5

Intelligent Real-time Expert System Environment in Process Control

Grantham K. H. Pang
The University of Hong Kong

Raymond Tang
Esso Petroleum Canada

Stephen S. Woo
Esso Petroleum Canada

This chapter discusses the fundamental issues in a real-time expert system environment for process control. A new methodology which integrates Petri Net, fuzzy logic, and real-time expert system called Continuous Fuzzy Petri Nets (CFPN) is presented. This methodology has already been applied to the monitoring of an oil refinery processing unit. The major advantage of CFPN is that it provides a novel approach for engineers to carry out system modeling, operational analysis, process monitoring, and control. The developed system can relieve the operator from monitoring sensor data information and allow him to concentrate on the higher level interpretation of process event occurrences.

5.1 Introduction

The process control industries embody a major sector of activities in our modern society. They include petrochemical, pharmaceutical, pulp and paper, rubber and plastics, food and beverage, glass, metal industries etc. Efficient and safe operation of these process control systems is mandatory in the advancement of any society or country. Automation of process control plants very often involves automating a process or a combination of a number of subprocesses that need to be controlled. At the top level of operation, a corporate computing system could have all the information of the process, including all the business and manufacturing aspects. At the middle level, a supervisory computer carries out the network communications and maintains the database and application. At the lowest level of operation, it consists of the control and instrumentation layer with instruments to monitor, sense, and manipulate the process variables. There are many opportunities to introduce artificial intelligence (AI) or expert systems (ES) into process control, especially at the supervisory host computer level. This level is tightly integrated into the control layer.

In recent years, many interesting applications for expert systems have been developed and introduced in process control systems. The applications of intelligent systems could appear in many areas. First, alarm management is one of them. In a plant environment, either a large quantities of alarms go off at the same time or no alarms go off. An expert system which would intelligently figure out what is really causing the problem and what to do about it would be of tremendous benefit in the safety and operational functioning of a plant. The second important use of an expert system is in scheduling. It is a very tough problem because it affects all of the other aspects of the plant environment. The problem is particularly complicated in the batch processing industries where a finite run of one item is followed by a finite run of another item. Proper scheduling affects the productivity and inventory control of the plant. Thus, it could have important financial consequences to the operation. The third area is in supervisory control and optimization. Supervisory control and optimization adjusts the parametric coefficients in the various traditional control structures throughout the plant. It would require an intelligent system to identify that there is a need to tune, to reason about which tuning method to employ, and then to go ahead and apply the new tuning coefficients.

5.2 An Expert Systems Approach

The AI/ES approach to process control is both viable and necessary in many situations. The first is the existence of manufacturing processes whose behavior does not make them suitable for conventional techniques. For example, there are system properties which are intractable using a model-based approach but can be overcome using a rule-based approach. That is, a collection of formulas cannot totally model the process and allow you to arrive at a solution functionally and analytically. These properties can include nonlinearities in the dynamics of the process and unreliable or scarce measurements of process variables. Under such circumstances, a control strategy which relies on mathematical models is often ineffective and valid in only a narrow range of operating conditions.

Another situation which points to the use of expert systems is when the control of the process mimics certain qualities that are normally associated with the way human beings function. It may exhibit some reasoning, based on broad understanding, to focus experience, skill, and knowledge on an aspect of a problem at just the right time. Also, there is a need to search, which is somewhat related to not being able to find a closed form analytical solution. Very often, the problem at hand could deal with objects and relationships between objects. There may be many non-numerical data and associations involved in the process.

The environmental and safety concerns are crucial in the control of many industrial processes. Three outstanding incidents that have raised the public awareness are the Three-Mile Island incident, the methyl isocyanate release at Bhopal, and the nuclear failure at Chenobyl. In some cases, it has been shown that traditional control schemes have had success in optimizing raw material and energy efficiencies but cannot fulfill the role of monitoring safety and environmental aspects of a process simultaneously. It has become evident that the human operator plays an extremely important role in the mitigation of process plant accidents. However, a typical process plant may include hundreds of interacting units of various types. A human operator of such a plant could be provided with computer displays of the plant status. The operator may be responsible for measurement readings and alarm indicators of the order of thousands, or even ten thousands. Also, the information is presented at a lower level than individual measurements or trend plots of measurements. The interpretation of these readings and displays is the operator's responsibility. Hence, it is clear that the operator suffers from an overload of information.

In process control plants, the human operator is responsible for choosing the operating setpoints which can affect the quality, safety, and economy of the plant. In addition, he is in charge of the plant startup, shutdown, load change, and production changes. Of particular significance is the handling of accidents and emergency conditions. A skillful operator may have years of experience in observing process behavior and the results of control action. He should be aware of the process equipment failure modes and the pattern of behavior they induce. The failure of sensors and the consequences on control and alarm behavior of the control plant is also very important. Very often, there is a trend in the measurement readings or a pattern of events which leads to an emergency situation. An experienced operator would be able to detect such a trend or sequence of events and apply the correct actions before the crisis occurs. In the face of a flood of low-level measurements and alarms, it is very difficult to filter the information and make the appropriate judgment. An expert systems approach can bring along great benefits in such a situation.

5.3 Real-time Control and Petri Nets

The field of Artificial Intelligence (AI) now embodies a broad range of tools and techniques that permit the representation and manipulation of knowledge. With the advancement of both computer hardware and software, AI will continue to have a profound effect in areas such as process control. In this chapter, a novel approach called Continuous Fuzzy Petri Net (CFPN) for real-time process control and modeling is described. Continuous Fuzzy Petri Net combines several paradigms and technologies–Fuszzy control, Petri nets and real-time expert systems. These three areas are integrated to produce a powerful tool in the area of real-time process control supervision.

Petri Nets were selected because of its inherent quality in representing logic in an intuitive and visual way. Brand and Kopainsky [4] discussed the principles and engineering method of process control with Petri nets. Techniques of fuzzy control was incorporated because it is well known that operators and engineers often provide inexact knowledge in the form of rules, heuristics, or even conflicting knowledge. This problem is especially prevalent in the process control industries where plant operation is sometimes more of an art than a science. Fuzzy logic provides a framework for amalgamating these uncertainties and ad-hoc techniques into a mathematically-sound method for logical inferencing. Hence, fuzzy control is appropriate for the process control area of application.

Real-time expert systems form the backbone of this CFPN approach. The issues of real-time expert system play an important role in setting constraints and providing goals for the proposed Continuous Fuzzy Petri Net. In particular, knowledge base validity maintenance over time is an important issue. A method for the aging of assertions based on temporal distance from the present state is adopted as a means of automatic truth maintenance.

The Fuzzy Petri Net method presented in this chapter was developed for ESSO (Imperial Oil Ltd.) in Canada. The objective was to provide advanced process monitoring at an oil refinery process in Sarnia, Ontario. The tool was implemented using the G2 [11] real time expert system, from Gensym Corporation. G2 is a very powerful expert system development environment for building intelligent real-time applications. It is an object-oriented and graphical tool, together with a structured natural language. The physical and abstract aspects of the application can be represented using the G2 objects. The object-oriented environment allows the user to create new object instances by cloning existing objects or by defining objects and their properties and behaviors. All objects are organized in a hierarchical class structure and multiple inheritance is also allowed. The objects have built-in connectability which allows the developer to connect objects graphically to represent the data and logic flows. This feature allows G2 to represent the Fuzzy Petri Net concept. The set of objects created for CFPN is shown in Fig. 5.1.

