Multilevel Transitive and Intransitive Non-Interference, Causally*

Paolo Baldan[1] and Alessandro Beggiato[2]

[1] Università di Padova, Dipartimento di Matematica,
baldan@math.unipd.it
[2] IMT School for Advanced Studies Lucca
alessandro.beggiato@imtlucca.it

Abstract. We develop a theory of non-interference for multilevel security domains based on causality, with Petri nets as a reference model. We first focus on transitive non-interference, where the relation representing the admitted flow is transitive. Then we extend the approach to intransitive non-interference, where the transitivity assumption is dismissed, leading to a framework which is suited to model a controlled disclosure of information. Efficient verification algorithms based on the unfolding semantics of Petri nets stem out of the theory.

1 Introduction

Starting with [1], the notion of non-interference has been widely used in the study of information flow security. In the simplest scenario, entities are classified according to two levels, a confidential level *High* and a public level *Low*. Information is allowed to flow from *Low* to *High*, but not vice-versa. When dealing with formalisms describing concurrent components that can interact and synchronize, like process calculi and Petri nets, a popular formulation of non-interference is Non-Deducibility on Composition (NDC). It states that a component S is free of interference whenever S running in isolation, seen from the low level, is behaviorally equivalent to S interacting with any parallel high level component [2,3,4,5,6,7,8,9]. Intuitively, the behavior of the *High* part of the system is required not to cause any modification in the behavior of the *Low* part.

This informal reference to causality is made formal in [7] that, relying on some previous work [5], provides a causal characterization of BNDC (Bisimulation-based NDC) on Petri nets, in terms of the unfolding semantics [10]. The interest for a causal characterization is not only of theoretical nature. On the pragmatic side the use of a true concurrent semantics, like the unfolding, which represents interleaving only implicitly, is helpful to face the state explosion problem which affects the verification of concurrent systems.

The approach in [7] works in a two-level setting, possibly with downgrading [11], while since its infancy (see, e.g., [12]) information flow security has

* Supported by MIUR project CINA and the Padua University project ANCORE.

recognized the usefulness of dealing with multilevel security domains where a relation between levels, referred as a security policy, specifies the admitted flows. The transitive nature of information flow – if information flows from level A to level B and from B to C then it necessarily flows from A to C – naturally leads to work in domains where the security policy is a partial order, only allowing a flow of information from lower to higher levels (no read-up, no write-down). The order can be total, expressing a hierarchy of confidentiality degrees (e.g., top secret, secret, confidential and unclassified in a military setting). It can also be partial, typically when various confidentiality criteria are combined into a single domain. E.g., an administration could keep public and sensitive citizen data concerning taxes and civil status. Independent access rights to sensitive tax and civil status data naturally leads to a lattice of security levels.

As argued, e.g., in [13] it can also be natural to consider *intransitive* policies, in a way that a direct flow between two levels, say from A to B, can be forbidden, while a flow mediated through a third level, say D, is admitted. Intransitive policies are suited, for instance, for representing downgrading of confidential information. This allows for a controlled form of leakage, making such policies more realistic than pure non-interference policies that require the complete isolation of confidential levels. More generally, intransitive policies allow one to describe the (possibly cyclic) paths on which information is allowed to flow in a system.

In this paper the approach of [7], providing a causal characterization of the BNDC (Bisimulation-based NDC) property for (safe) Petri nets based on the unfolding semantics, is extended to deal with multilevel transitive policies. Generalizing [11] we also treat the intransitive case, namely we develop a multilevel theory for BINI [6], an adaptation of BNDC to intransitive domains. The non-interference properties of interest are characterized in terms of the absence of suitable causal dependencies in the unfolding, witnessed by places where illegal interactions occur. This enables the definition of algorithms that checks the non-interference properties on a suitably defined complete prefix of the unfolding.

The unfolding-based algorithms are implemented in a tool MultiUBIC [14]. Compared to tools that exploit the reachability graph of the net, like ANICA (Automated Non-Interference Check Assistant) [15] and PNSC (Petri Net Security Checker) [16], thanks to the partial order representation of concurrency, MultiUBIC - as its predecessor UBIC – leads to a gain of efficiency for highly concurrent systems where the unfolding prefix can be exponentially smaller than the complete state space (see e.g. [17]). The verification of multilevel policies can be also reduced to a number of problems on two-level security domains (enriched with a downgrading level in the intransitive case). MultiUBIC comes equipped with facilities for performing the reduction. The experiments suggest that, in general, a direct multilevel verification is more efficient when the number of levels increases, but situations are singled out where the reduction is convenient.

Synopsis. In § 2 we define multilevel security domains and we review some Petri net notions. In § 3 we focus on transitive policies, providing a causal characterization of the BNDC property and a verification algorithm. In § 4 we extend the results to intransitive policies. In § 5 we describe the tool MultiUBIC. In § 6 we draw some conclusions.

2 Multilevel Security Domains and Petri Nets

In this section, after introducing multilevel security domains, we review some basic notions about Petri nets, with special attention to their unfolding semantics, later used to provide a causal characterization of the non-interference properties.

