Diagnosability evaluation by model-checking

Pascale Marangé, Alexandre Philippot, Jean-Francois Pétin, François Gellot

Diagnosability evaluation by model-checking

Marangé, P.*, Philippot, A.**
Pétin, J.F.* and Gellot, F.**

** CRAN, CNRS UMR 7039, University of Lorraine*
France (Tel: (33) 38-368-4420; e-mail: pascale.marange@univ-lorraine.fr).
***CReSTIC, University of Reims Champagne-Ardenne, France (e-mail: alexandre.philippot@univ-reims.fr)*

Abstract: In order to improve the availability and reliability of manufacturing systems, the diagnosis method is primordial. The literature around the diagnosis of Discrete Event Systems (DES) have proposed different approaches and diagnosability assessment. This paper presents a local modelling of diagnoser and a diagnosability evaluation by Model-Checking. This approach avoids the combinatory explosion problem of global approaches.

Keywords: Diagnosis, Discrete-event Systems, Modelling, Verification.

1. INTRODUCTION

In recent years, researches around diagnosis have expanded in the academic and industrial world due to the increasing complexity of the systems, but also the costs of maintenance policy. To improve the availability and reliability of installations, it is necessary to develop systematic approaches to diagnosis to detect and isolate defaults. Moreover, it has become important to develop approaches for assessing the performance of these diagnosis methods in terms of detection, localization and identification of a fault in a finite delay. Among the diagnosis approaches, literature has shown particular interest around the model-based approaches to the DES diagnosis and the notion of diagnosability.

The aim of this paper is to provide an assessment of diagnosability by model-checking. This approach consists in analyze dependence of local models in order to establish a distribution of the diagnosis. A model checker is then used to verify a number of properties on the failed states reachability. These properties allow us to assess the diagnosability of proposed models. This evaluation is firstly made locally. In case where the system is not locally diagnosable, local diagnoser evolves in a modular diagnoser. An assessment of modular diagnosability is then done. Finally, global diagnosability is checked. In addition, we see that the verification by model checking can assess K-diagnosability and give counterexample to complete the diagnoser. A state of the art on the DES diagnosis approaches and diagnosability notion are listed in section 2. In section 3, the proposed approach to formalize local diagnosers is presented and a diagnosability verification approach by model-checking is exposed. Section 4 illustrates on an academic example, the various concepts discussed in this paper. Before leaving our conclusions and research perspectives, section 5 provides a discussion around the contribution.

2. STATE OF THE ART

The diagnosis field is an important aspect in systems engineering. This importance is not only due to operational safety but also the need to achieve the objectives of maintenance. The objective of this section is to present a state of the art of Discrete Event Systems diagnosis approaches.

2.1 Literature approaches

DES diagnosis approaches can be classified according to the "without model" and "model-based" methods. The methods without model involve the availability of data from recordings made throughout the operation. They often come from expert systems (Tzafestas and Watanabe, 1990), (Alonso-Gonzalez et al., 2010). Therefore, the acquisition of knowledge from experts can be difficult and time consuming before have sufficient knowledge to obtain a reliable diagnosis is uncertain. The model-based methods compare the expected behavior represented by a model of the system, called diagnoser (Sampath, 1995) (Reiter, 1987) (Roth et al., 2009), (Cabasino et al . 2013). The modelling task is often tedious, and quality of the model influence the quality of results returned by the diagnoser. These approaches can also be distinguished by the way the system is modeled (in normal and/or abnormal operation) as well as the modelling tool used (Petri Net, Bayesian Net, automata …). In the context of this paper, the works presented are based on the use of a model-based approach by finite state automata.

