SAT-Based Testing of Diagnosability and Predictability of Centralized and Distributed Discrete Event Systems

Hassan Ibrahim and Philippe Dague

LRI, Univ. Paris-Sud, CNRS, Univ. Paris-Saclay
Orsay, France
Email: firstname.lastname@lri.fr

Laurent Simon

LaBRI, Univ. Bordeaux, CNRS
Bordeaux, France
Email: lsimon@labri.fr

***Abstract*—In the general framework of safety analysis, diagnosability of a system, i.e., the guarantee to surely identify any fault in a finite delay after its occurrence, based on the available observations, is a key property to be verified at design stage. Diagnosability analysis of discrete event systems received a lot of attentions in the past twenty years, firstly in the centralized, then in the distributed case. In particular, a satisfiability-based approach was proposed in 2007 in the centralized case. We extend in this work this approach to cover also distributed discrete event systems, by handling both observable and unobservable synchronous communication events at the same time. Then, we adapt the method to analyze, in both centralized and distributed cases, fault predictability, a stronger property than diagnosability, which guarantees that any fault can be correctly predicted before its occurrence, based on observations. We provide experimental results for both diagnosability and predictability.**

***Keywords*–Discrete Event Systems; Distributed Systems; Diagnosability; Predictability; Satisfiability.**

I. Introduction

Nowadays, there is an increasing interest to ensure from the design stage of a system that partial observations given by the sensors will allow a precise diagnosis of potential faults that could occur in that system, once built. This will actually save high costs of adding new sensors for this task during the operating mode of the system. This raises the problem of diagnosability and of predictability which are essential properties to verify while designing the system model. Once this verification has been done (possibly thanks to a modification of the system model), both the system and its diagnoser or predictor (which can be automatically derived from diagnosability or predictability analysis) can be built with a guarantee of correctness and precision of the diagnosis, at least for those faults anticipated at design stage. Diagnosability is a property that determines the possibility to distinguish any possible behavior in the system with a given fault from any other behavior without this fault. A fault is diagnosable if it can be surely identified from the partial observation available in a finite delay after its occurrence. A system is diagnosable if every possible fault in it is diagnosable. Predictability is similarly an important system property, stronger than diagnosability, that determines at design stage whether a considered fault can be correctly predicted before its occurrence, based on available observations. If a fault is predictable, the fault management system can be designed to warn the operator, to halt the system or to take preventive measures.

The main difficulty in diagnosability and predictability checking is related to the states number explosion. Methods to cope with this problem and scale the studied system size in the case of discrete event systems (DES) resort to adopting a succinct representation of the system or a distributed modeling and to use powerful checking tool. For these reasons, we extend in this work to distributed discrete event systems (DDES) the succinct representation and the use of a satisfiability (SAT) solver introduced in [1] for centralized DES. Then, we adapt the SAT-based method to predictability analysis, in both centralized and distributed cases.

The paper is structured as follows. We first present related works in section II. In section III, we introduce the system transition models for DES and recall the traditional definition of diagnosability in those models and the state of the art of encoding it as a satisfiability problem in propositional logic. Then, in section IV, we present our first contribution, an extension of this SAT-based diagnosability analysis to DDES with observable and unobservable synchronous communication events in the same model, and give experimental results of this extension. Then, in section V, after having recalled the usual definition of predictability in DES, follows our second contribution, the encoding of this property as a satisfiability problem for both DES and DDES, and presentation of experimental results. Finally, in section VI, we conclude and outline our perspectives for future work.

II. Selection of Related Works

The first introduction to the notion of diagnosability was by [2], who gave its formal definition (see Def. 2 in section III) and studied it for labeled transition systems (LTS) by constructing a deterministic diagnoser to test it. However, this approach is exponential in the number of states of the system, which makes it impractical. In order to overcome this limitation, [3] introduced the *twin plant* approach, a structure built by synchronizing on their observable events two identical instances of a nondeterministic fault diagnoser. Then a so-called *critical path* is searched in this structure, i.e., a path with an observed cycle made up of ambiguous states, i.e., states that are pairs of original states, one reached by going through a fault and the other not. Fault diagnosability is thus equivalent to the absence of such a critical path. This approach turns the diagnosability problem in a search for a path with a cycle in a finite automaton, and this reduces its complexity to be polynomial of degree 4 in the number of states. The works by

[4] and [5] generalize simple faults modeled as distinguished events to supervision patterns, given as arbitrary suffix-closed rational languages of events.

The first work that addressed diagnosability analysis in DDES was [6], who introduced an incremental diagnosability test that avoids to build the twin plant for the whole system if not needed. Thus, one starts by building a local twin plant for the faulty component to test the existence of a local critical path. If such a path exists one builds the local twin checkers of the neighboring components (structure similar to local twin plant, except that there is no fault information in it) and one tries to solve the ambiguity resulting from the local critical path by exploiting the observable events in the neighboring components. This is done by synchronizing on their *communication events* the local twin plant with the local twin checker of one neighboring component. The process is repeated until the diagnosability is answered, so only in the worst case has the whole system to be visited. The work by [7] has optimized this construction by exploiting the different identifiers given to the communication events at the observation synchronization level (depending on which instance, left or right, they belong to) to assign them directly to the two behaviors studied. This helped in deleting the redundant information, then in abstracting the amount of information to be transferred later to next steps if the diagnosability was not answered. The generalization to supervision patterns in DDES was introduced by [8].

