Korean J. of Chem. Eng., **10**(1), 28-35 (1993)

AUTOMATIC GENERATION OF THE SYMPTOM TREE MODEL FOR PROCESS FAULT DIAGNOSIS

Dong Soo Nam, Yong Jin Choe, Yeo Hong Yoon and En Sup Yoon

Department of Chemical Engineering, Seoul National University,
Kwanak-ku, Shinlim-dong, San 56-1, Seoul, Korea
(*Received 4 September 1992 • accepted 24 December 1992*)

Abstract—The Symptom Tree Model (STM) has been studied extensively as a model for fault diagnosis in chemical processes and has been applied to real processes. In this study, a program to build a model, AUSST (AUtomatic Synthesis of the Symptom Tree model), which generates the STM automatically is developed. The input information supplied to AUSST includes the process topology and the unit model library. The unit model library is represented in the form of mini-fault trees which can be constructed systematically through qualitative abstraction from the mathematical model or the operation data and experienced operators. AUSST has worked well, the generated symptom trees describe the paths of fault propagation sufficiently and contain all the possible primal faults. AUSST helps to assure the accuracy of the STM as well as managing the STM consistently. It is expected that AUSST reduces the engineering efforts required to develop a fault diagnostic system for a new process.

INTRODUCTION

Fault detection and diagnosis is the problem of identifying and isolating the root causes of process disturbances from observable symptoms. The root causes include physical failures and external disturbances. Physical failures include sensor and controller failures, leaking, blockage, and fouling. External disturbances are feedstock or utility variations outside the process boundaries. The symptoms are the deviations of measurable process variables.

The model-based approach has been extensively studied in computer-aided fault diagnosis in order to overcome the detriments of the experiential knowledge approach. Two classes of process models can be considered: (1) an underlying mathematical description, numerically or qualitatively and (2) a representation of the casual sequences. The former types of models are found in the papers by Isermann [1] and Kuipers [2, 3]. The latter types of models include the Fault Tree Model (FTM) [4-6], the Symptom Tree Model (STM) [7-9], the Signed Directed Graph (SDG) [10-12], and the Fault Consequence Digraph (FCD) [9].

Regardless of the type of model, the diagnostic model is one of the most important parts in the fault diagnostic system. When the model is built manually, it requires much time and effort. Also, the model depends on personal experience and is not easy to manage consistently with its increasing size. Therefore, it is natural that computer-aided building should be paid considerable attention.

From the early 70's to the middle 80's, several methodologies for computer-aided fault tree synthesis have been proposed. Since Fussel [4] initiated an automated construction of a fault tree for an electrical system with his Synthetic Tree Model, many studies have been done. One important piece of information for fault tree synthesis of a process plant is a description of the process. There are two approaches to develop a description of the process. The first approach is centered around the component units, where the unit model of a local causality among process variables is represented in terms of mini-fault trees (Powers and Tomkins [5], Martin-Solis et al. [6]). In the second approach, the process can be described by placing stress on the process flowsheet structure (Lapp and Powers [13], Shafaghi et al. [14]). In this approach, the process is represented in terms of a directed graph or a reliabilty graph. The process interaction and cause-and-effect relationships are obtained from this graph. Using this information, a complete fault

Table 1. Applications using the symptom tree model

Process	Methods	Workers
Cement calcination process	symptom tree symptom-failure cause table	Han & Yoon [7]
Naphtha furnace	symptom tree	Kim & Yoon [8]
Naphtha furnace	symptom tree FCD	Oh, Yoon & Choi [9]

tree is built up with an algorithm.

The building of the STM is generally a complicated task, like that of the FTM. The objective of this study is to build the STM automatically from a given process topology. This includes representation of process topology, representation and development of the chemical process' unit model for local causality, development of an algorithm for building the STM automatically, and interfacing with diagnostic systems. This model builder is applied to a simple buffer tank system for theoretical background, and has been tested on real processes.

SYMPTOM TREE MODEL

The Symptom Tree Model is a qualitative graph which represents fault propagation, cause-and-effect relationships between symptoms and their causes in a process plant. The STM is basically derived from the symptom-sub tree concept and the definition of a symptom variable. It is constructed in terms of Boolean logic like a fault tree. The difference between the symptom tree and the fault tree comes from the nature of their top events. The top event of the fault tree usually represents hazardous events in a system while that of the symptom tree represents a symptom variable which is to be measured. Practical implementations using the STM are shown in Table 1.