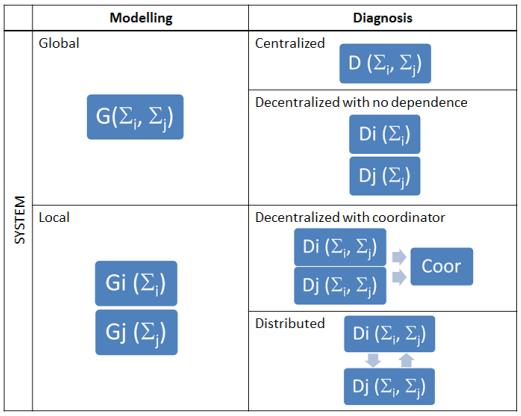
Approaches with representation of the faults in the model are a large part of the literature work. The observer model, often called diagnoser, must inform user of system status in the form of labels (Debouk et al., 2000) (Genc and Lafortune, 2003). Originally proposed in (Sampath, 1995), these approaches have two main steps: Make a model of normal and anormal behavior of the system, after build a labeled diagnoser providing information on the behavior of the system.

These approaches are only discrete in the sense that no other information than that given by the sensors and actuators is present. However, it is sometimes necessary to enrich the

knowledge of the system through temporal or delayed information. The model-based approaches using templates or chronic have been then developed (Holloway and Chand, 1994) (Pandalai and Holloway, 2000) (Milne et al., 1994).

The main trouble of model-based approaches remains in the size of the models to use and to implement. Table 1 shows the classically possible architectures. A global model of the system G can be decompose by local models G_i ($i \in 1,.. n$) in the case of complex systems. Definition of a global diagnoser D containing all the observations of the system (centralized approaches) can be made. But it is possible to obtain the decentralization of information across several local diagnosers D_i ($i \in 1,.. n$). However, when several local diagnosers are present, they should not be contradictory. If their observation Σ_i is exclusively local to the diagnoser D_i, the final decision is then a simple concatenation of local decisions. However, if the local diagnoser requires external information $D_i(\Sigma_i, \Sigma_j)$, we need to ensure the consistency of this information and remove ambiguities making. Therefore, you must use either a decisions coordinator noted Coor (in the form of high-level rules, for example), or communicate the status of this information between local diagnosers (distributed approaches).

Table 1: Diagnosis Architectures



For centralized structure (Sampath, 1995), the disadvantage is the combinatory explosion limiting the application in the case of complex systems. Decentralized and distributed structures can solve this problem (Su and Wonham, 2000) (Qiu, 2005) (Pencolé et al., 2001), but raise other issues in the audit capacity to diagnose all faults.

2.2 Concepts of diagnosability

The use of approaches to diagnosis is essential for complex systems. However, it is important to define whether a system can diagnose with certainty a number of faults in a finite delay. In other words, diagnosability assesses all identified faults are identifiable and locatable in a finite number of events. This is called diagnosability. Indeed, before applying a method on a system, we need to check whether it has sufficient information to perform the diagnosis.

In the Meera Sampath's thesis, a DES is said diagnosable for a set of partitions and for a set of observable events, if it is possible to detect the occurrence of any fault of a partition in a finite delay :

$$(\forall i \in \Pi_f)(\exists n_i \in \aleph)[\forall s \in \Psi(\Sigma_{fi})](\forall t \in L/s) :$$
$$[||t|| \geq n_i \Rightarrow w \in P_L^{-1}[P(st)] \Rightarrow \Sigma_{fi} \in w]$$

where $L/s = \{t \in \Sigma^* \mid st \in L\}$ is the set of all sequences of events after s. $\Psi(\Sigma_{fi})$ is the set of all sequences of events that ends with an event of default in Π_f. $P_L^{-1}[P(st)]$ is the set of all sequences of events that have a projection, an observable sequence of events, equivalent to st in a finite delay ni.

From the work on the decentralized diagnosis (Debouk et al. 2000), authors present a local diagnosis where the objective is to diagnose each component separately and obtain an equivalent diagnosability of the centralized case. This is called local diagnosability where observability is local. However, if two components are each diagnosable locally, the system may not be globally diagnosable with respect to the overall observability of the system.

Regarding distributed structures, joint diagnosability is found in (Qiu, 2005). This is an extension of the co-diagnosability since it is based on the local information of each diagnoser but also the information of neighboring diagnosers. Sometimes called modular diagnosis in literature.

