ADVANCED KNOWLEDGE ENGINEERING

Instructor:

KHUAT Thanh Tung

University of Technology Sydney

Subject’s Outline

- **Topic 1:** An Overview of Knowledge Engineering
- **Topic 2:** An Overview of Knowledge-based Systems
- **Topic 3:** Knowledge Acquisition
- **Topic 4:** Knowledge Representation and Reasoning

Mid-term assessment

- **Topic 5:** Ontology
- **Topic 6:** Knowledge Graphs
- **Topic 7:** Expert Systems
- **Topic 8:** Uncertain Reasoning
- **Topic 9:** Hybrid Knowledge-based Systems
- **Topic 10:** Automated AI Planning

Group projects for the advanced topics

Subject’s objectives

- Understanding the Knowledge Base, Knowledge-based Reasoning and Representation
- Learn how to build fundamental components of Knowledge-based systems
- Learn how to integrate the knowledge base, and knowledge-based reasoning components into Machine Learning Systems
- Figure out several advanced topics related to Knowledge Engineering: Ontology, Knowledge Graphs, Hybrid knowledge-based systems, and Automated AI Planning using knowledge base
- **Programming languages**: General purpose programming language: Python, C++, Java,… + PROLOG, PDDL

Course outcomes

- **Remembering**: Techniques for knowledge acquisition, representation, reasoning, and KBS development.
- **Understanding**: Fundamental components of Knowledge-based systems, methods of building a specific KBS
- **Applying**:
  - Have the ability to build small applications using knowledge and related components
  - Able to use several programming languages to build core reasoning engines within a knowledge-based system such as PROLOG
- **Analyzing**:
  - Analyze the given problems and choose suitable techniques, knowledge presentations and a KBS
  - Able to integrate the knowledge engineering into machine learning and AI systems.

Assessment methods

- Learning progress assessment (**20%**):
  - Random attendance checking
  - Small exercises
- Mid-term: A set of quizzes based on the lectures (**20%**)
- Final assignment (**60%**): A group project of 3-5 students for provided topics.

References

1. Textbook: **An Introduction to Knowledge Engineering**, written by S.L. Kendal and M. Creen, published by Springer
2. **An Introduction to Knowledge Graphs**, written by Umutcan Serles and Dieter Fensel, published by Springer
3. **Logic Programming with Prolog**, written by Max Bramer, published by Springer
4. **Knowledge Engineering Tools and Techniques for AI Planning**, edited by Mauro Vallati and Diane Kitchin, published by Springer
5. **Ontologies - A Handbook of Principles, Concepts and Applications in Information Systems**, edited by Raj Sharman, Rajiv Kishore, Ram Ramesh
6. **Giáo trình các hệ cơ sở tri thức**, biên soạn bởi GS. Hoàng Văn Kiếm, TS. Đỗ Phúc, TS. Đỗ Văn Nhơn, xuất bản bởi Nhà xuất bản đại học quốc gia thành phố Hồ Chí Minh

AN OVERVIEW INTRODUCTION TO KNOWLEDGE ENGINEERING

TOPIC 1

Objectives of this topic

By the end of this topic, you will be able to:

- ✓ define knowledge and explain its relationship to data and information
- ✓ distinguish between knowledge management and knowledge engineering
- ✓ explain the skills required of a knowledge engineer
- ✓ define Knowledge-based systems
- ✓ explain what a KBS can do

Agenda

- What is knowledge?
- Types of knowledge
- Knowledge engineering
- Knowledge engineers

Philosophical Basis

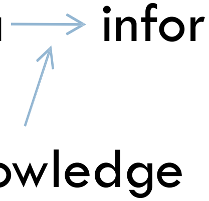
- Traditional questions that have been analyzed by philosophers, psychologists, and linguist:
  - What is knowledge?
  - What do people have inside their head when they know something?
  - Is knowledge expressed in words?
  - If so, how could one know things that are easier to do than to say, like tying a shoestring or hitting a baseball?
  - If knowledge is not expressed in words, how can it be transmitted in language?
  - How is knowledge related to the world?
  - What are the relationships between the external world, knowledge in the head, and the language used to express knowledge about the world?

Philosophical Basis

- With the advent of computers, the questions addressed by the field of artificial intelligence (AI) are:
  - Can knowledge be programmed in a digital computer?
  - Can computers encode and decode that knowledge in ordinary language?
  - Can they use it to interact with people and with other computer systems in a more flexible or helpful way?

Information Processing Views of Knowledge

- □ **Hierarchical view:** data → information → knowledge
  - Data are streams of raw facts representing events before they have been arranged into a form that people can understand and use
  - Information is the input or raw material of new knowledge
  - Knowledge is authenticated/personalized information
- □ **Reversed hierarchical view:** knowledge → information → data
  - Knowledge must exist before information can be formulated and before data can be collected
- □ **Non-hierarchical view:** data → information



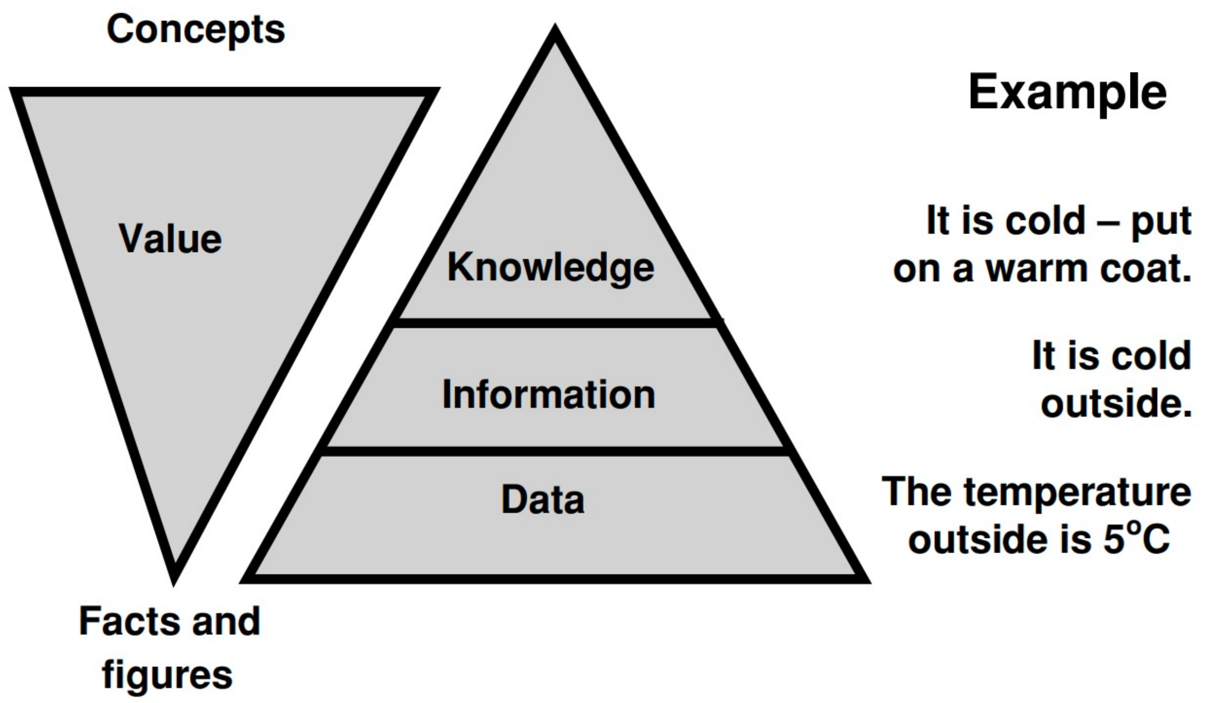
Knowledge

  - Knowledge is needed in converting data into information
  - Knowledge is the accumulation of experiences vs. knowledge is created through conjectures and refutations.

Information Processing Views of Knowledge

□ The movement from data to knowledge implies a shift from facts and figures to more abstract concepts

□ From a knowledge engineering perspective, knowledge can be considered as something that can be expressed as a rule or useful to assist a decision



Alternative Perspectives on Knowledge

- Knowledge can be defined as a justified belief that increases an entity's capacity for effective action.
- It may be viewed from several perspectives:

  (1) a state of mind – knowledge is the state of knowing and understanding

  (2) an object – knowledge is an object to be stored and manipulated

  (3) a process – knowledge is a process of applying expertise

  (4) a condition – knowledge is organized access to and retrieval of content

  (5) a capability – knowledge is the potential to influence action

Taxonomies of Knowledge

- Tacit vs. explicit
  - Explicit knowledge refers to knowledge that is transmittable in formal, systematic language
  - Tacit knowledge is deeply rooted in actions, experience, and involvement in a specific context. It consists of cognitive element (mental models) and technical element (know-how and skills applicable to specific work).
- Individual vs. social
  - Individual knowledge is created by and exists in the individual whereas social knowledge is created by and exists in the collective actions of a group.

