CHAPTER
SIXTEEN

ARTIFICIAL INTELLIGENCE AND PLC SYSTEMS

Computers can figure out all kinds of problems, except the things in the world that just don't add up.

—James Magary

Chapter Highlights

In previous chapters, we highlighted both simple and complex PLC applications. In this chapter, we will present an area of PLC applications that goes one step beyond these—artificial intelligence. We will explain the basics of artificial intelligence (AI) systems by explaining their organization and methodology. We will also discuss how each of the three types of artificial intelligence systems—diagnostic, knowledge, and expert—work. Finally, we will present an example of an AI application to further explain how these complex systems operate. After you finish learning about artificial intelligence, you will be ready to explore fuzzy logic, another advanced application that involves PLCs.

16-1 Introduction to AI Systems

Artificial intelligence (AI) is an area of computer science that has been around for some time. In fact, the conceptual design of AI was first developed in the early 1960s. The definition of artificial intelligence varies among people in the computer industry, making the concept somewhat difficult to perceive and understand. In general, AI can be defined as the subfield of computer science that encompasses the creation of computer programs to solve tasks requiring extensive knowledge.

The software programs that form an AI system are developed using the knowledge of an expert person (or persons) in the field where the system will be applied. For instance, a food-processing AI system that involves the making and packaging of a food product will consist of knowledge obtained from chemists, food technologists, packaging experts, maintenance personnel, and others closely associated with the operation.

In this chapter, we will present AI techniques that can be implemented through a PLC-based process control system. These techniques will define the methods for implementing AI into the process. The result will be a system that can successfully diagnose, control, and predict outcomes based on resident knowledge and program sophistication.

16-2 Types of AI Systems

An exact classification of the types of artificial intelligence systems is very difficult to obtain because of the varying definitions of AI applications. For the purposes of this text, however, we will divide artificial intelligence into three types of systems:

- diagnostic
- knowledge
- expert

Each of these types of AI systems have similar characteristics, and in fact, the systems evolve sequentially. As the systems become more sophisticated, the size of the database grows and the extent of how the process data is compiled and interpreted increases.

Diagnostic Systems

Diagnostic AI systems are the lowest level of artificial intelligence implementation. These systems primarily detect faults within an application, but they do not try to solve them. For example, a diagnostic system can diagnose a pump fault by detecting a loss of tank pressure or by reading flow meter values.

A diagnostic system reaches a fault conclusion through inferring techniques based on known facts (knowledge) introduced into its detection system. This type of AI is used in applications that have a small knowledge and database structure. Diagnostic systems typically make GO or NO GO decisions and sometimes provide information about the fault's probable cause.

Knowledge Systems

A **knowledge AI system** is, in reality, an enhanced diagnostic system. Knowledge systems not only detect faults and process behaviors based on resident knowledge, but also make decisions about the process and/or the probable cause of a fault.

In the batching system example mentioned in the diagnostic system section, a knowledge system would go beyond just diagnosing the fault. It would also provide suggestions about probable faulty devices, as well as make a decision about whether to continue the process (if the fault is noncritical) or to shut down (if the fault is critical). The system bases these decisions on its programmed knowledge and a set of rules that defines each fault condition.

It is possible that the detection of a fault in the previous example could have been a false alarm. As part of its enhanced features, a knowledge system checks whether the elements signaling the fault condition (i.e., flow meter, pressure transducer) are operating correctly. It then compares these observations (process feedback) with the procedures and measures based on this information. For example, if a fault does occur and it is a valid noncritical fault, the control system may issue *continue process, stop after finished*, and *alert personnel* commands.

Expert Systems

An **expert AI system** is the top of the line in AI-type applications; it has all of the capabilities of a knowledge system and more. An expert system provides an additional capability for examining process data using statistical analysis. The use of statistical data analysis lets the system predict outcomes based on current process assessments. The outcome prediction may be a decision to continue a process in spite of a fault detection.

For the example used in the other two types of AI systems, an expert system may decide to continue the batching operation until the noncritical fault generates another fault. The system might arrive at this decision because the average pressure sensed in the mixing reactor tank is within tolerance limits (i.e., readings observed about the mean). Thus, the system continues the batching operation in spite of the fact that the flow meter reported a loss of flow. The system then continues production and alerts personnel that pump and flow meter feedback may have been lost.

The knowledge introduced into an expert system is more complex than in the other types of AI systems; therefore, expert systems generate more data verification (feedback information). The decisions made by expert systems also require more sophisticated software programming, since their decision trees involve more options and attributes.

The implementation of an expert AI system requires not only extra programming effort but also more hardware capability. The total system will need more transducers to check other transducers and field devices. Moreover, the PLC will require the use of two or more processors to implement the control and intelligence programs. The speed of the system must also be fast so that it can operate in real time. Furthermore, the system's memory requirements will be larger, since knowledge data must be incorporated and stored into the AI system.

16-3 Organizational Structure of an AI System

A typical artificial intelligence system consists of three primary elements:

- a global database
- a knowledge database
- an inference engine

Figure 16-1 shows a block diagram of an AI system's architecture. As the figure illustrates, the AI system must receive its knowledge from a person who thoroughly understands the process or machine being controlled. This individual, called the *expert*, must communicate all information about system maintenance, fault causes, etc. to the *knowledge engineer*, the person responsible for system implementation. The process of gathering data from the expert and transmitting it to the knowledge engineer is known as *knowledge acquisition*.

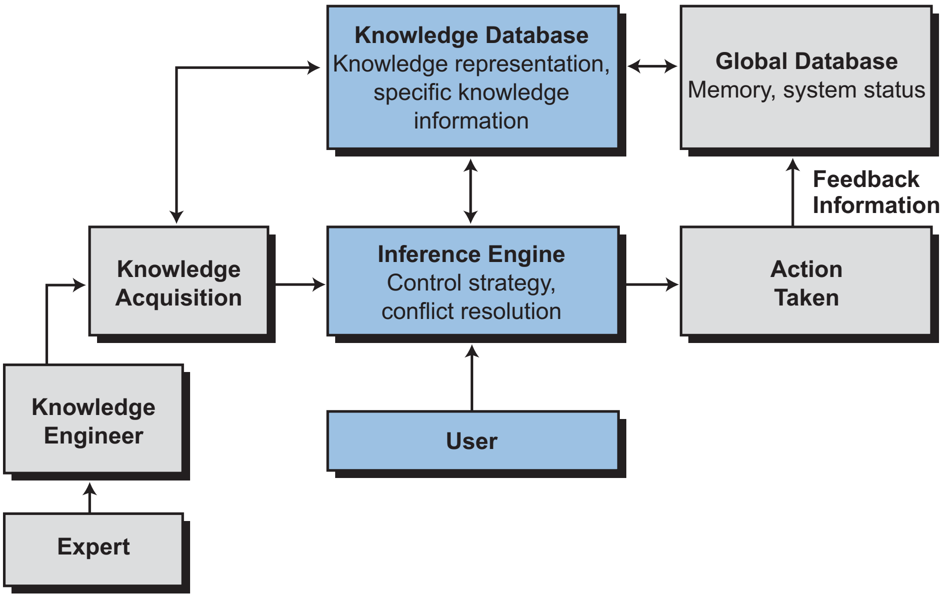

Figure 16-1. Artificial intelligence system architecture.