Other definitions exist for diagnosability. In this paper, we summarize the following cases:

1. Local Diagnosability: Failure F_i is said locally diagnosable in a ***subsystem G_i*** iff there exists a finite sequence of observable events ***subsystem G_i*** after the F_i occurrence, F_i is occurred with certainty.
2. Modular Diagnosability: Failure F_i is said modularly diagnosable in a ***subsystem G_i*** and only one iff there exists a finite sequence of observable events of the ***system G*** after the F_i occurrence, F_i is occurred with certainty.
3. Global Diagnosability: Failure F_i is generally said diagnosable in a ***system G*** iff there exists a finite sequence of observable events of the ***system G*** after the F_i occurrence, F_i as occurred with certainty.

2.3 Evaluation of diagnosability

Before checking diagnosability of a system, (Sampath, 1995) has identified two conditions:

1. There is at least one state of the diagnoser which the diagnoser decides with certainty the occurrence of a fault belonging to partition Π_{Fi}.
2. There must not be any cycles called "indeterminate" for which the diagnoser is unable to determine with certainty the occurrence of a fault.

In (Jiang et al., 2001), an algorithm for testing the diagnosability a system has been defined. This is to build for a system G, an automaton G_d by synchronous composition of a diagnoser G_o with himself called twin plant. The algorithm then checks that for every cycle of G_d there diagnoser in a cycle which all states are uniquely labeled. Other methods, for the construction of a non-deterministic automaton G_d in (Yoo and Lafortune, 2002) or empty test in a Büchi automaton in (Tripakis 2008), have been proposed but for centralized approaches.

In the context of decentralized structures (or distributed), very few algorithms concern verifying local modular and global diagnosability. In (Pencolé, 2004), a local checker is constructed for each subsystem. The work of (Saddem et al., 2012) uses the model-checking (Clarke et al., 1999) to check modular diagnosability. In (Cimatti et al., 2004), a formal verification diagnosability by Model-Checking is also available for global approaches. (Grastien 2009) extends this approach to decentralized structures but failed to return a counterexample when the diagnosability is not verified.

In the context of this paper, our work presents an evolutionary approach of diagnosability evaluation by model-checking. Through counterexample analysis, it is possible to propose a transformation of local diagnosers to check the property.

3. DIAGNOSABILITY EVALUATION

In this paper, a distributed diagnostic approach is proposed around a components modelling. The systems studied are manufacturing systems composed of discrete sensors and actuators. The originality is to prevent the global modelling step of the plant G and avoid the risks of combinatorial explosion. Local diagnosers are obtained from local modelling of a component, called Part of Plant (PoP).

After modelling diagnosers, diagnosability is verified at various levels by model-checking.

3.1 Modelling of normal behavior

In industrial processes, a production system is defined as a functional chain composed of a controller (PC) that transmits control signals to the plant and receives sensor values. Plant represents the mechanical part while the controller is the logic one, which describes the desired behavior. This exchange of information between the plant and the controller is the only online observable information. A production system is composed of mechanical elements (actuator / sensor) that interact with each other or not. Each component can be modeled by normal behavior as PoPs. These PoPs models incorporate technical specifications for realistic models (Philippot et al., 2010). Therefore, a component i can be modeled by an automaton $PoP_i = (X_i, \Sigma o_i, \delta e_i, x_{i0}, I_i)$, where X_i is the state space, Σo_i is the set of observable events, δe_i is the function transition, x_{i0} is the initial state and I_i is a set of time intervals where the transition functions is awaited. A transition function δe_i corresponds to an expected logical expression in a time interval I during which an event must occur.

3.2 Local diagnosers and faults partition

A local diagnoser is obtained after identification of all possible faults for each normal condition of each component. This is an analysis by an expert who sets the associated all faults to a label diagnoser. A faults partition Π_{Fj} is associated with several labels indicating the type of failure. To model the fault in the diagnosers, the following two assumptions are considered:

- Only one fault can occur simultaneously on a component (but several on several components);
- The controller is assumed dependable and safe. Therefore, the controller cannot be responsible for a fault. In our case, we use the filter approach to ensure the safety of C (Marangé et al., 2007).

