

## CHAPTER

# 7 Structuring Requirement: Logic Modeling

## LEARNING OUTCOMES

By the end of this chapter, you should be able to:

1. Explain the use of Logic Modeling;
2. Use structured English, decision tables and decision trees to analyse, describe and document structured decisions;
3. Describe the guidelines in choosing appropriate logic modeling techniques for different situations; and
4. Explain software development strategies.

## INTRODUCTION

As good as data flow diagrams, they are not very good at showing the logic inside the processes. This is because data flow diagrams are not really designed to show the detailed logic of processes, you must model process logic using logic modeling techniques.

This topic introduces you to techniques used for modeling process decision logic. Logic modeling involves representing internal structure and functionality of processes depicted on a DFD. Logic modeling can also be used to show when processes on a DFD occur. Processes must be clearly described before they can be translated into a programming language. In this topic you will be introduced to techniques that you can use during the analysis phase to model the logic within processes: Structured English, Decision Tables and Decision Tree. This topic concludes with explanation on development strategies in creating new software which are by: Custom software, purchase commercial off-the-shelf software package and outsourced to an application service provider.

### 7.1 LOGIC MODELING TECHNIQUES

Modeling a system's logic is part of requirement structuring, just as was representing the system with data flow diagrams. Here our focus is on the processes pictured on the data flow diagrams and the logic contained within each process. Figure 7.1 shows the three goals of producing logic modeling.

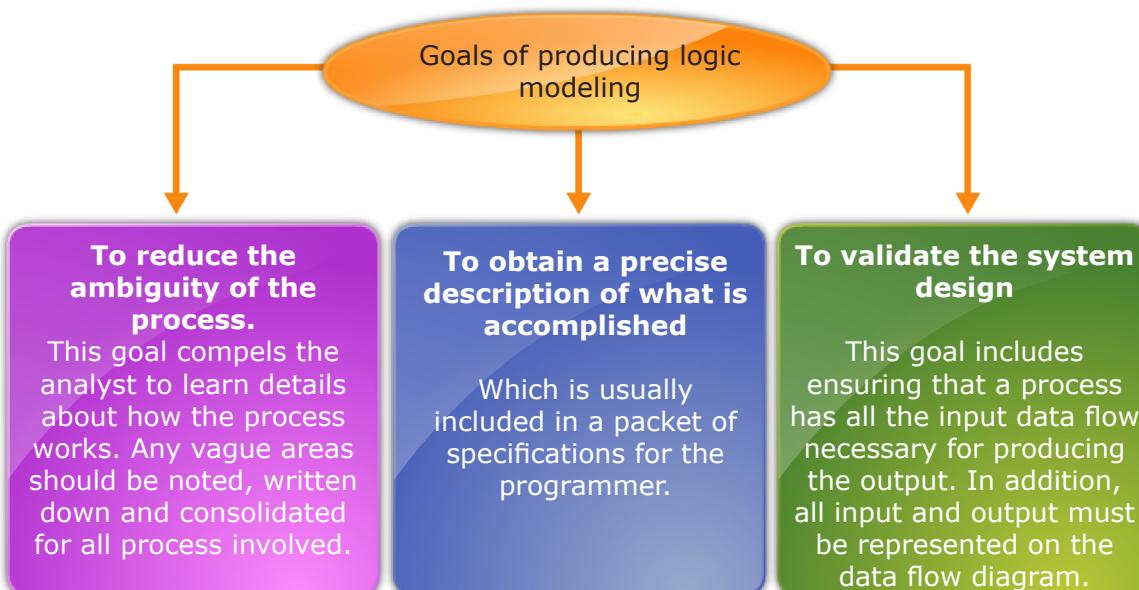


Figure 7.2: Organisation of the Classification Essay

In the next section, we will learn in detail how to model the logic within processes using several techniques that are Structured English, Decision Tables and Decision Tree.

### 7.1.1

### Structured English

Structured English is a modified form of English language used to specify the logic of information system processes.

The language structure is simple and clear. It describes the process logic precisely in sequence. Normally, Structured English uses:

- Statements in the form of sequence, selection and repetition.
- Indentation, which provides the structure, so that it is easy to read and understand.
- Use a limited vocabulary, including standard terms used in the data dictionary and specific words that describe the processing rules.

Structured English might look familiar to programming students because it resembles pseudo-codes, which are used in program design. Although the techniques are similar, the primary purpose of structured English is to describe the underlying business logic, while programmers, who are concerned with coding, mainly use pseudo codes as a shorthand notation for the actual code.

Examples of the Structured English are shown in Figure 7.2. After you study the sales promotion policy, notice that the structured English version describes the processing logic that the system must apply. Following these structured English rules ensures that your process descriptions are understandable to users who must confirm that the process is correct, as well as to other analysts and programmers who must design the information system from your descriptions.