GLOBAL DATABASE

The **global database** section of an AI system contains all of the available information about the system being controlled. This information mainly deals with the input and output data flow from the process. The global database resembles a storage area where information about the process is stored and updated. The AI system can access the data in this area at any time to perform statistical analysis on historical process control data, which in turn can be used to implement AI decisions.

The global database resides in the memory of the control system implementing the artificial intelligence. If a PLC is used to implement a diagnostic AI system, the global database will most likely be located in the storage area of the PLC's data table. If a PLC is used in conjunction with a computer or computer module to implement an AI system, then the global database will probably be located in the computer, the computer module's memory, or a hard disk storage subsystem.

KNOWLEDGE DATABASE

The **knowledge database** section of an AI system stores the information extracted from the expert. Like the global database, this database includes information about the process; however, it also stores information about faults, along with their probable causes and possible solutions. Moreover, the knowledge database stores all of the rules governing the AI decisions to be made. The more involved the AI system, the larger the knowledge database.

Accordingly, the knowledge database of a diagnostic system is less complex than that of a knowledge system; likewise, the knowledge database of a knowledge system is less sophisticated than that of an expert system. The knowledge database is stored in the section of the system memory that implements the AI techniques.

INFERENCE ENGINE

An AI system's **inference engine** is the place where all decisions are made. This section uses the information stored in the knowledge database to arrive at a decision and then execute all applicable rules and decisions about the process. The inference engine also constantly interacts with the global database to examine and test real-time and historical data about the process.

The inference engine usually resides in the main CPU (i.e., the one that performs the AI computations). However, in a PLC-based system, the inference engine may or may not be stored in the main CPU, depending upon the system's complexity (i.e., diagnostic, knowledge, or expert).

16-4 KNOWLEDGE REPRESENTATION

Knowledge representation is the way the complete artificial intelligence system strategy is organized—that is, how the knowledge engineer represents the expert's input. This representation is stored in the knowledge database of the AI system. In rule-based knowledge representation, the expert's knowledge is transformed into IF and THEN/ELSE statements, which facilitate actions and decisions.

All control systems that implement artificial intelligence, whether diagnostic, knowledge, or expert, execute the control strategy (via the software control program) in the inference engine. Whenever a decision must be made due to a fault or another situation, the inference engine refers to the knowledge representation to obtain a decision about the probable cause. This decision is the result of a group of software subroutines. Once the knowledge database reaches an AI decision, the inference engine will determine the appropriate course of action. Depending on the control strategy formulation (main program), the inference engine may, at this time, refer to the global database to verify data or obtain more information.

RULE-BASED KNOWLEDGE REPRESENTATION

Rule-based knowledge representation defines how the expert's knowledge is used to make a decision. The rules used are either *antecedent* (IF something happens) or *consequent* (THEN take this action). For example, to the question, What causes the volume in the tank to drop?, the expert may respond with the answer, a malfunctioning tank system. The knowledge engineer may implement this information as the following rule: IF the volume is less than the set point, THEN annunciate a system malfunction due to a loss of volume.

Rules can be as long and complex as needed for the process, and they usually define the involvement of the AI system. For instance, a simple rule-based system (few rules, not very complex) may formulate a simple diagnostic rule, such as:

IF the temperature is less than the set point, THEN open the steam valve

A more complex diagnostic formula would involve rules that depend on parent rules:

```
IF case 1, THEN — ELSE nothing
               |
               IF case 2, THEN — ELSE something
                              |
                              IF case 3 — THEN nothing
                                          •
                                          •
                                          •
```

where each of the case conditions represents a particular measurement, comparison, or situation. Figure 16-2 illustrates a decision tree for forming AI rules.

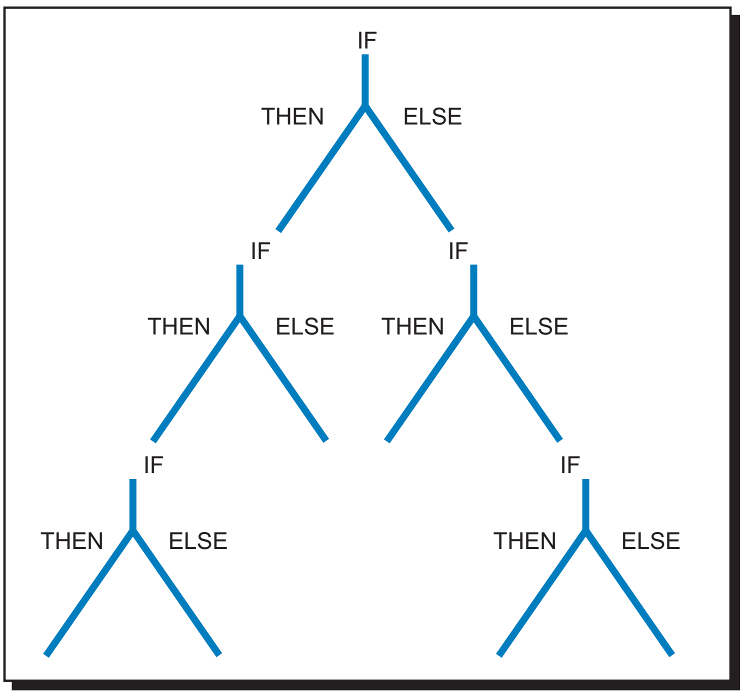

Figure 16-2. Decision tree.

A slightly different degree of complexity occurs in a rule-based knowledge representation when the rule has several probable causes. For example:

$$\text{IF volume drops, THEN} \left\{ \begin{array}{c} \text{valve failure} \\ \text{or} \\ \text{pump failure} \\ \text{or} \\ \text{feedback failure} \end{array} \right.$$

In this case, the consequents must be further investigated to arrive at a complete formal rule. The inference engine can use the consequents derived in the knowledge representation to obtain a better definition of the problem's cause. Knowledge and expert AI systems use this process to provide advanced decision-making capabilities.