To determine all the possible candidates responsible for a faulty behavior in a PoP, knowledge can come from an analysis of experts and/or documents such as failure mode and effects analysis (FMEA).
Take a diagnoser D in Figure 1, where each state x is composed of a input/output vector V(x) with 3 variables (e_1 e_2 e_3) and where the decision function gives a normal functioning label N, or a label corresponding to a faulty operation $\{F_1, F_2, F_3\}$. From the normal state x_0, the state vector $V(x_0) = (0\ 0\ 0)$, it is possible to distinguish the expected observable events ($\uparrow e_1$), by a function of prediction FPx for example, unexpected events ($\uparrow e_2$, $\uparrow e_3$) possible from this vector, but also to determine the unobservable faults when expected event does not occur within a specified time period (not appearance $\uparrow e_1$).

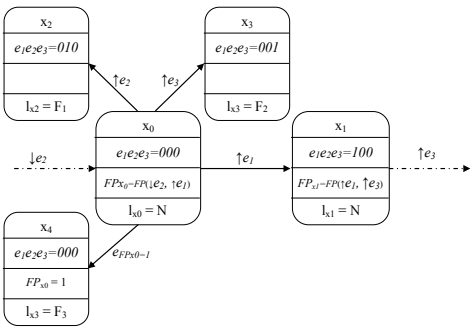


Fig. 1. Extract from a diagnoser D

The faults can then be modeled as observable and/or unobservable. If observable faults seem to be trivial, it is often a source of errors or ambiguities decision in the case of distributed approaches. The problem is to identify the actual cause of an observation, we will continue to handle this case.

3.3 Modelling the diagnoser

More formally, and from the modelling of normal behavior PoP_i and associated faults partiton Π_{Fj}, a local diagnoser can be represented by an automaton $D_i = (X_i \cup XDF_i, \Sigma o_i, \delta_i, x_{i0}, I_i, l_i)$ with X_i, Σo_i, δe_i, x_{i0} and I_i as defined in automaton PoP_i, XDF_i is the set of abnormal states, $\delta_i : X_i \times \Sigma_i^* \rightarrow X_i \cup XDF_i$ is the transition function with the expected functions (δe_i) and unexpected (δu_i) from state x and l_i is the set of functions decision of the local diagnoser D_i with $l_i(x)$ the decision based on the state x. In the state x, the local decision making is carried out as follows

- If the label is $l_i(x)=\{N\}$, the diagnoser decide with certainty the non-presence of a fault.
- If the label is $l_i(x)=\{F_j\}$ the diagnoser indicates with certainty the presence of a fault F_j.
- If the label is $l_i(x)=\{N, F_j\}$, the diagnoser cannot decide and the system is in a uncertainty case.

To define unexpected functions (δu_i), it is possible to define

all the transition functions for 2^n possibilities (with n the number of events and intervals).
The diagnosers are independent of the order, so to ensure that the faults come from the plant and not the order, we combine the diagnosers with a control filter (Marangé et al., 2007). For more information , readers can read the paper (Philippot et al., 2014). However, the mechanical structure of the components and the use of control filters, make impossible some combinations. For example, one interval is 1 at a time, or through control filter, contrary commands cannot be sent. Accordingly, the complexity depends on the granularity of the local models, but also control filter performance (Marangé et al., 2007). The intervals are needed to improve the system diagnosability. For each event occurrence, a corresponding tolerance interval is also defined.

3.4 Evaluation diagnosability by the Model-Checking

After defining diagnosers, it is necessary to evaluate the diagnosability for diagnosis performance. Formal verification approach is proposed to assess the system diagnosability, using a model checker. In contrast to conventional approaches that assess analytically diagnosability, we propose to test the diagnosis models in real situations ensuring that whatever events received by the diagnoser of plant or PC, it is able to determine with certainty the occurrence of a fault. For this, we drew the audit work (Riera et al., 2011), who propose to model the system taking into account the execution environment to ensure that whatever the command sent by PC, the control filter ensures the safety of the plant. These models are not presented in this paper, to focus on the methodology.

