CHAPETR 3: PROBLEM SOLVING

1 INTRODUCTION

Blake planned a career in operation research, but a summer internship at the World Bank changed his mind. When asked why, he said,” I didn’t see one application of OR at the World Bank last summer.” The next summer, Tania interned at the World Bank and developed a linear programming model to allocate resources in the Egyptian agricultural sector. They hired her for another year to continue work on the model. Why was the experience of the two students different?

Blake expected someone to provide a well defined problem he could plug into a computer program and solve. This seldom happens. Tania examined issues important to the World Bank and modeled a problem solvable by linear programming. Tania was a problem solver; Blake was not.

Problem solving is difficult to teach; it is more art than science. To learn problem solving, you must do it. As in riding a bicycle, explanations are helpful, but eventually you must do it yourself. You may fall, but that is usually necessary in learning to ride a bike. There is no magic formula for problem solving.

1.1 Problems

What is a problem? Defining it is not so easy. A **problem** exists when what is happening differs from what should happen. What is happening is the current state, and what should happen is the goal state.

Problems may be one-time problems or recurring problems. Deciding how many widgets to produce each month is a recurring problem, but determining why a machine failed is a one-time problem. Many concepts for problem solving apply to both kinds of problems, but these problems have different situations. Recurring problems require continuing data collection, report, and other infrastructure.

Every problem has a lifespan. Some problems must be solved quickly, but others are not urgent. We cannot take too long to develop a schedule for programs run on a computer, because the time to run programs id usually short. On the other hand, we have more time to determine the location of departments in a plant to be built next year.

Problems have also different impact. Problems solved should be worthy of the resources required to solve them. If it costs more to do the study than it will save, don’t do it. The problem impact should determine the around of effort we put into solving the problem.

1.2 Solutions

Problems typically do not go away unless we do something to resolve them. This intervention is **problem solving**. We look for easy and quick solutions, but complex problems often require complex solutions. Solving most well-structured problems is easy. To find the minimum of a quadratic function, we use first and second derivatives. Solving ill-structured problems is not obvious. How would you reduce world hunger? A large part of problem solving is transforming ill-structured problems into well structured problems.

Ackoff (1991) discusses four approaches to a problem. The first, **Absolution,** ignores the problem and hopes it gone away, which is seldom a good approach. The second approach is resolution. **Resolution** finds an acceptable solution to the problem using common sense; resolution is usually better than absolution, but the answer may not be a good one. **Solution** is his third approach. It uses quantitative and experimental methods to get the "best" answer under the current conditions. The fourth approach, d**issolution,** redesigns the system to eliminate the cause of the problem. This approach, if possible and not too costly, is the preferred one.

To solve a problem five conditions should exist:

1. A gap between the current state and the goal state,i.e., a problem exists
2. An awareness of the gap, in which we recognize the problem
3. Motivation to close the gap- the problem is important to someone and has impact, and resources will be committed to solve it
4. The ability to "measure" the size of the gap; we have an idea of the severity of the problem and know when improvement occurs
5. The ability and resources to close the gap; we have the methodology to solve the problem and the resources to carry out the solution.

If one or more of these conditions is missing, successfully solving the problem is unlikely. If these conditions exist, we can proceed with problem solving. Although there is no one best way to solve ill-structured problems, we present a framework that may prove useful.

1.3 Problem solvers

Who is the problem solver? The person who has the problem or someone paid to solve problems could be the **problem solver**. In the production arena, the problem solver may be a manager, analyst, or industrial engineer. Often there are several problem solvers working together; for case of discussion, we will use the singular.

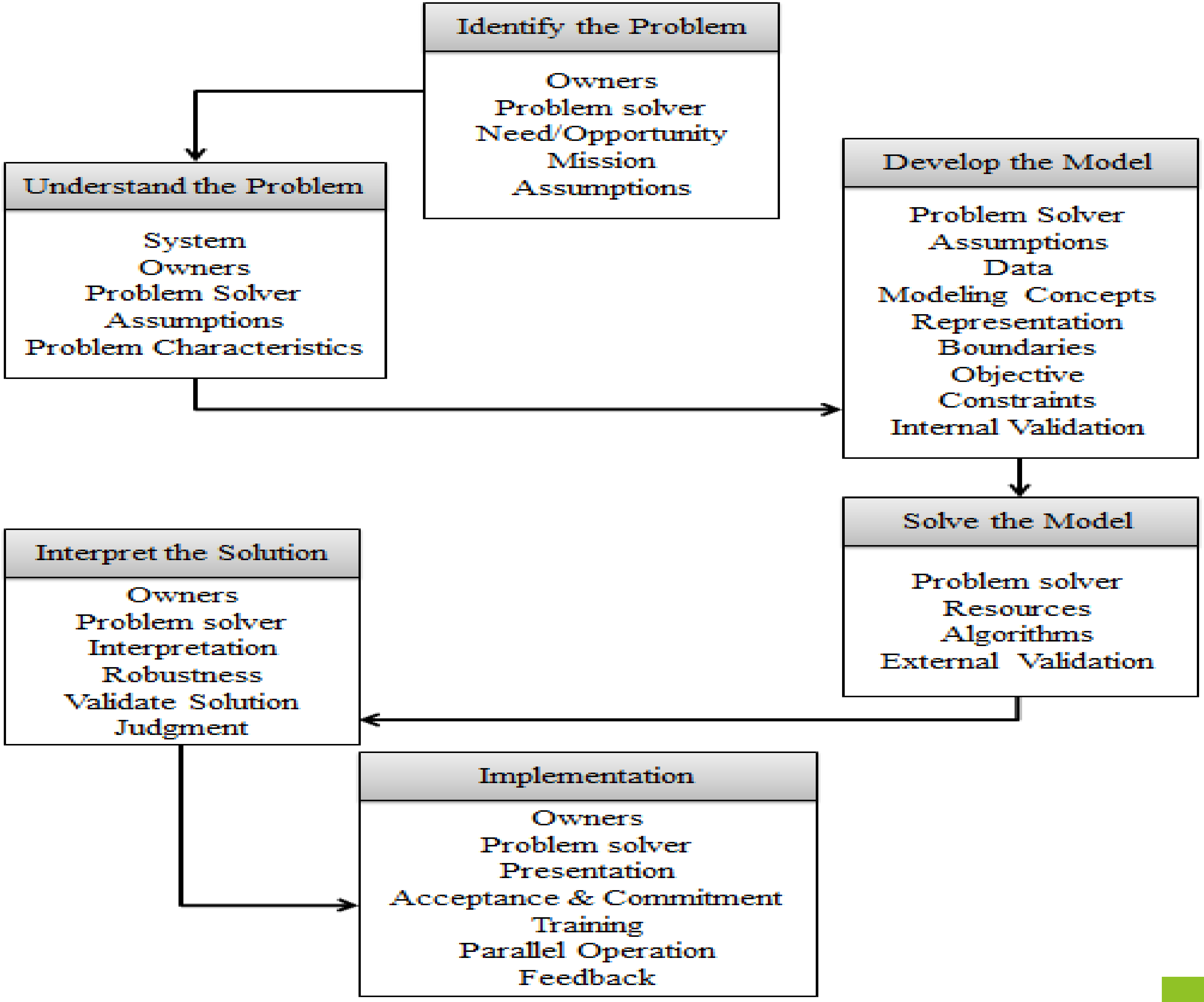
Because problem solvers are people, they are not infallible. Personal values, biases, and judgment at affect the problem-solving process. Whether a problem exists or not is affected by a person's point of view, but recognizing that bias exists should minimize its impact.

