PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1992,
edited by ICOT.

A Diagnostic and Control Expert System Based on a Plant Model

Junzo SUZUKI* Chiho KONUMA Mikito IWAMASA Naomichi SUEDA
Systems&Software Engineering Laboratory, Toshiba Corporation
70, Yanagi-cho, Saiwai-ku, Kawasaki 210, Japan

Shigeru MOCHIJI Akimoto KAMIYA
Fuchu Works, Toshiba Corporation
1, Toshiba-cho, Fuchu 183, Japan

Abstract

A conventional expert system for plant control is based on heuristics, which are a priori knowledge stored in a knowledge base. Such a system has a substantial limitation in that it cannot deal with "unforeseen abnormal situations" in a plant due to the lack of heuristics. To realize a flexible plant control system which can overcome this limitation, we focus on model-based reasoning. Our system has three major functions: 1) model-based diagnosis for unforeseen abnormal situations, 2) model-based knowledge generation for plant control, and 3) knowledge-based plant control both with generated and a priori stored knowledge.

In this paper, we focus on the function of model-based knowledge generation. First, we show an overview of our system which has an integrated architecture of deep reasoning with shallow reasoning. Next, we explain the theoretical aspects of model-based knowledge generation. Finally, we show the experimental results of our system, and discuss the system's capabilities and some open problems.

1 Introduction

Currently in the field of diagnosis and control of thermal power plants, the more intelligent and flexible systems become, the more knowledge they need. Conventional diagnostic and control expert systems are based on heuristics stored a priori in knowledge bases, so they cannot deal with unforeseen abnormal situations in the plant. Such situations could occur if knowledge engineers forgot to implement some necessary knowledge.

A skilled human operator is able to operate the plant and somehow deal with such unforeseen abnormal situations because he has fundamental knowledge about the structure and functions of component devices of a plant, the principles of plant operations, and the laws of physics. His thought process is as follows.

- Diagnosis of an unforeseen abnormal situation
- Generation of plant control knowledge
- Verification of generated knowledge

A skilled human operator can deal with unforeseen abnormal situations by repeatedly executing these steps using the fundamental knowledge mentioned before. Therefore, the concepts of our diagnostic and control expert system are based on the same steps.

In this paper, we focus on the generation and verification of plant control knowledge. First, we show an overview of our system. Next, we explain the model representations and the model-based reasoning mechanisms. After that, we describe the experimental results and discuss the system's capabilities. Finally, we discuss some open problems and related work.

2 A System Overview

The model-based diagnostic and control expert system (Figure 1) consists of two subsystems: the *Shallow Inference Subsystem (SIS)* and the *Deep Inference Subsystem (DIS)*.

The *SIS* is a 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 occurrences of unforeseen abnormal situations, and then activates the *DIS*.

The *DIS* consists of the following modules: the *Diagnosor*, the *Operation-Generator*, the *Precondition-Generator*, and the *Simulation-Verifier*. The *Diagnosor* utilizes the *Qualitative Causal Model* for plant process parameters to diagnose unforeseen abnormal situations. The *Operation-Generator* figures out which plant operations are necessary to deal with these unforeseen abnormal situation. It utilizes the *Device Model* and the

*Email:suzuki%ssel.toshiba.co.jp@uunet.uu.net

Operation Principle Model. The *Precondition-Generator* attaches the preconditions to each plant operation above, and as a result, generates rule-based knowledge for plant control. The *Simulation-Verifier* predicts plant behavior which is to be observed when the plant is operated according to the generated knowledge. It utilizes the *Dynamics Model*, verifies the generated knowledge using predicted plant behavior, and gives feedback to the *Operation-Generator* to refine the knowledge if necessary.

The knowledge compiled from models by the *DIS* is transmitted to the *SIS*. The *SIS* executes the plant operations accordingly, and as a result, the unforeseen abnormal situations should be handled properly.

