Checking diagnosability on centralized model of the system

Mehdi Chankate, Alexandre Philippot, Véronique Carré-Ménétrier, Pascale Marangé

Checking Diagnosability on Centralized Model of the System

M. CHANKATE, A. PHILIPPOT and V. CARRE-MENETRIER
CReSTIC
University of Reims Champagne-Ardenne (URCA)
Reims, France
mehdi.chankate@univ-reims.fr
alexandre.philippot@univ-reims.fr
veronique.carre@univ-reims.fr

P. MARANGE
CRAN
University of Lorraine, UMR 7039, Campus Sciences, BP 70239, 54506
Vandoeuvre-les-Nancy, France CNRS, CRAN, UMR 7039, France
pascale.marange@univ-lorraine.fr

***Abstract*—In this work, the problem of checking diagnosability on Discrete Event System (DES) is considered especially in centralized architecture. Many approaches in literature deals with diagnosability using one or more intermediate models. In this paper, we present a new model based diagnosability algorithms in the framework of set theory for deciding diagnosability, without any intermediate constructions or models and considering several faults at the same time.**

***Index Terms*—Diagnosability, Diagnosis, Model based system, Finite state machine, Discrete Event System (DES).**

I. Introduction

In recent years, work around diagnosis attracted the attention of the academic and the industrial world due to the increasing complexity of systems, the costs of maintenance interventions but also avoiding catastrophes and loss of life. To improve the availability and reliability of industrial installations, it is necessary to develop systematic diagnostic approaches to detect and isolate faults. Moreover, it has become essential to implement approaches to evaluate the performance of these methods in terms of detection, location, and identification of a fault in a finite time. This is called diagnosability. Among the diagnostic approaches, the literature has shown an interest in model-based approaches for the diagnosis of Discrete Event Systems (DES) [1].

The concept of diagnosing using the observer of the system appeared for the first time in the work of [2]. It is also in these works that the notion of "diagnosability" appeared, which means evaluating the possibility to guarantee the diagnosis of a fault in a finite time. The diagnosability becomes essential information on the system. A fault is defined as any deviation of the system from its specified behavior. The fault diagnosis is the process that detects and identifies the fault and its type based on the observable symptoms. Various evolution of the diagnosability is found in the literature, notably according to the structure of the system and its modeling [3], [4], [5]. But regardless of this diagnosability, it is essential to be able to evaluate its capacity. In [2], diagnosability is checked by detecting fault-indeterminate cycles in a global diagnose. In [6], the detection of such behaviors is based on the twin plant construction and in [7] it is based on the construction of verifiers.

In [8], an empty test in a B*ü*chi automaton has been treated as centralized approaches. The check of diagnosability is done in two stages: the construction of the automaton B based on two models one represents the whole system, while the other conserves only the nominal behavior of the system, then a composition is done. And an algorithm looking for a cycle that proves the system is not diagnosable is applied. In [9], the same diagnosability problem is proposed to be solved as a model-checking problem by the use of two models of the system, the first represents the global behavior of the system and the second considers only the faulty behavior in order to reduce the number of state of the synchronous composition, several symbolic model-checking algorithms are proposed to test diagnosability using twin plant construction with fairness properties.

In [10] a variant of the classical diagnoser is introduced to explicitly separate the normal states from the faulty ones in each node of the diagnoser. Most of these approaches highlight efficient resolution algorithms as long as the system remains reasonable and the fault partition is very small (often a single type of fault considered). The approach presented in this paper is based on a verification method of monolithic diagnosability in the case of centralized systems. This proposal allows, in particular, the analysis of the diagnosability from the overall model of the system without the construction of intermediate models. We are interested in this work to determine a solution in a centralized architecture to prove the feasibility of the approach. In section 2, some notions on the modeling of DES and related operations are recalled, followed by the definition of diagnosability. Section 3 focuses on the proposed approach of verification of diagnosability on the centralized systems. A comparison between our approach and main approaches is discussed in the section 4. Finally, we conclude the paper with prospects.

978-1-7281-2292-2/19/$31.00 ©2019 IEEE

II. PRELIMINARIES

A. DES Modeling

To study the logical behavior of DES, one of the formal ways is based on the language theory and automata. Where, the temporal aspect of the system is ignored, only the order and identity of the events generated by the system are considered. Therefore, a logic model where only the sequences of events are manipulated is perfectly adapted to this type of problem. Discrete-event systems are quite convenient to perform the safety analysis of complex systems in a sufficiently high abstraction level.