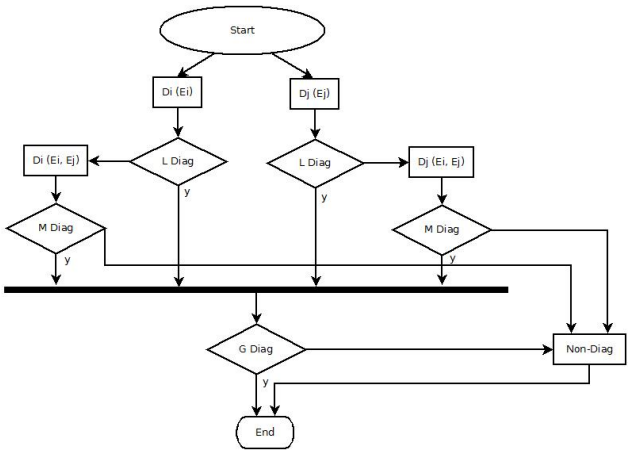


Fig. 2. Flowchart of checking diagnosability

The assessment of diagnosability is according to the flowchart in Figure 2. Each local diagnoser depending only on events PoP is evaluated for local diagnosability of the component. If all component faults cannot be diagnosed, then the local diagnoser is enriched by external events to the component and modular diagnosability is checked. If the verification of ownership is not satisfied, then the conclusion is that the system is not diagnosable tested for the identified partition. Otherwise, the algorithm will check the global diagnosability of system.

For local, modular and global diagnosability tests, we will define a property for each diagnosability condition from the definitions given by (Sampath, 1995):

- The existence of at least one state of the diagnoser which it is decided with certainty the fault occurrence. This condition can be trivial and can be visually verified. However, we can use logic CTL (Clarke et al., 1999) to check the property (1) for each fault F_i

$$EF\ (Fi \wedge \neg N). \qquad (1)$$

ie : in the future a state exist where the diagnoser has no ambiguity in the presence of a defect

- In case of uncertainty between several faults, the diagnoser must be able to solve this uncertainty with the occurrence of new events.
  - First, we will determine if there is a state where two faults are detectable at the same time. For this, we use the property of equation (2).

$$EF(F_i \wedge F_j) \qquad (2)$$

  - If there is an uncertainty state, that is to say that the previous property is checked, we must determine whether it is possible in the future to solve this uncertainty. The property checked is defined in equation (3) where after having a state where F_i and F_j are true at the same time, there is a state where F_i is real and not F_j or vice versa.

$$A((F_i \wedge F_j)\ U\ ((F_i \wedge \neg F_j) \vee (\neg F_i \wedge F_j))) \qquad (3)$$

For the verification of these properties, we model the comportment of the system, control and diagnosers in timed automata. The modelling part is not detailed in this paper. If a diagnoser is diagnosable, all the properties 1 are true, all properties 3 are true if the corresponding property 2 is true.

4. ACADEMIC EXAMPLE.

In this article, we focus on a valve consists of three components: two sensors for closed position *fsc* and open position *fso* and a double acting valve operated by *O* for opening order and *C* for closing order. For this element, it is possible to identify the following faults:

- Sensor *fsc* stuck to 0 (F_1) or to 1 (F_2)
- Sensor *fso* stuck to 0 (F_3) or to 1 (F_4)
- Valve blocked on the sensor *fsc* (F_5) or *fso* (F_6)
- Unexpected passage from 0 to 1 of sensor *fsc* (F_7) or *fso* (F_9)
- Unexpected passage from 1 to 0 of sensor *fsc* (F_8) or *fso* (F_{10})
- Unexpected movements from *fsc* to *fso* (F_{11}) or *fso* to *fsc* (F_{12})