EXAMPLE 16-1

A PLC-controlled box conveyor transports two sizes of boxes that are diverted to different palletizer operations according to their size. A solenoid activates the diverter that sorts the boxes. Write the rules that a knowledge database could use to detect a possible cause for the solenoid's malfunction.

SOLUTION

One of two factors can result in solenoid failure according to the situation presented: coil burnout or mechanical damage. The conditions and causes in Figure 16-3 describe these two possible factors that could lead to a fault.

Rule #1

Result	Condition	Cause
Burned-out coil	Excessive temperature is developed due to continuous high-current input	—Low line voltage causes failure to pull plunger —High ambient temperature —Mechanically blocked plunger —Operations too rapid

Rule #2

Result	Condition	Cause
Mechanical damage	Excessive force exerted on the plunger	—Overvoltage —Reduced load

Figure 16-3. Knowledge database rules for conveyor fault.

16-5 KNOWLEDGE INFERENCE

Knowledge inference is the methodology used for gathering and analyzing data to draw conclusions. Knowledge inference occurs in the inference engine during the execution of the main control strategy program. It also occurs in the knowledge database during the comparison and computation of rule solutions.

The system's software program determines the approach used to derive AI solutions. Operator interaction on control problems can enhance the solution-finding process. For example, if the system detects a failure due to a misreading in an inspection system, it may alert the operator to the problem and advise him/her of probable causes. Furthermore, the system may wait for the operator's input (e.g., check for laser intensity in the receiver side to determine if the laser beam is reflecting at the correct angle) and then use the operator's input to develop more intelligent solutions to the problem.

In small systems, knowledge inference occurs on a local basis. That is, the control system houses the resident software for the inference engine. In large, distributed, intelligent systems, knowledge inference often occurs at a main host in the hierarchical system.

Remember that the degree of AI involvement in the system will determine how much hardware is required (e.g., computer modules, powerful PLCs, small PLCs with personal computers, etc.). When all global databases are in constant network communication, allowing knowledge inference information to be passed from one controller to another, the intelligent system is said to have a *blackboard architectural structure*.

In all types of intelligent systems, certain methods of rule evaluation are used to implement knowledge inference. These methods include *forward chaining* and *backward chaining*. Intelligent systems also analyze statistical information as part of knowledge inferencing to obtain predictions about outcomes.

BLACKBOARD ARCHITECTURE

Large, complex, distributed control systems involve the interaction of several subsystems, which continuously communicate with each other either directly or over a local area network. When artificial intelligence is added to these large systems, system elements, such as knowledge inferencing and the global and knowledge databases, are distributed throughout the architecture of the control system. Whether or not each of the controllers in the network has a local inference engine, global database, and knowledge database depends on the degree of inferencing that occurs on a local basis. **Blackboard architecture** is the name given to this type of large system, which utilizes several subsystems containing local global and knowledge databases.

Figure 16-4 illustrates a blackboard configuration of an intelligent control system. The PLCs at the subsystem level may contain computer modules, which help them perform inference engine computations. The hierarchy of the control system allows the supervisory PLC controller to poll each of the subsystems and obtain all or part of their local global database information. The host computer element in this control structure holds the blackboard, the area that stores all of the information obtained from the subsystems by the supervisory PLC. The inference engine of the host element then implements the complex AI solution according to its knowledge inference about the total control system.

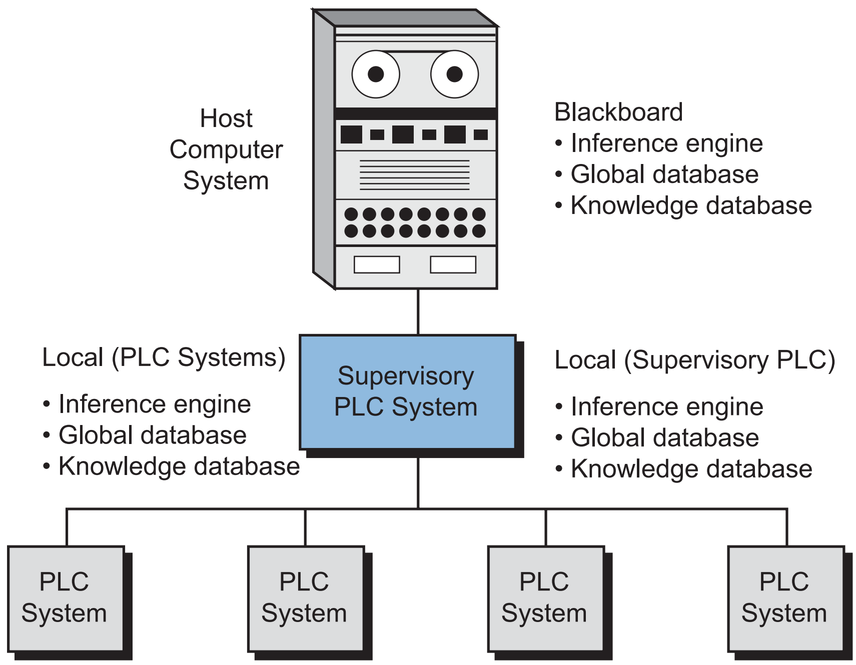

Figure 16-4. Example of blackboard architecture.

FORWARD CHAINING

Forward chaining is a method used to determine possible outcomes for given data inputs. Forward chaining inference engines typically receive process information via the global database and monitor specific inputs to the control system to determine the outcomes. For instance, in Example 16-1, forward chaining specifies the following consequences for a failed solenoid: a jammed conveyor or misplaced boxes in the two palletizers.

Two different types of fact searching occur within the forward chaining method: *depth first* and *breadth first*. Both searches deal with how the outcome is obtained. A depth-first search, shown in Figure 16-5, evaluates the rules that form the knowledge database (A, B, C, etc.) on a priority basis going *down* the tree. In the conveyor example mentioned earlier, when the control system detects the solenoid failure (A), it will evaluate a new rule to see if jamming has occurred (B). If the conveyor has jammed, then the system will evaluate the consequences that can occur (e.g., material inside box may break or material could spill (D)).