2.1 Multilevel Security Domains

Definition 1 (multilevel security domain). *A* multilevel security domain *(MSD)* $(\mathscr{L}, \rightsquigarrow)$ *is a finite set of security levels* $\mathscr{L}$*, endowed with a reflexive relation* $\rightsquigarrow \subseteq \mathscr{L} \times \mathscr{L}$ *called a* security policy*. When* $\rightsquigarrow$ *is transitive we call* $(\mathscr{L}, \rightsquigarrow)$ *a* transitive multilevel security domain*.*

The security policy specifies the legal information flows. It is reflexive because entities at the same level should be able to freely exchange information. Without loss of generality, a transitive MSD will be assumed to be a partial order. In fact, if $\rightsquigarrow$ is a proper preorder (i.e, not antisymmetric), we can equivalently consider the partial order obtained as its quotient under the equivalence $\rightsquigarrow \cap \rightsquigarrow^{-1}$. Since equivalent levels can communicate in either direction, they can be safely collapsed. Examples of MSD will be discussed later, after introducing also net systems. Given $S \subseteq \mathscr{L}$ we write $\overline{S}$ for its complement $\mathscr{L} \setminus S$.

Definition 2 (upper sets and targets). *Let* $(\mathscr{L}, \rightsquigarrow)$ *be a MSD. An* upper set *is a subset* $U \subseteq \mathscr{L}$ *such that if* $L \in U$ *and* $L \rightsquigarrow L'$ *then* $L' \in U$*. Given a security level* $L \in \mathscr{L}$ *its set of* targets *is* $\uparrow L = \{L' \in \mathscr{L} \mid L \rightsquigarrow L'\}$*, while the* strict targets *are* $\mathrel{\overset{\circ}{\uparrow}} L = \uparrow L \setminus \{L\}$*.*

An entity (user, program, variable, instruction) with associated security level L has permission to influence, or to write, or to pass information only to entities with security level in $\uparrow L$. Any other information flow is a violation of the policy. Targets are defined on sets $U \subseteq \mathscr{L}$ by letting $\uparrow U = \bigcup_{L \in U} \uparrow L$ and $\mathrel{\overset{\circ}{\uparrow}} U = \uparrow U \setminus U$.

2.2 Petri Nets and Net Systems

A *(Petri) net* is a tuple $N = (P, T, F)$ where P, T are disjoint sets of *places* and *transitions*, respectively, and $F : (P \times T) \cup (T \times P) \to \{0, 1\}$ is the *flow function*. Graphically places and transitions are drawn as circles and rectangles, respectively, while the flow function is rendered by means of directed arcs connecting places and transitions. For $x \in P \cup T$ we define its *pre-set* ${}^\bullet x = \{y \in P \cup T : F(y, x) = 1\}$ and its *post-set* $x^\bullet = \{y \in P \cup T : F(x, y) = 1\}$. A *marking* of N is a function $m : P \to \mathbb{N}$. A transition $t \in T$ is *enabled* at a marking m, denoted $m[t\rangle$, if $m(p) \geq F(p, t)$ for all $p \in P$. If $m[t\rangle$ then t can be *fired* leading to a new marking m', written $m[t\rangle m'$, defined by $m'(p) = m(p) + F(t, p) - F(p, t)$

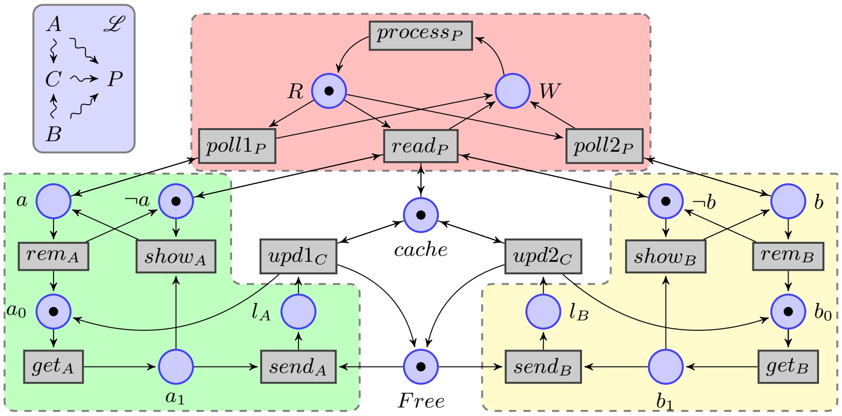

Fig. 1. A non-BNDC net system under the security domain $\mathscr{L}$ (top left).

for all places $p \in P$. The enabling and firing relations are extended to $\sigma \in T^*$ (finite sequences of elements of T) by defining $m[\varepsilon\rangle m$ (where ε is the empty sequence) and $m[\sigma\rangle m'[t\rangle m''$ imply $m[\sigma t\rangle m''$. Markings are represented as black dots, called *tokens*, inside places. A *marked net* is a pair $\mathbf{N} = (N, m_0)$ where N is a net and m_0 is a marking of N. A marking m' is *reachable* if there exists $\sigma \in T^*$ such that $m_0[\sigma\rangle m'$. The set of reachable markings of $\mathbf{N}$ is denoted by $[m_0\rangle$. When $m[t\rangle m'$, the marking m', uniquely determined by m and t, is denoted by $\langle m[t\rangle$. Analogously, for $\sigma \in T^*$, if $m[\sigma\rangle$ we can define the marking $\langle m[\sigma\rangle$. A net $\mathbf{N}$ is *safe* if for every $p \in P$ and every $m \in [m_0\rangle$ we have $m(p) \leq 1$.

In order to formalize information flow properties in the setting of Petri nets, an MSD $\mathscr{L}$ is fixed and, as in [5,6], transitions are associated with security levels.

