A Short Overview on Diagnosability of Patterns in Timed Petri Net

Éric Lubat and Silvano Dal Zilio

LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
eric.lubat@laas.fr

Diagnosability is a basic property of Discrete Event Systems (DES) that relates to the "observability" of concealed events. Basically, it means that every failure (a distinct instance of unobservable event) can be eventually detected after a finite number of observations. In this work, we are interested by the diagnosability of systems modelled using labelled Time Petri nets (TPN), an extension of Petri nets in which we can associate timing constraints to transitions [7]. This means that we take into account the date at which events are observed and that we want to detect failures in a bounded time. We are also interested by the detection of patterns of events (sequence of observable events that are part of some given regular language) instead of the occurrence of a single fault.

The problem of fault diagnosis was introduced by Sampath et al [8] and is well studied in the context of discrete event systems [4,11]. The problem has also been studied on timed models, see for instance the work of Tripakis [9] with Time Automata. In these works, diagnosability is often reduced to properties on the *trace language* of systems and their composition. These problems are much more difficult when we need to take timing constraints into account. For instance, one of the main problem when considering Time Petri nets is that it is not possible to build the synchronous composition of two transitions when they have non-trivial time constraints. Therefore it is not possible to "syntactically" define the intersection of the trace languages of two TPN.

We tackle the diagnosability problem using model-checking techniques, which means that we reduce diagnosability to the problem of checking the validity of a temporal logic formula on a finite-state model. This approach relies on a new class of TPN, called Product TPN (PTPN), introduced in [5]. The idea is to enrich TPN with a new synchronization operator ($\times$) that can be used to compute the (language) intersection of two or more TPN. We show in [5] that it is possible to extend the classical State Class Graph construction to this model.

This work makes two main contributions. First, we describe how to use PTPN to decide the diagnosability of a system. Our approach extends the classical techniques based on the twin-plant construction, which relies on the composition of two copies of (the model of) a system. Next, we show that we can also define a notion of *pattern diagnosability* as a straightforward extension of the single fault problem. We conclude with a short discussion on future work.

Diagnosability of Single Faults. There are two main problems when checking timed extensions of Petri nets. First, the state space of a net is typically infinite when we use a dense time model, that is when time delays can be arbitrarily small. Therefore we need to work on an abstraction of the transition system. Second, there is no natural way to define the (structural) composition of two transitions that have non-trivial time constraints.

A solution to the first problem was proposed by Berthomieu and Menasche in [2], where the authors define a state space abstraction based on *state classes*. A state class captures a convex set of constraints on the time at which transitions can fire. This approach is used in several model-checking tools, such as Tina [3] for instance, and makes it possible to check LTL properties on bounded TPN. To solve the second problem, we propose to use Product PTPN, a new model where we can define "groups of transitions", in a net, that should fire concurrently, meaning together, and at the same date. The main idea behind this product operator is to mimic the behaviour of a synchronous product between transitions. Given two TPN, N and N', we can define a PTPN $N \times N'$ that has exactly the (observable) behaviours that are in both N and N'.

Armed with PTPN, we can easily define the *twin-plant* construction of a net N (see e.g. [10]), as the composition of two copies of the net, say $N.1 \times N.2$, on the set of their observable labels. In the following, we consider that failures are transitions on a common unobservable label, say f. As a consequence, a system is diagnosable when we cannot find a pair of execution traces such that: (1) they are *compatible* (observable events occur in the same orders and at the same date); and (2) only one of them *exhibits a failure*. This would entail that we have an execution, with the occurrence of a failure, that is indistinguishable from a nominal one. This case (often referred to as the existence of a *critical pair*) can be directly translated into traces in the twin-plant. In particular, system N is diagnosable if all the (maximal) executions of the twin-plant $N.1 \times N.2$ satisfies the LTL formula $(\Diamond f.1) \Leftrightarrow (\Diamond f.2)$. This support the idea that diagnosability can be reduced to checking some temporal properties on the twin-plant model, an idea that was already exploited in [9,10].

In this context, we use the general assumption that systems are *ultimately observable*, meaning that they do not block and that, on every execution, we will always eventually find an observable event (after a bounded number of transitions). Hence a system is diagnosable when pairs of compatible traces have the same failures, or if one of them block, meaning that it is not possible to extend them with some compatible event. We can further simplify this property by using the inherent symmetry of the problem.