In former studies, the building of an STM is performed manually. The method that Han and Yoon [7] proposed is as follows: First, the fault tree for a target process is constructed, then it is divided to obtain the symptom tree. Specifically, the hazardous event is selected as a top event. All the specified component failures and the process variable deviations leading to the top event are represented in one tree. Then, the symptom trees whose sensor variables become the top events, are extracted from the fault tree.

Kim et al. [8] obtained the symptom tree by combining successively the symptom-sub trees which are obtained for each process variable deviation. The events which are inconsistent with the top event are removed. The events which are overlooked by the symptom-sub tree, which represents the local causality, are added.

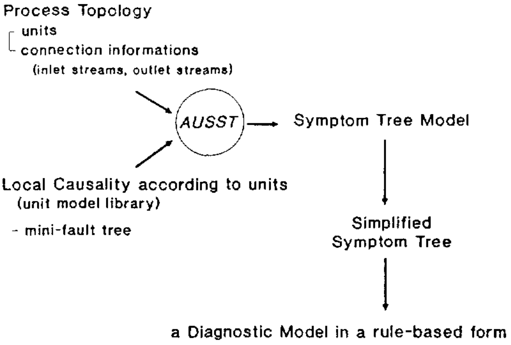


Fig. 1. Diagram of the procedure for obtaining diagnostic model using AUSST.

Oh et al. [9] derived the symptom tree from the FCD [9]. This method was proposed to maintain integrity between the STM and the FCD, and reduced the effort of constructing the symptom trees. Since the symptom trees were constructed without symptom-sub trees for unmeasured variables, a number of mistakes and awkwardness were reduced.

There are several problems with the above methods. The detriment of Han's approach is the requirement that a large fault tree be constructed in order to obtain the symptom trees. Oh's method which derives symptom trees from the FCD is more easily constructed, but it is cumbersome to construct the FCD first. Kim's method needs systematic refinement. Computer-aided generation of the STM could be possible using the symptom tree concept and the methodology of computer-aided fault tree synthesis.

AUTOMATIC BUILDING OF THE STM

The model building program, AUSST (AUtomatic Synthesis of the Symptom Tree model), which builds the STM automatically is developed. AUSST runs on a SUN4 workstation and was developed under NEXPERT, an expert system shell developed by Neuron Data Inc. for interfacing to the research.

The procedure for obtaining a diagnostic model using AUSST is shown in Fig. 1. The input information given to AUSST are the process topology and the unit model library. From this, the STM can be built. Also, simplified symptom trees can be obtained by removing the intermediate events from the STM, excluding the deviation of measured variables and primal faults. Fi-

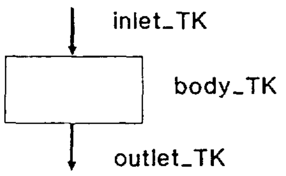


Fig. 2. Buffer tank unit model.

Table 2. Variables for buffer tank unit model

	Variables	Top event variables
inflow_TK	F_i_TK, X_i_TK, T_i_TK, P_i_TK	P_i_TK
body_TK	L_b_TK, X_b_TK, T_b_TK	L_b_TK, X_b_TK, T_b_TK
outflow_TK	F_o_TK, X_o_TK, T_o_TK, P-O-TK	F_o_TK, X_o_TK, T_o_TK

nally, a diagnostic model in a rule-based form can be produced, which can be used in diagnostic systems.

1. Unit Model Library

The process unit model is represented in terms of a mini-fault tree which represents local relationships between the process variables. Primal faults possibly occurring are defined for the concerned unit. Inlet streams, outlet streams, and the variables in each stream and unit's body are identified. The direction of the influnces between the variables must be identified. These can be achieved from the following two methods: If it is difficult to obtain a mathematical model, a mini-fault tree is constructed using the cause-and-effect relationship between the variables obtained from the operation data and the experienced operators. Otherwise, it can be obtained from the mathematical model. The mathematical model usually consists of ordinary differential equations (ODE's) and algebraic equations (AE's). In general, ODE's can be written as Eq. (1), where the direction x_j's influence on x_i is the sign of $\partial f_i/\partial x_j$. Also, AE's can be written in the form of Eq. (2), where the direction x_j's influence on x_i is the sign of a_{ij}. The deviation of a process variable on the left-hand side of Eqs. (1) and (2) is the top-event of the mini-fault tree.