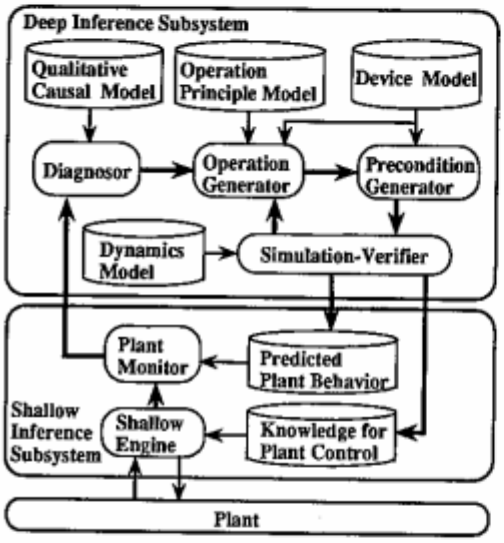

Figure 1: An overview of the system

3 Model-Based Generation and Verification of Knowledge

The main purpose of this section is to present a generation and verification procedure for plant control knowledge to deal with unforeseen abnormal situations. This knowledge is in IF-THEN format.

3.1 Model Representation

The *Device Model* and the *Operation Principle Model* are used to generate the knowledge. The *Dynamics Model* is used to verify the knowledge. We explain these models briefly.

1. Device Model

   The *Device Model* represents the fundamental knowledge about the functions, structure and characteristics of a plant. Because a plant consists of component devices, a *Device Model* can be defined for each component device. Figure 2 shows the *Device Model* representation for a *boiler-feeding-water-pump*, which supplies water to a boiler.

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_bfif)**
flow_in :	***(defined at system)***
flow_out :	***(defined at system)***
system :	**bfp_system(a_bff , a_bfif)**

Figure 2: An example of the Device Model

The demands for each component device are described in the **demand** slot, and their constraints to be satisfied are described in the **goal** slot.

The functions of each component device 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.

Direct and indirect influences to plant processes by operations are described in the **operation** and **quality** slots respectively.

The structure of a plant is described in the **flow_in** and **flow_out** slots. In addition, hierarchical modeling can be done as shown in Figure 3.

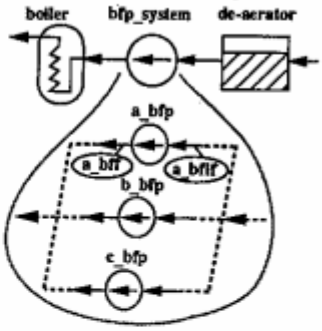

Figure 3: Hierarchical modeling of plant devices

2. Operation Principle Model

   The *Operation Principle Model* is concerned with the principles for safe and economical plant control. It consists of the following two rules.

   - Strict Accordance Rule

     The purpose of this rule is to ensure plant safety throughout a series of plant operations. It consists of the following two components: a rule to use a device within its own allowable range, and a rule to keep a faulty device out of service.

- Preference Rule

  The purpose of this rule is to ensure an economical plant operation. It consists of the following two components: a rule to keep the number of in-service devices to a minimum, and a rule to equalize the service-time of each device.

3. Dynamics Model

   The *Dynamics Model* represents the dynamic characteristics of the plant. In the area of plant control, the *Dynamics Model* is concerned either with the functions of traditional plant controllers based on *PID-control* or with the characteristics related to physical laws. Figure 4 shows the model of a *water-flow-controller*. Kp and T are constants. $1/s$ means the integral operator.

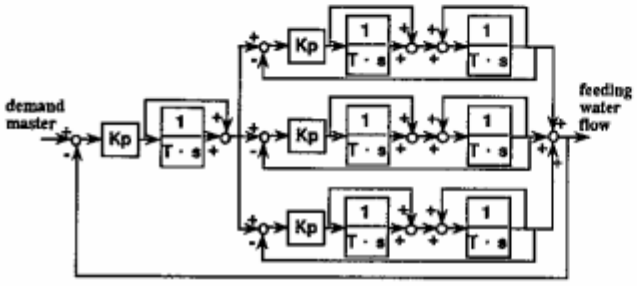

Figure 4: An example of the Dynamics Model

3.2 Model-Based Reasoning Mechanism

We briefly explain the model-based reasoning mechanism of these modules: the *Operation Generator*, the *Precondition Generator* and the *Simulation-Verifier*.

