MODULE 5

I. MODEL BASED DECISION MAKING

9.2 DECISION SUPPORT SYSTEMS MODELING

Applications of Models in DSS

Many readily accessible applications describe how the models incorporated in Decision Support Systems (DSS) contribute to organizational success. Examples include Pillowtex (ProModel, 2013), Fiat (ProModel, 2006), and Procter & Gamble (Camm et al., 1997). INFORMS publications such as *Interfaces*, *ORMS Today*, and *Analytics magazine* include stories that illustrate successful applications of decision models in real settings. Simulation models, for example, can enhance an organization's decision-making process and enable it to see the impact of its future choices. Fiat (ProModel, 2006) saves $1 million annually in manufacturing costs through simulation. IBM has used a combination of weather and sensor data to build a river system simulation application capable of simulating thousands of river branches at a time. This application not only predicts the imminent flood of the Guadalupe River days in advance but also supports irrigation planning to avoid the effects of droughts and surplus water. Pillowtex, a $2 billion company that manufactures pillows, mattress pads, and comforters, employed a simulation model to reorganize its plants and maximize net profits after filing for bankruptcy, resulting in immediate savings of over $12 million. Christiansen et al. (2009) describe how TurboRouter, a DSS for ship routing and scheduling, helped a shipping company better utilize its fleet and generate $1–2 million in profit in just three weeks.

CURRENT MODELING ISSUES

(1) IDENTIFICATION OF THE PROBLEM AND ENVIRONMENTAL ANALYSIS

Environmental scanning and analysis, which involve monitoring, scanning, and interpreting collected information, are crucial to understanding the problem context. No decision is made in a vacuum; it is essential to analyze the scope of the domain and the forces and dynamics of the environment. Decision-makers need to identify organizational culture and corporate decision-making processes, such as who makes decisions and the degree of centralization. Environmental factors may have contributed to the current problem, and business analytics (BA) tools can help identify problems by scanning for them. A shared understanding of the problem among all stakeholders ensures the model accurately represents the issue and aids decision-making effectively.

(2) VARIABLE IDENTIFICATION

Identifying the variables within a model—decision variables, result variables, and uncontrollable variables—is critical to the modeling process, as are the relationships among these variables. Influence diagrams, which are graphical representations of mathematical models, can simplify the identification process, while cognitive maps, a broader form of influence diagrams, help decision-makers understand the problem more comprehensively, especially concerning variables and their interactions.

(3) FORECASTING (PREDICTIVE ANALYTICS)

Forecasting involves predicting the future and is an essential aspect of predictive analytics. This process is critical for constructing and manipulating models since decisions implemented today affect future outcomes. While traditional MIS focus on reporting what is or was, DSS enable forecasting what will be, allowing for what-if analyses and sensitivity analyses. E-commerce has created immense opportunities for forecasting, providing vast amounts of information to predict demand and analyze product life-cycle needs, marketplace trends, and customer behavior. Forecasting models can optimize customer relationship management (CRM) and revenue management systems (RMS), which predict customer profitability and identify the right products, prices, and customers to target.

MODEL CATEGORIES

Table 9.1 categorizes DSS models into seven groups, each with representative techniques. These models may be static or dynamic and are developed under environments of certainty, uncertainty, or risk. Special decision analysis systems, such as spreadsheets, OLAP systems, and data mining tools, embed modeling languages and capabilities that facilitate model construction and expedite decision-making processes.

TABLE 9.1 Categories of Models

Category	Process and Objective	Representative Techniques
Optimization of problems with few alternatives	Find the best solution from a small number of alternatives	Decision tables, decision trees, analytic hierarchy process
Optimization via algorithm	Find the best solution from a large number of alternatives, using a step-by-step improvement process	Linear and other mathematical programming models, network models
Optimization via an analytic formula	Find the best solution in one step, using a formula	Some inventory models
Simulation	Find a good enough solution or the best among the alternatives checked, using experimentation	Several types of simulation
Heuristics	Find a good enough solution, using rules	Heuristic programming, expert systems
Predictive models	Predict the future for a given scenario	Forecasting models, Markov analysis
Other models	Solve a what-if case, using a formula	Financial modeling, waiting lines

MODEL MANAGEMENT

Models, like data, require systematic management to maintain integrity and applicability. Model Base Management Systems (MBMS) are analogous to Database Management Systems (DBMS) and play a critical role in managing and maintaining models effectively.

KNOWLEDGE-BASED MODELING

While DSS predominantly rely on quantitative models, expert systems utilize qualitative, knowledge-based models. Constructing solvable models requires domain knowledge and expertise, which can be integrated into predictive analytics techniques like classification and clustering. These techniques help build robust knowledge-based models by incorporating expertise and insights.

