TR-682

The Theory and Application of Diagnostic and Control Expert System Based on Plant Model

by
J. Suzuki & M. Iwamasa (Toshiba)

September, 1991

ICOT

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

(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

Chapter 1

The Theory and Application of Diagnostic and Control Expert System Based on Plant Model

JUNZO SUZUKI[1] and MIKITO IWAMASA[2]
Systems & Software Engineering Laboratory
TOSHIBA Corporation
70 Yanagi-cho, Saiwai-ku, Kawasaki-city 210, Japan

1. Introduction

Currently in the field of diagnosis and control of thermal power plants, the trend of the systems is that the more intelligent and flexible they become, the more knowledge they need [13]. As for the knowledge, conventional diagnostic and control expert systems are based on heuristics stored a priori in knowledge bases. So, they could not deal with unforeseen events if they occurred in the plant [7][11]. Unforeseen events are abnormal situations which were not expected to happen when the plant was designed. A skilled human operator has some kind of fundamental knowledge about plant control, so he could operate the plant and somehow deal with the unforeseen events. He would deal with them according to the following thought process.

1. Interpretation of disorders
   First, he would bring to mind his fundamental knowledge about the causal relations among process parameters of the plant. He would be able to diagnose the plant and interpret possible disorders which caused the unforeseen event.

2. Generation of plant-operations-sequence
   Second, he would bring to mind his fundamental knowledge about the structure and functions of the plant and about the principles of plant operations. According to the above interpretation, he would be able to think about what operations he should do to deal with the unforeseen event and when he should execute them. We call a series of the operations with their preconditions the operations-sequence, which is a set of the rule-based knowledge for plant control. The rule-based knowledge is in IF-THEN format.

[1] E-mail:suzuki%ssel.toshiba.co.jp@uunet.uu.net
[2] E-mail:iwamasa%ssel.toshiba.co.jp@uunet.uu.net

3. Verification of generated knowledge
   Third, he would bring to mind his fundamental knowledge about the dynamic characteristics of plant controllers and physical laws. He would be able to predict the plant behavior when operating the plant according to the generated knowledge. If he found the occurrence of undesirable events for plant control in this prediction, he would go back to (2) to solve it.

A skilled human operator could deal with unforeseen events by repeatedly executing the above steps. He would need some fundamental knowledge about diagnosis and control to deal with them. The concept of an intelligent diagnostic and control expert system is based on the above steps.

In this paper, first, we describe the theoretical aspect of these steps. Next, we describe the practical aspect of these steps, and explain the model-based plant control expert system. Finally, we also describe an approach to speed up its model-based reasoning using a parallel computer and discuss the open problems and the related works.

2. Theories of Model-Based Diagnostic and Control Expert System

This section describes a theoretical approach to realize an intelligent diagnostic and control expert system.

2.1. A Theory of Model-Based Diagnosis

When an unforeseen event occurs, a skilled human operator diagnoses its cause from the model about the structure of the plant and the physical processes in plant devices qualitatively. The main purpose of this section is to present the theoretical aspect of diagnosing an unforeseen event and to formalize the model-based diagnosis using a *Qualitative Causal Model* of the plant.

2.1.1. What to Express as Model

Firstly, we define a *Qualitative Causal Model* which represents the physical processes in the plant devices qualitatively.

Definition

A *Qualitative Causal Model* is $< H, Mn, Pt, In, Ds, Ob, R >$.

Here H, Mn, In, Pt, Ds, Ob means,

H	: Plant parameters ($H = Mn \cup Pt \cup In \cup Ds$)
Mn	: Manifestation parameters which has sensor data
In	: Input parameters
Ds	: Design parameters
Ob	: Symptom parameters ($Ob \subset Mn$)
Pt	: Pathological parameters ($H - Mn - In - Ds$)

The domain of H is three qualitative values,

$$Q \stackrel{\text{def}}{=} domain(H) = \{[+], [-], [0]\}$$

and each symbol in the brackets [ ] represents their qualitative variation from normal value.

$p \in Ob$ is a parameter which has sensor data and indicates the qualitative variation from normal value.

We define symptoms in the *Qualitative Causal Model*.

Definition

$< h, q >$ ($h \in H, q \in Q$) is a parameter which has qualitative value q.

Definition

A symptom is a pair $< p, q >$ ($p \in Ob, q \in Q$) which indicates that a qualitative variation is observed on the the sensored parameter.

Let R be a set of the qualitative causal relations between the plant parameters. Each relation corresponds to a physical process in a plant device. We define a *qualitative sentence* to represent these relations.

Definition

A *qualitative sentence* is (r, p) ($r \in R, p \in Mn \cup Pt$) [3] which describe the qualitative causal relation of the parameter p via relation r, where

$$r : p \Leftarrow \{\{+ : p_1, ...p_n\}, \{- : p_{n+1}, ..., p_m\}\} \qquad (r,p)$$

This relation means that the p and $p_i (1 \leq i \leq m)$ have a causal connection via relation r. The relation between p and $p_j (1 \leq j \leq n)$ is monotonic increase, and the relation between p and $p_j (n + 1 \leq j \leq m)$ is monotonic decrease.

We call p in (r, p) the *effect parameter* of (r, p), and $p_j (1 \leq j \leq m)$ the *cause parameter* of (r, p).

[3] We use round brackets to express a label for a *qualitative sentence*. There is no *qualitative sentence* which contains input parameters and design parameters, because these parameters are not affected by any other parameters

We compile the *qualitative sentences* into a network which is based on the connections of the parameters. We call this network Qualitative Causal Network: *QCNet*. A simple example of the *qualitative sentences* is shown in Figure 1, and Figure 2 is the *QCNet* for this example.

$$
\begin{array}{ll}
r1 : a \Leftarrow \{\{+ : b\}, \{- : c\}\} & (r1, a) \\
r2 : c \Leftarrow \{\{+ : d, e\}, \{- :\}\} & (r2, c) \\
r3 : f \Leftarrow \{\{+ : c\}, \{- :\}\} & (r3, f)
\end{array}
$$

Figure 1: An example of the *Qualitative Sentences*

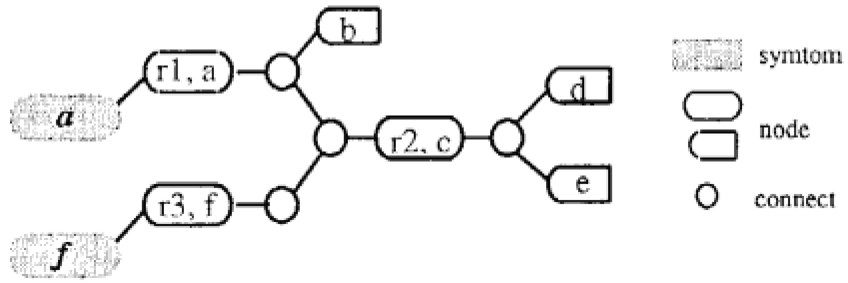

Figure 2: An example of *QCNet*

2.1.2. What to Deduce with Model

Disorders in continuous physical devices are represented in the *Qualitative Causal Model* $< H, Mn, Pt, In, Ds, Ob, R >$ described above. Disorders are classified in the following two types.

Type 1 Abnormal variation on the input parameters.
This disorder represents an external change in a plant. (ex. the temperature of sea water is high)

Type 2 Fault in plant devices, or abnormal variation on the design parameters.
This disorder represents an internal change in a plant. (ex. a pump has broken down, a pipe has gotten thin)

These disorders are translated into qualitative disorders.

A qualitative disorder can be either a pair of two elements or a triple of three elements in the *Qualitative Causal Model* $< H, Mn, Pt, In, Ds, Ob, R >$. Each type of disorder described above is represented as follows.