1. Operation Generator

   This module determines the *goal-state* where all of the constraints defined by the *Device Model* and the *Operation Principle Model* are satisfied. Generally, an unforeseen abnormal situation causes a state change of a plant, and this change can make the above constraints unsatisfied. To estimate this unsatisfied constraints, the following functions are needed.

   (a) Verification of Constraints

   All the constraints defined by the *Device Model* should be verified to see if they are still satisfied after the unforeseen abnormal situation. This function (Figure 5) consists of the following two sub-functions: propagating the change at each device to the others according to the connections of devices, and locally verifying the constraints at each device.

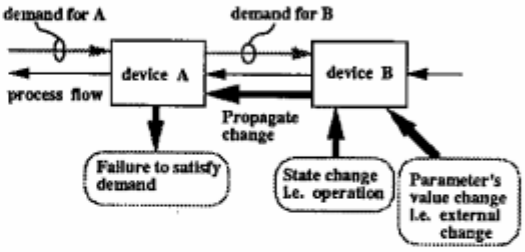

Figure 5: Constraints verification function

   (b) Update of the Goal-State

   If some of the constraints at a certain device are proved not to be satisfied, a new state for this device should be sought in order to satisfy them. This function (Figure 6) consists of the following sub-functions: searching for a 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. The plant operations are deduced by taking the difference between the initial *goal-state* and the updated one.

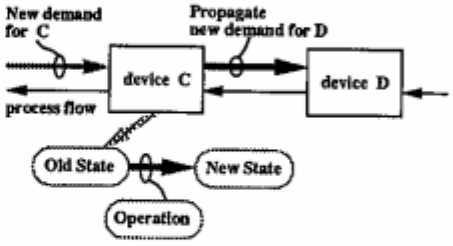

Figure 6: Goal-State update function

2. Precondition Generator

   In the domain of thermal power plant control, preconditions of each plant operation can be classified into the following five generic classes [Konuma 1990].

   - Preconditions for the state before an operation
   - Preconditions for the order of operations

- Preconditions for safety during an operation
- Preconditions for the timing of an operation
- Preconditions for completion of an operation

This module generates the above preconditions for each operation by analyzing the *goal-state* according to the constraints defined by the *Device Model*. An image of their generation process is shown in Figure 7.

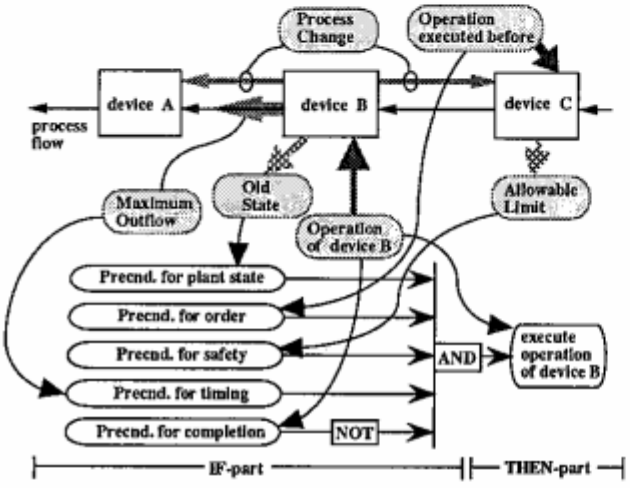

Figure 7: Generation process of preconditions

3. Simulation Verifier

This module predicts plant behavior using the *Dynamics Model* to verify the knowledge compiled from models by the *Operation Generator* and *Precondition Generator*. The prediction of plant behavior can be realized through simulation methods [Suzuki 1990]. After the prediction, the module examines whether or not undesirable events have occurred. Undesirable events can be defined by several criteria, but one of the most important is the transient violation of the allowable range 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. If this change is beyond the allowable range of a current plant state, it is detected as a violation.