CURRENT TRENDS IN MODELING

Recent trends in modeling include the development of model libraries and solution technique libraries. These libraries allow decision-makers to access and apply powerful optimization and simulation packages via web-based platforms, such as the NEOS Server for Optimization and INFORMS resources. While these tools simplify modeling, effective use requires a solid understanding of model construction and analysis. Revenue management has expanded beyond airlines and hotels to retail, insurance, and entertainment, while CRM continues to leverage predictive models to optimize decision-making. However, challenges such as limited expertise and large data requirements persist, necessitating the use of data warehouses and parallel computing for effective modeling. The trend toward transparent analytics models, such as those seen in OLAP systems, has made modeling more accessible but has also limited consideration of advanced and nuanced model classes.

9.3 STRUCTURE OF MATHEMATICAL MODELS FOR DECISION SUPPORT

The Components of Decision Support Mathematical Models (IMPORTANT)

All quantitative models typically comprise four basic components: result (or outcome) variables, decision variables, uncontrollable variables (and/or parameters), and intermediate result variables. Figure 9.1 illustrates these components. Mathematical relationships link these components in quantitative models, while in non-quantitative models, the relationships are symbolic or qualitative. The results of decisions are determined by the values of decision variables, the uncontrollable factors in the environment, and the relationships among these variables. The modeling process involves identifying these variables and their interconnections. Solving a model helps determine the values of the components and the result variable(s).

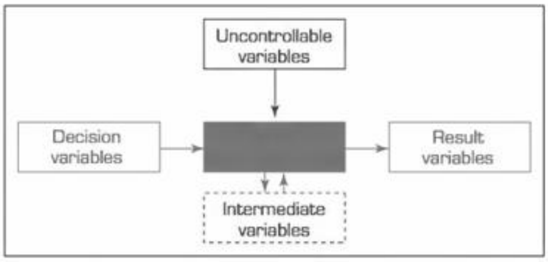

FIGURE 9.1 The General Structure of a Quantitative Model.

1. **RESULT (OUTCOME) VARIABLES**

- Result (outcome) variables reflect the level of effectiveness of a system and indicate how well the system performs or attains its goal(s). These variables serve as outputs.
- Examples of result variables are shown in Table 9.2. They are considered dependent variables, as their values depend on other factors in the model.
- Intermediate result variables may also be used in models to capture intermediate outcomes. For dependent variables, the occurrence of another event is a prerequisite before the event described by the variable can take place.
- In essence, result variables rely on the occurrence of decision variables and uncontrollable variables.

2. **DECISION VARIABLES**

- Decision variables describe the various alternative courses of action available to a decision maker. These variables are under the control of the decision maker.
- For instance, in an investment scenario, the amount to invest in bonds is a decision variable. Similarly, in a scheduling problem, decision variables include the allocation of people, times, and schedules.
- Table 9.2 provides additional examples of decision variables that can arise in various contexts.

3. **UNCONTROLLABLE VARIABLES, OR PARAMETERS**

- In any decision-making scenario, there are factors that influence result variables but are beyond the control of the decision maker. These factors may either be fixed, in which case they are termed uncontrollable variables or parameters, or they may vary, in which case they are referred to as variables.

- Examples of such factors include the prime interest rate, tax regulations, a city's building code, and utility costs. These factors are generally uncontrollable because they are determined by elements of the system environment where the decision maker operates. Some of these variables also act as constraints, thereby limiting the decision maker's available choices.

4. INTERMEDIATE RESULT VARIABLES

- Intermediate result variables represent intermediate outcomes in mathematical models.
- For example, in a machine scheduling scenario, spoilage serves as an intermediate result variable, while total profit is the final result variable, with spoilage contributing to the determination of total profit.
- Another example is employee salaries, which act as a decision variable for management. Salaries influence employee satisfaction, which constitutes an intermediate outcome, and this satisfaction level, in turn, affects productivity, which is the final result variable.

TABLE 9.2 Examples of the Components of Models

Area	Decision Variables	Result Variables	Uncontrollable Variables and Parameters
Financial investment	Investment alternatives and amounts	Total profit, risk Rate of return on investment (ROI) Earnings per share Liquidity level	Inflation rate Prime rate Competition
Marketing	Advertising budget Where to advertise	Market share Customer satisfaction	Customer's income Competitor's actions
Manufacturing	What and how much to produce Inventory levels Compensation programs	Total cost Quality level Employee satisfaction	Machine capacity Technology Materials prices
Accounting	Use of computers Audit schedule	Data processing cost Error rate	Computer technology Tax rates Legal requirements
Transportation	Shipments schedule Use of smart cards	Total transport cost Payment float time	Delivery distance Regulations
Services	Staffing levels	Customer satisfaction	Demand for services

The Structure of Mathematical Models

The components of a quantitative model are linked together by mathematical (algebraic) expressions—equations or inequalities.

A very simple financial model is

$$P = R - C$$

where P = profit, R = revenue, and C = cost. This equation describes the relationship among the variables. Another well-known financial model is the simple present-value cash flow model, where P = present value, F = a future single payment in dollars, i = interest rate (percentage), and n = number of years. With this model, it is possible to determine the present value of a payment of $100,000 to be made 5 years from today, at a 10 percent (0.1) interest rate, as follows:

$$P = 100{,}000/(1 + 0.1)^5 = 62{,}092$$

We present more interesting and complex mathematical models in the following sections.