Taxonomies of Knowledge

- Five Types of Knowledge
  - Declarative knowledge
    - Know-about
  - Procedural knowledge
    - Know-how
  - Causal knowledge
    - Know-why
  - Conditional knowledge
    - Know-when
  - Relational knowledge
    - Know-with
- Meta-knowledge
  - Knowledge about knowledge

Four Modes of Knowledge Conversion (Nonaka, 1991)

To

From	Tacit knowledge	Explicit knowledge
Tacit knowledge	Socialization	Externalization
Explicit knowledge	Internalization	Combination

Socialization is the process of sharing tacit knowledge through observation, imitation, practice, and participation in formal and informal communities.
Externalization is the process of articulating tacit knowledge into explicit concepts
Combination is the process of integrating concepts into a knowledge system.
Internalization is the process of embodying explicit knowledge into tacit knowledge

Knowledge Engineering

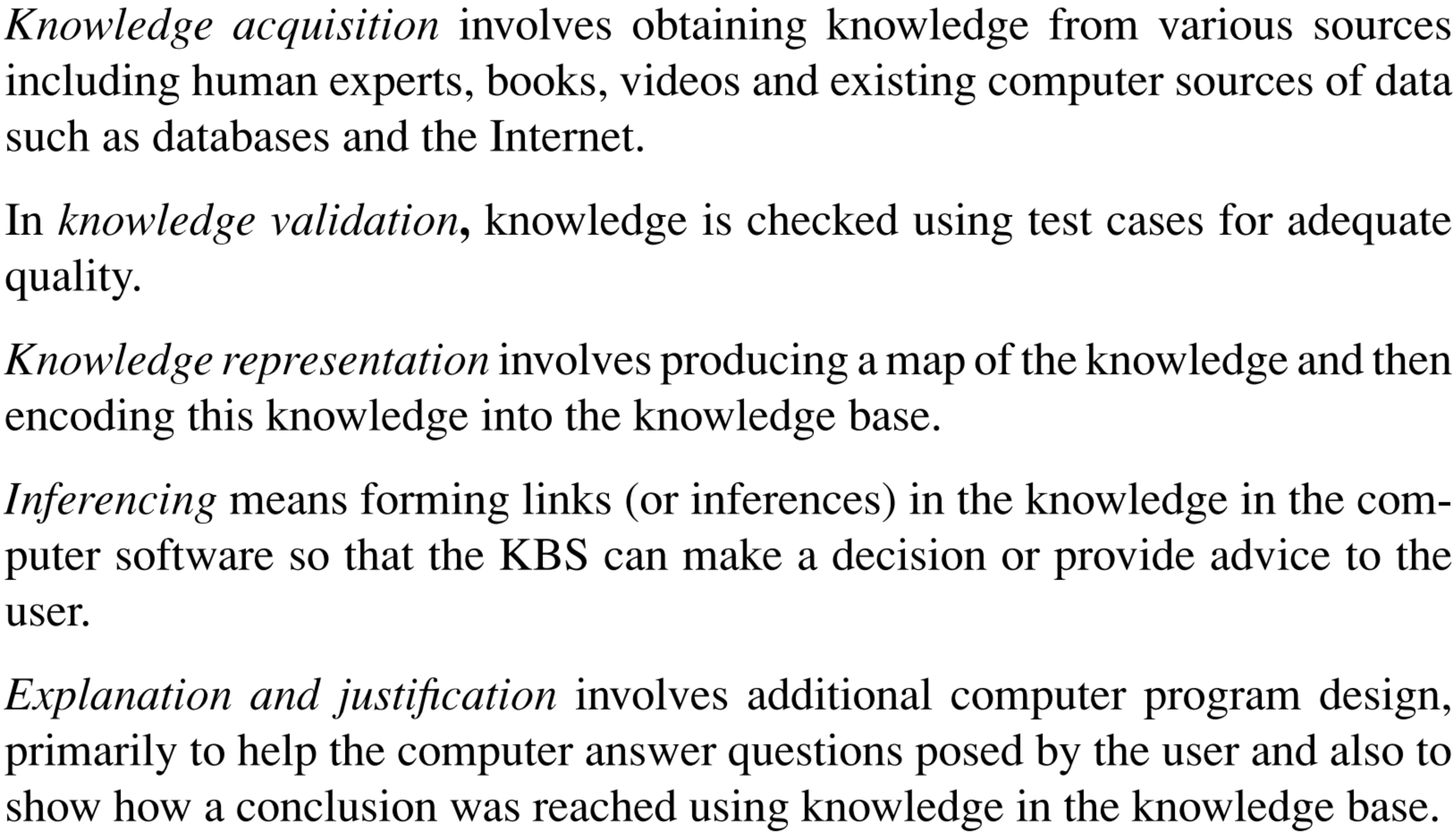
- An engineering discipline that involves integrating knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise (Feigenbaum and Pamela, 1983)

- It normally involves five distinct steps in transferring human knowledge into some form of knowledge based systems (KBS)

Five Steps of Knowledge Engineering

- Knowledge acquisition
- Knowledge validation
- Knowledge representation
- Inferencing
- Explanation and justification

Five Steps of Knowledge Engineering



Knowledge acquisition involves obtaining knowledge from various sources including human experts, books, videos and existing computer sources of data such as databases and the Internet.

In *knowledge validation*, knowledge is checked using test cases for adequate quality.

Knowledge representation involves producing a map of the knowledge and then encoding this knowledge into the knowledge base.

Inferencing means forming links (or inferences) in the knowledge in the computer software so that the KBS can make a decision or provide advice to the user.

Explanation and justification involves additional computer program design, primarily to help the computer answer questions posed by the user and also to show how a conclusion was reached using knowledge in the knowledge base.

Two Main Views of Knowledge Engineering

- Transfer view – This is the traditional view. In this view, the key idea is to apply conventional knowledge engineering techniques to transfer human knowledge into the computerized system.
- Modeling view – In this view, the knowledge engineer attempts to model the knowledge and problem solving techniques of the domain expert into the computerized system.

Knowledge Engineering (KE) vs. Knowledge Management (KM)

- KE is primarily concerned with constructing a knowledge-bases system while KM is primarily concerned with identifying and leveraging knowledge to the organization’s benefit.
- KE and KM activities are inherently interrelated.
- Knowledge engineers are interested in what technologies are needed to meet the enterprise’s KM needs.

Knowledge Engineers

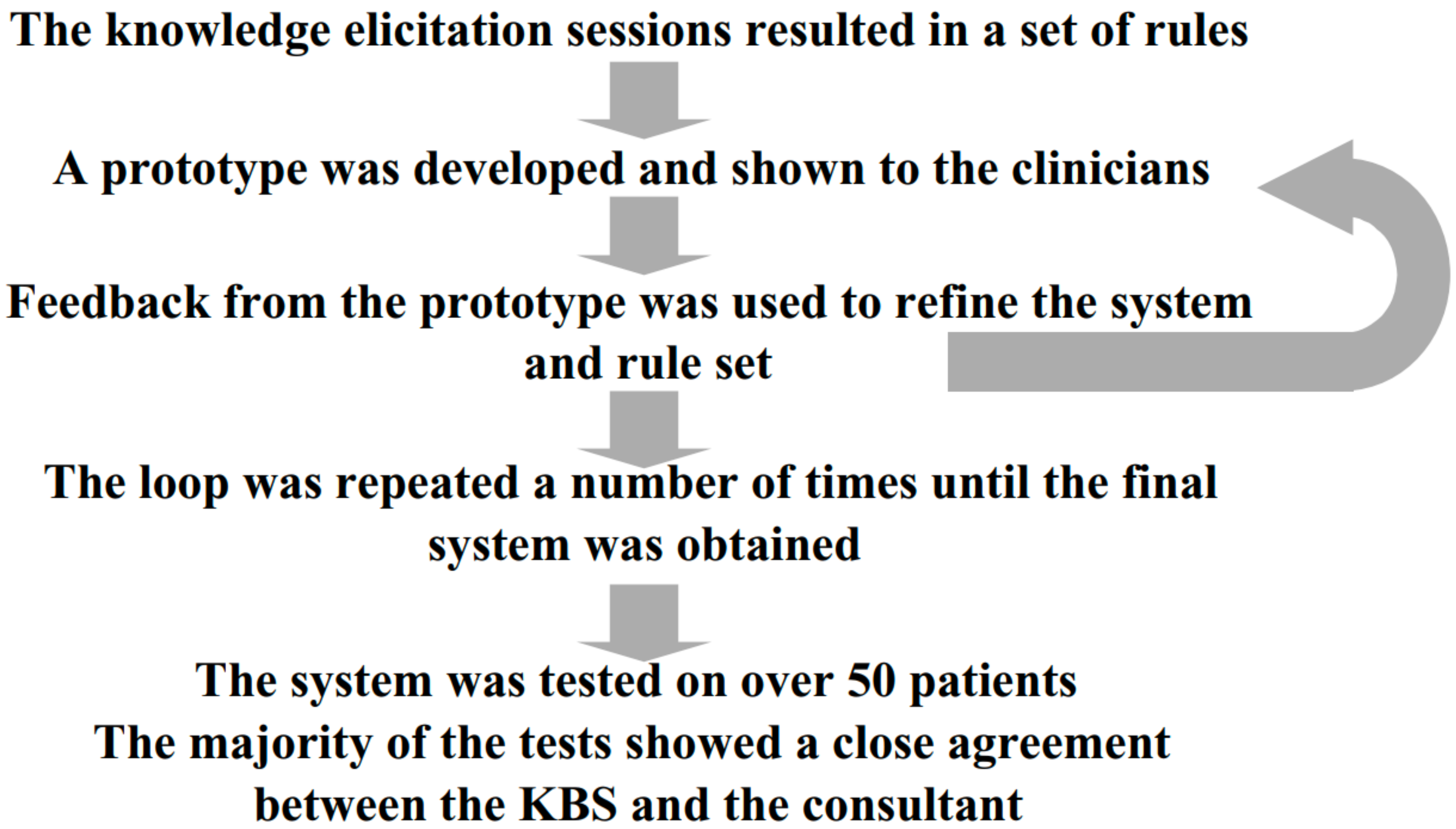
- A knowledge engineer is responsible for obtaining knowledge from human experts and then entering this knowledge into some form of KBS.
- In developing KBS, the knowledge engineer must apply methods, use tools, apply quality control and standards, plan and manage projects, and take into account human, financial, and environmental constraints.
- Required skills of a knowledge engineer
  - Knowledge representation
  - Fact finding (knowledge elicitation)
  - Human skills
  - Visualization skills
  - Analysis
  - Creativity
  - Managerial