After the reduction of the diagnosability problem to a path finding problem by [3], it became transferable to a satisfiability problem as for planning problems [9]. This was done by [1] which formulated the diagnosability problem (in its twin plant version) into a SAT problem, assuming a centralized DES with simple fault events, modeled as a succinct labeled transition system (SLTS). We provide in subsection III-B a summary of this approach, on which our work is based. Our prior work [10] focused on using incremental SAT for diagnosability analysis in DDES.

Works on the predictability property for DES are fewer and more recent. A deterministic diagnoser approach was proposed by [11], with exponential complexity in the number of system states, and later a polynomial method by [12], that checks predictability directly on a twin plant. But the whole twin plant is built, which we avoid here by forcing the search after the fault occurrence in the correct sequence only (see subsection V-B). The generalization to supervision patterns was introduced by [13] and to DDES by [14] and [15].

III. SAT-Based Diagnosability Analysis of Centralized Systems

We recall the definitions of DES models we use, of the diagnosability property and of its SAT-based analysis.

A. Preliminaries

Traditionally, since the seminal work [2], LTS are used as a modeling formalism, where faults are simply modeled as particular unobservable events. Following [1] we will use an equivalent but more compact representation than LTS called SLTS, that are expressed in terms of propositional variables, allowing an easier translation to a SAT problem of the twin plant method proposed by [3] for checking diagnosability. The system states are represented by the valuations of a finite set A of Boolean state variables where valuation changes reflect the transitions between states according to the events. The set of all literals issued from A is $L = A \cup \{\neg a | a \in A\}$ and $\mathcal{L}$ is the full propositional language over A that consists of all formulas that can be formed from A and the connectives $\vee$, $\wedge$, $\neg$, $\rightarrow$ and $\leftrightarrow$. Each event is described by a set of pairs $\langle \phi, c \rangle$ which represent its possible ways of occurrence by indicating that the event can be associated with changes $c \in 2^L$ in states that satisfy the condition $\phi \in \mathcal{L}$.

Definition 1. A **succinct labeled transition system** (SLTS) is described by a tuple $T = \langle A, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0 \rangle$ where:

- A is a finite set of state variables,
- Σ_o is a finite set of observable correct events,
- Σ_u is a finite set of unobservable correct events,
- Σ_f is a finite set of unobservable faulty events,
- $\delta : \Sigma = \Sigma_o \cup \Sigma_u \cup \Sigma_f \rightarrow 2^{\mathcal{L} \times 2^L}$ assigns to each event a set of pairs $\langle \phi, c \rangle$,
- s_0 is the initial state (a valuation of A).

It is straightforward to show that any LTS with a set of states X can be represented as an SLTS with $\lceil log(|X|) \rceil$ Boolean variables and reciprocally that any SLTS can be mapped to an LTS (see Definition 2.4 in [1]).

The formal definition of diagnosability of a fault f in a centralized system modeled by an LTS or SLTS T was proposed by [2] as follows.

Definition 2. Diagnosability. A fault f is diagnosable in a system T if and only if (iff)

$$\exists k \in \mathbb{N}, \forall s^f \in L(T), \forall t \in L(T)/s^f, |t| \geq k \Rightarrow$$
$$\forall p \in L(T), (P(p) = P(s^f.t) \Rightarrow f \in p).$$

In this formula, $L(T)$ denotes the prefix-closed language of T whose words are called trajectories, s^f any trajectory ending by (a first occurrence of) the fault f, $L(T)/s$ the post-language of $L(T)$ after the trajectory s, i.e., $\{t \in \Sigma^* | s.t \in L(T)\}$ and P the projection of a trajectory on its observable events. The above definition states that for each trajectory s^f ending with fault f in T, for each t that is an extension of s^f in T with enough (depending only on f, not on its occurrences) events, every trajectory p in T that is equivalent to $s^f.t$ in terms of observation should contain in it f. As usual, it will be assumed that $L(T)$ is live (i.e., for any state, there is at least one transition issued from this state) and convergent (i.e., there is no cycle made up only of unobservable events).

A system T is said to be diagnosable iff any fault $f \in \Sigma_f$ is diagnosable in T, which is equivalent to each fault being separately diagnosable (i.e., the other faults being considered as unobservable correct events). Thus, to avoid exponential complexity in the number of faults during diagnosability analysis, only one fault's diagnosability is checked at a time, without loss of generality. It will thus be assumed in the following that there exists only one fault event f ($\Sigma_f = \{f\}$), without restriction on the number of its occurrences. Diagnosability checking has been proved in [3] to be polynomial in the number $|X|$ of states for LTS, so exponential in the number $|A|$ of state variables for SLTS (actually the problem is NLOGSPACE-complete for LTS and PSPACE-complete for SLTS [16]).