The knowledge and experience of a problem solver also comes into play. If a person has more tools, that person has a wider range of options for solving problems. Experience teaches which tools to use in certain situations and even helps in inventing new ones or adopting old tools to new situations. If a particular tool is not in a tool kit, you will not use it.

2 PROBLEM SOLVING APPROACH

There are many approaches to problem solving. A general one includes problem identification, generation of solutions, and choosing a solution. We present a six step process, outlined in Figure 3-1. These steps are:

1. Identify the problem
2. Understand the problem
3. Develop a model
4. Solve the model
5. Interpret the solution
6. Implementation



Identify the Problem

Owners
Problem solver
Need/Opportunity
Mission
Assumptions

Understand the Problem

System
Owners
Problem Solver
Assumptions
Problem Characteristics

Develop the Model

Problem Solver
Assumptions
Data
Modeling Concepts
Representation
Boundaries
Objective
Constraints
Internal Validation

Solve the Model

Problem solver
Resources
Algorithms
External Validation

Interpret the Solution

Owners
Problem solver
Interpretation
Robustness
Validate Solution
Judgment

Implementation

Owners
Problem solver
Presentation
Acceptance & Commitment
Training
Parallel Operation
Feedback

FIGURE 3-1 An overview of the problem-solving process

If all steps are successful, the process begins at *Identify the Problem* and then proceeds to understand the problem, *Develop the Model*, and so forth. No step can be skipped. It is likely that a step still not be successfully completed; we must then return to previous step. If we do not the complete the *Solve the Model* step, to be determined by the reason we could not solve the model. When we return to a step, the following steps must be done consequently.

The first step is to identify the problem, which includes identifying the problem owners and ,together, determining the problem mission. Assumptions are made at this step. After successfully identifying the problem, the problem solver and owners must understand the problem, which includes examining the system within which the problem occurs, specifying problem characteristics, including goals, and possibly making more assumptions. Validation ensures that the current problem is solved at later steps. If this step is unsuccessful, a return to problem identification will be necessary. Once the problem solver and owners agree on the problem, a formal model is developed. Modeling concepts and data availability determine a representation for the model. Then, boundaries, constraints, objectives, and relationships are used to produce a formal model. Validate the model to make sure it does what the problem solver intended to do. Failure at this step may require to return a either problem identification or problem understanding. After constructing the formal model and collecting data, an appropriate algorithm is used to solve the model. Again, unsatisfactory results force a return to a previous step. Once we have the model solution. We interpret it considering the actual problem. Robustness, solution validation, and judgement lead to a problem solution. If necessary, return to a previous step to resolve differences. Finally, implement the solution. Implementation begins by proposing a solution. Once the solution is accepted, resources are committed to solve the problem. Appropriate people are trained and the new solution is implemented in parallel with the old procedure. Be sure the monitor progress for continued success.

We discuss each of these steps in some detail. Some topics, such as assumptions and owner implement, appear in several steps. Unless there is something different about them in particular step, we only discuss them the first time they occur.

3 PROBLEM IDENTIFICATION

Initially symptoms arise; from these we must make a problem diagnosis. The problem may arise from a need, an opportunity, or both. The problem solver and owners develop a problem mission, which determines what needs to be accomplished. To do so, assumptions are often necessary. Do not try to visualize a solution now; it will only restrict the ideas on problem identification. This step is a continuous interaction between owner and problem solver and results in an initial problem statement.

Problem identification converts a “mess” into a simple problem statement. Tentatively, we describe our current state and goal state. If a production line produces many items rejected by the customer, the current state is a line that is producing poor quality items; the goal state would be a line that produces perfect items.

Identification is an important step in problem solving. A problem never recognized will never be solved. Ackoff (1991) believes it is better to get the wrong solution to the right problem than the right solution to the wrong problem. Solutions to the wrong problem are ignored, but wrong answers to the right problem create interest and are corrected and used.

