Model based process diagnosis using graph methods

MIKLÓS GERZSON[1], ADRIEN LEITOLD[2], KATALIN M. HANGOS[1,3]
[1]Department of Electrical Engineering and Information Systems
University of Pannonia, Veszprém, HUNGARY
gerzson@almos.vein.hu
[2]Department of Mathematics, University of Pannonia, Veszprém, HUNGARY
leitolda@almos.vein.hu
[3]Process Control Research Group, Computer and Automation Research Institute HAS,
Budapest, HUNGARY
hangos@scl.sztaki.hu

Abstract: A novel structure comparison procedure for discrete event systems described by Petri nets are proposed in this paper for model-based diagnosis that utilizes the graph distance metric method. A priori given structurally different discrete event system models that describe the process in its normal and/or faulty modes are used for model-based determination of the severity of the considered faulty modes. The determination is performed by comparing the Petri nets describing the normal operation of the real process with the Petri nets that correspond to its different faulty modes. This comparison is based on the graph distance metric using the maximal common subgraph of the Petri nets to be compared. The proposed procedure is illustrated on a simple example with three faulty modes.

Key-Words: discrete event systems, structure identification, graph representation, Petri nets, fault isolation

1 Introduction

Model-based techniques [1] are widely used and very popular in control and diagnostic applications because of their efficiency and good performance both for systems with continuous and discrete range spaces. The appropriate models in the discrete range space case are built using the tools and techniques of discrete event systems [2], and these are mainly in the form of Petri nets. When used for model-based fault detection and isolation, one not only needs a model for the normal operation of the system, but also other models are required that describe the considered faulty modes. This gives the possibility to isolate the actual faulty mode from measured data and the structurally different faulty models the comparison of which is the subject of the present paper.

The idea of using model structure identification and comparison of discrete event systems models [3] is not new, but the field has matured only recently by a review paper [4] that focuses on Petri net models used for model-based diagnosis.

As a first step, the distance of the describing Petri net models should be defined that may characterize the severity of the fault if its model is compared to the normal model. As a next step, the actual model of the real operating system determined by process mining [5] from the measured data are compared to the models of the different faulty modes in order to achieve fault isolation.

The aim of this paper is to propose and analyze a graph distance based on the maximal common subgraph for the above purpose, and illustrate its use on a simple manufacturing process with three faulty modes.

2 Basic notions

2.1 Petri net model of a process

Petri nets enable both the mathematical and the graph representation of a discrete event system to be modeled, where the signals of the system have discrete range space and time is also discrete [2]. Petri nets can be used for describing a controlled or open-loop system, for modeling the events occurring in it and for analyzing the resulted model. During the modeling and analysis process we can get information about the structure and dynamic behavior of the modeled system.

One of the main advantages of modeling with Petri nets is the ability of describing sequences of events. In a real system the events occur both in a serial and in a parallel way. In case of parallelism we can distinguish two different situations. In the first case the two or more series of events can take place independently from each other. In the other case only one of the sequences can take place because two or more events have the same precondition, and the occurrence any of them makes this precondition invalid. This kind of parallelism is called *conflict situation*. In the Petri net the conflict can be recognized when the same place is the input of two or more transitions. While the real parallelism can occur in normal operational circumstances, the conflicts mean fault situations in general. Although faults also have their preconditions, but these are frequently invisible for the operator and it seems to be the effect of randomness which event takes place.

For the different application purposes various modifications of the original Petri net were carried out, and several types of nets were introduced by researchers from all over the world since the first application of Petri nets by C. A. Petri. The aim of these modifications is to improve the modeling capabilities of this method. One of them is the class of *work flow nets (WF-nets)* which can be applied for modeling of business processes. In our paper we use the WF-nets for diagnosis purposes [5], i.e. for the determination of faulty operational modes of the investigated system.

2.2 WF- nets

The WF-nets form a special subclass of Petri nets. The most important property of the WF-nets is the *soundness*. Soundness can be interpreted as a correctness criterion of WF-net. A WF-net is sound if all of its firing sequences are sound, and a firing sequence is sound if it terminates properly, i.e. when it terminates only the terminal place has one token.

