HUNGARIAN JOURNAL
OF INDUSTRY AND CHEMISTRY
VESZPRÉM
Vol. 41(1) pp. 59-64 (2013)

A STRUCTURED MODEL-BASED DIAGNOSIS METHOD FOR DISCRETE DYNAMIC PROCESSES USING EVENT SEQUENCES

ATTILA TÓTH ✉[1], KATALIN M. HANGOS[1,2], AND ÁGNES WERNER-STARK[1]

[1]Department of Electrical Engineering and Information Systems, University of Pannonia, Veszprém H-8200, HUNGARY
[2]Computer and Automation Research Institute, Budapest, HUNGARY
✉ E-mail: atezs82@gmail.com

A novel model-based fault detection and diagnosis method is proposed that is based on following event sequences measured in a discrete dynamic process. The model of the nominal and faulty operation modes is given in the form of event sequences, that are decomposed according to the components and sub-components present in the process system. The faulty event sequences are defined using extended procedure HAZID tables. A diagnostic algorithm is also presented that uses a component-wise decomposed form of the event sequences. The operation of the algorithm is illustrated on a simple example of a process system consisting of three similar tanks.

Keywords: Process monitoring, diagnosis, discrete event systems, qualitative models

Introduction

Fault prevention and mitigation in the field of process system management is a task of crucial importance in avoiding serious accidents. Thus, numerous hazard identification (HAZID) techniques have been developed in the past decades to ensure the safe operation of process systems and to relieve effects of faults (see [1] for a broad presentation of the field). Among these techniques, the most important methodologies involve the function-driven HAZOP (HAZard and OPerability, see [2]) analysis and the component-driven FMEA (Fault Mode and Effects Analysis). There have been results in the past decade for automating the creation of HAZOP analysis ([3] with a concrete application described in [4]). Blending the component-driven and function-driven analyses also resulted in a novel hazard identification approach described in [5].

Although the information collected in the HAZOP and FMEA studies serve the purpose of hazard identification, these studies can be the basis of diagnostic procedures, too. A model-based diagnostic method based on HAZOP and FMEA information is reported in [6].

It is important to note that the above techniques concentrate on the static case when the deviation from a normal steady-state behaviour is of importance. Therefore, the transient case when the plant is controlled by an operational procedure is not addressed in these results. A recent study [7] tries to deal with the diagnostic task by using a specially constructed P-HAZID analysis and a diagnostic algorithm. In this paper, this diagnostic idea is extended to be able to handle more complex diagnostic tasks - by taking advantage of a possible decomposition of typical process systems along their similar components.

Basic notions

In a complex system the full dynamic model that describes its behaviour under normal and faulty operation models is rarely available, therefore one should base the diagnosis on qualitative information both in terms of the dynamic models and in the measured data. Here we briefly summarize the basic notions for qualitative model based diagnosis.

Qualitative range spaces

Current values of continuous measurable outputs in process systems can be described using a properly selected qualitative range space. For example, to describe the value of a level sensor in a tank, the following range space can be used:

$$Q_e = \{\mathbf{e}-, 0, L, N, H, \mathbf{e}+\}. \qquad (1)$$

Here, 0 means an empty tank, L, N and H means low, normal and high fluid level, respectively, while $\mathbf{e}-$ and $\mathbf{e}+$ refer to unmeasurably low and high fluid levels (this might mean a failure in the level sensor itself). This range space will be used to describe system outputs during operation.

Input-output event sequences

Operational procedures in process systems are detailed list of instructions for the plant operator personnel to perform certain operations on the plant. Procedures can be formally described using finite input-output event sequences where a single event describes a change in either the inputs or the outputs of the system at a specific time instant. Therefore the syntax of a single input-output event (at time instant t) is the following

$$\mathrm{event}_t = (t; \mathrm{input\ values}; \mathrm{output\ values}).$$

The input in an event always refers to a state of an actuator component in the process system (e.g., in the case of a valve it can be **open** (op) or **closed** (cl)). On the other hand, the output in an event refers to a value of an output of the process system in the qualitative range space using the qualitative set defined in *Eq.* (1). Sequences formed from these events are called traces and defined as

$$T(t_1, t_n) = \mathrm{event}_{t_1}, ..., \mathrm{event}_{t_n}.$$

Separate events in a trace contain the same inputs and outputs. Note that the discrete event time instances t_i are abbreviated by their indexes i in the description, i.e. 2 stands for t_2.