Let $G = (Q, \Sigma, \delta, q_0)$ be a finite state machine model of the system to be diagnosed, where Q is the set of states in the model, the finite set of events with two subsets: Σ_o denote the set of observable events and Σ_{uo} denote the set of unobservable events such that $\Sigma = \Sigma_o \cup \Sigma_{uo}$. δ is the partial transition function $(q, \sigma, q') \in Q \times \Sigma \times Q$ with $q' \in \delta(q, \sigma)$ and q_0 is the initial state of the system. The model G accounts for the normal and failed behavior of the system. The behavior of the system is described by the prefix-closed language $L(G)$ to denote the generated language of G. $L(G)$ is a subset of Σ^*, where Σ^* denote the Kleene closure of the set Σ. In the context of the diagnosis, faults are basically assumed to be unobservable events $\Sigma_f \subseteq \Sigma_{uo}$, the set of fault events can be partitioned as disjoint fault classes $\Sigma_f = \Sigma_{f1} \cup \Sigma_{f2} \cup ... \cup \Sigma_{fm}$ denoted by $\prod_f = \Sigma_{f1}, ..., \Sigma_{fm}, m \geq 1$.

Since a diagnoser is a model based on the observable events of the system, then it is necessary to recall the operation to capture the observed behavior. Therefore, the projection function of all the observable events by $P : \Sigma^* \rightarrow \Sigma_o^*$, i.e. The effect of P on an event sequence $s \in \Sigma^*$ is simply to erase the unobservable events in it. Generally, $P(\sigma) = \sigma$ if $\sigma \in \Sigma_o$ and $P(\sigma) = \epsilon$ if $\sigma \in \Sigma_{uo}, P(s.\sigma) = P(s)P(\sigma)$ with $s \in \Sigma^*$ and $\sigma \in \Sigma$. The inverse projection P_L^{-1} is defined by $P_L^{-1}(y) = \{s \in L \subseteq \Sigma^* : P(s) = y\}$. Moreover, a synchronous composition of two models G_1 and G_2, noted $\|$, is a well-known operation denoted, $G_1 \parallel G_2 = (Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta, q_1 \times q_2)$. It represents the evolution of the global model.

B. Definition of Diagnosability

The diagnosability of a system has been formulated in the seminal work of [2]. Based on the construction of a global diagnoser as an automata and their extensions [2] and [4], or Petri nets. Checking diagnosability is equal to look for the conformity with two sufficient conditions:

- C1. There is at least one state of the diagnoser for which the diagnoser decides with certainty the occurrence of a fault belonging to a partition $\prod_{fi}$.
- C2. There must be no so-called indeterminate cycles for which the diagnoser is unable to decide with certainty the occurrence of a fault belonging to a partition $\prod_{fi}$.

In addition, the following two assumptions must be accepted:

- The automaton G is alive, i.e., that from each state q of G, there is at least one possible transition;
- There is no cycle containing only unobservable events.

The absence of unobservable event cycles makes it possible to ensure that regular observations are available to observe the effects of a fault. $L(G)$ is therefore diagnosable with respect to the projection $P : \Sigma \rightarrow \Sigma_o$ and with respect to the partition $\prod_f$ of Σ_f, if the following condition is satisfied.

$$(\forall_i \in \prod_f)(\exists_{ni} \in N)[\forall s \in \Psi(\Sigma_{fi})](\forall t \in L/s) : [\| t \| \geq n_i \implies \omega \in P^{-1}L(G)[P(st)] \implies \Sigma_{fi} \in \omega] \quad (1)$$

With :

- $\Psi(\Sigma_{fi})$ the set of traces ending in a fault of Σ_{fi}.
- L/s the set of suffix sequences of s in $L(G)$.
- $\| t \|$ the length of the sequence t.
- $P^{-1}L(G)[P(u)]$ all traces v in $L(G)$ such that $P(v) = u$.