An example of a KBS Project

- ❑ The **goal**: assist the clinician in the intensive care unit (ICU)
- ❑ Problems need to be addressed:
  - ❑ interpretation of measurement values with respect to historical information about changes in a patient's status and therapy.
  - ❑ The difficulty of directly relating measurement values to a therapeutic recommendation
- ❑ Tasks of the system:
  - ❑ Predict the initial setting of the mechanical ventilator to assist the patient to breathe
  - ❑ Suggest adjustments to treatment by continuous reassessment of the patient's condition
  - ❑ Summarize the patient's physiological status
  - ❑ Maintain a set of patient's specific expectations and goals for future evaluations
  - ❑ Aid in the stabilization of the patient's condition

An example of a KBS Project

- ❑ KBS development process



The knowledge elicitation sessions resulted in a set of rules

A prototype was developed and shown to the clinicians

Feedback from the prototype was used to refine the system and rule set

The loop was repeated a number of times until the final system was obtained

The system was tested on over 50 patients
The majority of the tests showed a close agreement between the KBS and the consultant

An example of a KBS Project

- ❑ One of the main queries in the project was from the experts providing knowledge for the system
  - ❑ it was essential that the system provided accurate answers, otherwise patients lives could be at risk
- ❑ Experts providing the knowledge did not want to be blamed if an incorrect response was given by the KBS

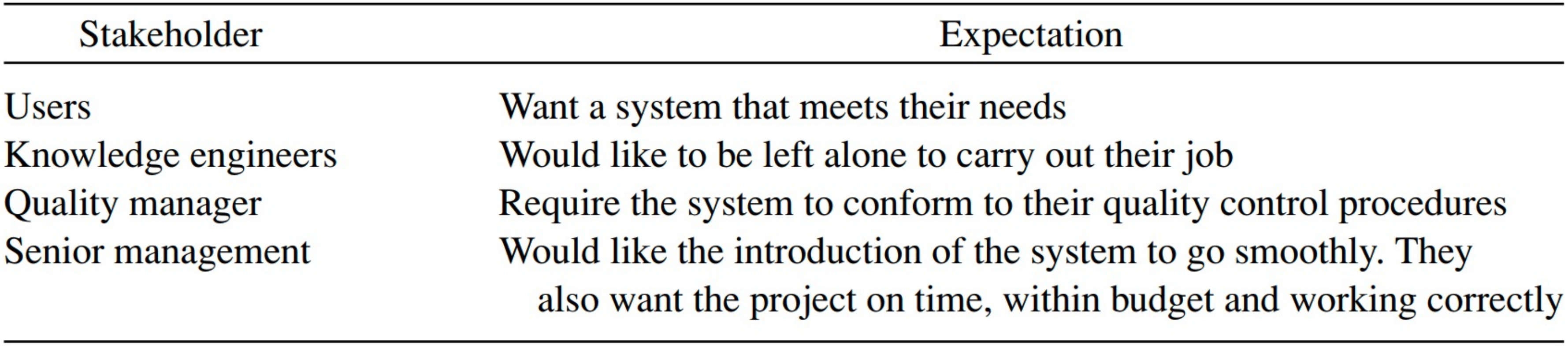
Whose fault is it if the system fails?
Has the system been developed to quality standards?
Do we really know if it is correct?

An example of a KBS Project

- The knowledge engineers and project managers need to provide the main assurance that the built system is in accordance with quality assurance standards
- Quality assurance is an essential part of the design of any KBS, especially
  - railway signaling systems
  - alarm systems
  - detection of gas leaks
  - nuclear power station monitoring and control
  - Other critical systems in which an error could result in significant risk, including loss of life

An example of a KBS Project

- The Project Manager’s Dilemma
  - Like the general software project management, a project manager for the KBS project needs some negotiating skills to try and match the expectations of all parties involved in a project



Stakeholder	Expectation
Users	Want a system that meets their needs
Knowledge engineers	Would like to be left alone to carry out their job
Quality manager	Require the system to conform to their quality control procedures
Senior management	Would like the introduction of the system to go smoothly. They also want the project on time, within budget and working correctly

KNOWLEDGE-BASED SYSTEMS

TOPIC 2

Objectives of this topic

By the end of this topic, you will be able to:

- ✓ describe the characteristics of a knowledge-based system
- ✓ explain the main elements of knowledge-based systems and how they work
- ✓ evaluate the advantages and limitations of knowledge-based systems
- ✓ identify appropriate contexts for the use of particular types of knowledge-based systems
- ✓ distinguish expert systems from machine learning systems
- ✓ state a brief definition of expert systems, neural networks, case-based reasoning, genetic algorithms, intelligent agents

Agenda

- Expert systems
- Neural networks
- Case-based reasoning
- Genetic algorithms
- Intelligent agents

What are KBSs?

- A knowledge based system is a system that uses artificial intelligence techniques in problem-solving processes to support human decision-making, learning, and action.
- Two central components of KBSs are
  - Knowledge base
    - Consists of a set of facts and a set of rules, frames, or procedures
  - Inference engine
    - Responsible for the application of knowledge base to the problem on hand.
- There are pros and cons of using KBSs, compared to human expertise.

Types of KBSs

- Expert systems
- Neural networks
- Case-based reasoning
- Genetic algorithms
- Intelligent agents

Expert Systems

- An expert system is a computer program designed to emulate the problem-solving behavior of an expert in a specific domain of knowledge
- In order to qualify as an expert system, a system must have the capability of explaining or justifying its conclusions.
- A system which can explain its reasoning process is said to demonstrate meta-knowledge (knowledge about its own knowledge).

Features of Problem Solvers

- Human experts exhibit certain characteristics and techniques which help them perform at a high level in solving problems in their domain.
  - ❑ Solve the problem
  - ❑ Explain the results
  - ❑ Learn
  - ❑ Restructure knowledge
  - ❑ Break rules
  - ❑ Determine relevance
  - ❑ Degrade gracefully

Characteristics of Expert Systems

- ❑ The system performs at a level generally recognized as equivalent to that of a human expert or specialist in the field.
- ❑ The system is highly domain specific.
- ❑ The system can explain its reasoning.
- ❑ If the information with which it is working is probabilistic or fuzzy, the system can correctly propagate uncertainties and provide a range of alternative solutions with associated likelihoods.