B. SLTS Diagnosability as Satisfiability

An immediate rephrasing of definition 2 shows that T is nondiagnosable iff it exists a pair of trajectories corresponding to cycles (and thus to infinite paths), a faulty one and a correct one, sharing the same observable events. This is equivalent to the existence of an ambiguous cycle in the product of T by itself, synchronized on observable events, which is called *twin plant* structure introduced in [3]. A cycle is ambiguous iff it is made up of pairs of states respectively reachable by a faulty path and a correct path. This nondiagnosability test was formulated in [1] as a satisfiability problem in propositional logic and we recall below this encoding, where superscripts $t \in \mathbb{N}$ refer to time points and (e_o^t) and $(\hat{e}_o^t)$ refer respectively to the faulty and correct events occurrences sequences of a pair of trajectories witnessing nondiagnosability. These two sequences share the same observable events represented by (e^t) and forming a cycle. The states are described by valuations of (a^t) and $(\hat{a}^t)$.

In order to represent the occurrence of the fault f and differently from the original encoding in [1], which does not exploit any relation between the fault occurrences at the different time steps, we added the variables f^t, whose truth value is $True$ iff f has occurred before t. This helped us to propagate the fault information automatically and guide the solver to search this specific information about the fault occurrence which is essential to decide the diagnosability test (it will be required also for our predictability encoding in SAT). Each time step increase corresponds to triggering at least one transition and so the extension by an event of at least one of the two trajectories. $T = \langle A, \Sigma_u, \Sigma_o, \Sigma_f, \delta, s_0 \rangle$ being an SLTS, the propositional variables required for the encoding are:

- a^t and $\hat{a}^t$ for all $a \in A$ and $0 \leq t \leq n$,
- e_o^t for all $e \in \Sigma_o \cup \Sigma_u \cup \Sigma_f$, $o \in \delta(e)$ and $0 \leq t \leq n-1$,
- $\hat{e}_o^t$ for all $e \in \Sigma_o \cup \Sigma_u$, $o \in \delta(e)$ and $0 \leq t \leq n-1$,
- e^t for all $e \in \Sigma_o$ and $0 \leq t \leq n-1$,
- f^t for all $0 \leq t \leq n$.

The following formulas express the constraints that must be applied at each t or between t and $t+1$.

1) The event occurrence e_o^t must be possible in the current state:
$$e_o^t \rightarrow \phi^t \quad \text{for } o = \langle \phi, c \rangle \in \delta(e) \tag{1}$$
and its effects must hold at the next time step:
$$e_o^t \rightarrow \bigwedge_{l \in c} l^{t+1} \quad \text{for } o = \langle \phi, c \rangle \in \delta(e) \tag{2}$$
We have the same formulas with $\hat{e}_o^t$.

2) The present value ($True$ or $False$) of a state variable changes to a new value ($False$ or $True$, respectively) only if there is a reason for this change, i.e., because of an event that has the new value in its effects (so, change without reason is prohibited). Here is the change from $True$ to $False$ (the change from $False$ to $True$ is defined similarly by interchanging a and $\neg a$):
$$(a^t \wedge \neg a^{t+1}) \rightarrow (e^t_{{i_1}_{o_{j_1}}} \vee \cdots \vee e^t_{{i_k}_{o_{j_k}}}) \tag{3}$$
where the $o_{j_l} = \langle \phi_{j_l}, c_{j_l} \rangle \in \delta(e_{i_l})$ are all the occurrences of events e_{i_l} with $\neg a \in c_{j_i}$.
We have the same formulas with $\hat{a}^t$ and $\hat{e}^t_{{i_l}_{o_{j_l}}}$.

3) At most one occurrence of a given event can occur at a time and the occurrences of two different events cannot be simultaneous if they interfere (i.e., if they have two contradicting effects or if the precondition of one contradicts the effect of the other):
$$\neg(e_o^t \wedge e_{o'}^t) \quad \forall e \in \Sigma, \forall \{o, o'\} \subseteq \delta(e), o \neq o' \tag{4}$$
$$\neg(e_o^t \wedge e'^t_{o'}) \quad \forall \{e, e'\} \subseteq \Sigma, e \neq e', \forall o \in \delta(e),$$
$$\forall o' \in \delta(e') \text{ such that } o \text{ and } o' \text{ interfere} \tag{5}$$
We have the same formulas with $\hat{e}_o^t$.

4) The information about f occurrence is propagated by expressing that f has occurred before $t+1$ ($t \leq n-1$) iff it has occurred either before t or between t and $t+1$.
$$f^{t+1} \leftrightarrow f^t \ \vee \bigvee_{e \in \Sigma_f, o \in \delta(e)} e_o^t \tag{6}$$

5) The formulas that connect the two events sequences require that observable events take place in both sequences whenever they take place (use of e^t):
$$\bigvee_{o \in \delta(e)} e_o^t \leftrightarrow e^t \text{ and } \bigvee_{o \in \delta(e)} \hat{e}_o^t \leftrightarrow e^t \quad \forall e \in \Sigma_o \tag{7}$$

6) To avoid trivial cycles (silent loops with no state change at some step) we require that at every time point at least one event takes place:
$$\bigvee_{e \in \Sigma_o} e^t \quad \vee \bigvee_{e \in \Sigma_u \cup \Sigma_f, o \in \delta(e)} e_o^t \quad \vee \bigvee_{e \in \Sigma_u, o \in \delta(e)} \hat{e}_o^t \tag{8}$$

The conjunction of all the above formulas for a given t is denoted by $\mathcal{T}(t, t+1)$.