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \cdots, x_n) \tag{1}$$

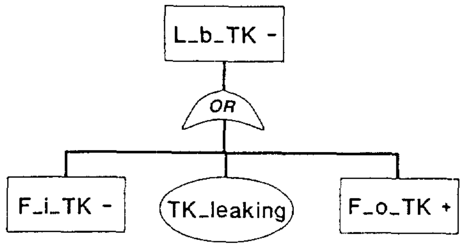


Fig. 4. Mini-fault tree for top event L_b_TK- in buffer tank unit model.

$$x_i = \sum_{j=1}^{n} a_{ij} x_j \tag{2}$$

One primal fault which can occur in the buffer tank, as shown in Fig. 2, is leaking. A buffer tank is divided into three parts, the inlet stream, the outlet stream, and the tank body. These parts are named inlet_TK, outlet_TK, and body_TK respectively. These will be objects in NEXPERT as shown in Fig. 3. Top event variables of the mini-fault trees and the variables in each part are shown in Table 2. Mini-fault trees for the states of tank level high and low can be obtained as follows: when a model equation of buffer tank is given as Eq. (3), the direction inlet flow rate's (F_i_TK) influence on tank level(L_b_TK) is positive, $\partial(dL_t/dt)$

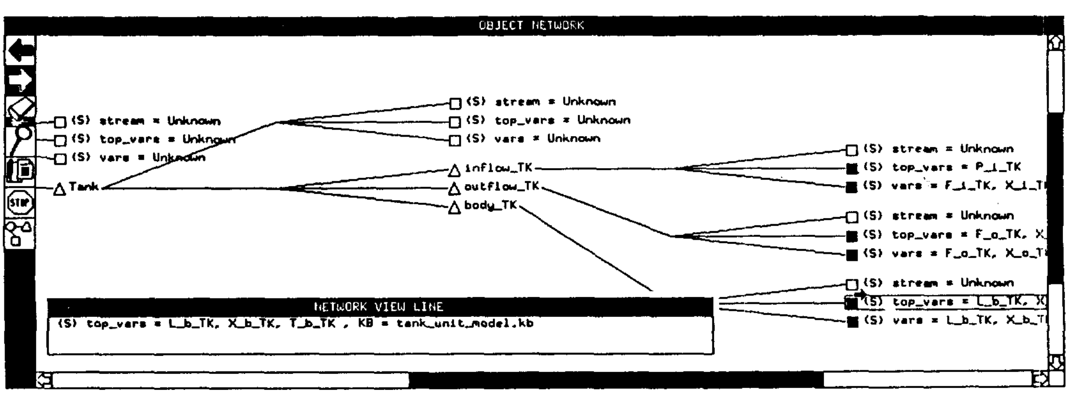


Fig. 3. NEXPERT OBJECT representation of buffer tank unit model.

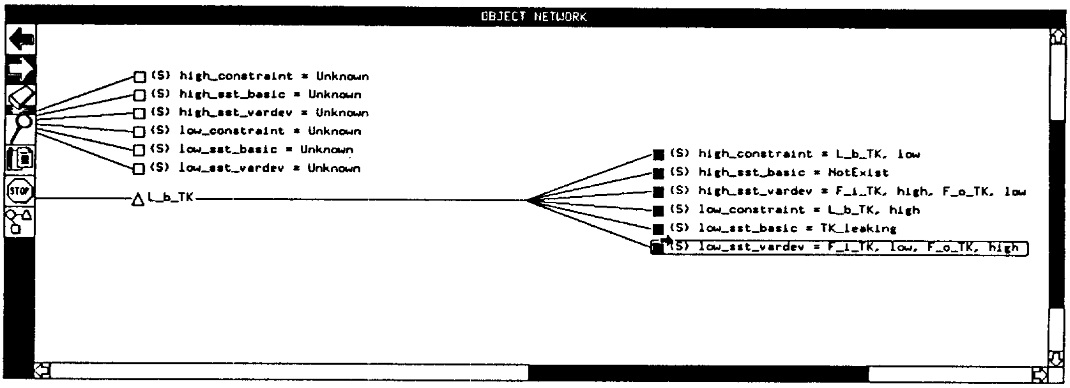


Fig. 5. NEXPERT OBJECT representation of mini-fault tree in Fig. 4.