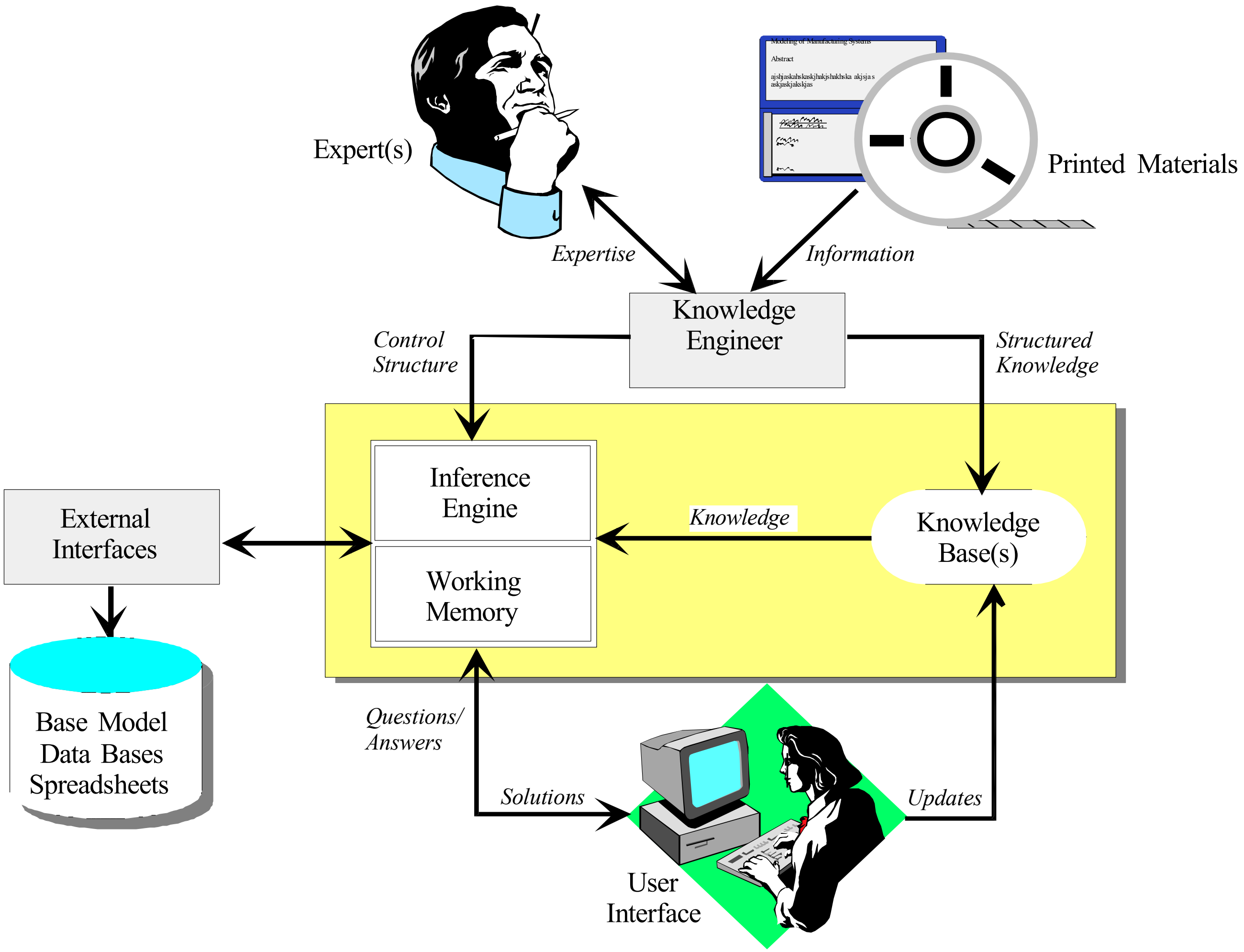
Applications of Expert Systems

- DENDRAL
  - Applied knowledge (i.e., rule-based reasoning)
  - Deduced likely molecular structure of compounds
- MYCIN
  - A rule-based expert system
  - Used for diagnosing and treating bacterial infections
- XCON
  - A rule-based expert system
  - Used to determine the optimal information systems configuration
- New applications: Credit analysis, Marketing, Finance, Manufacturing, Human resources, Science and Engineering, Education, …

Components of Expert Systems

- Knowledge base
  - Consists of facts and rules
  - Rules are commonly expressed in if-then structure (production rules)
    - If-premise then conclusion
    - If-condition then action
- Inference engine
  - Responsible for rule interpretation and scheduling
  - Forward chaining vs. backward chaining
- User interface
- Working memory
- Explanation facility

Conceptual Architecture of a Typical Expert System



Expert(s)

Printed Materials

Expertise

Information

Knowledge Engineer

Control Structure

Structured Knowledge

Inference Engine

Working Memory

Knowledge

Knowledge Base(s)

External Interfaces

Base Model Data Bases Spreadsheets

Questions/ Answers

Solutions

Updates

User Interface

Expert System Building Tools

- Programming language
  - An expert system can be implemented using a general purpose programming language. However, the programming language LISP and PROLOG are typically used in expert systems implementation, in particular Artificial intelligence applications.
- Shells
  - A shell consists mainly of an inference engine and an editor to assist developers in building their knowledge base.
  - Example: CLIPS is an expert system shell developed by NASA

Strengths and Limitations of Expert Systems

- ❑ Strengths
  - ❑ Human expertise can be expensive
    - ❑ Expert Systems only require the costs of hardware to run
  - ❑ Human advice can be inconsistent
    - ❑ Human advice may be adversely affected by tiredness, busy diaries, personal problems, *etc*.
    - ❑ Computer advice will always be based on the rules within the expert system, and those rules can be checked by other experts to ensure their validity.
  - ❑ Human knowledge may be lost
    - ❑ humans tend to die eventually
    - ❑ Human knowledge may be lost in other ways such as brain disease or simply changing jobs
  - ❑ Human knowledge can only be accessed in one place at one time
    - ❑ where the expert happens to be

Strengths and Limitations of Expert Systems

- Limitations
  - Lack of common sense
    - humans may draw conclusions based on their overall view of the world; expert systems do not have this information
  - Lack of inspiration or intuition
  - Lack of flexibility to apply their knowledge outside a relevant domain
    - Humans understand the limits of their knowledge and will seek help when confronted by complex or novel situations
    - Unless programmed specifically, expert systems will not recognise their limitations and fail when confronted with new situations

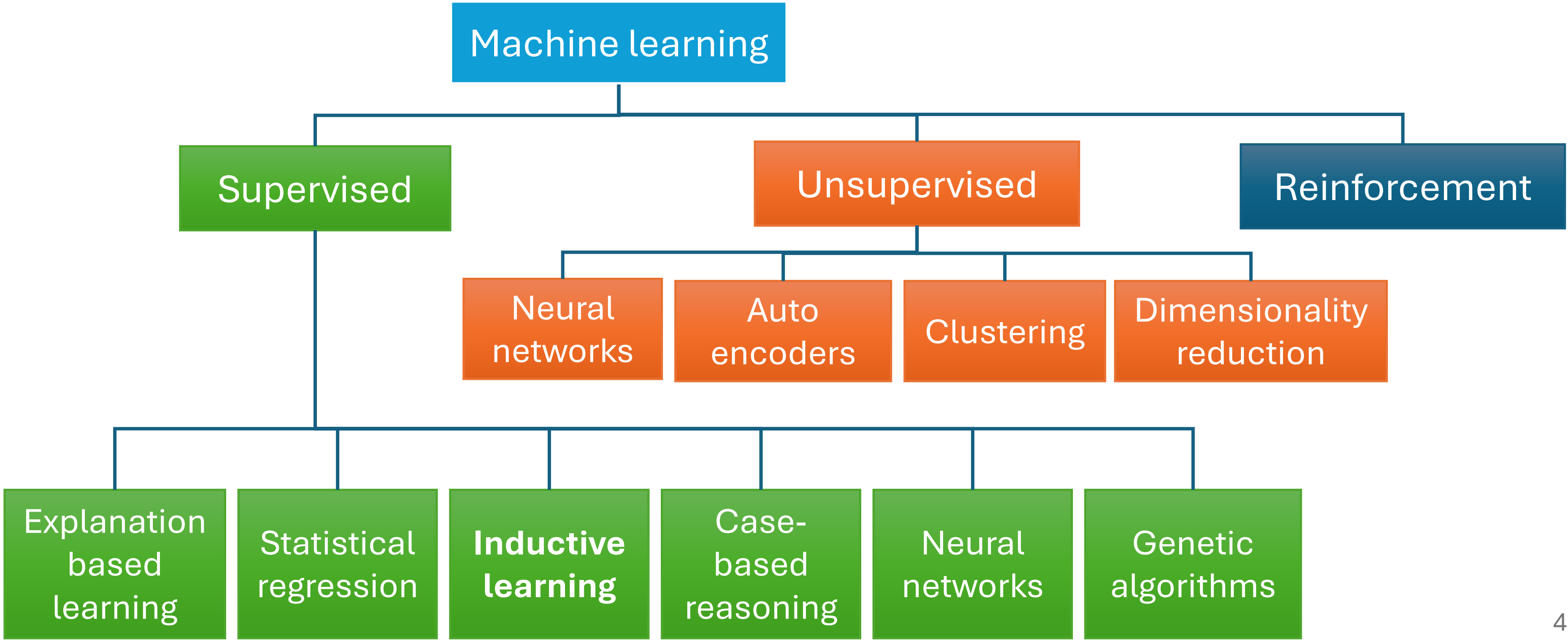
Where Are Expert Systems Used?

- ❑ The problem is important to business
  - ❑ time or money or both can be saved by using the expert system
- ❑ The expertise required is available and stable
  - ❑ Able to build the expert system rule base
- ❑ The knowledge required is scarce
  - ❑ at least in terms of human experts available to provide answers
- ❑ The problem is recurrent
- ❑ The problem is at the right level of difficulty
- ❑ The domain is well defined and of a manageable size
  - ❑ Particularly large domains or domains with no easily defined limits (e.g., object identification systems in the autonomous cars) are difficult to program due to the large number of required rules
- ❑ The solution depends on logical reasoning, not 'common sense' or general knowledge