Examples of single events for a two-input single output case include

$$(1; \mathrm{cl}, \mathrm{op}; \mathrm{N}) \quad , \quad (2; \mathrm{cl}, \mathrm{op}; \mathrm{L})$$

where at time instances t_1 and t_2 the two valve inputs are held closed and open, respectively, while the level decreases form its normal value to the low level. The trace formed form the above two consecutive events is written as

$$T(1, 2) = ((1; \mathrm{cl}, \mathrm{op}; \mathrm{N}), (2; \mathrm{cl}, \mathrm{op}; \mathrm{L})) .$$

For every operational procedure there exists a trace (called the nominal trace) which describes its behaviour under fault-free conditions. The diagnostic method compares this trace to other traces which may have been executed under faulty conditions (called characteristic traces), and the differences (called deviations) are later used to find possible malfunctions of components in the system.

Deviations

Nominal and characteristic traces can be compared by comparing their corresponding event fragments. The difference between two corresponding event fragments is described by a deviation. Deviations are formed from a deviation guideword and the nominal event from which the corresponding characteristic trace event is deviating from. The following deviation types are used during diagnosis:

- **never-happened**: When the particular event never happened in the characteristic trace.
- **later**: When the event happened in the characteristic trace, but at a later time instant.
- **earlier**: When the event happened in the characteristic trace, but at an earlier time instant.
- **greater**: When a particular output's qualitative value was higher in the characteristic event than that of the nominal trace.
- **smaller**: When a particular output's qualitative value was lower in the characteristic event than that of the nominal trace.

For the detailed description of the **greater** and **smaller** qualitative relations, please refer to [7].

Procedure HAZID

As a combination and extension of the widely used FMEA and HAZOP analyses (for details, refer to [7] or [1], and in particular to [5]), the procedure HAZID (abbreviated as P-HAZID) analysis can be used for fault diagnosis during operational procedures in a given process system. The result of this P-HAZID analysis is given in the form of a spreadsheet and it consists of deviations together with their implications and possible (root) causes. A cause is considered to be a root cause if it is a non-measurable failure mode of a system component (which is an elementary part of the system). For example, a leak on a tank is considered as a root cause. A simple example of a P-HAZID table can be found in *Table* 1.

Using the initial set of differences (deviations) between the characteristic trace and the nominal trace, the set of possible root causes can be found using simple reasoning. For details, refer to *Ref.* [7].

The diagnostic algorithm uses this technique first to find possible P-HAZID row(s) to start from (using the set of initial deviations). Then, following the deviation chains defined by these rows, the algorithm proceeds towards a possible root cause by traversing new rows based on the initial set of deviations. Using this procedure, it may end up at a root cause or at a row with deviations from which it cannot proceed forward, because they are not contained in the initial set of deviations. The algorithm assumes that the root causes are static and they happened before the execution of the procedure began.

Component based diagnosis

It is widely known that complex systems can often be decomposed in a hierarchical way using simple non-dividable elements that are called components. The connection of these components is usually specified in terms of a graph called flowsheet. Such a decomposition can be used effectively for the operation of a reasoning-based diagnostic algorithm.

Component based structural decomposition

The above mentioned fault diagnosis based on the P-HAZID analysis is only developed for process systems

Table 1: A simple example of a P-HAZID table. Inputs: op=**open**, cl=**closed**. Outputs: 0=**no**, L=**low**, N=**normal**. Deviations: NH=**never-happened**, LAT=**later**, EAR=**eariler**, SML=**smaller** and GRE=**greater**. Faults: **TANK-LEAK** is the leak of the tank and **POS-BIAS** is the positive bias failure of the tank level sensor.