Another fundamental feature of G2 is the ability to capture knowledge by creating generic rules, procedures, formulas, and relationships, that apply across entire classes of objects. The basis of developing expert systems is that knowledge can be captured and represented in the form of rules. The rules of G2 work in real time and mimic the human ability to reason on specific situations. The rules can be data-driven (or event-driven) and get triggered when new measurements are obtained from the process.

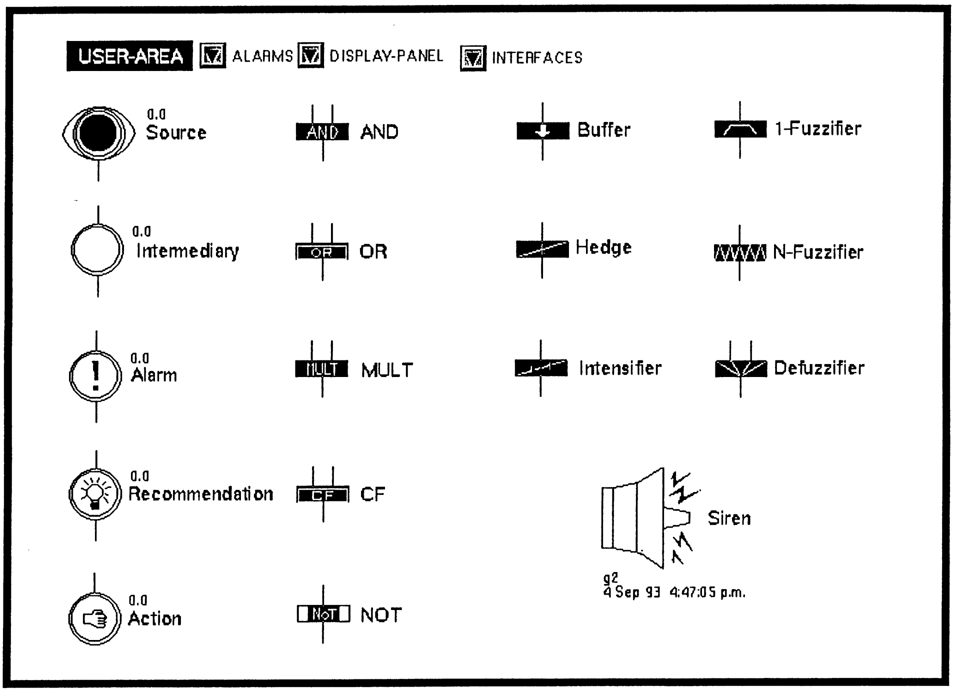

FIGURE 5.1 CFPN Symbol.

Alternatively, rules can be data seeking through backward chaining to invoke other rules, procedures, or formulas. This goal-driven feature is also implemented in CFPN and it is useful as a trouble-shooting and diagnostic tool. When the process has deviated from its desirable performance, CFPN helps to analyze the possible causes of the problem. The graphical objects and generic rules embody the knowledge of the process and its structure. The appropriate rule would be invoked automatically to infer the causes of the variance in the system operation and the possible causes of the problem and their remedy procedures.

As a tool for intelligent control, Fuzzy Petri Net can be used in many ways. It can be used to improve operator decision support. This, in turn, can improve the performance of the plant. For example, the aim could be an increase of octane content of the petroleum products, which means an increase of yield and profit. Another aim could be to optimize operations like continuous catalytic reformer (CCR). The Fuzzy Petri Net can also be used as a diagnostic tool. It can act an assistant to the operator and help to monitor the operation and trend of hundreds or thousands of sensors and actuators. If it senses an impending incident (the possibility of an undesirable event is high), it can alert the engineer with a color coded alarm. Remedy procedures can also be suggested to the operator. Thus, it is seen as an "extra pair of eyes" against problems.

5.4 Overview of Fuzzy Logic

In 1965, Lotfi A. Zadeh developed the idea of fuzzy logic and fuzzy sets [27]. Developing his ideas further in 1973, the concept of linguistic variables was introduced [28, 29, 30, 31]. Linguistic variables have values which are words rather than numbers. For instance, a linguistic variable "speed" may have values "fast," "slow," and "not very fast" and so on. Essentially, fuzzy logic provides a mathematical approach to dealing with the world which is full of imprecision and vagueness. In combination with fuzzy **IF-THEN** rules, the concept has been found useful in dealing with uncertainty in real world tasks.

Fuzzy Expert Systems

Many areas of research, such as medical diagnosis and process control, have taken advantage of the use of expert systems. However, imprecision and uncertainty were found to be an important aspect of these fields. Thus, traditional expert systems have been evolved to fuzzy expert systems to tackle these problems.[12]

The method for handling imprecision must be excellent in order for an expert system to succeed in becoming a useful tool. It has to be simple and natural so that a domain expert, such as a plant operator, can transform his knowledge into an expert system knowledge-base without difficulty.

Visualization is a powerful method in creating an environment which facilitates the translation from expert knowledge to expert system. One fuzzy expert system tool which is particularly relevant to this chapter is CASNET [26]. CASNET is a semantic net-based expert system where each node represents a state and has an associated certainty factor. There are forward and reverse weights associated with each node that give the strengths of causation between nodes. These weights correspond to the following interpretations: sometimes, often, usually, always, etc. Rules attached with a confidence value between -1 and 1 are used to link observations with states. The CFPN also takes advantage of the visual qualities in a net-based approach to expert system knowledge representation.

Fuzzy Control

One application area which has caught the attention of fuzzy logic researchers was control systems. Thus, fuzzy logic control was born and has made fuzzy logic an important and popular research topic. One of the first industrial applications of fuzzy logic control was F. L. Smidth Corp.'s cement kiln which became operational in Denmark during 1975 [13]. In a conventional control scheme, what is modeled is the system or process being controlled, whereas, in a fuzzy logic controller, the focus is on the human operator's behavior. Fuzzy controllers adjust the system parameters through a fuzzy rule-base expert system. The rule-base forms a logical model of the thought processes a human operator might go through in manipulating the system. In conjunction with the two key processes of *fuzzification* and *defuzzification*, the link between vagueness and exactness is established to make fuzzy logic a practical tool in process control.

The process of *fuzzification* is used to evaluate an exact or crisp value to its linguistic equivalent. The *membership function* plays a key role in this process. The membership function does not necessarily represent a probability density function. In fact, it is more of a similarity measure. Although probabilities may be a good starting point, the shape of the membership function can be quite arbitrary. In practice, it may often be obtained by gathering information from an experienced operator's opinion. The membership functions in practical fuzzy control may need to be fine tuned further to achieve better control. The reciprocal process of *defuzzification* brings the vague and imprecise logic of a fuzzy rule-base expert system back to the exact world of the system it is controlling. The fuzzy rule-base reaches some conclusions and makes recommendations on how to change the system parameters. Membership functions from various recommendations are combined and weighted, and through defuzzification, a crisp control value is obtained. There are various methods for defuzzification. The centroid method is a popular approach, where the centroid of the various weighted membership functions of all recommendations is taken as the crisp control value.

5.5 Overview of Petri Nets

Petri Net, as an approach to discrete event system modeling and analysis, has found great diversity and versatility. Since its creation in 1962 by Carl A. Petri [20], a rich body of knowledge concerning Petri Net theory and applications has been developed. Petri Nets have been used with varying success in modeling logic systems in many diverse fields, such as flexible manufacturing systems [16, 32, 6, 33], software engineering [14, 23, 24, 3], as well as process control and monitoring [22, 4, 1].