Type 1 $\{< p, q > | p \in In, q \in Q\}$

Type 2 $\{< p, r, q > | p \in Mn \cup Pt, r \in R, q \in Q\}, \{< p, q > | p \in Ds, q \in Q\}$

For example, $< b, [+] >$ $(b \in In)$ means input parameter b has the qualitative value $[+]$. $< b, r1, [+] >$ means that causal relation $r1$ has become abnormal because of a fault in a relevant plant device and this abnormality directly causes parameter b to have the qualitative value $[+]$.

These qualitative values affect other parameters via causal relations and as a consequence, the symptoms are detected by the sensors.

In the example of *QCNet* in Figure 2, the qualitative disorders are expressed as the nodes of the network.

2.1.3. How to Deduce with Model

In this section, we formalize the thought process of a skilled human operator. He diagnoses the cause using the *Qualitative Causal Model* described in section *2.1.2.*

In an unforeseen event, several symptoms with abnormal sensor data is observed.

Each symptom $s \in \{< p, q > | p \in Ob, q \in Q\}$ is connected to a set of qualitative disorders which has causal relations with it.

Qualitative propagation on *QCNet* starts with one symptom and gathers all the qualitative disorders which have a causal relation with the symptom.

The *Qualitative Propagation Module* calculates this set of qualitative disorders for all the input symptoms. Using these sets of disorders, all the possible combinations of the qualitative disorders which can explain all the symptoms are calculated in the *BIPARTITE Module*.

The calculated combination of qualitative disorders represents the faults which are occurring simultaneously. We call this combination *explanation*. There are plural possible *explanations* for the symptoms.

We take the following two steps to calculate all the *explanations* for the symptoms.

Step 1 Calculate the set of possible disorders for each symptom.
The *Qualitative Propagation Module* calculates the set $cause(s_i)$ for each symptom $s_i \in \{< p, q > | p \in Ob, q \in Q\}$. This $cause(s_i)$ is a set of possible disorders for symptom s_i. As a result, the collection of the sets $\{cause(s_i) | s_i \in \{< p, q > | p \in Ob, q \in Q\}\}$ is calculated.

Step 2 Calculate the *explanation* for $\{cause(s_i) | s_i \in OBS\}$
Calculate all the *explanations* for $\{cause(s_i) | s_i \in OBS\}$ calculated in Step 1. This calculation is executed in the *BIPARTITE Module* based on the set covering technique.

We briefly outline these two steps.

- Each qualitative disorder in *QCNet* has causal relations with several symptoms via the *qualitative sentences*. On the contrary, each symptom has causal relations with several qualitative disorders. The qualitative propagation on *QCNet* calculates these qualitative disorders for a given symptom. The propagation mechanism is based on a search.

  To avoid the problem of qualitative combinatorial explosion, we adopt a new semantic for propagation. This semantic does not consider any combinations. For example, the cause of $< a, [+] >$ when

  $$r : a \Leftarrow \{\{+ : b\}, \{- : c\}\} \qquad (r, a)$$

  is calculated as follows.

  $$cause(< a, [+] >) = \{< b, [+] >, < c, [-] >\}$$

  This equation means that the cause of $< a, [+] >$ is either $< b, [+] >$ or $< c, [-] >$. This semantic is acceptable in the sense that at least one of $< b, [+] >$ or $< c, [-] >$ is occurring.

  A function PROPAGATE which calculates $cause(< h, q >)$ on *QCNet* for symptom $< h, q >$ using this semantic is shown in Figure 3.

- In the *Qualitative Propagation Module*, the diagnostic problem is transformed into a simpler problem of $\{cause(s_i) | s_i \in \{< p, q > | p \in Ob, q \in Q\}\}$. *Set covering* calculates the minimal set of disorders which can explain all the symptoms using $\{cause(s_i) | s_i \in \{< p, q > | p \in Ob, q \in Q\}\}$ [12].

  The basic idea of *set covering* is to calculate all the possible classifications of the symptoms. All the symptoms are classified into several groups according to whether or not $cause(s)$ of each symptom in a group has common qualitative disorders. Each group satisfies following conditions.

  Let S_i $(1 \leq i \leq n)$ be a group of symptoms.

  Then each S_i satisfies

  $$\bigcap^{s_j \in S_i} cause(s_j) \neq \phi$$

  Let $G = \{S_1, S_2, \ldots, S_n\}$ be one possible classification of all the symptoms. We define disorders D_i for S_i as follows.

  $$D_i = \bigcap^{s_j \in S_i} cause(s_j)$$

  This D_i corresponds to one fault occurring. The existence of the symptoms in S_i can be explained by this fault. The number of the elements of G equals to

function propagate($h, q, QCNet$)
$D = \{\}$
begin
$Sent$ = qualitatlive sentences for h; : get relevant qualitative sentences
if $Sent \neq \phi$ **then** : if h is not input parameter
while $Sent \neq \phi$ **do** : calculate for all relations
$(r, h) = NextSentence$ from $Sent$;
$Para$ = cause parameters for $< h, q >$ in (r, h); : get cause from sentence (r,h)
while $Para \neq \phi$ **do** : get cause for all cause parameter
$(p, q') = NextPara$ from $Para$;
$D2 = propagate(p, q', QCNet)$; : recursive call on propagate/4
$D1 = D1 \cup D2$;
endwhile;
$D = D \cup \{< u, r, q >\} \cup D1$;
endwhile;
else $D = \{< h, q >\}$; : input parameter or design paramter
endif;
return D;
end.

Figure 3: *Qualitative Propagation*

the number of faults occurring simultaneously. Calculating one possible classification of the symptoms is equivalent to diagnosing one possible combination of disorders to explain the symptoms [20].

There may exist more than one possible classification which can explain all the symptoms. *Set covering* calculates all these possible classifications.

Peng and Reggia [12] justified the idea of *set covering* in "*Parsimonious Covering Theory*".

The BIPARTITE algorithm [12] calculates all the possible classifications of the symptoms incrementally using the $cause(s_i)$ $(s_i \in \{< p, q > | p \in Ob, q \in Q\})$.

The *BIPARTITE Module* utilizes the BIPARTITE algorithm.

2.1.4. *Diagnostic Strategy*

As described above, the *BIPARTITE Module* calculate all the possible classifications. In order to select the most probable classification from all possible classifications, a skilled human operator would consider the frequency of the occurring fault. He would adopt the *single fault strategy*.

The selected qualitative disorders are translated into *interpreted disorders*, which are directly related to the plant devices. For example, the qualitative disorder ($< pressure, [-] >$) is translated into the abnormal inside-pressure of the *condenser*. The disorder ($< parameter, r_{sensor}, [+] >$) is translated into the fault of the pressure-sensor.

2.2. *A Theory of Model-Based Generation of Plant Control Knowledge*

The main purpose of this section is to present a generation procedure of the operations-sequence, namely rule-based knowledge, which can be done by a skilled human operator using some fundamental knowledge.

2.2.1. *What to Express as Model*

A skilled human operator needs at least the following fundamental knowledge when he thinks about how to operate a plant when dealing with unforeseen events. We call this fundamental knowledge *model*.

- *model of plant structure*, which defines what devices the plant consists of and how each device is connected to the others.
- *model of plant functions*, which defines what functions the each component device has and how each function interacts.
- *model of plant characteristics*, which defines the relations among process parameters.
- *model of principles of plant operations*, which defines default rules to maintain safety, efficiency and economy in plant operations.

The former three models are related to the plant design, and the latter one is related to the plant operation task itself. We aggregate the former three into the *Device Model* and the latter one the *Operation Principle Model*. The representation of both models corresponds to the constraints to be satisfied in order to generate the knowledge for plant control.