The *Simulation-Verifier* supports the *Generate&Test* algorithm of knowledge [Suzuki 1990] as illustrated in Figure 8. This process can be formalized as updating the *goal-state* according to the degree of the violation.

```
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 : plant state	M : output of Diagnosor
Di : demand for a device	Op : plant operations
PS : plant behavior	NG : flag for allowable range violations
Ki : plan of plant operations	
[ ] : list expression	<= : substitution expression

Figure 8: Generate&Test algorithm of the knowledge

4 Experiments

We have implemented the expert system on Multi-PSI [Taki 1988]. To realize a rich experimental environment, we have also implemented a plant simulator instead of an actual plant on a mini-computer G8050. Both computers are linked by a data transmission line. This section describes the results of some experiments.

4.1 Configuration of a Thermal Power Plant

Figure 9 shows the configuration of the thermal power plant. It consists of controllers (hatched rectangle) and devices. 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 moved 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 *fuel-system* supplies pulverized coal to the boiler.

4.2 Experimental Results

The total of the *Device Models* in the system amounts to 78 (Table 1). In this table, the difference between the numbers in the left and right columns is due to hierarchical modeling.

The experiments were performed as follows.

1. First, we selected appropriate faults of the following devices: a *coal-pulverizer*, a *boiler-feeding-pump*, a *condensation-pump*, a *circulation-water-valve*, and a *water-heater*. We made these faults the malfunctions of the plant simulator. We also set them up for multiple faults.

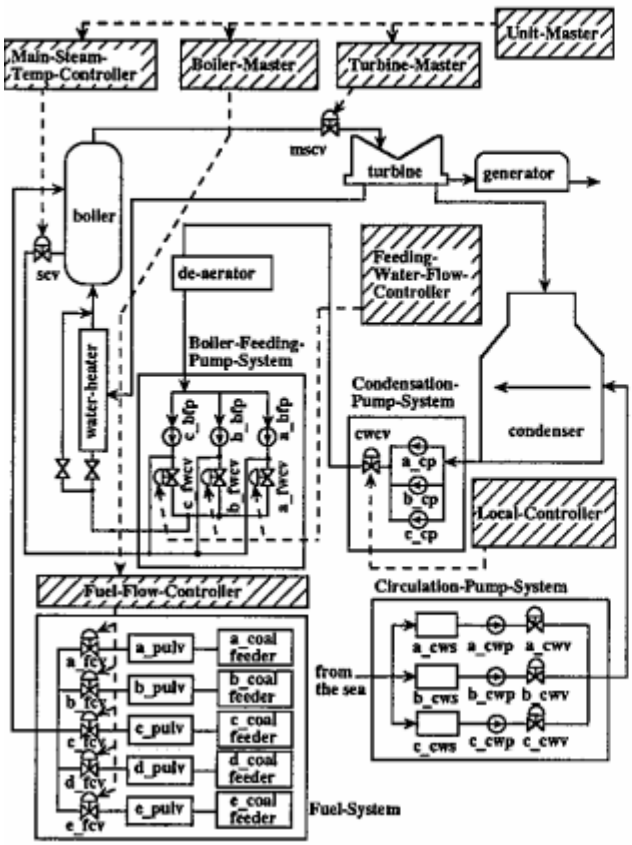

Figure 9: Configuration of a thermal power plant

2. Next, we extracted some specific knowledge for plant control from the knowledge base in the *SIS*. This specific knowledge was necessary to deal with the selected faults. As a result, the selected faults were equal to unforeseen abnormal situations.

3. Finally, after activating the malfunctions of the plant simulator, we confirmed that the *DIS* compiled the knowledge from the models and that the *SIS* executed the operations accordingly.

We explain the quality of generated knowledge for a single fault, because the results in multiple faults are the same as in a single fault. In the experiments, the contents of generated knowledge are concerned with switching from a faulty device to a backup one. Table 2 summarizes all the generated plant operations. In the case of a *water-heater* fault, the system failed to generate plant operations. In other cases, the system succeeded in generating plant operations. We estimate the quality of the generated knowledge in terms of its preconditions. This table lists columns consisting of the following items for each operation: the number of the preconditions encoded by a human expert ($N1$), the number of the essential ones

Table 1: The amount of devices and controllers

	Amount in the plant	Amount in the Device Models
Devices	43	63
Controllers	7	15
Total	50	78

in $N1$ ($N2$), the number of generated ones by the system ($N3$), the covered ratio of $N2$ by the system ($CR1$), and the uncovered ratio of $N2$ (ER), namely, the ratio of $N2$ missed being generated or incorrectly generated by the system. ($CR2$ will be explained in Section5.)

The difference between $N1$ and $N2$ is due mainly to the following reason. Although a human expert specifies the preconditions of the knowledge as generally as possible, the system generates specialized preconditions for each occurring unforeseen abnormal situation. With this point in mind, we determine $N2$ by eliminating unnecessary preconditions from $N1$. CRi (i=1,2) and ER are calculated by the following formulas.

