Unfold and Cover: Qualitative Diagnosability for Petri Nets

Stefan Haar
INRIA- IRISA
Campus de Beaulieu
35042 Rennes cedex, France.
stefan.haar@irisa.fr
shaar@site.uottawa.ca

***Abstract*— In recent years, classical discrete event fault diagnosis techniques have been extended to Petri Net system models under partial order semantics [4]–[6]. We propose here to take further advantage of the partial order representation of concurrent processes; we explore the relational structure of occurrence nets to derive a *covering relation*. It indicates that occurrence of some event a inevitable leads to occurrence of some event b, before a, after a, or concurrently. Covering defines a decomposition of occurrence nets into *facets*; we introduce the facet-based concept of q-*diagnosability* - for *qualitative* diagnosability as opposed to quantitative criteria like in [6], [15] -, which is specific to partial order semantics. All objects considered can be computed from a finite unfolding prefix of bounded length.**

I. Introduction

Petri nets (see e.g. [9], [14]) and their partial order unfoldings [2], [10], [12] have been increasingly used in recent years for both fault diagnosis [4]–[6] and control (see e.g. [8]) of asynchronous discrete event systems. The advantage of this semantics lies in the space reduction for representing nonsequential processes that have a high degree of parallelism. In unfoldings, sets of concurrent events are not ordered, which means they have to be represented only once (by one partial order) rather than by giving all their interleavings, whose number is exponential in the size of the concurrent set. The gain in space therefore depends heavily on the degree of parallelism; the motivation is thus very strong in highly distributed systems such as telecommunication networks, see [5]. See also the discussion in the reference [3], entirely dedicated to the necessity of *true concurrency* in the study of distributed discrete event systems. In [4]–[6], fault diagnosis for a Petri net model $\mathscr{N}$ is performed by unfolding the labeled product of $\mathscr{N}$ and an observed alarm pattern $\mathscr{A}$, also in Petri net form. A recursive procedure filters out, on-the-fly, those runs that explain exactly $\mathscr{A}$. In [6], we have presented a characterization of diagnosability adapted to partial order semantics of 1-safe Petri nets; it extends the well-known characterization of diagnosability developed in [15] for FSM models, adapting them to the specificities of nonsequential systems. As in all fault diagnosis, observability and diagnosability of the model must be ensured in order for the diagnosis algorithm to work. In the context of partial order semantics, we introduced and characterized *weak* and *strong diagnosability* [6]: a system is strongly diagnosable if faults are detected a bounded number of events after their occurrence, regardless of the interleaving of events; this generalizes diagnosability in the sense of [11], [15]. Here, we introduce a non-quantitative concept of **q**-*diagnosability*, or *qualitative* diagnosability as opposed to quantitative criteria like those in [6], [15]; it is specific to partial order semantics, and uses a decomposition of occurrence nets into *facets*. The key observation concerns the relational structure of occurrence nets, consisting of a partial order $<$ and a conflict relation #; pairs that are neither ordered, nor in conflict, are collected in the complement relation **co** that indicates concurrency. Within this structure, we identify pairs (x,y) of nodes such that x *logically covers* y in the sense that whenever x occurs, y must eventually occur as well. The reader familiar with occurrence nets may jump ahead to Figure 2 for an illustration: e.g. the pair (a,g) exhibits this covering, i.e. observing a implies that g inevitably has to occur, if it has not already. *Facets* are subnets of the unfolding in which *any* two events cover one another. As a consequence, if some event in a facet occurs, eventually all other events of the facet have to occur in any fair execution (i.e. assuming progress: no enabled event remains enabled forever without occurring). We will define logical covering and facets, prove their key properties and show their link with diagnosability.

Overview

The paper is organized as follows: We start in Section II by recalling definitions for Petri net models and partial order diagnosis techniques. In Section III, we introduce the covering relation and facets, and study their properties. Section IV discusses **q**-diagnosability, and Section V concludes.

II. Petri Nets, Unfoldings, and Diagnosis

A. Nets and homomorphisms

Definition 1 *A* net *is a triple $N = (\mathscr{P}, \mathscr{T}, F)$, where $\mathscr{P}$ and $\mathscr{T}$ are disjoint sets of* places *and* transitions*, respectively, and $F \subset (\mathscr{P} \times \mathscr{T}) \cup (\mathscr{T} \times \mathscr{P})$ is the* flow relation. *In figures, places are represented by circles, and marked places are highlighted in thick; rectangular boxes represent transitions, and arrows represent F. Let $<$ be the transitive closure of F and $\leqslant$ the reflexive closure of $<$. For node $x \in \mathscr{P} \cup \mathscr{T}$, call ${}^\bullet x \triangleq \{x' \mid F(x',x)\}$ the* preset *and $x^\bullet \triangleq \{x' \mid F(x,x')\}$ the* postset *of x; further, let $\lceil x \rceil \triangleq \{x' \mid x' < x\}$ be the* prime

configuration *(see below) or* cone *of x. A* net homomorphism *from N to N′ is a map* $\pi : \mathscr{P} \cup \mathscr{T} \longmapsto \mathscr{P}' \cup \mathscr{T}'$ *such that:*

1) $\pi(\mathscr{P}) \subseteq \mathscr{P}'$, $\pi(\mathscr{T}) \subseteq \mathscr{T}'$, *and*
2) $\pi_{|^\bullet e} : {}^\bullet e \to {}^\bullet\pi(e)$ *and* $\pi_{|e^\bullet} : e^\bullet \to \pi(e)^\bullet$ *induce bijections, for every* $e \in \mathscr{E}$.