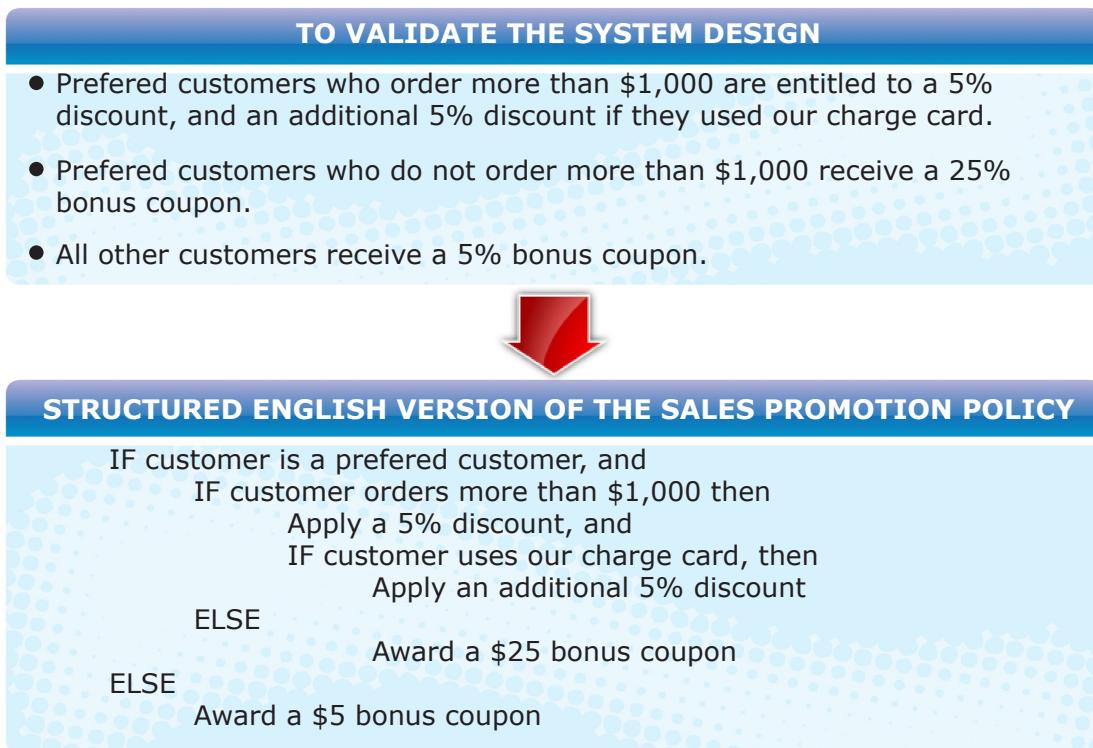


Figure 7.2: Sample of policy and structured English version of policy  
Source: Adapted from Shelly et al (2006)

Let's look at another example of how structured English would represent the logic of some of the processes identified in the current logical Inventory Control System as shown in Figure 7.3.