Cause	Deviation	Implication
TANK-LEAK	NH(2;op,cl;L)	NH(3;op,cl;N)
NH(2;op,cl;L)	NH(3;op,cl;N)	NH(4;op,op;N)
TANK-LEAK	SML(2;op,cl;L)	SML(3;op,cl;N)
SML(2;op,cl;L)	SML(3;op,cl;N)	SML(4;op,op;N)
POS-BIAS	GRE(1;op,cl;0)	GRE(2;op,cl;L)
GRE(1;op,cl;0)	GRE(2;op,cl;L)	GRE(3;op,cl;N)
GRE(2;op,cl;L)	GRE(3;op,cl;N)	NH(4;op,op;N)
POS-BIAS	NH(1;op,cl;0)	EAR(2;op,cl;L)
NH(1;op,cl;0)	EAR(2;op,cl;L)	EAR(3;op,cl;N)

consisting of different individual components in [7]; the possible redundancy of such systems (e.g. multiple components of the same characteristics) were not taken into account. However, complex process systems in practice can be decomposed into a connected network of more simple components. For example, the process system in *Fig.* 1 can be decomposed into three smaller similar components each formed by an input and an output valve and a tank.

When developing the decomposed form of a process system it can happen that some elements are part of multiple subsystems as in the case of valves VB and VC in *Fig.* 1. These elements are called boundary components, and are assumed to be error-free during the diagnosis.

Traces affecting different components can also be decomposed into a chain of trace fragments each referring to a single component of the trace. Events in such a trace fragment have only a subset of inputs and outputs of the united trace (only the inputs and outputs of the particular component that is present in them). Fragments also have information about the next trace fragment (called the next trace), and there is a starting condition (an event) that is associated with them to help the diagnosis. Along with the trace fragment, each component has its own associated P-HAZID spreadsheet.

The component based diagnostic algorithm

Applying the diagnostic approach described in [7] on a decomposed process system, the components can be diagnosed separately against faults, treating them as a whole system during diagnosis. After the separate diagnosis, the root causes can be collected and the resulting set of root causes yields to the set of root causes in the united system. Using the component decomposition, the size of the HAZID information required can be made lower in cases when similar connected subsystems form the process system to be diagnosed. On the other hand, system-level deviations need to be converted into component deviations by aligning their time and reducing the inputs and outputs in their associated events to component-level inputs and outputs.

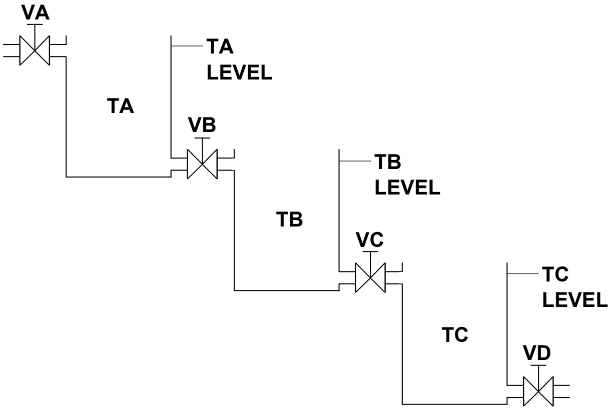

Figure 1: Process system consisting of 3 similar tanks

The diagnosis is then performed by comparing the whole nominal trace with the characteristic trace, and then distributing the deviations (differences) among the components. Before distributing, time alignment and reduction of input and output states to component level are performed (including the component's boundary elements). After the distribution, component-level diagnosis may begin to explore possible (root) causes on the component-level.

The diagnosis proceeds along the flow direction by starting from the first component, takes the deviations, generates the set of possible root causes from them, and then tries to proceed to the next component by checking the starting condition of the next trace fragment - if there is any. If the start condition is fulfilled, the diagnosis continues, otherwise it halts. For example, in the case of *Fig.* 1 the second fragment might have a start condition containing a statement about the minimum level of fluid in tank TA, and in the case of the congestion of valve VA no fluid is coming into a system, therefore, tank TA is not even filling up to the specified minimum level. In this case the diagnosis stops. The result of the diagnostic algorithm is always the union of identified and non-identified root causes created by the consecutive diagnostic algorithm that runs on the components of the consequent nominal trace fragments.