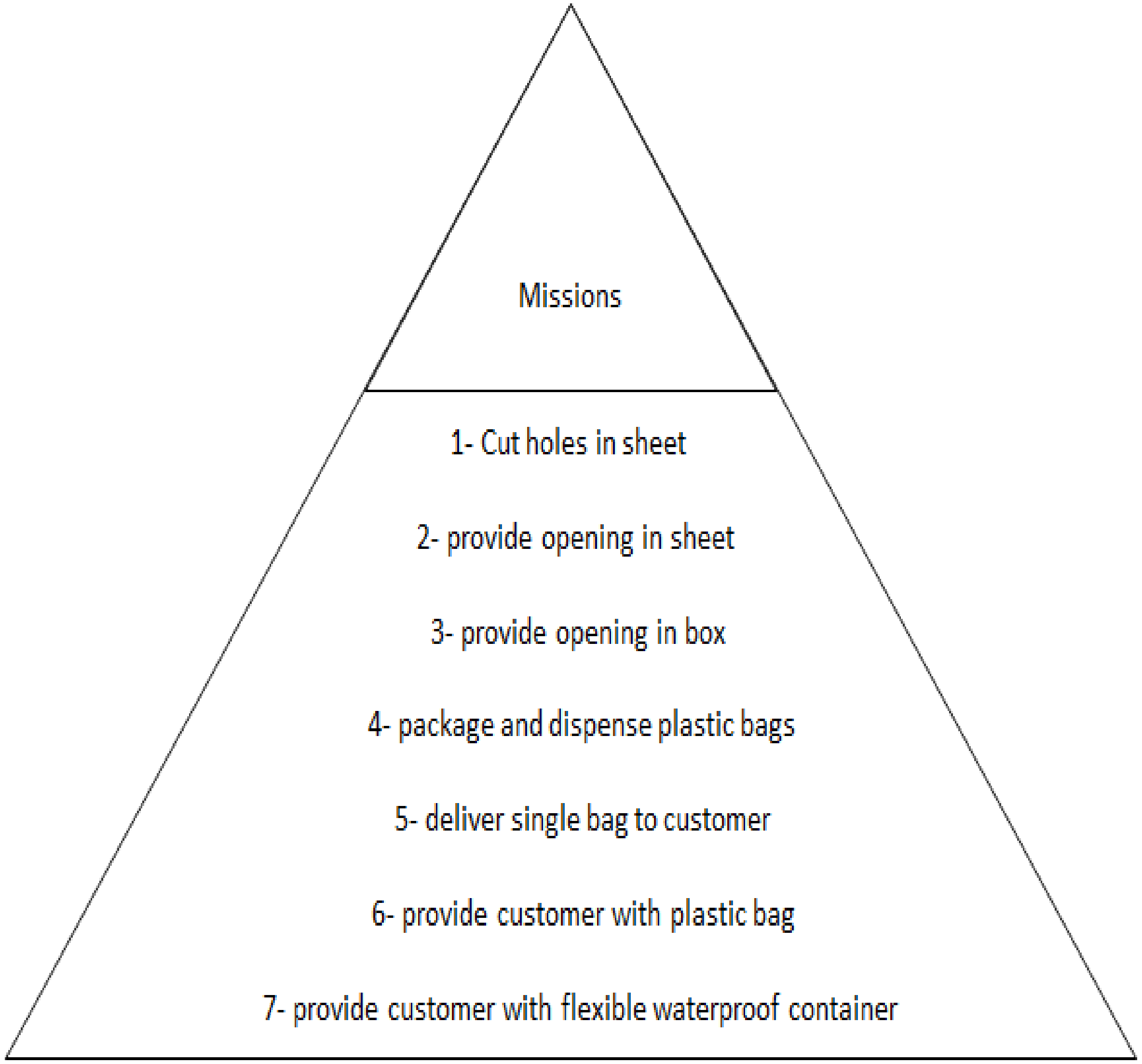
Two sources of problem are need and opportunity. A broken machine is need driven problem. Sometimes, need-driven problems are hard to recognize. In 1962, General Motors made 51percent of all cars and light trucks sold in the United States. By 1991, their market share was only 35 percent. Their failure to respond to a changing market is a classic example of unrecognized problem.

Even if there are no complaints, we may still have a problem. Satisfied customers are not enough; we may want to improve quality to increase our competitive edge. This situation is opportunity driven. If it isn’t broke, make it better! The Japanese insistence on continuous improvement is an example of opportunity-driven problem solving. To recognize opportunity-driven problems, we should follow the advice of Shaw who said, “I dream things that never were, and say ‘why not?’ ”

3.1 Problem mission

The most important phase of problem identification is to determine the **problem mission**. The mission is the overall purpose – what we want to accomplish. The mission will later be translated into goals, and the objectives. Different missions result in different solutions. For any problem, we could give many missions.

We may view missions as a pyramid with several levels. Each level represents a different mission, the most specific at the top and becoming more general as we move towards the base. As an example, in a company that makes plastic bags. The bags are packaged in a cardboard box. One side of the box has a tab that can be pulled to create an opening. Individual bags are pulled out of the box through this hole. Currently, the company has problems cutting the perforations in the sheet of cardboard that becomes the box. List of missions is given in figure 3-2.



Missions

1- Cut holes in sheet

2- provide opening in sheet

3- provide opening in box

4- package and dispense plastic bags

5- deliver single bag to customer

6- provide customer with plastic bag

7- provide customer with flexible waterproof container

FIGURE 3-2 Hierarchy of problem missions

The first mission is quite different from the last. Cutting holes in the sheet restricts the possible solutions, while providing a customer with a flexible waterproof container places the problem in a new perspective. Pursuing a more general mission will probably result in greater change; in this case the company might change its entire product line.

It is not always best to pursue the most general mission. Time and resource constraints, or even policy decisions, may limit mission. For example, the package and dispense plastic bags may be as far as we are willing to go. Choosing this mission eliminates the original problem because we may no longer need to cut holes in the sheet. Of course, we must now solve the problem of how to package and dispense bags.

Defining the correct mission will prevent confusing symptoms with problems. Problem identification has much in common with a doctor making a diagnosis. The problem solver looks at symptoms and determines the correct problem statement. A melt-spun ext-ruder broke the key connecting a gear to the main drive shaft. After breaking twice, a mechanic assumed the mission was to keep the key from breaking, so he replaced it with a case-hardened steel key. Naturally, the key did not break again, but the shaft would up looking like a pretzel.

The broken key was a symptom rather than the problem. In fact, the key did its job- it protected the shaft and gear. The true mission was finding out what caused the excess force applied to the key. Once the mission was known, a maintenance engineer found and fixed the problem. Often, identifying the right problem is more difficult than getting a solution.

To differentiate between symptoms and causes, ohno (1988) recommends asking why?, five times. When the key broke, ask why. The answer would be that there was too much force applied. Now ask why there was too much force. When that is known, continue to ask why until the true problem is discovered. İf the problem solver asks why at least five times, the real problem will probably be found.

Once the right mission has been chosen, identifying the problem is easier. Be careful to spend enough time on this phase. Do not try to incorporate solutions into problem identification- that will come later. Observing what is happening now will be very helpful. Also, brainstorming with a group of knowledgeable people can help clarify the problem.

3.2 Problem owners

Problem owners are people who must live with the solution. It would be unusual in a production environment (or many other environments) to have a single owner. Often, different owners have different stakes in the problem and even different goals. Carefully review these stakes and goals and *continually* involve the owners in the solution process. Do not try to place blame for a problem. Make the problem a common enemy so that work with you rather than against you.

