ICOT Technical Memorandum: TM-0890

TM-0890

Plant Control Expert System Coping with Unforeseen Events-Model-based Reasoning Using Fuzzy Qualitative Reasoning-

by
J. Suzuki, N. Sueda, Y. Gotoh & A. Kamiya
(Toshiba)

May, 1990

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

Plant Control Expert System Coping with Unforeseen Events - Model-based Reasoning Using Fuzzy Qualitative Reasoning -

Junzo Suzuki, Naomichi Sueda, Yasuo Gotoh and Akimoto Kamiya
Systems and Software Engineering Lab., Toshiba Corp.
70 Yanagi-cho, Saiwai-ku, Kawasaki-city, 210 Japan

ABSTRACT

An ordinary expert system controls a plant according to heuristics. So, it fails to control the plant for lack of heuristics if unforeseen events occur as a result of abnormal situations. We propose a new framework of model-based reasoning that can dynamically generate the knowledge for plant control against unforeseen events. This proposed framework consists of three functions: (a) generation of the goal state after recovery from the unforeseen events; (b) generation of knowledge for plant control; (c) prediction of process trend curves and estimation of the generated knowledge. In the proposed framework, various kinds of models which correspond to the fundamental knowledge about plant control are used. We have implemented a thermal power plant control expert system on the basis of this proposed framework. This paper describes the model-based reasoning mechanism of the experimental plant control expert system to realize each of three functions. Especially as for (c), this paper explains qualitative reasoning mechanism using fuzzy logic.

1 Introduction

In the area of thermal power plant control, conventional expert systems based on heuristics cannot deal with unforeseen events that occur in the plant[1]. Against this limitation of the conventional system's faculty, we propose a framework of model-based reasoning[2][3]. This paper describes the mechanism to realize this framework, which can automatically generate the knowledge for plant control to solve the problems arising from unforeseen events. To generate precise knowledge, this mechanism utilizes the qualitative reasoning using fuzzy logic[4]. It's a simulation mechanism to predict the trend curves of plant processes when operating the plant according to the generated knowledge. We are developing a more flexible plant control expert system with these two mechanisms.

A skilled human operator could operate the plant somehow and deal with the unforeseen events, which are abnormal situations he couldn't foresee happening. At this time, he would think as follows.

(a) Generation of the goal state

First, he brings to mind his fundamental knowledge about plant control, namely knowledge about the plant's structure and the principles of the plant's operations. He then generates a goal state of the plant where the problems of the unforeseen events are solved.

(b) Generation of the knowledge for plant control

Second, he finds the essential operations needed to bring the plant from the current state to the goal state. He generates the conditions that he must check before doing each operation, and as a result, he generates knowledge for plant control in IF-THEN form.

(c) Estimation of the generated knowledge

Third, he predicts changes in the value of plant processes when he operates the plant according to the generated knowledge. To do this, he brings to mind his fundamental knowledge about plant dynamics, namely knowledge about relations among the plant's process parameters and knowledge about plant

Acknowledgment

Some pages of this slide are writings of Dr.Takashi Chikayama, Dr.Kazunori Ueda, Dr.Atsuhiro Goto, Mr.Masakazu Furuichi, Ms.Kumiko Wada, Mr.Katsuto Nakajima, Mr.Hiroshi Nakashima and some other researchers in ICOT and cooperating companies.

Condition Generator, the *Simulator*, and the *Knowledge Estimator*.

The Diagnosor utilizes a *Causal Model* about plant process parameters to deduce the cause of abnormality.

The Operation Generator generates the goal state of the plant where the problems of the unforeseen events is solved. The goal state of the plant is represented by the combination of each piece of equipment's state. It uses two kinds of models: an *Object Model* and an *Operation Principle*. The Object Model is fundamental knowledge of the structure and function of equipment. The *Operation Principle* is fundamental knowledge about the rules of plant operations. After generating the goal state, the Operation Generator generates the essential operations which bring the plant to the goal state. These operations are equal to the difference between the current state of plant and the goal state.

The *Condition Generator* generates the conditions of each operation generated by the Operation Generator, which are then checked before executing each operation. As a result, the knowledge for plant control is generated in IF-THEN form. The Condition Generator also utilizes an Object Model, which is fundamental knowledge about not only the structure of the plant but also the qualitative relations among plant process parameters.

