COLORED PETRI NET BASED DIAGNOSIS OF PROCESS SYSTEMS

Anna I. Pózna[1], Miklós Gerzson[1,3], Adrien Leitold[2] and Katalin M. Hangos[1,3]

[1]Department of Electrical Engineering and Information Systems,
[2]Deparment of Mathematics,
University of Pannonia
H-8201 Veszprém, P.O.B. 158,
Hungary

[3]Process Control Research Group,
Computer and Automation Research Institute
H-1518 Budapest, P.O.B. 63,
Hungary

E-mail: pozna.anna@virt.uni-pannon.hu

KEYWORDS
fault diagnosis, qualitative model, colored Petri net, structural decomposition

ABSTRACT

An improved diagnosis method of technological systems is proposed in this paper that is based on their qualitative colored Petri-net model (Leitold et al. 2014). The proposed diagnosis method can be used when more than one fault occur in the system and if the fault evolves during the operation of the system. In order to make the diagnosis of complex process systems computationally feasible, a structural decomposition method is introduced to reduce the size of their occurrence graphs and to make the diagnosis easier. The proposed methods are illustrated on examples.

INTRODUCTION

The fault diagnosis problem includes the specific sub-tasks: fault detection, isolation and identification. Occurrence of faults can be determined by fault detection, the type or location of faults can be found by fault isolation methods while fault identification is used for characterizing the occurred faults.

Fault diagnosis of discrete event systems was originally studied within the framework of automata theory Sampath et al. (1995). Other popular modeling formalisms are Petri nets and their different extensions such as labeled, timed or coloured Petri nets, state-charts and hierarchical state machines. The main advantages of Petri nets are that they give both structural and mathematical representation of the system. Therefore many different techniques can be used for diagnosis with Petri nets, for example the analysis of the occurrence graph, marking estimation, linear algebra, integer linear programming, diagnoser nets and reverse nets.The most frequently used methods are based on the idea of unobservable transitions and using labeled Petri net models. Besides the observability of transitions, the set of places may have observable and unobservable subsets too. In Basile et al. (2009) sufficient conditions of diagnosability are given and an on-line fault detection algorithm is developed based on ILP and checking the fault diagnosability conditions. In Cabasino et al. (2010) the markings reachable by unobservable transitions are taken into account at the construction of the occurrence graph. In Lefebvre and Aguayo-Lara (2015) firing times of transitions are taken into account and the diagnosis is based on generating residuals.

Complex systems can be represented in a compact form by using colored Petri nets (CPN). CPN can be used as a colored diagnoser Pencole et al. (2015), which has usually smaller size than the colored one, or backward reachability can be used to find the source of failures Bouali et al. (2012).

In case of large systems the computational effort of diagnoser algorithms can be extremely large therefore making effective algorithms is a very important task. Distributed diagnosis is a popular method to solve the problem however it raises the question how the global diagnosis result can be obtained from the local results. Usually some kind of communication protocol between the local diagnoser modules is required to get the total diagnosis result in Genc and Lafortune (2007).

In our previous paper (Leitold et al., 2014) a novel on-line diagnosis method is introduced for hazard identification of process systems. The method is a hybrid procedure for on-line diagnosis that combines the availability and flexibility of HAZID information-based diagnosis with the computational power and tools available for CP-nets. The deviations between the nominal and characteristic traces stem from the technological system can be identified on the occurrence graph of CP-net model. The occurrence graph of the system to be diagnosed can be constructed in advance and with the on-line searching on the graph the possible fault can be determined.

BASIC CONCEPTS

A brief introduction of the basic concepts used in this paper is given here. The detailed description can be found in Leitold et al. (2014).

Qualitative Ranges, Events, Traces and Deviations

In many cases it is enough to know whether a measured value is in a range specified in advance. For example, for a sensor S the following ranges can be defined if the rough resolution is enough:

$$Q_S = \{e0, 0_S, L_S, N_S, H_S, e1\}, \qquad (1)$$

where 0_S, L_S, N_S, H_S refer to zero, low, normal and high value measured by the sensor S, respectively, while $e0$ and *e1* may refer to outlier value caused by a bias failure. The qualitative range of binary state actuators is as follows:

$$Q_S = \{op, cl\}, \qquad (2)$$

where *op* refer to the open state while *cl* to the close state of the actuator.