9.4 CERTAINTY, UNCERTAINTY, AND RISK (IMPORTANT)

As part of Simon's decision-making process discussed in Chapter 2, evaluating and comparing alternatives necessitates predicting the future outcome of each proposed alternative. Decision situations are typically classified based on what the decision maker knows or believes about the forecasted results. This knowledge is divided into three categories (see Figure 9.2), which range from complete knowledge to complete ignorance:

- Certainty
- Risk
- Uncertainty

When developing models, any of these conditions can occur, and different types of models are appropriate for each case. The following sections define these terms and highlight important modeling issues associated with each condition.

Decision Making Under Certainty

- In decision making under certainty, it is assumed that complete knowledge is available so that the decision maker knows exactly what the outcome of each course of action will be (as in a deterministic environment).
- While outcomes may not be 100% certain, this assumption simplifies the model and makes it more manageable. The decision maker is considered a perfect predictor of the future, as only one outcome is assumed for each alternative.
- For example, investing in U.S. Treasury bills involves complete information about future returns if held to maturity. Decision making under certainty is often associated with structured problems and short time horizons (up to one year). Certainty models are relatively straightforward to develop and solve, often yielding optimal solutions. Many financial models assume certainty, despite the inherent uncertainties of financial markets.

Decision Making Under Uncertainty

- In decision making under uncertainty, multiple outcomes are possible for each course of action, but the probabilities of these outcomes are unknown or cannot be estimated.

This makes decision making under uncertainty more complex than under certainty due to insufficient information.

- Modeling in such scenarios often involves assessing the decision maker's or organization's attitude toward risk.
- Managers typically attempt to minimize uncertainty, sometimes by seeking additional information to shift the problem into a more manageable certainty or risk framework.
- If no further information is available, the problem must be addressed under uncertainty, which is inherently less definitive and more challenging than other categories.

Decision Making Under Risk (Risk Analysis)

- Decision making under risk (also referred to as probabilistic or stochastic decision making) involves considering multiple possible outcomes for each alternative, each associated with a specific probability of occurrence.
- These probabilities, assumed to be known or estimable, allow the decision maker to calculate and assess the degree of risk associated with each alternative—this is known as calculated risk.
- Most major business decisions are made under conditions of assumed risk.
- Risk analysis involves calculating the expected value of each alternative and choosing the one with the best expected value.
- By analyzing known probabilities, decision makers can make more informed decisions under risk.

9.7 MULTIPLE GOALS, SENSITIVITY ANALYSIS, WHAT-IF ANALYSIS, AND GOAL SEEKING (IMPORTANT)

Multiple Goals

- Management decision analysis seeks to evaluate how effectively each alternative advances managers toward their goals.
- Managerial problems often involve multiple, simultaneous goals, some of which may conflict.
- Stakeholders, such as shareholders, managers, and employees, frequently have different and sometimes opposing objectives.
- For example, a profit-making firm may aim to earn revenue, grow, develop its products and employees, provide job security, and serve the community. Simultaneously,

shareholders demand profitability, managers desire higher salaries, and employees want increased benefits. In such scenarios, analytical methods, like the analytic hierarchy process (AHP) combined with integer programming, can evaluate multiple goals.

- To manage multiple goals, decision-making often involves transforming them into a single-measure-of-effectiveness problem, enabling comparison among solutions. Linear programming (LP) models frequently employ this approach.

Certain **difficulties** may arise when analyzing multiple goals:

- It is usually difficult to obtain an explicit statement of the organization's goals.
- The decision maker may change the importance assigned to specific goals over time or for different decision scenarios.
- Goals and sub-goals are viewed differently at various levels of the organization and within different departments.
- Goals change in response to changes in the organization and its environment.
- The relationship between alternatives and their role in determining goals may be difficult to quantify.
- Complex problems are solved by groups of decision makers, each of whom has a personal agenda.
- Participants assess the importance (priorities) of the various goals differently.

Several **methods** of handling multiple goals can be used when working with MSS. The most common ones are:

- Utility theory
- Goal programming
- Expression of goals as constraints, using LP
- A points system

Sensitivity Analysis

- Sensitivity analysis evaluates the impact of changes in input data or parameters on the solution of a model.

- It is essential for understanding model flexibility and adaptability to dynamic conditions.
- Sensitivity analysis aids in refining models, increasing decision-making confidence, and providing insights into the relationships between variables.