The *Simulator* predicts plant behavior, namely the trend curves of plant processes, when the plant is operated according to the generated knowledge. It utilizes a *Dynamics Model*, which is fundamental knowledge about the physics in the plant and the plant control equipment.

The *Knowledge Estimator* checks whether undesirable events caused by the dynamics of the plant occur in the predicted trend curves. If they occur, the Knowledge Estimator gives feedback to the Operation Generator to generate supplementary knowledge to solve the problems of these undesirable events.

The knowledge generated in the DIS is added to the knowledge base in the SIS and utilized by the SIS, which operates the plant according to the generated knowledge, and then the problems of unforeseen events is solved. Our main concern is how to generate the knowledge for plant control against the unforeseen events. Therefore, this paper focuses on aspects of the DIS other than the Diagnosor.

3 Model based Generation of Knowledge for Plant Control

Utilizing as input the cause of abnormality deduced by the Diagnosor, the Operation Generator and the Condition Generator dynamically generate the knowledge for plant control against the unforeseen events. This section explains the model-based reasoning mechanism of the Operation Generator, using an example. Details of the Condition Generator are omitted here.

3.1 Plant Configuration

Fig.2 shows the feed water system configuration of a plant. Either water or steam flows clockwise through two kinds of water pump systems, namely the boiler-feed-water-pump system (bfp-sys) and the condensation-pump system (cp-sys). Each pump system consists of two pumps connected in parallel. Currently, only a-bfp and a-cp are activated and the volume of flowing water is 390(T/H).

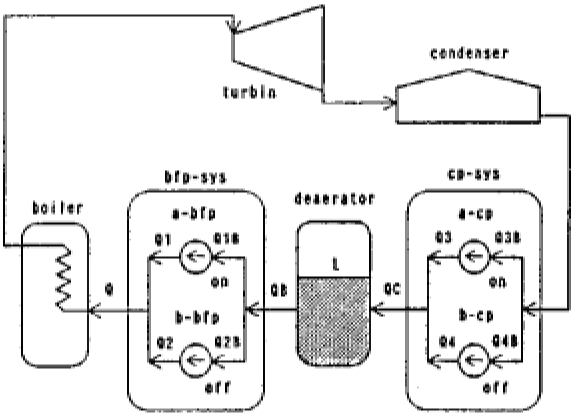


Figure 2 Feed water system configuration

3.2 Operation Generator

The Operation Generator generates the goal state of the plant where the problems of the unforeseen events are solved and generates the operations to change the plant state to the goal state. Its model representation and its model-based reasoning mechanism are described below.

control equipment. If he finds the occurrence of undesirable events in the predictions, he goes back to (a) to solve this.

A skilled human operator can solve the problems of unforeseen events by repeatedly executing steps (a) to (c). The proposed mechanism is based on the models, which are the fundamental knowledge the skilled human operator brings to his mind. This mechanism realizes his thinking process.

This paper describes the mechanism in detail. Section 2 describes the overall configuration of an experimental plant control expert system. Section 3 explains model-based reasoning mechanism to realize the above steps (a) and (b). Especially, we focus on step (a) because it's very important. Section 4 explains qualitative reasoning mechanism using fuzzy logic to realize the above step (c). Additionally, this section explains how to go back to step (a) when undesirable events occur. Section 5 is the conclusion of this paper.

2 System Configuration

The system consists of two subsystems: the *Shallow Inference Subsystem* (SIS) and the *Deep Inference Subsystem* (DIS) as illustrated in Fig.1. To perform the real experiment, this system is linked to a thermal power plant simulator for training operators, instead of an actual plant.

The SIS is the conventional plant control system based on heuristics. It consists of two modules: the *Plant Operation Executor* and the *Plant Monitor*. The Plant Operation Executor decides and executes plant operations according to heuristics, namely the knowledge for plant control in the knowledge base. The *Plant Monitor* detects occurrences of unforeseen events and activates the DIS.