The dynamic evolution of course of a process system can be characterized by time dependent variables. Output variables are the measured values, while the input variables are the actions performed by the operators. An *event* is an ordered list of a time stamp and the values of input and output variables belonging to this time value. The *event list* or operational procedure contains the possible events during a course of a process system. The set of consecutive events is called *trace*. The traces can be categorized into two main groups: to the *nominal trace* describing the normal or faultless work of the system and to the *faulty traces* referring to the different faulty modes. During the course of the system the *characteristic trace* is recorded that describes the occurring events.

The proposed fault diagnosis method is based on the comparison of a nominal trace and a characteristic trace. If there is a deviation between these traces then the system works probably in a faulty mode. Assuming that the events in a trace are ordered by the time stamps, the most important deviation types are the following:

- *never-happened* - if the coherent input and output values do not occur in the characteristic trace at any time stamp;
- *later* or *earlier* - if these values occur but at a later or earlier time stamp than in the nominal trace;
- *greater* or *smaller* var_out_i - if the value of a output variable is greater or smaller in the characteristic trace than in the nominal trace at a given time instant τ.

Colored Petri Nets

According to the formal definition (see details in (Jensen 1997)) a CP-net model consists of places, transitions, guard and arc functions, colors and tokens. For diagnostic purposes the following modelling principles are used.

- The input and output variables, the operational mode and the deviation are assigned to *places*.
- *Color of tokens* describes the variables' value, the type of the fault and the emergent deviation from the nominal trace.
- The *transitions* execute the timing of the system. The operation can be divided into user defined time period, and the values of variables can change at the end of a period.
- The *guard functions* assigned to the transitions contain the fault generation function (Gerzson et al. 2012).
- *Arcs* connect coherent places and transitions.
- The *arc functions* describe the change of colors.

The behavioral analysis can be done with the occurrence graph (Jensen 1997). The occurrence graph contains all of the reachable markings (system states) from the initial one in a form of a graph. The nodes of the graph refer to the color distribution in a given system state and based on this information the diagnosis can be performed.

DIAGNOSIS USING THE CPN MODEL OF THE SYSTEM

Having CP-net form model of a complex process system the diagnosis of the actual course can be done using the occurrence graph. Comparing the token distribution of the nodes in the graph with the characteristic trace the fault can be identified.

In order to use a CP-net for diagnosis a special CP-net model was constructed. In Leitold et al. (2014) the model was described in detail here the most important elements and some extensions are highlighted based on Fig. 1. The values of input and output variables of the system are represented as the color of the tokens on the places (var_{in_n/out_m}) in the net. Place *fault* is devoted to register the randomly generated fault. Places *dev*, *never* and transition *t3* are to store and manage the deviations from the nominal course and the never occurring events during the course.

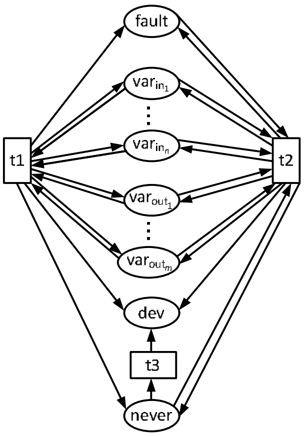


Figure 1: Structure of the Generalized CP-net Model

Transition *t1* initializes the values at the beginning with the help of special guard functions and it determines the occurring fault in the system. Transition *t2* schedules the work of the system; the changes take place as time evolves.

In a real process system faults can occur at any time and not only one fault can influence the course of the system. To fulfill these requirements the basic model was modified in the following way.

For the management of the effect of two or more faults at the same time new fault types were introduced. These new types were added to the list of possible faults, so the adequate faulty operational mode can be generated at the initialization of the system. The traces of these faulty modes also generated and added to the set of traces.

To model the fault occurring at any time it is assumed that the system works in normal way until this point of time. At the recent stage of our work it is assumed that only one fault occurs on the fly and this fault is constant until the end of course of the system. Using this assumption the faulty operational mode has to be modelled from this point of time. For the diagnosis the traces describing the events from this step should be generated and added to the model.