Definition 3 (net system). *A* net system *is a tuple $N = (P, T, F, \lambda)$ where (P, T, F) is a Petri net and $\lambda : T \to \mathscr{L}$ is a function that assigns a security level to each transition. For $S \subseteq \mathscr{L}$ we define $T_S = \{t \in T \mid \lambda(t) \in S\}$, the set of transitions whose security level is in S. An S-*system *is a net system such that $T = T_S$, i.e. a system only capable of performing actions whose security level belongs to S.*

Consider the net system and security domain in Fig. 1. It represents a device consisting of two independent sensors getting new measures for a processor, that, in turn, can poll them to acquire more recent data. Each sensor has a cyclic behavior. For instance, the left sensor is capable to get a measure (get_A). Such measure can be exposed at its interface ($show_A$) and then removed after a while (rem_A), restarting the cycle. Alternatively, the measure can be sent to a shared cache ($send_A$) which is thus updated ($upd\,i_C$). Note that when a place is both in the pre- and post-set of a transition (like *cache* for $upd\,i_C$) instead of an ingoing and an outgoing arrow, we draw a single double arrow. The presence or absence of a datum at the interface is represented by a token in place a or $\neg a$, respectively. The access to the cache by the two sensors via transitions $upd\,i_C$ is mutually exclusive (the cache stores a single measure), as guaranteed by the use of place *Free*, consumed by transitions $send_X$ and produced by $upd\,i_C$. The processor cyclically gets some value for the measure. If a value is exposed at the interfaces of the sensors (places a or b marked) then one of such values is taken ($poll\,i_P$), otherwise (places $\neg a$ and $\neg b$ marked) the cached value is read ($read_P$).

The security level of transitions is given by their subscript (namely, $\lambda(t_L) \mapsto L$). Transitions modeling the left and right sensors have security level A and B. The processor and the cache have security levels P and C, respectively. The intuition is that the two sensors should not interfere with each other, and they can send information to the processor directly or through the cache. The processor and the cache should not affect the behavior of the sensors.

In order to formalize the non-interference notions we will need some operations on net systems, specifically (parallel) composition and restriction [6].

Definition 4 (composition). *Let N and N' be two net systems such that $P \cap P' = \emptyset$ and for all $t \in T \cap T'$ it holds $\lambda t = \lambda' t$. The* composition *of N and N' is the net system $N|N' = (P \cup P', T \cup T', \lambda \cup \lambda', F \cup F')$. The composition of $\mathbf{N} = (N, m_0)$ and $\mathbf{N'} = (N', m_0')$ is the marked net system $\mathbf{N}|\mathbf{N'} = (N|N', m_0 \cup m_0')$.*

Definition 5 (restriction). *Given a net system N and a subset $T_1 \subseteq T$, the* restriction *of N by T_1 is the net system $N \setminus T_1 = (P, T - T_1, \lambda', F')$ where λ' and F' are the obvious restrictions of λ and F. For a marked net system $\mathbf{N}$, the restriction $\mathbf{N} \setminus T_1$ is $(N \setminus T_1, m_0)$.*

Intuitively $N|N'$ is the parallel composition of N and N', synchronized on the common transitions. Restriction simply removes the restricted transitions.

2.3 Unfolding semantics and related notions

The behavior of a Petri net can be represented by its unfolding $\mathcal{U}(\mathbf{N})$ [10], an acyclic net constructed inductively starting from the initial marking of $\mathbf{N}$ and then adding, at each step, an occurrence of each enabled transition of $\mathbf{N}$. In what follows we indicate by π_1 the projection over the first component of pairs.

Definition 6 (unfolding). *Let $\mathbf{N} = ((P, T, F), m_0)$ be a marked net. Define the net $U^{(0)} = (P^{(0)}, T^{(0)}, F^{(0)})$ as $T^{(0)} = \emptyset$, $P^{(0)} = \{(p, \bot) : p \in m_0\}$ and $F^{(0)} = \emptyset$, where $\bot$ is an element not belonging to P, T or F. The unfolding is the least net $\mathcal{U}(\mathbf{N}) = (P^{(\omega)}, T^{(\omega)}, F^{(\omega)})$ containing $U^{(0)}$ and such that*

- *if $t \in T$ and $X \subseteq P^{(\omega)}$ with X reachable, and $\pi_1(X) = {}^\bullet t$, then $(t, X) \in T^{(\omega)}$;*
- *for any $e = (t, X) \in T^{(\omega)}$, the set $Z = \{(p, e) : p \in \pi_1(e)^\bullet\} \subseteq P^{(\omega)}$; moreover ${}^\bullet e = X$ and $e^\bullet = Z$.*

Places and transitions in the unfolding represent tokens and firing of transitions, respectively, of the original net. Each place in the unfolding is a tuple recording the place in the original net and the "history" of the token. For historical reasons transitions and places in the unfolding are also called *events* and

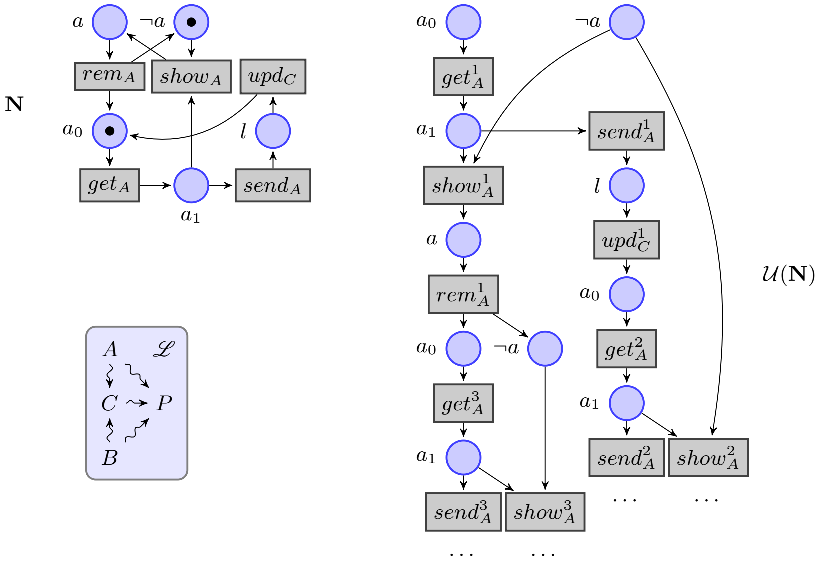

Fig. 2. A net system and the initial part of its unfolding.