Sensitivity analysis tests relationships such as the following:

- The impact of changes in external (uncontrollable) variables and parameters on the outcome variable(s)
- The impact of changes in decision variables on the outcome variable(s)
- The effect of uncertainty in estimating external variables
- The effects of different dependent interactions among variables
- The robustness of decisions under changing conditions

Sensitivity analyses are used for:

- Revising models to eliminate too-large sensitivities
- Adding details about sensitive variables or scenarios
- Obtaining better estimates of sensitive external variables
- Altering a real-world system to reduce actual sensitivities
- Accepting and using the sensitive (and hence vulnerable) real world, leading to the continuous and close monitoring of actual results

Types of Sensitivity Analysis:

1. **Automatic Sensitivity Analysis:**
   Standard quantitative models, such as LP, can perform automatic sensitivity analysis, identifying ranges within which input changes do not significantly impact solutions. It is limited to single-variable changes but is efficient and precise. For example, LP sensitivity analysis can calculate profit changes based on marketing constraints.
2. **Trial-and-Error Sensitivity Analysis:**
   This approach involves adjusting variables and re-solving the problem to observe impacts. It allows for experimentation and discovering better solutions, especially when using tools like Excel.

<u>What-If Analysis</u>

What-if analysis explores the effects of changing input variables, assumptions, or parameters on solutions. Example scenarios include:

- How will a 10% increase in inventory carrying costs affect total inventory cost?
- What market share will result from a 5% increase in the advertising budget?

Using tools like Excel, managers can modify data inputs directly and receive instant feedback, enabling iterative exploration of scenarios without needing technical expertise.

<u>Goal Seeking</u>

Goal seeking calculates the necessary input values to achieve a desired output level. It represents a backward problem-solving approach. Examples include:

- Determining the R&D budget required to achieve a 15% annual growth rate by 2018.
- Calculating the number of nurses needed to reduce emergency room waiting times below 10 minutes.

Goal seeking applications, such as calculating the internal rate of return (IRR) in Excel, identify inputs required to achieve specified outcomes. For example, finding the IRR where net present value (NPV) equals zero demonstrates how goal seeking operates.

Break-Even Analysis with Goal Seeking:
Some software can compute break-even points, such as the quantity of goods required to achieve zero profit. These tools are particularly valuable in decision support systems (DSS) for enabling easy sensitivity and goal-seeking analyses.

II. EXPERT SYSTEM

11.4 BASIC CONCEPTS OF EXPERT SYSTEMS (IMPORTANT)

- Expert systems (ES) are **computer-based information systems** that use **expert knowledge** to attain high-level decision performance in a narrowly defined problem domain.
- (MYCIN, developed at Stanford University in the early 1980s for medical diagnosis, is the most well-known ES application).

- ES has also been used in taxation, credit analysis, equipment maintenance, help desk automation, environmental monitoring, and fault diagnosis.
- ES have been popular in large and medium-sized organizations as a sophisticated tool for improving productivity and quality.
- The basic concepts of ES include how to determine who experts are, the definition of expertise, how expertise can be extracted and transferred from a person to a computer, and how the expert system should mimic the reasoning process of human experts.

Experts

- An expert is a person who has the special knowledge, judgment, experience, and skills to put his or her knowledge in action to provide sound advice and to solve complex problems in a narrowly defined area.
- An expert knows which facts are important and also understands and explains the dependency relationships among those facts.
- In diagnosing a problem with an automobile's electrical system, for example, an expert mechanic knows that a broken fan belt can be the cause for the battery to discharge.
- Typically, human experts are capable of doing the following:
    - Recognizing and formulating a problem
    - Solving a problem quickly and correctly
    - Explaining a solution
    - Learning from experience
    - Restructuring knowledge
    - Breaking rules (i.e., going outside the general norms), if necessary
    - Determining relevance and associations
    - Declining gracefully (i.e., being aware of one's limitations)

Expertise

- Expertise is the extensive, task-specific knowledge that experts possess.
- The level of expertise determines the performance of a decision.

- Expertise is often acquired through training, reading, and experience in practice. It includes explicit knowledge, such as theories learned from a textbook or in a classroom, and implicit knowledge, gained from experience.
- The following is a list of possible knowledge types:
    - • Theories about the problem domain
    - • Rules and procedures regarding the general problem domain
    - • Heuristics about what to do in a given problem situation
    - • Global strategies for solving these types of problems • Metaknowledge (i.e., knowledge about knowledge)
    - • Facts about the problem area These types of knowledge enable experts to make better and faster decisions than nonexperts when solving complex problems.
- Expertise often includes the following characteristics:
    - • Expertise is usually associated with a high degree of intelligence, but it is not always associated with the smartest person.
    - • Expertise is usually associated with a vast quantity of knowledge.
    - • Expertise is based on learning from past successes and mistakes.
    - • Expertise is based on knowledge that is well stored, organized, and quickly retrievable from an expert who has excellent recall of patterns from previous experiences.

Features of ES (IMPORTANT)

ES must have the following features:

• **Expertise**. Experts differ in their level of expertise. An ES must possess expertise that enables it to make expert-level decisions. The system must exhibit expert performance with adequate robustness.

• **Symbolic reasoning**. The basic rationale of artificial intelligence is to use symbolic reasoning rather than mathematical calculation. This is also true for ES. That is, knowledge must be represented symbolically, and the primary reasoning mechanism must be symbolic. Typical symbolic reasoning mechanisms include backward chaining and forward chaining, which are described later in this chapter.