The DIS utilizes various kinds of models to realize the thinking process of a skilled human operator and generates the knowledge for plant control to solve the problems arising from unforeseen events. It consists of five modules: the *Diagnosor*, the *Operation Generator*, the

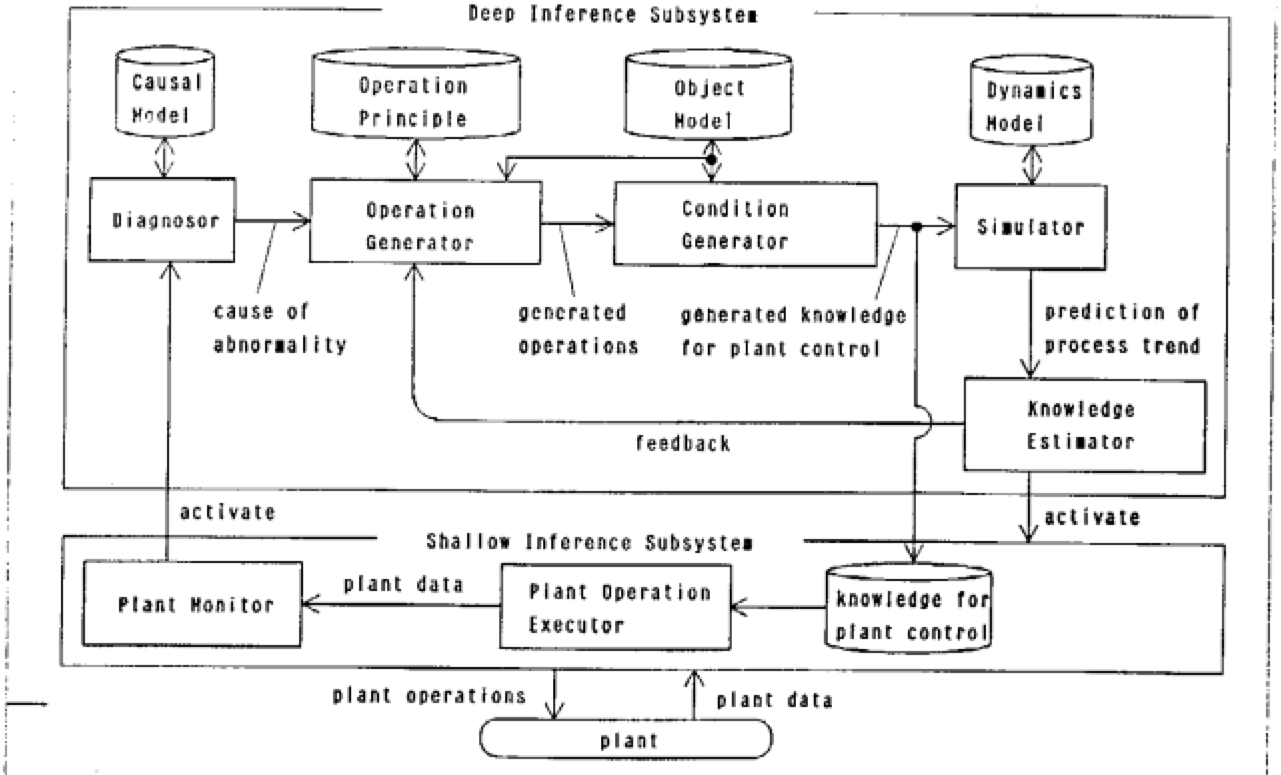


Figure 1 Plant control expert system configuration

3.2.2 Model-based Reasoning Mechanism

The Operation Generator generates the goal state according to the Object Model and Operation Principle. The plant component equipment has the state which is defined in the "states" slot of its Object Model, so the goal state of the plant is represented by the combination of each piece of equipment's state. The mechanism consists of two kinds of inference engines as follows.

(1) Operation Verification Inference Engine

The *Operation Verification Inference Engine* supports the process of checking whether the demand for a piece of equipment is still satisfied after the operation to the equipment changes its state. It gets information about the changed state from the "states" slot and the demand from the "demand" slot. Then, it checks whether the "goal" slot description still holds true. Furthermore, after checking the operated equipment itself, it can also check other connected equipment according to the "forward" slot description.

(2) Operation Derivation Inference Engine

The *Operation Derivation Inference Engine* supports the process of searching for states where the demand for a piece of equipment is satisfied. It utilizes the Object Model and Operation Principle by getting the the candidates for the state from the "states" slot and checking each candidate state to determine whether the "goal" slot description holds true. Then, the Strict Accordance Rule checks the validity of the demand to see whether the demand is within the limitations defined by the "min" and "max" slot and the demand for damaged equipment is zero. Additionally, the demand for equipment of upper hierarchy is distributed to equipments of lower hierarchy according to Preference Rule. For each piece of lower hierarchy equipment, this engine searches for the state that satisfies the distribution demand, depending on the descriptions of the "comp" and "relation" slots. After deciding the state, this engine proceeds to search for the state of connected equipment according to the "backward" slot description.

