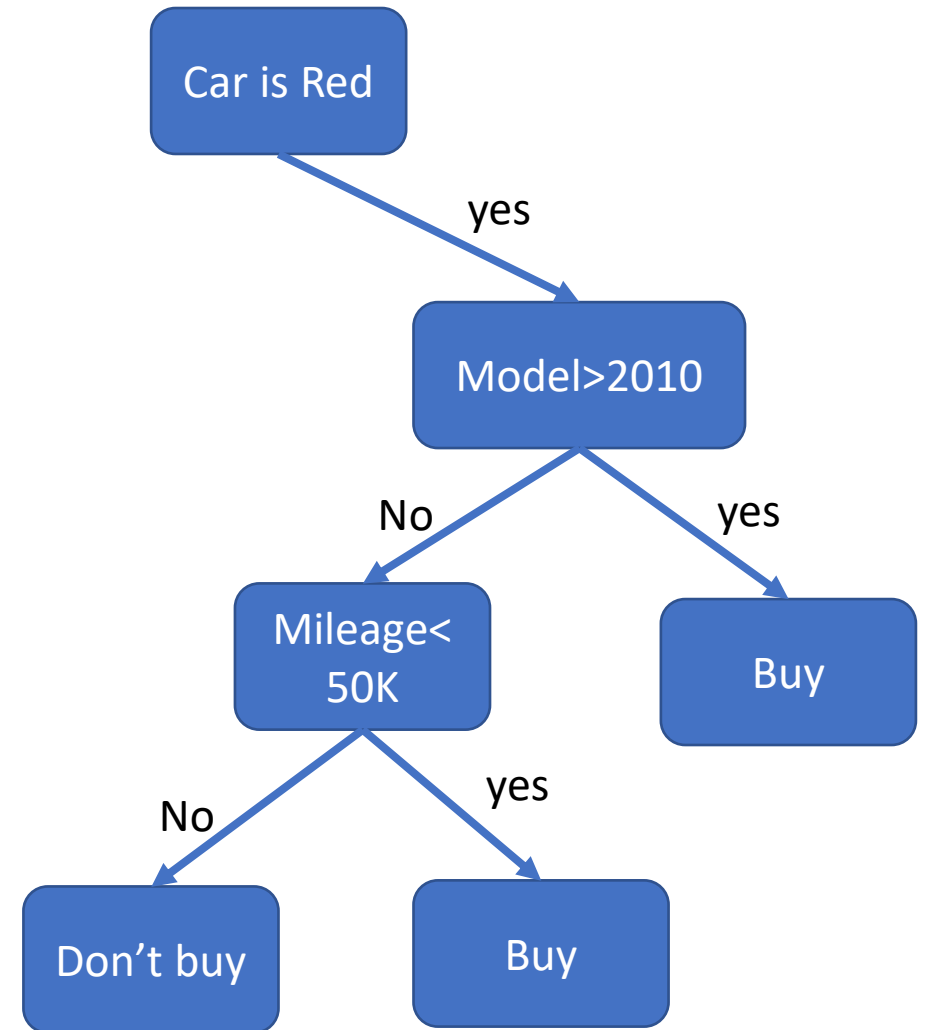
Relation between Rules and decision tree

Decision tree can be converted into a set of decision rules:

- Each path from the root node to the leaf node in the tree can be represented by one rule.
- Rule conclusion represents the leaf node.

Example:

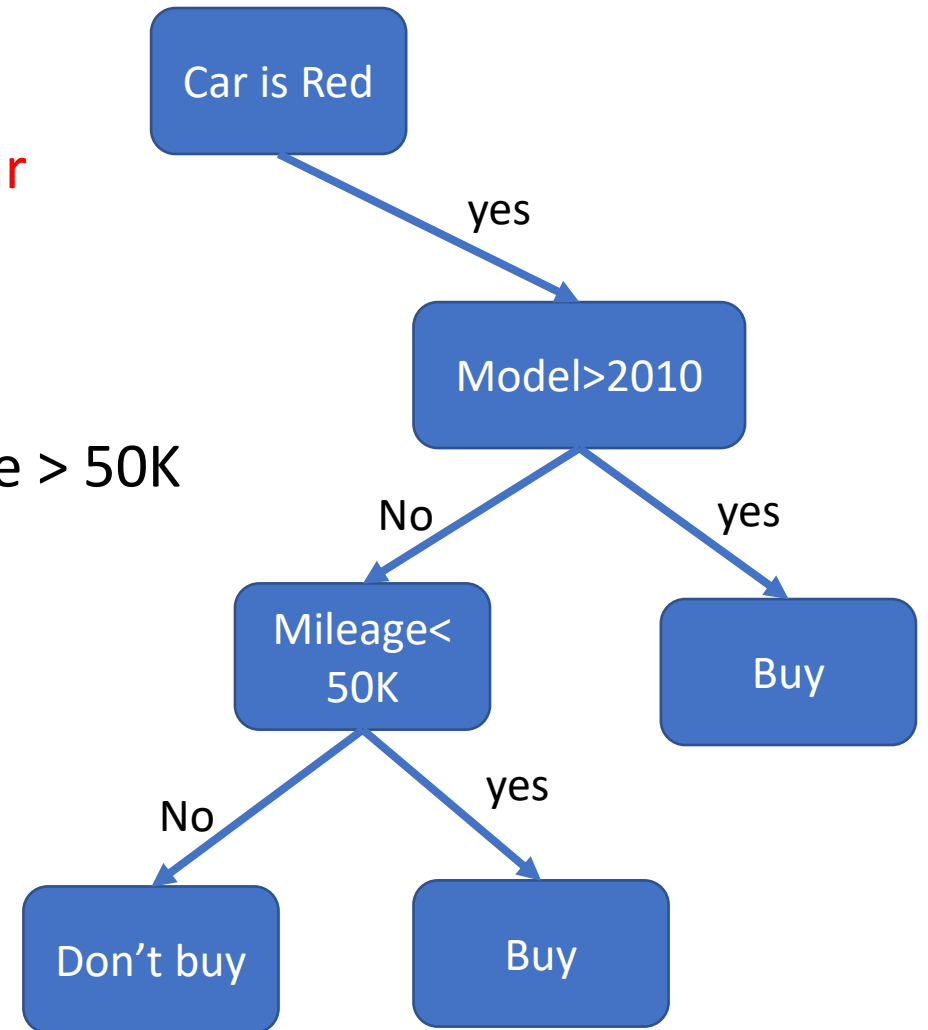
If the car is red and the model < 2010 and the Mileage < 50K then **buy the car (outcome)**