Failure to involve the owners continually may be disastrous. There are many honor stories of the problem solvers developing an initial problem describing and going back to the office to solve the problem. After spending time, effort, and resources, they present their solution to the owners, only to discover the problem they solved does not really exist. Turning in an assignment that was not what the professor wanted is a typical example.

Often owners only recognize symptoms and don’t recognize the problem, which hampers problem identification. A good diagnosis is necessary and requires a continuous dialogue with the owners.

3.3 Assumptions

When identifying the problem, we seldom know all the facts, so we make assumptions. If you work on well-structured problems, assumptions may be necessary. However, relationships between various parts of most problems are uncertain, which requires the problem solver to make assumptions about them.

It is very important to state assumptions explicitly. Then everyone can question and comment on them. Explicit assumptions remind us to check their influence on the solution. If an assumption is questionable and will have a large impact on the solution, try to use preliminary experiments to justify or change it.

Sometimes we must make questionable assumptions in spite of everything. It is better to make a questionable explicit assumption than not to state an assumption. Sensitivity analysis can determine the effect of assumptions. In any event, assumptions should be “reasonable”, i.e. fit within the general problem environment. If possible, justify assumptions by observation, empirical data or evidence, or judgment of the owners.

3.4 Initial problem statement

Once you identify a problem, write down a “formal” problem statement. Include a one or two sentence description of the mission and a brief description of the current state and the goal state. Do not include restrictions, and do not go into great detail. List all assumptions. We will talk more about it in our case study.

4 UNDERSTAND THE PROBLEM

From the initial problem statement, we refine our understanding of the problem. Because problems do not exist in a vacuum; the problem solver must understand how the problem fits into its environment. We must describe the system within which it occurs. Once the boundaries of the problem are determined, the solver and owners identify problem characteristics. They also make the problem mission more specific by identifying solution goals. New assumptions may be needed in this step, leading to a deeper understanding of the problem. The problem must be validated to ensure the problem is the right one. Methods for problem understanding are similar to those used for problem identification, but at a more detailed level. We may need to redefine the system, make assumptions, or even return to problem identification. A detailed problem statement is a result of problem understanding.

4.1 The systems perspective

A **system** is a collection of interacting components; its function cannot be done by any single component. Machines are one component of a production system. By itself, a machine cannot make a finished product, but correctly coupled with other machines, people and raw material, it becomes a system capable of making a product. The machine is also a system made up of components; e.g. power supply, tools, etc. Thus whether something is a system or a component depends on the particular problem. Problems often occur in the way parts of the system interact with each other. We must understand problems within the system’s framework.

Analysis is one way to study a system. The system is taken apart and each component studied separately to see how it works. The knowledge of the components is combined to gain knowledge about the system, which can usually tell us how a system works.

Synthesis is another way to look at a system. Synthesis views the system as a component of a larger system and we try to explain the behavior of the larger system. It tells us how the system operates and tells us why it operates as it does.

Figure 3-3 shows the difference between analysis and synthesis. Suppose the problem occurs within the system outlined in solid bold lines. Analysis is represented by the arrow pointing left. It looks within the system to see how the components interact, which tells how the system works. On the other hand, synthesis, represented by the arrow going out of the system, views the system as a component of a larger system, shown by the dashed lines. By considering its interaction with the components of the larger systems, we can discover why the system works as it does.

When examining a problem within the systems framework, important question to as are who, what, why, when, where, and how does it; what do they do; why do they do it; when do they do it; where do they do it; and how do they do it? These questions further define the current state.

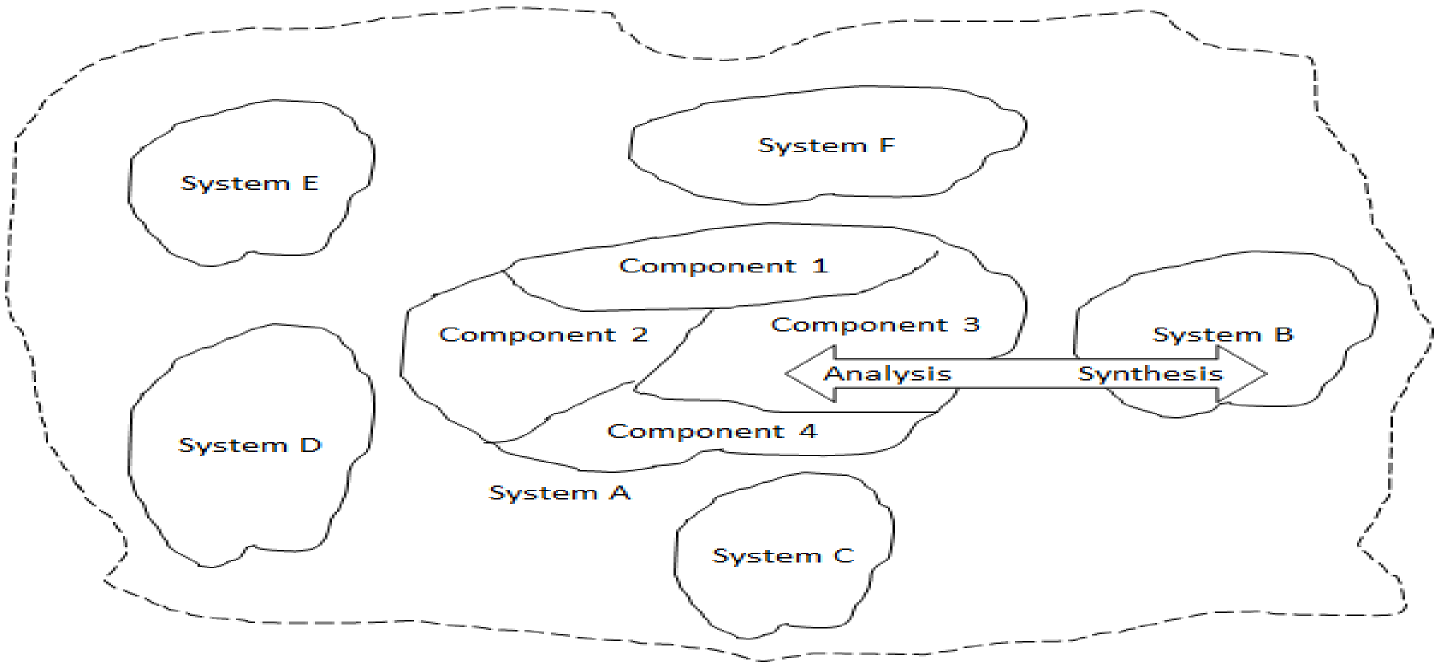


Figure 3-3 Analysis and synthesis of a system

4.2 Goals:

Given an initial problem statement and further understanding of the problem we define the solution **goal**. The mission is our overall purpose, but the goal should be one or more accomplishments that will lead to fulfilling the mission. If our mission is to have satisfied customers, the goal might be to improve the quality of our product or service. Other common goals might be to reduce the time to carry out a task or provide a product to reduce the associated cost. There is no need to describe the goals in minute detail, only to state them in a general way.