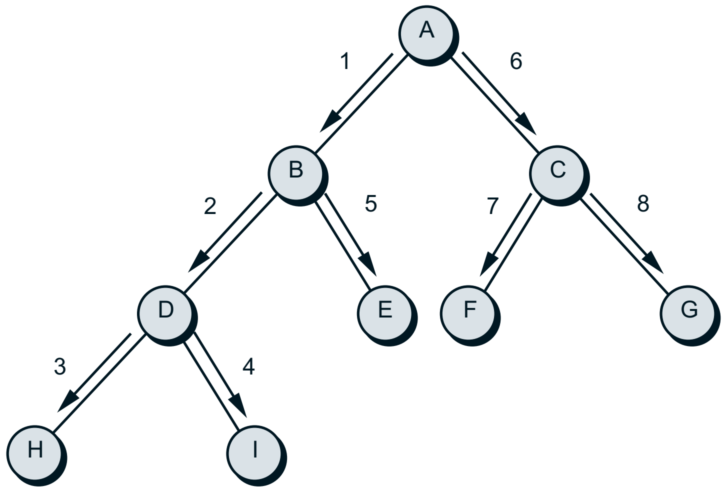

Figure 16-5. Forward chaining depth-first search.

In contrast, the breadth-first method evaluates each rule in the *same level* of the tree before proceeding to the next level down (see Figure 16-6). In our conveyor example, a breadth-first evaluation of the rules means that after the solenoid failure (A) the system will check for a possible jam (B), then it will check for palletizer misplacement (C), and so on.

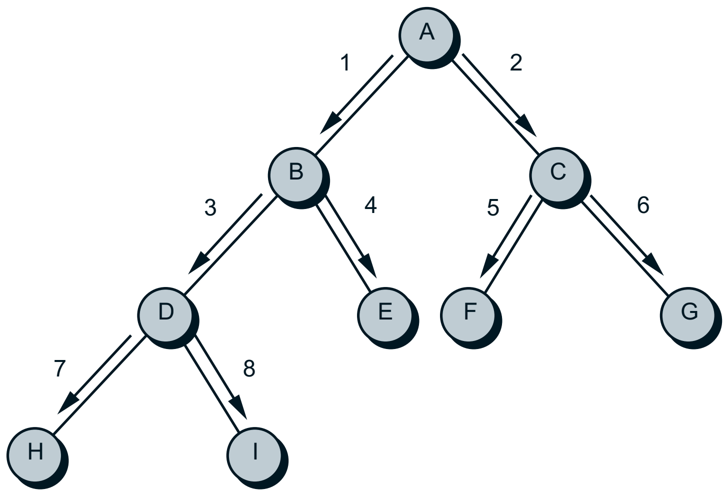

Figure 16-6. Forward chaining breadth-first search.

BACKWARD CHAINING

Backward chaining is a method for finding the causes of an outcome. Referring to Example 16-1, the rule tables present backward chaining information—that is, causes for the solenoid failure outcome. Basically, backward chaining analyzes the consequences to obtain the antecedents.

Similar to forward chaining, backward chaining uses both the depth-first and breadth-first search methods. In our conveyor example, after the solenoid failure occurs, a backward chaining depth-first search will first check one condition rule then check each possible cause of that condition. On the other hand, a breadth-first search will first examine both of the condition rules and then obtain the causes for each of the conditions.

STATISTICAL AND PROBABILITY ANALYSIS

Statistical analysis and probability play a large role in artificial intelligence systems. These aspects of AI are particularly important in expert systems, which predict outcomes. The system's global database stores the process information that will be used in the AI statistical analysis.

In Chapter 13, we explained how to interpret and obtain statistical data, such as the mean, mode, median, and standard deviation. These statistical computations help determine a future outcome based on what is happening in the current process. Decisions based on statistics can be related to the consequences of the rules described in the knowledge representation. For example, just because a system detects an error fault does not mean that the fault actually occurred, even though the feedback data transducer devices may be operating correctly. Using statistical analysis, the inference engine may decide not to advise personnel or apply the corresponding control to the fault, but instead to continue monitoring the situation more closely.

EXAMPLE 16-2

A control system monitors and controls a cooker in a temperature loop with specifications as shown in Figure 16-7. Indicate how AI can be added to the system to detect real temperature problems. Also, indicate how the system can screen out false temperature faults.

SOLUTION

Figure 16-7a shows a profile of temperature readings from the system. The PLC can monitor and accumulate temperature data continuously from time t_0 to time t_1 using FIFO instructions, storing this data in a storage area with a fixed number of registers (see Figure 16-7b). The program can also compute the mean, median, and standard deviation of the current temperature readings.