The most basic type of Petri Net is referred to as Ordinary Petri Net. It consists of components called places, transitions, tokens, and arcs. Hence, it is also known as Place/Transition Nets [5,2] Petri Nets have been applied in many diverse areas, which have prompted many variations and extensions of the Ordinary Petri Net. Some of the major variations are Color Petri Nets [15], Timed Petri Nets [21], Stochastic Petri Nets [19], Fuzzy Petri Nets [18, 25, 6, 7, 10] and Hybrid Petri Nets [17].

Fuzzy Petri Nets

Fuzzy Petri Nets [18, 25, 6, 7, 8, 9, 10] have been developed to model uncertainty in a knowledge base system. Tokens in these approaches represent uncertain assertions with a truth value between zero and one.

Looney [18] first defined the concept of fuzzy Petri net and applied it for rule-based decision making. Since that time, various models have been developed. Cardoso et al. [6] have attempted to introduce uncertainty in the marking of a Petri net with application in the monitoring of Flexible Manufacturing Systems. They aimed to reduce the combinatorial explosion of the complexity of the Petri net by considering a larger set of transitions as enabled by a marking with uncertainty.

In the paper by Chen et al. [7], they have developed a fuzzy Petri Net (FPN) which can work as a tool for real-time expert system modeling. However, CFPN offers more, as it is also useful as a tool for real-time expert control of both continuous and discrete systems. The fuzzification and defuzzification elements in the CFPN allows it to do this. The FPN does not include these elements. The FPN of Chen et al. is capable of dealing with uncertainty, but does not address the final step which is to bring that uncertainty to a crisp action.

Garg et al. [8] have also made modifications to the original Petri net model which is used for the representation of a set of fuzzy formulas. The resulting model can be used for automated reasoning and decision making in a fuzzy environment. However, the model is simple as it only represents knowledge in the form of propositional or first-order logic. Bugarin and Barro [9] have also proposed a fuzzy Petri net model for the knowledge representation of fuzzy production rules. Algorithms for the execution of the fuzzy Petri nets are developed and the process is carried out for the sup-min compositional rule of inference. A continuous fuzzy Petri net tool was described by Tang et al. [10]. An object-oriented description of the approach was presented. In this chapter, a formal definition of the Continuous Fuzzy Petri Net is given.

Hybrid Petri Nets

In the paper by Le Bail, Alla and David [17], a type of Petri Net which combined continuous Petri Nets with traditional Petri Nets, call Hybrid Petri Nets was introduced. Their approach was based on dividing a token into smaller units. Therefore, a continuous firing of transitions will build up sufficient of those smaller units to form one token unit.

In CFPN, the token at a place does not get accumulated. The certainty factor associated with the token represents the certainty of the assertion associated with the place. The CFPN approach is meant to provide continuous real-time inferencing. It has the ability to continuously make new inferences as sensor readings are updated.

5.6 The Continuous Fuzzy Petri Net Concept

In this section, a brief discussion of Continuous Fuzzy Petri Net is given. Continuous Fuzzy Petri Net as a modeling tool combines the paradigms of Fuzzy Logic and Petri Nets, each having different characteristics and advantages in one integrated tool. Therefore, a new paradigm that takes advantage of both approaches, through visual programming, is formed.

The Continuous Fuzzy Petri Net extends from an ordinary Petri net which consists of places, transitions, arcs, and tokens. In CFPN, places can be used to denote fuzzy propositions or other declarative knowledge.

The presence of a token represents the actual assertion and the degree of which an assertion holds true. The certainty of an assertion is color coded to provide added visual stimulus to the user. Transitions are used as functional nodes which can be used to represent linguistic hedges, fuzzification and defuzzification procedures, as well as logical operations such as AND, OR, and NOT. Arcs are used to interconnect the elements to form the logical structure of the Continuous Fuzzy Petri Net. Together, they form an intuitive visual representation of a fuzzy expert control system.

This Continuous Fuzzy Petri Net approach also introduces two important extensions to the Petri Net concept. These are the addition of a time based pattern matching algorithm for fuzzification and negative certainty values in the fuzzy logic paradigm.

The implication of the use of *continuous* to describe this approach is to contrast the difference between some of the previous work in the area of Fuzzy Petri Nets, where the Fuzzy Petri Net was used as a one-shot inferencing mechanism for knowledge representation or deals with strictly discrete event systems. This approach is meant to provide continuous real-time inferencing for the purpose of process control and modeling. Hence, it has the ability to continuously make new inferences as sensors are updated in real-time.

5.7 Definition of a Continuous Fuzzy Petri Net

A generalized Continuous Fuzzy Petri Net structure can be defined as a 12-tuple:

CFPN $= (P, T, \mathcal{P}, \mathrm{I}, O, \Phi, \Theta, \Psi, \tau, \delta, \nu, \sigma)$, where

$P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of *fuzzy places*.
$T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of *fuzzy transitions*.
$\mathcal{P} = \{\rho_1, \rho_2, \ldots, \rho_n\}$ is a finite set of *propositions*.
I : $P \times T \to \{0, 1\}$ is an input function that defines the set of directed arcs from P to T.
O : $T \times P \to \{0, 1\}$ is an output function that defines the set of directed arcs from T to P.
Φ : $P \to$ {*Source, Intermediary, Action, Alarm, Recommendation*}
is a mapping of fuzzy place to fuzzy place subclass.
Θ : $T \to$ {*AND*, *OR*, *MULT*, *CF*, *NOT*, *Buffer*, *Hedge*,
Intensifier, 1 − *Fuzzifier*, *N* − *Fuzzifier*, *Defuzzifier*}
is a mapping of fuzzy transition to fuzzy transition subclass.
Ψ : $P \to \mathcal{P}$ is a bijective mapping from fuzzy places to propositions.
τ : $T \to [0, 1]$ is a mapping of fuzzy transition to a real value between zero and one, denoting a threshold value.
δ : $T \to \{d\}$ is a mapping of fuzzy transition to time delay d, expressed as a time interval.
ν : $T \cup I \cup O \to [-1, 1]$ is a mapping of fuzzy transition and arc to a real value between zero and one, denoting a weighting.
σ : $P \to [-1, 1]$ is a mapping of fuzzy place to a real value between zero and one, and denotes the certainty value of the place.

P and T define a set of fuzzy-places and fuzzy-transitions, respectively. Ψ maps each fuzzy-place to a fuzzy proposition. (e.g., "*temperature is high*"). τ and δ define, respectively, the threshold and time delay associated with each fuzzy-transition. These establish a set of basic functions that a fuzzy-transition is to perform.

The functions I and O establish the presence of fuzzy-arc connections between fuzzy-places and fuzzy-transitions. ν defines the weighting of these fuzzy-arcs.

The associated mapping functions Φ and Θ determine the particular subclass of fuzzy place and fuzzy transition which each node ($P \cup T$) belongs. These subclasses define the nature of each fuzzy-place and fuzzy-transition. The symbol $\mathcal{F}(\cdot)$ will be used to represent a formula associated to a particular fuzzy-transition such as min, max, or fuzzify, which will be explained in a later section. These characteristics of the fuzzy entities provide us with the reasoning power in the CFPN approach.