The criteria of sound WF-nets are as follows: there is (i) a single *Start* place and (ii) a single *End* place in the net, with (iii) every node (places and transitions) being connected to a path directed from the *Start* to the *End* places, (iv) they should be live, (i.e. no dead transitions), and (v) every process (i.e. firing sequence) started by a single token at the *Start* place should finish leaving a single token at the *End* place. The above requirements do not allow to have cyclic behavior and further input places (for example for describing fault indicators) in the system. Furthermore, each of the cycles should be implemented as separate path by inserting artificial *Start* and *End* places, while the faulty modes can be described using separate transitions for each that are in conflict to the transitions belonging to the other modes (resulting a WF-net with conflict). Sound WF-nets are also used as reference models in process mining techniques.

2.3 Describing fault modes using Petri nets

Let us assume that we have the Petri net of the normal operational course in the form of WF-net, the so called *normal reference model*. Let us construct the WF-net of the actual operational course based on the observed data of process. This list of data is called *event log* (see later in sub-section 3.1), and it contains the measured values and the performed actions. The difference between the two nets refers to the deviation from the normal operational course.

If we know the possible faulty cases then we can construct the WF-nets describing them. Comparing these pre-constructed WF-nets with the WF-net recovered from the event log we can establish the most likely operational course and in case of fault the most feasible reason of it can be diagnosed.

2.4. Graph distance metric based on the maximal common subgraph

Algorithms for graph matching include the detection of graph and subgraph isomorphism. However, due to errors and distortions in the input data and models, error-tolerant graph matching methods are needed in many applications. One way to achieve error-tolerant graph matching is to use the notion of *graph edit distance* [7]. Here one introduces a set of edit operations (deletion, insertion, substitution of nodes or edges and so on), and defines the similarity of two graphs in terms of the shortest sequence of edit operations that transforms one graph into the other.

Another approach to cope with errors and distortions is based on the *maximal common subgraph* of two graphs [6].

Let $G = (V, E)$ be a graph, where V is a set of finite vertices, $E \subseteq V \times V$ is a set of edges. The graph $S = (V_S, E_S)$ is a *subgraph of* G, if $V_S \subset V$ and $E_S = E \cap (V \times V)$. The notation $S \subseteq G$ is used to indicate that S is a subgraph of G.

A bijective function $f: V \rightarrow V'$ is *a graph isomorphism* from a graph $G = (V, E)$ to a graph $G' = (V', E')$ if $(v_1, v_2) \in E \Leftrightarrow (f(v_1),\ f(v_2)) \in E'$. An injective function $f: V \rightarrow V'$ is *a subgraph isomorphism* from G to G' if there exists a subgraph $S \subseteq G'$ such that f is a graph isomorphism from G to S.

Let G, G_1 and G_2 be graphs. G is *a common subgraph of* G_1 *and* G_2 if there exists subgraph isomorphism from G to G_1 and from G to G_2. A common subgraph G of G_1 and G_2 is *maximal* if there exists no other common subgraph G of G_1 and G_2 that has more nodes than G. The maximal common subgraph of two graphs G_1 and G_2 will be denoted by $mcs(G_1, G_2)$. Notice that $mcs(G_1, G_2)$ is not necessarily unique for two given graphs G_1 and G_2. Let us denote the number of nodes of a graph $G = (V, E)$ by $|G|$.

The distance of two non-empty graphs G_1 and G_2 is defined according to Eq. (1).

$$d(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{max(|G_1|, |G_2|)} \qquad (1)$$

It can be proved that the above distance measure d fulfills the properties of metric [6].

The graph distance measure based on the maximal common subgraph of two graphs can be defined in case of graphs with labeled nodes and edges, too. Classical algorithms for computing the maximal common subgraph of two graphs is based on maximal clique detection [8] or backtracking [9].

3 Structure identification using graph distances

Petri net models of a manufacturing process can be either constructed a priori from engineering knowledge, or from measured real data. Taking the data set of real process executions, the so called event logs, process mining techniques can be used for process discovery [5]. This section deals with the latter way of construction and with the comparison of the constructed Petri net models.

3.1 Reconstructing Petri net models from event logs

An *event log* is a set of finite event sequences, whereas each event sequence corresponds to a particular materialization of the process. We refer to an event sequence as a trace hereafter. We assume that it is possible to record events in a way that each event refers to an activity (i.e. a well defined step in the process), and each event belongs to a case (i.e. a process instance). In addition, each event can have a performer also referred to as originator (the person who executes or initiates the activity), and events have a time stamp, while they are totally ordered. An event log is used as the starting point for mining.