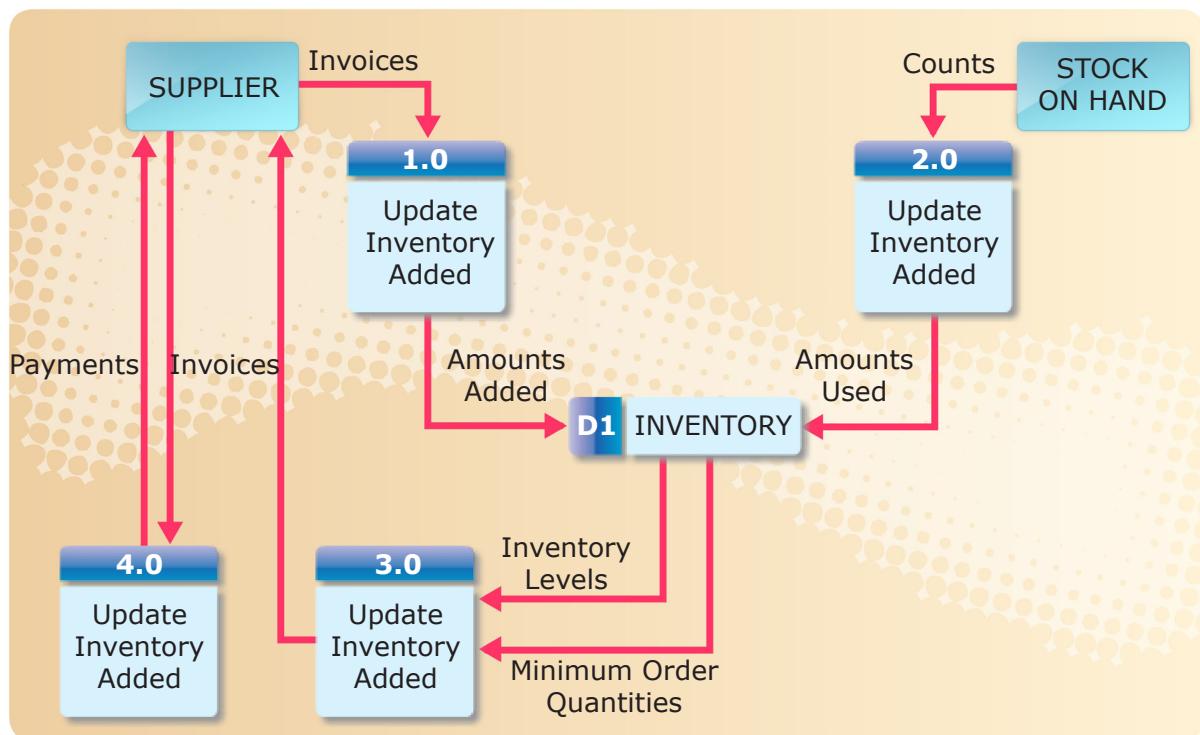


Figure 7.3: DFD for Inventory Control System  
 Source: Adapted from Hoffer et al (2008)

#### Process 1.0: Update Inventory Added

```

DO
  READ next Invoice-item-record
  FIND matching Inventory-record
  ADD Quantity-added from Invoice-item-record to Quantity-in-stock on
    Inventory-record
UNTIL End-of-file
  
```

#### Process 1.0: Update Inventory Added

```

DO
  READ next Stock-item-record
  FIND matching Inventory-record
  SUBTRACT Quantity-used on Stock-item-record from Quantity-in-stock on
    Inventory-record
UNTIL End-of-file
  