1. *Device Model*

   The *Device Model* represents the fundamental knowledge about the functions, structure and characteristics of a plant. A plant consists of some component devices, so a *Device Model* can be defined for each component device. Figure 4 shows an image of the *Device Model* representation for a *boiler-feeding-water-pump*, which supplies water to a boiler.

   In the field of plant control, because demands for each component device are very significant, their values are described in the *demand* slot and their constraints

name :	a_bfp
demand :	a_bff = 360 [ton/hr]
goal :	a_bff =< capacity(a_bff)
states :	on ; capacity(a_bff) = 615 [ton/hr] off ; capacity(a_bff) = 0 [ton/hr]
operation :	off → on ; time-lag = 0.1 [hr] , d/dt(a_bff) = + on → off ; time-lag = 0.1 [hr] , d/dt(a_bff) = -
quality :	d/dt(a_bff) = d/dt(a_bflf)
flow_in :	*(defined at system)*
flow_out :	*(defined at system)*
system :	bfp_system(a_bff , a_bflf)

Figure 4: An image of the *Device Model* representation

to be satisfied are described in the *goal* slot. Their values are determined not only according to the functions and structure of a plant but also by the intents of a human operator. We call the intents top-level demands.

Each component device has some functions to meet these demands. These functions are described as possible states of each device in the *states* slot.

The operations of a device are defined by the change of its state, which is described in the *states* slot. Direct and indirect influences to plant processes by operations are described in the *operation* and *quality* slots respectively.

As for the structure of a plant, the connections of devices are defined according to each process flow. These connections are described in *flow_in* and *flow_out* slot. In addition, the hierarchy of devices can be defined as shown in Figure 5 to enable efficient modeling.

2. *Operation Principle Model*

The *Operation Principle Model* represents the principles for safe and efficient plant control. It consists of the following two rules.

(a) *Strict Accordance Rule*

It is the rule to maintain safety in plant operations. It consists of the following two rules: the rule to use a device within its own allowable limit constraints, and the rule to keep a damaged device out of service.

(b) *Preference Rule*

It is the rule to maintain efficiency and economy in plant operations. It consists of the following two rules: the rule to keep the number of devices

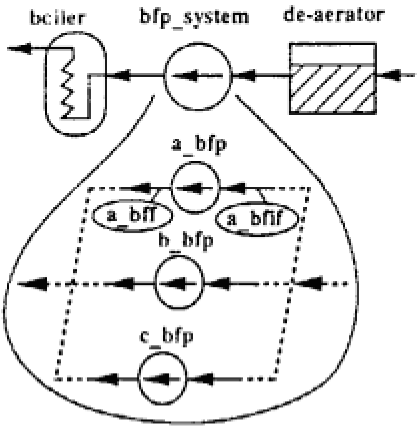

Figure 5: Hierarchical representation of the *Device Model*

in service to a minimum, and the rule to equalize the service-time of each device.

2.2.2. *What to Deduce with Model*

After generating *interpreted disorders* of an unforeseen event via the diagnostic phase, a skilled human operator would think about a target plant state, where he would like to bring the plant to deal with it. We call this target plant state a *goal-state*. A *goal-state* should be decided relating to the following items: interpreted disorders, the current plant state, the constraints defined by the *Device Model* and the *Operation Principle Model*, and the intents of a human operator.

After deciding a *goal-state*, he could easily know the operations to bring the plant from a current state to the *goal-state*. That is, these operations are easily derived from the difference between the current plant state and the *goal-state*.

In addition to knowing the operations, he should also know when to execute them. He should know a set of the preconditions to specify the timing to execute them. After knowing both the operations and their preconditions, he could get the operations-sequence, that is, a set of rule-based knowledge for plant control.

To summarize the above, the significant points are to decide the *goal-state* and to generate the operations-sequence.

2.2.3. *How to Deduce with Model*

In this section, we are going to formalize how a skilled human operator decides the goal-state and generate the operations-sequence.

1. Decision of the *Goal-State*

We can assume that deciding the *goal-state* corresponds to generating a new plant state where all of the constraints are satisfied. These constraints are defined by the models and redefined by faults of devices. This assumption makes sense on the following conditions: faulty devices must be out of service; the mapping from the *interpreted disorders* to the new state of their related devices is clear; the goal-state is selected to be as near to the current state as possible; and the intents of a human operator can be represented as the demands for devices.

The search for this new plant state should be done step by step for the reason of efficiency. A skilled human operator would try to get the goal-state by combining the local state of each device and by propagating it to the others. Therefore, an intelligent diagnostic and control expert system should have the following functions [14][19].

(a) Verification of Constraints

To generate the *goal-state* according to an unforeseen event, all the constraints defined by the *Device Model* should be verified to be still satisfied after the unforeseen event, because the unforeseen event often causes the change of plant state or parameter's value. This function (Figure 6) consists of the following two subfunctions: propagating the change according to the connections of devices, and verifying locally the constraints on each device.

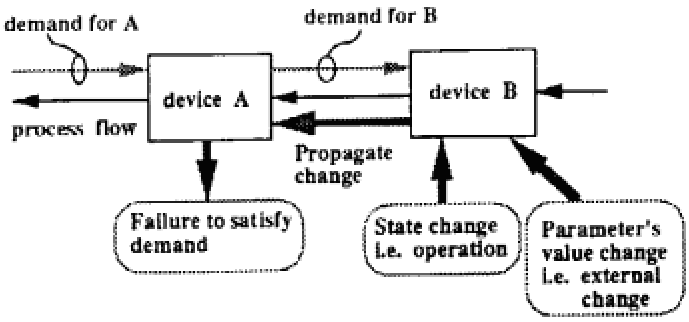

Figure 6: Constraints verification function

(b) Update of the *Goal-State*

If some of the constraints defined for a certain device by the *Device Model* are proved not to be satisfied, a new state for this device is searched to satisfy these constraints. This function (Figure 7) consists of the following subfunctions: searching the state of each device where all of its demands can be satisfied, distributing the demands for a device of higher hierarchy to devices of lower hierarchy according to the constraints defined by the *Operation Principle Model*, and generating new demands for connected devices according to the *Device Model* and propagating them.

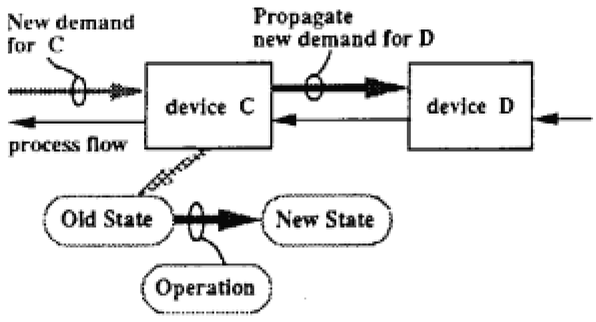

Figure 7: *Goal-State* update function

If an unforeseen event occurred, a skilled human operator would verify the constraints according to the *interpreted disorders*. If unsatisfied constraints were detected, he would update the *goal-state*, and as a result, generate a new *goal-state*.

2. Generation of the Operations-Sequence

In the domain of a thermal power plant control, the operations of devices need the following five generic preconditions, which are classified based on interviews with the experts [8].

(a) Precondition for the state before an operation

This precondition specifies the devices have not operated yet. The specified devices are both the target device to be operated and the one connected in parallel to it.

(b) Precondition for the order of operations

There is a principle about the order of plant operations. That is, operations to activate devices should be executed in the direction of process flow, and the ones to halt them should be in the reverse direction. This precondition specifies this principle.

(c) Precondition for safety during an operation

If operations are executed, the change of process occurs and influences several devices via their connections. This precondition specifies these influences to be within an allowable limit.