3.2.3 Example of Reasoning

Suppose that an unforeseen event occurs in Fig.2 and the Diagnosor deduces that it is caused by an abnormality of a-bfp and that a-bfp is going to stop. In this situation, Operation Generator works as follows.

(1) Operation Verification Inference Engine

It checks whether the state where a-bfp stops can be the goal state. If a-bfp stops, the capacity of "Q1" changes to "0" according to the "states" slot of the Object Model of a-bfp in Fig.3. So, the "goal" slot description becomes false. As a result, it is deduced that the state where a-bfp stops without any other operations in Fig.2 isn't the goal state to achieve.

(2) Operation Derivation Inference Engine

It searches for a new state to satisfy the "goal" slot description. Since the new state of a-bfp is fixed to "off" according to the output of the Diagnosor, its demand "Q1=390" in the "demand" slot will never be satisfied. As a result, this demand must be changed at the origin specified in the "demander" slot, namely bfp-sys of the upper hierarchy.

Bfp-sys has the Object Model in Fig.4. Its demand, "Q=390" in the "demand" slot, is distributed to the pumps of the lower hierarchy, "a-bfp" and "b-bfp" in the "comp" slot. The Strict Accordance Rule decides the new demand will be "0" for the damaged pump a-bfp. The Preference Rule decides the new demand will be "390" for another sound pump b-bfp. B-bfp has the same object model as a-bfp, so the new state of b-bfp is deduced to be "on".

After the new states of bfp-sys are fixed, the newly necessary volume of inflow "QB" is calculated according to the "relation" slot in Fig.4, and it becomes the new demand for the deaerator on the inflow side, according to the "backward" slot. As a result, the Operation Derivation Inference proceeds to all equipment on the inflow side of bfp-sys, and then it generates the goal state of the plant.

Taking the difference between the generated goal state and the current plant state in Fig.2, the essential operations to solve the problem of the unforeseen event are generated as follows:

a-bfp: "*on*"→"*off*" (halt pomp)
b-bfp: "*off*"→"*on*" (activate pomp)

3.2.1 Model Representation

The two kinds of models used by the Operation Generator are as follows.

(1) Object Model

The *Object Model* is fundamental knowledge about the structure of the plant. Every piece of component equipment in the plant has its own Object Model, that is, an Object Model is defined for each device. Fig.3 shows the Object Model of a-bfp. It is represented in the "slot-value" form like the frame representation. For example, the "states" slot says that a-bfp has binary state, "on" and "off", and the "capacity" of the outflow "Q1" is "400(T/H)" or "0(T/H)" respectively. The demand for the outflow of a-bfp is "390(T/H)" (in the "demand" slot) and the current state of a-bfp is "on" (in "status" slot) , so this demand is satisfied because the condition in the "goal" slot becomes true. The limitation constraints for the outflow are described in the "max" and "min" slots.

```
demand   : Q1 = 390(T/H).
goal     : Q1 ≤ capacity.
demander : bfp-sys.
status   : on.
states   : if status = on
           then capacity = 400(T/H),
           if status = off
           then capacity = 0(T/H).
system   : bfp-sys( Q1,Q1B ).
relation : Q1 = Q1B.
min      : Q1 = 0(T/H).
max      : Q1 = 400(T/H).
                   ⋮
```

Figure3 Object model of a-bfp

The Object Model can be described hierarchically. Fig.4 show the Object Model of bfp-sys, which is the parent Object Model of a-bfp and b-bfp. Bfp-sys consists of two pumps (in the "comp" slot) and is connected to the "deaerator" for inflow direction (in the "backward" slot) and to the "boiler" for outflow direction (in the "forward" slot). In Fig.3, the "system" slot says the parent Object Model of a-bfp is as the same as that for bfp-sys. Furthermore, the "relation" slot describes the relations among parameters of a-bfp and the "demander" slot describes the origin of the demand for a-bfp. The relations in the "relation" slot holds true in the static state of plant, so it is utilized for state generation.

```
demand   : Q = 390(T/H).
goal     : Q ≤ capacity.
demander : boiler.
comp     : a-bfp( Q1,Q1B ),
           b-bfp( Q2,Q2B ).
forward  : boiler( Q ).
backward : deaerator( QB ).
relation : capacity =
              capacity( a-bfp )
              + capacity( b-bfp ),
           Q = QB,
           Q = Q1 + Q2,
           QB = Q1B + Q2B.
                  ⋮
```