Having an event log, well established procedures of process mining [5] can be used for constructing a *describing Petri net in the form of a sound WF-net*. It is important to emphasize that an event log contains a set of event sequences that each corresponds to a particular behavior, and the events recorded in a log may have ”measurement errors”, that is, some of them may be omitted or have a perturbed time stamp, for example.

3.2 Comparison of the reference model and the reconstructed models

Two discrete event system models in the same form (e.g. both in Petri net forms) can be compared in two principally different ways.

1. *Comparison in the space of events*
   Here the comparison is performed by comparing the event log generated by the reconstructed model with the one generated by the reference model using some signal norm. As the event sequences (without the timing information but with their labels as symbols) can be seen as strings, the efficient algorithms of string comparison (see e.g. [10]) can be applied.
2. *Comparison in the space of Petri net models*
   Here one compares the structure of the two models by using some general graph comparison methods and related graph distance [6, 7] based thereon. Suppose we have a model of the process in the form of a WF-net *N1* (see subsection 2.2). This model is based on our original concepts about the system and on the experiences resulted from the logs of many executions of the process. Note that this model may describe both normal (i.e. non-faulty) operation courses and operation courses belonging to the known faulty modes (in a form of WF-net with conflict). Based on the workflow log of the actual operational courses and using some mining algorithm we construct another WF-net *N2*. The question is whether *N2* is a subgraph of *N1*. If it is true then we can determine whether the system works under normal operational conditions or we can isolate the fault. If it is not true then probably a new fault has been detected.

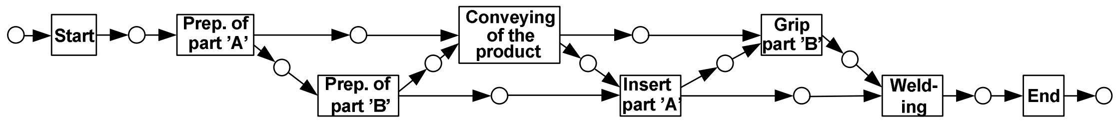

Fig.1. The WF-net G_n of the welding process in case of normal operation

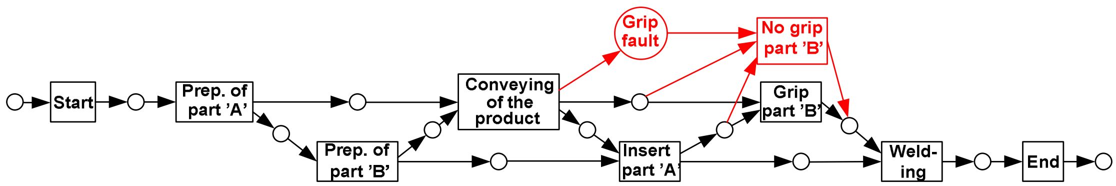

Fig.2. The WF-net G_1 of the welding process with Type 1 fault

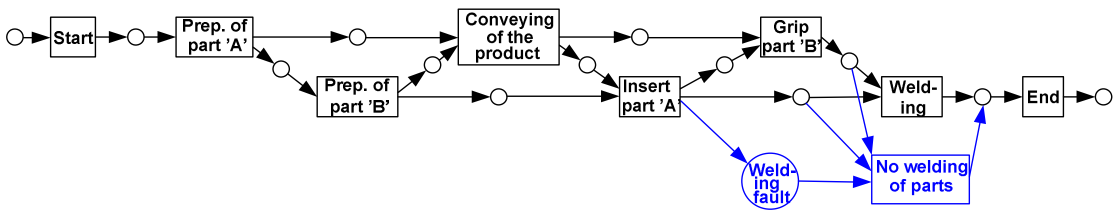

Fig.3. The WF-net G_2 of the welding process with Type 2 fault

4 Simple case study

The above graph comparison and distance determination method has been investigated using a simple illustrative case study. The investigated system is a welding procedure that represents a simple manufacturing process.

4.1 The welding system and its operating procedure

The welding system is a part of an assembly system in which a part ‘A’ and two parts ‘B’ are welded to each other.