$\Psi(\Sigma_f)$ is the set of finite sequences of events that end with a faulty event of (Σ_{fi}). This definition means that an L language is diagnosable if and only if, for any sequence s terminating with an offending event and any sequence of events t of s sufficiently long ($\mid t \mid \geq ni$), then any other sequence of events ω having the same observable projection as $(P(s.t) = P(\omega))$ necessarily contains a faulty event of Σ_f. The idea of the cycle is the main element in diagnosability check, so it is necessary to recall it as follows:

Property 1. (Cycle in G)

A cycle is a path of transitions and states wherein a state is reachable from itself. In our application, we denote by CL be the set of possible cycles cl in a system G such that $(q \in Q, \sigma \in \Sigma)$:

$$cl = (q_0, \sigma_0, q_1, \sigma_1, ..., q_p, \sigma_p, ..., q_p).$$

To check the diagnosability in the model G of the system it is necessary to check the absence of two cases:

- The system G is not diagnosable if it exists a cycle cl_1 in G with fault F_i and another cycle cl_2 not containing the fault F_i with respect to the same observable projection.

$$(\exists cl_1 \subseteq CL/\exists(q_j, \sigma_j \in L : \sigma_j \in F_i)(\exists cl_2 \subseteq CL/ \forall(q_k, \sigma_k) \in L : \sigma_k \in \Sigma_0)/P(cl_1) = P(cl_2) \text{ avec } j \in [0, n], k \in [0, p]. \quad (2)$$

- The system G is not diagnosable if it exists a cycle cl_1 in G with fault F_i and another cycle cl_2 with another fault F_j ($F_i \neq F_j$) and the cl_1 and cl_2 are sharing the same observable projection.

$$(\exists cl_1 \subseteq CL/\exists(q_j, \sigma_j \in L : \sigma_j \in F_i)(\exists cl_2 \subseteq CL/ \forall(q_k, \sigma_k) \in L : \sigma_k \in F_{s \neq i})/P(cl_1) = P(cl_2) \text{ avec } j \in [0, n], k \in [0, p]. \quad (3)$$

In the literature, the monolithic diagnosability is analyzed first in the frame work of automata, and depends on both the observable event and the set of faults in the system. Section 3 starts with definitions of the sets used in the proposed approach then presents the algorithm designed.

III. CHECKING DIAGNOSABILITY ON CENTRALIZED MODEL OF THE SYSTEM

The proposed approach is based on previous work on the diagnosability [11], it aims to avoid the construction of intermediate models and automata compositions by using class instances (G_1 and G_2) of the model G, then properties of diagnosability are verified in this basis using set theory.

A. Definitions

The analysis of diagnosability is based on the set theory, so it is necessary to define sets that are used as following.

Definition 1.
Let Sv denote the finite set of all the cycle sequences in G starting from the initial state q_0, where σ_i is the event related to a transition with respect to the event index i, where Sv_i denotes the sequences of observable and unobservable events constituting the cycle i from q_0.

Definition 2.

- Ev denotes the finite set of all observable sequences related to all cycles in G.
- Ev_i is the sequence i in G which is the concatenation of all the observable events occurred in the cycle i, starting from the initial state q_0.
- Ev_{c_i} is the sequence of all observable events belonging only to the loop in the cycle i.

Definition 3.
Sv_{Ev_i} is a sub-set of Sv which is named sequence set sharing the same observable sequence of Ev_i.

Definition 4.
Let Qv denotes the finite set of states visited in all cycles. We note that Qv_i is the set containing all the states visited in the cycle i.

Definition 5.
Let Fv be the finite set of faults in all cycles, where Fv_i is the set of faults occurred in the cycle i.
The definitions above will be illustrated in section III.C

B. Proposed approach

Figure 1 illustrates the different steps to check diagnosability based on the centralized model of the system. Model G is instantiated twice (G_1,G_2). These models account for both the normal and the faulty behavior of the system. The approach is detailed in five steps in Figure 1.