Relation between Rules and decision tree

If the car is red and the model > 2010 then **buy the car** (outcome)

If the car is red and the model < 2010 and the Mileage $> 50K$ then **Don't buy the car** (outcome)



Extracting rules from the tree

- Each leaf (outcome) in the tree is represented by a rule.
 - Antecedent is the conjunction of the value at each node.
 - The consequence is the outcome(class) of the leaf node.
 - The order of the execution is not important for the rules of the tree.
 - Pruning to remove the redundant rules is important to avoid over fitting.

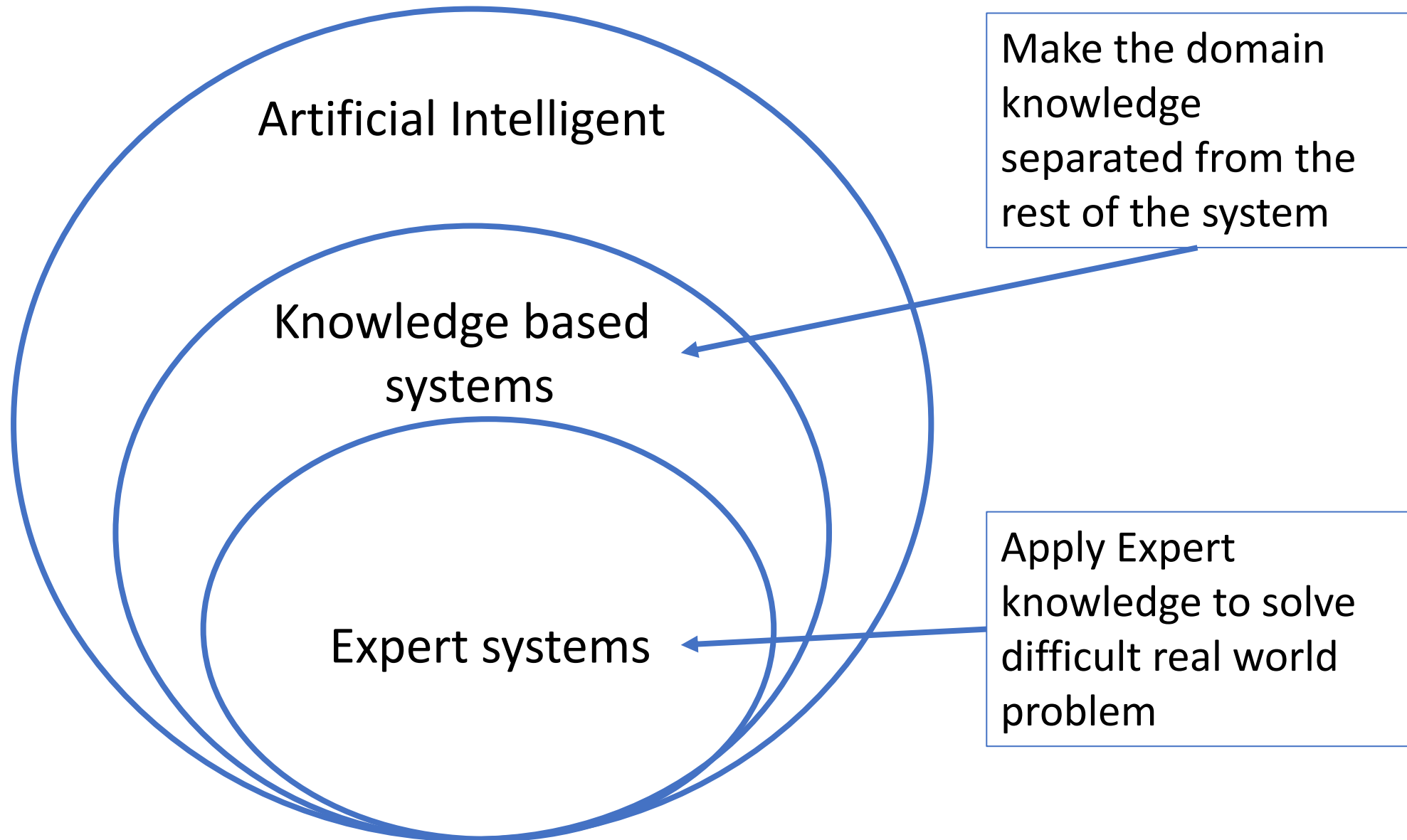
Converting rules to a tree is not simple



Expert Systems (ES)

Expert Systems (ES):

- Definitions:
 - “Computer system that operates by applying an inference mechanism to a body of specialist expertise represented in the form of knowledge”.
 - “A program designed to solve problem at a level comparable to the human expert in given domain”.
 - “A computer system that emulate, or acts in all respects, with the decision-making capabilities of human expert in a limited domain”.