Our diagnosis method (Leitold et al. 2014) is based on the generation of deviation between the characteristic and nominal traces and on the searching the node on the occurrence graph which token distribution refers to this deviation. Let us assume that all the fault modes of process system are known. The first step is to generate the *deviation list* describing the distinction between the nominal and the characteristic traces. The next step is the simulation of CP-net model from the given initial state and *the generation of the occurrence graph* with all considered faulty mode. The last step of the diagnosis is to find the node having the token distribution which refers to the deviation list on the place *dev*. Based on the token color on the place *fault* in this node the type of fault can be determined. If more than one node has the token distribution referring to the deviation list, then the set of possible faults can only be concluded. If no token distribution refers to the deviation list then an unknown faulty mode occurs in the system.

The main disadvantage of the occurrence graph based method is that the size of the graph increases together with the number of units of the process system or with the refinement of qualitative measuring range. The computational effort and time also increases in this case and it has negative impact of the diagnosis. The method of *structural decomposition* can be a solution for this problem.

The structural decomposition is based on the decomposition of the process system. In a process system, the structure and the connections of components are usually known. The system can be partitioned into smaller subsystems or technological units along the connection points. Having the decomposed units of the system, the diagnosis can be performed on them separately. In case of complex systems it should be considered that the fault occurred in one unit may affect the operation of other units connected to it. Knowing the technological sequence of the units the diagnosis should be started with the first unit and its result has to be taken into account when diagnosing the following units. For the diagnosis method described above the full trace of the system should be decomposed, too. To do this, first the time step should be selected when the unit begins to work. Then all the variables belonging to the operation of this unit should be picked out from the full trace. The time steps are shifted back such that the operation of each unit starts at step 1 so each unit has its own relative time. The time steps and the values of variables compose new traces which describes the operation of the units. The following step is the generation of the deviation list with the comparison of the nominal and characteristic traces of the subsystem. If this deviation list matches the token distribution of exactly one terminal node on the occurrence graph then the possible fault can be concluded from the color of the fault place. If more than one terminal node has the same token distribution as the deviation list then the set of possible fault can be determined. If no terminal node has the same token distribution then an unknown fault is detected.

If a fault is detected in a unit and this fault has an effect on the operation of the next unit then this fault should be taken into account during the diagnosis of the subsequent unit. To do this at the generation of the occurrence graph of this unit the fault of previous unit should be taken into account as an initial condition. If more than one fault is diagnosed in the previous unit then all of them should be treated separately. If more units have effect on the examined unit then all of the faults detected in them should be taken into account as initial conditions. For example if two units have effect on the actual unit then the initial condition contains two previously detected faults. As a result the occurrence graph contains those states of this unit that happen when the previous unit has the given fault. Then the diagnosis is performed using this occurrence graph and the deviation list belonging to the actual component. The result of the diagnosis of the entire system is the union of diagnosed faults of the units.

SIMPLE CASE STUDIES

In the following our diagnosis method is illustrated using a simple process system in case of multiple faults and of a fault occurring during the course of the system. A second more complex example is used for the introduction of the structural decomposition.

Example 1: Multiple Faults and Fault on the Fly

Let us assume the following simple process system. A tank having one input and one output pipe is filled up with liquid until a certain level when the output valve is opened and the unit works in continuous mode. The filling process is a time driven event it takes two time periods. The tank has an input and output valve and a level sensor. The data measured by the level sensor is used only for monitoring the work of the unit.

For the diagnosis of the effect of multiple faults it is assumed that the following faults or their combination can occur in the system:

- The bias fault of the level sensor. The measured value is less or greater than the actual value with one qualitative unit as the effect of bad operational mode.
- The leak of the tank. The level of the liquid remains zero in the tank.
- The combination of either of bias errors and the leak.

It is assumed the fault or faults had evolved before the process starts and remain constant during the operation.

Let the states of valves be the input variables and the measured level value be the output variable. Valves are binary actuators, and their qualitative range space is in Eq. (2), while the qualitative range space of the measured level is in Eq. (1). The structure of an event is as follows:

$$event_\tau = (\tau, \textit{state of input valve, state of output valve,}$$
$$\textit{measured value of level sensor}),$$

where τ is the time stamp. The trace for the normal operational course contains the following events:

$$T = event_0, event_1, event_2, event_3;$$

where $event_0$ meets the initialization, $event_1$ refers to the start of filling up process, $event_2$ is intermediate state and $event_3$ means that the filling up is ready and then the tank works in continuous mode. The value of variables can be found in the first column of Tab. 1. The other columns of Tab. 1 contain the traces for faults tank leakage, negative bias error of level sensor and for the case when these two faults occur at the same time in the system as illustration.