Homomorphisms between nets allow to formalize branching processes, see below.

Definition 2 *Two nodes* x, x' *of a net N are* in conflict, *written* $x \# x'$, *if there exist transitions* $t, t' \in \mathscr{T}$ *such that i)* $t \neq t'$, *ii)* ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$, *and iii)* $t \leqslant x$ *and* $t' \leqslant x'$. *A node x is said to be in* self-conflict *iff* $x \# x$. *An* occurrence net *(ON) is a net* $ON = (\mathscr{B}, \mathscr{E}, F)$, *with the elements of* $\mathscr{B}$ *called* conditions *and those of* $\mathscr{E}$ events, *satisfying the additional properties :*

1) *no self-conflict:* $\forall x \in \mathscr{B} \cup \mathscr{E} : \neg[x \# x]$;
2) $\leqslant$ *is a partial order:* $\forall x \in \mathscr{B} \cup \mathscr{E} : \neg[x < x]$;
3) $\forall x \in \mathscr{B} \cup \mathscr{E} : |\lceil x \rceil| < \infty$;
4) *no backward branching:* $\forall b \in \mathscr{B} : |{}^\bullet b| \leq 1$.
5) *the set* $\mathbf{c}_0 \triangleq \min(ON)$ *of minimal nodes of ON is contained in* $\mathscr{B}$.

Here, we add w.l.o.g. restriction 5.) for convenience; it is not required, e.g., in [1]. Note that, as a consequence of property 3), $\mathscr{B} \cup \mathscr{E}$ is well-ordered by $\leqslant$, i.e. there exist no infinite strictly decreasing sequences. Occurrence nets are useful to represent executions of Petri nets, see below: essential dynamical properties are visible via the topological structure of the acyclic graph. Nodes x and x' are *concurrent,* written x **co** x', if neither $x \leqslant x'$, nor $x' \leqslant x$, nor $x \# x'$ hold. A *co-set* is a set $\mathscr{X}$ of pairwise concurrent conditions; a maximal co-set $\mathscr{X}$ w.r.t. set inclusion is called a *cut*, and generically denoted by the symbol $\mathbf{c}$; in particular, $\mathbf{c}_0$ is a cut, called the *initial* cut of *ON*. To highlight the importance of $\mathbf{c}_0$, we will henceforth note occurrence nets as $ON = (\mathscr{B}, \mathscr{E}, F, \mathbf{c}_0)$.

We note for future reference that occurrence nets are a special case of *event structures* [13]:

Definition 3 *A tuple* $(E, <, \#)$ *is an* ***event structure*** *iff:*

1) $(E, <)$ *is a countable partially ordered set,*
2) $\lceil e \rceil$ *is finite for all* $e \in E$,
3) $\# \subseteq E \times E$ *is symmetric and irreflexive, and such that* $\forall\, x, y, z \in E$: $x \# y$ *and* $y < z$ *together imply* $x \# z$.

B. Petri Nets

Let $N = (\mathscr{P}, \mathscr{T}, F)$ be a finite net. A *marking* of net N is a multi-set $M \in \mathfrak{M}(\mathscr{P})$. A *Petri net* (PN) is a pair $\mathscr{N} = (N, M_0)$, where $M_0 \in \mathfrak{M}(\mathscr{P})$ is an *initial* marking. $\mathscr{T} \in \mathscr{T}$ is *enabled* at M, written $M \xrightarrow{t}$, if for all $p \in {}^\bullet t$, $M(p) \geqslant 1$. If $M \xrightarrow{t}$, then t can *fire*, leading to

$$M' \quad = \quad (M - 1_{^\bullet t}) + 1_{t^\bullet},$$

where symbol 1 denotes the set indicator function; write in that case $M \xrightarrow{t} M'$. The set $\mathbf{R}(M_0)$ contains the markings of $\mathscr{N}$ *reachable* through $\longrightarrow$. A Petri net $\mathscr{N} = (N, M_0)$ is *k-safe* if for all $M \in \mathbf{R}(M_0)$ and places p, $M(p) \leqslant k$. 1-safe nets are simply called *safe*. Only safe nets are considered in this article; we will represent reachable markings of nets simply as sets $M \subseteq \mathscr{P}$.

C. Branching Processes and Unfoldings

The branching process semantics reflects the partial order behavior of Petri nets in occurrence nets, thus allowing for structural analysis.

Definition 4 *A* branching process *of the safe Petri net* $\mathscr{N} = (\mathscr{P}, \mathscr{T}, F, M_0)$ *is given by a pair* $\pi = (ON, \pi)$, *where* $ON = (\mathscr{B}, \mathscr{E}, G, \mathbf{c}_0)$, *and* π *is a homomorphism from ON to N, such that i)* $\pi(\mathbf{c}_0) = M_0$, *and ii) for all* $e, e' \in \mathscr{E}$, ${}^\bullet e = {}^\bullet e'$ *and* $\pi(e) = \pi(e')$ *together imply* $e = e'$. *For* π, π' *two branching processes of* $\mathscr{N}$, π' *is a* prefix *of* π, *written* $\pi' \sqsubseteq \pi$, *if there exists an injective homomorphism* ψ *from ON′ into a prefix of ON, such that* ψ *induces a bijection between the initial cuts* $\mathbf{c}_0$ *and* $\mathbf{c}'_0$, *and the composition* $\pi \circ \psi$ *coincides with* π'. *Occurrence net* $\rho = (\mathscr{B}_\rho, \mathscr{E}_\rho, G_\rho, \mathbf{c}_0(\rho))$ *is a* (structural) prefix *of ON, written* $\rho \sqsubseteq ON$, *iff*