A formula for the initial state s_0 is:
$$\mathcal{I}_0 = \neg f^0 \ \wedge \bigwedge_{a \in A, s_0(a)=1} (a^0 \wedge \hat{a}^0) \ \wedge \bigwedge_{a \in A, s_0(a)=0} (\neg a^0 \wedge \neg \hat{a}^0) \tag{9}$$

At last, the following formula can be defined to encode the fact that a pair of executions is found with the same observable events and no fault in one execution but one fault in the other (first line), which are infinite (in the form of a cycle, necessarily non trivial by (8)) at step n (second line), witnessing non diagnosability:
$$\Phi_n^T = \mathcal{I}_0 \wedge \mathcal{T}(0,1) \wedge \cdots \wedge \mathcal{T}(n-1, n) \wedge f^n$$
$$\wedge \bigvee_{m=0}^{n-1} (\bigwedge_{a \in A} ((a^n \leftrightarrow a^m) \wedge (\hat{a}^n \leftrightarrow \hat{a}^m)))$$

From this encoding in propositional logic, follows the result (theorem 3.2 of [1]) that an SLTS T is not diagnosable iff $\exists n \geq 1, \Phi_n^T$ is satisfiable. It is also equivalent to $\Phi^T_{2^{2|A|}}$ being satisfiable, as the twin plant states number is an obvious upper bound for n, but often impractically high (see in the same reference some ways to deal with this problem).

IV. SAT-Based Diagnosability Analysis of Distributed Systems

We extend from centralized to distributed systems the satisfiability framework above for testing diagnosability and we provide some experimental results.

A. DDES Modeling

In order to model DDES with SLTS, we need to extend these ones by adding communication events *to each component*. So we introduce the following definition for a distributed SLTS with k different components (sites):

Definition 3. A **distributed succinct labeled transition system** (DSLTS) with k components is described by a tuple $T = \langle A, \Sigma_o, \Sigma_u, \Sigma_f, \Sigma_c, \delta, s_0 \rangle$ where (subscript i refers to component i):

- A is a union of disjoint finite sets $(A_i)_{1 \leq i \leq k}$ of component own state variables, $A = \cup_{i=1}^{k} A_i$,
- Σ_o is a union of disjoint finite sets of component own observable correct events, $\Sigma_o = \cup_{i=1}^{k} \Sigma_{oi}$,
- Σ_u is a union of disjoint finite sets of component own unobservable correct events, $\Sigma_u = \cup_{i=1}^{k} \Sigma_{ui}$,
- Σ_f is a union of disjoint finite sets of component own unobservable faulty events, $\Sigma_f = \cup_{i=1}^{k} \Sigma_{f\,i}$,
- Σ_c is a union of finite sets of (observable or unobservable) correct communication events, $\Sigma_c = \cup_{i=1}^{k} \Sigma_{ci}$, which are the only events shared by at least two different components (i.e., $\forall i, \forall c \in \Sigma_{ci}, \exists j \neq i, c \in \Sigma_{cj}$),
- $\delta = (\delta_i)$, where $\delta_i : \Sigma_i = \Sigma_{oi} \cup \Sigma_{ui} \cup \Sigma_{f\,i} \cup \Sigma_{ci} \rightarrow 2^{\mathcal{L}_i \times 2^{L_i}}$, assigns to each event a set of pairs $\langle \phi, c \rangle$ in the propositional language of the component where it occurs (so, for communication events, *in each component separately* where they occur),
- $s_0 = (s_{0i})$ is the initial state (a valuation of each A_i).

Synchronous communication is assumed. More precisely, a transition by a communication event c may occur in a component iff a simultaneous transition by c occurs in all the other components where c appears (has at least one occurrence). The global model of the system is thus the product of the models of the components, synchronized on communication events. Notice that we allow in whole generality communication events to be, partially or totally, unobservable (which is not allowed up to now in any model, to the best of our knowledge), so one has in general to wait further observations to know that some communication event occurred between two or more components. On the other side, assuming these communications to be faultless is not actually a limitation. If a communication process or protocol may be faulty, it has to be modeled as a proper component with its own correct and faulty behaviors. In this sense, communications between components are just a modeling concept, not subject to diagnosis. It will be also assumed that the observable information is global, i.e. centralized, allowing to keep definition 2 (as, when observable information is only local to each component, distributed diagnosability checking becomes undecidable [17]).

B. DSLTS Diagnosability as Satisfiability

Let T be a DSLTS made up of k components denoted by indexes i, $1 \leq i \leq k$. In order to express the diagnosability analysis of T as a satisfiability problem, we have to extend the formulas of the centralized case to deal with communication events between components. Let $\Sigma_c = \Sigma_{co} \cup \Sigma_{cu}$ be the communication events, with $\Sigma_{co} = \cup_{i=1}^{k} \Sigma_{coi}$ the observable ones and $\Sigma_{cu} = \cup_{i=1}^{k} \Sigma_{cui}$ the unobservable ones. The idea is to treat each communication event as any other event in each of its owners and, as it has been done with events e^t for $e \in \Sigma_o$ for synchronizing observable events occurrences in the two executions, to introduce in the same way a global reference variable for each communication event at each time step, in charge of synchronizing any communication event occurrence in any of its owners with occurrences of it in all its other owners. We use one such reference variable for each trajectory, e^t and $\hat{e}^t$, for unobservable events $e \in \Sigma_{cu}$, and only one for both trajectories, e^t, for observable events $e \in \Sigma_{co}$ as it will also in addition play the role of synchronizing observable events between trajectories exactly as the e^t for $e \in \Sigma_o$. So, we add to the previous propositional variables the new following ones:

- e_o^t, $\hat{e}_o^t$ for all $e \in \Sigma_c$, $o \in \delta(e) = \cup_i \delta_i(e)$ and $0 \leq t \leq n-1$,
- e^t for all $e \in \Sigma_c$, $\hat{e}^t$ for all $e \in \Sigma_{cu}$ and $0 \leq t \leq n-1$.