(d) Precondition for the timing of an operations

To keep the operations efficient, their executions should be delayed as long as possible. Moreover, to keep the operations safe, their executions must be delayed till they no longer cause any bad influences on the plant. This precondition specifies the timing of operations.

(e) Precondition for completions of an operation

All operations must be confirmed to be executed completely. This precondition specifies this confirmation by providing the results of their execution.

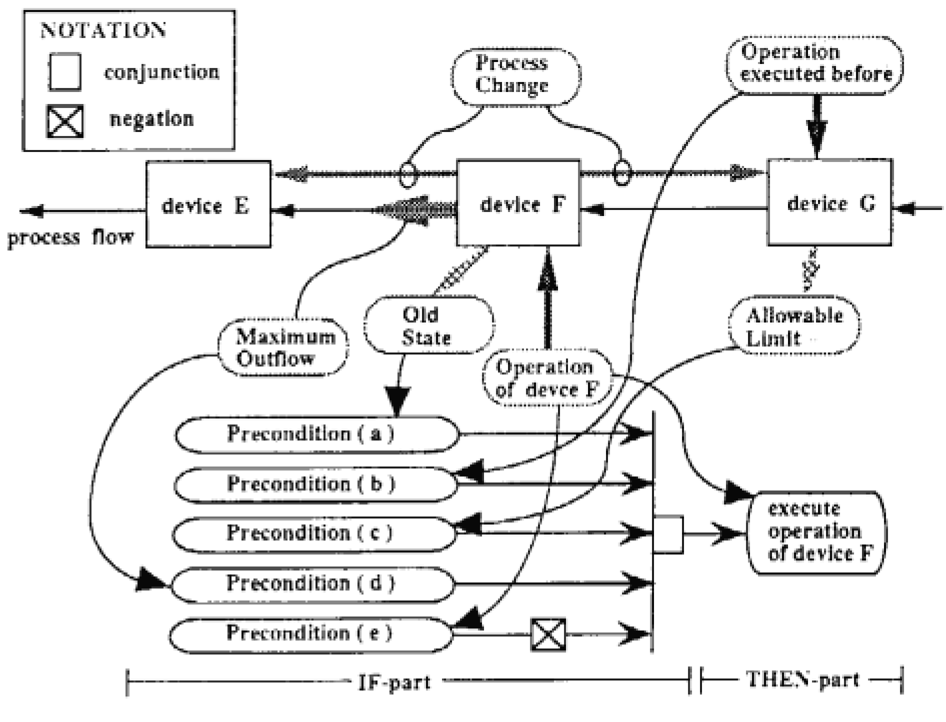

Figure 8: Generation process of preconditions of a plant operation

A skilled human operator could generate the above preconditions by analyzing the goal-state according to the constraints defined by the *Device Model*. We are going to explain this generation process below using Figure 8. Preconditions (a),(b) and (e) are easily generated. Precondition (c) can be generated according to the constraints defined by the *Device Model*, namely the *operation* and *quality*

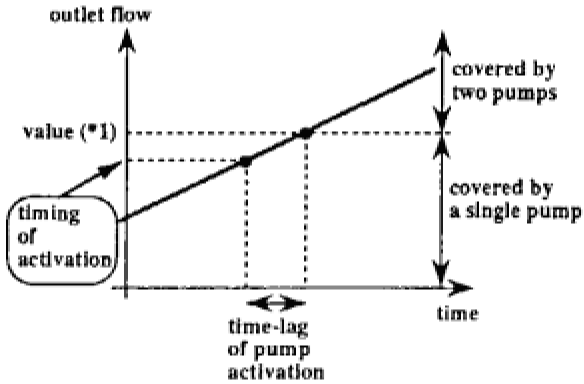

Figure 9: The timing of activation of a supplementary device

slots filler. For example, the activation of a pump would increase the outlet flow of water from a tank, so the water level of the tank would drop. Therefore, to execute this operation, the water level must be high enough so as not to drop below its allowable minimum limit. The threshold value of this precondition is calculated to be the minimum limit plus some margin. Precondition (d) is generated according to the maximum process outflow of the target device. For example in Figure 9, the total volume of outlet flow of water by a pump-system cannot exceed the value (*1) unless a supplementary pump is activated. Conversely, the timing of this activation is determined according to both this value (*1) and the time lag of this activation.

2.3. *A Theory of Model-Based Verification of Plant Control Knowledge*

As explained before, generating the operations-sequence is deeply based on the *goal-state*. Because the *goal-state* corresponds to the state at a given point in time, the models explained so far are based on static relations among process parameters and represent no time concepts. Generally speaking, time concepts are very important especially in the area of plant control. For this reason, the dynamic relations among process parameters must be taken into consideration.

The main purpose of this section is to present the theoretical aspect of verifying the generated operations-sequence by the fundamental knowledge.

2.3.1. *What to Express as Model*

A dynamic system, for example a plant, has transient changes in response to some external action on it due to its dynamic characteristics. For this reason, a skilled human operator would predict the plant behavior when he executed the generated operations-sequence. Therefore, he must know the dynamic characteristics of the plant, which we call the *Dynamics Model*. In the area of plant control, the *Dynamics Model* consists of the following two models.

1. Model of plant controllers

   This model represents the functions of traditional plant controllers based on PID control. The model of a water-level-controller of a *de-aerator* is described as Figure 10. A *de-aerator* is a kind of water tank to extract oxygen from the feeding water to the boiler.

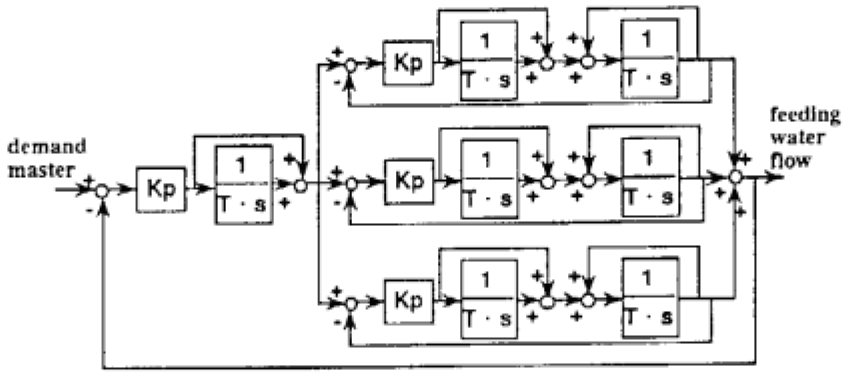

Figure 10: An example of the *Dynamics Model* of a plant controller

2. Model of plant processes

   This model represents the characteristics directly related to physical phenomena, e.g., the characteristics between temperature and pressure of gas, the characteristics of heat exchanging between steam and water, and so on. The model of the inside pressure of a *condenser* is described in Figure 11. A *condenser* is a cooling device of exhaust steam from the turbine.

2.3.2. What to Deduce with Model

Needless to say, the prediction of the plant behavior by the *Dynamics Model* is necessary to verify the generated operations-sequence. After this prediction, a skilled human operator would examine whether or not undesirable events occurred. The undesirable events can be defined by several criteria, but one of the most important ones is the transient violation of limit-constraints for each process parameter's value. The execution of plant operations usually causes the transient change of processes due to the dynamic characteristics of a plant, and if this change is beyond the allowable range of a current plant state, it is to be detected as a violation of limit-constraints. He could deal with the violation according to the degree of the violation.