$$CRi = \frac{Success(N2)}{N2}$$

$$ER = \frac{Miss(N2) + Fail(N2)}{N2}$$

$Success(N2)$ denotes the number of $N2$ generated by the system; $Miss(N2)$ the number of $N2$ which were not generated; and $Fail(N2)$ the number of $N2$ incorrectly generated.

We also consider the following in evaluating $Success(N2)$.

- Although the generated preconditions enumerate the individual state of each device, a human expert often represents them succinctly. For example, the conjunctive precondition "a_bfp = on" ∧ "b_bfp = on" ∧ "c_bfp = off" are represented as "the number of activated bfp = 2".

- The system often generates superfluous preconditions that a human expert does not mention.

- Although a human expert encodes preconditions for the selection of an in-service device, the system never generate them because they are already estimated in applying the *Operation Principle Model*.

None of the above devalues the quality of generated knowledge because the system is required only to generate specific preconditions for an occurring unforeseen abnormal situation. For this reason, we regard generated preconditions applicable to any of the above as $Success(N2)$.

We carried out the experiments under the following conditions.

Table 2: Quality of generated knowledge

Unforeseen situation	Precond. / (OP) Operation	The number of preconditions					
		Knowledge in KB (N1)	Knowledge in KB (N2)	Generated (N3)	Covered (CR1) ratio [%]	Error (ER) ratio [%]	C.R. after (CR2) refinement [%]
Pulverizer Fault	1. activate a Pulverizer	12	6	11	100	0	100
	2. halt a Pulverizer	8	8	12	100	0	100
BFP Fault	3. activate a BFP	42	18	8	22	78	100
	4. open a FWCV	26	10	8	40	60	90
	5. set FWCV auto	32	8	8	38	62	75
	6. set FWCV hand	12	4	7	75	25	100
	7. close a FWCV	14	6	9	83	17	100
	8. halt a BFP	23	8	11	87	13	100
CP Fault	9. activate a CP	17	7	6	57	43	100
	10. halt a CP	13	7	8	86	14	100
CWV Fault	11. activate a CWP	8	7	7	57	43	100
	12. open a CWV	4	3	7	67	33	100
	13. close a CWV	4	3	8	67	33	100
	14. halt a CWP	8	7	8	71	29	100
HTR Fault	15. open a HTR Bypass VLV	4	3	failed to generate OP	failed to generate OP	failed to generate OP	failed to generate OP
	16. close a HTR VLV	4	3	failed to generate OP	failed to generate OP	failed to generate OP	failed to generate OP

- Once *DIS* was activated, no further unforeseen abnormal situation occurred.
- The *Diagnosor* deduced the exact diagnostic results.

Because of the above conditions, *SIS* interpreted all the generated knowledge and handled the unforeseen abnormal situations. Figure 10 shows the generated knowledge and its corresponding knowledge encoded by a human expert for the operation no.5 in Table 2. We also show some additional information in Figure 10, which is referred to in the next section.

5 Discussion

In this section, we evaluate the system's capability to generate the necessary plant operations and to generate the correct preconditions for each operation. The former is concerned with performance of the *Operation Generator* and the latter is concerned with that of the *Precondition Generator*. In addition, we discuss the pros and cons of using Multi-PSI and some open problems.

1. Capability to generate plant operations

In the experiment, the system could generate all the necessary plant operations for each malfunction except the *water-heater* fault. We briefly explain the reason for this failure bellow.

At a *boiler*, the following approximation holds true for outlet steam pressure (P), inlet fuel flow (F), inlet water temperature (T) and inlet water flow (G).

$$P = \int (c_1 F + c_2 G(T - \alpha_1) + \alpha_2) dt$$

c_1, c_2 are positive constants, and α_1, α_2 are correction terms related to other process parameters.