Why do we need to develop/ use ES?

- Not enough human experts are available (e.g. Clinicians).
- Very expensive to have a human expert in a field.
- Human experts may not be available when we need them.
- Human experts may face difficulties in applying their knowledge in urgent situations.
- You can easily lose the experts (retirement, or resignations) .
- ES are able to document all the decisions.
- ES are able to teach or explain a complex knowledge.

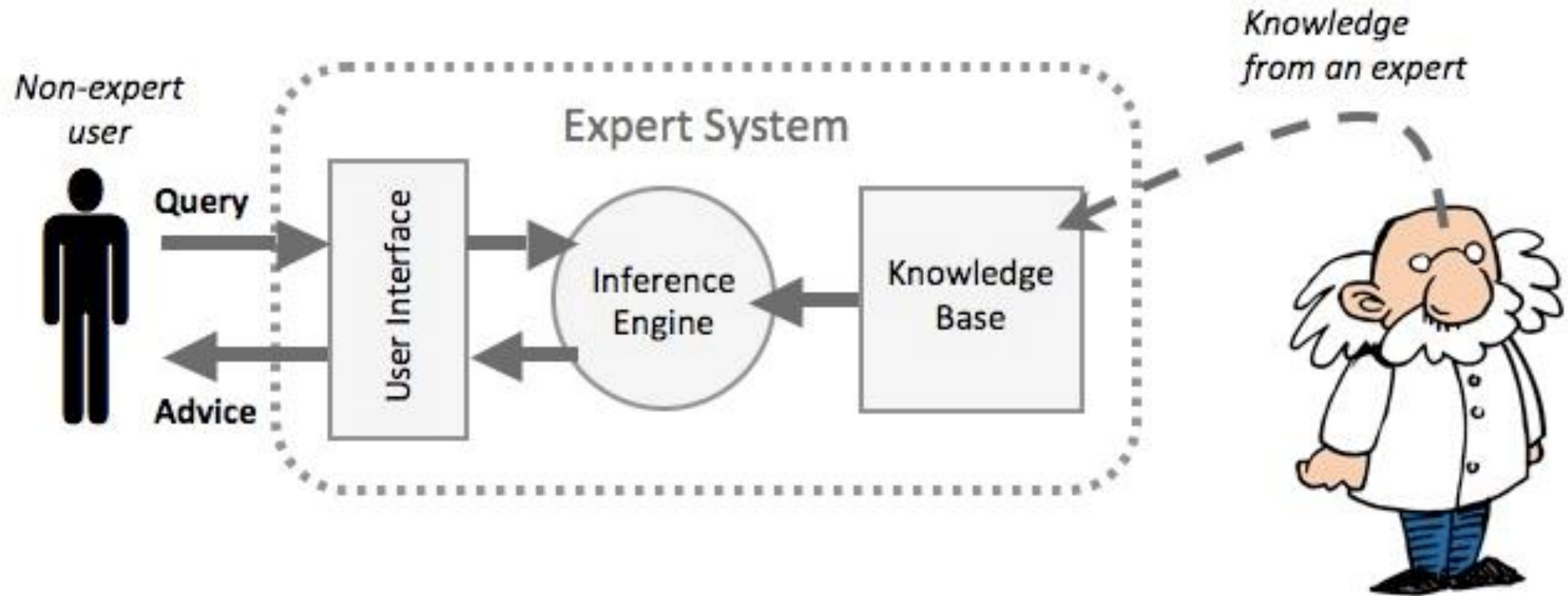
Applications of ES

- Medicine:
 - Disease diagnosis.
 - Healthcare management.
- Finance:
 - Credit analysis system.
- Manufacturing
 - Machine fault detection.
 - Robotics.

ES VS. the rest of AI

- ES can deal with problems with considerable complexity.
- The subject of ES is narrow and specialised.
- ES applies human expert solution to complex problems.
- The knowledge within ES is separated from the processing engine which provide plug and play capability.
- The conclusions of ES are interpretable.
- ES has flexibility in incorporating new information or knowledge.