• **Deep knowledge**. Deep knowledge concerns the level of expertise in a knowledge base. The knowledge base must contain complex knowledge not easily found among nonexperts.

• **Self-knowledge**. ES must be able to examine their own reasoning and provide proper explanations as to why a particular conclusion was reached. Most experts have very strong learning capabilities to update their knowledge constantly. ES also need to be able to learn from their successes and failures as well as from other knowledge sources.

TABLE 11.1 Comparison of Conventional Systems and Expert Systems

Conventional Systems	Expert Systems
Information and its processing are usually combined in one sequential program.	The knowledge base is clearly separated from the processing (inference) mechanism (i.e.,knowledge rules are separated from the control).
The program does not make mistakes (programmers or users do).	The program may make mistakes.
Conventional systems do not (usually) explain why input data are needed or how conclusions are drawn.	Explanation is a part of most ES.
Conventional systems require all input data. They may not function properly with missing data unless planned for.	ES do not require all initial facts. ES can typically arrive at reasonable conclusions with missing facts.
Changes in the program are tedious (except in DSS).	Changes in the rules are easy to make.
The system operates only when it is completed.	The system can operate with only a few rules (as the first prototype).
Execution is done on a step-by-step (algorithmic) basis.	Execution is done by using heuristics and logic.
Large databases can be effectively manipulated.	Large knowledge bases can be effectively manipulated.
Conventional systems represent and use data.	ES represent and use knowledge.
Efficiency is usually a major goal.	
Effectiveness is important only for DSS.	Effectiveness is the major goal.
Conventional systems easily deal with quantitative data.	ES easily deal with qualitative data.
Conventional systems use numeric data representations.	ES use symbolic and numeric knowledge representations.
Conventional systems capture, magnify, and distribute access to numeric data or information.	ES capture, magnify, and distribute access to judgment and knowledge.

- **TABLE 11.1 IMPORTANT**

- The development of ES is divided into two generations. Most first-generation ES use if-then rules to represent and store their knowledge. The second-generation ES are more flexible in adopting multiple knowledge representation and reasoning methods. They may integrate fuzzy logic, neural networks, or genetic algorithms with rule-based inference to achieve a higher level of decision performance.

11.5 APPLICATIONS OF EXPERT SYSTEMS (IMPORTANT)

Classical Applications of Expert Systems (ES):

- **DENDRAL:**
  - Initiated by Edward Feigenbaum in 1965.
  - Used rule-based reasoning to deduce molecular structures of organic compounds from chemical analyses and mass spectrometry data.
  - Demonstrated how rule-based reasoning could evolve into powerful knowledge engineering tools.
  - Led to the development of other rule-based reasoning programs, notably MYCIN.
- **MYCIN:**
  - A rule-based ES developed at Stanford University in the 1970s for diagnosing bacterial blood infections.
  - Utilized backward chaining with a rule base of about 500 rules.
  - Could recognize around 100 bacterial infections and recommend effective drug prescriptions.
  - Achieved diagnostic performance equal to that of human specialists in controlled tests.
  - Introduced reasoning and uncertainty processing methods with a long-term impact on ES development.
- **XCON:**
  - Developed by Digital Equipment Corp. for configuring VAX computer systems.
  - Used rules to determine optimal system configurations based on customer requirements.
  - Reduced processing time from 20–30 minutes to 1 minute.
  - Increased service accuracy from 65% (manual) to 98%, saving millions annually.

Newer Applications of ES:

- **Credit Analysis Systems:**
  - Support commercial lending institutions by analyzing customer credit records and determining proper credit lines.
  - Help assess risks and enforce risk-management policies.
  - Used by over one-third of the top 100 commercial banks in the United States and Canada.
- **Pension Fund Advisors:**
  - Example: Nestle Foods Corporation developed an ES for employee pension fund status updates.
  - Provides advice on regulation changes and compliance with new standards.
  - Example: Pingtung Teacher's College in Taiwan offers an Internet-based system for retirement planning with what-if analyses.
- **Automated Help Desks:**
  - Example: BMC Remedy's HelpDeskIQ (remedy.com), a browser-based rule-based help desk solution for small businesses.
  - Automatically processes incoming emails through a business rule engine, prioritizes them, and sends them to the appropriate technician.
  - Enhances problem resolution and issue tracking for help desk technicians.

Areas for ES Applications

As indicated in the preceding examples, ES have been applied commercially in a number of areas, including the following:

- **Finance.** Finance ES include insurance evaluation, credit analysis, tax planning, fraud prevention, financial report analysis, financial planning, and performance evaluation.
- **Data processing.** Data processing ES include system planning, equipment selection, equipment maintenance, vendor evaluation, and network management.
- **Marketing.** Marketing ES include customer relationship management, market analysis, product planning, and market planning.
- **Human resources.** Examples of human resources ES are human resources planning, performance evaluation, staff scheduling, pension management, and legal advising.