1) $\mathscr{B}_\rho \subseteq \mathscr{B}$, $\mathscr{E}_\rho \subseteq \mathscr{E}$, *and* $G_\rho = G_{|(\mathscr{B}_\rho \times \mathscr{E}_\rho) \cup (\mathscr{E}_\rho \times \mathscr{B}_\rho)}$;
2) $e \in \mathscr{E}_\rho \Rightarrow {}^\bullet e \cup e^\bullet \subseteq \mathscr{B}_\rho$
3) $\mathbf{c}_0 = \mathbf{c}_0(\rho)$;*and*
4) ρ *is* causally closed*: if* $x' \leqslant x$ *and* $x \in \rho$, *then* $x' \in \rho$.

A prefix κ *of ON is a* configuration *if* κ *is* conflict-free*, i.e. no two nodes from* κ *are in conflict; we require for convenience that configurations be condition-bordered, i.e. all* $<-$*maximal nodes of* κ *are conditions. A* maximal *configuration (w.r.t. set inclusion) is called a* run *and generically denoted* ω*; denote the set of runs as* Ω.

By theorem 23 of [1], there exists a unique (up to an isomorphism) $\sqsubseteq$-maximal branching process, called the *unfolding* of $\mathscr{N}$ and denoted $\mathscr{U}(\mathscr{N})$; by abuse of notation, we will also use $\mathscr{U}(\mathscr{N})$ for the occurrence net obtained by the unfolding. The principle for effectively constructing the unfolding (see [1], [4], [8]) is as follows: a copy of initial marking M_0 yields the initial conditions; events are appended to concurrent conditions that enable them, and are followed by the post-conditions they create. An illustration is given in Figure 1, taking up the running example from [4], [6]. Petri net $\mathscr{N}$ is shown on the left, and a branching process $\pi = (ON, \pi)$ of $\mathscr{N}$ on the right hand side. Conditions are labeled by places, events by transitions. A configuration is shown in gray. The mechanism for constructing the unfolding of $\mathscr{N}$ is illustrated in the middle.

D. Configurations

The nonsequential executions of safe Petri net $\mathscr{N}$ are in one-to-one correspondence with the configurations of $\mathscr{U}(\mathscr{N})$. in every configuration; further, every *finite* configuration κ terminates at a cut, which we denote $\mathbf{c}_\kappa$. The mapping $\kappa \mapsto \mathbf{c}_\kappa$ is bijective; for each cut $\mathbf{c}$, the union of the cones of all conditions in $\mathbf{c}$ yield the unique configuration κ such that $\mathbf{c} = \kappa_\mathbf{c}$. Moreover, one has the following correspondence: for every reachable marking $M \subseteq \mathscr{P}$ of $\mathscr{N}$, there exists at least one cut $\mathbf{c}$ of $\mathscr{U}(\mathscr{N})$ such that $\pi(\mathbf{c}) = M$

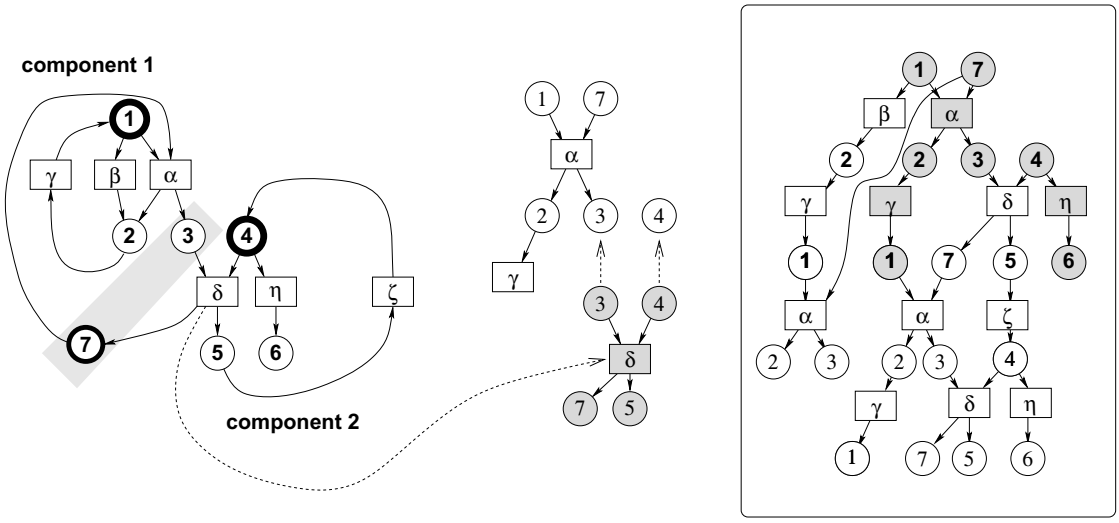

Fig. 1. Unfolding procedure in the context of the Petri net example from [4]

for all p, and the unique configuration κ such that $\mathbf{c}_\kappa = \mathbf{c}$ is such that execution of κ takes M_0 to M; write $M_0 \xrightarrow{\kappa} M$ for this. Conversely, every finite configuration κ corresponds to a unique reachable marking $M(\kappa)$ given by $M(\kappa) \triangleq \pi(\mathbf{c}_\kappa)$.