Three fault partitions are defined: Sensor *fsc*: $def\_fsc = \Pi_{fsc} = \{F_1, F_2, F_7, F_8\}$; Sensor *fso*: $def\_fso = \Pi_{fso} = \{F_3, F_4, F_9, F_{10}\}$ and Valve: $def\_V = \Pi_V = \{F_5, F_6, F_{11}, F_{12}\}$

In this example, we apply the methodology of the flowchart

of Fig. 2. To evaluate the local diagnosability of sensor *fsc*, we consider only the set Σ_{fsc}={↑*fsc*, ↓*fsc*, *fsc*, ¬*fsc*} and the partition Π_{fsc} = {F_1, F_2, F_7, F_8}. Local diagnoser is given in Fig. 3 and is obtained by considering only the events Σ_{fsc}. The reachability property of all faults is not verified by equation (1) and therefore, this diagnoser is not locally diagnosable. Indeed, for the same condition, it is possible to be in normal operation or in the presence of one or more fault. It is necessary to introduce an external information to the component and so check modular diagnosability. In this early work , the addition of information to pass the local model the modular model , does not happen automatically. We believe that an expert complete the model, and our work to help check diagnosticity .

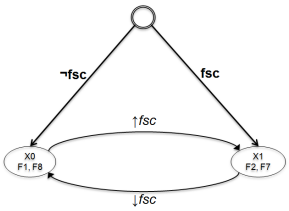


Fig. 3. Local Diagnoser of sensor *fsc*

To detect some defaults, it is necessary to take into account temporal information. However, this information is only available when an order is sent by the controller and when a sensor value change is attempt. If the opening signal is sent to the valve, then *fsc* is deactivated in an interval *t1*, and *fso* activated in an interval *t4*. These time intervals are determined by experts or learning according to the dynamics of the system and the desired behavior (Figure 4).

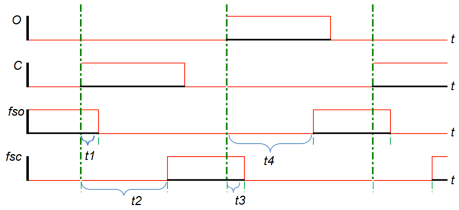


Fig. 4. Temporal interval estimation

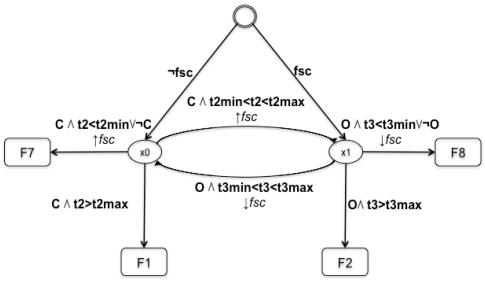


Fig. 5. Diagnoser of sensor *fsc*

From this information, an expert must analyze the possibility of faults occurrence, for each component state, and especially the ability to detect and isolate a fault. Modular diagnoser is initialized from a normal state according to the observation of the first sensors. Modular diagnoser of sensor *fsc* is shown in figure 5. To establish this, we can use the following information: *O*, *C*, *fsc*, ↑*fsc*, ↓*fsc*, ¬*fsc*, *t1*, *t2*. Indeed, depending on the diagnoser state and sent orders, the event occurrence may lead either to a fault (If the sensor is *fsc* to 0, the *O* is sent, the occurrence ↑*fsc* defines the default F_7) or be impossible (in the same configuration, it is impossible to observe the occurrence ↓*fsc*) or either no change (in the same configuration, non-event occurrence does not conclude).

Considering the events and states of the actuator in the diagnoser, reachability properties are checked for all faults, so the diagnoser is modularly diagnosable. We do the same for the actuator and we get the diagnoser in Fig.6. This diagnoser is also modularly diagnosable.