```

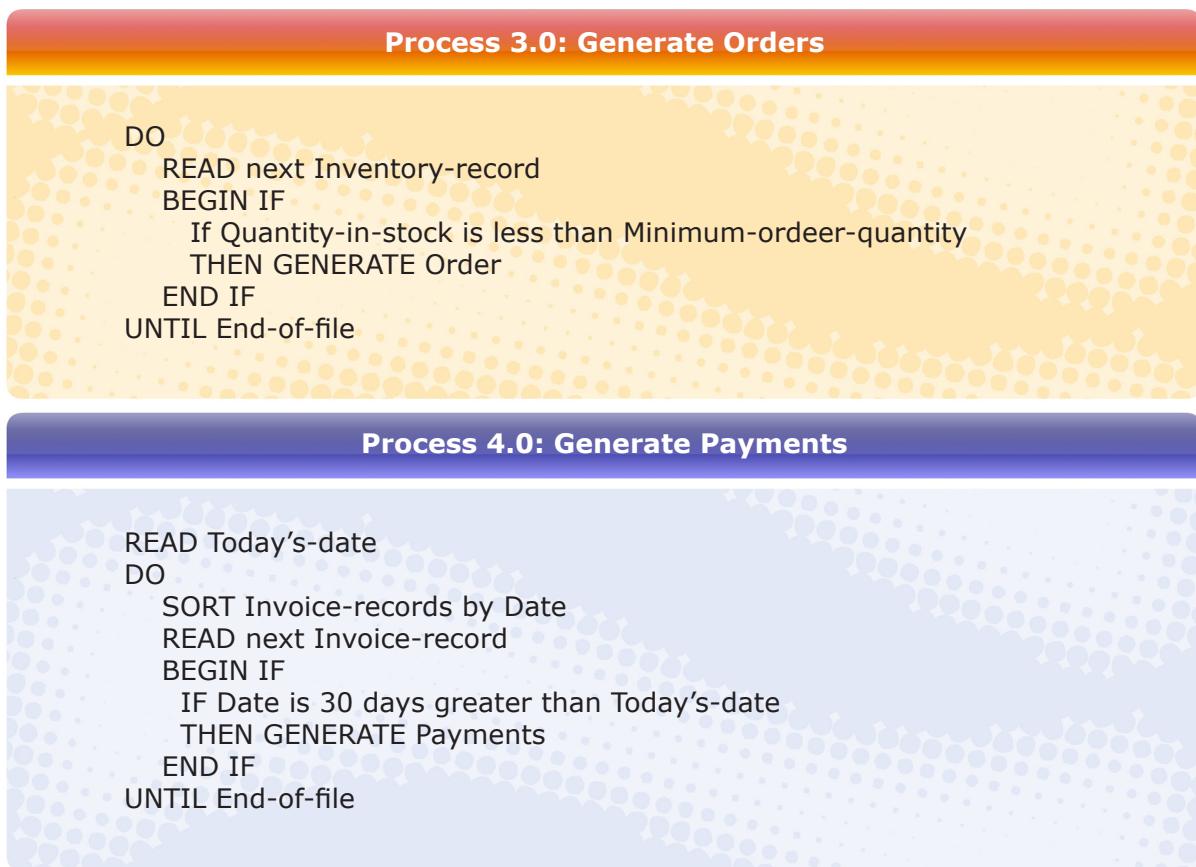


Figure 7.4: Structured English representations of four processes depicted in Figure 7.3

Source: Adapted from Hoffer et al (2008)

Besides the obvious advantage of clarifying the logic and relationships found in human languages, Structured English has another important advantage: It is a communication tool. Structured English can be taught to and hence understood by users in the organisation, so if communication is important, Structured English is a viable alternative for decision analysis.

### 7.1.2

### Decision Table

**Decision Table** shows a logical structure consisting of a combination of all process conditions and actions. Analysts often use decision table, in addition to structured English, to describe a logical process and ensure that they have not overlooked any logical possibility.

In certain situations, it provides the logic in a form that is shorter when compared with Structured English. A decision table is more suitable for representing very complex logic. It is sometimes used to check the accuracy of logic that is written in Structured English.

A decision table is a table of rows and columns, separated into four quadrants, as shown in Table 7.1. The upper left quadrant contains the condition(s); the upper right quadrant contains the condition alternatives. The lower half of the table contains the actions to be taken on the left and the rules for executing the actions on the right.

*Table 7.1: Standard Format Used for Presenting a Decision Table*

Conditions and Actions		Rules	
Conditions		Condition Alternatives	
Actions		Action Entries	

Figure 7.5 shows a decision table for Sales Promotion Policy that is equivalent to Figure 7.2. The company needs to categorise its customers before deciding what kind of discount to give. Here three condition exist; was the customer a preferred customer, did the customer order more than 1,000 and did the customer use our charge card? Based on these three conditions, four possible actions can occur, as shown in the Figure 7.5.

	1	2	3	4	5	6	7	8
CONDITION	Preferred customer	Y	Y	Y	Y	N	N	N
	Order more than \$1,000	Y	Y	N	N	Y	Y	N
	Used our charge card	Y	N	Y	N	Y	N	Y
ACTION	5% discount	X	X					
	Additional 5% discount	X						
	\$25 bonus coupon			X				
	\$5 bonus coupon	X	X	X				

*Figure 7.5: Sample decision table for Sales Promotion Policy  
Source: Adapted from Shelly et al (2006)*

Figure 7.6 shows the steps can be adopted in building a decision table:

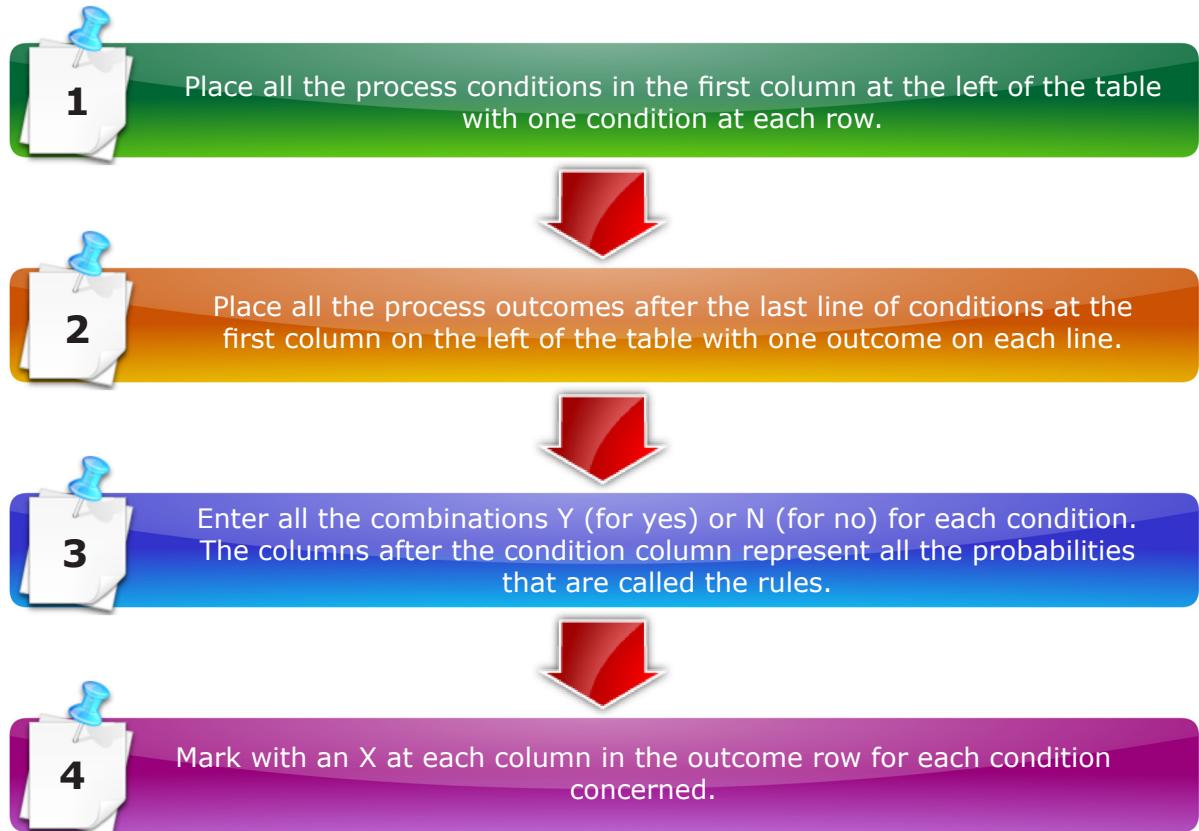


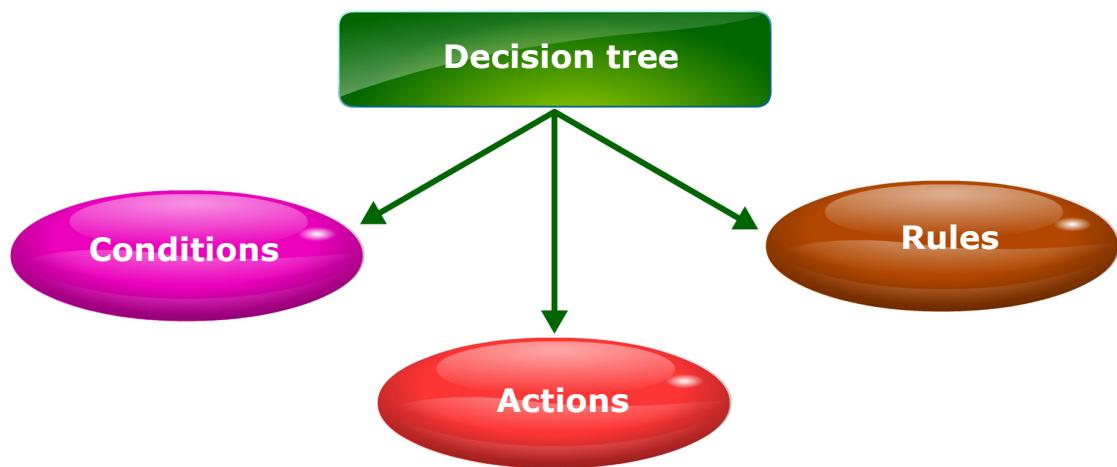
Figure 7.6: Steps can be adopted in building a decision table

Decision tables are an important technique in the analysis of structured decisions. One major advantage of using decision tables over other methods is that tables help the analyst ensure completeness. When using decision tables, it is also easy to check for possible errors, such as impossible situations, contradictions and redundancy.