E. Finite Complete Prefix

If $\mathscr{U}(\mathscr{N})$ is infinite, we are naturally interested in finite prefixes of $\mathscr{U}(\mathscr{N})$ that are *complete* in the sense that their analysis allows to derive results for all of $\mathscr{U}(\mathscr{N})$. The definition and size of such prefixes varies with the intended purpose; see [10] for a systematic treatment. We use here the following Definition, similar to that in [8]:

Definition 5 *The order 1 unfolding, denoted $\mathscr{U}_1(\mathscr{N})$, is a finite prefix of the unfolding obtained by stopping the construction of the unfolding when we reach a **cut-off** event e, i.e., an event such that:*
EITHER $\lceil e \rceil$ reproduces the initial marking: $M(\lceil e \rceil) = M_0$;
OR there exists an event $e' \neq e$ such that :

1) *The prime configuration for e' is a prefix of that of e: $\lceil e' \rceil \subseteq \lceil e \rceil$;*
2) *the markings reached firing the two configurations are equivalent: $M(\lceil e \rceil) = M(\lceil e' \rceil)$.*

In the following we call e' the mirror transition of e in $\tilde{\mathscr{N}_1}(M_0)$. Once $\mathscr{U}_1(\mathscr{N})$ constructed, assume we continue the unfolding until we reach an event e such that there exist another event e' with the following properties:

- *either e' does not belong to $\mathscr{U}_1(\mathscr{N})$ or it is a cut-off event of $\mathscr{U}_1(\mathscr{N})$;*
- *The prime configuration for e' is a prefix of that of e: $\lceil e' \rceil \subseteq \lceil e \rceil$;*
- *the two configurations are marking-equivalent: $M(\lceil e \rceil) = M(\lceil e' \rceil)$.*

The resulting net $\mathscr{U}_2(\mathscr{N})$ is called order 2 unfolding; order n unfoldings $\mathscr{U}_n(\mathscr{N})$ are defined recursively in the same manner.

Note that the initial definition from [12] used as cutoff criterion *cardinality*, i.e. $|\lceil e' \rceil| < |\lceil e \rceil|$ instead of $\lceil e' \rceil \subseteq \lceil e \rceil$. With this choice, the prefix obtain is in general shorter; however, it may not be complete w.r.t. computing the covering relation below.

F. Observability and Diagnosability

The asynchronous diagnosis of [4]–[6] proceeds as follows: Take the Petri net model $\mathscr{N}$ of the system, with transition labeling taking values in an alphabet A of alarms, and the Petri net representation $\mathscr{A}$ of the observed alarm pattern; that is, the events of $\mathscr{A}$ are labeled by the observed alarms in A, and in the order they were observed. Then form the product net $\mathscr{N} \times \mathscr{A}$ by fusing transitions carrying the same label. All executions of $\mathscr{N} \times \mathscr{A}$ correspond to executions of $\mathscr{N}$; the converse is obviously not true. Moreover, not all executions of $\mathscr{N} \times \mathscr{A}$ cover all of $\mathscr{A}$; in general, only a proper prefix of the observation is explained by a given run of $\mathscr{N} \times \mathscr{A}$. In the unfolding $\mathscr{U}(\mathscr{N} \times \mathscr{A})$, take all those branches that *fully explain* $\mathscr{A}$; the corresponding executions of $\mathscr{N}$ form the *diagnosis set* of all possible explanations of $\mathscr{A}$ in the model $\mathscr{N}$. Without going into the details of the diagnosis method, it emerges immediately that the algorithm can only converge if $\mathscr{N}$ cannot perform *invisible cycles*; that is, sufficiently many transitions of $\mathscr{N}$ must carry a non-empty label and thus be *visible*, such that the net cannot leave a marking M and then return to M without having produced a visible alarm on the record. That is (compare [6]), for any two configurations κ, κ' such that κ is a proper prefix of κ' and $\kappa \sim_M \kappa'$, there must be at least one visible event in $\kappa \backslash \kappa'$ for observability of the net. This property will be assumed throughout. *Diagnosability* in the sense of [15] is the capacity of detecting that a fault has occurred, a bounded number of steps after the fault. In [6], the metric criteria of [15] are lifted to partial orders via quantitative criteria using *height* of configurations. In contrast, our proposal here will

yield a definition of diagnosability in *qualitative* terms, and can be checked directly on a bounded prefix.