conditions, respectively. The projection π_1 over the first component maps places and transitions of the unfolding to the corresponding items of the original net $\mathbf{N}$. The initial marking is implicitly identified as the set of minimal places.

As an example, consider the net system in Fig. 2 (top left), a slightly simplified version of the subnet of Fig. 1 modeling one of the sensors. A fragment of its unfolding is provided in Fig. 2(right). Conditions and events are labeled with the name of the corresponding place and transition in the original net. Different occurrences of a transition are distinguished using a numeric superscript. The conditions labeled by a_0 and $\neg a$ on the top, according to Definition 6, are $(a_0, \bot)$ and $(\neg a, \bot)$, respectively. Event get^1_A is $(get_A, \{(a_0, \bot)\})$ and the condition a_1 in its post-set is (a_1, get^1_A). Similarly, event $show^1_A$ is $(show_A, \{(a_1, get^1_A), (\neg a, \bot)\})$.

Definition 7 (causality, conflict). Causality $<$ *is the least transitive binary relation on* $P^{(\omega)} \cup T^{(\omega)}$ *such that* $x < y$ *if* $x \in {}^\bullet y$. *By* $\leq$ *we denote the reflexive closure of* $<$. Conflict *is the least symmetric binary relation* $\sharp$ *on* $P \cup T$ *such that if* $t, t' \in T$, $t \neq t'$ *and* ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$ *then* $t \sharp t'$ *and if* $x < x'$ *and* $x \sharp y$ *then* $x' \sharp y$.

In the running example, $get^1_A \leq show^1_A$ and $get^1_A \leq send^1_A$, while $send^1_A \sharp show^1_A$ and $show^1_A \sharp rem^1_A$.

The runs of $\mathbf{N}$ are represented by the configurations of $\mathcal{U}(\mathbf{N})$, i.e., subsets of $T^{(\omega)}$ that are causally closed and conflict-free. For a transition $t \in T^{(\omega)}$ we define its *causes* $[t] = \{t' \in T^{(\omega)} : t' \leq t\}$ and its *strict causes* $[t) = [t] - \{t\}$.

Definition 8 (configuration). *A* configuration *of $\mathcal{U}(\mathbf{N})$ is a finite subset $C \subseteq T^{(\omega)}$ such that $(C \times C) \cap \sharp = \emptyset$ and $[e] \subseteq C$ for all $e \in C$. The set of all configurations of $\mathcal{U}(\mathbf{N})$ is denoted by $\mathcal{C}(\mathcal{U}(\mathbf{N}))$.*

The transitions of a configuration C can be fired in any order compatible with causality, producing a marking called the frontier $C^\circ = (P^{(0)} \cup \bigcup_{t \in C} t^\bullet) - (\bigcup_{t \in C} {}^\bullet t)$; in turn, this corresponds to a marking of $\mathbf{N}$ given by $\mathsf{M}(C) = \pi_1(C^\circ)$. For instance, in Fig. 2, the set $\{get_A^1, show_A^1, rem_A^1\}$ is a configuration, while $\{show_A^1, rem_A^1\}$ and $\{get_A^1, show_A^1, rem_A^1, show_A^2\}$ are not since the first is not causally closed ($get_A^1 < show_A^1$) and the second has a conflict ($show_A^1 \# show_A^2$).

The unfolding has been shown to be marking complete in the sense that $m \in [m_0\rangle$ iff there exists $C \in \mathcal{C}(\mathcal{U}(\mathbf{N}))$ such that $\mathsf{M}(C) = m$ (see [10,18]).

3 Transitive Multilevel Non-Interference

In this section we focus on transitive multilevel security domains and we define the reference security property in the paper as an instance of (Bisimulation-based) Non-Deducibility on Composition (BNDC) [5].

3.1 Bisimilarity-based Non-Deducibility on Composition

Let $(\mathscr{L}, \rightsquigarrow)$ be a transitive MSD, fixed throughout the section. The definition of BNDC can be obtained by adapting that in [5,7] to the multilevel setting. First, in order to formalize the idea of variations of the behavior which are visible at a given security level we introduce a *view function* (or purge function [19]).

Definition 9 (view function). *Given a subset of the domain $S \subseteq \mathscr{L}$ and a net system N, the* view function *$S(\cdot) : T^* \to T_S^*$, is defined inductively by $S(\epsilon) = \epsilon$, $S(t\sigma') = tS(\sigma')$ if $\lambda(t) \in S$ and $S(t\sigma') = S(\sigma')$ otherwise.*

The view function filters out transitions whose level is not in S. It is used to define a bisimulation capturing the observation power of a user able to observe only events with security level in a given set.