4.3 Problem characteristics

The time frame in which the problem exists is important. If it is a one-time problem, the solution must be given before the problem disappears or changes. For example, we must determine the best way to lay out a production facility before using it. Similarly, if a manager needs to know how many employees to hire for next month, giving an answer two months from now does not help. Thus, the time available to solve a problem determines the approach we take to solve it.

Does the problem recur or is it a one-time situation? Recurring problems often require more resources. Reports, data collection and other infrastructure needed for recurring problems are more elaborate than for one-time problems.

We also need to determine a proper level of detail for the problem. A dairy deciding how much skim, 2 percent, and whole milk to produce next week would not need to know the exact amount of milk an individual cow gives, only the total of all cows. Deciding which cows to breed requires knowing individual yields.

Next try to determine if any physical laws control part of the problem. Conservation of matter and the law of gravity are typical physical laws. In production systems, physical laws are seldom problems. Company policy usually places far more restrictions on the way systems function. Do not include policy restrictions now; they are not absolute physical laws.

The characteristic and impact of the problem determine how much time and effort can be spent solving the problem. Important problems with a long time frame are worthy of time and effort. However, if the potential payoff is small, we should resort to “quick and dirty” solutions or not to solve the problem at all. Quick and dirty methods may also be right for important problems with a short time horizon.

4.4 Validate understanding

Our understanding of the problem is an abstraction of the real problem. If we understand the actual problem, our abstraction should capture its important features. We must make an effort to ensure this; i.e. we **validate** our problem understanding…

Validating understanding is not easy. Try describing the problem to the owners. If you cannot, it is unlikely you understand it. Critically question every part of the problem description. It helps to have someone else critically question your understanding of the problem. A model based on faulty understanding does not produce good solution to the problem. If you cannot validate your understanding of the problem, return to the problem identification step.

4.5 Problem statement

We can now write a more precise, detailed problem statement. After further study of the problem, we may have revised the mission or modified assumptions. Document all changes.

As before, involve the problem owners. It is helpful to have someone who is unfamiliar with the problem involved which will help avoid the "can't see the forest for the trees" syndrome. Do not try to hurry through this phase to get a solution. Time spent here saves wasted time and effort later.

Now make the problem statement more formal, usually done by developing a formal model. We discuss this step in the next section.

5 DEVELOP A MODEL

In this step, we turn a detailed problem statement into a formal model. A **model** is a representation of something. The problem solver uses available data, modeling concepts, and assumptions to choose a type of model. Then, specific data and the problem boundaries help generate an objective and constraints applicable to the model. Assumptions affect the model.

A model is then proposed and structurally validated, which ensures the model works as it should. If unsuccessful, we return to any previous step. Eventually, we construct a formal model statement.

We discuss various types of models, followed by sources and uses of data; then we discuss modeling itself. Again, involve the owners, and clearly state all assumptions. During this step, we may need to return to a prior step.

5.1 Model representations

Models can be formal or informal, qualitative or quantitative. They are used to test an alternative, predict the behavior of a system, determine the best of many alternatives, or explore what-if questions. We can usually test more alternatives with a model than we could by direct experimentation. We briefly discuss iconic, analog, and symbolic models.

Iconic Models: are physical representations that usually have different scales that what they represent. Wind tunnel tests of scale model airplanes are less costly and time consuming than building actual planes to test. Traditionally, industrial engineers use iconic models of buildings and templates for machines to do layout studies. Iconic models are often used with computers. For example, CAD systems. These models are easy to explain, because they look like the real thing.

Analog Models: behave the same way the system does but do not look like the system. A flowchart showing the information flow in a production system is an analog model. We have used analog representations for the steps in problem solving. Historically, flows in systems were modeled by electrical circuits, because current flows in electrical circuits are easily calculated.

Symbolic Models: are important classes of models. As the name implies, the system is represented by symbols. This class includes graphical, tabular, and mathematical models.

For simple systems, a table of graph can be used as a model. Speed versus number of good products made could represent the output of a machine on the production floor. This relationship could be shown by a table or a graph or even as a mathematical question. Spreadsheets are helpful for these types of models.

Systems too complicated for tables and graphs use more complicated models, simulation models, which use relationships between components, are symbolic models. Often the relationships are stochastic, and they may not be defined as precisely as those used in mathematical models. Mathematical models capture relationships between components using mathematical equations. Linear programming formulations are typical mathematical models. Changes in input data usually do not affect the model structure, only the solution, which makes models easy to use repetitively.

5.2 Data

Data are often needed to identify and understand a problem. Data are used to validate assumptions, estimate model parameters and validate models and are usually collected at every step of problem-solving process. Availability of data may dictate the type of model used.

Data represents characteristics of people, objects or events. Examples of data would be the number of machine hours available in the finishing department; the number of units a worker produced in one shift; the number of gallons of crude oil needed to produce one gallon of kerosene; and the cost to make one unit of product. Some data may be functions of other data. But all data can be given values before solving the problem.

Data are not information; information describes or explains data. The number of units produced during a shift is data, and the average over many shifts is information. Be careful when extracting information from data; some information is unknowable exactly.

Data are used to get information. *Do not collect data for the sake of collection data*. Data collection is often the most expensive part of the problem solving. If it does not provide information needed to solve the problem, do not collect it. Many projects start by trying to get all the data possible, usually at a large cost of time and money. At this point, determine the data needed to solve the problem, and only collect this data.

A pessimistic view is that data are seldom available, and when they are, they are in the wrong form. If by chance data are in the right form, they are not accurate. More realistically, seldom will we have all the data we want, or even all we need. Availability of data may dictate the type of model used.

Where do you get the data? The answer to this question depends on the data. Company records, people, government, and trade associations are all potential sources.