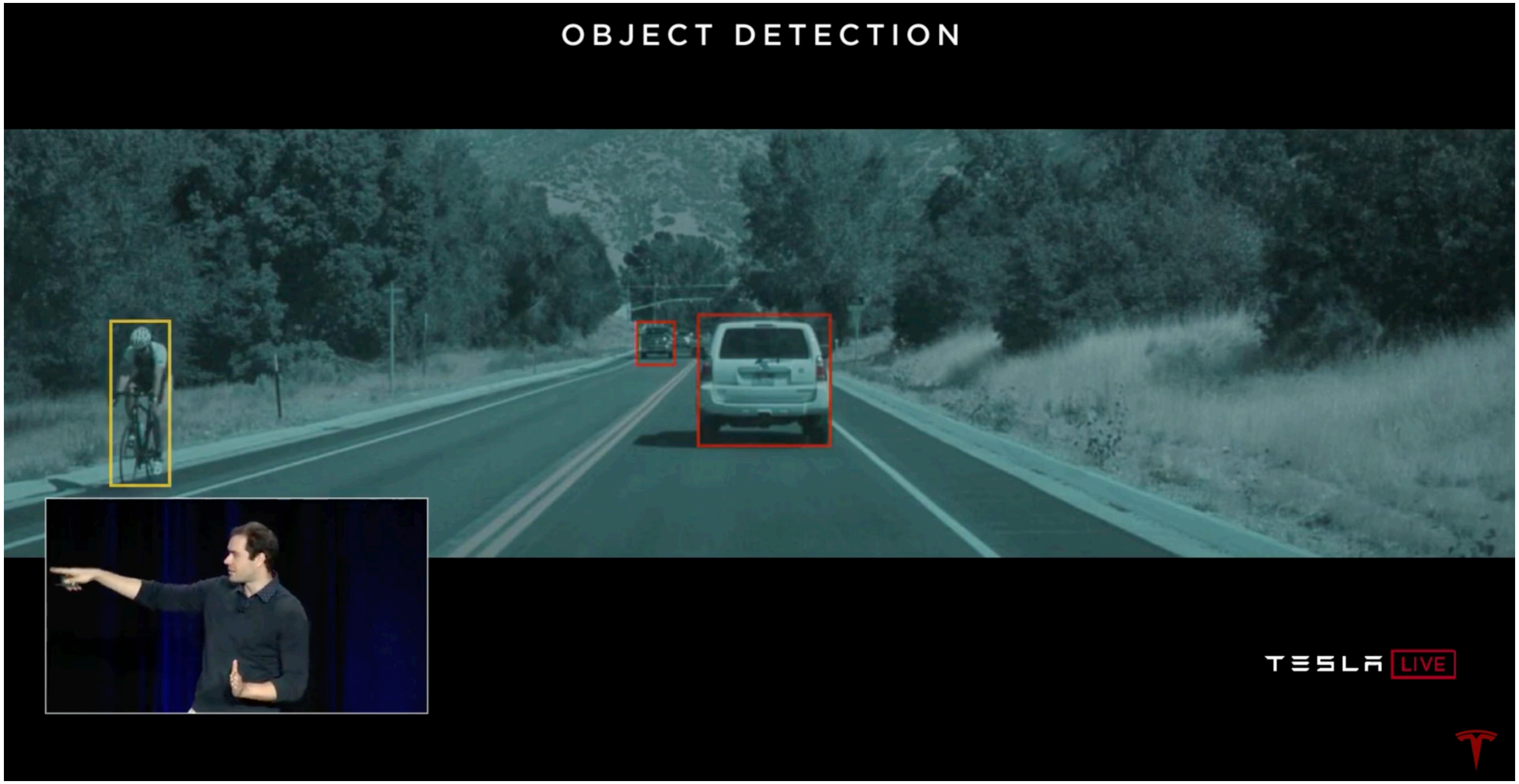
Machine Learning Methods

- **Machine learning**

  The process by which a computer learns from experience (e.g., using programs that can learn from historical cases)



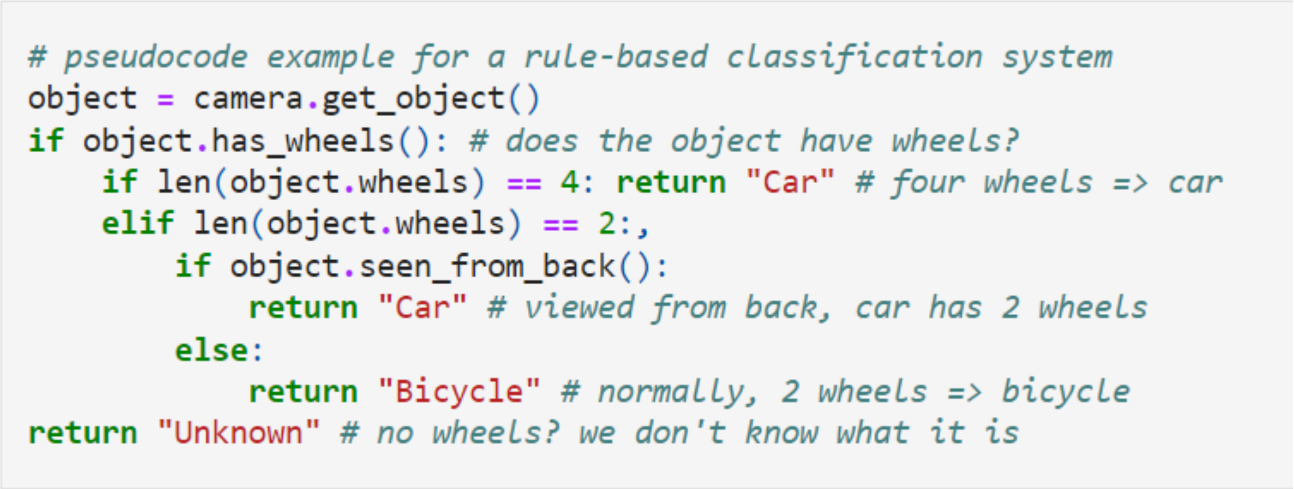
An examples of Machine Learning



A self-driving car system uses dozens of components that include detection of cars, pedestrians, and other objects

An examples of Machine Learning

One way to build a detection system is to write down rules.

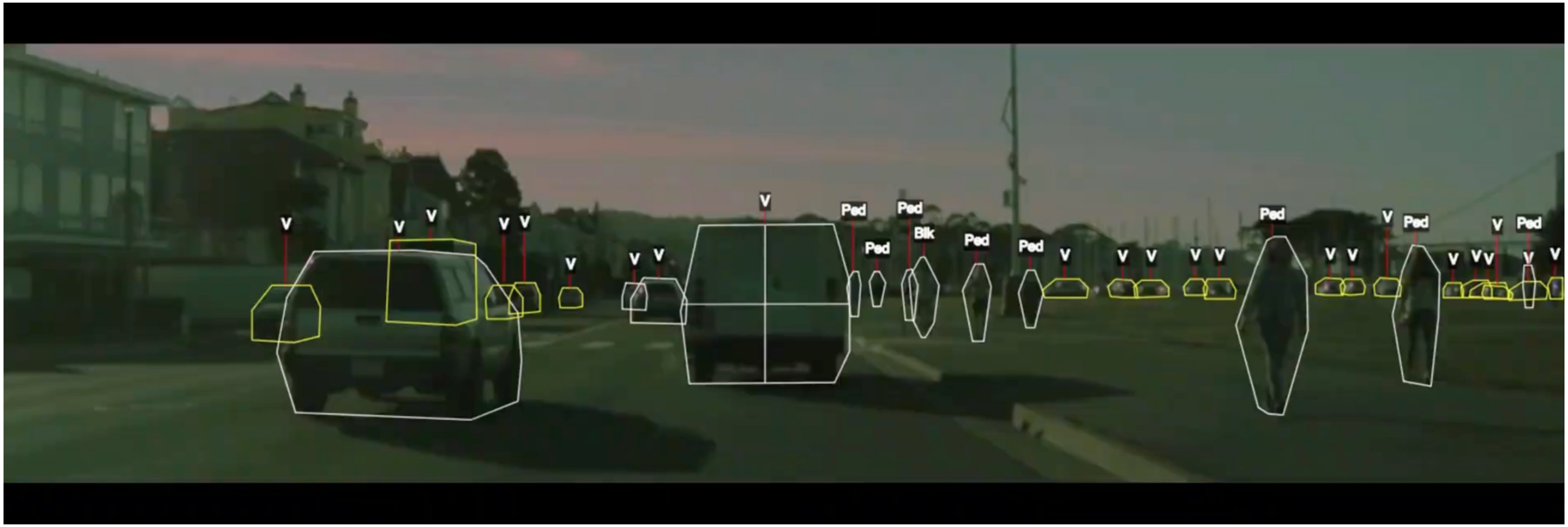


```
# pseudocode example for a rule-based classification system
object = camera.get_object()
if object.has_wheels(): # does the object have wheels?
    if len(object.wheels) == 4: return "Car" # four wheels => car
    elif len(object.wheels) == 2:,
        if object.seen_from_back():
            return "Car" # viewed from back, car has 2 wheels
        else:
            return "Bicycle" # normally, 2 wheels => bicycle
return "Unknown" # no wheels? we don't know what it is
```

In practice, it's almost impossible for a human to specify all the edge cases.

An examples of Machine Learning

The machine learning approach is to teach a computer how to do detection by showing it many examples of different objects.



No manual programming is needed: the computer learns what defines a pedestrian or a car on its own!

A formal definition of Machine Learning

Machine learning is a field of study that gives computers the ability to learn without being explicitly programmed.
(Arthur Samuel, 1959)

This principle can be applied to countless domains: medical diagnosis, factory automation, machine translation, and many more!

Neural Networks

- ❑ Neural networks represent a brain metaphor for information processing. Neural computing refers to a pattern recognition methodology for machine learning. The resulting model from neural computing is often called an artificial neural network (ANN) or neural network (NN).

- ❑ Due to their ability to learn from the data, their non-parametric nature (i.e., no rigid assumptions), and their ability to generalize, neural networks have been shown to be promising in many forecasting and business classification applications.