Topology
GIn *stream numbers*
GOut *stream numbers*
Unit *unit model name1* Name *particular name1*
In *inlet stream numbers* Out *outlet stream numbers*
Unit *unit model name2* Name *particular name2*
In *inlet stream numbers* Out *outlet stream numbers*
⋮

Fig. 6. The templet of input file to the AUSST.

$/\partial F_i = 1/(\rho \cdot A)$ or >0. That means that L_b_TK increases as F_i_TK increases and decreases as F_i_TK decreases.

$$dL_t/dt = (F_i - F_o)/(\rho \cdot A) \quad (3)$$

In Fig. 4, the causes which make L_b_TK decrease are inlet flow rate low, outlet flow rate high, and tank leaking connected by an OR gate. A mini-fault tree for the top event L_b_TK- is stored as low_sst_vardev and low_sst_basic's property value as shown in Fig. 5. Mini-fault trees for other top event variables such as F_o_TK+, P_i_TK+, etc. also can be obtained in a similar fashion and are stored as the above-mentioned method. In this study, a mini-fault tree is stored in one-layer in order to develop and maintain a unit model easily. Basides, it is easy to incorporate a new unit model into the library.

2. Process Topology

In addition to the unit model library, another piece of input information supplied to AUSST is the process topology. A process is decomposed according to units and transformed into a block diagram. Stream numbers are assigned to each stream without respect to material flow and electrical signal. The decomposed units and the connection information are given to AUSST as a process topology.

The process topology proposed in this study is shown in Fig. 6, as an input file templet. Gin represents inlet streams flowing into the concerned process from its surroundings and Gout represents outlet streams flowing into the surroundings from the process. Explicit notation of Gin and Gout distinguishes them from the outlet streams of the sensor unit and makes top events of symptom trees identified automatically.

Unit model name is an allowed name in the unit model library, such as Tank, Controller_L, Sensor_F, etc. *Particular name* is an arbitrary name given by the user. *Inlet and outlet stream numbers* must be filled according to the specified order which is determined when the unit model is developed. For example, with outlet streams of the buffer tank unit model, the first outlet stream number represents the material flow stream and the other stream numbers represent electrical signal streams.

3. Strategy

The overall strategy used to build the STM is shown in Fig. 7. First, a process topology is supplied to AUSST. This information is stored in inlet_strms, outlet_strms, and unit_model as shown in Fig. 8(a). *Particular name* in Fig. 6 is an object as an instance of UsedUnit class. The process topology is analyzed according to streams. The in_unit and out_unit properties are attached to each stream as shown in Fig. 8(b), which is used later as an information medium in tree expansion. The top events of the symptom trees are identified by isolating the sensor unit among given units. After developing each branch of the tree using the mini-fault trees in the unit model library, the full symptom trees are then generated for each top event.

The tree expansion in one-layer downward from one intermediate event is equivalent to that from top

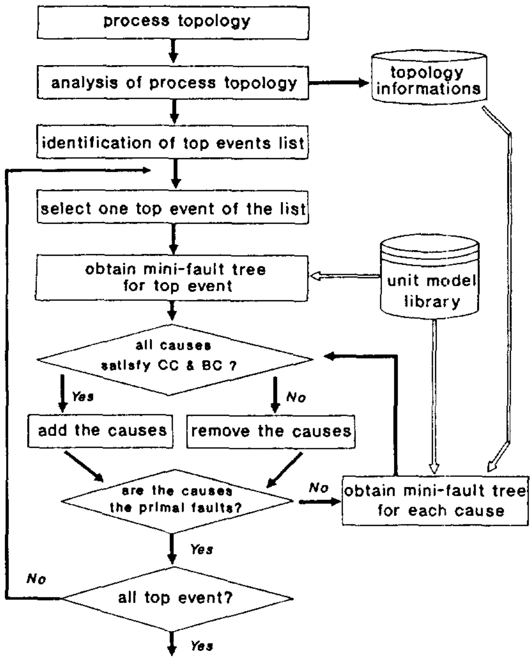


Fig. 7. Overall strategy of the Symptom Tree Model building.