Let the *normal operational procedure* is the following. First part ‘A’ is prepared then two pieces from part ‘B’. The welding process takes place in a revolving holder, where the next step is the rotation of this holder to the next position, i.e. the assembled piece of the previous process is removed and an empty slot appeares. When the holder has rotated the part ‘A’ is inserted into the empty slot. Then the two parts ‘B’ are gripped by robot arms and placed close to part ‘A’. Next the welding procedure takes place and the assembled product is ready for conveying to the next station. The above steps of the normal operational procedure of assembling parts ‘A’ and ‘B’ form an event sequence. The describing Petri net of the normal reference model as a sound WF-net is depicted in Fig.1.

Three types of *faulty modes* are considered. The first type of fault relates to the robot arms when they don’t grip well parts ‘B’. It could mean that they grip only one part ‘B’ or none of it. After the welding process the ‘assembled product’ consists of part ‘A’ having one part ‘B’ or having no part ‘B’. The WF-net of the operation with Type 1 fault can be seen in Fig. 2. A second fault possibility is when the welding doesn’t take place completely. Although at the end of this type of process the assembled piece consists of three parts but the assembling is wrong. The Petri net of the process with Type 2 fault is depicted in Fig. 3. These two kinds of faults can take place simultaneously in the same process execution. As a result of the two consecutive faults the product are assembled from part ‘A’ and one piece of part ‘B’ and the welding is also insufficient. The Petri net that corresponds to the two consecutive faults denoted by Type 1_2 is shown in Fig. 4. The third kind of fault is when the revolving holder does not rotate. As a consequence of this Type 3 error the assembled product of the previous step gets two other parts B during the welding and the product consists of five parts: one ‘A’ and four

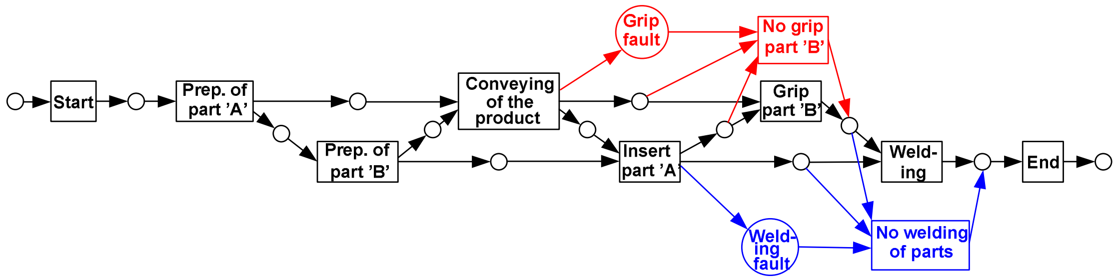

Fig.4. The WF-net $G_{1,2}$ of the welding process with Type 1_2 fault

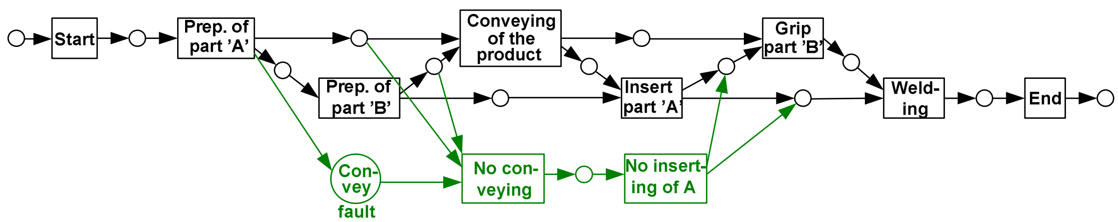

Fig.5. The WF-net G_3 of the welding process with Type 3 fault

'B'. The WF-net of the process with Type 3 faults can be seen in Fig. 5.

4.2 Results of graph comparison

Let G_n denote the WF-net belonging to the normal, error-free operation, and G_1, G_2 and G_3 the WF-nets relating to the three different kinds of error possibilities as they were described in previous section. Let $G_{1,2}$ denote the WF-net describing the double fault mode i.e. when the first and second type errors occur simultaneously.