Definition 10 (S-view bisimulation). *Let $\mathbf{N}$, $\mathbf{N}'$ be marked systems and $S \subseteq \mathscr{L}$. An S-*view simulation *of $\mathbf{N}$ by $\mathbf{N}'$ is a relation $R \subseteq [m_0\rangle \times [m_0'\rangle$ such that:*

- *$(m_0, m_0') \in R$;*
- *if $(m, m') \in R$ and $m[\sigma\rangle$ then there exists σ' such that $S(\sigma) = S(\sigma')$, $m'[\sigma'\rangle$ and $(\langle m[\sigma\rangle, \langle m'[\sigma'\rangle) \in R$.*

*An S-*view bisimulation *between $\mathbf{N}$ and $\mathbf{N}'$ is a relation $R \subseteq [m_0\rangle \times [m_0'\rangle$ such that R and R^{-1} are S-view simulations. If there exists an S-view bisimulation between $\mathbf{N}$ and $\mathbf{N}'$, we say that they are S-*view bisimilar *and write $\mathbf{N} \approx_S \mathbf{N}'$.*

In a two-level setting, i.e., in the domain $\{Low \rightsquigarrow High\}$, a system is non-interferent when the low level behavior is not influenced by high level interactions. Formally, a net system $\mathbf{N}$ is BNDC when $\mathbf{N} \approx_{Low} (\mathbf{N}|\mathbf{N}') \setminus (T_{High} - T')$ for any $\{High\}$-net $\mathbf{N}'$, i.e., the "low level" view of the behavior of $\mathbf{N}$ remains unchanged when the net interacts with any high level net system [6].

The generalization to the multilevel setting considers any partition of the security domain in an upper set $U \subseteq \mathscr{L}$ and its complement $\overline{U}$, and requires that U does not influence the view of $\overline{U}$.

Definition 11 (BNDC). *Let $\mathbf{N}$ be a marked net system. For an upper set $U \subseteq \mathscr{L}$, we say that $\mathbf{N}$ is U-BNDC if $\mathbf{N} \approx_{\overline{U}} (\mathbf{N}|\mathbf{N}') \setminus (T_U - T')$ for all marked U-systems $\mathbf{N}'$. The system is BNDC if it is U-BNDC for any upper set $U \subseteq \mathscr{L}$.*

The definition can be understood as follows. Given an upper set U, if the system is not U-BNDC then there is a flow from some level $L \in U$ to $L' \in \overline{U}$. This is a security violation since $L \not\rightsquigarrow L'$ otherwise L' would be in U. Vice versa, if there is a security violation, it will consist of a flow from some security level L to a level L' which cannot be influenced by L, namely $L \not\rightsquigarrow L'$. This is captured by the definition above when considering the upper set $U = \uparrow L$, since $L' \in \overline{U}$.

Note that the BNDC property for a multilevel domain reduces to the validity of BNDC in a number of two-level domains, one for each upper set, with U and its complement $\overline{U}$ playing the role of the high and low part of the system, respectively. Actually, as suggested by the considerations above, any security violation can be detected by analyzing upper sets of the kind $U = \uparrow L$ for $L \in \mathscr{L}$.

Proposition 1. *A net system $\mathbf{N}$ is BNDC iff $\mathbf{N}$ is $\uparrow L$-BNDC for every $L \in \mathscr{L}$.*

3.2 BNDC through Causal and Conflict Places

The characterization of BNDC based on causal and conflict places for the two-level case in [5,7], can be generalized to multilevel security domains. Roughly, a net system is BNDC when transitions with different security levels are never in conflict and there is no causal flow which is not allowed by the security policy.

Hereafter we focus on safe nets, which admit simpler and more effective notions of causal and conflict place (a weakening of those for general nets, whence the qualification "weak").

Notation. Given a net system $\mathbf{N}$ and a transition $t \in T$, we denote by $t^- = {}^\bullet t \setminus t^\bullet$ and, dually, $t^+ = t^\bullet \setminus {}^\bullet t$ the sets of places where the firing of t decrease and increase, respectively, the number of tokens.

Definition 12 (weak causal place). *A* weak causal place *in a net system $\mathbf{N}$ is any place $p \in {}^\bullet l \cap h^+$, for some $l, h \in T$ such that $\lambda h \not\rightsquigarrow \lambda l$, and some marking $m \in [m_0\rangle$ such that $m[h\tau l\rangle$, with $\tau \in T^*$.*

Intuitively, the firing sequence $h\tau l$ and the place $p \in {}^\bullet l \cap h^+$ witness a firing of l that depends on a token produced by the firing of h, representing an illegal flow from level λh to level λl. Conflict places are defined along the same lines.

Definition 13 (weak conflict place). *A* weak conflict place *in a net system* $\mathbf{N}$ *is any place* $p \in {}^{\bullet}l \cap h^{-}$*, for some* $l, h \in T$ *such that* $\lambda h \not\rightsquigarrow \lambda l$*, and some reachable marking* $m \in [m_0\rangle$ *such that* $m[h\rangle$ *and* $m[\tau l\rangle$*, with* $\tau \in T^{*}$*.*

The presence of weak causal or conflict places witnesses the failure of BNDC.