σ maps each fuzzy-place to a certainty value (or certainty factor, which we will use interchangeably). When this σ function assigns a certainty value to a fuzzy-place, the corresponding fuzzy proposition becomes a fuzzy assertion. The certainty value tells us the degree of belief or disbelief we have of the fuzzy propositions defined in the knowledge base embedded in the Continuous Fuzzy Petri Net. So, in other words, a fuzzy proposition is simply a logical statement (e.g., "*the humidity is low*"), about which we have no idea of its truth or falseness. However, a fuzzy assertion occurs when a fuzzy proposition is assigned certainty value (e.g., "*the humidity is low with a degree of truth of 0.7*"). The σ function also assigns a marking to the Continuous Fuzzy Petri Net and puts a *token* in a fuzzy-place to denote when a fuzzy proposition becomes a fuzzy assertion.

Figure 5.1 illustrates the symbols used to represent each type of fuzzy-place (mapped by Φ) and fuzzy-transition (mapped by Θ) within the CFPN approach.

Execution of a Continuous Fuzzy Petri Net

In the CFPN approach, which is similar to Looney's approach [18], tokens are not consumed when they are *fired*. This is in contrast to how they are commonly used in other Petri Net approaches. The presence of a token in a fuzzy-place denotes a fuzzy assertion. A fuzzy assertion does not become invalid after a fuzzy-transition fires; therefore, tokens remain within fuzzy-places regardless of the firing state of a fuzzy-transition. This is necessary because we want to represent the truth states of the system we are modeling and/or controlling rather then simply providing a reasoning mechanism.

The certainty value of each fuzzy-place is determined by whether or not it is being *substantiated*. A fuzzy-place is substantiated by any tokens that are fed into that particular fuzzy-place. Tokens are modeled as a continuous entity since the CFPN is trying to deal with continuous real-time inferencing. Tokens are created inevitably at a source place as sensors are updated. Tokens are fed into a fuzzy-place by any of its input fuzzy-transitions. A fuzzy-transition fires if the absolute value of the certainty factor, evaluated by the formula governing the behavior of the fuzzy-transition, equals or exceeds the threshold value of that particular fuzzy-transition. Hence, a fuzzy-place is substantiated by a token if:

$$\nu(t_i) \times \mathcal{F}(x(t)) \geq \tau(t_i)$$

where $\nu(t_i)$ is the certainty value modifier of the fuzzy-transition, (t_i) feeding the fuzzy-place, and, $\mathcal{F}(x(t))$ is the formula governing the behaviour of the fuzzy-transition. This function represents the mathematical or logical function performed by a fuzzy-transition such as min, max, fuzzify, etc. $x(t)$ is the set of inputs to the fuzzy-transition at time, t, representing the certainty value of tokens feeding a fuzzy-transition from all of its inputs. However, note that transitions such as buffer, not, intensifier, hedge and the fuzzifier have only one input. $\tau(t_i)$ is the threshold of the fuzzy-transition t_i.

While a fuzzy-place is being substantiated, the certainty value of a fuzzy-place p_j is given by:

$$\sigma_j(t + d) = \nu(t_i \times p_j)\nu(t_i) \times \mathcal{F}(x(t))$$

for one or more fuzzy-transition inputs. $\nu(t_i \times p_j)$ represents the modifier of the fuzzy-arc connecting a fuzzy-transition t_i to the fuzzy-place p_j, and d is the delay of the transition t_i, with the rest defined as above.

When a fuzzy-place p_j is not substantiated by any fuzzy-transition output, the certainty of p_j is given by:

$$\sigma_j(t) = \text{Age}(\sigma_j(t_o), t - t_o)$$

where Age$(\cdot)$ is a formula governing the aging process of the fuzzy-place p_j and t_o was the time the fuzzy-place p_j was last substantiated. The current time t will always be greater then or equal to t_o. The formula Age$(\cdot)$ is defined such that $\sigma_j(t) = \sigma_j(t_o)$, for $t = t_o$ and $\sigma_j(t) \leq \sigma_j(t_1)$, for $t \geq t_1$. That is Age$(\cdot)$ is a non-increasing function of time. In effect, the Aging function determines the certainty value of a token that has been put into a fuzzy-place. While a fuzzy-place is being substantiated, we can view the situation as a token continuously replacing an old one, thereby resetting t_o to the current time continuously. (e.g., $t = t_o$)

CFPN Places

The mapping function Φ determines the nature of a fuzzy-place within the Continuous Fuzzy Petri Net framework. Each place has a certainty factor associated with it and this represents the degree of truth or disbelief we have about a certain fuzzy assertion or state. The CFPN approach assigns a color to the associated token within each of these places to reflect the certainty to take advantage of the visual quality of Petri Nets. A different color coding scheme is used for each type of fuzzy-place. For example, intermediary places may have a color scheme that range from white through grey to black, while an alarm place may have a color scheme that range from green through yellow to red.

Source Place

This type of fuzzy-place is used to collect real time sensor data. Its certainty indicates the freshness of the data. As sensor readings are obtained, the certainty of a fuzzy-source is set to fuzzy-source places are the input interfaces of the CFPN to the external world.

Intermediary Place

The intermediary place is used to denote general fuzzy assertions and intermediary fuzzy assertions which may be used in combination to make conclusions. The certainty indicates the degree of truth of the assertion.

Action Place

The action place is used to store the results from a defuzzifier transition and specifies a particular control output value that should be sent to an actuator. This type of place is the output interface of the CFPN to the external world. This completes the final step in the aim of using CFPN to encompass fuzzy control from sensor gathering, through inferencing, to control decisions.

Alarm Place

The alarm place is used to assert fuzzy propositions that can be construed as an alarm condition (e.g., "*the boiler is overheating*"). The certainty factor indicates the degree of truth of the alarm condition. Also, a graded belief of an alarm condition can be used as a sort of early warning system, such that an alarm condition can be seen to develop (and action can be taken immediately) rather than wait until a full fledged alarm condition has occurred.

Recommendation Place

The recommendation place is used to make recommendations (e.g., "*set valve high*") to a defuzzifier transition. The certainty indicates the confidence of the particular action we are recommending. These objects are usually connected to a defuzzifier transition. As a tool for a practical application, it was found that it was necessary to define a reasonable action for all possible logical states. For example, we have a recommendation with a positive certainty to "set value high." If the statement is "do not set value high," it does not necessarily mean we actually want it to be set to anything in particular. That is, the certainty factor of that case is zero, while a negative certainty factor would have meant some particular action to take place. Each recommendation place also has a membership function associated with it. The membership function determines the range of valid control values for that recommendation and is required for the defuzzification process.

CFPN Transitions

The function $\mathcal{F}(\cdot)$ governing the behavior of a fuzzy transition is determined by the mapping function Θ. The actual functions for each of the possible mappings are described in this section.

AND Transition

This type of fuzzy-transition is used to form logical AND constructs for a fuzzy rule base. It uses the minimize function to model this behavior. That is, the lowest *certainty* value of all input **fuzzy-places** is obtained as the resultant certainty.

The AND transition function is defined as:

$$\mathcal{F}(x(t)) = \min(x(t))$$

where $x(t)$ is the set of inputs to the fuzzy-transition at time, t, representing the certainty value of tokens feeding a fuzzy-transition from all of its inputs. Each component of $x(t)$, x_i, is defined as $\sigma(p_i) \times \nu(I(p_i \times t_j))$ with p_i being an input fuzzy-place to the fuzzy-transition t_j.