The severity of the above error possibilities can be characterized by the difference from their corresponding process execution from that of the normal operational mode. The distance between WF-nets of the normal operation and the execution with the given faulty mode can be used for this purpose. Because of the way these models were constructed, the structure of the describing WF-nets exhibits a special property: *every WF-net describing faulty mode or modes contains the WF-net of the normal operation as a subnet.* Consequently, the comparison of the WF-nets describing the normal and one of the faulty modes results in the WF-net G_n of normal operation as a maximal common subgraph. The distance of the WF-net graphs can then be determined according to Eq. (2).

$$d(G_e, G_n) = 1 - \frac{|mcs(G_e, G_n)|}{max(|G_e|, |G_n|)} = 1 - \frac{|G_n|}{|G_e|} \quad (2)$$

where $G_e \in \{G_1, G_2, G_{1,2}, G_3\}$ refers to the net of the appropriate faulty mode.

The distances between the WF-net of the normal operation and the WF-net describing a given faulty mode are summarized in Table 1.

Type of error	Distances between graphs of normal and faulty modes
G_1	$d(G_1, G_n)$ = 1-21/23 = 0,087
G_2	$d(G_2, G_n)$ = 1-21/23 = 0,087
$G_{1,2}$	$d(G_{1,2}, G_n)$ = 1-21/25 = 0,16
G_3	$d(G_3, G_n)$ = 1-21/25 = 0,16

Table 1. Distances between graphs

It can be stated that the third type of error is the most severe from the three error possibilities based on the distances between WF-nets. At the same time, we get the same difference from the normal operation in case of simultaneous presence of the first and second type of error.

This way of comparison of WF-nets can be used not only for the characterization of the difference between the normal and a faulty operation, i.e. for characterizing the severity of a faulty mode, but it is also suitable for fault isolation. By reconstructing the Petri net in WF form from the event logs and comparing it with the predefined nets of normal and different known faulty modes, we can identify the type of error if it has occurred or its seriousness can be determined if an unknown type fault occurs.

5 Conclusion

A novel structure comparison procedure for discrete event systems described by Petri nets are proposed in this paper for model-based diagnostic purposes that utilize the notions and tools of graph distances.

Structurally different discrete event system models describing a system in its normal and/or faulty modes were used as reference models that were compared to each other in order to characterize the severity of the faults. The comparison was performed using a graph distance based on the maximal common subgraph..

The proposed procedure was illustrated on a simple manufacturing process with three faulty modes.

Further work includes the generalization of the structure comparison method for fault isolation using measured event logs from the real process and process mining to construct a describing sound WF-net from the data.

Acknowledgement

We acknowledge the financial support of this work by the Hungarian State and the European Union under the TAMOP-4.2.1/B-09/1/ KONV-2010-0003 project “Mobility and environment: Researches in the fields of motor vehicle industry, energetic and environment in the Middle- and West-Transdanubian Region”. Part of the work is also supported by the Hungarian National Research Fund through project K83440.

References:

[1] Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M.: *Diagnosis and Fault-tolerant Control*, Springer-Verlag, 2006.

[2] Cassandras, C.G., Lafortune, S.: *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, 1999.

[3] Meda, M.E., Ramirez, A., Malo, A.: Identification in discrete event systems, *IEEE International Conference on Systems, Man, and Cybernetics*, 1998, 740–745.

[4] Fanti, M.P., Seatzu, C.: Fault diagnosis and identification of discrete event systems using Petri nets, *9th International Workshop on Discrete Event Systems, WODES 2008*, 2008, 432–435.

[5] van der Aalst, W.M.P., van Hee, K.M.: *Workflow Management: Models, Methods and Systems*, MIT Press, Cambridge, MA, 2002.

[6] Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph, *Pattern Recognition Letters* 19, 1998, 255-259.

[7] Bunke, H. On a relation between graph edit distance and maximum common subgraph, *Pattern Recognition Letters* 18, 1997, 689-694.

[8] Levi, G. A note on derivation of maximal common subgraphs of two directed or undirected graph, *Calcols*, 9, 1972, 341-354.

[9] McGergor, J.J. Backtrack search algorithms and maximal common subgraph problem, *Software Practice and Experience*, 12, 1982, 23-34.

[10] Cohen, W., Ravikumar, P., Fienberg, S.: A comparison of string distance metrics for name-matching tasks, *Proceeding of the IJCAI*, 2003.