### 7.1.3

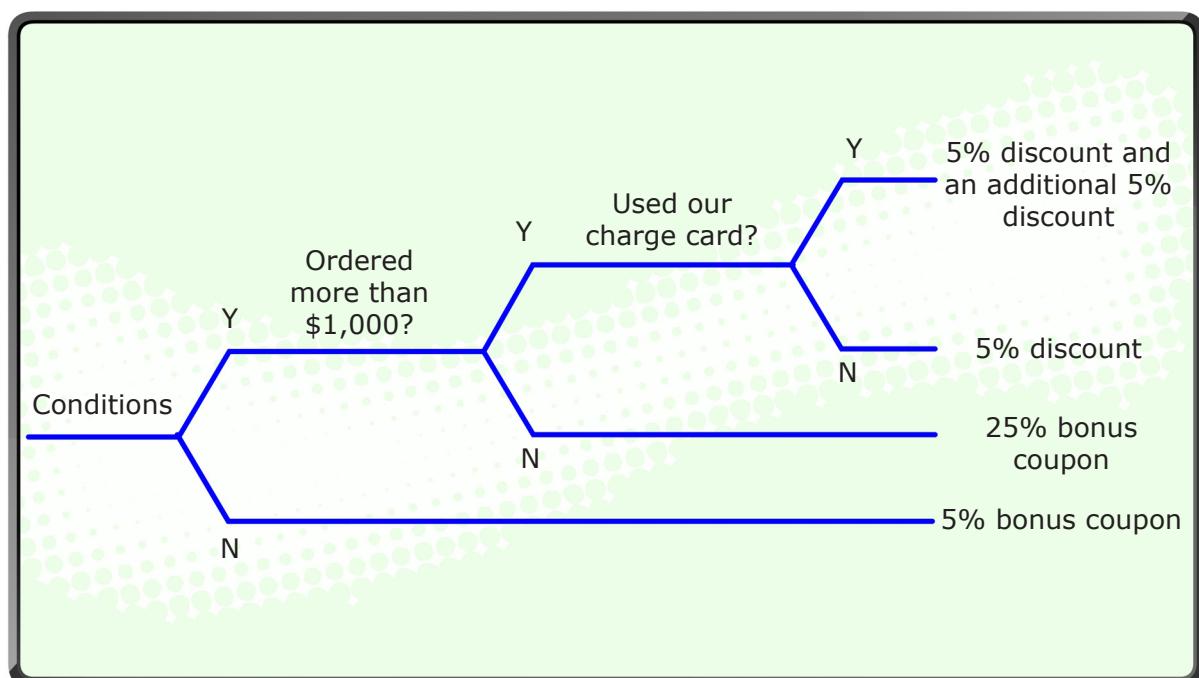
### Decision Tree

Decision Tree is quite similar to the decision table. Both represent the same elements (condition, action and rules) but in different forms - one in the table form, while the other in graphic form.



Decision tree is a graphical representation of the conditions, actions, and rules found in a decision table. It shows a logical structure, which appears like a tree. Beginning with root and stem on the left, it expands into branches and leaves towards the right.

Figure 7.7 shows the same Sales Promotion Policy in a decision tree format which is equivalent to the decision table in Figure 7.5. A decision tree is read from left to right, with the conditions along the various branches and the actions at the far right.



*Figure 7.7: Sample decision tree for Sales Promotion Policy*  
*Source: Adapted from Shelly et al (2006)*

Whether to use a decision table or a decision tree often is a matter of personal preference. A decision table might be a better way to handle complex combinations of conditions. On the other hand, a decision tree is an effective way to describe a relatively simple process.



What is one of the major advantages of decision tables over other methods of decision analysis?

#### 7.1.4

#### Choosing a Technique

We have examined the three techniques for modeling the logic in the process as shown in Figure 7.8. Although they need not be used exclusively, it is customary to choose one analysis technique for a decision rather than employing all three. The following guidelines provide you with a way to choose one of the three techniques for a particular case:

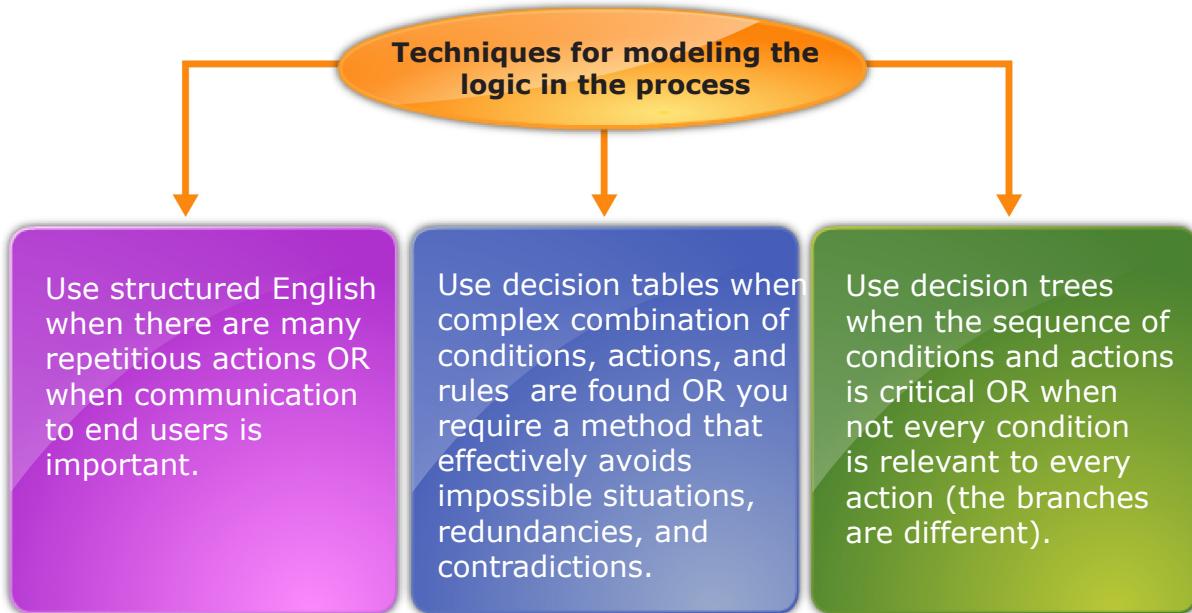


Figure 7.8: Three techniques for modeling the logic in the process

**7.2****DEVELOPMENT STRATEGIES**

Analysts and organisations are increasingly faced with a make, buy or outsource decision when assessing development strategies for information systems project. As an analyst, part of the expertise you are developing is to make sound judgments regarding developing software versus purchase of software or outsourcing for new and existing system. Each alternative comes with its own strengths and weaknesses. Therefore, one alternative is better than others in different situations.

**7.2.1****Create Custom Software**

Custom Software is also known as in-house software, is a type of software that is developed from the beginning for a specific organisation or function. Most of the project groups consider custom development as the best approach to develop a system because the team has complete control over the functions and shape of the system.

There are several situations that call for the custom development of a new software or software components. The most likely instance is when commercial off-the-shelf (COTS) software does not exist or cannot be identified for desired application. Alternatively, the software may exist but is unaffordable or cannot easily be purchased or licensed.

Custom software development should be done when the organisation is attempting to gain a competitive advantage through the leveraged use of information systems. This is often the case when an organisation is creating ecommerce or other innovative applications where none existed. It is also possible that the organisation is a “first pioneer” in the use of a particular technology or in its particular industry. Organisations that have highly specialised requirements or exist in niche industries can also benefit from custom development. Table 7.2 summarises the advantages and disadvantages of custom development.

*Table 7.2 Advantages and Disadvantages of Custom Development*

Advantages	Disadvantages
<ul style="list-style-type: none"><li>● Specific response to specialised business needs.</li><li>● Innovation may give firm a competitive advantage.</li><li>● In-house staff available to maintain software.</li><li>● Pride of ownership.</li></ul>	<ul style="list-style-type: none"><li>● May be significantly higher initial cost compared to COTS software or Application Service Provider.</li><li>● Necessity of hiring or working with a development team.</li><li>● Ongoing maintenance.</li></ul>

### 7.2.2

### Purchasing COTS Packages

Commercial Off-The-Shelf (COTS) software is a ready-made software product that can be easily obtained. COTS include such products as Microsoft Office suite which includes Word for word processing, Excel for spreadsheets, Access for building databases and other applications. Other types of COTS software from Enterprise Resource Planning (ERP) packages such as Oracle and SAP.

Consider using COTS software when you can easily integrate the applications or packages into existing or planned systems, and when you have identified no necessity to immediately or continuously change or customise them for users. There are some advantages to purchasing COTS software that you should keep in mind as you weigh alternative. One advantage is that these products have been refined through the process of commercial use and distribution, so that often there are additional functionalities offered. Another advantage is that packaged software is typically extensively tested, and thus extremely reliable. Table 7.3 summarises the advantages and disadvantages of purchasing COTS packages.

*Table 7.3 Advantages and disadvantages of purchasing COTS packages*

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>● Refined in the commercial world.</li> <li>● Increased reliability.</li> <li>● Increased functionality.</li> <li>● Often lower initial cost.</li> <li>● Already in use by other firms.</li> <li>● Help and training comes with software.</li> </ul>	<ul style="list-style-type: none"> <li>● Programming focused; not business focused.</li> <li>● Must live with the existing features.</li> <li>● Limited customisation.</li> <li>● Uncertain financial future of vendor.</li> <li>● Less ownership and commitment.</li> </ul>

Most of the COTS packages allow customisation or manipulation of system parameters for changing certain functions. These changes are good in creating functions that are needed, but the said functions might not be present inside the software package.

System Integration refers to the process of developing a new system by integrating or combining a software package with an existing legacy system. Many companies are skilful in system integration. It is not impossible for a company to select COTS packages, and then to outsource the job of integrating various packages to the consulting companies.

### 7.2.3

### Outsourcing

**Outsourcing** is contracting goods or services with another company or person to do a particular function such as to pay an external supplier to develop or to provide services for creating a system. This third option is to outsource some of the organisation's software needs to an Application Service Provider (ASP) that specialises in IT applications. Almost every organisation outsources in some way. Typically, the function being outsourced is considered non-core to the business.

There are specific benefits to outsourcing applications to an ASP. For example organisations that desire to retain their strategic focus and do what they are best at, may want to outsource the production of information systems applications. Additionally outsourcing one's software needs means that the organisation doing the outsourcing may be able to sidestep the need to hire, train and retain a large IT staff. This can result in significant saving. Table 7.4 shows the summarises the advantages and disadvantages of outsourcing.