Figure4 Object model of bfp-sys

(2) Operation Principle

The *Operation Principle* is the constraint for safe and efficient plant control. It is used to determine the demand, which is in the "demand" slot of the Object Model. It consists of two kinds of rules, as follows, which are both implemented as procedures in the operation derivation inference engine explained later.

(a) Strict Accordance Rule

The *Strict Accordance Rule* is the rule to maintain safety in plant operations. That is, it is a rule to keep damaged equipment out of service and to use equipment within its own limitation constraints, which are described in the "min" and "max" slot of its Object Models.

(b) Preference Rule

The *Preference Rule* is a rule to maintain efficiency and economy in plant operations. That is, it is the rule to keep the number of the activated equipment to a minimum and to equalize the activated time of each piece of equipment.

(3) **Fuzzy rules**

Fuzzy rules represent the qualitative calculation primitives. By categorizing the range of value by membership functions finely so far as no trivial category is generated, a reasonable solution can be calculated without combinational explosion by ambiguity. Table 1 illustrates the fuzzy rules for "add(X,Y,W)". For a pair with the fuzzy values of X and Y, all the fuzzy rules in Table 1 are applied to calculate the fuzzy value of W. So, for example, the rule if X="P" and Y="N" then W="Z" doesn't say that W is zero, but says that the grade of membership of W for the "zero" category is calculated by this rule.

Table1 Fuzzy rules for add(X,Y,W)

Y \ X	N	NS	Z	PS	P
N	N	N	N	NS	Z
NS	N	NS	NS	Z	PS
Z	N	NS	Z	PS	P
PS	NS	Z	PS	PS	P
P	Z	PS	P	P	P

(4) **Simulation procedure**

The simulation is executed with propagation and prediction steps in turn.

Step1) Calculate the fuzzy value of the input parameters.

Step2) Execute prediction by calculating the integration primitives.

Step3) Execute propagation to get the value of the other parameters at the new simulation time.

Step4) Go to Step1.

4.1.2 Example of Simulation

In Fig.2, two Dynamics Models are implemented for the flow control of bfp-sys and the water level control of the deaerator. Namely, bfp-sys is controlled to keep the total volume of its outflow at 390(T/H), and the deaerator is controlled to keep its water level at 0(mm). Fig.7 gives a block diagram of the model for the deaerator.

The Simulator predicts the trend curves of the plant process when operating the plant according to the generated knowledge explained in the previous section. Fig.8 shows the result of the simulation. The trend curves of the inflow to bfp-sys and the deaerator, "QB" and "QC", are simulated as illustrated in Fig.8(a) and Fig.8(b) respectively. The fluctuations are caused by the delay of plant control equipment.

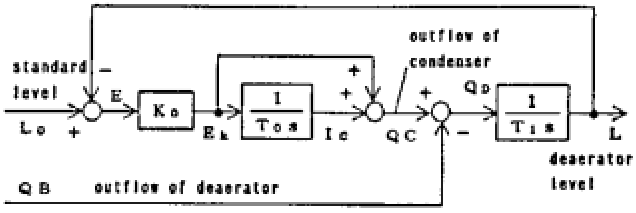
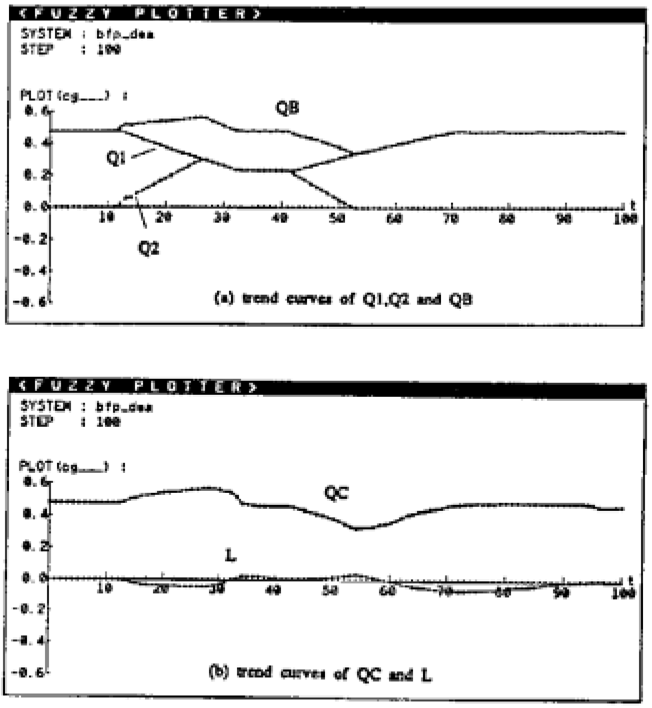