- **Step1: Verification of assumptions and research of the cycle.** (Figure 2)

After the activation of the instance G_1, the cycle search begins after the execution of the first possible transition (σ_i) of the system from its initial state q_0. After the occurrence of an event (observable or not), a test condition (4) is called to check if a cycle is detected or not as follows with (k) and $(k-1)$ refer to the step for the event or state.

$$\sum_{i=1,k=1}^{m,|Q|} q_i(k) \cap Q_{v_i}(k-1) \neq \emptyset \quad (4)$$

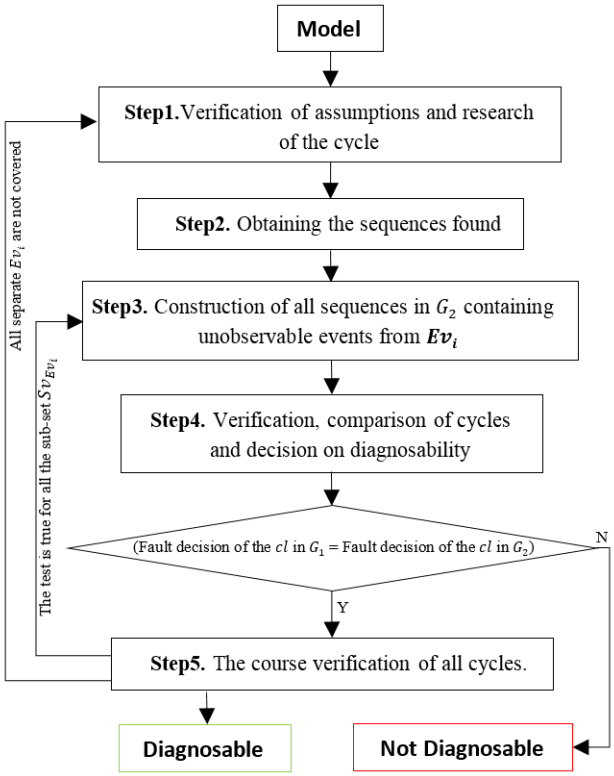

Fig. 1. The proposed approach

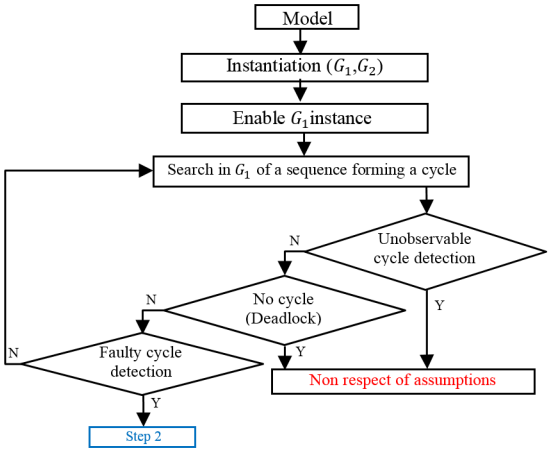

Fig. 2. Verification of assumptions and research of the cycle

With: m is the number of cycles in the model G, q_i (k) is the last visited state, and Qv_i $(k-1)$ is the set of states visited (q_0 to $q_{(k-1)}$). The test is based on the set of states visited with respect to the first instance G_1. In words, during the evolution of the system, if the new visited state q_i (k) already exists in the set of Qv_i $(k-1)$ that means a cycle detection $q_i(k) \cap Q_{v_i}(k-1) \neq \emptyset$. To have more information about the type of cycle detected, several cases are possible:
Case 1. The detected cycle (faulty or not) is formed only by

unobservable events.

$$\sum_{i=1,k=1}^{m,|Q|} q_i(k) \cap Q_{vi}(k-1) \neq \emptyset \quad and \quad Ev_i \neq \emptyset \qquad (5)$$

Case 2. The model has a deadlock during its evolution, it means that the model does not respond to the first hypothesis of liveliness,

$$\forall q \in Q \quad \delta\ (q, \sigma) = q' \qquad (6)$$

In case 1 and case 2 the verification of the diagnosability is impossible because of the non-respect of the assumptions in section II.

Case 3. The detection of a normal cycle in G_1 means that the algorithm must look for another cycle with fault in instance G_1.

Case 4. The detection of a faulty cycle (one or more type of faults) in G_1 means that the existence of all observable and unobservable events Sv_i forming this cycle. And the rest is to determine the sequences for the verification.

- **Step2: Obtaining the sequences found.** (Figure 3)

Based on the detection of the faulty cycle in G_1 (case 4), a requested backup of useful information related to the observable sequence is needed, by applying a projection and detection of repeated loops on Sv_i ($Ev_i = P(Sv_i)$). Then a call to memorize Ev_i, the loop cycle Evc_i and the faults appeared Fv_i is done.