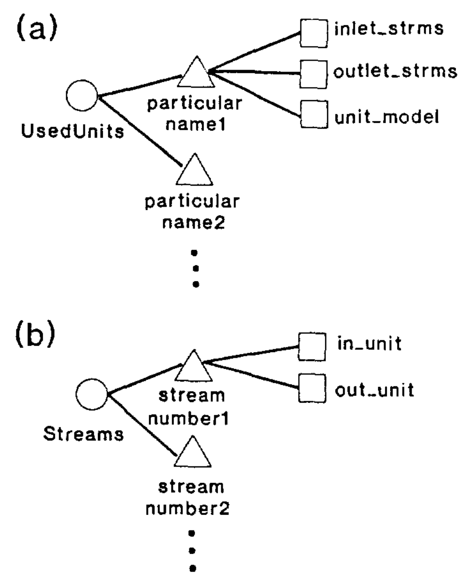


Fig. 8. NEXPERT OBJECT representation of process topology.

event and another intermediate event judging from logical point of view. The information to analyze process topology is used in one-layer tree expansion from

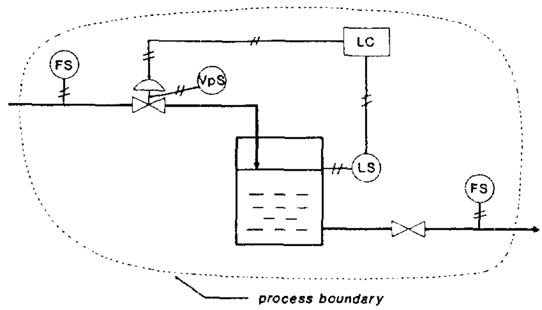
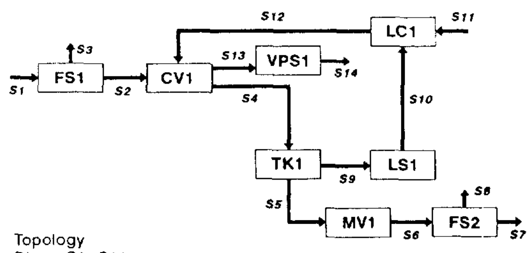


Fig. 9. Process flow diagram of a buffer tank system.




```
Topology
GIn   S1 S11
GOut  S7
Unit  Tank          Name  TK1   In  S4       Out  S5 S9
Unit  Valve_MV      Name  MV1   In  S5       Out  S6
Unit  Valve_CV      Name  CV1   In  S2 S12   Out  S4 S13
Unit  Controller_L  Name  LC1   In  S10 S11  Out  S12
Unit  Sensor_F      Name  FS1   In  S1       Out  S2 S3
Unit  Sensor_F      Name  FS2   In  S6       Out  S7 S8
Unit  Sensor_L      Name  LS1   In  S9       Out  S10
Unit  Sensor_VP     Name  VPS1  In  S13      Out  S14
```

Fig. 10. Block diagram of a buffer tank system and input file to AUSST.

each event and the sub-event is generated recursively.

It is necessary in developing the STM to avoid any inconsistencies and to restrict the expansion of tree. This is accomplished with two constraints, that is, the consistency constraint (CC) and the boundary constraint (BC). The consistency constraint says that positive deviation of a concerned variable cannot cause negative deviation of the variable and vice versa. The boundary constraint keeps an event in one tree from appearing in another tree and watches the process boundaries. The symptom trees which are generated using AUSST will be displayed in a NEXPERT window.