Table 1: Traces for different operational modes

normal	leak	negbias	leak_negbias
(0,*cl,cl,0*)	(0,*cl,cl,0*)	(0,*cl,cl,e0*)	(0,*cl,cl,e0*)
(1,*op,cl,0*)	(1,*op,cl,0*)	(1,*op,cl,e0*)	(1,*op,cl,e0*)
(2,*op,cl,L*)	(2,*op,cl,0*)	(2,*op,cl,0*)	(2,*op,cl,e0*)
(3,*op,op,N*)	(3,*op,op,0*)	(3,*op,op,L*)	(3,*op,op,e0*)

The software package CPNTools was used for modelling the different courses of the system, for the generation of the occurrence graph and for implementing the proposed fault diagnosis method. The CP-net model of the simple tank can be seen in Fig 2. The structure of the tank model refers to the general model in Fig. 1. The operation of the CPN model is the same as described earlier in Section Colored Petri Nets.

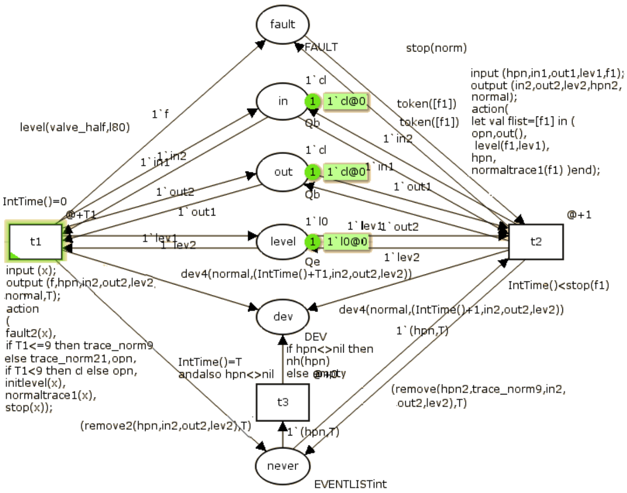


Figure 2: The CP-net Model of the Tank

The course of the diagnosis was performed using CPN model of the system (see earlier and in Leitold, 2014). It can be seen that the introduction of the occurrence two or more faults at same time does not cause any change in the diagnosis.

Let the next case be when *the fault occurs during the course of the system*. Assume that only one fault takes place and the system works in normal way until then. Based on these constraints it is enough to model the behavior of the system from this time. For the diagnosis the traces referring the system states from that time should be defined and added to the model. The course of diagnosis is the same as in case of described above. The occurrence graph will be smaller and the search needs less computational effort.

Example 2: Structural Decomposition

A composite system (see Fig 3.) is used here that consists of three tanks, two smaller and a larger one. First the two smaller tanks are filled up then their output valves are opened and the larger tank is filled up. Thereafter the system works in continuous mode. The set-up, the operation and the possible faults of these tanks are the same as it was described in the previous single tank example. Let us assume that only one fault can happen at one tank but the fault can occur anytime during the course.

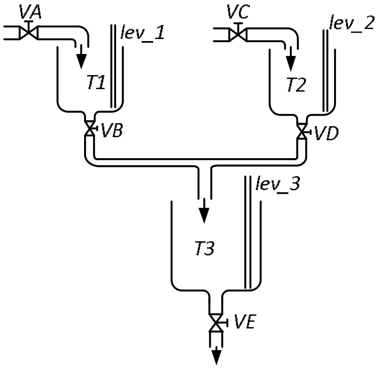


Figure 3: The Parallel Tank System

The CP-net model of the two smaller tanks is the same as in Fig. 2. In case of the third larger tank this model is extended with an extra input variable place because of the two input valves and this extra input variable is added also to the events. The nominal trace of the complex process system can be seen in Tab. 2. The cells belonging to time steps *1*, *2* and *3* and *VA*, *VB* and *lev_1* compose the trace of tank *T1* (framed with dashed line). Similarly the cells framed with dash-dot lines belong to the trace of tank *T2* and with dash-double-dot line to the trace of tank *T3*.