- **Manufacturing.** Manufacturing ES include production planning, quality man design, plant site selection, and equipment maintenance and
- **Homeland security.** Homeland security ES include terrorist threat assessment and terrorist finance detection.
- **Business process automation.** ES have been developed for help desk automation, call center management, and regulation enforcement.
- **Healthcare management.** ES have been developed for bioinformatics and other healthcare management issues.

11.6 STRUCTURE OF EXPERT SYSTEMS (Refer Textbook, Pg.No. 484**)**

11.7 KNOWLEDGE ENGINEERING (IMPORTANT)

What is Knowledge Engineering (KE)? With a neat diagram, discuss the role of various activities in the process of knowledge engineering.

Role of Various Activities in the Process of Knowledge Engineering

Knowledge engineering is defined as "the art of bringing the principles and tools of artificial intelligence research to bear on difficult application problems requiring the knowledge of experts for their solutions." It involves intensive activities to acquire, represent, validate, and utilize expert knowledge for intelligent systems.

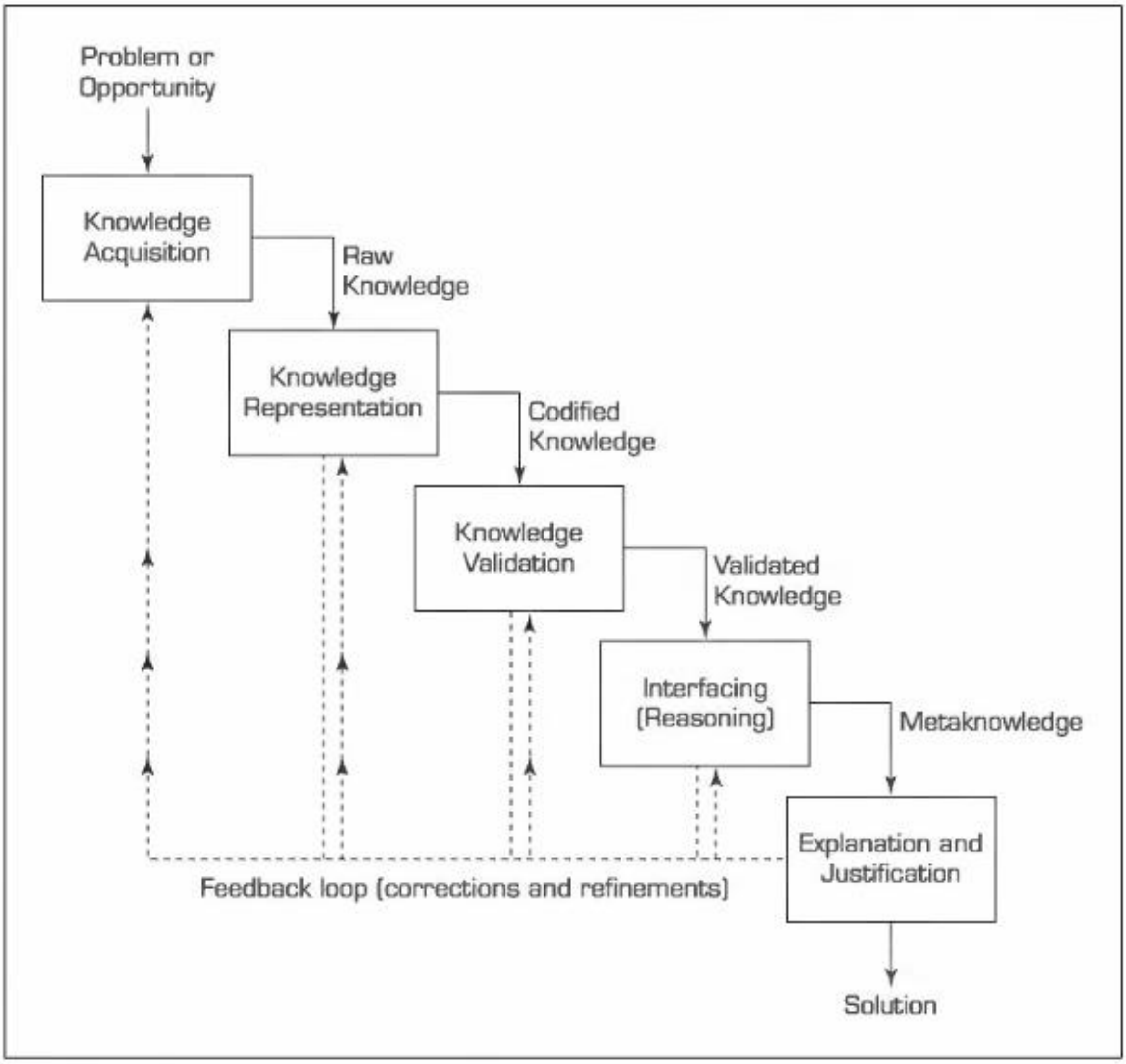

FIGURE 11.5 The Process of Knowledge Engineering.