CASE STUDY

AUSST is applied to the simple process shown in Fig. 9, which was used by Yoon [15], Han [7] and Kim [8]. To generate the STM for this process the process boundary is set as shown in Fig. 9. There are four measured variables and a feedback control

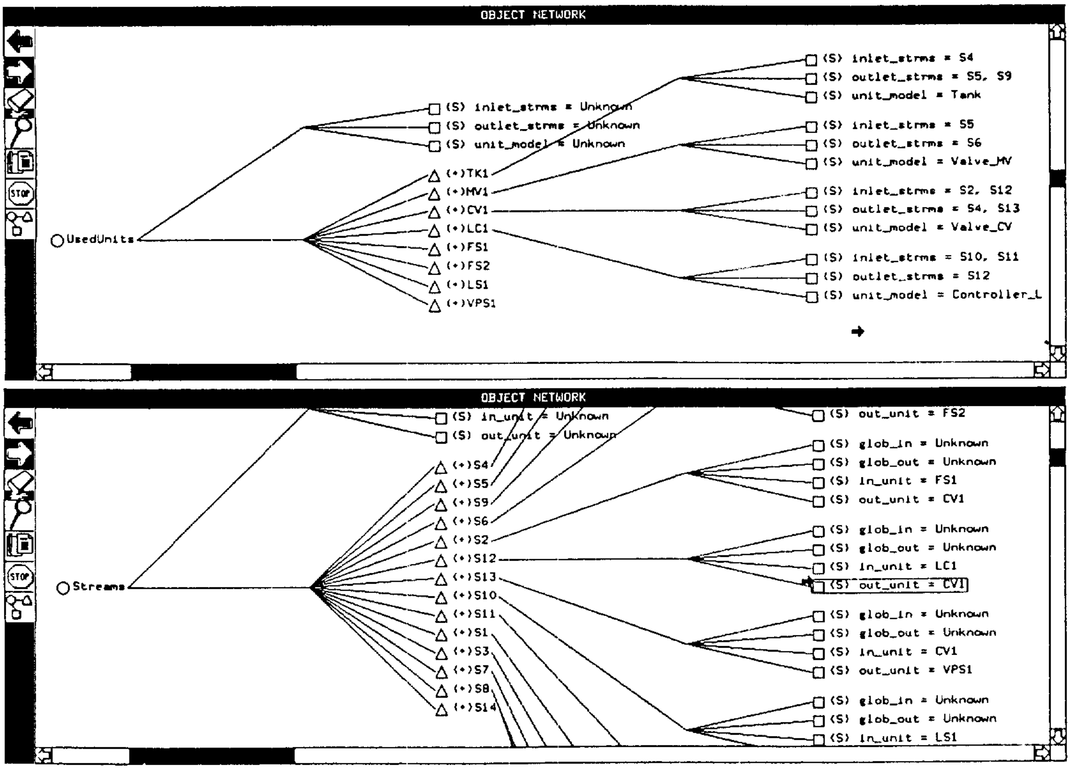


Fig. 11. NEXPERT OBJECT representation of process topology (screen dump).