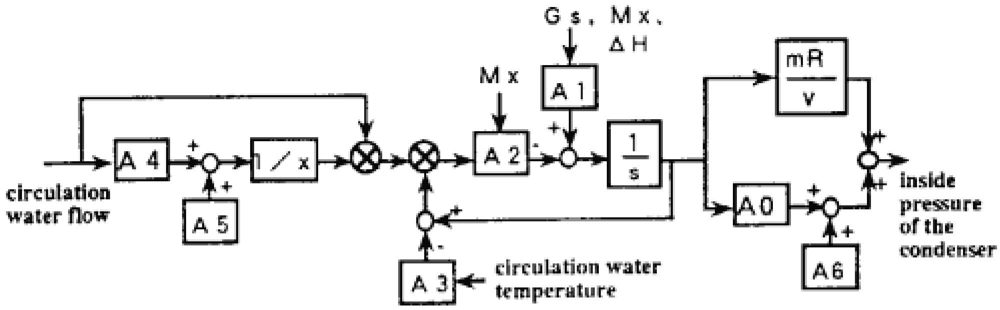

Figure 11: An example of the *Dynamics Model* of a plant process

2.3.3. *How to Deduce with Model*

The prediction of a plant behavior can be realized by simulation methods. When simulating it, a skilled human operator would not always predict the exact numeric value of parameters. That is, he would often predict only the qualitative trend. To get the degree of the violation of limit-constraints requires the numeric information, but to get the approximate trend of a plant behavior requires only the qualitative information.

Therefore, an intelligent diagnostic and control expert system can adopt the following method: numeric simulation to get the exact predictions, qualitative simulation [10][2] [5] to get the approximate trend, and fuzzy-based simulation [15][9] as the intermediate approach between the former two.

When determining the violation of limit-constraints in the prediction, a skilled human operator would try to deal with it according to the degree of the violation. This process can be formalized as updating the goal-state according to this degree. Therefore, an intelligent diagnostic and control expert system should support the Generate&Test algorithm of the operations-sequence [15][16] as illustrated in Figure 12.

3. An Application of Model-Based Diagnostic and Control Expert System

This section describes a practical application of an intelligent diagnostic and control expert system.

3.1. *Configuration of The Expert System*

The model-based diagnostic and control expert system (Figure 13) consists of two subsystems: the *shallow inference subsystem* (*SIS*) and the *deep inference subsystem*

```
procedure Generate&Test ( M or D0 , S0 )
  begin
      [ Se, Op ] <= Operation_Generate ( M or D0, S0 ) ;
               K1 <= Precondition_Generate ( S0, Se, Op ) ;
               PS <= Simulate ( S0, K1 ) ;
    [ NG, D1, S1 ] <= Verify ( PS ) ;
    if NG =\= constraint_violation
    then return( K1, Se ) ;
    else
         [ K2, S3 ] <= Generate&Test ( D1, S1 ) ;
         [ K3, SE ] <= Generate&Test ( M, S3 ) ;
               K4 <= FIX( K1 ) + K2 + K3 ;
         return( K4, SE ) ;
    endif
  end.
```

NOTATION :

Si, Se, SE : plant state	M : interpreted disorder
Di : demand for a device	Op : plant operations
PS : plant behavior	NG : flag for constraint-violations
Ki : knowledge for plant control	
[ ] : list expression	<= : substitution expression

Figure 12: Generate&Test algorithm of the operations-sequence

(*DIS*). To perform the 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, namely the shallow knowledge for plant control. It selects and executes plant operations according to the heuristics stored in the knowledge base. The plant monitor detects occurrence of an unforeseen event, and then activates the *DIS*.

The *DIS* utilizes various kinds of models to realize the thinking process of a skilled human operator and generates the operations-sequence to deal with the unforeseen events. It consists of the following modules: the *Diagnosor*, the *Operation-Generator*, the *Precondition-Generator*, and the *Simulation-Verifier*.

The *Diagnosor* utilizes the *Qualitative Causal Model* about plant process parameters to deduce the *interpreted disorders* causing the unforeseen event.

The *Operation-Generator* generates the *goal-state* to deal with the unforeseen event, and then generates the operations-sequence. It utilizes the *Device Model* and the *Operation Principle Model*.

The *Precondition-Generator* generates the preconditions of each generated operation by the *Device Model*, and as a result, generates rule-based knowledge for plant control.

The *Simulation-Verifier* predicts the plant behavior using the *Dynamics Model* when the generated operations-sequence is executed, and then checks whether or not undesirable events occur in the prediction. If they occur, it gives feedback to the *Operation-Generator* to deal with them.

The generated and verified operation-sequence by the *DIS* is transmitted to the *SIS*. The *SIS* executes the plant operations according to it, and as a result, the unforeseen event would be covered.

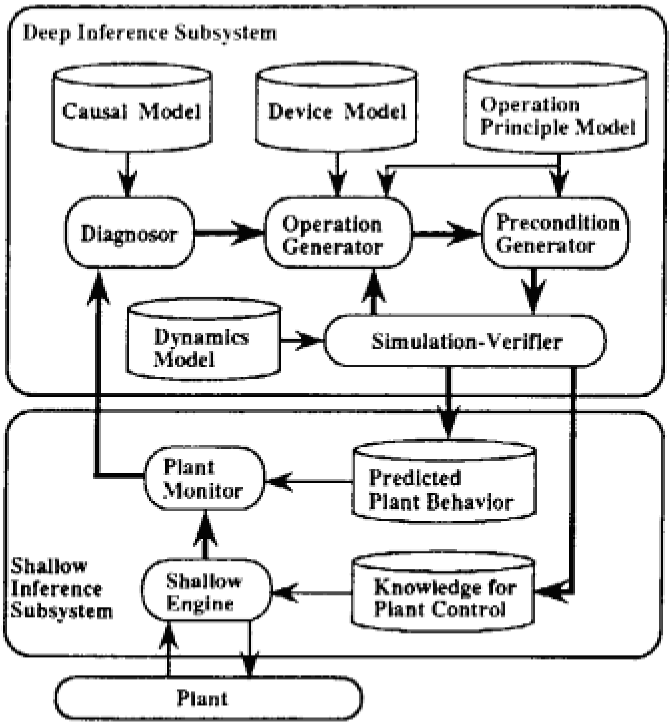

Figure 13: Configuration of the model-based diagnostic and control expert system

3.2. Configuration of The Target Thermal Power Plant

Figure 14 shows the water feeding system of the target thermal power plant to control. The *condenser* is a device for cooling the *turbine*'s exhaust steam; the steam is reduced to water using cooling water taken from the sea. The reduced water is transmitted through the *de-aerator* to the *boiler* by the *condensation-pump-system* and the *boiler-feeding-pump-system*. The cooling water is provided by the *circulation-pump-system*. The *air-ejector* extracts the air inside the *condenser* to facilitate the inflow of the exhaust steam from the *turbine*.

3.3. Experiments

We have implemented the expert system on Multi-PSI [18], which is the parallel inference machine developed by the Institute for New Generation Computer Technology

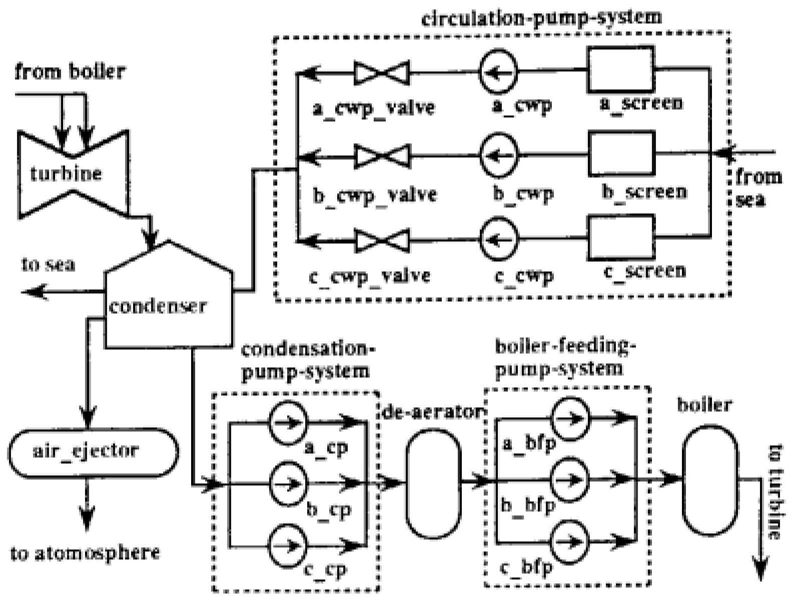

Figure 14: Configuration of the target thermal power plant

(ICOT). To realize a rich experimental environment, we have also implemented a plant simulator instead of an actual plant on G8050, which is the mini-computer developed by Toshiba Corporation. Both computers are linked by a data transmission line as described in Figure 15. We performed some experiments with the system to estimate the ability of model-based reasoning. The experiments were performed along with the following procedure.