Basic Concepts of Neural Networks

- The human brain is composed of special cells called nuerons.
- Neural network elements
  - **Nucleus**

    The central processing portion of a neuron
  - **Soma**

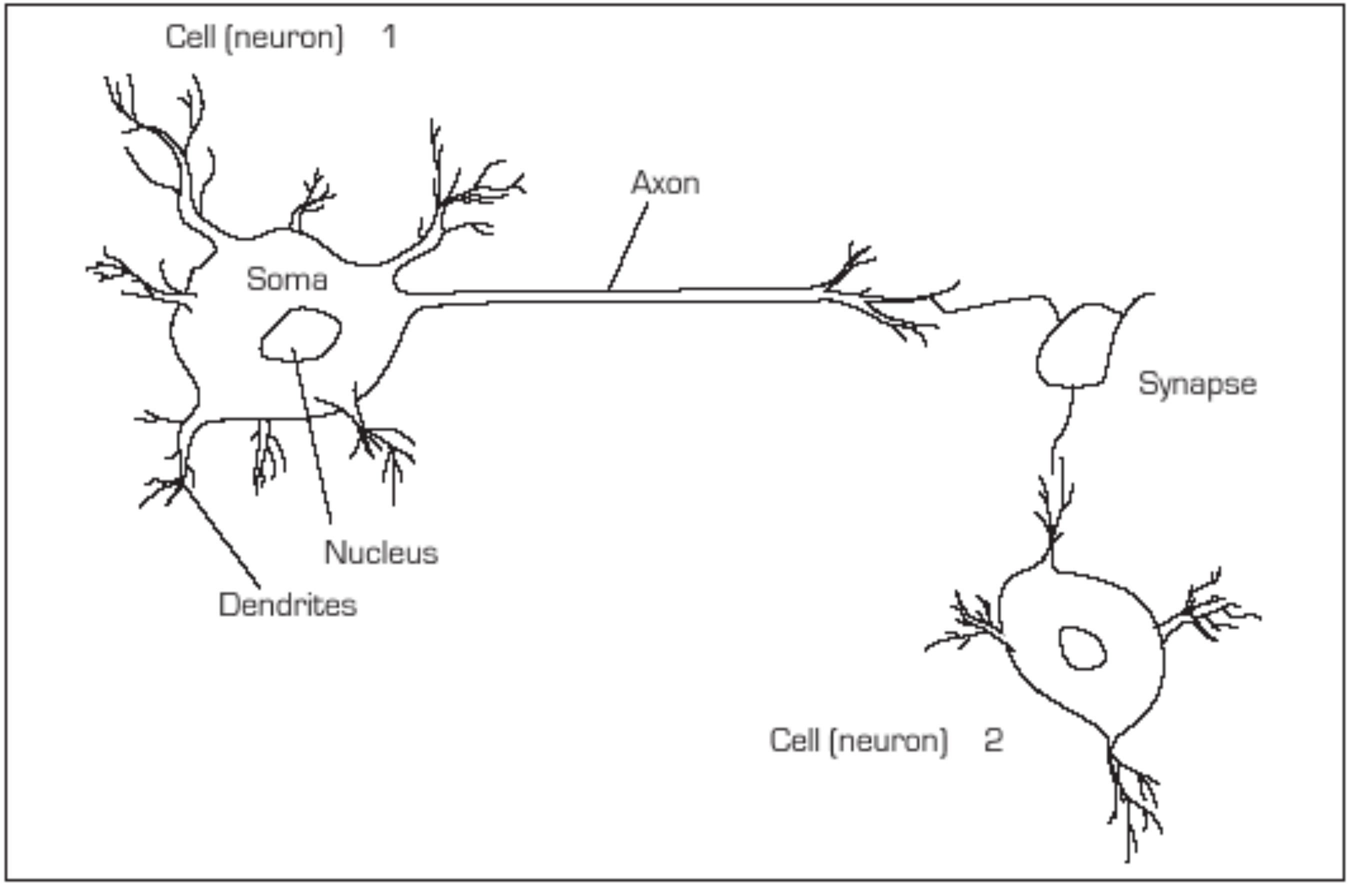
    The main body of the neuron in which the cell nucleus is contained
  - **Dendrite (sợi nhánh)**

    The part of a biological neuron that provides inputs to the cell
  - **Axon (sợi trục)**

    An outgoing connection (i.e., terminal) from a biological neuron
  - **Synapse**

    The connection (where the weights are) between processing elements in a neural network

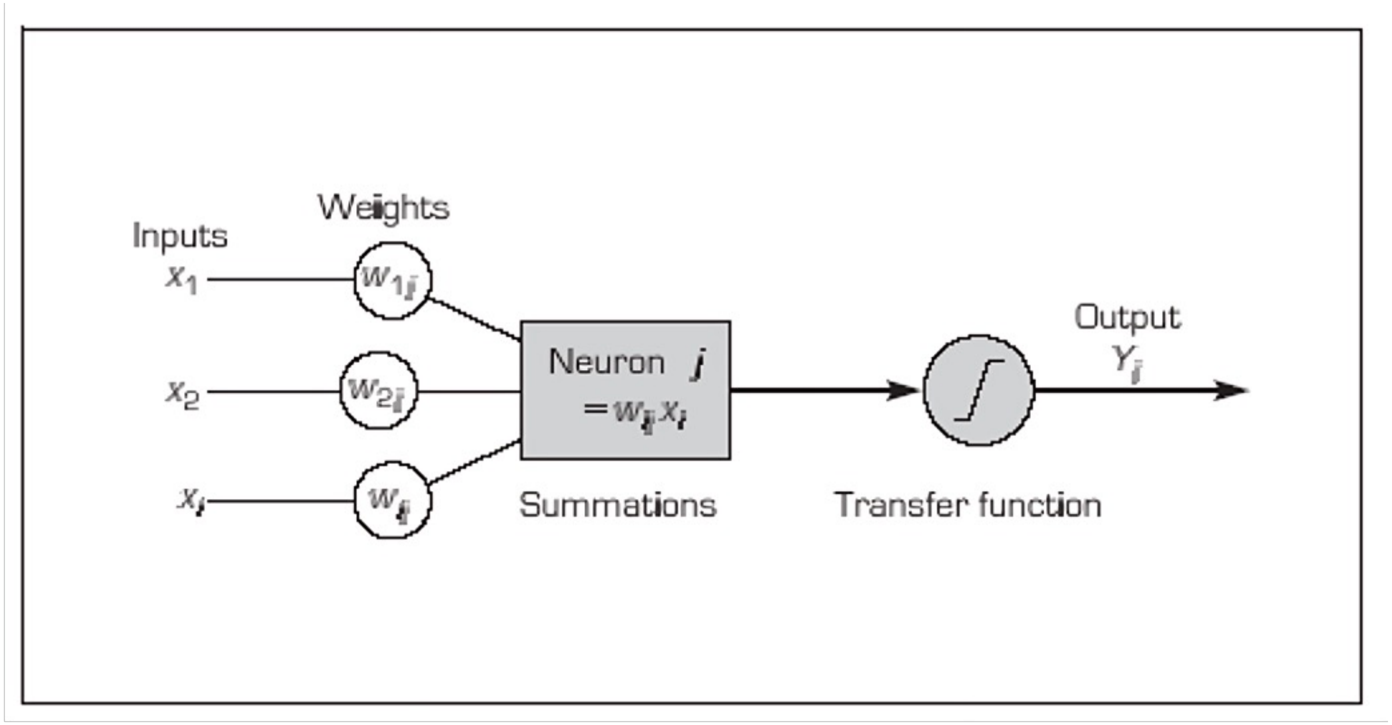
Structure of a Biological Neural Network



Artificial Neural Network

- An ANN model emulates a biological neural network
- Neural concepts are usually implemented as software simulations of the massive parallel processes that involve processing elements (also called artificial neurons) interconnected in a network structure.
- Connections between neurons have an associated weight.
- Each neuron calculates a weighted sum of the incoming neuron values, transforms this input, and passes on its neural value as the input to subsequent neurons or external outputs.

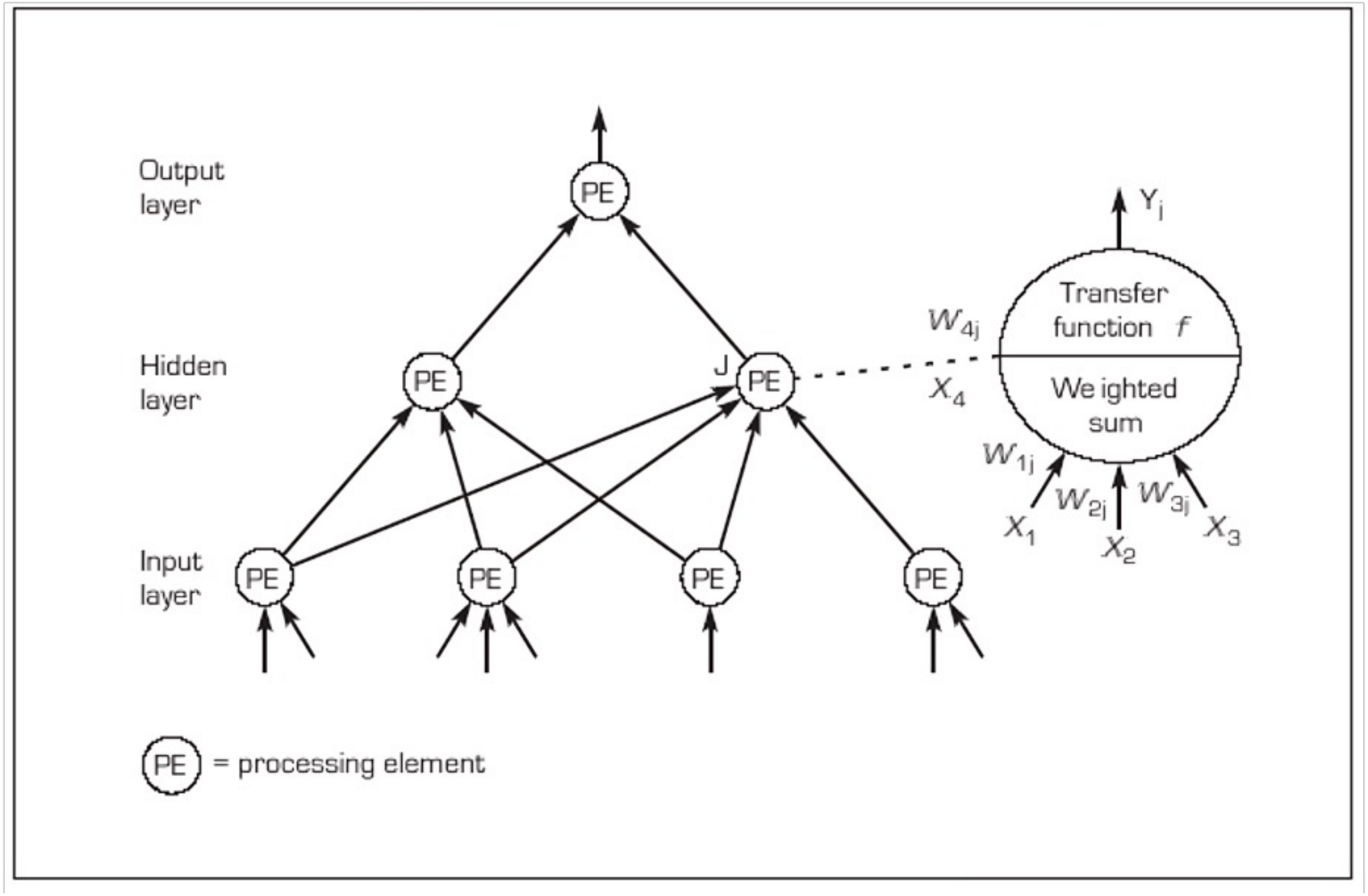
Processing Information in an Artificial Neuron