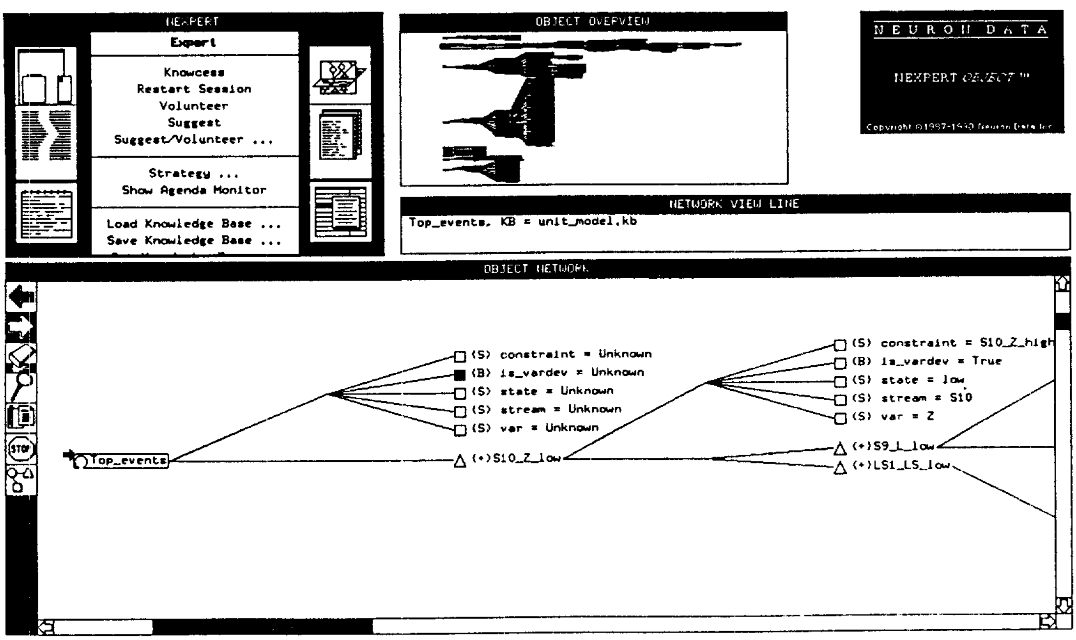


Fig. 12. Generated symptom tree for tank level decrease (screen dump).

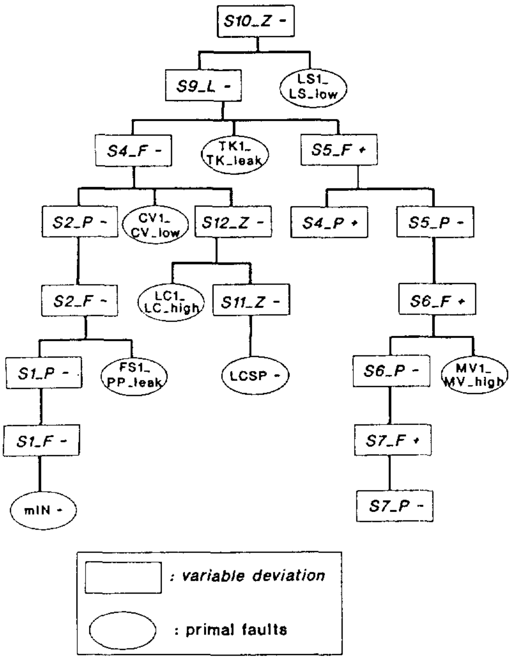


Fig. 13. Generated symptom tree for tank level decrease, S10_Z-.

Table 3. Primal faults in buffer tank system

Primal faults	Meaning
LS1_LS_low	level sensor 1 fail low
TK1_TK_leak	tank 1 leaking
CV1_CV_low	control valve 1 fail closed
LC1_LC_high	level controller 1 fail high
FS1_PP_leak	pipe leaking before inlet flow sensor
LCSP-	level controller 1 set point low
MV1_MV_high	manual valve 1 open
mIN-	inlet flowrate from process outside low

loop to control the level of tank. For this buffer tank system, a corresponding block diagram and the input file supplied to AUSST are shown in Fig. 10.

The unit models used in this process are Tank, Controller_L, Valve_CV, Valve_MV, Sensor_F, Sensor_L, Sensor_VP. They are prepared using the method described in section 1. A full description of these models, which includes variables, top event variables, and corresponding mini-fault trees, can be found in reference [16].

The process topology and information after analysis are shown in Fig. 11. After the top events of the symptom trees are identified, eight symptom trees can be obtained, one for each of the top events of the four

1st Kind

```
(setf 'S10_Z_low  '( (LS1_LS_low) (TK1_TK_leak)
                     (CV1_CV_low) (LC1_LC_high)
                     (FS1_PP_leak LCSP- MV1_MV_high)
                     (mIN-))
```

(a)

2nd Kind

```
(setf 'S10_Z_low  '( LS1_LS_low TK1_TK_leak
                     CV1_CV_low LC1_LC_high
                     FS1_PP_leak LCSP-
                     MV1_MV_high mIN-)
```

(b)

Fig. 14. Two kinds of abstraction of the generated symptom tree.

measured variables' high and low states. One of the symptom trees, the top event of which is the low state of the tank level, is shown in Fig. 12 and Fig. 13. The primal faults which can cause the low state of tank level are LS1_LS_low, TK1_TK_leak, CV1_CV_low, etc. The meaning of these are shown in Table 3. Every branch in the symptom tree is connected by an OR gate. The generated symptom trees describe the paths of fault propagation sufficiently and contain all the possible primal faults.

The computational time including the analysis of topology information used by AUSST on this example was 9.6s. The processing of one top event required about 1s. The computation speed depends on computing environment and the process scale but it is not critical problem because the generation of the STM is done off-line.