Figure 7 Block diagram for water level control in the deaerator



(a) trend curves of Q1,Q2 and QB

(b) trend curves of QC and L

p.u. : per-unit (Fuzzy Value)

Figure 8 Simulator output

4.2 Knowledge Estimator

4.2.1 Item to Estimate

The Knowledge Estimator checks whether the violation of the limitation constraints occurs in the Simulator output. These constraints are defined in the "min" and "max" slots of the Object Model. If a violation occurs, it gives feedback to the Operation Generator to generate knowledge to deal with such an undesirable violation. Currently no priority for the

3.3 Condition Generator

This module generates the conditions for each operation generated by the Operation Generator. It utilizes the Object Model where additional information not used by the Operation Generator is described. This is knowledge about the qualitative relations of changing processes caused by operations, the time required by operations, and the maximum changing ratio of processes. To make use of the qualitative propagation mechanism, it generates four kinds of conditions according to the Object Model. Fig.5 shows the generated conditions for b-bfp activation. Due to limited space, more details about this mechanism are omitted here.

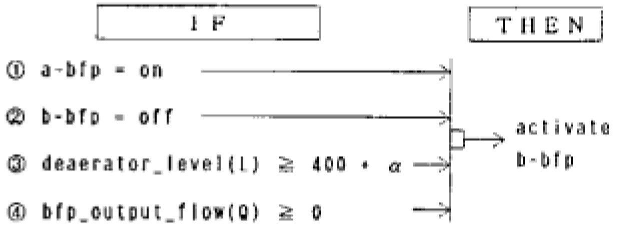


Figure 5 Conditions for b-bfp activation

4 Model-based Estimation of the Knowledge for Plant Control

The Operation Generator and Condition Generator generate knowledge by generating the goal state of the plant. This goal state is regarded as the state at a given time point, so the model they use is based on static relations among process parameters and represents no time concepts. In general, time concepts are very important, especially in the area of control, so the dynamic relations among process parameters must be taken into consideration. From this point of view, we incorporate a simulation mechanism with the model-based reasoning mechanism[5].

First in this section, we explain a qualitative reasoning mechanism using fuzzy logic to predict plant behavior. Next, we explain how to detect the occurrence of undesirable events and how to give feedback to the Operation Generator to solve this.

4.1 Simulator

4.1.1 Qualitative Reasoning Using Fuzzy Logic

The Simulator predicts the trend curves of plant processes caused by the plant operation. It utilizes a Dynamics Model. A quantitative simulation based on a detailed model can predict the exact trend curves, but this detailed model is often hard to build. On the other hand, a qualitative simulation based on a qualitative model is useful because qualitative modeling is easier and because the qualitative reasoning process better matches to the thinking process of a skilled human operator[6][7]. However, a qualitative simulation has the following disadvantages:

(a)Because of the ambiguity in qualitative calculus, there is the possibility that many solutions are calculated.

(b)A qualitative solution cannot be used directly by the plant control system.

In order to solve these disadvantages while keeping the advantages explained above, we propose a method that employs a qualitative reasoning mechanism using fuzzy logic[8]. This mechanism is explained below.