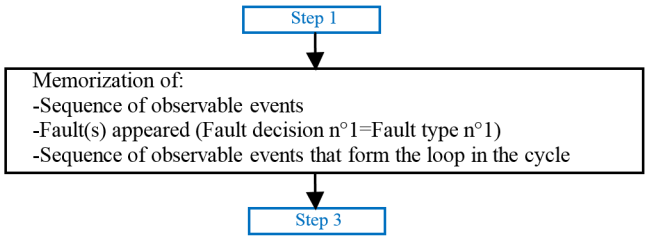

Fig. 3. Obtaining the sequences found.

- **Step3: Construction of all sequences in G_2 containing unobservable events from Ev_i.** (Figure 4)

The search of the cycle in G_2 respects the order of execution using Ev_i and makes it possible to execute unobservable events accepted in the sequence of the future cycle. The search for the inverse projection of Ev_i is necessary in order to get a sub-sequence Sv_{Ev_i} accepted in $L(G)$. The idea is to find a sequence Sv_j with observable and unobservable events, this sequence must form a cycle in G_2. The inverse projection of Ev_i is defined as follows:

$$P_L^{-1}(Ev_i) = \{Sv_j \ \in L \subseteq \Sigma^* : \ P(Sv_j) = Ev_i\}$$

This inverse projection is ensured by the algorithm using (7). On the basis of the sequence Ev_i saved in G_1, a construction is set up depending on two elements: an observable events $\sigma_{Ev_i}(k')$ of Ev_i with respect to the order in the set, or a possible unobservable event $\sigma_{Nobs_{possible}}$. This specific execution is ensured as follows:

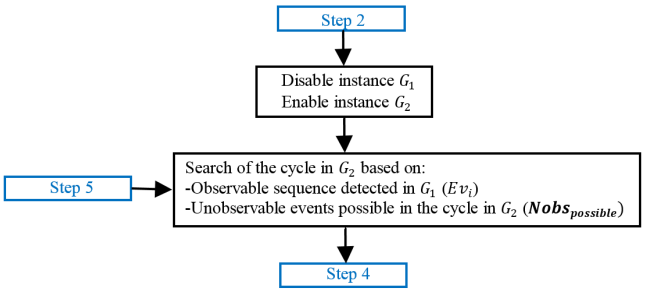

Fig. 4. Construction of all sequences in G_2 containing unobservable events from Ev_i.

$$\sigma_j(k) = \begin{cases} \sigma_{Ev_i}(k') \\ \sigma_{Nobs_{possible}} \end{cases} \qquad (7)$$

With k is the index of the event generated in Sv_j and $k' \in [1\ , Dim(Ev_i)]$. Dim is the size of the set Ev_i and $Nobs_{possible}$ denotes unobservable events possible to occur in the cycle. If the Ev_i sequence is executed completely and the cycle is not yet detected then Evc_i will be looped until the cycle detection or the achievement of $|Q|$ states which represents the longest cycle.

- **Step4: Verification, comparison of cycles and decision on diagnosability** (Figure 5)

After the detection of a cycle in G_2, which shares the same observable sequence Ev_i, a step relating to the fault decision is needed to decide on diagnosability. The objective is to determine the fault types of the two cycles detected. This test is the analysis of the faulty sets to decide whether the two decisions of fault share the same type of fault or not.

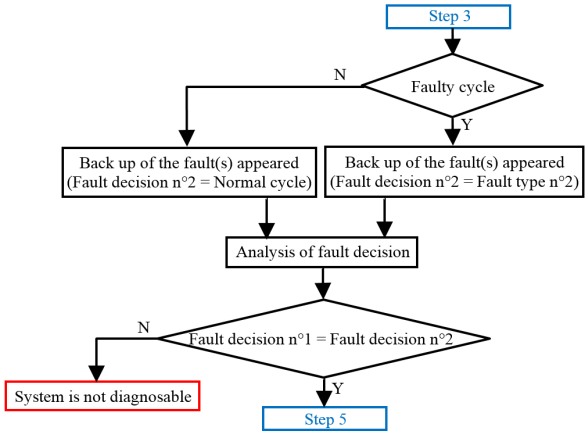

Fig. 5. Verification, comparison of cycles and decision on diagnosability.