- First, we selected the appropriate disorders of the plant. We set up them as malfunctions of the plant simulator.
- Next, we extracted the specific knowledge for plant control from the knowledge base in the *SIS*. This specific knowledge was necessary to deal with the selected disorders. As a result, the selected disorders were equal to unforeseen events.
- Finally, after activating the malfunctions of the plant simulator, we confirmed that the DIS generated the operations-sequence and the SIS executed the operations according to it.

This section shows the results of two experiments. One is the unforeseen event caused by the external change of a plant, and another is the one caused by the internal change. Because any disorder belongs to either cases of above experiments, we believe that the selected two experiments are enough to discuss the ability of the system.

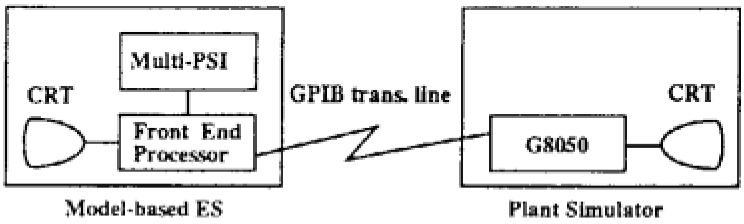

Figure 15: Configuration of the experimental environment

3.3.1. Unusual Increase of The Water Temperature of The Sea

This case is an example of the unforeseen events caused by the external change of a plant. In this section, we focus on the *DIS*.

1. Diagnostic phase

   The qualitative causal network for the *condenser* model is shown in Figure 16.

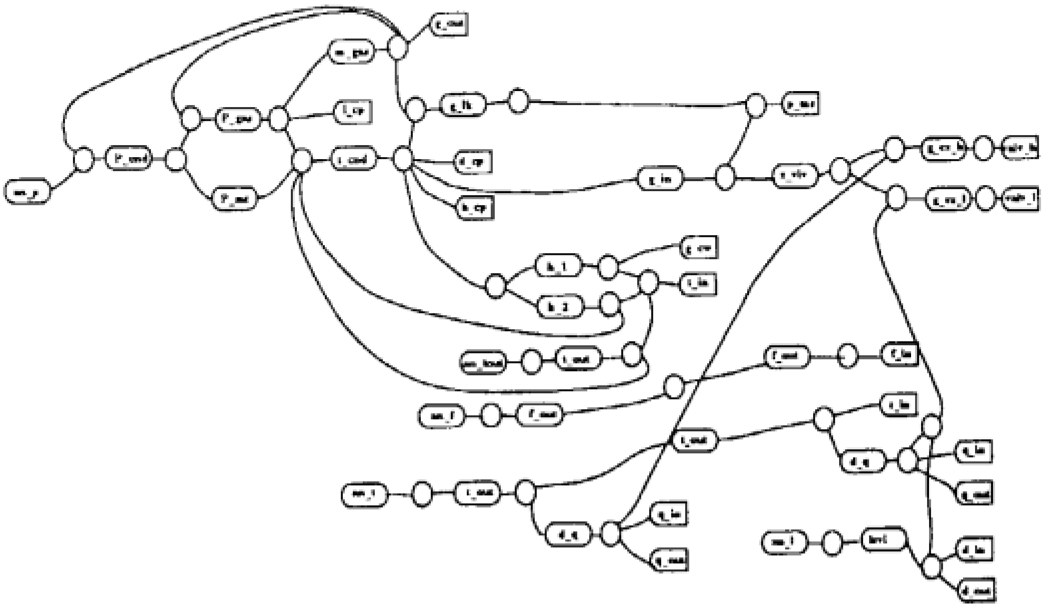

Figure 16: QCNet for the *condenser*

The abnormal increase of the pressure in the *condenser* and abnormal increase of the outlet temperature of the cooling water were detected. The qualitative propagation module calculated the disorders for each symptom, and the

BIPARTITE module calculated the *generator set*. Using *single fault strategy* the increase of the inlet temperature of the cooling water was selected as the interpreted disorder.

2. Generation phase of operations-sequence

The *Operation-Generator* was activated to verify the constraints for the *condenser* because the inlet temperature of the cooling water influenced the function of the *condenser*. As a result, it generated three operations as indicated under the hatched symbols of devices in Figure 17.

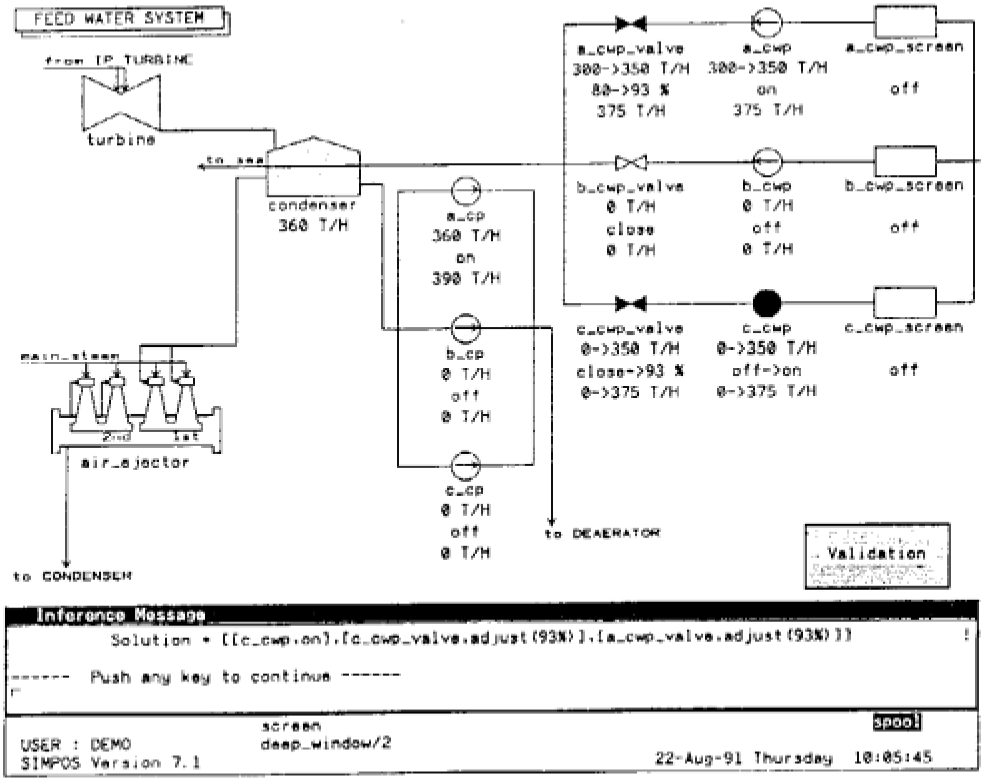

Figure 17: Hardcopy of the output by the *Operation-Generator*

The *Precondition-Generator* deduced the preconditions for each operation to generate the rule-based knowledge for control. Figure 18 is a logic-chart description of one of the generated knowledge.