Theorem 1 (BNDC through weak causal/conflict places). *A safe net system* $\mathbf{N}$ *is BNDC iff* $\mathbf{N}$ *contains no weak causal nor weak conflict place.*

Consider the running example in Fig. 1. The system is not BNDC. In fact place a_0 is causal, as witnessed by the firing sequence $get_A\, send_A\, upd1_C\, get_A$, with $a_0 \in {upd1_C}^{+} \cap {}^{\bullet}get_A$ and $\lambda(upd1_C) = C \not\rightsquigarrow A = \lambda(get_A)$. Analogously, place b_0 is causal and place *Free*, is both causal and conflict. The interference seems unavoidable given that the cache is accessed in mutual exclusion and a value sent to the cache must determine an update. In § 4 we will show how these occurrences of interference can be amended with the use of intransitive policies.

3.3 Non-Interference in the Unfolding

Occurrences of causal and conflict places in the unfolding of safe net systems can be given a structural characterization, which, thanks to Theorem 1, leads to a unfolding-based characterization of the BNDC property.

Notation. For a condition b and an event t in the unfolding $\mathcal{U}(\mathbf{N})$ we set $t^{+} = \{b \in P^{(\omega)} : \pi_1(b) \in \pi_1(t)^{+}\}$ and $t^{-} = \{b \in P^{(\omega)} : \pi_1(b) \in \pi_1(t)^{-}\}$.

Proposition 2 (BNDC in the unfolding). *A safe net system* $\mathbf{N}$ *is not BNDC iff there are events* h', l' *such that* $\lambda h' \not\rightsquigarrow \lambda l'$ *and a condition* b *in* $\mathcal{U}(\mathbf{N})$ *such that either (i)* $b \in {}^{\bullet}l' \cap h'^{+}$ *or (ii)* $b \in {}^{\bullet}l' \cap h'^{-}$ *and* $[h') \cup [l'] \in \mathcal{C}(\mathcal{U}(\mathbf{N}))$*.*

Note that condition (ii) is harder to check than (i), as it involves an exploration of the history of the interacting transitions. In the verification procedure it is convenient to look only for causal interference. This can be done, thanks to the fact that for safe nets all occurrences of interference can be reduced to causal ones. We omit the details which largely overlap with those for the two-level case [7]. We only remark that the causal reduction causes an expansion of the size of the net that is at most quadratic in the number of transitions.

Proposition 3 (BNDC in the causal reduct). *Let* $\mathbf{N}$ *be a safe net system. It is possible to build a safe net* $\gamma(\mathbf{N})$*, called* causal reduct *of* $\mathbf{N}$*, such that* $\mathbf{N}$ *is BNDC iff* $\gamma(\mathbf{N})$ *has no weak causal places.*

3.4 Unfolding-based Algorithm for BNDC

The unfolding of a net can be infinite (when it includes a cycle). Starting with [18] techniques have been developed for efficiently constructing finite prefixes of the unfolding which are complete with respect to properties of interest [20].

Here, as a first step, we identify a completeness criterion ensuring that an unfolding prefix includes at least a representative for a causal interference, when a net system is not BNDC. This is used for developing an algorithm for checking BNDC for a safe net. Interestingly, while Definition 11 reduces multilevel noninterference to a number of checks in a two-level setting, here the verification is performed by constructing a single unfolding prefix.

As discussed in the two-level case [7], a prefix complete for reachability could omit information relevant for interference. In order to capture all occurrences of interference, in the two-level case, markings were enriched by recording which tokens were generated by high transitions. Here we record the level of transitions generating the tokens, adapting the notion of completeness accordingly.

Definition 14 (c-marking, c-complete prefix). *Let* $\mathbf{N}$ *be a safe net system and let* $C \in \mathcal{C}(\mathcal{U}(\mathbf{N}))$. *The* confidentiality marking (c-marking) *of* C *is* $\mathsf{M}^*(C) = \langle \mathsf{M}(C), \Lambda_C \rangle$, *where* $\Lambda_C : \mathsf{M}(C) \to \mathscr{L}$ *is a partial function defined as follows. For any* $b \in C^\circ$, *if* ${}^\bullet b = \{t'\}$ *then* $\Lambda_C(\pi_1(b)) = \lambda t'$, *otherwise, if* ${}^\bullet b = \emptyset$ *then* $\Lambda_C(\pi_1(b))$ *is undefined. A prefix* U *of* $\mathcal{U}(\mathbf{N})$ *is* complete for c-marking reachability, *or simply* c-complete, *when for any configuration* $C \in \mathcal{C}(\mathcal{U}(\mathbf{N}))$ *there exists* $C' \in \mathcal{C}(U)$ *such that* $\mathsf{M}^*(C) = \mathsf{M}^*(C')$.

In words, Λ_C maps each marked place to the level of the transition that generated the corresponding token. It is undefined on tokens of the initial marking.

When checking BNDC on a complete prefix U, we need to consider also events at the “border” of U, i.e., events that are enabled by configurations of U and which could be could be added by a further unfolding step. In the procedure for generating the prefix these transitions will be added and marked as cut-offs. The prefix obtained from U by adding such transitions is denoted $U^{\rhd}$.

We can now show that a c-complete prefix U of $\mathcal{U}(\mathbf{N})$, includes sufficient information for deciding whether or not $\mathbf{N}$ contains a weak causal place.

Theorem 2 (weak causal places in c-complete prefixes). *Let* $\mathbf{N}$ *be a safe net system and let* U *be a c-complete prefix of* $\mathcal{U}(\mathbf{N})$. *Then* p *is a weak causal place in* $\mathbf{N}$ *iff there exists in* $U^{\rhd}$ *a condition* b *and events* h', l' *such that* $\pi_1(b) = p$, $b \in {}^\bullet l' \cap h'^+$ *and* $\lambda h' \not\rightsquigarrow \lambda l'$.