1. Knowledge Acquisition

Definition:

"Knowledge acquisition involves the acquisition of knowledge from human experts, books, documents, sensors, or computer files. The knowledge may be specific to the problem domain or to the problem-solving procedures; it may be general knowledge (e.g., knowledge about business), or it may be metaknowledge (knowledge about knowledge)."

Key Points:

- Manual methods include structured, semi-structured, and unstructured interviews, reasoning process tracking, and observation.
- Automated AI-based techniques aim to reduce manual involvement but are still less common in real-world projects.
- Involves addressing challenges such as unstructured expert knowledge or conflicting expertise by narrowing the domain or engaging multiple experts.

Role:

- Forms the foundation by gathering essential knowledge from diverse sources to solve domain-specific problems.

2. Knowledge Representation

Definition:
"Acquired knowledge is organized so that it will be ready for use, in an activity called knowledge representation. This activity involves preparation of a knowledge map and encoding of the knowledge in the knowledge base."

Key Points:

- Common methods include production rules, semantic networks, frames, objects, decision tables, decision trees, and predicate logic.
- Production rules are most common, structured as "IF-THEN" pairs. Example:
  - *If the stop light is red AND you have stopped, THEN a right turn is okay.*
  - *If an international conflict begins, THEN the price of gold goes up.*
- Separates **knowledge rules** (declarative) and **inference rules** (procedural), which work together during problem-solving.

Role:

- Converts acquired knowledge into a usable format that enables problem-solving in intelligent systems.

3. Knowledge Validation

Definition:

"Knowledge validation (or verification) involves validating and verifying the knowledge (e.g., by using test cases) until its quality is acceptable. Test results are usually shown to a domain expert to verify the accuracy of the ES."

Key Points:

- Evaluation ensures the system is usable, efficient, and cost-effective.
- Validation focuses on ensuring acceptable performance compared to the expert's.
- Verification ensures that the system is implemented correctly per its design specifications.

Role:

- Ensures the accuracy, reliability, and usability of the system by validating the correctness of the knowledge base.

4. Inferencing (Reasoning)

Definition:

"Inferencing (or reasoning) is the process of using the rules in the knowledge base along with the known facts to draw conclusions. Inferencing requires some logic embedded in a computer program to access and manipulate the stored knowledge."

Key Points:

- **Forward chaining** (data-driven): Starts with available information and works towards conclusions.
- **Backward chaining** (goal-driven): Starts with a goal or hypothesis and seeks supporting evidence.
- Pattern matching is the process of identifying which rules in the knowledge base can be "fired" based on the current known facts.

(Refer to the textbook for the explanation of forward and backward chaining.)

Role:

- Implements the reasoning process that allows the system to make decisions and solve problems.

5. Explanation and Justification

Definition:
"An explanation is an attempt by an ES to clarify its reasoning, recommendations, or other actions (e.g., asking a question). The part of an ES that provides explanations is called an explanation facility (or justifier)."

Key Points:

- **Why explanations:** Explain why certain information or actions are required (e.g., "Why do you need to know my income?").
- **How explanations:** Show how a conclusion or recommendation was reached by tracing the chain of rules that were fired.
- Purposes of explanation:
  - Debugging systems.
  - Enhancing user understanding.
  - Conducting sensitivity analyses.
  - Building user confidence in the system.

Role:

- Makes the system more transparent and understandable, increasing user trust and allowing for debugging and refinement.

(Summary

The process of knowledge engineering involves the following activities:

1. **Knowledge acquisition** establishes the knowledge foundation by eliciting expert insights.
2. **Knowledge representation** structures the knowledge for computational use.
3. **Knowledge validation** ensures the system performs accurately and reliably.

4. **Inferencing** enables the system to draw conclusions and solve problems.
5. **Explanation and justification** provide transparency and user confidence.)

11.8 PROBLEM AREAS SUITABLE FOR EXPERT SYSTEMS

Expert Systems (ES) are applied in several problem areas. These systems are tailored to address specific challenges, as described below:

- **Interpretation Systems:** Infer situation descriptions from observations. Applications include surveillance, speech understanding, image analysis, signal interpretation, and intelligence analyses. These systems explain observed data by assigning symbolic meanings.
- **Prediction Systems:** Forecast outcomes in fields such as weather, demographics, economics, traffic, crop production, and military or financial planning. Examples include weather forecasting and economic predictions.
- **Diagnostic Systems:** Identify underlying causes of observed irregularities. Common applications include diagnosing medical conditions, electronic failures, mechanical issues, and software bugs.
- **Design Systems:** Develop configurations of objects that meet constraints. Examples include circuit layouts, building designs, and plant layouts. These systems ensure the designs satisfy specified constraints.
- **Planning Systems:** Solve planning problems in areas such as project management, routing, communications, product development, military applications, and financial planning.
- **Monitoring Systems:** Compare observations of system behavior with standards to identify potential flaws. Applications range from air traffic control to fiscal management tasks.
- **Debugging Systems:** Develop solutions to correct diagnosed problems using planning, design, and prediction capabilities.
- **Repair Systems:** Create and implement plans to remedy diagnosed issues. These systems integrate debugging, planning, and execution functionalities.
- **Instruction Systems:** Diagnose and address student knowledge deficiencies. These systems interpret student behavior, identify weaknesses, plan tutorials, and deliver remedial knowledge.