(1) **Model representation**

The Dynamics Model is represented by the ambiguous quantitative values (fuzzy values) of input parameters and constants, and the relations among all parameters. This model consists of two kinds of models, which describe the physics of the plant and the plant control equipment. The following six primitives of fuzzy calculation are used:

Integration	$X = \int Y dt$	integ(X,Y)
Sign reversal	$Y = -X$	minus(X,Y)
Addition	$W = X + Y$	add(X,Y,W)
Multiplication	$Q = X \times Y$	mult(X,Y,W)
Coefficient	$Q = a \times X$	coef(a,X,W)
Equality	$W = X$	equal(X)

(2) **Membership functions**

Five membership functions, illustrated in Fig.6 ,are used.

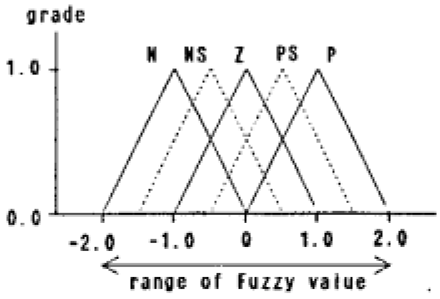


Figure 6 Membership functions

5 Conclusion

This paper proposes a mechanism of model-based reasoning using fuzzy qualitative reasoning, which realizes a flexible expert system for plant control that can solve the problems of unforeseen events. The mechanism actualizes the thinking process of a skilled human operator. This mechanism has two characteristic features: the generation of a goal state by a model of static relations and the estimation of the generated knowledge by a model of dynamic relations. As our second contribution, we apply to this estimation the technique of qualitative reasoning using fuzzy logic. This mechanism is implemented on a PSI-II(Personal Sequential Inference Machine) using ESP(Extended Self-contained Prolog), and is under experimental implementation.

The open problems remain as follows:

(1) Refinement of Knowledge Estimator

A new mechanism is needed to deal with the priority of the limitation constraints and to analyze the effect of the temporal operations against the constraints violation.

(2) Refinement of Simulator

In the experimentation, the qualitative reasoning using fuzzy logic doesn't produce reasonable results in all cases. In some cases, tuning up of the model or the fuzzy rules is needed.

(3) Dynamic generation of model

The example presented in this paper is a very simple one, and we make no mention of the generation of the model used. However, it is very important to generate dynamically the model which reflects unforeseen events and to transform the model to a suitable grain size for each inference module[9].

Acknowledgement

This research is carried out under the auspices of the Institute for New Generation Computer Technology (ICOT). We deeply appreciate the advice of the other researchers and of Mr.Ikoma, manager of the 5th Lab. of ICOT.

References

[1] Suzuki,M. et al. ,"Recent Thermal Power Plant Automation", Journal of the Society of Instrument and Control Engineers, vol.22, no.12, pp.1021-1028, 1982 (in Japanese)

[2] Suzuki,J. et al. ,"Deep Knowledge based Expert System for Plant Control", Proc. of 9th Knowledge Engineering Symposium, Society of Instrument and Control Engineers, pp.153-158, 1989 (in Japanese)

[3] Taoka,N. et al. ,"Deep Knowledge based Expert System for Plant Control - Generation of Plant Operations -", Proc. of 38th Annual Convention, Information Processing Society of Japan, pp.599-600, 1989 (in Japanese)

[4] Washio,T. et al. ,"Attempting of Fuzzy Qualitative Reasoning", 5th Knowledge Engineering Symposium, Society of Instrument and Control Engineers, pp.147-152, 1987 (in Japanese)

[5] Suzuki,J. et al. ,"Deep Knowledge based Expert System for Plant Control - Combination of Deep Inference Mechanism with Knowledge Estimation Mechanism -", 11th Knowledge and Intelligent System Symposium, Society of Instrument and Control Engineers, pp.7-12, 1990 (in Japanese)

[6] Kuipers,B ,"Qualitative Simulation", Artificial Intelligence, vol.29, pp.289-338, 1986

[7] de Kleer,J. ,"A Qualitative Physics based on Confluence", Artificial Intelligence, vol.24, pp.7-83, 1984

[8] Konuma,C. et al. ,"Plant Control Expert System against the Unforeseen Events - Inference Mechanism with Qualitative Reasoning -", Proc. of 40th Annual Convention, Information Processing Society of Japan, pp.298-299, 1990 (in Japanese)

[9] Falkenhainer,B. ,"Setting up Large Scale Qualitative Models", Proc. of AAAI'88, pp.301-306, 1988

limitation constraints is considered, so all the constraints are treated equally.