The Relationship Between Biological and Artificial Neural Networks

Biological	Artificial
Soma	Node
Dendrites	Input
Axon	Output
Synapse	Weight
Slow speed	Fast speed
Many neurons (10^9)	Few neurons (a dozen to hundreds of thousands)

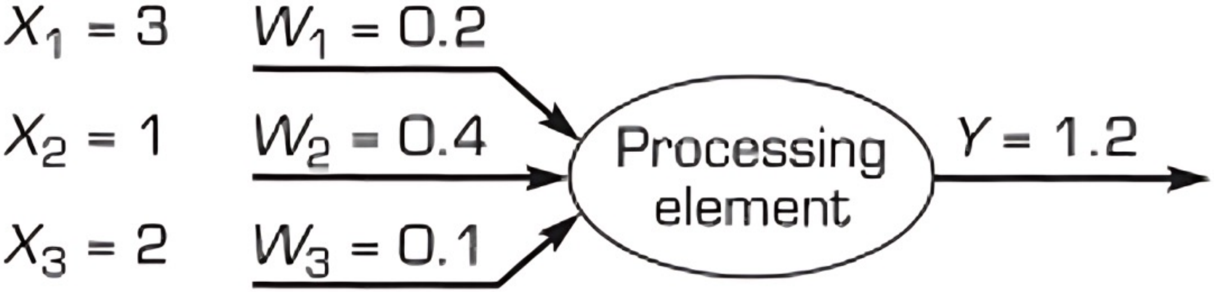
Neural Network with One Hidden Layer



Example of ANN Functions

Summation function: $Y = 3\,(0.2) + 1(0.4) + 2(0.1) = 1.2$

Transformation (transfer) function: $Y_T = 1/(1 + e^{-1.2}) = 0.77$



$X_1 = 3$ $W_1 = 0.2$

$X_2 = 1$ $W_2 = 0.4$

$X_3 = 2$ $W_3 = 0.1$

Processing element

$Y = 1.2$

Learning in ANN

- **Supervised learning**
  - Uses a set of inputs for which the desired outputs are known
  - *Example*: Backpropagation algorithm

- **Unsupervised learning**
  - Uses a set of inputs for which **NO** desired output are known
  - The system is self-organizing; that is, it organizes itself internally. A human must examine the final categories to assign meaning and determine the usefulness of the results.
  - *Example*: Self-organizing map

Learning in ANN

- ❑ **Supervised learning**
  - ❑ Uses a set of inputs for which the desired outputs are known
  - ❑ *Example*: Backpropagation algorithm

- ❑ **Unsupervised learning**
  - ❑ Uses a set of inputs for which NO desired outputs are known.
  - ❑ The system is self-organizing; that is, it organizes itself internally. A human must examine the final categories to assign meaning and determine the usefulness of the results
  - ❑ *Example*: Self-organizing map.

Characteristics of ANNs

- Adaptive learning
- Self-organization
- Error tolerance
- Real-time operation
- Parallel information processing

Benefits and Limitations of Neural Networks

- Benefits
  - Ability to tackle new kinds of problems
  - Robustness
- Limitations
  - Performs less well at tasks humans tend to find difficult
  - Lack of explanation facilities
  - Requires large amounts of training and test data

Case-Based Reasoning (CBR)

- ❑ A case has two parts: a **problem** and a **solution**
- ❑ Cases represent **experience**; that is, they record how a problem was solved in the past
- ❑ CBR is a methodology in which knowledge and/or inferences are derived from historical cases. It is based on the premise that **new problems are often similar to previously encountered problems** and that, past solutions may be of use in the current situations.
- ❑ CBR is particularly applicable to problems in which the domain is **NOT understood well** enough for a robust statistical model or system of equations to be formulated.

Process of CBR

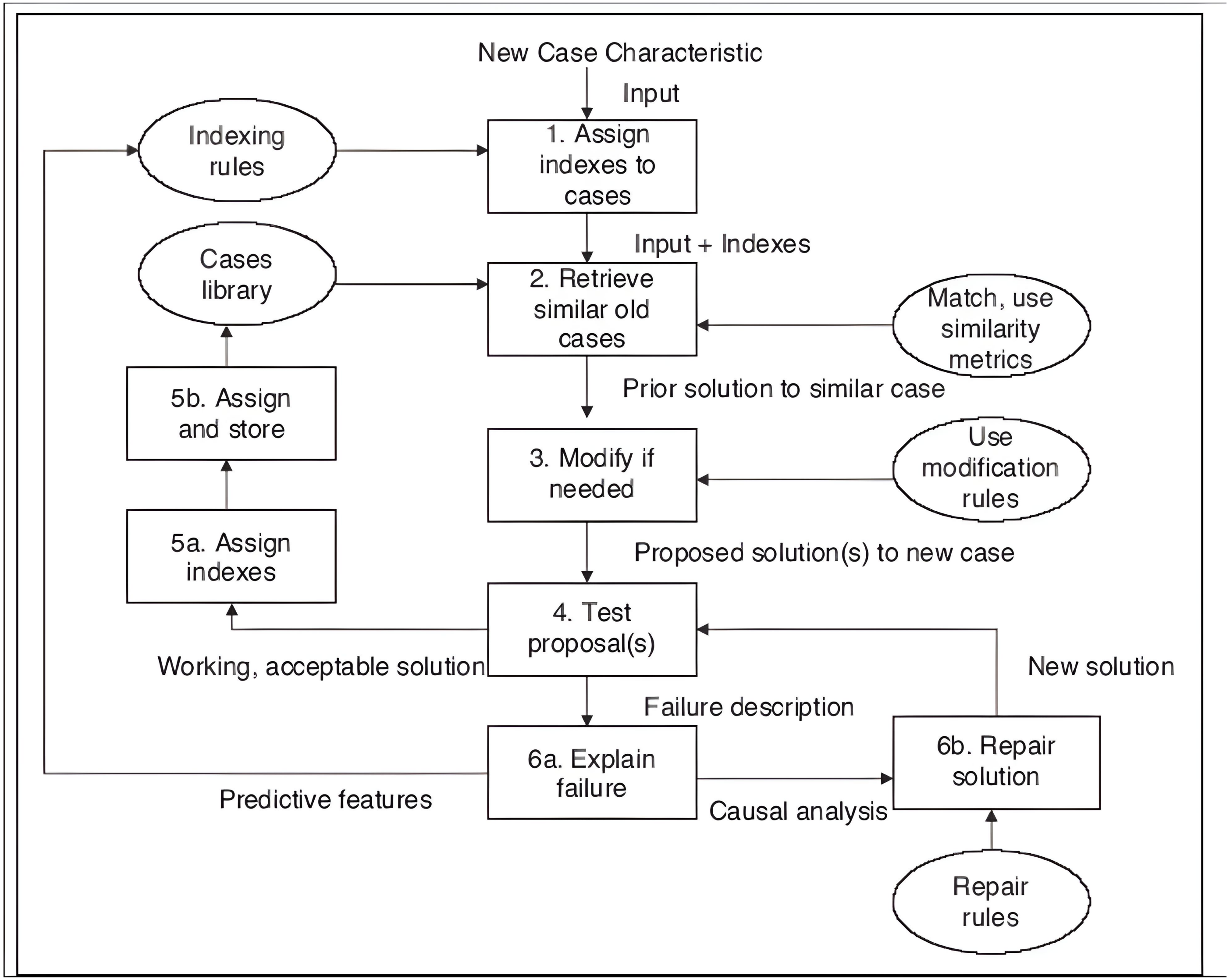
1. **Retrieve**
   - Given a target problem, retrieve the most similar cases

2. **Reuse**
   - Map the solution and reuse the best old solution to solve the current case

3. **Revise**
   - Test the solution and, if necessary, revise the old case to come up with the solution

4. **Retain**
   - After the solution has been successfully adapted to the target problem, store the resulting experience as a new case

Step-by-Step Process of CBR



Similarity Computation

- ❑ Cases are ranked according to their similarity based on the similarity of each feature
- ❑ The degree of similarity can be expressed by a **real number** between **0** (not similar) and **1** (identical).
- ❑ The importance of different features may be different. In that case, similarity is computed by weighted average.