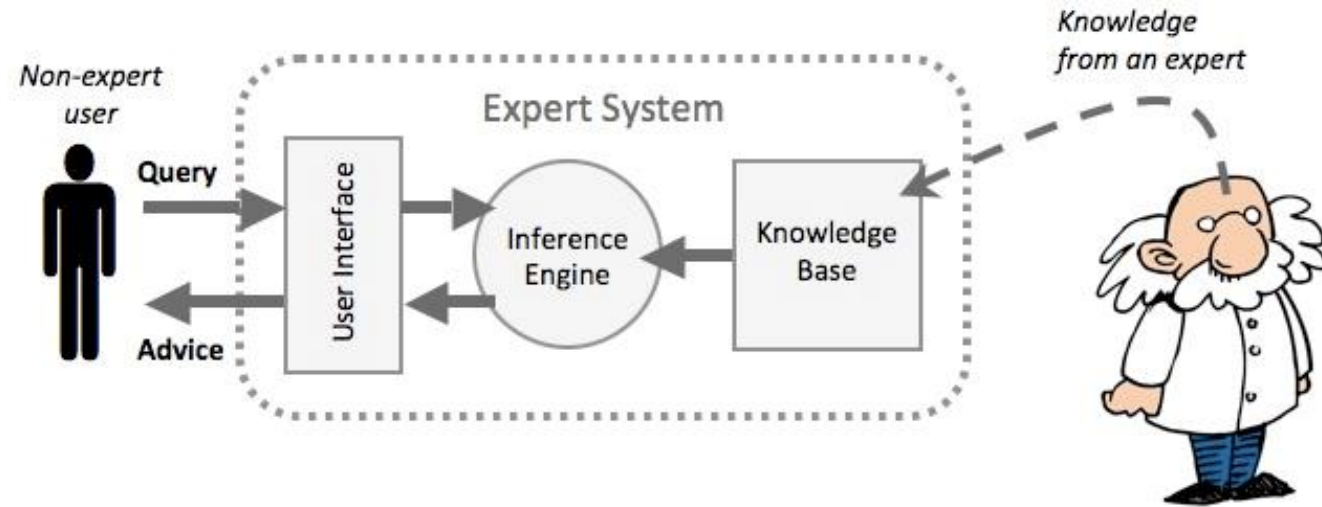
Expert system block diagram



The main aim of ES:

To provide support to human in making decisions related to a real world problem

Key Components of ES

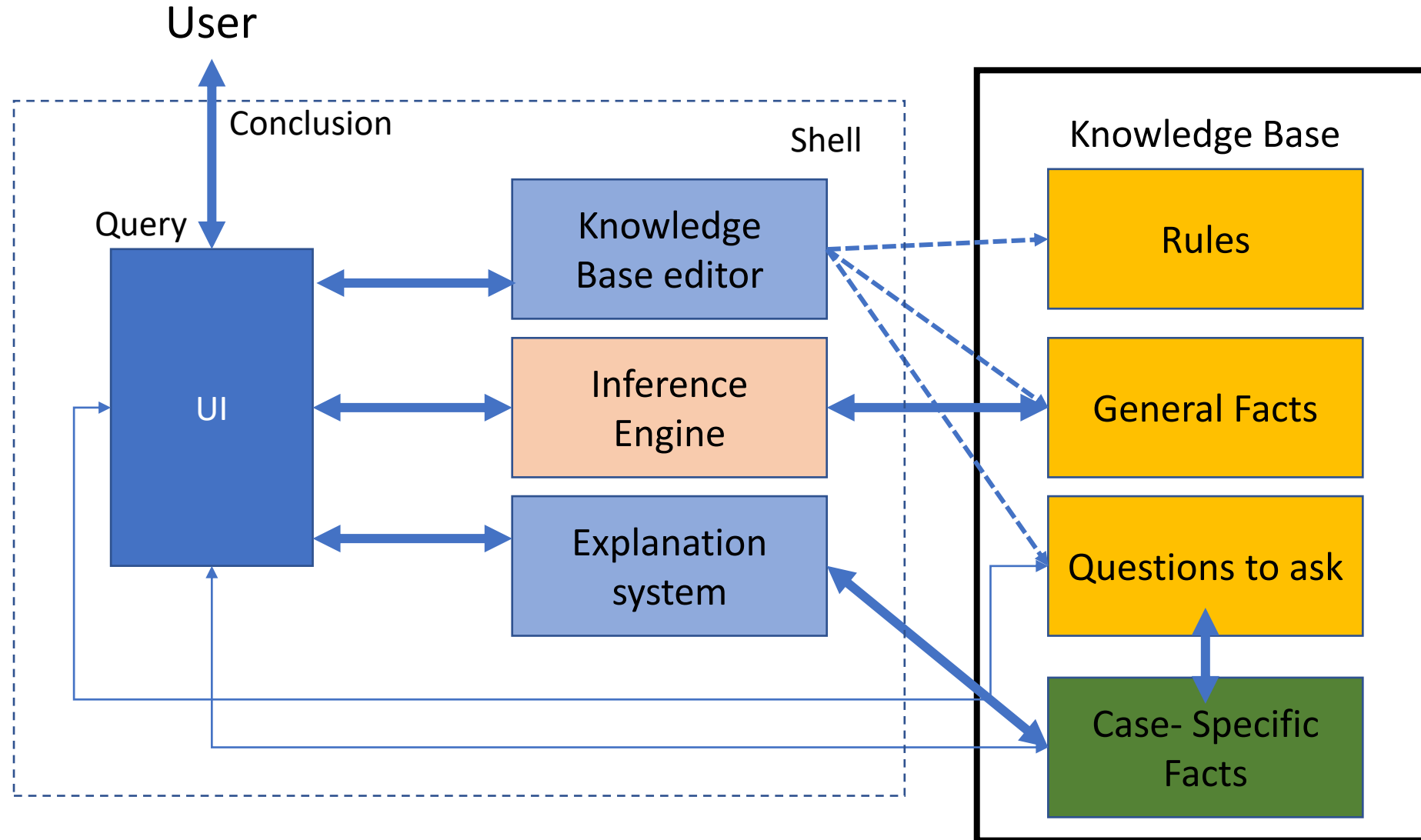


- Knowledge base:
 - Used to store the knowledge which can be represented as Rules or Trees, or Semantic network, or Bayesian network etc.
- Inference Engine:
 - It is responsible for reasoning in other word finding which rule conclusions are satisfied.
 - The inference engine should match the representation format of the knowledge.
 - It is responsible for firing (activating) the consequence of the selected rule.