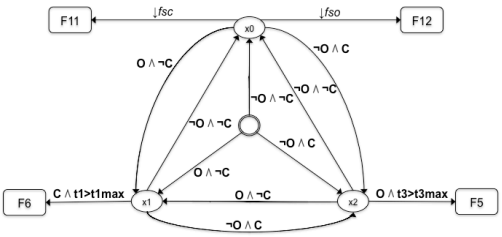


Fig. 6. Diagnoser of actuator

We must now ensure that if this set of three diagnosers is globally diagnosable. For that, we check initially the property of equation (2) with *i* ranging from 1 to 11 and *j* ranging from *i*+1 to 12. This property verifies that the same event cannot lead to two faults of two different components. Four combinations do not have this property (F_2/F_5, F_8/F_{11}, F_4/F_6, F_{10}/F_{12}).

For these four combinations, it is necessary to check the second property (equation 3) to check if these faults can be isolated in the future. For our models, this property is never verified. It means that our models are detectable but not globally diagnosable. High level information to distinguish these combinations must be added by coordinator for example.

5. DISCUSSION

The first advantage is that the local modelling PoPs provides local diagnosers never building global model of the plant. Furthermore, the use of a control filter ensures the safety of the system and thus prevents errors that would result from the controller. This use of the filter implies a simplification of PoPs models by excluding some behaviors.

On the second point, evaluation of the overall diagnosability by Model-Checking seems interesting in order not to rebuild global model to check reachability properties.

Currently, the proposed approach only determines whether the developments of the diagnosis are diagnosable but offers no improvement. For this, we plan to continue this work by filling the approach by modification of the diagnosers to become globally diagnosable. Indeed, in case of non-satisfaction of a property, the Model-Checker can return a trace to the user to analyze if a solution is feasible or not. These traces evolutions to isolate faults may be offered to enrich new local diagnosers Di+.

Furthermore, if the global diagnosability property is checked, an assessment of the number of events required may be returned by the K-diagnosability. This criterion is studied in the works of (Qiu and Kumar, 2008) in the context of collaborative fault detection, or even in (Liu, 2014) with the use of labeled Petri nets.

The model-checking tool used allows us to quantify this assessment but through obtaining some traces and not the shortest. Indeed, obtaining the required minimum number of events involves browse all of the state space and then is liable to a problem of combinatorial explosion.

6. CONCLUSIONS

The paper presents an algorithm for checking the diagnosability for a distributed approach to DES diagnosis. Local diagnosers are obtained from events locally observable by the component. The diagnoser becomes modular when observations from neighboring components are needed to check modular diagnosability. To test this property, an approach by Model-Checking is used to verify the reachability of single fault.

A prospect of this work is a comparative study with conventional approaches to literature around the same benchmark. Furthermore, a dependency analysis on the local diagnosers must be performed. Indeed, the various components are not necessarily dependent on the equipment but can dependent on the product (non-observable events). This dependence on the proceeds may lead to inconsistencies or making false alarms to other equipment. A product diagnoser may be possible.

REFERENCES

Alonso-Gonzalez, C., Moya, N., Biswas, G. (2010). Factoring dynamic bayes networks using possible conflicts. In: Proceeding of the 21th Int. Workshop on Principles of Diagnosis (DX10).

Cabasino M.P., Giua A. and Seatzu C. (2013). *Diagnosis of Petri Nets. Control of Discrete-Event Systems*, Chapter 14, Vol. 433, pp.279-300.

Cimatti A., Pecheur C., and Cavada R. (2003). Formal verification of diagnosability via symbolic model checking. In Int. Joint Conference on Artificial Intelligence (IJCAI'03), pp.363-369.

Clarke E. M., Grumberg O., Peled D. A. (1999). *Model Checking*. The MIT Press, Cambridge, Massachusetts.

Debouk R, Lafortune S and Teneketzis D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete event systems. Discrete Event Dynamic Systems, Vol.10, pp.33-86.

Genc S and Lafortune S. (2003). Distributed diagnosis of discrete-event systems using Petri nets. Lecture Notes in Computer Science, 2679, pp.316-336.

Grastien A. (2009). Symbolic testing of diagnosability. In Int. Workshop on Principles of Diagnosis, pp.131-138.

Holloway, L.E. and Chand, S. (1994). Time templates for discrete event fault monitoring in manufacturing systems. American Control Conference, Baltimore, USA.