The *Operation Generator* calculates F, G and T from P using this formula defined in the *Device Model*. P is the demand for the *boiler*. After that, the *Operation Generator* propagates F to the *fuel-system*, and G and T to the *water-heater* as a new demand respectively. In this time, the *Operation Generator* must evaluate the above formula from left side to right side, but possible value combinations of F,G and T cannot be decided using the single input value P. To deal with this undecidability, the *Operation Generator* utilizes the *Operation Principle*

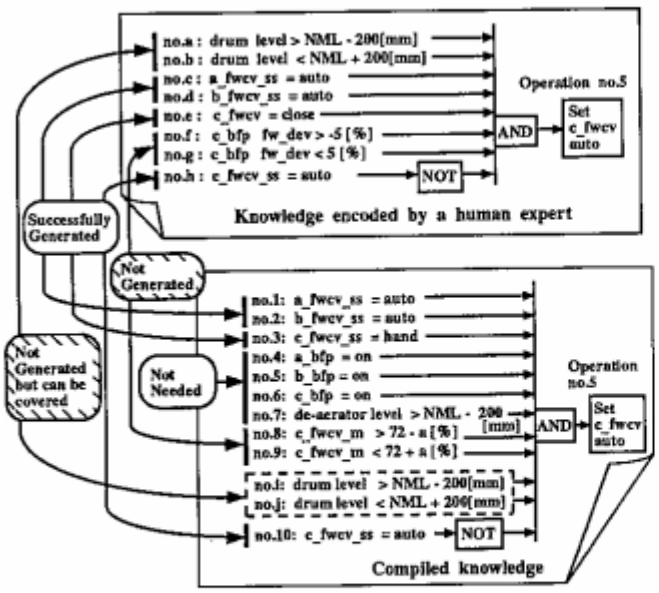

Figure 10: Knowledge for c_bfp controller

Model and approximation functions supplemented with the *Device Model*. The failure in *water-heater* fault is caused by this reasoning mechanism. We believe that additional principles are needed to evaluate such a process balance.

2. Capability to generate preconditions

   From $CR1$ and ER in Table 2, we can see that most of the generated preconditions are imperfect, namely $ER > 0$. The reasons are as follows.

   - The *Precondition Generator* failed to generate preconditions related to devices not modeled in *Device Model*. An example is the set of preconditions to establish the electric power supply for the pump. We can resolve this problem easily by augmenting the *Device Model*.
   - Although all the necessary preconditions could be checked in the *goal-state* search, the *Precondition Generator* missed analyzing them. No.i to no.j in Figure 10 illustrates this point. The system focuses only on the neighbor devices of the operated device. Because the system is required only to generate specific preconditions for an occurring unforeseen abnormal situation, we can resolve this problem easily by extending the focusing area.
   - The *Precondition Generator* generated incorrect preconditions for the timing of operations, as shown by no.8 to no.9 in Figure 10. Although the system is based on the concept that the timing of operations can be determined from the maximum outlet process flow of each device, this concept does not hold true for devices such as PID-controllers or devices placed under the control of PID-controllers.

   Although we can resolve the former two problems easily, the last problem is serious because it is closely related to the basic concept for the generation of preconditions. It is still an open problem. In Table 2, column $CR2$ represents the expected results after the refinements against the former two problems. The remaining uncovered parts for operations 4 and 5 (ER is 10% and 25% respectively) are related to the last mentioned problem above.

3. Real-time reasoning using Multi-PSI

   Although our system does not require of the severe real-time reasoning capability to cover either *PID-control* or *adaptive-control*, it requires at least the ability to compile the knowledge within a few minutes. To guarantee this performance, we have been investigating a parallel reasoning mechanism with Multi-PSI [Suzuki 1991]. We can use KL1 language on Multi-PSI, which is a profitable language to implement a multi-process system concisely. In particular, its process synchronization mechanism by "suspend" is an advantage for our system implementation. In spite of this point, it is very difficult to achieve a drastic speedup using KL1 and Multi-PSI. We have already demonstrated a threefold to fivefold improvement of reasoning time by using Multi-PSI with 16 processor elements. To achieve more improvement, we think we must make a more elaborate implementation.

4. Utility of the compiled knowledge

   In contrast to the classical approach by shallow knowledge, our proposed model-based reasoning mechanism succeeded to deal with unforeseen abnormal situations in a plant. This point is the utility of the compiled knowledge.