G. Definitions and Conventions

For the remainder of the paper, fix an observable Petri net $\mathscr{N} = (N, M_0)$ with $N = (\mathscr{P}, \mathscr{T}, F)$, a set A of alarm labels, $\varepsilon \in A$ the empty symbol and $\lambda : \mathscr{T} \to A$ a labeling function . Denote as $\mathscr{I} \triangleq \lambda^{-1}(\{\varepsilon\})$ the set of *invisible* transitions; dually, let $\mathscr{T}_A \triangleq \mathscr{T} \setminus \mathscr{I}$ be the set of *visible* transitions. Let $\mathscr{U}(\mathscr{N}) = (\mathscr{B}, \mathscr{E}, G, \mathbf{c}_0)$ be the unfolding of $\mathscr{N}$ with homomorphism π, and denote as $O \triangleq \pi^{-1}(\mathscr{T}_A)$ the set of observable events. Further, let $\xi \in \mathscr{I}$ be a fault to be observed; let $\mathscr{E}_\xi \triangleq \pi^{-1}(\{\xi\})$. For configurations κ, κ' of $\mathscr{N}$, write $\kappa \sim_A \kappa'$ iff the sets κ_A, κ'_A of observable events of κ and κ', respectively, are isomorphic partially ordered sets (with the order relation induced by $\leqslant$). Let $\xi \in \mathscr{I}$ be a *fault*[1] to be diagnosed. Configurations κ, κ' are ξ-*equivalent* iff either both contain a ξ-event, or neither of them does:

$$\kappa \sim_\xi \kappa' \quad \textit{iff} \quad \left[\kappa \cap \mathscr{E}_\xi \neq \emptyset \iff \kappa' \cap \mathscr{E}_\xi \neq \emptyset.\right] \tag{1}$$

III. Covering and Facets

Before turning to diagnosability below, we prepare the ground with structural analysis of occurrence nets.

A. Covering Relation

Consider the occurrence net in Figure 2: for any run ω,

$$k \in \omega \quad \Rightarrow \quad e \in \omega \Rightarrow b \in \omega, \tag{2}$$

$$a \in \omega \quad \iff \quad \neg(b \in \omega) \iff c \in \omega, \tag{3}$$

$$e \in \omega \quad \iff \quad f \in \omega, \tag{4}$$

etc. Property (2) reflects the inheritance of # under $<$; the reader is invited to check that (3) and (4) follow from the maximality of runs. One might suspect that, to derive (3,4) from the relational structure, one would have to explore the entire set of configurations. We will show here that it suffices to derive a binary *covering* relation, computable from a finite bounded prefix ρ of the unfolding.We begin with the notion of *root conflict sets* :

Definition 6 *The* conflict set *of node $x \in (\mathscr{B} \cup \mathscr{E})$ is*

$$\#[x] \quad \triangleq \quad \{x' \mid x \# x'\}.$$

The root conflict set *of x is given by*

$$\underline{\#}[x] \quad \triangleq \quad \{y \mid x \# y \ \wedge \ \forall\, z : z < y \Rightarrow \neg(z \# x)\}.$$

Node x implies or covers y, written $x \triangleright y$, iff $\#[x] \supseteq \#[y]$; that is, iff for all z, $z\#y$ implies $z\#x$.

One immediately checks that $\triangleright$ is a reflexive and transitive relation. Define the *covering range* of node x as

$$\triangleright[x] \quad \triangleq \quad \{y \mid x \triangleright y\}.$$

The covering relation mirrors run inclusion:

[1]we only consider single fault types to avoid technicalities w.l.o.g.

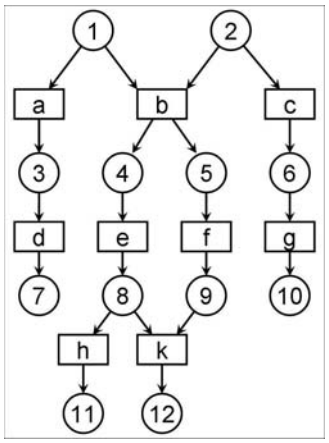

Fig. 2. Occurrence net whose covering and facets are given in the text

Lemma 1 *$x \triangleright y$ holds iff for all runs ω,*

$$x \in \omega \quad \Rightarrow \quad y \in \omega \tag{5}$$

Proof: If $x \in \omega$ and $y \notin \omega$, there exists a node $z \in \#[y] \cap \omega$; in fact, otherwise $\omega \cup \lceil y \rceil$ would be a configuration, and ω could not be maximal. If $x \triangleright y$, $z \in \#[x] \cap \omega$, which is impossible, so $\neg(x \triangleright y)$. Conversely, suppose that (5) holds for every ω; then there exists z such that $z\#y$ and $\neg(z\#x)$. But then there exists a run ω_z such that $x, z \in \omega_z$, but by assumption $y \notin \omega_z$, hence (5) is violated for ω_z. □

Relation $\triangleright$ is not a partial order: consider $e \triangleright f$ and $f \triangleright e$ in Figure 2. This is a crucial fact behind the definition of *facets* below. However, the following holds:

Lemma 2 *$x < y$ implies that $y \triangleright x$.*

Proof: By inheritance of #, $x < y$ implies $\#[x] \subseteq \#[y]$. □

As a consequence, we have:

Lemma 3 *$\triangleright[x]$ is a configuration.*

Proof: Since $\lceil x \rceil \subseteq \triangleright[x]$ by Lemma 1, we have $\mathbf{c}_0 \subseteq \triangleright[x]$; thus Lemma 2 implies the result. □

In Figure 2, we have the following covering ranges:

$$\begin{aligned} \triangleright[b] = \triangleright[e] = \triangleright[f] &= \{b, e, f\} \\ \triangleright[h] &= \{b, e, f, h\} \\ \triangleright[k] &= \{b, e, f, k\} \\ \triangleright[a] = \triangleright[d] = \triangleright[c] = \triangleright[g] &= \{a, d, c, g\} \end{aligned}$$

The following result is crucial for the feasibility of our approach: it shows that in order to decide whether $x \triangleright y$, it suffices to know $\underline{\#}[x]$ and $\underline{\#}[y]$:

Theorem 1 *The set $\#[x]$ is generated by $\underline{\#}[x]$ through inheritance:*

$$\#[x] \quad = \quad \{z \mid \exists\, y \in \underline{\#}[x] :\ y \leqslant z\}. \tag{6}$$

As a consequence, $x_1 \triangleright x_2$ iff $\underline{\#}[x_1] \supseteq \underline{\#}[x_2]$.