Theorem 1. *a TPN N is diagnosable if and only if all the maximal executions of the PTPN $N.1 \times N.2$ satisfy the LTL formula $\Box(f.1 \Rightarrow \Diamond(dead \vee f.2))$.*

Theorem 1 provides a constructive method for checking the diagnosability of a TPN, as long as the net is bounded. We have implemented this method inside a tool called TWINA.

Diagnosability of Patterns. It is possible to extend this method to the observability of "patterns of events" [6]. We define a *pattern* as a labelled Petri net, P, representing the set of behaviours (sequences of labels) that we want to detect. We also identify a distinguished place in P, say *found*, that corresponds to the detection of the pattern. We say that the pattern is detected in the net N when the place *found* is marked in $N \times P$. Drawing a parallel with the case of single failure, we say that a TPN N is *diagnosable for the pattern* P if we cannot find a pair of compatible (and maximal) executions in $N \times P$ such that the pattern is detected in one and not the other. Again, we propose to reduce this diagnosability problem to a model-checking problem on the twin-plant, this time on the product of the system and the pattern.

Theorem 2. *Assuming some well-formedness conditions on the pattern P, TPN N is diagnosable for pattern P iff all maximal executions of the PTPN $(N.1 \times P.1) \times (N.2 \times P.2)$ satisfy the LTL formula $\Box(\mathit{found}.1 \Rightarrow \Diamond(\mathit{dead} \vee \mathit{found}.2))$.*

This time, we need to impose some extra restrictions. Intuitively, patterns are very much like observers in observer-based verification methods. Like with observers, we want to make sure that P does not interfere with the system it interacts with. For example, it should not prevents the execution of the system. To this end, we impose three "well-formedness conditions", which are sufficient conditions for Theorem 2 to hold: (1) the pattern must to be total (it should never block or delay a transition on one of its label, so there is always at least one transition for any given label); (2) the pattern must be deterministic (the same observations should lead to the same states, so there is at most one possible transition for any given label); and (3) the net P is bounded and place *found* is a *trap* (once it is marked, it stays marked).

Future Work. We give a very broad overview of a constructive method to decide the diagnosability of single failures and patterns of failures in TPN. Our presentation emphasize the parallel structure between these two properties. In the future, we hope to benefit from this interconnection in order to define more elaborate observability properties. For instance we would like to study the link with the notion of opacity and see if we can come up with an equivalent formulation with patterns.

References

1. Arnold, A.: Nivat's processes and their synchronization. Theoretical Computer Science **281**(1), 31–36 (2002)
2. Berthomieu, B., Menasche, M.: An enumerative approach for analyzing time Petri nets. In: Proceedings IFIP (1983)
3. Berthomieu, B., Ribet, P.O., Vernadat, F.: The tool TINA–construction of abstract state spaces for Petri nets and time Petri nets. International Journal of Production Research **42**(14), 2741–2756 (2004)
4. Cassandras, C.G., Lafortune, S.: Introduction to Discrete Event Systems. Springer-Verlag (2009)
5. Dal Zilio, S.: TWINA: A realtime model-checker for analyzing Twin-TPN. https://projects.laas.fr/twina/ (2019)

6. Gougam, H.E., Pencolé, Y., Subias, A.: Diagnosability analysis of patterns on bounded labeled prioritized Petri nets. Discrete Event Dynamic Systems **27**(1), 143–180 (2017)
7. Merlin, P.M.: A study of the recoverability of computing systems. Ph.D. thesis, Department of Information and Computer Science, University of California (1974)
8. Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D.: Diagnosability of discrete-event systems. IEEE Transactions on automatic control **40**(9), 1555–1575 (1995)
9. Tripakis, S.: Fault diagnosis for Timed Automata. In: 7th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT). pp. 205–224 (2002)
10. Yoo, T.S., Lafortune, S.: Polynomial-time verification of diagnosability of partially observed discrete-event systems (2002)
11. Zaytoon, J., Lafortune, S.: Overview of fault diagnosis methods for discrete event systems. Annual Reviews in Control **37**, 308–320 (2013)