For reference, the whole diagnostic algorithm is presented as a pseudo-code in Algorithm 1. The algorithm collects all root causes (sets $\mathbf{INC}$ and $\mathbf{IRC}$) given a component decomposition, a starting component, and a possibly faulty characteristic trace.

Case study

In the case study the diagnosis procedure for the simple process system in *Fig.* 1 containing three sequentially connected tanks is used. The measured output of the tanks

Algorithm 1 Component-based reasoning procedure

```
$\mathbf{INC} \leftarrow \{\emptyset\}$
$\mathbf{IRC} \leftarrow \{\emptyset\}$
$actualComponent \leftarrow startComponent$
$continue \leftarrow \mathbf{true}$
$shift \leftarrow 0$
while $continue$ do
    $\mathbf{DEV} \leftarrow$ GENERATEDEVIATIONS$(actualComponent,\ chrTrace,\ shift)$
    $\mathbf{FDP} \leftarrow$ COLLECTFINALDEVIATIONPAIRS$(\mathbf{DEV})$
    for all $pair \in \mathbf{FDP}$ do
        $startDeviation \leftarrow proj_1(pair)$
        $startImplication \leftarrow proj_2(pair)$
        REASON$(startDeviation, startImplication, actualComponent.phazid)$
    end for
    if $actualComponent$ has successive component and its start condition evaluates to true then
        $shift \leftarrow length(actualComponent.trace) - 1 + shift$
        $actualComponent \leftarrow$ GETCOMPONENT$(actualComponent.successiveComponent)$
    else
        $continue \leftarrow$ false
    end if
end while
function GENERATEDEVIATIONS(component, chrTrace, shift)
    $\mathbf{DEV} \leftarrow \{\emptyset\}$
    $nomTrace \leftarrow component.trace$
    $reducedTrace \leftarrow$ TRIMTRACE$(chrTrace, shift, shift + component.trace.length)$
    $locChrTrace \leftarrow$ CONVERTTOCOMPONENTLEVEL$(reducedTrace)$
    for $T := 1$ to $length(nomTrace)$ do
        for all deviation $D$ of $locChrTrace$ from $nomTrace$ at time $T$ do
            $\mathbf{DEV} \leftarrow \mathbf{DEV} \bigcup (D)$
        end for
    end for
    return DEV
end function
procedure REASON(deviation, implication, phazid)
    if $\exists R \in$ ROWS$(phazid), deviation = dev_{phazid}(R), implication = imp_{phazid}(R)$ then
        for all $\{R, dev_{phazid}(R) = deviation$ and $imp_{phazid}(R) = implication\}$ do
            if $cause_{phazid}(R) \in$ RC then
                $\mathbf{IRC} \leftarrow \mathbf{IRC} \bigcup cause_{phazid}(R)$
                return
            else if $cause_{phazid}(R) \in \mathbf{DEV}$ and $cause_{phazid}(R) \prec dev_{phazid}(R)\ in\ \mathbf{DEV}$ then
                REASON$(cause_{phazid}(R),\ dev_{phazid}(R),\ phazid)$
            else
                $\mathbf{INC} \leftarrow \mathbf{INC} \bigcup cause_{phazid}(R)$
                return
            end if
        end for
    else
        $\mathbf{INC} \leftarrow \mathbf{INC} \bigcup cause_{phazid}(R)$
        return
    end if
end procedure
```

Table 2: Tank fill operational procedure.

Time	Input values				Output values		
	VA	VB	VC	VD	TA	TB	TC
1	op	cl	cl	cl	0	0	0
2	op	cl	cl	cl	L	0	0
3	op	op	cl	cl	N	0	0
4	op	op	cl	cl	N	L	0
5	op	op	op	cl	N	N	0
6	op	op	op	cl	N	N	L
7	op	op	op	op	N	N	N

Table 3: Normal fill in a single tank with no faults.