OR Transition

The OR transition is used to form logical OR constructs for a fuzzy rule base. The maximize function is used to model this behavior. The highest certainty value of all input **fuzzy-places** is obtained as the resultant certainty.

The OR transition function is defined as:

$$\mathcal{F}(x(t)) = \max(x(t))$$

with $x(t)$ defined as previously.

MULT Transition

The MULT transition is used to multiply all input certainty values which can be used to represent conditional probabilities. This is an alternative definition for the logical AND condition. Depending on the purpose of their application (i.e., to base the conclusion on either the conditional probabilities or the MYCIN approach), one would choose between these two types of AND transitions.

The MULT transition function is defined as:

$$\mathcal{F}(x(t)) = \Pi_{i=1}^{n} x_i(t)$$

where $x_i(t)$ is the certainty value of each input fuzzy-place weighted by their connecting fuzzy-arcs. Each $x_i(t)$ is defined as a component in $x(t)$ and n is the number of components.

CF Transition

This transition is used as a certainty factor (CF) combiner (from which its name is derived). It is used to combine positive and negative evidence for a fuzzy assertion. The algorithm adopted allows any number of fuzzy conditions whose certainty can range from -1 to 1, to be combined to form a resulting fuzzy assertion with certainty within a range of -1 to 1. All positive evidence strengthens the measure of belief (as do negative evidence for the measure of disbelief).

Just as the MULT transition is an alternative model for a logical AND operator, the CF transition can be used as an alternative model for a logical OR. This type of fuzzy-transition is appropriate in modeling fault detection rules. Often, a fault condition depends not only on the strength of a single condition but upon several highly coupled conditions. One condition may add confidence to another while others may reduce the degree of confidence to form a combined level of confidence.

The CF transition function is defined by the following algorithm:

1) $MB = 0$
2) $MD = 0$
3) for all $x_i(t) \geq 0$ $\{MB = MB + x_i(t) - MB \times x_i(t)\}$
4) for all $x_i(t) < 0$ $\{MD = MD + |x_i(t)| - MD \times |x_i(t)|\}$
5) $CF = (MB - MD)$

where $x_i(t)$ is the certainty value of tokens of each input fuzzy-place weighted by their connecting fuzzy-arcs. Each $x_i(t)$ is defined as $x(t)$ in the AND-transition. *MB* and *MD* represent the intermediate level of belief and disbelief of a fuzzy assertion. The difference between these two values is the actual certainty value (or degree of truth) of the fuzzy assertion we are deducing.

Buffer Transition

The buffer transition allows only one input place. This transition simply passes the certainty value it receives from an input fuzzy-place to its outputs after applying the usual modifiers and fuzzy-transition actions.

The buffer transition function is defined as:

$$\mathcal{F}(x(t)) = x(t)$$

where $x(t)$ is defined as in the AND-transition. It can be used to model simple time delay or certainty factor modifier like an arc.

NOT Transition

This type of transition is used to model logical NOT constructs for a fuzzy rule base. It also allows only one input place.

The NOT transition function is defined as:

$$\mathcal{F}(x(t)) = (1 - |x(t)|) \times \text{sign}(x(t))$$

where $x(t)$ is defined as in the AND-transition.

Intensifier Transition

An intensifier transition is used to apply a contrast intensifier upon input certainty values. This serves to increase the strength of the certainty value when beyond a threshold value (i.e., 0.5), but decrease it when below this threshold. Essentially, it serves to create a steeper slope on membership functions, bringing it closer to traditional bi-level logic. That is, it has the effect of reducing the fuzziness of an assertion. Like the Buffer transition, an intensifier allows only one input fuzzy-place.

The intensifier transition function is defined as:

$$\begin{aligned}
\mathcal{F}(x(t)) &= 2 \times x(t)^2 \times \text{sign}(x(t)), && \text{for } 0.0 \geq |x(t)| \leq 0.5 \\
\mathcal{F}(x(t)) &= (1 - (2 \times (1 - x(t))^2)) \times \text{sign}(x(t)), && \text{for } 0.5 \geq |x(t)| \leq 1.0
\end{aligned}$$

where $x(t)$ is defined as in the AND-transition.

Hedge Transition

The hedge transition is used to apply linguistic hedges (e.g., "*very,*" "*slightly*") to fuzzy assertions. This is accomplished by multiplying the magnitude of the incoming certainty value by a power factor. This transition allows only one input fuzzy place.

The hedge transition function is defined as:

$$\mathcal{F}(x(t)) = |x(t)|^{\alpha} \times \text{sign}(x(t))$$

where $x(t)$ is the certainty value of each token in each input fuzzy-place weighted by their connecting fuzzy-arcs. α is the power factor attribute of the fuzzy hedge transition, where $\alpha > 0$.

1-Fuzzifier Transition

A 1-fuzzifier transition contains a fuzzy membership function which can be used to map a single sensor data value to a certainty factor.

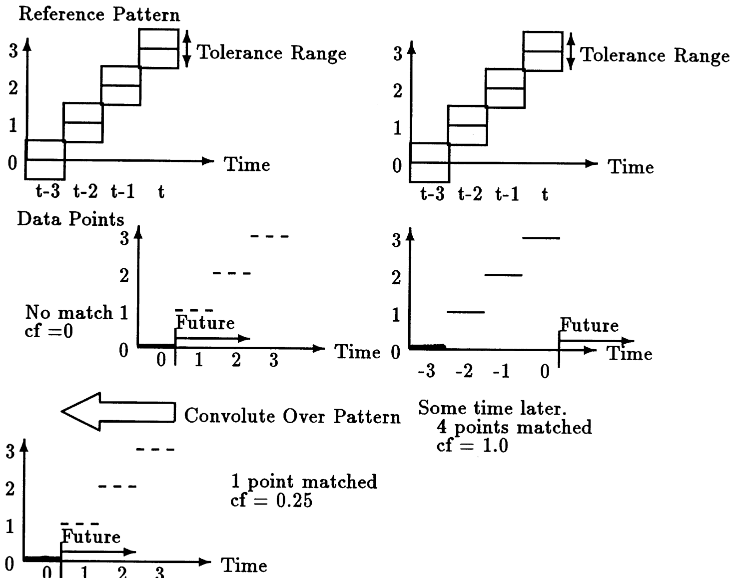

FIGURE 5.2 N-Fuzzifier Pattern Matching Procedure.

The 1-Fuzzifier transition function is defined as:

$$\mathcal{F}(x(t)) = \mu_F(\mathrm{Val}(p_i)) \times x(t)$$

where $\mu_F(\cdot)$ is a membership function of a fuzzy set F, $\mathrm{Val}(p_i)$ is the data value of the input fuzzy-source place p_i. $x(t)$ is defined as in the AND-transition and it can be used to model sensor uncertainty.

N-Fuzzifier Transition

These transitions contain a *pattern* and a *tolerance* value for each data point in a reference pattern. Each point of the reference pattern is compared with our historical data, with each point being evaluated for its similarity to this reference pattern and combined with equal weighting to form a single membership rating from 0 to 1. This evaluation also checks for partial matching of the reference pattern to detect conditions that maybe building up to the situation which we are trying to detect with our reference pattern. This is a very important function, as it allows the CFPN to provide a sort of early warning method, by producing a certainty factor evaluation which gradually increases as the data pattern becomes a closer match to the reference pattern. Figure 5.2 illustrates this procedure.