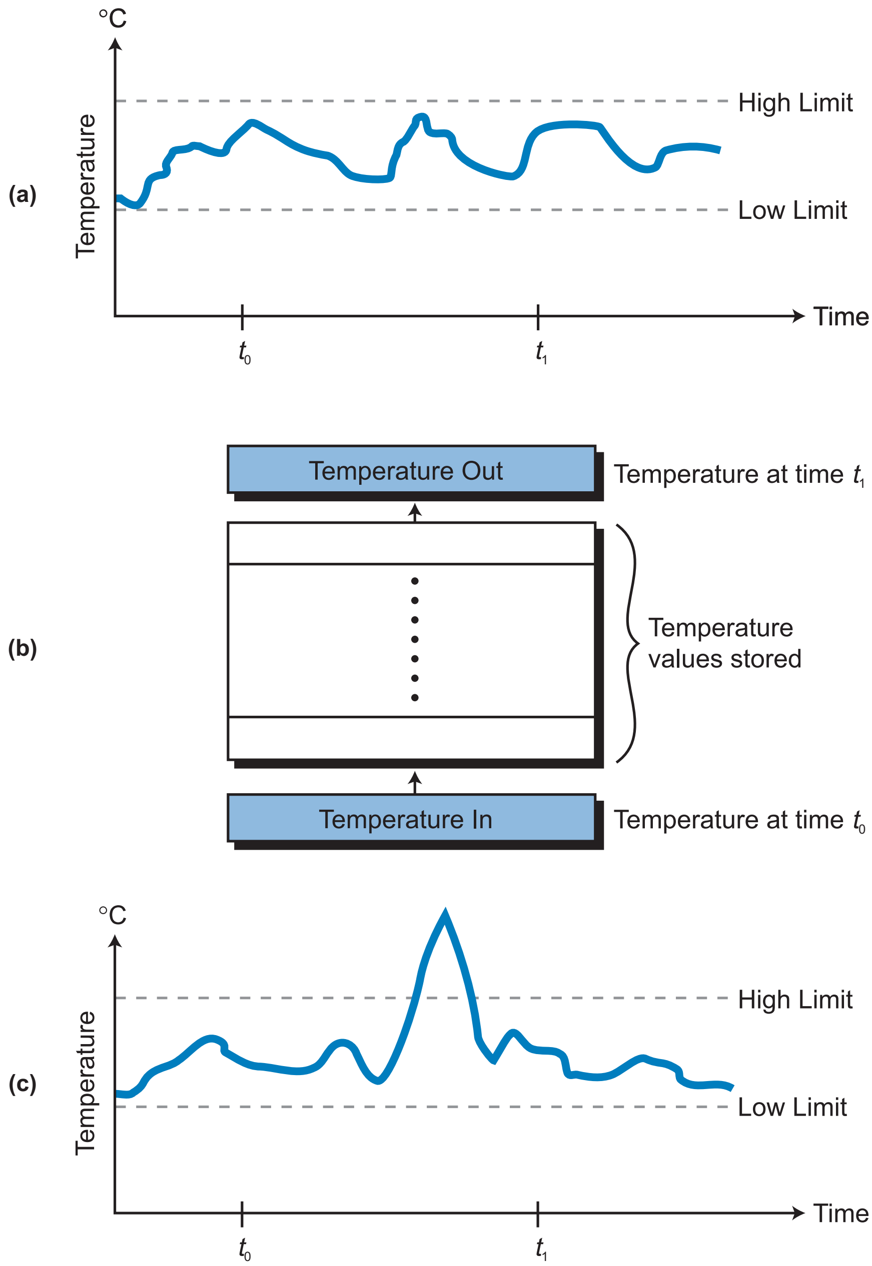

Figure 16-7. (a) Profile of temperature readings, **(b)** FIFO storage method, and **(c)** high-limit alarm value in the example process.

If a high-limit alarm occurs (see Figure 16-7c), a normal system would control the cooker by adjusting its temperature loop. However, the temperature fault may not have been caused by a temperature loop malfunction; a noise spike near the temperature transducer could have caused it.

An intelligent system would detect this sudden temperature increase by recognizing that it is well beyond the mean and median of the readings for the t_0 to t_1 period, therefore exhibiting a large standard deviation. By implementing a rule that considers the statistics of the process, the AI system will ignore the false alarm and not add the temperature reading value to the average calculations report. Furthermore, the global database of the system will receive information, for future use, about the time of day, location, and level of the spike reading. The system will closely analyze the temperature increase in case it is a true alarm. It will use temperature rate of change computations to help determine if it is a true fault.

Probability can be useful when determining or approximating the possible cause of a fault in a diagnostic ruling. One of the most commonly used probability methods is **Baye's theorem** of conditional probability. The use of this type of probability in an AI system is known as *conditional probability inferencing*. To employ probability computations in any system, however, the system must maintain historical information about the process. The expert generally provides this type of data.

Baye's theorem defines the probability of X event occurring based on the fact that Y has already occurred [$P(X/Y)$] as:

$$P(X/Y) = \frac{[P(Y/X)][P(X)]}{[P(Y/X)][P(X)] + [P(Y/\overline{X})][P(\overline{X})]}$$

where:

$P(Y/X) =$ the probability that Y occurs when X has occurred

$P(X) =$ the prior probability that X has occurred

$P(Y/\overline{X}) =$ the conditional probability that Y occurs if X does not occur

$P(\overline{X}) =$ the prior probability that X has not occurred

EXAMPLE 16-3

Part of a conveyor system controls a solenoid-operated diverter, which sends two types of boxes to two different repackaging areas. The system uses several photoelectric eyes to determine which box goes where.

The material-handling expert indicates that, due to the size and type of the boxes and environment, the following probabilities exist for conveyor faults:

For a solenoid-caused fault:

- The prior probability of a solenoid fault is 20% (80% probability that it does not fault).

- The probability that the boxes will go to the right place when the solenoid is faulty is 35%.
- The probability that the boxes will go to the right place when the solenoid is good is 60%.

For a photoeye-caused fault:

- The probability of a photoeye fault is 35% (65% probability that it does not fault).
- The probability that the boxes go to the right place when the eye is faulty is 25%.
- The probability that the boxes go to the right place when the eye is good is 45%.

Find the most probable cause of a conveyor fault when the boxes are going to the right place.

Solution

We can include the expert's data in the knowledge representation by calculating which element has a higher percentage probability of having occurred. Using Baye's theorem, the probability that the solenoid is faulty (*S*) even though the boxes are going to the right place (*B*) is:

$$\begin{aligned} P(S/B) &= \frac{[P(B/S)][P(S)]}{[P(B/S)][P(S)]+[P(B/\bar{S})][P(\bar{S})]} \\ &= \frac{(0.35)(0.20)}{(0.35)(0.20)+(0.60)(0.80)} \\ &= 12.73\% \end{aligned}$$

The probability that the photoeye is faulty (*E*) even though the boxes are going to the right place (*B*) is:

$$\begin{aligned} P(E/B) &= \frac{[P(B/E)][P(E)]}{[P(B/E)][P(E)]+[P(B/\bar{E})][P(\bar{E})]} \\ &= \frac{(0.25)(0.35)}{(0.25)(0.35)+(0.45)(0.65)} \\ &= 23.03\% \end{aligned}$$

The computations indicate that a photoeye fault is most likely to have occurred in the conveyor system. In this event, the operator should be alerted and the system temporarily halted. Also, the global database should be updated with the statistics of the fault occurrence, so that this information can be used in the future.

Conflict Resolution

A conflict occurs when more than one rule is triggered at the same time in an AI system. Normally, a system starts executing rules based on the order of occurrence of the situation. However, when situations happen at the same time, a conflict may occur in the system. For example, a system may receive information indicating that a high temperature, a low pressure, and a flow obstruction have all occurred. These three situations, on their own or in combination, can trigger the following rule consequents:

Rule 1: IF high temperature, THEN start cooling procedure.

Rule 2: IF low pressure and flow obstruction, THEN open relief valve in main supply pipe.

Rule 3: IF high temperature and low pressure, THEN open relief valve in main supply pipe and alert personnel in the area.

Therefore, the system must make a decision about which of these three rules to implement. It must select the rule that exhibits the greater priority—in this case, rule 3. The expert provides the system with this information about the priority of rule execution.

16-6 AI Fault Diagnostics Application

The example presented in this section illustrates the use of the methodology described in the previous sections. For simplicity, we will not elaborate on the PLC program coding for the application, but we will describe the rules used to define the knowledge representation.