Faults are represented by indexes and each index belongs to a fault partition or several indices belong to the same partition. For that we carry out two operations: the product ($Pdtc$) represents the product of the indices of faults detected in the cycle and the sum (Smc) means the sum of the indices relating to the faults for each cycle detected in G. The products ($Pdtc_1$,$Pdtc_2$) and the sums (Smc_1,Smc_2) are the decision

makers for each cycle detected in G_1 and G_2. The fault decision is defined as follows:

$$Pdtc = \prod_{k=1}^{|F|} F(k) \quad Smc = \sum_{k=1}^{|F|} F(k) \tag{8}$$

Fault decision in G for a cycle cl

If the two decisions are different i.e. ($Fault\ decision\ n^o1 \neq Fault\ decision\ n^o2$), in other ways ($Pdtc_1 \neq Pdtc_2$) OR ($Smc_1 \neq Smc_2$) then it means that two sequences have the same observation, but the cycle in G_1 has a type of fault and for the cycle in G_2 has a different type of fault (3) or without (2), so both cycles have two different fault decisions and therefore the system G is Not Diagnosable. On the other hand, if the two decisions are equal ($Fault\ decision\ n^o1 = Fault\ decision\ n^o2$), which means ($Pdtc_1 = Pdtc_2$ AND $Smc1 = Smc2$) then the fault decision is the same and the algorithm looks for another cycle of the sub-set Sv_{Ev_i} of G_2.

- **Step5: The course verification of all cycles.** (Figure 6)

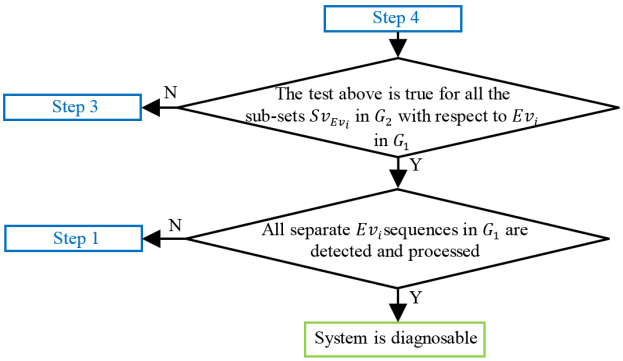

Fig. 6. The course verification of all cycles.

If all cycles of Sv_{Ev_i} in G_2 relating to the observable sequence Ev_i detected in G_1 are covered, it means that the test of fault decision is true for all the sub-sets Sv_{Ev_i} in G_2. Then the algorithm looks for a new sequence Ev_{i+1}. In every new search for a cycle in G_1, the algorithm checks whether the new sequence found ev_i is already processed in the previous sequences of Ev_i (9).

$$\sum_{i=1,k=1}^{m,(i-1)} ev_i \ \cap \ Ev_k \ \neq \varnothing \tag{9}$$

If the new sequence ev_i already exists in the set of Ev_k then the algorithm looks for the next sequence ev_{i+1} and then the check (9) is recalled until the algorithm finds a new sequence never occurred before in all sets of Ev_k. The system is **Diagnosable** if for all the sequences found Ev_i in G_1 and all the sub-sequences found Sv_{Ev_i} in relation to Ev_i in G_2, any case does not involve at least once the property ($Fault\ decision\ n^o1 \neq Fault\ decision\ n^o2$)

C. Application example

To evaluate and compare the proposed approach, let us consider the automaton G in figure 7 from the work of [2]. The subsets of the events are $\Sigma_o = \{a, b, c\}, \Sigma_{uo} = \{uo, f_1\}, \Sigma_f = \{f_1\}$.

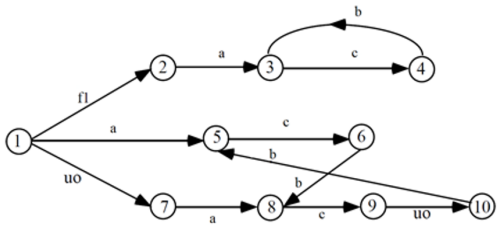

Fig. 7. Model G of the example

The model G is the system to be diagnosed and it is characterized by the presence of a single fault f_1 and a single unobservable event uo. To decide on diagnosability of G the steps presented must be followed.