Input valve	Output Valve	Tank Level
op	cl	0
op	cl	L
op	op	N

is the tank level that takes its values from Q_e in *Eq.* (1), and the input variables are the valve positions (**open** (op) or **closed** (cl)).

Components, operational procedure, and nominal trace

Every tank may contain no fluid (the tank level is equal to **no** (0)), may be low on fluid (level value is **low** (L)), or might have normal fluid level (level value is **normal** (N)). In every time instant the level increases by one qualitative magnitude (i.e. from **no** to **low** or from **low** to **normal**) if fluid is coming through the input valve but the output valve is closed. Due to the same size of the valves the effect of fluid flow out of the system is similar, but in the opposite direction (from **normal** to **low** or from **low** to **no**). The valve positions (**open** (op) or **closed** (cl)) can be changed by the operator; they are considered as inputs of the system. Leak in the tank is assumed to be equal to the size of an open valve (i.e. a quite substantial leak).

The considered operational procedure is the initial filling of all the three tanks with fluid (the fill operational procedure in short), and is described in detail in *Table* 2.

The process system can be decomposed into three components, therefore the fill operational procedure can also be partitioned into three identical procedure fragments along the component boundaries (the VB and VC valves).

The fragment abstracted from the three identical trace fragments associated to the three tanks can be observed in *Table* 3. It has only the subset of inputs and outputs which are directly related to the particular tank component - the input and output valve and the tank level.

The corresponding component P-HAZID table can be found in *Table* 4 with some of the component faults and deviations associated to them. Instances of this P-HAZID table are used in the case of all three tanks during diagnosis.

Table 4: P-HAZID table of a single tank component with two valves for a reference trace of *Table* 3. Faults: **TANK-LEAK** is leak of the tank, **POS-BIAS** is the positive bias fault of the level sensor and **NEG-BIAS** is the negative bias failure of the level sensor.

Cause	Deviation	Implication
TANK-LEAK	NH(2;op,cl;L)	NH(3;op,op;N)
TANK-LEAK	SML(2;op,cl,L)	SML(3;op,op;N)
NEG-BIAS	LAT(1;op,cl;0)	NH(2;op,cl;L)
LAT(1;op,cl;0)	NH(2;op,cl;L)	NH(3;op,op;N)
NEG-BIAS	SML(1;op,cl;0)	SML(2;op,cl;L)
SML(1;op,cl;0)	SML(2;op,cl;L)	SML(3;op,op;N)
POS-BIAS	NH(1;op,cl;0)	EAR(2;op,cl;L)
NH(1;op,cl;0)	EAR(2;op,cl;L)	NH(3;op,op;N)
POS-BIAS	GRE(1;op,cl;0)	GRE(2;op,cl;L)
GRE(1;op,cl;0)	GRE(2;op,cl;L)	GRE(3;op,op;N)

Table 5: Tank fill operational procedure with a leak in the second tank TB. The leak caused two different events in the operational procedure related to TB (in **bold**), these differences resulted in the four deviations the diagnosis could start from.

Time	Input values				Output values		
	VA	VB	VC	VD	TA	TB	TC
1	op	cl	cl	cl	0	0	0
2	op	cl	cl	cl	L	0	0
3	op	op	cl	cl	N	0	0
4	**op**	**op**	**cl**	**cl**	**N**	**0**	**0**
5	**op**	**op**	**op**	**cl**	**N**	**0**	**0**
6	op	op	op	cl	N	0	0
7	op	op	op	op	N	0	0

The operation of the diagnostic algorithm

The operation of the diagnostic algorithm is illustrated with the case when a rupture of the second tank is present as a root case (fault).

A characteristic trace with the leak of the second tank can be seen in *Table* 5. The size of the leak is assumed to be larger or equal to the size of an outbound pipe, therefore, the tank cannot fill up and no fluid can flow to the third tank TC.

The starting condition of the TB and TC tank components is the appropriate "normal" level in the preceding tank. In that way it is ensured that diagnosis is done on the operational components only.