The *Simulation-Verifier* predicted and verified the plant behavior, but in this case, no violations caused by the dynamic characteristics of the plant were detected.

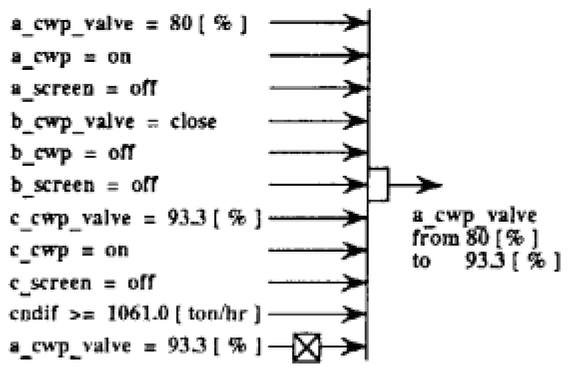

Figure 18: An example of generated knowledge for control

The *SIS* could deal with this unforeseen event by executing the plant operations according to the generated and verified operations-sequence. We confirmed the expected result via the CRT of the plant simulator.

3.3.2. Unforeseen Fault of The Boiler-Feeding-Pump

This case is an example of the unforeseen events caused by the internal change of a plant, usually faults of devices. In this section, we also focus on the *DIS*.

1. Diagnostic phase

   The abnormal decrease of inlet water flow rate in the *boiler* and the abnormal decrease of outlet water flow rate in the *boiler-feeding-pump-system* were detected. The common disorder was calculated in the same way as in the *condenser* example. As a result, the fault of the *boiler-feeding-pump* was selected as the *interpreted disorder*.

2. Generation phase of operations-sequence

   As in the another case, the *Operation-Generator* and the *Precondition-Generator* generated the knowledge to switch from the faulty pump to one of the two other sound pumps.

   The *Simulation-Verifier* predicted the plant behavior in Figure 19 when the switching operations were executed. The initial plant state S0 persisted till time 10 and the *goal-state* Se was reached at time 92 after the operations, when all processes settled down. All the parameters' values in this figure are regularized to take the value 1.0 [p.u.] when the plant is fully loaded. Here, we define the parameter QC to be the total outflow of the *condensation-pump-system*. The value of QC was over 0.5 [p.u.] from time 14 to 33. The maximum outflow of each *condensation-pump* was designed to be 400 [ton/hr], namely 0.5 [p.u.] and only one pump was in service throughout from S0 to Se. For this reason, the maximum possible value of QC, which is the total outflow of all the *condensation-pump*, was 0.5 [p.u.]. Therefore, the violation of the limit-constraint for QC occurred from time 14 to 33. In this figure, the maximum value of QC was 0.57 [p.u.] equal to 456 [ton/hr], at time 28. According to the algorithm of Figure 12, the *Simulator-Verifier* selected this value as the new demand for QC and gave the feedback to the *Operation-Generator* with the initial state S1 just before time 14. As shown in Figure 20, the *Operation-Generator* generated the *intermediate goal-state* S3 where this new demand was satisfied, and as a result, the knowledge was generated to activate another *condensation-pump* to deal with that violation. According to the algorithm of Figure 12, the operations-sequence was generated completely by the *DIS*. It consisted of the following operations: switching from the faulty *boiler-feeding-pump* to the sound *boiler-feeding-pump* connected in parallel, and temporally activating the additional *condensation-pump*.

The *SIS* could deal with this unforeseen event by executing the plant operations according to the generated and verified operations-sequence. We also confirmed the expected results via the CRT of the plant simulator as well.

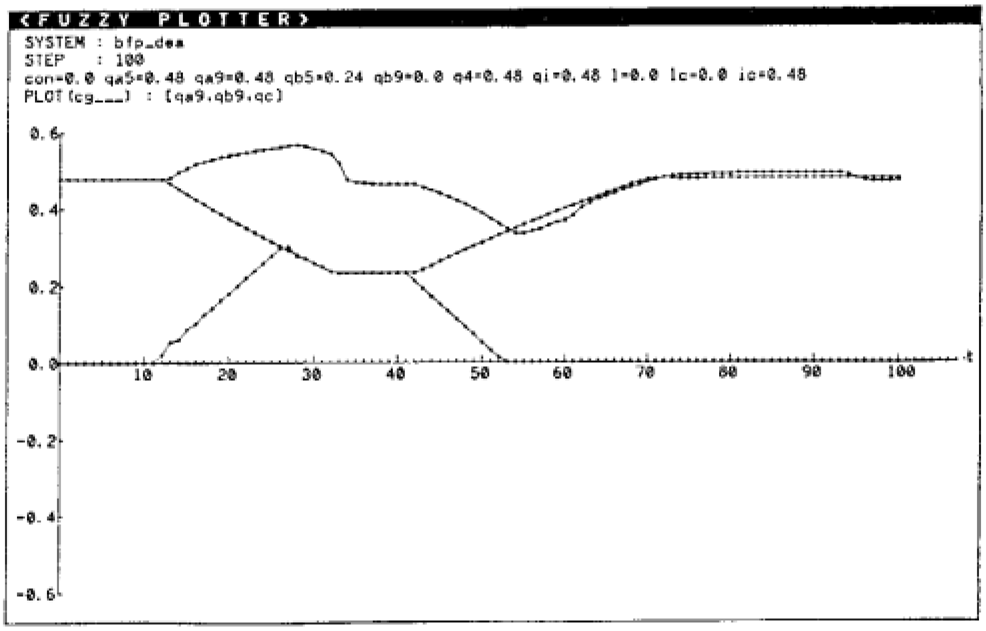

Figure 19: An example of predicted plant behavior

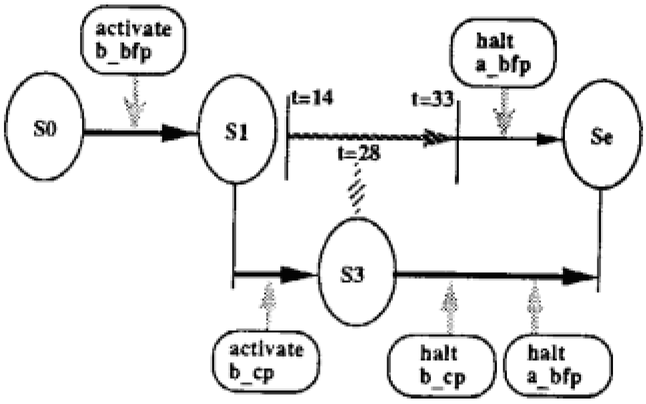

Figure 20: An image of the output by the Generate&Test algorithm

3.4. Discussions

We have confirmed by some experiments the ability and utility of the model-based diagnostic and control expert system. Although model-based reasoning is one of the significant mechanisms to realize an intelligent diagnostic and control expert system, the following open problems must be also resolved for its practical use.

1. Facility of model acquisition

   The utilized models in the system, namely the *Qualitative Causal Model*, the *Device Model*, the *Operation Principle Model* and the *Dynamics Model*, could be built from the plant design, and should be consistent with each other. In the current implementation of the system, each model is built and implemented without any relation to the others. Therefore, the real model sharing is not realized now.

   To support the consistent acquisition and utilization of the models, we should have a framework to support the maintenance and transformation of the shared model. As for this problem, we believe that the modeling aspect of QPT [6] is a useful way to resolve it.

   Some related works are in the area of the qualitative reasoning. Crawford [1] attempted to maintain and support the qualitative modeling environment by QPT. The research on hierarchical modeling is described in Falkenhainer [4] and Yoshida [21].