Proof: The inclusion

$$\#[x] \quad \supseteq \quad \{z \mid \exists\, y \in \underline{\#}[x] :\ y \leqslant z\} \tag{7}$$

being obvious, it remains to show

$$\#[x] \quad \subseteq \quad \{z \mid \exists\, y \in \underline{\#}[x] :\ y \leqslant z\}. \tag{8}$$

Take any $y \in \#[x] \backslash \underline{\#}[x]$. Since $x\#y$, there exist a condition b_1 and events x_1, y_1 such that

1) $x_1 \neq y_1$;
2) $b_1 \in {}^\bullet x_1 \cap {}^\bullet y_1$;
3) $x_1 \leqslant x$ and $y_1 \leqslant y$.

Let $n \geq 1$. If $y_n \in \underline{\#}[x]$, we are done; otherwise there exist a condition b_{n+1} and events x_{n+1}, y_{n+1} such that

1) $x_{n+1} \neq y_{n+1}$;
2) $b_{n+1} \in {}^\bullet x_{n+1} \cap {}^\bullet y_{n+1}$;
3) $x_{n+1} \leqslant x$ and $y_{n+1} < y_n$.

If we find recursively infinitely many $y_1, y_2, \ldots$, we have a contradiction with property 3) of Definition 2, since

$$y \quad \geqslant \quad y_1 > y_2 > \ldots.$$

We conclude that there exists $n \in \mathbb{N}$ such that $y_n \in \underline{\#}[x]$, and this proves (8). □

B. Facets

The occurrence net *ON* decomposes into *facets*:

Definition 7 *A **facet** of ON is a strongly connected component of $\triangleright$; that is, a maximal set $\mathsf{D} \subseteq (\mathscr{E} \cup \mathscr{B})$ of nodes such that for any $x, y \in \mathsf{D}$, one has*

$$x \triangleright y \quad \textit{and} \quad y \triangleright x.$$

Denote as $\mathsf{D}(x)$ the unique facet that contains x.

In figure 2, the facets are

$$\{a, d, c, g\},\ \{b, e, f\},\ \{h\},\ \{k\}.$$

Concerning the shape of facets, we obtain:

Lemma 4 *Facets are convex, i.e. $x, y \in \mathsf{D}$ and $x < z < y$ together imply $z \in \mathsf{D}$.*

Proof: By Lemma 2, $\#[x] \subseteq \#[z] \subseteq \#[y]$; assumption $\#[x] = \#[y]$ yields $\mathsf{D}(x) = \mathsf{D}(y) = \mathsf{D}(z)$. □

However, more is true: facets behave to the outside world like their elements, i.e. *ON* can be represented by the quotient facet structure. To make this precise, let x_i be a node of *ON*, let $\mathsf{D}_i \triangleq \mathsf{D}(x_i)$, and set

$$\mathsf{D}_1 \prec_\Delta \mathsf{D}_2 \quad \Longleftrightarrow \quad \left\{ \begin{array}{ll} & \mathsf{D}_1 \neq \mathsf{D}_2 \\ \exists & y_1 \in \mathsf{D}_1, y_2 \in \mathsf{D}_2 : \\ & y_1 < y_2 \end{array} \right. \tag{9}$$

$$\mathsf{D}_1 \#_\Delta \mathsf{D}_2 \quad \Longleftrightarrow \quad [\exists\, y_1 \in \mathsf{D}_1, y_2 \in \mathsf{D}_2 :\ y_1 \# y_2] \tag{10}$$

Relation $\prec_\Delta$ from Definition (9) is a partial order by Lemma 4; $\#_\Delta$ is well-defined since $y_1 \# y_2$ implies $z_1 \# z_2$ for all z_1 from D_1 and z_2 from D_2. Note that $y_1 \# y_2$ also implies that $\mathsf{D}_1 \neq \mathsf{D}_2$: *Facets are conflict-free*. One checks easily that

$$\mathsf{D}_1 \# \mathsf{D}_2 \prec_\Delta \mathsf{D}_3 \quad \Rightarrow \quad \mathsf{D}_1 \# \mathsf{D}_3, \tag{11}$$

and finds that $(\Delta, \prec_\Delta, \#_\Delta)$ is an event structure in the sense of Definition 3. We denote as $\lceil \mathsf{D} \rceil$ the set of facets

$$\lceil \mathsf{D} \rceil \quad \triangleq \quad \{\mathsf{D}' \mid \mathsf{D}' \prec_\Delta \mathsf{D}\}.$$

By Lemma 4, the set union of all facets in $\lceil \mathsf{D} \rceil$ spans a configuration of *ON*; we denote this configuration as $\kappa(\mathsf{D})$.

Finally, since we will be dealing with infinite unfoldings in general, two problems have to be considered:

1) Are there facets of *infinite size* ?
2) Are there *infinitely many different facets* ?