The AI setup in this example is a diagnostic-level system implemented by a PLC-based control system. The method of rule evaluation is backward chaining (i.e., once the system detects a fault, it searches for the cause of the fault). In the batching system, the control program implements AI fault detection for only one of the two ingredients. The rules for the second ingredient are similar to the first ingredient.

Definition of the Process

Two ingredients, A and B, are to be mixed in the tank of the batching system shown in Figure 16-8. Figures 16-9 through 16-12 show the flowcharts of the process, as well as the steam valve–versus–temperature relationships. The process is as follows:

- A flow meter counts the number of pulses to monitor the amount of the ingredients in the tank (in gallons).

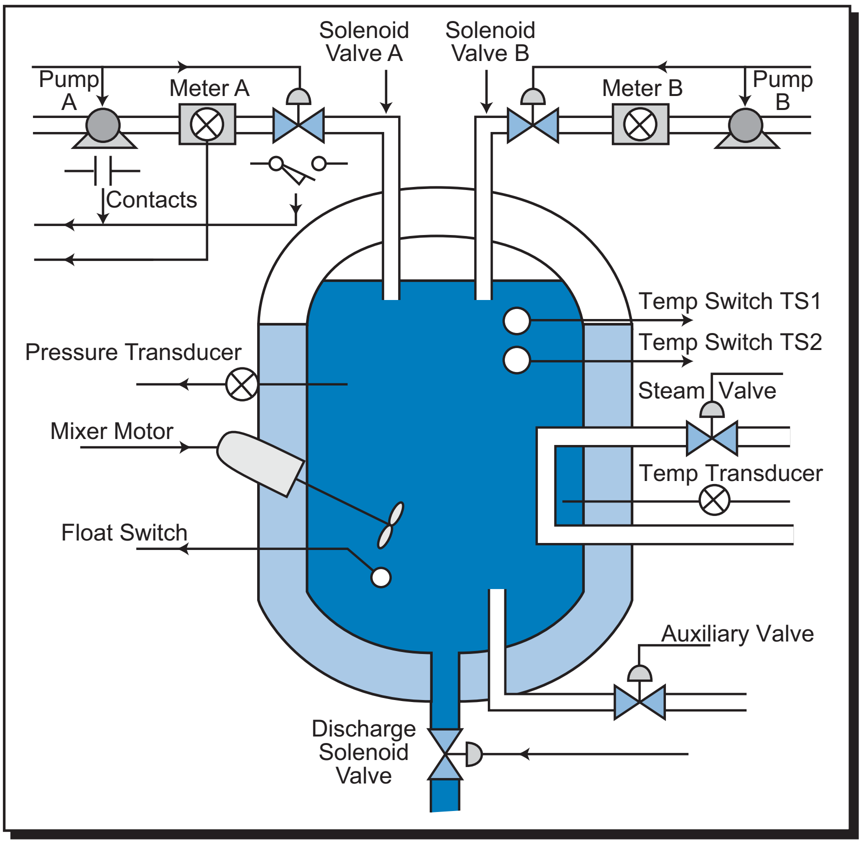

Figure 16-8. Batching system configuration.

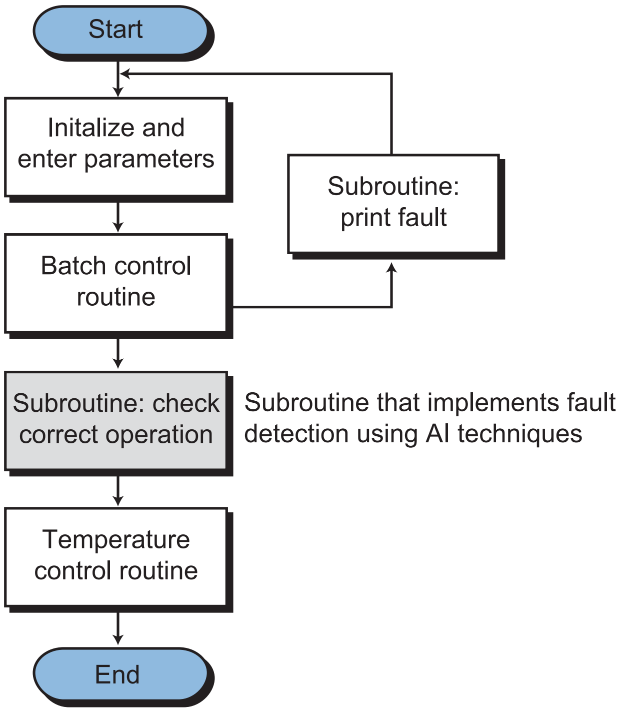

Figure 16-9. Main control program flowchart.

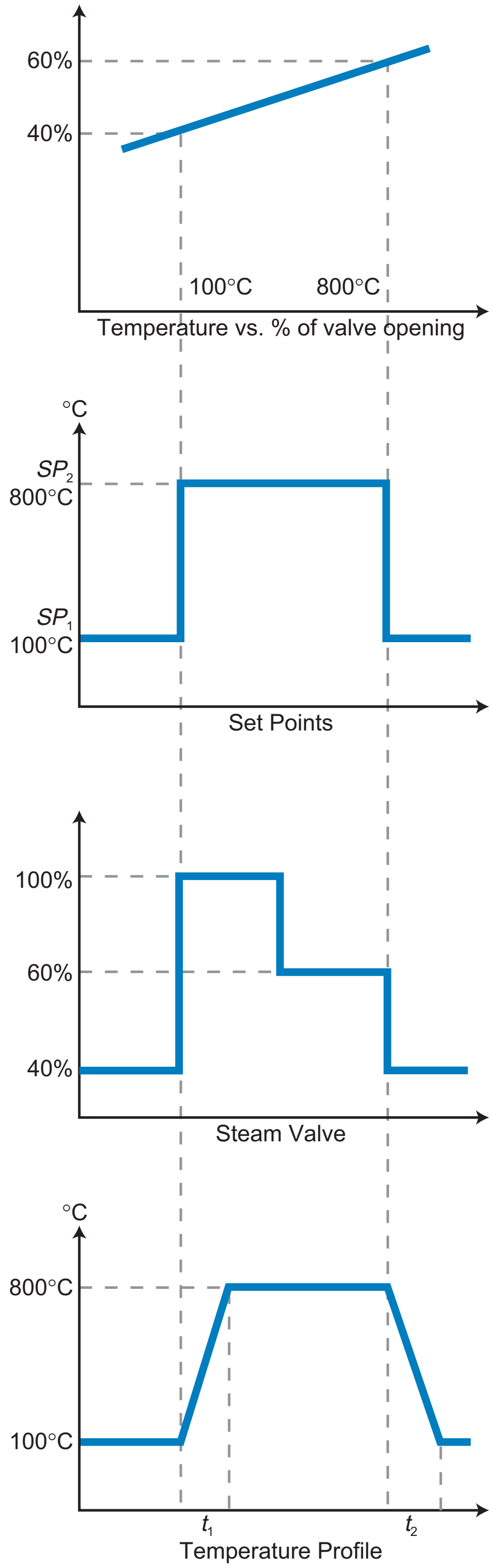

Figure 16-10. Temperature and steam valve relationship.