Table 2: Structural Decomposition of the Nominal Trace

time	Input variables					Output variables		
	VA	*VB*	*VC*	*VD*	*VE*	*lev_1*	*lev_2*	*lev_3*
1	*op*	*cl*	*op*	*cl*	*cl*	*0*	*0*	*0*
2	*op*	*cl*	*op*	*cl*	*cl*	*L*	*L*	*0*
3	*op*	*op*	*op*	*op*	*cl*	*N*	*N*	*0*
4	*op*	*op*	*op*	*op*	*cl*	*N*	*N*	*L*
5	*op*	*op*	*op*	*op*	*op*	*N*	*N*	*N*

Let us assume that the characteristic trace of the actual course is the one in Tab. 3. The trace is decomposed the same way as the nominal trace. The initial time is shifted to 1 in case of the third unit. The resulted event list of the three tanks can be seen in Tab. 4.

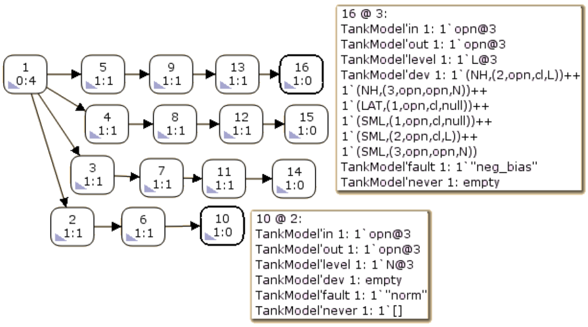
Table 3: Structural Decomposition of the Characteristic Trace

time	Input variables					Output variables		
	VA	*VB*	*VC*	*VD*	*VE*	*lev_1*	*lev_2*	*lev_3*
1	*op*	*cl*	*op*	*cl*	*cl*	*e0*	*0*	*0*
2	*op*	*cl*	*op*	*cl*	*cl*	*0*	*L*	*0*
3	*op*	*op*	*op*	*op*	*cl*	*L*	*N*	*0*
4	*op*	*op*	*op*	*op*	*cl*	*L*	*N*	*L*
5	*op*	*op*	*op*	*op*	*op*	*L*	*N*	*0*

Table 4: Characteristic Traces after Structural Decomposition

T1	T2	T3
(1,*op*,*cl*,*e0*)	(1,*op*,*cl*,*0*)	(1,*op*,*op*,*cl*,*0*)
(2,*op*,*cl*,*0*)	(2,*op*,*cl*,*L*)	(2,*op*,*op*,*cl*,*L*)
(3,*op*,*op*,*L*)	(3,*op*,*op*,*N*)	(3,*op*,*op*,*op*,*0*)

The diagnostic process is started with the first tank. The deviation list is generated as a first step by comparing the nominal and the characteristic traces belonging to this unit. Then this list is searched among the terminal nodes of the occurrence graph (see in Fig. 4.). It can be stated that the node No. 16 contains the same deviation list and based on the token of the place *fault* the type of fault can be determined: the level sensor has negative bias error.



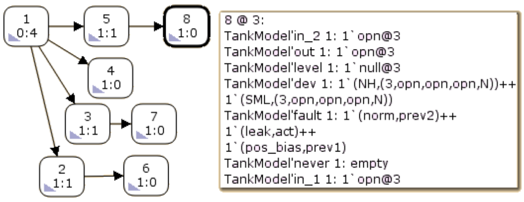
```
16 @ 3:
TankModel'in 1: 1`opn@3
TankModel'out 1: 1`opn@3
TankModel'level 1: 1`L@3
TankModel'dev 1: 1`(NH,(2,opn,cl,L))++
1`(NH,(3,opn,opn,N))++
1`(LAT,(1,opn,cl,null))++
1`(SML,(1,opn,cl,null))++
1`(SML,(2,opn,cl,L))++
1`(SML,(3,opn,opn,N))
TankModel'fault 1: 1`"neg_bias"
TankModel'never 1: empty
```

```
10 @ 2:
TankModel'in 1: 1`opn@3
TankModel'out 1: 1`opn@3
TankModel'level 1: 1`N@3
TankModel'dev 1: empty
TankModel'fault 1: 1`"norm"
TankModel'never 1: 1`[]
```

Figure 4: The occurrence graph of tanks *T1* and *T2*

Because of the parallel connection of tanks *T1* and *T2*, the diagnosis of tank *T2* can be performed irrespectively of the fault of tank *T1*. On the other hand this tank has the same model and the same occurrence graph so its diagnosis can be done in the same way as at tank *T1*. The result of the diagnosis is: this tank works in normal way (see node No. 10 in Fig 4.)