Other component:

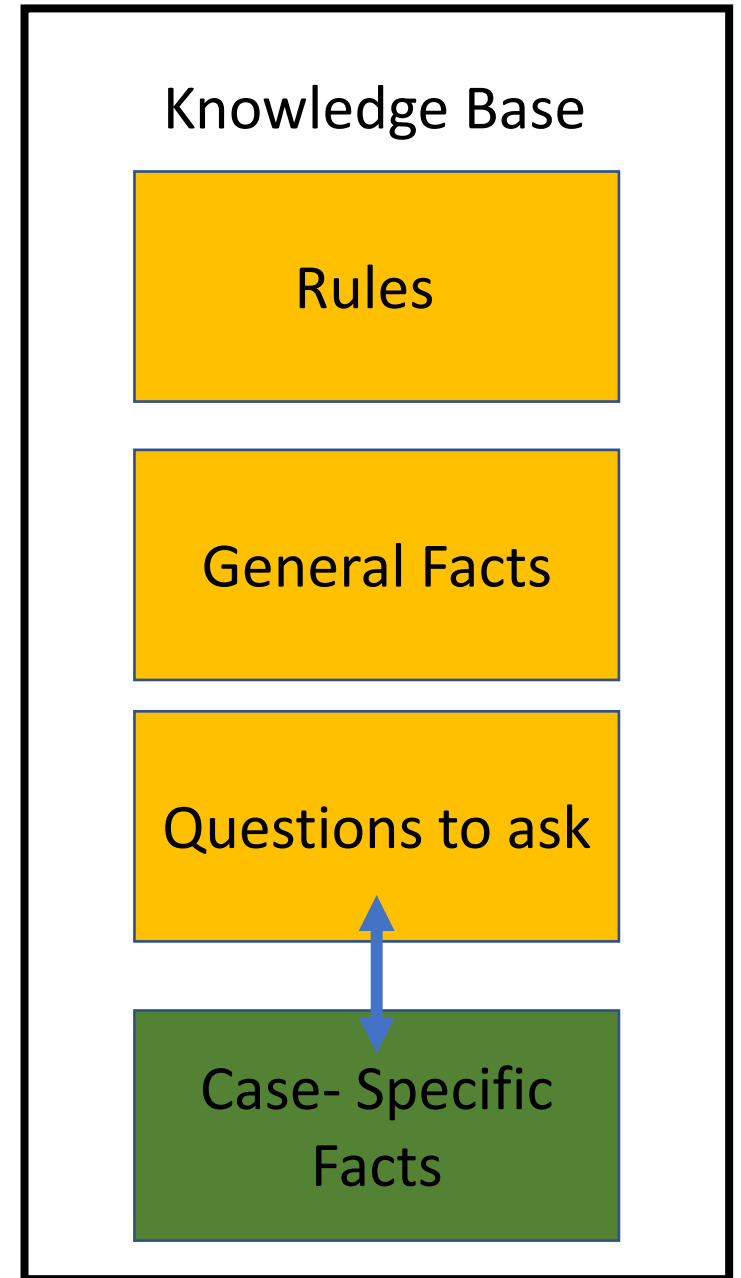
- User interface: The component that the user uses to interact with the system.
- Knowledge based editor: It allows the user to enter new knowledge to the system without any need for coding expertise.
- Explanation system: It Justifies and explains the conclusion in the form of the natural language.

Architecture of ES



Knowledge Base

- Holds the expertise that the system can deploy.
- Developed for a specific expert task.
- This component is independent from the rest of ES.