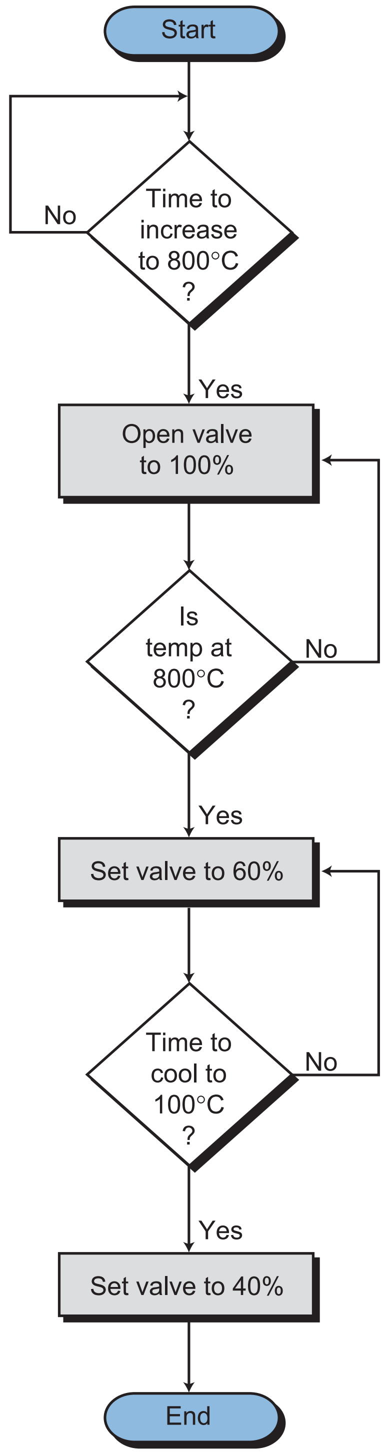

Figure 16-11. Temperature control subroutine.

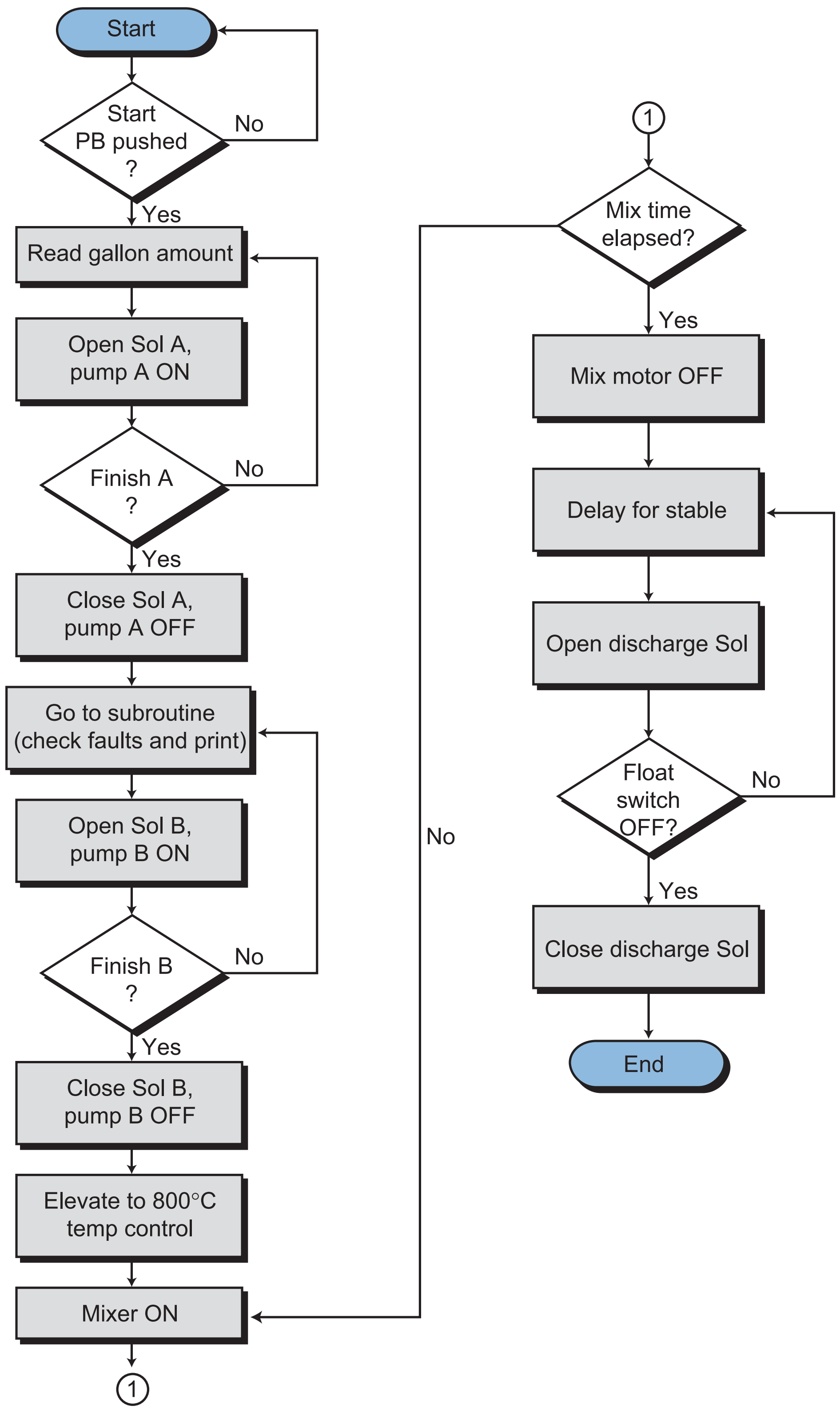

Figure 16-12. Batching control routine.