The **N-Fuzzifier** transition function is defined as:

$$\mathcal{F}(x(t)) = \max_{j=0}^{t}\left(\frac{\sum_{i=t-n+1}^{t} \mu_{q(i)}^{(w(i-j))}}{n}\right) \times x(t)$$

where $\mu_{q(i)}(\cdot)$ is a membership function for each reference point $q(i)$. The pattern q has n data points. $w(i)$ is the $i'th$ data point of the input fuzzy-source place. The max operator is used to represent the operation of shifting the reference pattern over the data points. Both the pattern and data points are stored in an array. The array is matched as is, and then shifted by one point to check for gradual matching up to the present time t.

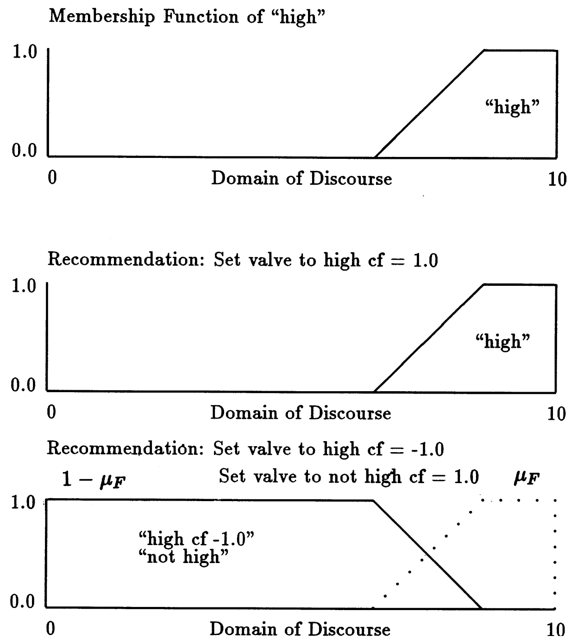

FIGURE 5.3 Defuzzifying with Negative Certainty Values.

Defuzzifier Transition

The defuzzifier transition is used to defuzzify recommendations for actions deduced from a fuzzy rule base. Fuzzy assertions are converted into crisp control output values through a defuzzifer transition. Defuzzifiers must have at least one fuzzy-recommendation and one fuzzy-action place connected to it in order to operate properly.

A special case occurs when $x_i(t)$ is a negative number. In order for the CFPN to provide a concise method for representing control actions such as "*set valve to not low*," as well as reconciling the incorporation of a measure of disbelief, negative certainty factors were used to model this behavior. We propose that when a defuzzifier transition receives a recommendation with a negative certainty, the formula $1 - \mu_F$ should be used in place of the normal membership function μ_F defined for an input fuzzy-recommendation. This provides us with a logical and consistent method for extending the fuzzy logic paradigm to include -1 to 1 logic. Figure 5.3 illustrates this procedure.

The control output value is calculated as:

$$\frac{\int_D y \times g(y)\, dy}{\int_D g(y)\, dy}$$

where D is the domain of discourse, which is the continuous range of possible real values from which crisp values are to be mapped onto fuzzy sets. $g(y)$ is defined as $\Sigma(\mu_{Fi}(y) \times x_i(t))$, over all input fuzzy-recommendation places. $\mu_{Fi}(y)$ is the membership function of a recommendation such as "set power high" and $x_i(t)$ is the certainty of the $i'th$ recommendation, thus it is used as a weighting factor on the membership function associated with that recommendation. $x_i(t)$ is defined as in the AND-transition.

5.8 Examples

A Simple Control Example

This example illustrates how one might develop an application from input sensor values to control output, by combining various logic elements of the CFPN approach to form a fuzzy rule-base. Objects such as the **Buffer** transition and the **OR** transition are introduced. **Fuzzy-recommendation** places, a **Defuzzifier** transition and an **Fuzzy-action** place are also used in the example to illustrate the control aspect of the Continuous Fuzzy Petri Net.

The CFPN constructed for this example represents the following two rules:

IF the temperature is low OR
the pressure is increasing as (0,1, 2, 3, 4, 5) then
set power to high

and another rule

IF the temperature is high then
set power to low

Referring to Fig. 5.4, the readings of the temperature and pressure sensors are simulated by a sawtooth waveform which varies from 0 to 10 units. The temperature sensor (**source place**) feeds into two different **1-fuzzifiers**: *temperature high* and *temperature low*. A simple linear mapping of the temperature values to a certainty factor is done by each of these two fuzzifier transitions. The **N-fuzzifier** transition attempts to match the pressure sensor reading to a reference pattern which varies as (0, 1, 2, 3, 4, 5). Each of the **recommendation** places, *power-low* and *power-high*, contains a membership function which determines what is considered as a low and high power setting, respectively. For the power-low setting, a triangular membership function where the power setting of 0 has a certainty of 1.0 and a power setting of 10 has

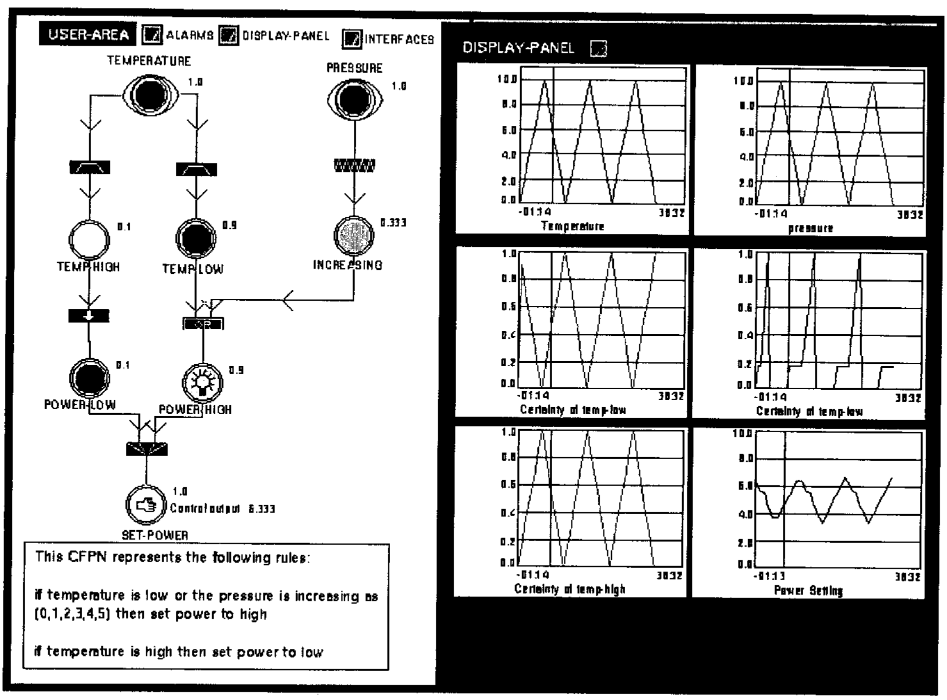

FIGURE 5.4 A Simple Control Example.