CBR Examples

- Intelligent customer support and sales support
- Retrieval of tour packages from travel catalogs
- Conflict resolution in air traffic control
- Conceptual building design aid
- Conceptual design aid for electronic devices
- Medical diagnosis
- Aircraft troubleshooting
- Heuristic retrieval of legal knowledge
- Computer supported conflict resolution through negotiation or mediation

Advantages and Disadvantages of Using CBR

- **Advantages**
  - Improved knowledge acquisition
  - Reduced development time
  - Easier explanation
  - Learning over time

- **Disadvantages**
  - Storing of cases in the Knowledge base
  - Implicit link between problem and solution
  - Access and retrieval speed.

Genetic Algorithms

- Programs that attempt to find optimal solutions to problems by conceptually following steps inspired by the biological processes of evolution
- The method learns by producing offspring that are better and better, as measured by a fitness-to-sur vive function, until an optimal or near-optimal solution is obtained.

Genetic Algorithm Fundamentals

- **Chromosome**

  A candidate solution for a genetic algorithm

- **Fitness function**

  A measure of the objective to be obtained.

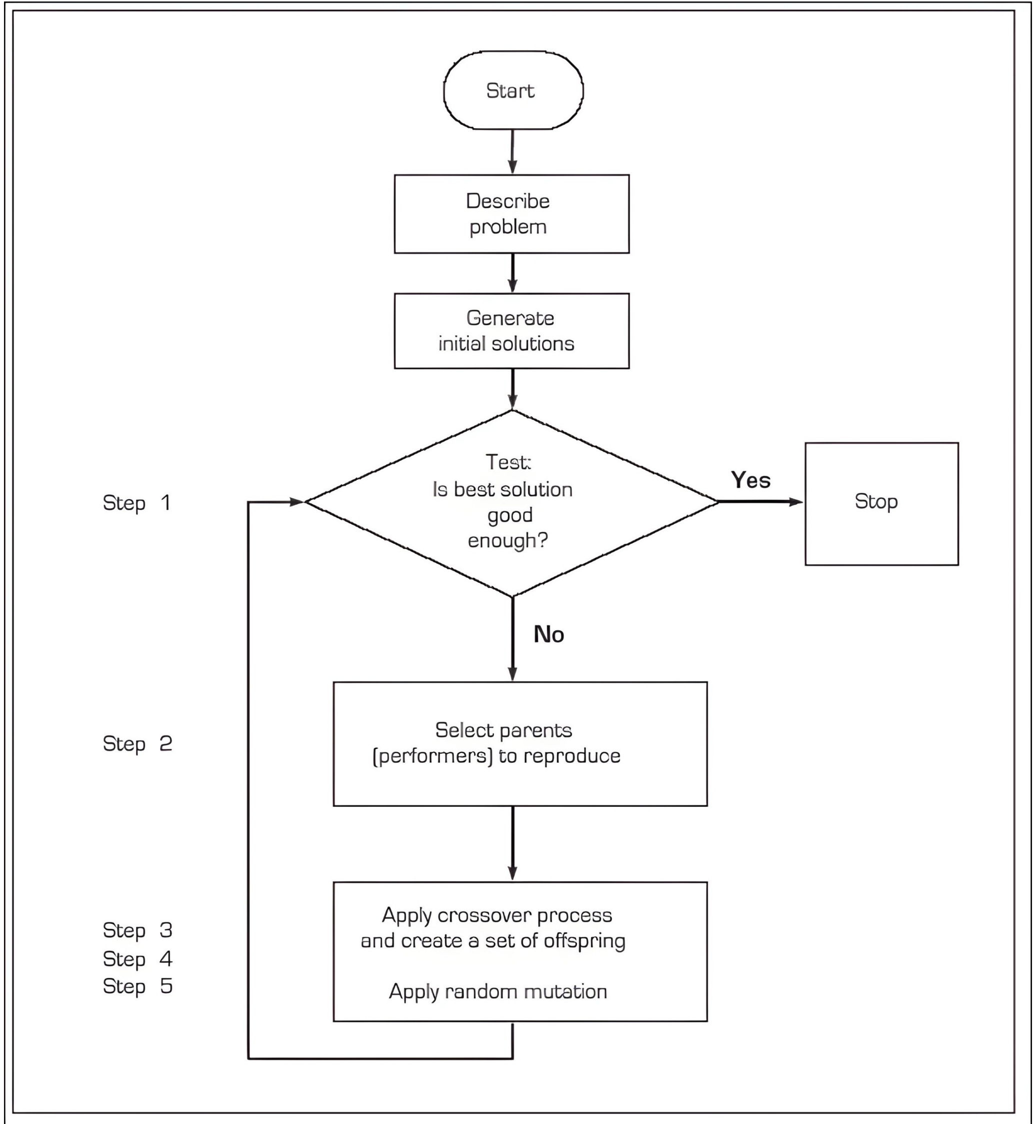
- **Generation**

  An iteration of the genetic algorithmic process in which candidate solutions are combined to produce offspring

Processes within Genetic Algorithm

- **Reproduction**
  - Through reproduction, genetic algorithms produce new generations of improved solutions by selecting parents with higher fitness ratings or by giving such parents a greater probability of being contributors and by using random selection.
- **Crossover**
  - The combining of parts of two superior solutions by a genetic algorithm in an attempt to produce an even better solution
- **Mutation**
  - A genetic operator that causes a random change in a potential solution

Genetic Algorithm Process


Start
Describe problem
Generate initial solutions
Step 1
Test: Is best solution good enough?
Yes
Stop
No
Step 2
Select parents (performers) to reproduce
Step 3
Step 4
Step 5
Apply crossover process and create a set of offspring
Apply random mutation

Genetic Algorithm Parameters

- Some parameters must be for the genetic algorithm
  - Number of initial solutions to generate
  - Number of offspring to generate
  - Number of parents and offspring to keep for the next generation
  - Mutation probability
  - Probability distribution of crossover point occurrence
- Their values are dependent on the problem being solved and are usually determined through trial and error

Genetic Algorithm Benefits and Limitations

- Genetic algorithms are particularly useful for complex problems that require rapid development of set of good solutions

- **Limitations**
  - Not all problems can be framed in the mathematical manner that genetic algorithms demand.
  - Development of a genetic algorithm is complex.
  - In some situations, the "genes" from a few comparatively highly fit (but not optimal) individuals may come to dominate the population, causing it to converge on a local maximum
  - Most genetic algorithms rely on random number generators that produce different results each time the model runs

Genetic Algorithm Applications

- Genetic algorithms provide a set of efficient, domain-independent search heuristics for a broad spectrum of applications including
  - Dynamic process control
  - Complex design of engineering structures
  - Scheduling
  - Transportation and routing
  - Layout and circuit design
  - Telecommunications
  - Discovery of new connectivity typologies

Intelligent Agents

- ❑ A computer program that carries out a set of operations on behalf of a user or another program, with some degree of autonomy, and in doing so, employs some knowledge or representation of the user's goals or desires.
- ❑ Agents in various forms
  - ❑ Software agents, wizards, software daemons
  - ❑ e-mail agents (mailbots), web browsing assisting agents
  - ❑ intelligent search agents (Web robots, spiders)
  - ❑ Internet softbots, network management and monitoring agents
  - ❑ e-commerce agents

Features of Intelligent Agents

- ❑ **Reactivity**
  - ❑ Agents perceive their environment and respond in a timely fashion to changes that occur in it.
- ❑ **Proactiveness**
  - ❑ Agents are able to exhibit goal-directed behavior by taking initiative
- ❑ **Social ability**
  - ❑ Agents are capable of interacting with other agents in order to satisfy their design objectives.
- ❑ **Autonomy**
  - ❑ Agents must have control over their own actions and be able to work and launch actions independently of the user or other actors

Why Use Intelligent Agents

- The Gartner Group findings on information overload:
  - The amount of data collected by large enterprises doubles every year
  - Knowledge workers can analyze only about 5% of this data
  - Most of the knowledge workers' efforts are spent in trying to discover important patterns in the data (60% or more), a much smaller percentage in determining what these patters mean (20% or less), and very little time (10% or less) is spend actually doing something about the patterns.
  - Information overload reduces our decision-making capabilities by 50 percent.
- A major value of intelligent agents is that they are able to assist in searching through all the data.
- Intelligent agents save time by making decisions about what is relevant to the user as well as by automating routine tasks.

Intelligent Agents: How Smart Are They?

- Intelligence levels
  - Level 0 - Agents retrieve documents for a user under straight orders
  - Level 1 - Agents provide a user-initiated searching facility for finding relevant Web pages
  - Level 2 - Agents maintain users' profiles
  - Level 3 - Agents have a learning and deductive component to help a user who cannot formalize a query or specify a target for a search

Intelligent Agents Vs. Expert Systems

- Agents and expert systems are **similar** in that they both intend to **incorporate domain knowledge to automate decision making.**
- They are different in the following aspects:
  - Classic ESs are not coupled to any environment in which they act; they act through a user as a middle man. Agents can actively search information from the environment in which they reside.
  - ESs are not generally capable of reactive and proactive behavior.
  - ESs are not generally equipped with social ability in the sense of cooperation, coordination, and negotiation.

Internet-based Software Agents

- Nine major application areas:
  - Assisting in workflow and administrative management
  - Collaborating with other agents and people
  - Supporting e-commerce
  - Supporting desktop applications
  - Assisting in information access and management, including searching and FAQs
  - Processing e-mail and messages
  - Controlling and managing network access
  - Managing systems and networks
  - Creating user interfaces, including navigation (browsing)

Issues to Consider for Intelligent Agents

- Learning
- Performance
- Multiagents
- Cost justification
- Security and privacy
- Ethical issues
- Acceptance