Before the diagnosis of tank *T3* the diagnosed faults of tanks *T1* and *T2* are added to the model of tank *T3* as initial conditions of place fault. Then the occurrence graph is generated. Comparing the trace piece describing the course of third tank and the trace piece referring to the normal course it can be stated that there is no deviation until time step 2. But in time step 3 a deviation appears, and it can be concluded that the fault occurred at time step 3. The occurrence graph on Fig. 5 contains the token distributions if the negative bias fault in tank *T1*, no fault in tank *T2* and a fault occurring in tank *T3* at time step 3 are the initial conditions for the simulation.



```
8 @ 3:
TankModel'in_2 1: 1`opn@3
TankModel'out 1: 1`opn@3
TankModel'level 1: 1`null@3
TankModel'dev 1: 1`(NH,(3,opn,opn,opn,N))++
1`(SML,(3,opn,opn,opn,N))
TankModel'fault 1: 1`(norm,prev2)++
1`(leak,act)++
1`(pos_bias,prev1)
TankModel'never 1: empty
TankModel'in_1 1: 1`opn@3
```

Figure 5: The occurrence graph of tank T3

The token distribution at terminal node No. 8 refers to the determined deviation list and based on the token on the place fault the tank leakage can be concluded.

CONCLUSION

A novel diagnosis method was introduced in our contribution, which is based on the qualitative CP-net model of the system. The proposed diagnosis method can also be used if more than one fault occur in the system and if the fault evolves during the course of the system. A structural decomposition method is also proposed that reduces the size of occurrence graphs and so the diagnosis needs less computational effort. The proposed method was illustrated on simple case studies.

ACKNOWLEDGEMENT

This research is partially supported by the National Research, Development and Innovation Office through grant No. K115694.

REFERENCES

Basile, F., P. Chiacchio and G. De Tommasi 2009. “An Efficient Approach for Online Diagnosis of Discrete Event Systems” *IEEE Transactions on Automatic Control*, 54, 4, 748-759

Bouali, M., P. Barger and W. Schon 2012. “Backward reachability of Colored Petri Nets for systems diagnosis” *Reliability Engineering & System Safety*, 99, 1-14

Cabasino, M. P., A. Giua, and C. Seatzu 2010. “Fault detection for discrete event systems using Petri nets with unobservable transitions” *Automatica*, 46, 9, 1531-1539

Genc, S. and S. Lafortune 2007. “Distributed Diagnosis of Place-Bordered Petri Nets” *IEEE Transactions on Automation Science and Engineering*, 2, 206-219

Gerzson, M., B. Márczi and A. Leitold 2012. “Diagnosis of Technological Systems based on their Coloured Petri Net Model” ARGESIM Report 38, 358/1-6

Jensen, K. 1997. “Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use” Springer-Verlag

Lefebvre, D. and E. Aguayo-Lara 2015. “Initial study for observers application to Fault Detection and Isolation with continuous timed Petri nets” *IFAC-PapersOnLine*, 48, 7, 97-103

Leitold, A., M. Gerzson, A. I. Pózna and K. M. Hangos 2014. “ On-Line Qualitative Model-Based Diagnosis of Technological Systems using Colored Petri Nets” *28th European Simulation and Modelling Conference – ESM’2014,* Porto, Portugalia, 332-336.

Pencolé, Y., P. Romain and P. Fernbach 2015. “Modular fault diagnosis in discrete-event systems with a CPN diagnoser” *IFAC-PapersOnLine*, 48, 21, 470-475

Sampath, M., R. Sengupta, S. Lafortune, K. Sinnamohideen and D. Teneketzis 1995. “Diagnosability of discrete-event systems” *IEEE Transactions on Automatic Control*, 40, 9, 1555-1575

WEB REFERENCES

CPNTools 2.2.0 http://wiki.daimi.au.dk/ cpntools/, University of Aarhus, Denmark, CPN Group