2. Over-sensitive verification of the plant behavior

   In the current implementation of the Generate&Test algorithm of the operations-sequence, the priority or importance of each limit-constraint is not considered at all. So, even though the violation of these constraints is slight enough to be ignored, this algorithm sensitively tries to deal with it. This sensitivity is meaningless for all practical purposes because the plant would be designed with enough capacity to absorb it. For this reason, the system should check the constraints with some allowable degree of violations. We are now investigating the mechanism to check the following limit-constraints: the constraint whose limit value is variable, and the one whose violation is allowable in some period of time depending on the degree of the violation.

3. Monitoring the execution of the generated operations-sequence

The *Diagnosor* selects one classification of the observed symptoms using the *single fault strategy*. However, this classification can be invalid. This validity should be estimated by the plant monitoring after the plant operations.

The *DIS* generates the operations-sequence according to the outputs of the *Diagnosor*, and the operations are executed by the *SIS*. Monitoring whether the output of the *Diagnosor* is valid or not is important. If they are not valid, another diagnosis must be started. In the new diagnosis, the executed plant operation must be taken into consideration. This information should be used in the selection of possible classifications.

As for the related work, Dvorak [3] utilizes the QSIM [10] to monitor a plant. His research is important because it is clear that the system's ability to build a fault model is directly related to the one to monitor a plant. Because the task of his research is the plant monitoring, he does not refer to the generation of the operations-sequence for unforeseen events.

4. Real-time reasoning by the DIS

The target task of our developing system is not PID control nor adaptive control in the area of the traditional control theory but sequence-control. Although our system is not required of the severe real-time reasoning capability to cover either PID control or adaptive control, it is at least required of the ability to complete the reasoning in a few ten second. To guarantee this performance, we have been investigating the parallel reasoning mechanism with a parallel computer [17]. The target points of parallel reasoning are as follows:

- Because the *BIPARTITE algorithm* implemented in the *Diagnosor* generates the *interpreted disorders* incrementally, each generation step can be calculated in pipeline manner.
- Because the reasoning at the *Operation-Generator* is based on the each local computation, it can be done in parallel by assigning a processor element to local computation for each device. .

- Because the *Precondition-Generator* generates five types of preconditions for each operation, each precondition can be generated in parallel by assigning a processor element to each type of each operation.
- The *Simulation-Verifier* predicts the plant behavior by utilizing the *Dynamics Model*. Because the *Dynamics Model* can be usually divided into some calculation parts, simulation can be performed in parallel or in pipeline manner by assigning a processor element to each area.

We have demonstrated a five times improvement of reasoning time by using 16 processor elements. We use the MIMD type parallel computer called Multi-PSI developed by ICOT. To achieve more improvement, we must investigate a better parallel algorithm.

4. Conclusion

In this paper, we have discussed a intelligent diagnostic and control expert system based on plant model from the viewpoint of both its theoretical and practical aspects. The main target of our approach is a system which could deal with unforeseen events. Our approach adopts a model-based architecture to realize the thinking process of a skilled human plant operator.

This proposed architecture can support the integrated framework which could generate the *interpreted disorders* of unforeseen events and could generate and test the operations-sequence to deal with them. It utilizes the following models : the *Qualitative Causal Model*, the *Device Model*, the *Operation Principle Model*, and the *Dynamics Model*.

In this architecture, the generation of the *interpreted disorders* is performed by qualitative propagation using the *Qualitative Causal Model*. The generation of the operations-sequence is performed by the constraints satisfaction and analysis method using the *Device Model* and the *Operation Principle Model*. The test of the operations-sequence is performed by the prediction of the plant behavior using the *Dynamics Model*.

In the domain of plant control, model-based diagnosis or operations guidance is about to be used practically. We believe the model-based reasoning to be significant and essential as the basis of a intelligent diagnostic and control expert system.

Acknowledgements

The authors thank Toshiba Corporation for supporting this research project in the Systems & Software Engineering Laboratory. Thanks also to other researchers of this project. This research is carried out under auspices of the Institute for New Generation Computer Technology (ICOT).

References

[1] Crawford, J., Farquhar, A. and Kuipers, B. *QPC: A Compiler from Physical Models into Qualitative Differential Equations*, Proc. of AAAI-90, pp.365–372 (1990).

[2] deKleer, J. and Brown, J. S. *A qualitative physics based on confluence*, Artificial Intelligence, vol.24, pp.7–83 (1984).

[3] Dvorak, D. and Kuipers, B. *Model-Based Monitoring of Dynamics Systems*, Proc. of IJCAI-89, pp.1238–1243 (1989).

[4] Falkenhainer, B. and Forbus, K. D. *Setting up Large–Scale Qualitative Models*, Proc. of IJCAI–89, pp.301–306 (1989).

[5] Forbus, K. D. *The qualitative process engine*, Reading in Qualitative Reasoning about Physical Systems, Weld,D. and deKleer,J.(Eds.), Morgan Kaufmann, pp.220 235 (1989).

[6] Forbus, K. D. *Qualitative Process Theory*, Artificial Intelligence **24**, pp.85–168 (1984).

[7] Hart, P. E. *Direction for AI in the eighties*, SIGART, **79**, pp.79 (1982).

[8] Konuma, C., et. al. *Deep Knowledge based Expert System for Plant Control – Development of Conditions Generation Mechanism of Plant Operations –*, Proc. of 12th Intelligent System Symposium, Society of Instrument and Control Engineers, pp.13–18 (1990) (in Japanese).

[9] Konuma, C., et al. *Plant Control Expert System Coping with Unforeseen Events*, Proc. of 40th Annual Convention, Information Processing Society of Japan, pp.298–299 (1990) (in Japanese).

[10] Kuipers, B. *Qualitative Simulation*, Artificial Intelligence, 29, pp.289–338 (1986).

[11] Michie, D. *High-road and low-road programs*, AI Magazine, vol.3, no.1, pp.21–22 (1982).

[12] Peng, Y. and Reggia, J. A. *Basics of Parsimonious Covering Theory in Abductive Inference Models for Diagnostic Problem-Solving*. Springer-Verlag, pp.49–98 (1990).

[13] Sone, H., et al. *Hierarchically integrated digital control system for the Shin-Onoda Power Station*, Journal of Thermal and Nuclear Power Engineering Society vol.87 no.2 (1985) (in Japanese).

[14] 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.153–157 (1989) (in Japanese).

[15] Suzuki, J., et al. *Plant Control Expert System Coping with Unforeseen Events – Model-based Reasoning Using Fuzzy Qualitative Reasoning –*, Proc. of Third International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-90), ACM, pp.431–439 (1990).

[16] Suzuki,J., et al. *Deep Knowledge based Expert System for Plant Control – Integration of Deep Inference Mechanism and Knowledge Estimation Mechanism –*, Proc. of 11th Knowledge and Intelligent System Symposium, Society of Instrument and Control Engineers, pp.7–12 (1990) (in Japanese).

[17] Suzuki, J., et al. *Plant Control Expert System on Multi-PSI Machine*, Proc. of KL1 Programming Workshop, pp.101–108 (1991) (in Japanese).

[18] Taki, K. *The parallel software research and development tool: Multi-PSI system, Programming of Future Generation Computers*, North-Holland, pp.411–426 (1988)

[19] 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).

[20] Wu, T. D. *Efficient diagnosis of multiple disorders based on a symptom clustering approach*, Proc. of the Eighth National Conference on Artificial Intelligence , pp.357–364 (1990).

[21] Yoshida, K., et al. *An Approach to Hierarchical Qualitative Reasoning – Constructing Shallow Knowledge from Deep Knowledge –*, Journal of Japanese Society for Artificial Intelligence, vol.4, no.4, pp.447–455 (1989) (in Japanese).