Formulas in $\mathcal{T}(t, t+1)$ are extended as follows.

1) Formulas (1), (2), (3) and (5) extend unchanged to e_o^t and $\hat{e}_o^t$ $\forall e \in \Sigma_c$.
2) Formulas (4) extend to prevent two simultaneous occurrences of a given communication event in the same owner component, i.e. apply $\forall e \in \Sigma_c, \forall i, \forall \{o_i, o_i'\} \subseteq \delta_i(e), o_i \neq o_i'$ (the same with $\hat{e}$)
3) The new following formulas express the communication process itself, i.e. the synchronization of the occurrences of any communication event e in all its owners components ($S(e)$ being the set of indexes of the owners components of e) and extend also formulas (7) to observable communication events:

$$\bigvee_{o_i \in \delta_i(e)} e_{o_i}^t \leftrightarrow e^t \text{ and } \bigvee_{o_i \in \delta_i(e)} \hat{e}_{o_i}^t \leftrightarrow \hat{e}^t \ \forall e \in \Sigma_{cu} \ \forall i \in S(e)$$

$$\bigvee_{o_i \in \delta_i(e)} e_{o_i}^t \leftrightarrow e^t \text{ and } \bigvee_{o_i \in \delta_i(e)} \hat{e}_{o_i}^t \leftrightarrow e^t \ \forall e \in \Sigma_{co} \ \forall i \in S(e)$$

4) Finally, the clause (8) is adapted to take into account both observable and unobservable communication events:

$$\bigvee_{e \in \Sigma_o \cup \Sigma_c} e^t \quad \vee \quad \bigvee_{e \in \Sigma_{cu}} \hat{e}^t \quad \vee \quad \bigvee_{e \in \Sigma_u \cup \Sigma_f, o \in \delta(e)} e_o^t \quad \vee \quad \bigvee_{e \in \Sigma_u, o \in \delta(e)} \hat{e}_o^t$$

We have thus the result that a DSLTS T is not diagnosable iff $\exists n \geq 1, \Phi_n^T$ is satisfiable (proof analog to that in the centralized case). It is also equivalent to $\Phi^T_{2^{2\sum_{i=1}^{k} |A_i|}}$ being satisfiable.

C. Implementation and Experimental Testing

We have implemented the above extension in Java, our experiments were run on 64-bit Windows 7 machine with an Intel(R) Xeon(R) CPU @2.80GHz processor and 8 GB of RAM. We used the well designed API of the SAT solver Sat4j [18] as it fitted well our clause generator written in Java. We have tested our tool on small examples with several communication events with multiple occurrences, with global communication (all components share the same event) or partial communication (only some components share the same event), as in Fig. 1, adapted from the running example in [6], which is made up of three communicating components. The results are in Table I, where the columns show the system and the fault considered (4 cases separated by horizontal lines), the steps number n, the answer of the SAT solver, the numbers of variables and of clauses and the runtime in ms.

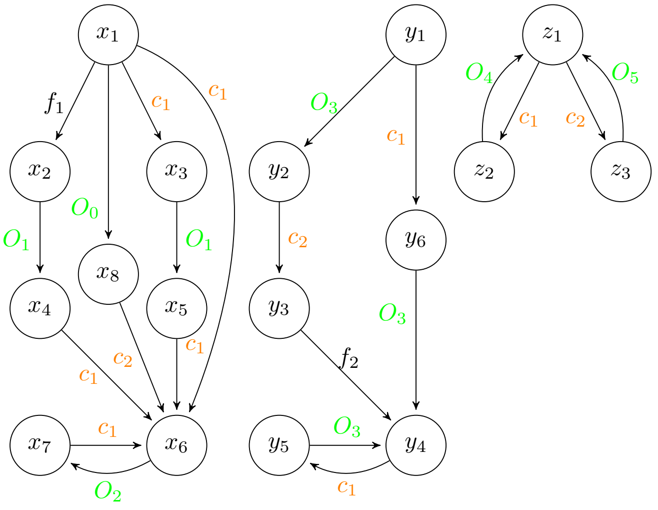

Figure 1. A DDES made up of 3 components C1, C2 and C3 from left to right. $ci,_{1\leq i\leq 2}$ are unobservable communication events, $Oi,_{0\leq i\leq 5}$ are observable events and $fi,_{1\leq i\leq 2}$ are faulty events.

TABLE I. DIAGNOSABILITY RESULTS ON THE EXAMPLE OF FIG 1.