Knowledge Base

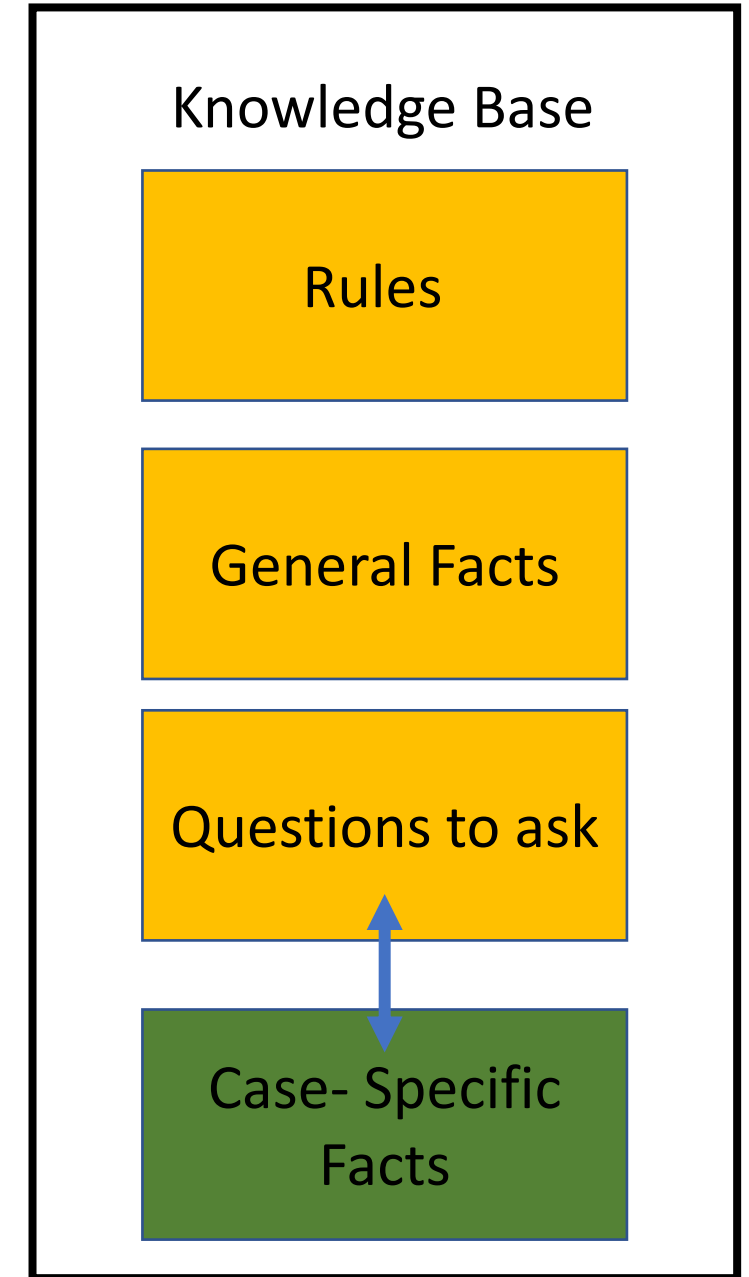


Long term memory



Short term (case specific facts)

- Rule Base:
 - The rule that model the domain knowledge.
- Fact Base:
 - General facts.
 - Key information.
- Question Base:
 - They are set of question written in natural language that the system used to ask the user to input specific facts.
 - The answers of these questions are used by the inference engine to fire the rule.
- Case-specific facts:
 - The facts that are provided by the user as response to the query.



Rule Base

- Unordered non-repeated rules, they can be up to thousands of rules if the problem is complex.
- Only simple rules are used (IF, AND, THEN), OR can be broken into individual rules.
- Example of Rules:
 - IF son of(son, father)
 - Then father of(father, sone)

Fact Base

- Unordered set of facts that includes values for the propositions.
- Facts should follow the same syntax of the rules to allow the inference engine to match the rules with the required facts.

Example of Rules:

IF son of(son, father)

Then father of(father, son)

Example of Facts:

son of(Mike, Simon)

son of(Chris, Neil)

The facts that Mike is the son of Simon, and Chris is the son of Neil are stored in Fact Base.

Question Base

- Unordered set of questions that are applied by ES to get an input from the user. The answers can take the following forms:
 - Yes/No
 - True/False.
 - One answer from multiple choices.
 - A value.
- Example:
 - Who father(son) → who is your father?
 - Answer= select 1
 - Niel
 - Mike
 - Chris
 - Phil