Jiang S., Huang Z., Chandra V., and Kumar R. (2001). A polynomial algorithm for diagnosability of discrete-event systems. IEEE Transactions on Automatic Control, Vol.46, pp.1318–1321.

Marangé P., Gellot F., Riera B. (2007). Remote control of automation systems for D.E.S. course, *revue IEEE Transaction on Industrial Electronics Special Section*, pp.3103-3111.

Milne R., Nicol C., Ghallab M., Trave-massuyes L., Bousson, Quevedo J., Dousson C., Aguilar J., and Guasch. Tiger A. (1994). Real-time situation assessment of dynamic systems. Intelligent Systems Engineering.

Liu B. (2014). An efficient approach for diagnosability and diagnosis of DES based on labelled Petri Nets – Untimed and Timed contexts. Thesis, Ecole Centrale de Lille.

Pandalai D. and Holloway, L.E. (2000). Template languages for fault monitoring of timed discrete event processes. IEEE transactions on automatic control, Vol.45(5), pp.868-882.

Pencolé Y., Cordier M.O., and Rozé L. (2001). Incremental decentralized diagnosis approach for the supervision of a telecommunication network. In: Int. Workshop on Principles of Diagnosis (DX01).

Pencolé Y. (2004). Diagnosability analysis of ditributed discrete event systems. Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-04), pp.43-47.

Philippot A., Sayed Mouchaweh M. Carré-Ménétrier. V. (2010). Chapter 16: Component models based approach for failure diagnosis of Discrete Event Systems. Intelligent Industrial Systems: Modelling, Automation and Adaptive Behaviour, IGI.

Philippot, A., Marangé, P., Gellot, F., Pétin, J. F., Riera, B. (2014) Fault Tolerant Control for Manufacturing Discrete Systems by Filter and Diagnoser Interactions. Annual Conference of Prognostic and Health Management Society.

Qiu W. (2005). Decentralized/distributed failure diagnosis and supervisory control of discrete event systems. Thesis, Iowa State University.

Qiu W. and Kumar R. (2008). Distributed diagnosis under bounded-delay communication of immediately forwarded local observations; IEEE Trans. On Systems, man, and cybernetics – Part A: Systems and Humans, Vol. 38 (3), pp628-643.

Reiter R. (1987). A theory of diagnosis from first principles". Artificial Intelligence, Vol.32(1), pp.57-95.

Riera B., Benlorhfar R., Annebicque D., Gellot F. , Vigario B. (2011). Robust control filter for manufacturing systems : application to PLC training. 18th World Congress of the International Federation of Automatic Control, Milano, Italy.

Roth M., Lesage J.-J., and Litz L. (2009). An FDI method for manufacturing systems based on an identified model. Proc. of the 13th IFAC Symposium on Information Control Problem in Manufacturing, INCOM'09, Moscow, Russia, pp.1389–1394.

Sampath M. (1995). A discrete Event Systems Approach to Failure Diagnosis. Thesis, University of Michigan.

Saddem R., Toguyeni A. K. A. et Tagina M. (2012). Algorithme d'interprétation d'une base de signatures temporelles causales pour le diagnostic en ligne des Systèmes à événements Discrets. 9ème Conférence Internationale de Modélisation, Optimisation et SIMulation, MOSIM 2012. - Bordeaux, France.

Su R. and Wonham W.M. (2000). Decentralised fault diagnosis for discrete event systems. In Proceeding CISS. TP Transactions on Automatic Control, Vol.45, pp.1620:1638.:1-6.

Tripakis S. (2008). Checking Timed Büchi Automata Emptiness on Simulation Graphs. ACM Transactions on Computational Logic.

Tzafestas S, and Watanabe K. (1990). Modern approaches to system/sensor fault detection and diagnosis. Journal A, Vol.31(4).

Yoo T. and Lafortune S. (2002). Polynomial-Time Verification of Diagnosability of Partially-Observed Discrete-Event Systems. IEEE Trans. on Automatic Control, Vol.47(9), pp.1491–1495.