The diagnosis of this faulty scenario begins by starting with the first tank component TA. There are no differences (and therefore no deviations) regarding this component. The start condition of the second component is fulfilled, therefore, the diagnosis moves towards the next component TB.

However, in the case of TB the following deviations are found after comparing the nominal and characteristic traces (due to that two consequent events did not happened, instead, two events happened with lower output values at time instant 4 and 5 in the operational procedure):

- **never-happened**(4;**open**,**closed**;**low**)
- **never-happened**(5;**open**,**open**;**normal**)
- **smaller**(4;**open**,**closed**;**low**)
- **smaller**(5;**open**,**open**;**normal**)

The time instant of the deviations are shifted back by 2 units because the second component's first event happens at the third system-level time instant. After that, the diagnosis is initiated on the HAZID table. Searching for the already found deviations and connecting them to possible root causes (as in the case of the original diagnostic idea in [7]) the leak of the second tank can be found.

Because of the lack of fluid in the second tank, the start condition of the third component is not fulfilled, therefore, the diagnostic process halts at this step resulting in a single possible root cause being the **TANK-LEAK**.

Conclusions

A novel component based extension of the single component diagnostic algorithm presented in [7] is described in this paper. Using the extension, the domain of application can be extended to more complex composite process systems. Driven by the decomposition of the overall system into components, the P-HAZID tables used for diagnosis are processed at component level by the diagnostic algorithm. The extended method is efficient in the cases when the overall process system consists of similar small components.

The component-based diagnostic procedure was described on a formal level, along with its proposed pseudocode. A case study for a process system of multiple components and a simple failure was also provided.

The following improvements are planned to extend the component-based diagnostic approach:

- The procedure is based on the assumption that the boundary elements between different components are free of failures. As a future work, this limitation might be removed by using a higher level reasoning above the components (as in the form of a system-level HAZID table, for example).
- At the moment, the algorithm is only working for already coded static event information in order to find faults in the system. Diagnosis would be more valuable if events could be processed dynamically, thus the diagnostic procedure could be executed real-time along with the operational procedures.
- Diagnosis would be more accurate if the derivatives of internal states (e.g. the derivative of the tank level) were present in the events.

Acknowledgements

The research was supported by the Hungarian Research Fund through grant K83440. We acknowledge the financial support of the Hungarian State and the European Union under the TAMOP-4.2.2.A-11/1/KONV-2012-0072.

REFERENCES

[1] CAMERON, I. T., RAMAN, R. Process Systems Risk Management. Vol. 6 of Process Systems Engineering (Elsevier Academic Press, San Diego, CA) 2005

[2] AS IEC 61882-2003: Hazard and operability studies (HAZOP studies) Application Guide

[3] VENKATASUBRAMANIAN, V., RENGASWAMY, R., YIN, K., KAVURI, S. N. A review of process fault detection and diagnosis Part I: Qualitative model-based methods, Computers and Chemical Engineering, *2003*, 27(3), 293–311

[4] VENKATASUBRAMANIAN, V., ZHAO, J. S., VISWANATHAN, S. Intelligent systems for HAZOP analysis of complex process plants, Computers and Chemical Engineering, *2000*, 24(9-10), 2291–2302

[5] SELIGMANN, B. J., NÉMETH, E., HANGOS, K. M., CAMERON, I. T. A blended hazard identification methodology to support process diagnosis, Journal of Loss Prevention in the Process Industries, *2012*, 25 , 746–759

[6] NÉMETH, E., LAKNER, R., CAMERON, I., HANGOS, K. M. Fault diagnosis based on hazard identification results, in Preprints of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Barcelona, Spain, June 30 - July 3, pp. 1515–1520

[7] TÓTH, A., HANGOS, K. M. , WERNER-STARK, A. HAZID information based operational procedure diagnosis method, in 12th International PhD Workshop on Systems and Control, Veszprém, 2012. aug. 27 (A. MAGYAR, ed.), ISBN 978-615-5044-71-7 (University of Pannonia, Veszprem), pp. 1–6 (on CD)