Step 1. The system G respects the assumptions of section II. We first instantiate the model G twice (G_1,G_2), and the cycle search begins after the execution of the first possible event (f_1 or a or uo) in G_1, i.e. one cycle of CL is going to be started. Then a test condition (4) is called to check if a cycle is detected, else the next possible event is occurred until the detection of a cycle. According to (4) the system is composed by three cycles (CL):

$$cl_1 = (q_1, f_1, q_2, a, q_3, c, q_4, b, q_3)$$
$$cl_2 = (q_1, a, q_5, c, q_6, b, q_8, c, q_9, uo, q_{10}, b, q_5)$$
$$cl_3 = (q_1, uo, q_7, a, q_8, c, q_9, uo, q_{10}, b, q_5, c, q_6, b, q_8)$$

In the case where the first cycle search in G_1 detects the cycle cl_1 or cl_2 which are non-faulty cycles, the algorithm decides to look for another cycle in G_1 until find a faulty cycle (case 3 in step1).

Step 2. According to definition.1, Sv_i is the sequences of observable and unobservable events constituting the cycle i.

$$Sv_1 = (f_1, a, c, b)$$
$$Sv_2 = (a, c, b, c, uo, b)$$
$$Sv_3 = (uo, a, c, uo, b, c, b)$$

To find all the observable events occurred in the cycles Ev_i, a projection on Sv_i is done, then a recovery of sequence Evc_i is applied. Although we have three cycles in this example, the sets Ev_2 and Ev_3 share the same observable sequence. So, we have:

$$Ev_1 = (a, c, b), Evc_1 = (c, b)$$
$$Ev_2 = (a, c, b, c, b), Evc_2 = (c, b, c, b) = (c, b)$$

Lets take the detection of the first cycle cl_1 in G_1, this one, is a faulty cycle considering the presence of f_1 in Sv_1, and consequently the algorithm continues the rest of steps since a faulty cycle in G_1 is detected, parallel to the cycle detection, Ev_1 is determined by $Ev_1 = P(Sv_1)$, the sequence of the loop in the detected cycle $Evc_1 = (c, b)$ is determined and the memorization of the faults appeared $Fv_1 = (f_1)$ is done.

Step 3. The observable sequence of Ev_1 is applied in G_2 by respecting the order of the sequence and adding all possible unobservable events. In other words, according to (7), the algorithm run $\sigma_j(k)$ wich is the first event a in Ev_i or uo from the list of all unobservable events possible in this cycle. One of these events will activate the next state, then (4) is called for the cycle detection. In absence of cycle, another event $\sigma_j(k+1)$ is called based on c or uo then the test of detection cycle is recalled until it becomes true. As a result of the inverse projection, all the possible cycles in G_2 which are likely to be covered is the sub-set Sv_{Ev_i}.

$$P_L^{-1}(Ev_i) = \{Sv_1, Sv_2, Sv_3\} = Sv_{Ev_i}$$

Step 4. According to definition.3, and thanks to the share of the same observable sequence Ev_1 and the same sequence in the loop of cycle Evc_1. The evolution of the model G will detect for example the cycle cl_2, which is a not faulty cycle since $Fv_2 = \varnothing$ then the test of fault decision is called to analyze the labels relating to the two cycles cl_1 and cl_2, in this case the system is non-diagnosable according to (2) and (4).

$$\begin{array}{ll} Pdtc_1 = f_1 & Smc_1 = f_1 \\ Pdtc_2 = 0 & Smc_2 = 0 \end{array}$$

Step 5. Since the system G has only one faulty cycle, step 5 will not be automatically executed, because the model of the system accepts a single set Ev_1, this step is important if the model contains more than one set Ev_i, that means several subsets in Sv_{Ev_i}, in this case steps 3 and 4 must be checked for all Sv_{Ev_i} sequences and steps 1 and 2 must be checked for all Ev_i sequences.

To check the diagnosability in this example the best case is to cover the faulty cycle cl_1 in G_1 formed by 5 states then cl_2 in G_2 formed by 7 states so globally 12 states to decide on diagnosability. The worst case is to cover the normal cycle cl_2 formed by 7 states in G_1, then the second normal cycle cl_3 formed by 8 and then the faulty cycle cl_1 with 5 states in G_1. After that, the same faulty cycle cl_1 formed by 5 states in G_2 then cl_3 composed by 8 states in G_2, i.e. 33 states to decide on diagnosability.