Table 7.4 Advantages and Disadvantages of Custom Development

Advantages	Disadvantages
<ul style="list-style-type: none"><li>Organisations that do not specialise in information systems can focus on what they do best (their strategic mission).</li><li>There is no need to hire, train or retain a large IT staff.</li><li>There is no expenditure of employee time on nonessential IT tasks.</li></ul>	<ul style="list-style-type: none"><li>Loss of control of data, systems, IT employees and schedules.</li><li>Concern over the financial viability and long-run stability of the ASP.</li><li>Security, confidentiality and privacy concerns.</li><li>Loss of potential strategic corporate advantage regarding innovativeness of applications.</li></ul>

Some believe that the heart of their business is their information, so the thought of relinquishing control over it is distressing. Thus, there are a lot of risks if outsourcing is chosen. To reduce the risks, the following steps need to be taken:

- Identify your project requirements completely. Do not outsource something that you do not understand at all.
- Choose your suppliers and developers carefully, and check their service records to see proofs of the system and the technology that you need.



What is the difference between outsourcing and COTS software?

## 7.2.4

**Evaluation of Software Guidelines**

Whether you purchase a COTS package or contract for ASP services, you will be dealing with vendors or suppliers who may have their own best interests at heart. You must be willing to evaluate software with users and not be unduly influenced by vendors' sales pitches. Specifically there are six main categories on which to grade software as shown in Table 7.5.

*Table 7.5: Guidelines for Evaluating Software*

Source: Adapted from Hoffer et al (2008)

Software Requirements	Specific Software Features
<b>Performance Effectiveness</b>	<ul style="list-style-type: none"> <li>• Able to perform all required tasks</li> <li>• Able to perform all tasks desired</li> <li>• Well-designed display screens</li> <li>• Adequate capacity</li> </ul>
<b>Performance Efficiency</b>	<ul style="list-style-type: none"> <li>• Fast response time</li> <li>• Efficient input</li> <li>• Efficient output</li> <li>• Efficient storage of data</li> <li>• Efficient backup</li> </ul>
<b>Ease of use</b>	<ul style="list-style-type: none"> <li>• Satisfactory user interface</li> <li>• Help menus available</li> <li>• "Read Me" files for last minute changes</li> <li>• Flexible interface</li> <li>• Adequate feedback</li> <li>• Good error recovery</li> </ul>
<b>Flexibility</b>	<ul style="list-style-type: none"> <li>• Options for input</li> <li>• Options for output</li> <li>• Usable with other software</li> </ul>
<b>Quality of Documentation</b>	<ul style="list-style-type: none"> <li>• Good organisation</li> <li>• Adequate online tutorial</li> <li>• Web site with FAQ</li> </ul>
<b>Manufacturer Support</b>	<ul style="list-style-type: none"> <li>• Technical support hot line</li> <li>• Newsletter/email</li> <li>• Web site with downloadable product updates</li> </ul>

Evaluated packaged software based on a demonstration with test data from the business considering it and an examination of accompanying documentation. Vendors' descriptions alone will not suffice. Vendors typically certify that their software is working when it leaves their supply house, but they will not guarantee that it will be error-free in every instance or that it will not crash when incorrect actions taken are by users. Obviously, they will not guarantee their packaged software if used in conjunction with faulty hardware.

## SUMMARY

- Various techniques are available for modeling the decision logic information system process. One method is to use Structured English as a technique for representing steps in logical processes in data flow diagrams. Another is to use decision tables to represent the logic of choice in conditional statements. Decision tree can also be used to represent decision table especially to describe a relatively simple process. There is no single best technique to use for structuring requirements, an analyst must be proficient at techniques used which required by different situations.
- Part of the expertise to be a system analyst is to make sound judgments on development strategies. There are three approaches in new software development. Software can be created as a custom product, purchased commercially off-the-shelf software package or outsourced to an application service provider. Each alternative comes with its own strengths and weaknesses. Therefore, one alternative is better than others in different situations.

## KEY TERMS

Commercial Off-The-Shelf (COTS)

Custom Software

Decision Table

Decision Tree

Logic Modeling

Outsourcing

Structured English

System Integration

## REFERENCES

1. Shelly, G., T. Cashman and H. Rosenblatt. (2006) System Analysis and Design, 6th ed. Cambridge, MA: Course Technology
2. Hoffer, J.A., George, J.F. and Valacich (2008) 5th ed., Modern Systems Analysis and Design, Benjamin/Cummings, Massachusetts.
3. Yourdon, E. 1989. Managing Structured Techniques, Prentice Hall, Upper Saddle River, New Jersey