- A pump motor provides the necessary pressure to send the ingredients through the line.
- Before any of the ingredients are poured into the tank, the temperature inside the tank should be 100°C. A solenoid opens a steam valve to 40% to achieve the proper temperature in the tank.
- A load cell pressure transducer reads the volume inside the tank. It detects whether an ingredient is entering the tank, serving as a feedback device in the event of a faulty signal.
- After the two ingredients are in the tank, the temperature must be at 800°C before mixing can occur. The steam valve opens to 100% until the temperature reaches 800°C, then it remains at 60% open to maintain 800°C.
- Two thermoswitches detect the two desired temperatures (100°C and 800°C) and serve as feedback in case of a fault.
- A steam valve heats up the tank. A temperature transducer controls the temperature, maintaining it at the desired level.
- A motor agitates the two ingredients.
- An auxiliary valve disposes of the ingredients in the event that they are not mixed properly.
- When the mixing is finished, a discharge valve drains the desired solution (mixture) into the next step of the process. The steam valve returns to 40% open to cool the temperature in the tank to 100°C for the next batch.
- A float switch detects an empty tank.

PROCESS CONTROL FAULT DETECTION

Fault detection in the system occurs during three major stages of the process:

1. when the ingredients are being poured
2. during the elevation of temperature
3. during the cooling of the tank

For each of these stages, the system can provide fault–versus–possible cause information. It detects the fault through feedback information from each of the controlling and measuring devices. It then verifies this fault information by comparing it with feedback data from additional control devices. Table 16-1 shows the control and feedback devices used to perform the system check.

Control Devices	Feedback	Purpose
Valve	Limit switch and pressure transducer	Check solenoid actuation in valve
Pump	Contacts and pressure transducer	Check pump operation
Flow meter	Pressure transducer	Check ingredient flow
Steam valve	Temperature switch	Check steam valve

Table 16-1. Control and feedback devices used in batching system.

RULE DEFINITIONS

Based on the process control description and the possible failures, the system has the rules described in Table 16-2. These rules specify actions based on process occurrences and measurements.

Given the AI system's rules, we can define a set of faults *F,* representing the possible malfunctions, as:

$$F_{n,i} \text{ for } n = 0 \text{ to } 9,\ i = 1 \text{ to } 2$$

where:

$$\begin{aligned} n &= \text{ rule number} \\ i &= \text{ type of fault } (1 = \text{critical},\ 2 = \text{noncritical}) \end{aligned}$$

We can divide the set of faults ($F_{n,i}$) into two subsets—critical faults ($F_{n,1}$) and noncritical faults ($F_{n,2}$):

$$F_{n,i} \in F_{n,1} \text{ or } F_{n,2} \text{ for } n = 1 \text{ to } 9$$

The actions taken for critical faults are abort batch process, alert operator of critical fault, open auxiliary valve, and inform operator of possible faulty devices. The actions taken for noncritical faults are alert the operator, continue process and stop at end of batch, and inform operator of possible faulty devices.

APPLICATION SUMMARY

Applying AI techniques to a control system usually involves adding hardware and software to the system. The complexity of the AI program varies depending on how much fault detection is desired. The previous example presented only the rules for one ingredient. Although the rules for the second ingredient would be similar, the control system would still have to be programmed with them, and this could be time consuming.

	Rules
1	IF there is no fault, THEN operation OK. Proceed with entire process.
2	IF there is a fault in the valve, THEN check pressure. IF pressure is OK, THEN there is a fault in the limit switch. Continue process and stop. IF pressure is not OK, THEN there is a fault in the valve. Stop process and alert operator; open auxiliary valve; fix fault.
3	IF there is a fault in the pump, THEN read pressure. IF pressure is OK, THEN there is a fault in the pump's contact. Continue process and stop. IF pressure is not OK, THEN there is a fault in the pump. Stop process and alert operator; open auxiliary valve; fix fault.
4	IF there is a fault in the meter, THEN read pressure. IF pressure is OK, THEN there is a fault in the flow meter. Continue process and stop. IF pressure is not OK, THEN there is a fault in the pump and/or valve. Stop process and alert operator; open auxiliary valve; fix fault.
5	IF there is a fault in the meter and valve, THEN read pressure. IF pressure is OK, THEN there is a fault in the flow meter and/or limit switch. Continue process and stop. IF pressure is not OK, THEN there is a fault in the valve and/or pump. Stop process and alert operator; open auxiliary valve; fix fault.
6	IF there is a fault in the meter and pump, THEN read pressure. IF pressure is OK, THEN there is a fault in the meter and/or pump contacts. Continue process and stop; fix fault. IF pressure is not OK, THEN there is a fault in the pump and/or valve. Stop process and alert operator; open auxiliary valve; fix fault.
7	IF there is a fault in the pump and valve and flow meter, THEN read pressure. IF pressure is OK, THEN there is a fault in the flow meter and/or pump and/or limit switch. Continue process and stop. IF pressure is not OK, THEN there is a fault in the valve and/or pump. Stop process and alert operator; open auxiliary valve; fix fault.
8	IF there is a fault during the heating of the tank to 800°C and TS2 is ON in a set time, THEN there is a fault in the temperature transducer. Continue process and stop. IF TS2 does not respond in the set time period, THEN there is a fault in the steam valve. Stop process and alert operator; open auxiliary valve; fix fault.
9	IF there is a fault during the cooling of the tank to 100°C and TS1 is ON in a set time, THEN there is a fault in the temperature transducer. IF TS1 does not respond in the set time period, THEN there is a fault in the steam valve. Both faults are noncritical, since the batch is finished; they do not require abort batch commands.

Table 16-2. Batching system rules.

We could add intelligence to the system by storing data from the process (e.g., how many times the pump has been turned ON, the contact status feedback to the system, how many times the valve has been turned ON and OFF, which limit switch responded, etc). This data, in conjunction with information about the last time and type of failure, when and how it was fixed, and when the last maintenance was performed, would allow the system to identify whether two possible causes generated a single fault. The global database would store this additional information, allowing the system to make decisions based on the probabilities assigned or calculated throughout several past process performances. Undoubtedly, the more intelligent a system is, the more productive it will be. Additional intelligence means less downtime and a safer process environment.

KEY TERMS

artificial intelligence (AI)
backward chaining
Baye's theorem
blackboard architecture
diagnostic AI system
expert AI system
forward chaining
global database
inference engine
knowledge AI system
knowledge database
knowledge inference
knowledge representation
rule-based knowledge representation

This page intentionally left blank.