Company records are probably the best source of data. These records include accounting data, standards, sales and inventory reports, financial reports, engineering drawings and blueprints, or product brochures. Make sure you understand what these data are before they are used. For example, a cost accountant may have three or four ways to determine the value of one unit of product in inventory.

People are another source of data. Be careful to get facts rather than opinions when dealing with people. Problem owners are a source of data, as are customers, vendors, and even competitors. Interviews and questionnaires are two ways to elicit information from people, but the questions asked should be throughout so that the data will provide useful information.
Governments and trade associations often provide data; various industry reports produced by the government contain useful data. Research reports and regulator publications are also sources of data. Magazine and journals published by trade associations may help.

The cost to get data will vary greatly depending on the source. Collecting data is expensive, but data from an existing database are relatively cheap. If the cost of the data is too high, the model may have to be changed so that the data are no longer needed. Also, if the data cannot be gotten before the problem must be solved; it will do little good to collect it.

Data must reflect the physical phenomenon, which we call data integrity. If data show a product needs two units of a particular component, it should use two, not one or three. Loss of data integrity occurs in several ways, but usually it is either an input problem or an accuracy problem. Control input by standard data processing techniques.

Data must be accurate, and the accuracy required should match the problem. To determine how many widgets to produce next month, you do not need to know the time to make one widget to six decimal places. Also, if there are several types of data, the least accurate data determine the model’s accuracy. Once we develop a model, we can determine how variability of the data affects its solution.

Data may not been accurate enough. At a warehouse, bar code readers used to pull cases of yarn for shipping are 99.9 percent accurate. Each truck holds about 250 cases, so about one in four trucks has a wrong case on it- no longer considered good service.

No matter where data come from, do not trust the data without verifying it yourself. An industrial engineer’s first job was to lay out a storage area. After getting floor plans for the area, a layout was drawn up, and a work order paint lines on the floor was prepared. After painting, the engineer visited the area. Amazingly, there were columns right in the middle of the twelve-foot-wide aisles! The building had been renovated, and no one had updated the plans; they were completely wrong. If the engineer had measured the building and compared the measurements to the plans, the problem would have been avoided.

Be wary of single-source data. If not verified by observation or another source, consider the data questionable. Plan to use sensitivity analysis on the values of the data when you generate solutions. Also, the further removed from the shop floor the source of the data, the more you should question it.

5.3 Modeling Concepts

Modeling is an art; there is no recipe to tell how to do it. There are, however, some concepts that may help, we give some now.

It is impossible to completely understand most problems. There is too much uncertainty. Thus, several models of the problem may be valid. The modeler should develop the simplest correct model the owner understands.

Constructing a model is in itself a worthwhile task, even without a solution! Modeling requires a thorough understanding of the problem, which increases insight. Folklore insists that developing the model provides 90 percent of the value of the simulation project. In fact, when the model is "solved," the problem solvers often find the solution obvious because they now understand the problem so well.

In addition to using the models to solve a particular problem, we also use them to gain insight into current operations. Ackoff (1962) discusses retrospective optimization, in which we use the model to impute data from a given situation. This model is similar to reverse engineering, in which a product is torn apart to deduce how it was made. In the classical price elasticity model, an estimated parameter gives the relationship between the price of a good and the number sold. This model is typically used to predict the sales, given a selling price. However, we could use it to determine an imputed parameter value by plugging in the actual price and number sold.

The modeler should ask if the problem fits a standard model, such as a problem network. If so, the modeling task is much easier. We simply define needed data, collect them, and apply the right solution algorithm for that model. If the problem does not fit a standard mode, we try simplifying assumptions so that it does. The assumptions must not hide important information about the problem.

Paraphrasing Einstein, “Models should be as simple as possible, but no simpler.” Start with broad models and add detail as needed; simple models are more general and give more insight. More detail requires more data or assumptions, making the model more “accurate,” but less general. Also, owners understand simple models more easily.

To determine key factors and important interactions between problem components, try rephrasing the problem to give a new perspective. If possible, draw diagrams or construct tables that highlight important aspects of the problem.

Break complicated problems into easier pieces. Try to recognize parts of the problem that are standard, and deal with them first. When you understand these parts, go back and include the unfamiliar parts into the model. It is often difficult for novice modelers to ignore parts of the problem. Remember that ignored parts will be included later.

Define notation, parameters, and variables carefully and precisely. Use them to express relationships that exist between parts of the problem, and discuss them with someone who can critically comment on them. Define any underlying physical laws that are appropriate. Also, search for empirical laws that may help simplify or provide understanding for the model.

We now discuss some common components of models:

5.3.1 Boundaries: the first step in describing the problem is to determine the important parts of the problem. Important information should be included, and irrelevant details should be left out in order to determine appropriate **boundaries** that contain the problem. We do not necessarily exclude everything outside the boundaries, but treat anything outside as given input to the problem.

For example if new are considering equipment and layout for a production facility, the boundaries may be the building walls.

Be careful in defining boundaries. A problem solution for a particular department is a plant that ignores interaction with other departments could be disastrous.

5.3.2 Objectives: Once we tentatively set boundaries, we determine the objectives. An objective is a refinement of a goal. It must be measurable. If the goal is to improve the quality of a product, the objective might be to decrease the number of items defective.

Be certain that the stated objectives accurately reflect the goals. One company decided to have all shipments on time. As the shipping date approached, a truck containing only one or two cases of goods was sent out rather than risk a late shipment. The company did not understand why their trucking costs increased dramatically. Although timelines of deliveries is important, other factors, such as trucking costs, should be considered.

5.3.3 Constraints: With unlimited resources, solving most problems is easy, but there are usually limitations or **constraints** on what we can do. These limits may be on people, time, knowledge, data, capacity, technology, money, or other resources. Constraints are inviolate- they must hold. Defining them is very important. Ignoring a constrain results in a problem solution that will not work, but including nonexistence constraints restricts the solution alternatives.

Question the validity of constraints. Often, people believe something done a particular way for a long time must be done that way. Thus, a policy is seen as a constraint rather than something that can change. One company stated that changeovers could only be done early in the morning. When questioned, the foreman admitted that they had always done them then. No one could give a reason why they could not be done anything during the day, even on second and third shifts. Dropping this necessary "constraint" on changeovers resulted in a much more flexible production schedule.