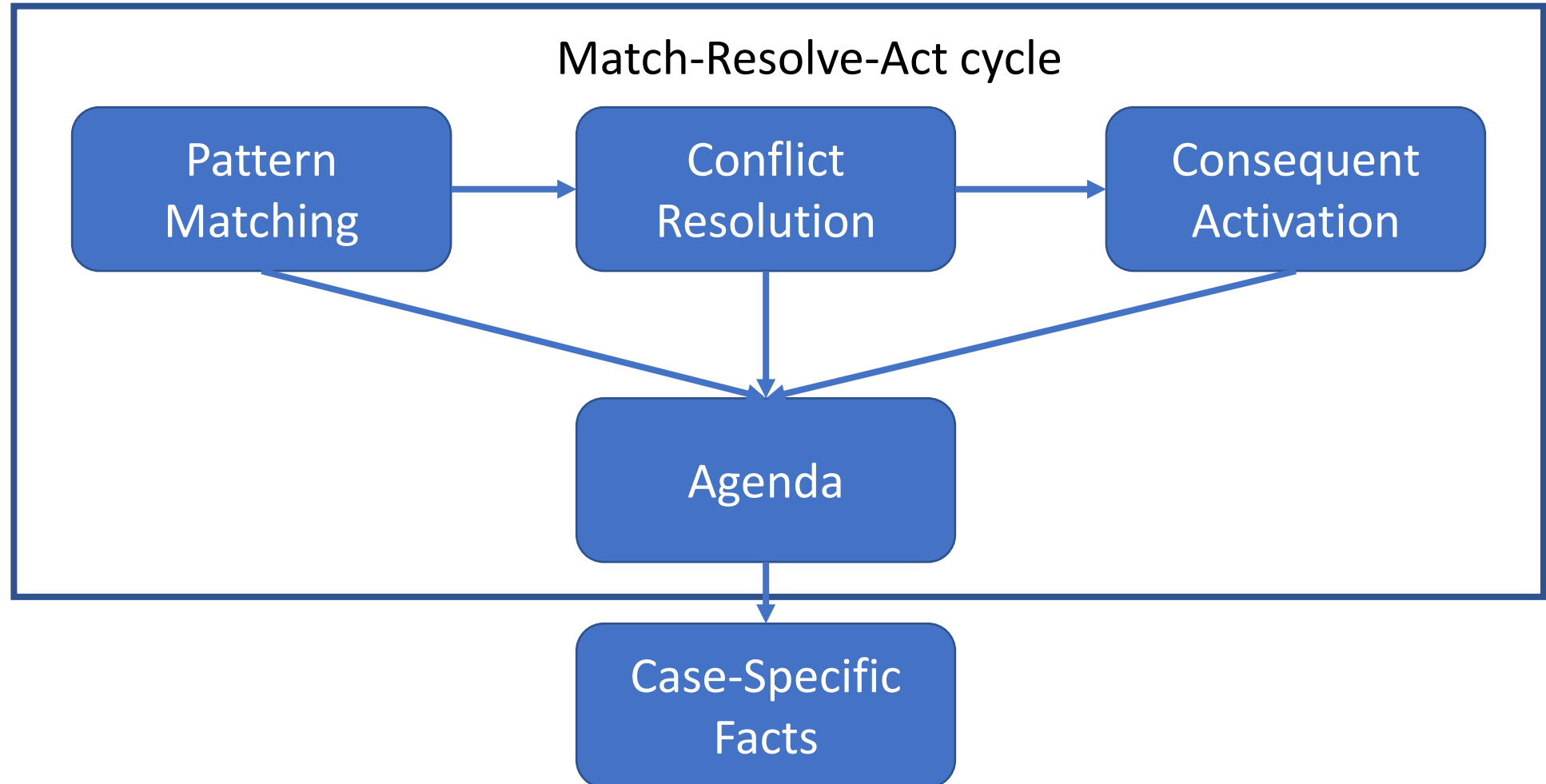
Inference engine

- Plays the central role in how the ES works. It is known as the rule interpreter, and also as the brain of the system.
- Interacts with the knowledge base
 - Control how the rules are applied to the facts.
- Resolves conflict between conclusions.
- Interacts with the user interface.
 - Allows using input via questions.
- Links rules with facts to generate new facts.
- Uses forward or backward chaining.

Rule inference

- Forward chaining (data driven):
 - Works by matching the fact with the IF part of the rule.
 - Starts with a fact in the knowledge base.
 - Finds the applicable rules and apply them.
 - Work well with control ES.
- Back chaining (goal based):
 - Matches the goal with the THEN part of the rule.
 - Starts with the user entering the goal
 - Looks for the rule that decompose a specific goal; then solve the smaller goals.
 - Works well for diagnosis ES.

Inference engine structure



Rule Matching

- Matching is the process of searching the knowledge to find the applicable rules.
- The matched rules are added to the agenda.
- When the set of rules is huge, matching facts against rules can be computationally expensive.

Variable Binding

- Binding Variables is:
 - Variables in rules acquire values during the matching process.
- Rule antecedent and consequent may contain variables.
- Example:
 - Rule:
 - If son_of(\$son,\$father)then father_of(\$father,\$son)
 - Fact:
 - Son_of(\$Mike,\$Chris)
 - Daughter_of(\$Linda,\$ Chris)
- Assume we do forward chaining, we look at the fact, the variables are bounded : Mike to \$son, and Chris to Father.
- When the rule is executed new fact (father_of(,\$Chris, (\$Mike)) will be added to case specific facts(working memory)

Conflict Resolution

- Rules matching often produce more than one applicable rule which are kept in the agenda component.
 - The question here which one should be fired first, how we should order them?
 - Is only one rule should be fired like one decision should be take, then we have to resolve this conflict to pick one.
- Selecting the rule with the highest priority from the agendas, but how can be measure the priority?

Conflict Resolution strategies

- Flow the order in rule base, and fire the first rule in the order.
 - Not preferred for the large rule set, as we need to order the rules in the knowledge base during the process of generating the rules by the human experts. This can be time consuming.
- Picking the rule that is more specific, if the condition of one rule is a subset of the condition of the second rule, select the second one.
 - For example:
 - if we have the two rules: IF A AND B THEN D & IF A THEN D , pick the first rule because it is more specific.
- Follow alphabetical order.

Conflict Resolution strategies

- Pick the rule with the highest priority. In this case the rules in the knowledge base should be combined with priority score.
- Pick the rule that is the most recently added or modified. This requires a time stamp for each rule.

Consequent Activation

- This component activates the selected rule which may generate new fact that will be added to Case Specific facts component.
- The rule is removed from the agenda once it executed.
- The cycle of the execution is ended when there is no any more rules in the agenda.

User interface

- It is used to allow the user to interact with the system:
 - Edit facts or rules.
 - Input data or answer questions.
 - make query.
- It can be connected to all the components in the system.
- It can be graphical user interface that allow the user to answer questions and input new knowledge or update old rules.

Knowledge Base editor

- It also known as knowledge acquisition facility. Which allow the integration of new knowledge or editing existing knowledge.
- It provides an automatic way to enter the knowledge rules into the system without any need for coding expertise.