System	Fault	|Steps|	SAT?	|Variables|	|Clauses|	Time(ms)
$C2$	$f2$	4	No	112	561	6
$C2$	$f2$	5	No	138	699	11
$C2$	$f2$	6	Yes	164	837	15
$C1, C2$	$f2$	6	No	356	356	25
$C1, C2$	$f2$	32	No	1838	12751	94
$C1, C2$	$f2$	64	No	3662	25487	225
$C1, C2$	$f2$	128	No	7310	50959	112
$C1, C2$	$f2$	256	No	14606	101903	180
$C1, C2$	$f2$	512	No	29198	203791	1855
$C1, C2$	$f2$	1024	No	58382	407567	784
$C1, C2$	$f2$	4096	No	233486	1630223	23453
$C2, C3$	$f2$	6	No	252	1237	15
$C2, C3$	$f2$	32	No	1292	6541	46
$C2, C3$	$f2$	64	No	2572	13069	71
$C2, C3$	$f2$	128	No	5132	26125	61
$C2, C3$	$f2$	256	No	10252	52237	216
$C2, C3$	$f2$	512	No	20492	104461	143
$C2, C3$	$f2$	1024	No	40972	208909	381
$C1, C2, C3$	$f1$	8	No	586	3723	40
$C1, C2, C3$	$f1$	9	Yes	657	4186	45
$C1, 10 \times C2, 10 \times C3$	$f1$	9	Yes	3862	22907	342
$C1, 20 \times C2, 20 \times C3$	$f1$	9	Yes	7112	42087	592
$C1, 50 \times C2, 50 \times C3$	$f1$	9	Yes	16862	99627	3141
$C1, 100 \times C2, 100 \times C3$	$f1$	9	Yes	33372	196723	26930

Which means that $f2$ is not diagnosable in $C2$ alone while it becomes diagnosable when synchronizing $C2$ with either $C1$ or $C3$. For proving these two last results, we have increased the steps number, verifying that the answer remained UNSAT, until reaching the theoretical upper bound $2^{2\sum_i |A_i|}$ (equal to $2^{2(3+2)} = 1024$ for $\{C2, C3\}$ and to $2^{2(3+3)} = 4096$ for $\{C1, C2\}$). While $f1$ is not diagnosable even after synchronizing all three components together. These 4 tests are mentioned as a proof of concept. But actually, numbers of variables and clauses are small in comparison to what SAT solvers can handle (up to hundred thousands propositional variables and millions of clauses). This is why we extended the last case (non-diagnosability of $f1$) to bigger systems obtained by duplicating (10, 20, 50 and 100 times) components $C2$ and $C3$, keeping unchanged their communication events and renaming their proper local events. This shows the efficiency of the method (less than 27s for 201 components). Notice that here the steps number remains unchanged as occurrences of non-interfering events are processed simultaneously in the same step, thanks to the succinct encoding in this representation. The number of states in the last tested system is very large, which proves the efficiency of this approach in detecting the non-diagnosability of a system if the length of a potential critical path stays short. The case where diagnosability analysis requires checking very long potential critical paths is still impractical and needs a more abstract induction-proof approach.

V. PREDICTABILITY AS SATISFIABILITY

We recall the definition of the predictability property, adapt the framework above to define SAT-based predictability analysis for both centralized and distributed systems and provide experimental results.

A. *Definition*

The formal definition of predictability of a fault f in a centralized system modeled by an LTS or SLTS T was proposed by [11] as follows.

Definition 4. Predictability. A fault f is predictable in a system T iff

$$\exists k \in \mathbb{N}, \forall s^f \in L(T), \exists \eta \in \overline{s^f}, \forall p \in L(T), \forall p' \in L(T)/p$$
$$(P(p) = P(\eta) \wedge f \notin p \wedge |p'| \geq k \Rightarrow f \in p')$$

The above definition, where $\overline{t}$ denotes the set of strict prefixes of t, states that a fault f is predictable iff for any trajectory s^f ending with a first occurrence of f, there exists at least one strict prefix of s^f, denoted by η (thus η does not contain f), such that for every correct trajectory p with the same observations as η, all the long enough (depending only on f) continuations of p should contain f. In other words, the non-predictability of f is equivalent to the existence of a finite faulty sequence that ends with a first occurrence of f and of an infinite (i.e. corresponding to a cycle) correct sequence that is synchronized with the faulty sequence on observable events before the occurrence of f. Predictability is thus stronger than diagnosability (if f is predictable, then f is diagnosable).

B. *SLTS Predictability as Satisfiability*

Unlike diagnosability, predictability checking process has two different phases, before and after the fault occurrence in the faulty sequence: the synchronization on observable events between the two sequences is required only up to this fault occurrence and, after it, only the correct sequence has to be extended and searched for the presence of a cycle in it. The new or modified formulas expressing the constraints to be applied at each time step t are as follows.

1) The synchronization of observable events between the two sequences holds only up to the fault occurrence, i.e. (7) is replaced by:

$$\begin{aligned} f^t \;\vee\; (\bigvee_{o\in\delta(e)} e^t_o \leftrightarrow e^t) \quad \forall e \in \Sigma_o \\ f^t \;\vee\; (\bigvee_{o\in\delta(e)} \hat{e}^t_o \leftrightarrow e^t) \quad \forall e \in \Sigma_o \end{aligned} \tag{10}$$

2) The formula (8), requiring that at every time point at least one event takes place in either one or the other sequence, remains valid up to the fault occurrence; after it, we require that at least one event takes place in the correct sequence:

$$f^t \;\vee \bigvee_{e\in\Sigma_o} e^t \;\;\vee \bigvee_{e\in\Sigma_u\cup\Sigma_f, o\in\delta(e)} e_o^t \;\;\vee \bigvee_{e\in\Sigma_u, o\in\delta(e)} \hat{e}_o^t$$

$$\neg f^t \;\vee \bigvee_{e\in\Sigma_o\cup\Sigma_u, o\in\delta(e)} \hat{e}_o^t \qquad (11)$$

The conjunction of the formulas (1-6), (10) and (11) for a given t is denoted by $\mathcal{S}(t, t+1)$.