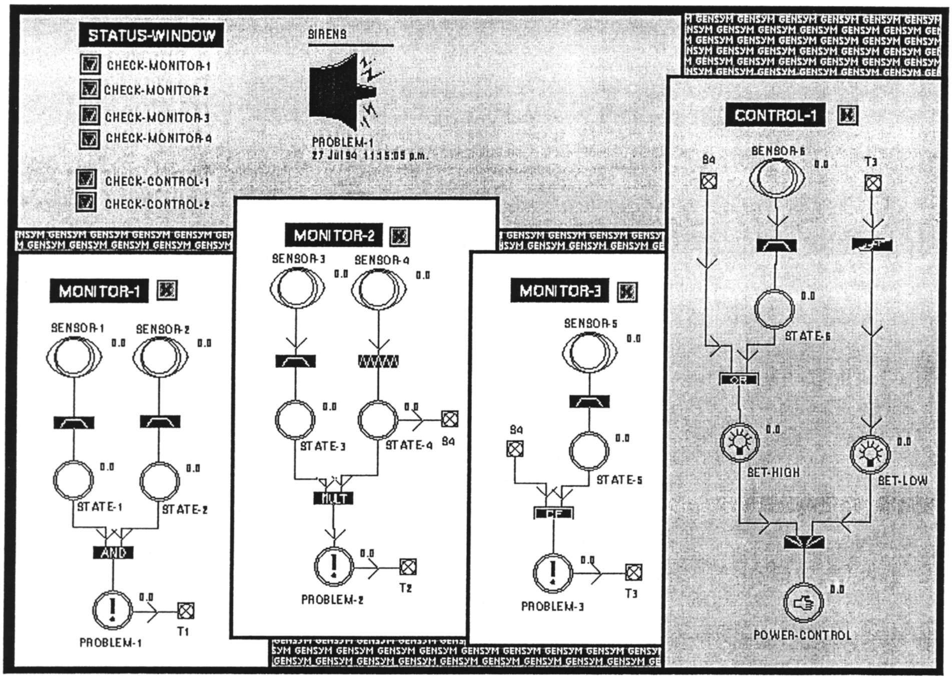

FIGURE 5.5 Sub-Windows for large CFPN networks.

a certainty of 0.0 is defined. The membership function of the power-high setting is simply the reverse with a power setting of 0 having a certainty of 0.0, and a setting of 10 having a certainty of 1.0.

The **DISPLAY-PANEL** shows the temperature and pressure sensor readings, along with their corresponding certainty values for each fuzzy assertion. The last graph in this example shows the control output (*power setting*) produced by this fuzzy rule-base.

Dealing with Large CFPN Networks

For a large industrial process, one may be faced with the problem of constructing a large CFPN to model the complex process control and diagnostic knowledge-base. To combat this problem, a separate window can be used to hold each fuzzy rule.

Within the G2 framework, a large Continuous Fuzzy Petri Network can be subdivided into several workspaces. Various "windows" can be used to access these workspaces. Some may be used to hold alarm conditions and *sirens* or to access other sub-windows. Figure 5.5 illustrates how this windowing structure operates. In this example, the CFPN is divided into six sub-workspaces or "windows" (four of which are shown). In addition, a **STATUS-WINDOW** is used to access these sub-workspaces and to hold any *alarms* created by the fuzzy rule-base represented by the CFPN. The CFPN has also been used for the monitoring of a Waste Effluent Treatment System (WETS) was developed at an ESSO oil refinery site in Sarnia, Ontario, Canada. The objective was to provide an enhanced detection and response system so as to avoid any reportable environmental incidents. A component of WETS, an Oily Water System, is shown in Fig. 5.6.

5.9 Conclusions

In this chapter, an intelligent real-time expert system environment based on the Continuous Fuzzy Petri Net (CFPN) is described. The CFPN concept has been implemented using the G2 real-time application development tool. G2 is a powerful tool ideal for the application due to its object-oriented and graphical tools, as well as the features of concurrent real-time execution, real-time rules, and structured natural language that support rapid prototyping. A formal definition of the Continuous Fuzzy Petri Net is

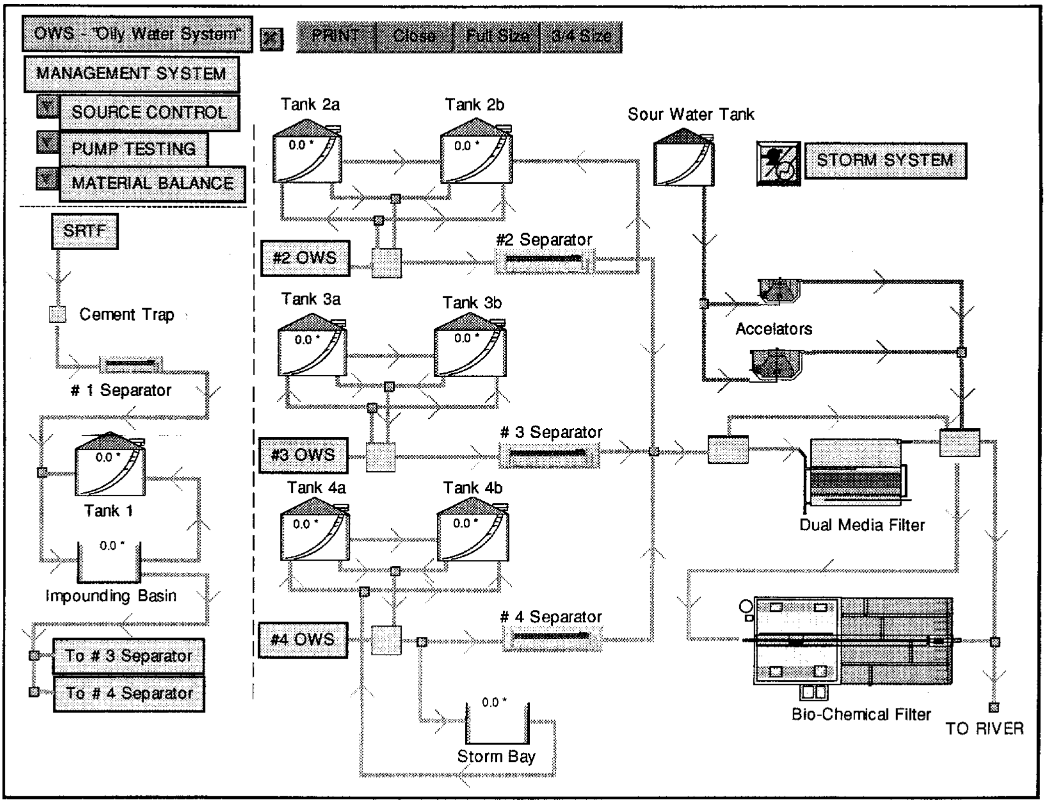

FIGURE 5.6 Oily water system.

presented. The execution and the function of the various components of the Continuous Fuzzy Petri Net are described. A time based pattern matching algorithm for fuzzification is also given. In addition, negative certainty values in fuzzy logic has been introduced as well as a method for handling negative certainty values in the defuzzification process.

A simple control example, using a Continuous Fuzzy Petri Net, is presented to illustrate the concepts developed in this chapter. ESSO Canada Ltd. has adopted the CFPN approach and has integrated it successfully with their refinery process monitoring system in Sarnia, Ontario, Canada. The Continuous Fuzzy Petri Net has addressed areas not covered by previous work on Fuzzy Petri Nets and it operates on a different level of logic compared to David and Alla's Continuous Petri Nets. Hence, the development of CFPN has made a contribution to the field, especially in that it has been used in a real industrial setting. Indeed, CFPN has provided significant insight into the understanding of the dynamics of the system at ESSO and has helped them to develop a process diagnostic system for the monitoring of an oil refinery.

The Continuous Fuzzy Petri Nets approach is a new direction in Petri Net development. It combines the flexibility of fuzzy logic and the graphical nature of Petri Nets to form a tool which can be useful for monitoring, diagnosis, decision support, and intelligent control.