4.2.2 Procedure to Solve Constraint Violation

The detected violations are transitory and all processes settle down for the static goal state generated by the operation decision module. So, the operations against the violation are temporal. To generate knowledge about these operations, feedback to the Operation Generator is given. That is, the new demand for the equipment where the violation occurs is the value of the process parameter violating the constraints, and then the Operation Derivation Inference generates a temporal goal state where this new demand is satisfied. Fig.9 illustrates the general procedure.

Against the unforeseen event "M" at the current state "S0", the Operation Generator and Condition Generator generate the goal state "Se" and the knowledge "K1" for plant control. The Simulator predicts the trend curves "PS" when the plant is operated according to "K1". If there is no violation of the limitation constraints in "PS", no knowledge other than "K1" is needed to solve "M". If the violation occurs after the intermediate state "S1" between "S0" and "Se", the value "D1" of the violating parameter is taken. The Operation Generator generates the temporal goal state "S3" where the demand "D1" for the violating equipment is satisfied. After generating "S3", the knowledge "K2" to change the plant state from "S1" to "S3" is generated and estimated. This process is performed with a recursive call of this procedure. The knowledge "K3" from "S3" to "Se" is generated in the same way. As a result, all the knowledge to solve "M" is the addition of "K2","K3" and a part of "K1" which is used to change the plant state from "S0" to "S1" ("fix(K1)" in Fig.9). This procedure is executed against the first detected violation. So, if many violations occur, they are solved one by one in chronological order.

4.2.3 Example of Solving the Constraint Violation

In Fig.8, the current state "S0" is at time 10 and the goal state "Se" is at time 92, when all processes settle down. The fuzzy value of each parameter is 1.0(p.u.) when the related equipment is fully loaded. The value of "QC" is beyond 0.5 from time 14 to 33. On the other hand, in Fig.2, the maximum outflow of a-cp and b-cp is 400(T/H) namely 0.5(p.u.) and currently only a-cp is activated. So, the maximum value of "QC", which is the total outflow of a-cp and b-cp, is 0.5(p.u.). As a result, the violation of the limitation constraint for "QC", the demand for cp-sys, occurs from time 14 to 33. In Fig.8(b), the maximum volume of "QC" is 0.57(p.u.), namely 456(T/H), at time 28. And then, the new demand for "QC" is 456(T/H) and the Operation Generator generates the temporal goal state "S3" where this demand is satisfied. As a result, the temporal operation to activate b-cp is generated to change the plant state from the state "S1" before time 14 to "S3". Finally, all knowledge to change the plant state from "S0" to "Se" via "S1" and "S3" is generated as illustrated in Fig.10.

```
procedure  Estimate(M or D0, S0)
   begin
      [Se, Op] <= Operation_Generator(M or D0, S0);
      K1         <= Condition_Generator(S0, Se, Op);
      PS         <= Simulator(S0, K1);
      [NG, D1, S1]
                 <= Knowledge_Estimator(PS);
      if NG =/= constraint_violation
      then return( K1, Se);
      else
         [K2, S3] <= Estimate(D1, S1);
         [K3, Se] <= Estimate(M, S3);
         K4 <= fix(K1) + K2 + K3;
         return(K4, Se);
   end:

Notation
   Si,Se :  plant state  M: cause of abnormality
   Di :  demand          Op :  operations
   PS :  trend curves of process parameters
   NG :  flag for constraint violation occurrence
   Ki :  knowledge for plant control
   [..] :  list expression     <= :  substitution
```

Figure 9 General procedure against the constraint violation

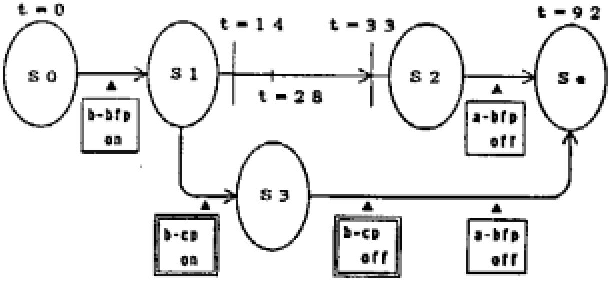


Figure10 Example for constraint violation solving

Parallel Inference Machines and Programming in the FGCS Project

Kazuo Taki

Institute for New Generation

Computer Technology

(ICOT)