Resources can be increased. If knowledge is a constraint, hiring experts may remove the constraint. Budget constraints are rarely absolute; showing that a small increase in budgets save money may increase the budget. The problem time frame also affects changes in resources. To increase capacity, working overtime is a short-term solution; increasing equipment is a long-term solution.

5.3.4 Relationships: once boundaries, objectives, constraints, and data are defined, we determine the **relationships** between them. First we define notation; it should be precise, with no ambiguity. Then, using this notation we describe relationships between various parts of the problem, which are linked by the variables.

Variables are the unknowns in the problem. Examples of variables are the number of workers to hire next month; the number of workers to have at the fast food counter during a day.

5.4 Assumptions and involvement

Although we may sound like a broken CD player, we again stress the need to explicitly state all assumptions made at this step. Involve the problem owners in modeling to ensure that the model is a reasonable representation of the problem. Depending on the results, we may need to return to a previous step to make revisions.

5.5 Internal validation

Validation ensures that the solution will be relevant to the real problem, and it occurs throughout the problem-solving process. After constructing the model, we check to see if it is doing what we intended, which we call **internal validation**.

Internal validation begins with a careful evaluation of the logic of the model. This process should be done by someone familiar with the problem that knows modeling concepts but did not actually construct the model. If not possible, try going over the model with someone who is familiar with the problem. Explaining the model to them may uncover modeling flaws; flow charts are helpful tools in explaining logic to others. Dimensional analysis can also prove helpful; if parts of the model have inconsistent units of measurement, the model is incorrect. Logic in computer programs should be checked and the programs debugged. Data required by the model should be available. If not, either a plan to get the data or a change in the model is necessary.

6 SOLVE THE MODEL

Starting with the model statement, the problem solver chooses and algorithm. This choice depends on the resources available to solve the problem, the available algorithms and the precision of solution desired. Solve simplified problems to confirm that the model is appropriate for the problem. We may need to return to some previous step in the problem-solving process if our results are not satisfactory. When the model and solution algorithm are satisfactory, we solve the model and get a model solution.

6.1 External Validation:

External validation tests the model solution. It finds inconsistencies in the model not fond during internal validation and validates the solution procedure as well. Here, we are not transforming the model solution into the problem solution; we are just making sure that the solution to the model is correct. For simple models this may seems a waste of time, but for models with many variables it is sometimes difficult to know if they are being solved correctly. **Model simplification** and **historical analysis** are two strategies for external validation.

6.1.1 Model simplification:

Simplification changes the model or data so that we know the model solution. If it correctly solves the simplification, we have more confidence in its solution to the actual problem. İf there is difference between the model solution and the "known" solution, we reevaluate and change the model or problem statement.

The easiest simplification is to look at small instances of the problem. For very small instances, we can list all the possibilities and hence know the correct solution. We can examine the entire model for these small problems and do calculation by hand to check the logic.

Another approach is to make simplifying assumptions about the data and system configuration. Suppose that we are simulating a queuing system for which there is no analytical solution. By assuming exponential inter-arrival and service times and allowing only one server, we can get an analytical solution. Then we can run the simulation program under the same conditions and compare the results. İf they are close, we have more confidence in the model.

We can force certain variables in the model to take on specific values, thus effectively removing them from the model. We can then examine the solution of the simplified model.

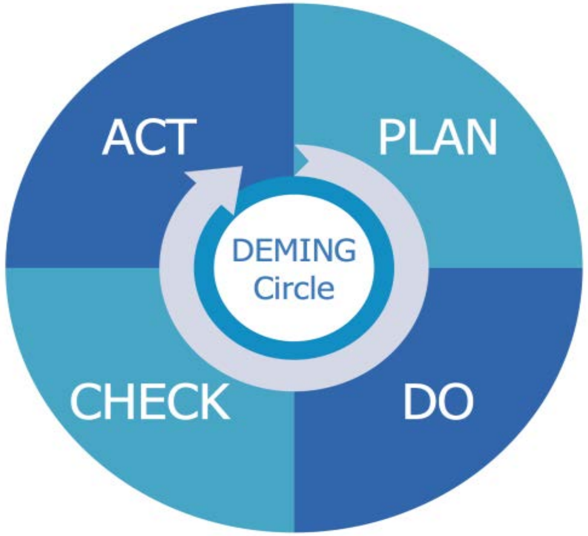
We also could combine variables by aggregating them. For example, suppose we have a production model with many products. A model with a single product having the characteristics of the average of all products would be one simplification.

This concept is very close to Deming-circle.

The **Deming Cycle** or PDCA Cycle is a four stage change management model used by companies for continuous improvement and problem solving

The four phases are:

- **Plan:** identify and analyze the problem or opportunity, develop hypotheses about what the issues may be, and decide which one to test.
- **Do:** test the potential solution, ideally on a small scale, and measure the results.
- **Check/Study:** study the result, measure effectiveness, and decide whether the hypothesis is supported or not.
- **Act:** if the solution was successful, implement it.



FOCUS PDCA is an improvement methodology of PDCA circle that many organizations use to guide their improvement efforts. It's simply a formalized process for solving different types of problems.
Now let's go through each one to understand its meaning.
***F*ind** a process or identify a problem that needs improvement. Problems are pretty easy to identify. Just think about the chronic complaints you get or those things that simply frustrate you at work.

***O*rganize** a team that understands or works with the process or problem. The team consists of people who know the process well and can speak to what works and what needs changing.

***C*larify** the knowledge. Clarifying the knowledge of the process can help to ensure there's agreement on what the real issues are. Every person who walks through the process or experiences the problem sees things from a little different perspective making it important to clarify the knowledge from every perspective.

***U*nderstand** the process variations. There are variations in every process. The trick is to discover what causes the variations so you can minimize the peaks and valleys.

***S*elect** a solution to test. Have the team determine what solution you'd like to test and create a goal for the improvement.

6.1.2 HISTORICAL ANALYSIS

A second strategy for external validation is **historical perspective**. Its basis is the judgment of people familiar with the problem and environment; the problem solver applies the model to historical data, and experts examine the model solution to see if it is reasonable. We could solve problem instances and extreme data sets to make sure that the expert recognize unusual situations. Ackoff (1962) proposed “murder boards,” a committee of people whose job is to find fault with the model. İf the results are not acceptable, we return to a previous step.

Sometimes we cannot model our problem exactly. One course of action then is to use the model and ignore the fact that it does not fit the problem. Forgetting the model and taking a seat-of-the-pants approach is another choice. Both courses, however, are dangerous. Using the model while realizing the solution is flawed is probably the better alternative because doing so provides additional insight, which is always welcome. Always remember, however, that the model is flawed.