- **Control Systems:** Manage the overall behavior of a system adaptively. Tasks include interpreting situations, predicting problems, diagnosing issues, planning solutions, and monitoring execution.

11.9 DEVELOPMENT OF EXPERT SYSTEMS

The development of Expert Systems (ES) involves multiple stages and requires careful planning and execution. Key steps include defining the problem, acquiring knowledge, selecting tools, coding, and evaluating the system.

Defining the Nature and Scope of the Problem

- Identify the problem's nature and define its scope.
- ES is not suitable for problems solvable by mathematical optimization algorithms.
- ES is best for qualitative problems with explicit knowledge and available experts.
- The problem scope should be specific and narrow due to current technological limitations.
- For example, ES can detect abnormal trading but not confirm criminal transactions.

Identifying Proper Experts

- Proper experts are crucial for developing the knowledge base.
- Experts should have problem-solving expertise, decision support knowledge, and good communication skills.
- A project may involve one expert or a group of experts.

Acquiring Knowledge

- Knowledge acquisition, or knowledge engineering, involves eliciting decision knowledge from experts.
- A knowledge engineer interacts with experts to document knowledge.
- Challenges in knowledge acquisition include:
    - Experts' reluctance due to proprietary knowledge.
    - Difficulty in articulating tacit knowledge.
    - Limited time availability of experts.
    - Confusing or contradictory knowledge.

    - Misunderstanding by the knowledge engineer.
- Knowledge is usually represented as **if-then rules**, decision trees, or decision tables.
- Consistency and applicability of the knowledge base must be evaluated.

Selecting the Building Tools

There are three main types of tools for building ES:

1. **General-Purpose Development Environment:**
    - Use languages like C++, Prolog, and LISP.
    - Prolog and LISP are easier to use than C++, but all require professional programming skills.
    - Java and .NET platforms are useful for Web-based applications.
2. **Expert System (ES) Shells:**
    - Pre-designed for ES development with built-in inference capabilities and user interfaces.
    - The knowledge base is empty and needs to be populated.
    - Example: **Corvid System by Exsys** with variables, logic blocks, and command blocks.
3. **Tailored Turn-Key Solutions:**
    - Tailored to specific domains and adaptable for similar applications.
    - Adjusts the user interface or system to meet unique needs.

Choosing an ES Tool:

- Consider cost benefits; tailored solutions are the most expensive.
- Evaluate technical functionality and flexibility.
- Check compatibility with the organization's existing infrastructure.
- Assess tool reliability and vendor support.

Coding the System

- Code the knowledge base based on the tool's requirements.
- Focus on efficient and error-free coding with skilled programmers.

Evaluating the System

- **Verification:** Ensures the knowledge base matches the expert's knowledge, avoiding coding errors.
- **Validation:** Confirms the system solves the problem correctly and effectively.

BENEFITS, LIMITATIONS, AND CRITICAL SUCCESS FACTORS OF EXPERT SYSTEMS (ES)

Benefits

1. **Consistency in Decision-Making**: Expert systems ensure consistent application of rules and knowledge, reducing variability in decision-making outcomes.
2. **Expertise Accessibility**: They allow nonexperts to perform tasks at a higher level by providing access to expert-level knowledge and advice.
3. **Knowledge Preservation**: ES store expert knowledge, ensuring it is retained and reused even after experts are unavailable.
4. **Improved Efficiency**: They speed up the decision-making process, saving time in diagnosing problems or performing tasks.
5. **Cost Reduction**: By automating tasks typically done by experts, ES reduce costs associated with expert consultation and training.

Limitations

1. **Narrow Domain Knowledge**: Expert systems are effective only within a well-defined domain and struggle with generalized or cross-domain problems.
2. **Lack of Adaptability**: Unlike humans, ES do not adapt to new or unforeseen situations unless explicitly updated.
3. **High Development Costs**: The process of creating and maintaining an expert system, especially acquiring knowledge, can be time-consuming and expensive.
4. **Limited Common Sense**: ES lack intuitive understanding and rely strictly on programmed rules and data.

Critical Success Factors

1. **Well-Defined Scope**: Clearly defining the problem area ensures the ES focuses on solvable tasks with qualitative knowledge.

2. **Expert Collaboration**: Identifying and working closely with knowledgeable and cooperative experts is essential for effective knowledge acquisition.
3. **Quality Knowledge Base**: Accurate and well-organized knowledge representation, such as rules or decision trees, is crucial.
4. **Effective Tools and Technologies**: Selecting appropriate development tools, such as ES shells or tailored turn-key solutions, enhances system functionality and integration with existing infrastructure.
5. **Evaluation and Validation**: Continuous testing and validation ensure the ES performs accurately and reliably