The formula (9) for the initial state s_0 is unchanged.

Finally, the formula to encode the non predictability property is obtained as Φ_n^T, where the presence of a cycle at step n is required only in the correct sequence:

$$\Psi_n^T = \mathcal{I}_0 \wedge \mathcal{S}(0,1) \wedge \cdots \wedge \mathcal{S}(n-1,n) \wedge f^n$$
$$\wedge \bigvee_{m=0}^{n-1} \big(\bigwedge_{a\in A} (\hat{a}^n \leftrightarrow \hat{a}^m)\big)$$

It follows that an SLTS T is not predictable iff $\exists n \geq 1, \Psi_n^T$ is satisfiable, which is also equivalent to $\Psi_{2^{2|A|}}^T$ being satisfiable (proof analog to that for diagnosability).

C. DSLTS Predictability as Satisfiability

Let T be now a DSLTS. The extension of predictability analysis to distributed systems adapts what we presented for diagnosability analysis. As the synchronization of observable events holds only before the fault occurrence, we will decouple it from the synchronization of communication events. So, the only change concerning the variables is that we use now one reference variable for each sequence for observable communication events, as for unobservable ones, i.e. we have:

- e^t, $\hat{e}^t$ for all $e \in \Sigma_c$ and $0 \leq t \leq n-1$.

Formulas in $\mathcal{S}(t, t+1)$ are extended as those in $\mathcal{T}(t, t+1)$ were extended, except the following.

1) The synchronization of the occurrences of any communication event e in all its owner components in $S(e)$ is expressed in each sequence and in the same way for both observable and unobservable events:

$$\bigvee_{o_i\in\delta_i(e)} e_{o_i}^t \leftrightarrow e^t \text{ and } \bigvee_{o_i\in\delta_i(e)} \hat{e}_{o_i}^t \leftrightarrow \hat{e}^t \;\forall e\in\Sigma_c \;\forall i\in S(e)$$

while the synchronization of the occurrences of any observable event in the two sequences before the fault occurrence, expressed in the centralized case by formulas (10), is extended to any observable communication event:

$$f^t \;\vee\; (e^t \leftrightarrow \hat{e}^t) \quad \forall e\in\Sigma_{co}$$

2) The clauses (11) are extended to take into account communication events:

$$f^t \;\vee \bigvee_{e\in\Sigma_o\cup\Sigma_c} e^t \;\vee \bigvee_{e\in\Sigma_{cu}} \hat{e}^t \;\vee \bigvee_{e\in\Sigma_u\cup\Sigma_f, o\in\delta(e)} e_o^t \;\;\vee \bigvee_{e\in\Sigma_u, o\in\delta(e)} \hat{e}_o^t$$

$$\neg f^t \;\vee \bigvee_{e\in\Sigma_c} \hat{e}^t \;\vee \bigvee_{e\in\Sigma_o\cup\Sigma_u, o\in\delta(e)} \hat{e}_o^t$$

We have thus the result that a DSLTS T is not predictable iff $\exists n \geq 1, \Psi_n^T$ is satisfiable, which is also equivalent to $\Psi_{2^{2\sum_{i=1}^k |A_i|}}^T$ being satisfiable (proof analog to that for diagnosability).

D. Experimental Results

We used the same example (Fig. 1) as for diagnosability and studied the predictability of the faulty events $f1$ and $f2$. Table II shows the results obtained. It is found that $f2$ is not predictable in $C2$ alone, which was expected as it is not diagnosable in $C2$. We saw that it became diagnosable in the system composed of $C1$ and $C2$ and we find that it is actually even predictable in this system, by obtaining the UNSAT answer up to the theoretical upper bound 4096. On the contrary, although we saw it became also diagnosable in the system composed of $C2$ and $C3$, we find that it remains not predictable in this system. And here again, we extend this test to bigger systems by duplicating component $C3$, with the same steps number and very good efficiency. Concerning the fault $f1$, it is found not predictable in the whole system made up of the three components, which was expected as it has been shown not diagnosable in this system.

TABLE II. PREDICTABILITY RESULTS ON THE EXAMPLE OF FIG 1.