The above result and Proposition 3 implies that, given a safe net system, one can check for BNDC on a c-complete prefix of the unfolding of its causal reduct.

Corollary 1 (BNDC on c-complete prefixes). *Let* $\mathbf{N}$ *be a safe net system and let* U *be a c-complete prefix of* $\mathcal{U}(\gamma(\mathbf{N}))$. *Then* $\mathbf{N}$ *is not BNDC iff there exist events* $h', l' \in U^{\rhd}$ *such that* $\lambda h' \not\rightsquigarrow \lambda l'$ *and* ${}^\bullet l' \cap h'^+ \neq \emptyset$.

Corollary 1 leads to an algorithm for checking BNDC on safe net systems. Given $\mathbf{N}$ first it computes its causal reduct $\gamma(\mathbf{N})$. Then it builds a c-complete prefix of the unfolding $\mathcal{U}(\gamma(\mathbf{N}))$ by adding, at each step, a transition occurrence and checking if its direct causalities satisfy the conditions in Corollary 1.

Corollary 2 (correctness of the algorithm for BNDC). *Let* $\mathbf{N}$ *be a safe net system. The algorithm outlined above always terminates and answers 'yes' iff* $\mathbf{N}$ *is BNDC.*

4 Intransitive Multilevel Non-Interference

In this section we focus on intransitive policies. The idea is that some information flows between levels that cannot communicate directly become allowed if they are mediated by a chain of trusted intermediaries.

4.1 Bisimilarity-based Intransitive Non-Interference

Inspired by the idea of separability in [19], in order to check whether there are illegal flows from a set of levels U, we artificially isolate that set by removing from the system all of its legal targets in $\twoheaduparrow U$. If, afterwards, the levels in U can still influence other levels in the rest of the system, the influence is certainly illegal. In fact, it cannot be mediated by a chain of legal intermediaries since any such chain has been certainly broken by the construction. This leads to a multilevel generalization of BINI (Bisimulation-based Intransitive Non-Interference) [6].

Definition 15 (BINI). *Given* $U \subseteq \mathscr{L}$*, a net system* $\mathbf{N}$ *is* U*-BINI if for all reachable markings* $m \in [m_0\rangle$ *the system* $(N \setminus T_{\twoheaduparrow U}, m)$ *is* U*-BNDC in the domain* $\mathscr{L}' = (\mathscr{L} \setminus \twoheaduparrow U, \rightsquigarrow^*)$*. The system* $\mathbf{N}$ *is BINI if it is* U*-BINI for all* $U \subseteq \mathscr{L}$*.*

As explained above, for each set of levels U we consider the net $N \setminus T_{\twoheaduparrow U}$, obtained by pruning the transitions with level in $\twoheaduparrow U$, to which a flow from U is admitted. The presence of an illegal flow from U is thus reduced to the presence of any flow from U in the pruned subsystem. In turn, the presence of a flow from U is formalized by resorting to the notion of BNDC previously introduced (Definition 11). It is easy to see that the definition is well-given, i.e., U is an upper set in $\mathscr{L}' = (\mathscr{L} \setminus \twoheaduparrow U, \rightsquigarrow^*)$. Note that an illegal flow from U could occur at any reachable marking m of the original system, but clearly the pruning operation can make m unreachable. This is the reason why the pruned net $N \setminus T_{\twoheaduparrow U}$ is checked with any marking reachable in the original net system $\mathbf{N}$.

Consider the running example in Fig. 1, which is not BNDC due to an interference between the cache and the sensors, and between the sensors themselves. In both cases the interference stem out from the mutually exclusive access to the cache. If this mode of access is an hardware constraint, it might be the case that the designer intends to ignore such occurrences of interference, deeming them inevitable and not problematic. This can be modeled by adding a number of "downgrading" levels to the domain, and modifying the net adding downgrading transitions. In Fig. 3 we show how this can be done in order to make the old net BINI (we only show a part of the system: the processor is unchanged

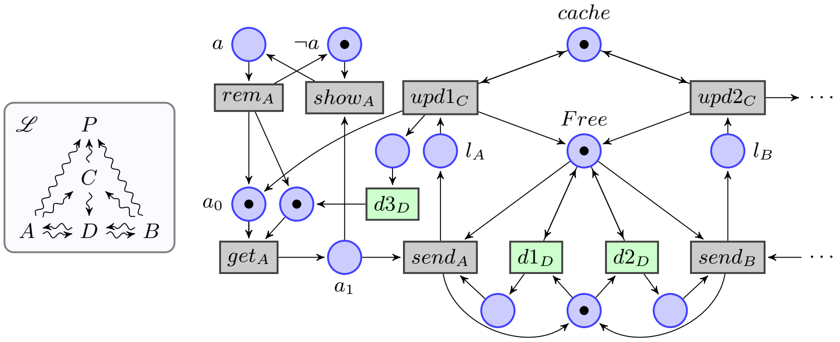

Fig. 3. A fix for the sensor net that makes it BINI. Only part of the system is shown, and as usual $\lambda(x_L) = L$. The downgrading transitions are highlighted in green.

and the second sensor is symmetric to the first one). Note, e.g., that $C \not\rightsquigarrow A$ but transition $upd1_C$ can obviously influence get_A, since we can have a causal chain $upd1_C \; d3_D \; get_A$. However, this is not a violation of BINI because the interference occurs through $d3_D$, which is a legitimate intermediary ($C \rightsquigarrow D \rightsquigarrow A$). More formally, if we take $U = \{C\}$, according to Definition 15, we have to consider the net $N \setminus T_{\Uparrow U}$, where legal intermediaries for C, namely transitions with level in $\Uparrow\{C\} = \{D, P\}$ are pruned. In particular, the pruned net does not include transition $d3_D$ and thus the interference of $upd1_C$ on get_A is correctly hidden. Similarly, transitions $d1_D$ and $d2_D$ mediate the conflict between $send_A$ and $send_B$.