Explanation System

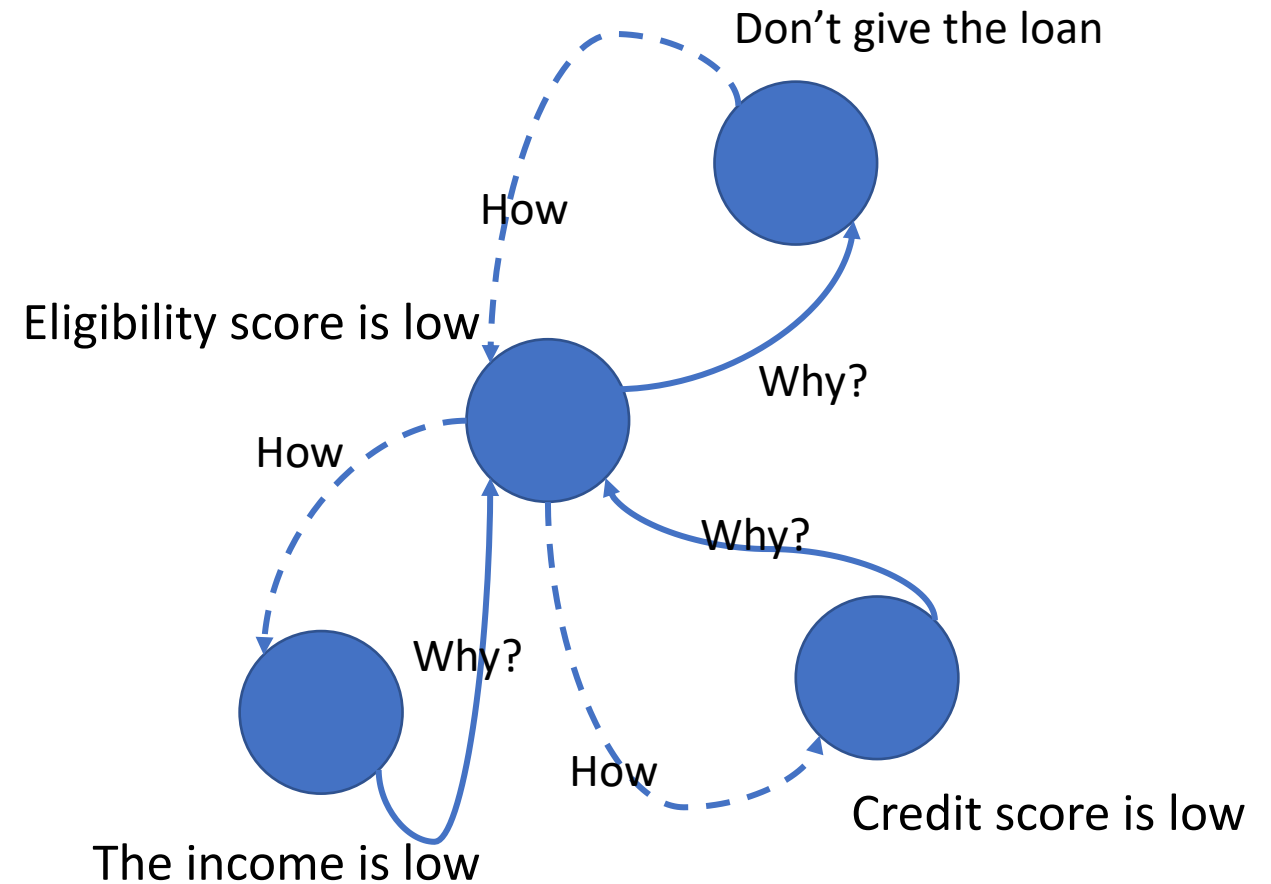
- It is also known as justifier. It explains the reason for producing such conclusions.
 - Which rules have been fired, and which facts have been used.
 - Why was another conclusion not produced?
 - Why a specific fact is used or not used?

In some fields users prefer to understand the reasoning for the conclusions, for e.g. clinicians like to know why this recommendation was reached.

Explanation System Example

The decision of not approving the loan can be because of bad credit history, the income is low or both of them together.

To know why, we can flow all the why down the chain of the questions.



Other Types of ES

- Fuzzy ES:
 - Applying the fuzzy logic with rules.
- Frame-based ES:
 - Uses frame representation for the knowledge.
- Hybrid ES:
 - Apply RULES and ANN.

Advantages of ES

- ES uses natural knowledge representation: (IF-THEN)
 - Easy to read rules.
 - Rules have similar structure
 - Rules are Interpretable.
 - Each rule represent independent amount of knowledge.
- Allows the transfer of knowledge, which may be needed in less developed or remote areas.
- Can be developed in stages, as new rules can be added later, and the existing rules can be modified.
- ES can deal with incomplete information.
- The output (conclusions, advices, actions, ..etc) of the ES are consistent for repetitive processes.

Advantages of ES

- Allows combining the knowledge from multiple human experts (e.g. many clinicians can collaborate to generate the rules).
- Can standardise the decision making process.

Disadvantages of ES

- They are very specific and can't be generalised.
- They may fail to come to a conclusion.
- They are not data driven. Therefore, they are not creative and can't learn.
- It can be slow and unsuitable for real time applications.
- Can't provide a conclusion about new situations.
- They are not adaptive to the change in the environment.

Summary

knowledge representation.

Expert Systems (Why, what, when?)

- Rule-based systems
- Knowledge base
- Inference Engine
- Other components.

Other Types of Expert Systems

Advantages and disadvantage.