System	Fault	|Steps|	SAT?	|Variables|	|Clauses|	Time (ms)
$C2$	$f2$	3	No	92	414	7
$C2$	$f2$	4	Yes	120	549	12
$C1, C2$	$f2$	1024	No	66574	404495	10109
$C1, C2$	$f2$	4096	No	266254	1617935	91299
$C2, C3$	$f2$	4	No	196	817	14
$C2, C3$	$f2$	5	No	242	1018	21
$C2, C3$	$f2$	6	Yes	288	1219	27
$C1, C2, C3$	$f1$	3	No	267	1399	29
$C1, C2, C3$	$f1$	4	Yes	350	1859	40
$C2, 10 \times C3$	$f2$	6	Yes	1408	5219	24
$C2, 20 \times C3$	$f2$	6	Yes	2528	9219	50
$C2, 50 \times C3$	$f2$	6	Yes	5888	21219	125
$C2, 100 \times C3$	$f2$	6	Yes	11488	41219	277

VI. Conclusion and Future Works

By extending the state of the art work for centralized DES [1], we have expressed diagnosability analysis of DDES as a satisfiability problem by building a propositional formula whose satisfiability, witnessing non-diagnosability, can be checked by SAT solvers. We allow both observable and unobservable synchronous communication events in our model. We have then applied the same method to express predictability analysis as a SAT problem, both for centralized DES and for DDES. In each case, we have provided experimental results.

In order to conduct more experiments to check precisely the scalability of the method and to compare it with other approaches referenced above (for which no software is available and in general no experimental results are given), we have implemented classical twin plant based algorithms and achieve implementing an automatic benchmarks generator, tuned by several parameters and whose diagnosability and predictability will be known by construction. We have also designed and are implementing the extension of this work from simple faulty events to supervision patterns. All our programs will be made available as open source. We also aim at investigating relations between our work and the problem of opacity of discrete event systems [19], in order to treat this problem with SAT-based methods. Finally, relationships between satisfiability and bounded or unbounded model checking methods to encode and analyze fault diagnosability and predictability will be studied. In particular, SAT-based model checking [20] allows incremental solving, which significantly improves both the capacity and the speed of solving. Research of invariants by full-proof methods like temporal induction should avoid unrolling to a theoretical bound, as it is the case here when the system is not diagnosable.

References

[1] J. Rintanen and A. Grastien, "Diagnosability testing with satisfiability algorithms." *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pp. 532–537, 2007.

[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, 1995.

[3] S. Jiang, Z. 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, 2001.

[4] T. Jéron, H. Marchand, S. Pinchinat, and M.-O. Cordier, "Supervision Patterns in Discrete Event Systems Diagnosis." *Proceedings of the 8th International Workshop on Discrete Event Systems (WODES'06)*, pp. 262–268, 2006.

[5] S. Genc and S. Lafortune, "Diagnosis of patterns in partially-observed discrete-event systems." *Proceedings of the 45th IEEE Conference on Decision and Control (CDC'06)*, pp. 422–427, 2006.

[6] Y. Pencolé, "Diagnosability Analysis of Distributed Discrete Event Systems." *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI'04)*, pp. 43–47, 2004.

[7] L. Ye and P. Dague, "An optimized algorithm for diagnosability of component-based systems." *Proceedings of the 10th International Workshop on Discrete Event Systems (WODES'10)*, pp. 143–148, 2010.

[8] Y. Yan, L. Ye, and P. Dague, "Diagnosability for patterns in distributed discrete event systems." *Proceedings of the 21st International Workshop on Principles of Diagnosis (DX'10)*, pp. 345–352, 2010.

[9] H. Kautz and B. Selman, "Planning as Satisfiability." *Proceedings of the 10th European Conference on Artificial Intelligence (ECAI'92)*, pp. 359–363, 1992.

[10] H. Ibrahim, P. Dague, and L. Simon, "Using Incremental SAT for Testing Diagnosability of Distributed DES." *Proceedings of the 26th International Workshop on Principles of Diagnosis (DX'15)*, pp. 51–58, 2015.

[11] S. Genc and S. Lafortune, "Predictability in discrete-event systems under partial observation." *Proceedings of the 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS'06)*, pp. 1461–1466, 2006.

[12] S. Genc and S. Lafortune, "Predictability of Event Occurrences in Partially-observed Discrete-event Systems." *Automatica*, vol. 45, no. 2, pp. 301–311, 2009.

[13] T. Jéron, H. Marchand, S. Genc, and S. Lafortune, "Predictability of Sequence Patterns in Discrete Event Systems." *Proceedings of the 17th World Congress*, pp. 537–453, 2008.

[14] L. Ye, P. Dague, and F. Nouioua, "Predictability Analysis of Distributed Discrete Event Systems." *Proceedings of the 52nd IEEE Conference on Decision and Control (CDC'13)*, pp. 5009–5015, 2013.

[15] ——, " A predictability algorithm for distributed discrete event systems." *Proceedings of the 17th International Conference on Formal Engineering Methods (ICFEM'15)*, pp. 201–216, 2015.

[16] J. Rintanen, "Diagnosers and diagnosability of succinct transition systems." *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pp. 538–544, 2007.

[17] L. Ye and P. Dague, "Undecidable Case and Decidable Case of Joint Diagnosability in Distributed Discrete Event Systems." *International Journal On Advances in Systems and Measurements*, vol. 6, no. 3 and 4, pp. 287–299, 2013.

[18] D. Le Berre and A. Parrain, "The Sat4j library, release 2.2." *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 7, pp. 59–64, 2010.

[19] F. Lin, "Opacity of discrete event systems and its applications." *Automatica*, vol. 47, no. 3, pp. 496–503, 2011.

[20] E. Clarke, A. Biere, R. Raimi, and Y. Zhu, "Bounded model checking using satisfiability solving." *Formal Methods in System Design*, vol. 19, no. 1, pp. 7–34, 2001.