   Although our proposed mechanism is powerful to deal with unforeseen abnormal situations, it is weak with respect to the acquisition of knowledge which is reusable in the *SIS*. Because the system generates specific knowledge only for occurring unforeseen abnormal situations, the generated knowledge is either too general with respect to the lack of some conjunctive preconditions or too specific with respect to their enumerative representations from the viewpoint of its reusability.

5. Facility of model acquisition

   The system utilizes the *Qualitative Causal Model*, the *Device Model*, the *Operation Principle Model* and the *Dynamics Model*. These models 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 separately. Therefore, model sharing is not yet realized.

In a diagnostic task, Yamaguchi[Yamaguchi 1987] refers to the facility of model acquisition. Some other related works are in the area of the qualitative reasoning. Crawford [Crawford 1990] attempted to maintain and support the qualitative modeling environment by *QPT*.

6. Over-sensitive verification of the plant behavior

   In the current implementation of the *Generate&Test* algorithm for the knowledge, the priority of each allowable range is not considered at all. Therefore, even though the violation of the range is slight enough to be ignored, the system tries to deal with this violation sensitively. This sensitivity is meaningless for all practical purposes because a plant would be designed with enough capacity to absorb the violation. For this reason, the system should check the range with some allowable degree of violation. We are now investigating the mechanism.

7. Monitoring the execution of the generated knowledge

   In this paper, we supposed that the *Diagnosor* can diagnose unforeseen events exactly. However, in general, this supposition can be invalid. Diagnostic results should be estimated by plant monitoring following the plant operations.

   As for the related work, Dvorak [Dvorak 1989] utilizes the *QSIM* [Kuipers 1986] to monitor a plant. However, he does not refer to the generation of the knowledge for unforeseen events.

6 Conclusion

We proposed a diagnostic and control expert system based on a plant model. The main target of our approach is a system which could deal with unforeseen abnormal situations. Our approach adopts a model-based architecture to realize the thought process of a skilled human plant operator.

In this paper, we focused on model-based generation of plant control knowledge, and explained the details of the model-based reasoning. Our system utilizes the following models: the *Device Model*, the *Operation Principle Model* and the *Dynamics Model*. We also discussed its ability as demonstrated through some experimental results. The results encourage us to make sure the model-based reasoning capabilities in plant control.

Acknowledgements

This research was carried out under the auspices of the Institute for New Generation Computer Technology (ICOT).

References

[Crawford 1990] 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).

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

[Konuma 1990] 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).

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

[Suzuki 1990] 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).

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

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

[Yamaguchi 1987] Yamaguchi, T., et al. *"Basic Design of Knowledge Compiler Based on Deep Knowledge"*, Journal of Japanese Society for Artificial Intelligence, vol.2, no.3, pp.333–340 (1987) (in Japanese).