A model that “passes” validation tests is not guaranteed to be a good model. There is no way to ensure a good model; we can only weed out the bad ones. Validation increases confidence in the model and hence the chance of using the results for the actual problem.

6.2 Solution strategy

The solution procedure depends on the model, and the model depends on available solution procedures. Having a great model that cannot be solved is frustrating. From the system approach, we know that it is better to get a partial solution to the whole problem than a complete solution to each part.

Solving a model can be as simple as plugging a few members into a single equation or as complicated as collecting a lot of data. Writing a matrix generator to put the data into proper formats, and solving a linear programming problem. Often, a computer is helpful- if not necessary- in solving problems.

Some models – for example, certain simulation models- only evaluate a given alternative. Generating alternatives is the problem solver’s responsibility. But can we generate a good a good alternative to evaluate? Usually we cannot guarantee that a single alternative generated is good, so, intuitively, the more alternatives we generate, the better our chance of finding a good one is. If it is expensive to evaluate them and they are easily generated, then we should look at many alternatives. Conversely, if it is hard to generate alternatives and costly to evaluate them, we cannot examine too many. Experimental design methods can help us trade off the expense of generating alternatives with the quality of the solution.

Other models can be solved by algorithms that generate alternatives as well as evaluate them. **Optimal**, or exact, algorithm look implicitly at all alternatives and choose the best; a critical path algorithm is an example. Other algorithms, called **heuristic** algorithms, generate alternatives that are not guaranteed to be optimal, but are, hopefully, close to the optimal solution. Choosing project with the largest return until the budget is used up is an illustration of a heuristic algorithm.

Even if a model can theoretically be solved optimally, a solution may not be possible in practice. Total enumeration soles integer-programming problems, but for reasonably sized problems, it will take centuries on the fastest computer. To get solutions in a reasonable amount of time we may have to use heuristic algorithms for these problems.

Heuristic algorithms provide uncertain solutions; because they do not guarantee optimally, we do not know the quality, i.e., how good this solution is. Some heuristic algorithms have performance guarantees, in which, for any set of data, the algorithm guarantees a solution within a given percentage of the optimal solution. We also can generate bounds on the optimal solution for a particular data set. These bounds give an indication of how far from optimal the heuristic solution is. İf the heuristic solution has a profit of $1000 and an upper bound on the profit is $1010, the heuristic solution is close enough. İf the bound is $2000, we are uncertain about the quality of the heuristic solution. The solution could be far from optimal, or we could have a loose bound.

Even optimal algorithms have uncertainty. How well the model fits reality is one source of uncertainty, and data are another. If data estimates are not exact, what does an optimal solution mean? Data uncertainty may be greater than the uncertainty of a heuristic solution, so a heuristic algorithm may be as good as an optimal algorithm.

After we validate the model and determine an appropriate algorithm, we solve the model. The next step is to interpret the model solution.

7 INTERPRET THE SOLUTION

Model solutions are not necessarily problem solutions. If the model is a good representation for the problem, it may lead to the solution of the problem. Because it is difficult to include all interactions in a model and still solve it, be careful in applying the model solution to the actual problem. Together, the problem solver and problem owner must translate the model solution into a useful solution to the problem. Examining the stability of the solution to data inaccuracies would be discussed now. Also, they must determine the effects of the assumptions made earlier on the solution. The judgment of both solver and owner plays an important role here. Again, we may need to return to a previous step to change or refine the process. If not, we have a problem solution.

Check this solution for reasonableness. Give the solution to the people who will use it, and get their comments. İf they think it is reasonable and would use the solution, we can have more confidence in our model and solution. İf they believe the solution is unreasonable, , the solution is not necessarily wrong, but their opinion indicates that there is more work do.

Because the data used in the model may be inaccurate; we may question whether the solution will change as the data changes. İdeally, however, The solution does not change significantly with small changes in the data; such a solution is **robust**. If we solve the model by linear programming, we can use sensitivity analysis to resolve this question. Other models have similar ways to determine the sensitivity of the solution to the data. If such methods do not exist, we can solve the model multiple times with slightly different data sets and examine the solutions. If results at this step are not satisfactory, we need to modify the model and repeat the problem solving process.

Robustness of assumptions also should be examined. Suppose we assumed a linear relationship for costs but were unsure about it. We might solve the model with the linear cost equation replaced by one having nonlinear costs. If the solutions to the two models are similar, our linearly assumptions is robust. On the other hand, if the solutions are very different, the assumption is invalid. We must determine the true nature of the cost relationship so that the model more correctly represents the system.
Assuming we can derive the problem solution from the model solution, we are now ready for implementation.

8 IMPLEMENTATION

Once a problem solution is found, implementation begins. All concerned must accept the solution. Necessary resources are committed and appropriate people are trained to carry out the new solution. It is good practice to have parallel operation of the new and old methods. Finally, the progress of the new solution must be monitored to make sure nothing goes wrong.

To implement a solution, it is usually necessary to change some system. İt is difficult to overcome the inertia of a well-established system.

People are notoriously resistant to change and accept change only when its need is clear and positive and they help plan the change. Unsatisfied owners will not accept a solution, so continually involving owners in the problem-solving process makes them more likely to understand and accept the final results.

Present your results with the enthusiasm of a salesperson, use simple language and give a clear statement of importance of the problem, and use facts to back up your position. Present both the positive and negative aspects of your solution, stress key points, and try to anticipate questions. If possible, give several alternative solutions covering a range of benefits and costs.

9 **Software**

Many software packages aid problem solving and some problems cannot be solved without a computer. Spreadsheets (Excel, Lotus) are easy to use and very helpful in problem solving.

For more mathematical problems, Computational algebra packages (Mathcad, Matlab, Mathematica) sometimes called electronic chalkboards, are useful. Statistical Analysis Packages (SAS, Minitab, SPSS) can extract information from data. . Optimization solvers (Lindo, CPLEX, OSL) are readily available and provide optimal solutions to many models. Modeling languages (AMPL, GAMS) make creating optimization models easier, but at the expense of learning them simulation languages (GPSS, SIMAN) and packages (promodel, witness) can help provide insight into many problems.

There are also a number of software packages tailored especially to Production Planning & Control courses (STORM) and so on.