AUSST has been tested on real processes which include distillation columns, heat exchangers, pumps, etc. When the concerned process is large and complex, it is divided into several sub-processes in order to prevent the symptom trees from growing very large. Division criteria include the interaction and delay times between sub-processes. Symptom trees are generated for each sub-process. In these cases, the symptom trees which are generated by AUSST are useful as a supplement for manual construction of the whole process.

INTERFACE WITH DIAGNOSTIC SYSTEMS

Another issue to consider when developing a diagnostic model for fault diagnosis is simplifying the STM and interfacing with diagnostic systems.

The symptom tree in Fig. 13 includes the deviations of unmeasured variables. Unmeasured variables cannot be used unless they are estimated using an appropriate process model. They can be, therefore, removed to represent the paths of fault propagation more easily.

The STM which is generated using AUSST must be abstracted to be used in real-time diagnostic systems such as EXFAST [8] and OASYS [9]. The first type of abstraction is shown in Fig. 14(a). It is represented in terms of a nested list expression. This type of expression is needed for generating a symptom pattern for a fault using the STM. It can be used as a diagnostic model of EXFAST in order to present fault candidates for symptoms and validate the fault candidates. The second type which is shown in Fig. 14(b) is represented in terms of one list, without respect to position in the symptom tree. This type of expression can be used in OASYS only to present fault candidates.

CONCLUSION

In this paper, we have presented a method to build a diagnostic model, the symptom tree model, automatically. The developed AUSST has worked well and generated satisfactory symptom trees as shown in the illustrative case study. One of the most important features using AUSST is the ability to generate all the possible primal faults which may be ommitted in manual construction.

AUSST also has been tested on real processes which include distillation columns, heat exchangers, pumps, etc. In these cases, the symptom trees which are generated by AUSST are useful as a supplement to manual symptom tree construction. It is expected that AUSST reduces the engineering efforts required to develop a fault diagnostic system for a new process as well as to manage the STM more consistently.

ACKNOWLEDGEMENT

This research was supported financially by the Automation Research Center, POSTECH/ERC.

NOMENCLATURE

A : area
F : flow rate
L : level
P : pressure
T : temperature
X : composition
Z : electrical signal
x : state variable
t : time
ρ : density

Subscripts

i : inlet stream
o : outlet stream
t : tank

REFERENCES

1. Isermann, R.: *Automatica*, **20**, 387 (1984).
2. Kuipers, B.: *Artific. Intell.*, **29**, 289 (1986).
3. Kuipers, B.: *IEEE Trans. Syst. Man Cybern.*, **17**, 432 (1987).
4. Fussell, J. B.: NATO Advanced Study Institute on Generic Techniques in Systems Reliability Assessment, Nordhoff (1973).
5. Powers, G. J. and Tomkins Jr, F. C.: *AIChE J.*, **20**, 376 (1974).
6. Martin-Solis, G. A., Andow, P. K. and Lees, F. P.: *Trans. IChemE*, **60**, 14 (1982).
7. Han, J. H. and Yoon, E. S.: IFAC Workshop on Fault Detection and Safety in Chemical Plants, Kyoto, 126 (1986).
8. Kim, C. J., Oh, J. K. and Yoon, E. S.: *HWAHAK KONGHAK*, **28**, 417 (1990).
9. Oh, J. K., Yoon, E. S. and Choi, B. N.: *KACC*, **2**, 805 (1989).
10. Iri, M., Aoki, E., O'Shima, E. and Matsuyama, H.: *Comput. & Chem. Eng.*, **3**, 489 (1979).
11. Kramer, M. A. and Palowitch Jr., B. L.: *AIChE J.*, **33**, 1067 (1987).
12. Oyeleye, O. O., Finch, F. E. and Kramer, M. A.: *Chem. Eng. Comm.*, **96**, 205 (1990).
13. Lapp, S. A. and Powers, G. J.: *IEEE Trans. Reliab.*, **26**, 2 (1977).
14. Shafaghi, A., Andow, P. K. and Lees, F. P.: *Chem. Eng. Res. Des.*, **62**, 101 (1984).
15. Yoon, E. S.: Ph.D. Thesis, MIT (1982).
16. Nam, D. S.: M. S. Thesis, SNU (1992).