References

1. K-E. Årzén, *Grafcet for intelligent real-time systems,* In *Proceedings of 12th IFAC World Congress,* Vol.4, pp. 497–500, Sydney, Australia, July 1993.
2. R. Y. Al-Jaar and A. A. Desrochers, *Petri Nets in Automation and Manufacturing,* In G.N. Saradis (ed), *Advances in Automation and Robotics: Knowledge-Based Systems for Intelligent Automation,* Vol. 2, pp. 153–225, Greenwich, Connecticut, JAI Press, Inc., 1990.

3. M. Baldassari and G. Bruno, *PROTOB: An Object Oriented Methodology for Developing Discrete Event Dynamic Systems,* In K. Jensen and G. Rozenberg (eds), *High-level Petri Nets: Theory and Application,* Springer-Verlag, pp. 624–648, 1991.
4. K. P. Brand and J. Kopainsky, *Principles and engineering of process control with petri Nets,* In *IEEE Transactions on Automatic Control,* Vol. 33 No. 2, pp. 138–149, Feb., 1988.
5. W. Brauer, W. Reisig, and G. Rozenberg (eds.), *Advances in Petri Nets 1986,* (Parts I and II), *Petri Nets: Central Models and Their Properties* and *Petri Nets: Applications and Relationships to Other Models of Concurrency. Proceedings of an Advanced Course,* Bad Honnef, Germany, Sept. 8–19, 186. Lecture Notes in Computer Science, Vol. 254 and 255, Springer-Verlag, New York, 1987.
6. J. Cardoso, R. Valette, and D. Dubois, *Petri nets with uncertain markings,* In G. Rozenberg (ed.), *Advances in Petri Nets,* Lecture Notes in Computer Science, Vol. 483, Springer-Verlag, pp. 65–78, 1990.
7. S. M. Chen, J. S. Ke, and J. F. Chang, *Knowledge Representation Using Fuzzy Petri Nets,* In *IEEE Transactions on Knowledge and Data Engineering,* Vol. 2, No. 3, Sept. 1990.
8. M. L. Garg, S. I. Ahson, and P. V. Gupta, *A fuzzy Petri net for knowledge representation and reasoning,* Information Processing Letters, 39, pp. 165–171, 1991.
9. A. J. Bugarin and S. Barro, *Fuzzy reasoning supported by Petri nets,* IEEE Trans. on Fuzzy Systems, Vol. 2, No. 2, pp. 135–150, May 1994.
10. Tang, R., G. K. H. Pang, and S. S. Woo, *A Continuous Fuzzy Petri Net Tool for Intelligent Process Monitoring and Control,* IEEE Trans. on Control Systems Technology, Vol. 3, No. 3, pp. 318–329, Sept. 1995.
11. *G2 Reference Manual,* G2 Version 3.0, Gensym Corp., 125 CambridgePark Drive, Cambridge MA 02140, U.S.A, 1992.
12. L. O. Hall and A. Kandel, The Evolution from Expert Systems to Fuzzy Expert Systems, In A. Kandel (ed.), *Fuzzy Expert Systems,* CRC Press, Boca Raton, 1992.
13. P. Holmblad and J-J. Stergaard, *Control of a Cement Kiln by Fuzzy Logic,* In M. M. Gupta and E. Sanchez (eds.), *Fuzzy Information and Decision Processes,* Amsterdam, North-Holland, pp. 398–399, 1982.
14. G. S. Hura, H. Singh, and N. K. Nanda, *Some design aspects of databases through Petri Net modeling,* In *IEEE Transactions on Software Engineering,* Vol. 12 No. 4, pp. 505–510, April 1986.
15. K. Jensen, *Coloured Petri Nets and the invariant method,* In *Theoretical Computer Science,* Vol. 14, pp. 317–336, Springer-Verlag, 1981.
16. E. Kasturia, F. DiCesare, and A. Desrochers, *Real time control of multilevel manufacturing systems using colored petri nets,* In *IEEE Int'l Conference on Robotics and Automation,* pp. 1114–1119, 1988.
17. J. Le Bail, H. Alla, and R. David, *Hybrid Petri Nets,* In *Proceedings of 1'st European Control Conference,* Grenoble, France, July 1991.
18. C. G. Looney, *Fuzzy Petri Nets for Rule-Based Decision making,* In *IEEE Trans. on Systems, Man, and Cybernetics,* Vol. 18, No. 1, pp. 178–183, Jan/Feb 1988.
19. S. O. Natkin, *Les reseaux de Petri stochastiques et leur application a l'evaualtion des systems informatiques,* These de Docteur-Ingenieur, Conservatoire National des Arts et Metiers (CNAM), Paris, June 1980.
20. C. A. Petri, *Kommunikation mit Automaten,* Ph.D. dissertation, University of Bonn, Bonn, West Germany, 1962.
21. C. Ramachandani, *Analysis of asynchronous concurrent systems by timed Petri Nets,* Ph.D. dissertation, Dept. of Electrical Engineering, MIT, Cambridge, MA, Sept. 1973.
22. W. H. Ray, *Advanced Process Control,* McGraw-Hill, New York, 1981.
23. W. Reisig, *Petri Nets for software engineering,* In *Petri Nets: Applications and Relations to Other Models of Concurrency,* Springer-Verlag, Berlin, pp. 63–96, 1986.
24. W. Reisig, *Petri Nets in Software Engineering,* In W. Brauer, W. Reisig and G. Rozenberg (eds), *Petri Nets: Applications and Relationships to Other Models of Concurrency,* Springer-Verlag, New York, pp. 63-96, 1987.

25. R. Valette, J. Cardoso, and D. Dubois, *Monitoring manufacturing systems by means of Petri nets with imprecise markings*, In *Proceedings of the IEEE Int'l Symposium on Intelligent Control 1989,* pp. 233–238, Albany, New York, Sept. 1989.
26. S. M. Weiss and C. A. Kulikowski, *Representation of expert knowledge for consultation: the CASNET and EXPERT projects,* In P. Szolovits (ed), *Artificial Intelligence in Medicine, AAAS Symp. Series,* Westview Press, Boulder, CO, 1982.
27. L. A. Zadeh, *Fuzzy Sets,* In *Information and Control,* V. 8, pp. 338–353, 1965.
28. L. A. Zadeh, *Outline of a new approach to the analysis of complex systems and decision processes,* In IEEE Transactions on Systems, Man, and Cybernetics, Vol. 3, pp. 28–44, 1973.
29. L. A. Zadeh, *The concept of a linguistic variable and its applications in approximate reasoning—Part 1,* In information Sciences, Vol. 8, pp. 199-249, 1975.
30. L. A. Zadeh, *The concept of a linguistic variable and its applications in approximate reasoning—Part 2,* In information Sciences, Vol. 8, pp. 301-357, 1975.
31. L. A. Zadeh, *The concept of a linguistic variable and its applications in approximate reasoning—Part 3,* In information Sciences, Vol. 9, pp. 43-80, 1975.
32. W. X. Zhang, *Representation of assembly and automatic robot planning by petri net,* In *IEEE Trans. on Systems, Man, and Cybernetics,* Vol. 19 No. 2, pp. 416–422, March/April 1989.
33. M. C. Zhou, F. DiCesare, and D. Rudolph, *Control of a flexible manufacturing system using petri nets,* In *1990 IFAC Congress,* Vol. 9, pp. 43–48, Tallinn, USSR, 1990.