If the unobservable event uo becomes a faulty event, so the system becomes diagnosable. In this case all the cycles in G are faulty and share the same Ev_i, therefore, the algorithm in the worst case will cover cl_3 with 8 states in G_1 then compare it with cycles in G_2: cl_1 formed by 5 states, cl_2 with 7 states and cl_3 formed by 8 states, in global 28 states.

IV. DISCUSSION

We note that the classical approaches require the construction of an intermediate model whereas the approach proposed in this paper does not require it. The presence of several types of faults is considered in the proposed approach. The algorithm has been tested on several examples of different sizes with different faults, and we are able through this method to identify certain, uncertain and ambiguous faults cases. In the aim of detecting possible limitations, the use of the algorithm on complex examples takes longer time and sometimes does not give a decision on the diagnosability.

V. CONCLUSION

We have first used the conditions of checking diagnosability and applied them directly on the model of the system without construction of intermediate models and considering several faults. This article presents diagnosability analysis in the context of set theory. To test the approach, a centralized example with a single fault in the literature is studied, where G_1 and G_2 are instances of G. We tested other examples with several faults, and the approach showed interesting results consistent with those of the literature.

This study should be continued to show the possibility of directly detecting cycles, then switch to larger systems to compare our approach in terms of computed time, search algorithm and state space. What seems interesting in this work is to arrive at the same decision of diagnosability based on the same system structure, consider several faults at the same time, and the uncertainty cases without need to build intermediary models. In the same concept, we tested whether it was possible to apply the approach in the decentralized architecture. In this case G_1 is instantiated for one of the models and G_2 on the other model. Again, the first results look promising.

REFERENCES

[1] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*, 2nd ed. New York, NY: Springer, 2008, oCLC: 255370614.

[2] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, Sep. 1995.

[3] O. Contant, S. Lafortune, and D. Teneketzis, "Diagnosis of modular discrete event systems 1," *IFAC Proceedings Volumes*, vol. 37, no. 18, pp. 327–332, Sep. 2004.

[4] R. Debouk, S. E. Lafortune, and D. Teneketzis, "Coordinated Decentralized Protocols for Failure Diagnosis of Discrete Event Systems," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 10, no. 1-2, pp. 33–86, 2000.

[5] Wenbin Qiu and R. Kumar, "Decentralized failure diagnosis of discrete event systems," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 36, no. 2, pp. 384–395, Mar. 2006. [Online]. Available: http://ieeexplore.ieee.org/document/1597408/

[6] Shengbing Jiang, Zhongdong Huang, V. Chandra, and R. Kumar, "A polynomial algorithm for testing diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 46, no. 8, pp. 1318–1321, Aug. 2001.

[7] Tae-Sic Yoo and S. Lafortune, "Polynomial-time verification of diagnosability of partially observed discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 9, pp. 1491–1495, Sep. 2002.

[8] F. Cassez and S. Tripakis, "Diagnostic des systèmes temporisés," in *Systèmes embarqués – Approches formelles*, ser. Traite IC2, Roux, O. H., Jard, and Claude, Eds. Hermes Science, Oct. 2008, pp. 145–176.

[9] A. Grastien, "Symbolic Testing of Diagnosability," *The 20th International Workshop on Principles of Diagnosis (DX-09)*, no. 131-138, p. 8, Jun. 2009.

[10] A. Boussif, M. Ghazel, and K. Klai, "Combining Enumerative and Symbolic Techniques for Diagnosis of Discrete-Event Systems," in *VECOS 2015 - 9th Workshop on Verification and Evaluation of Computer and Communication Systems*, Bucarest, Romania, Sep. 2015, p. 11p.

[11] M. Chankate, A. Philippot, V. Carre, and P. Marange, "Conception D'un Système De Vérification De La Diagnosticabilité Par Model Checking A Partir Du Modèle Du Système," *12th International Conference on Modelling, Optimization and SIMulation MOSIM18*, no. 157-164, p. 8, Jun. 2018, toulouse, France.