Consider first the possibility of *infinite facets*. Since it is easily checked that the width of runs (i.e. the size of a maximal concurrent set) in the unfolding of a safe Petri net $\mathscr{N}$ is bounded by the number of places in $\mathscr{N}$, an infinite facet is possible only if a run has a suffix without any branching conditions. This means that for all facets to be finite, it suffices to forbid marking-reproducing cycles with no conflict resolution:

Assumption A. *For any two distinct configurations κ, κ' of $\mathscr{N}$ such that κ is a prefix of κ' and $\kappa \sim_M \kappa'$, there must exist a configuration κ'' such that κ is a prefix of κ'', and for some $x' \in \kappa'$ and some $x'' \in \kappa''$ we have $x' \# x''$.*

Under assumption **A**, we now address the second question. Call facets D_1 and D_2 *π-isomorphic* iff there exists a net isomorphism ψ mapping D_1 to D_2 in such a way that the relations induced by *ON*'s $<, \#$, **co** are preserved and respected, and $\pi_{|\mathsf{D}_1} = \pi_{|\mathsf{D}_2} \circ \psi$.

Theorem 2 *Let $ON = (\mathscr{B}, \mathscr{E}, G, \mathbf{c}_0) = \mathscr{U}(N, M_0)$. Under Assumption A, every π-isomorphism class of facets in $\mathscr{N}$ has at least one element in $\mathscr{U}_2(\mathscr{N})$.*

Proof: Consider first and second order unfoldings $\mathscr{U}_1(\mathscr{N})$ and $\mathscr{U}_2(\mathscr{N})$ of $\mathscr{N}$. The only facets from $\mathscr{U}_1(\mathscr{N})$ that may not have a π-isomorphic counterpart in $\mathscr{U}_2(\mathscr{N}) \backslash \mathscr{U}_1(\mathscr{N})$ are those that hit the initial cut; moreover, some facets might be partly in $\mathscr{U}_1(\mathscr{N})$ and partly in $\mathscr{U}_2(\mathscr{N}) \backslash \mathscr{U}_1(\mathscr{N})$. By Assumption **A**, however, every facet that intersects $\mathscr{U}_1(\mathscr{N})$ lies entirely within $\mathscr{U}_2$. Since $\mathscr{N} = (N, M_0)$ is safe, only a finite number of markings are reachable. Combining these arguments yields the result. □

Since $\mathscr{U}_2(\mathscr{N})$ is finite, Theorem 2 entails the number of π-isomorphism classes of facets in *ON* is finite.

Remark: Recent results ([7]) show that Assumption A can be dropped in the above, yet these results require too much space to be developed here.

IV. Application: q-Diagnosability

We are now ready to introduce a concept of diagnosability that combines diagnosis of past events with prediction of inevitable ones. In Figure 2, supposing we observe a; we are then assured that b is going to occur, but also g and a fortiori c. That is, the implications of a single observation stretch into

the future and into processes that are parallel to the observed one. Petri net unfoldings thus call for a different notion of diagnosability that uses the above relational structure, rather than the length of interleaved sequences.

Let us formalize one of the notions that spring from the above analysis (a vaster exploration is in preparation [7]) Keeping the same setting and notations, let us define the *pro-cone* of a node $x \in \mathscr{E} \cup \mathscr{B}$ as

$$\lceil\lceil x \rceil\rceil \quad \triangleq \quad \kappa(\mathsf{D}(x)); \tag{12}$$

the *closure* of a configuration κ is then

$$\mathbf{Cl}(\kappa) \quad \triangleq \quad \bigcup_{x \in \kappa} \lceil\lceil x \rceil\rceil. \tag{13}$$

Configuration κ is *closed* iff $\mathbf{Cl}(\kappa) = \kappa$. Notice that $\mathbf{Cl}(\kappa)$ coincides with the configuration obtained by intersecting all runs that extend κ; this makes closed configurations key entities for asynchronous diagnosis. We are now ready to give the definition of **q**-*diagnosability* (compare (1)):

Definition 8 *If $\mathscr{N}$ is observable, a fault ξ is* **q**-*diagnosable iff for any two configurations κ, κ',*

$$\mathbf{Cl}(\kappa) \sim_A \mathbf{Cl}(\kappa') \quad \Rightarrow \quad \mathbf{Cl}(\kappa) \sim_\xi \mathbf{Cl}(\kappa'). \tag{14}$$

In words, ξ is **q**-diagnosable iff for any two configurations κ, κ' the following holds: if the *inevitable common part* of all runs extending κ and κ', respectively, are observationally equivalent, they have to be fault equivalent. Returning to Figure 2, suppose events h and k are both unobservable, and that $\lceil e \rceil \cup \lceil f \rceil$ is fault-free; then, for **q**-diagnosability, h and k have to be either both faulty or both fault-free. Suppose now that f is a fault, i.e. $\pi(f) = \xi$, and that $\mathbf{Cl}(\lceil f \rceil) = \{b, e, f\}$ and $\mathbf{Cl}(\lceil a \rceil) = \{a, c, d, g\}$ satisfy $\mathbf{Cl}(\lceil f \rceil) \sim_A \mathbf{Cl}(\lceil a \rceil)$; then there must be a fault event x, $\pi(f) = \xi$, in $\mathbf{Cl}(\lceil a \rceil) = \{a, c, d, g\}$, or $\mathscr{N}$ is not **q**-diagnosable. We observe that **q**-diagnosability includes both diagnosis of the past as 'prediction' of concurrent or future events. This notion of diagnosis is thus well adapted to asynchronous systems where the the precise interleaving of events is not available; concurrent events will occur and go unnoticed *unless* they change future branchings. It is therefore natural to focus diagnosis on the *closed configurations*. Now, the latter are obtained as the configurations of the *facet* event structure $(\Delta, \prec_\Delta, \#_\Delta)$. Offline verification of **q**-diagnosability can therefore proceed using the truncation of Δ to $\mathscr{U}_k(\mathscr{N})$ (for suitable k, see above), which is a much smaller structure than $\mathscr{U}_k(\mathscr{N})$ itself. Effective algorithms to exploit $(\Delta, \prec_\Delta, \#_\Delta)$ for diagnosability checks and diagnosis itself are the topic of future work.

V. Conclusion and Outlook

We have introduced a quantitative diagnosability criterion that can be effectively verified on a finite prefix of the unfolding. In fact,

- for determining covering relations and facets, it suffices to compute the root conflict set for each node x considered (Theorem 1), and
- $\mathscr{U}_2(\mathscr{N})$ contains all facets of $\mathscr{U}(\mathscr{N})$ up to π-isomorphic repetition (Theorem 2).

We have shown recently [7] that the covering relation can always be effectively computed: in fact, for any safe net $\mathscr{N}$, there exists a constant $K(\mathscr{N})$ such that for all n and all nodes x, y of $\mathscr{U}_n(\mathscr{N})$ such that $\neg(x \triangleright y)$ there exists a *witness* z in $\mathscr{U}_{n+K+1}(\mathscr{N})$, i.e. $z \# x$ and $\neg(z \# y)$ [7].

It remains to optimize the exploration of the data structures of $\mathscr{U}(\mathscr{N})$ and Δ for a most efficient verification of diagnosability. Computing the covering relation is polynomial in the size of $\mathscr{U}_2(\mathscr{N})$; on the other hand, the worst case size of $\mathscr{U}_2(\mathscr{N})$ is exponential in the sized of $\mathscr{P}$. However, as mentioned in the introduction, the cases where the modeling with Petri nets is well suitable in the first place - namely for highly distributed and asynchronous systems -, generally also have an order 2 unfolding of reasonable size.

Acknowledgments: This work was funded in part under the SWAN contract [16] and completed during the author's sabbatical leave, supported by INRIA, at School of Information Theory and Engineering, University of Ottawa, Canada.

References

[1] J. Engelfriet. *Branching Processes of Petri Nets.* Acta Informatica **28**:575–591, 1991.

[2] J. Esparza, S. Römer, W. Vogler. An improvement of McMillan's unfolding algorithm. *Form. Meth. in Sys. Des.* **20**(3):285-310, 2002.

[3] E. Fabre and A. Benveniste. Partial Order Techniques for Distributed Discrete Event Systems: why you can't avoid using them. *INRIA Research report* **5916**, Feb. 2007; http://hal.inria.fr/inria-00130025. Extended version of a plenary Address at WODES 2006.

[4] E. Fabre, A. Benveniste, C. Jard, and S. Haar. Diagnosis of Asynchronous Discrete Event Systems, a Net Unfolding Approach. *IEEE Trans. Aut. Control* **48**(5)·714–727, May 2003.

[5] E. Fabre, A. Benveniste, C. Jard, and S. Haar. Distributed monitoring of a concurrent and asynchronous systems. *Disc. Event Dyn. Sys.* **15**(1):33–84, Mar. 2005

[6] S. Haar, A. Benveniste, E. Fabre, and C. Jard. Partial Order Diagnosability of Discrete Event Systems Using Petri Net Unfoldings. In: *Proc. 42nd CDC,* 2003. Extended version: S. Haar. Diagnosability Of Asynchronous Discrete Event Systems in Partial Order Semantics. Research Report INRIA, No 5248, 2004.

[7] S. Haar. Covering relation and Diagnosability. In preparation.

[8] A. Giua and C. Xie. Control of safe ordinary Petri nets using unfolding. *Disc. Event Dyn. Sys.* **15**(4):349–373, Dec. 2005

[9] L.E. Holloway, B.H. Krogh and A. Giua. A Survey of Petri Net Methods for Controlled Discrete event systems. *Disc. Event Dyn. Sys.* **7**:151–190, 1997.

[10] V. Khomenko, M. Koutny, and W. Vogler. Canonical Prefixes of Petri Net Unfoldings. *Acta Informatica* **40**:95–118, 2003.

[11] F. Lin. Diagnosability of discrete event systems and its applications. *Disc. Event Dyn. Sys.* 4(1), 1994, pp. 197-212.

[12] K. McMillan. Using Unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. *4th Workshop on Computer Aided Verification* 164–174, 1992.

[13] M. Nielsen, G. Plotkin, G. Winskel. Petri nets, event structures, and domains, Part I. *TCS* **13**:85–108, 1981.

[14] J.L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice-Hall, 1981.

[15] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *IEEE Trans. Aut. Control* 40(9), 1555-1575, 1995.

[16] SWAN project, RNRT, Decision number 03 S 481. See URL: http://swan.elibel.tm.fr (in French).

[17] G. Winskel. Event structures. *Advances in Petri nets,* LNCS **255**: 325–392, Springer Verlag, 1987.