Although not immediate, as a sanity check, it can be proved that BINI and BNDC coincide on transitive domains.

Proposition 4 (BINI is BNDC on transitive domains). *In a transitive MSD $\mathscr{L}$, a net system $\mathbf{N}$ is BINI if and only if $\mathbf{N}$ is BNDC.*

Additionally, BINI can be characterized by replacing the quantification over all subsets $U \subseteq \mathscr{L}$ of Definition 15 with a quantification over single levels.

Proposition 5 (multilevel BINI on single levels). *A net system $\mathbf{N}$ is BINI iff $\mathbf{N}$ is $\{L\}$-BINI for each $L \in \mathscr{L}$.*

4.2 BINI through Causal and Conflict Places

A characterization of BINI amenable of effective verification in the unfolding of safe nets, relies on intransitive variants of weak causal and conflict places.

Definition 16 (intransitive weak causal/conflict place). *Let $\mathbf{N}$ be a safe net system. An* intransitive weak causal place *is $p \in {}^{\bullet}l \cap h^{+}$, for $l, h \in T$ such that $\lambda h \not\rightsquigarrow \lambda l$, and there is a reachable $m \in [m_0\rangle$ such that $m[h\tau l\rangle$, with $\tau \in T^{*}_{\overline{\Uparrow \lambda h}}$.*

An intransitive weak conflict place *is* $p \in {}^{\bullet}l \cap h^{-}$, *for* $l, h \in t$ *such that* $\lambda h \not\rightsquigarrow \lambda l$, *and there is a reachable* $m \in [m_0\rangle$ *such that* $m[h\rangle$ *and* $m[\tau l\rangle$, *with* $\tau \in t^{*}_{\overline{\twoheaduparrow \lambda h}}$.

The difference with respect to the notions of weak causal and conflict place in § 3.2 for transitive policies is that here τ is required not to contain any transition to which information can could legally flow from h. Intuitively, the reason is that, otherwise, the flow from h to l would be mediated by such transition, possibly amending the violation represented by p. As an example, in Fig. 3 place *Free* is not an intransitive conflict place, despite the fact that $\mathit{Free} \in {}^{\bullet}\mathit{send}_A \cap \mathit{send}_B{}^{-}$ and $B \not\rightsquigarrow A$. The reason is that, in any firing sequence starting from place *Free* marked, an occurrence of send_A is necessarily preceded by $d1_D$.

Theorem 3 (BINI through intransitive weak places). *A safe net system* $\mathbf{N}$ *is BINI iff it contains no intransitive weak causal or conflict place.*

4.3 BINI in the Unfolding

Occurrences of intransitive weak causal places can be characterized in the unfolding of safe nets.

Theorem 4 (intransitive weak causal places in the unfolding). *Let* $\mathbf{N}$ *be a safe net system. A place* p *in* $\mathbf{N}$ *is an intransitive weak causal place iff there exists a condition* b *in* $\mathcal{U}(\mathbf{N})$ *such that* $\pi_1(b) = p$ *and there are events* h', l' *such that (i)* $b \in {}^{\bullet}l' \cap h'^{+}$ *and (ii)* $\forall t' : h' < t' \leq l' \,.\, \lambda h' \not\rightsquigarrow \lambda t'$.

The above, together with the possibility of resorting, as in the intransitive case, to the causal reduct, leads to the following characterization of BINI.

Proposition 6 (BINI in the causal reduct). *Let* $\mathbf{N}$ *be a net system. Then* $\mathbf{N}$ *is BINI iff the causal reduct* $\gamma(\mathbf{N})$ *contains no intransitive causal places.*

For building a complete prefix, we still need to enrich the marking associated with a configuration C with a function Λ_C, mapping each token to the security level of the generating transition. However, due to the intransitivity of the policy, this is no longer sufficient to detect a violation. In fact, assume that an event l of level L consumes a token of level H such that $H \not\rightsquigarrow L$. Apparently this is a violation of the policy since the presence of a token of level H reveals that an event, say h, of the same level has been executed before, and this fact is visible at level L. However, this might not be a problem, since it could be that a token of a level D such that $H \rightsquigarrow D \rightsquigarrow L$, is also in the pre-set of l, produced by an event d such that $h < d < l$. In this case, the flow of information from L to H is legitimately mediated by D. Roughly, we can think that the token of level D absorbs the token of level H to its level. We then enrich the markings with an *absorbing relation* δ over the conditions in the frontier of a configuration.

Definition 17 (i-marking, i-complete prefix). *Let* $\mathbf{N}$ *be a safe net system and let* $C \in \mathcal{C}(\mathcal{U}(\mathbf{N}))$. *The* intransitive confidentiality marking (i-marking) *of* C *is* $\mathsf{M}^{*}_{i}(C) = \langle \mathsf{M}(C), \Lambda_C, \delta_C \rangle$, *where* $\Lambda_C : P \to \mathscr{L}$ *is as in Definition 14 and* $\delta_C : \pi_1(C^{\circ}) \times \pi_1(C^{\circ})$ *is the relation:*

$$\{(\pi_1(p), \pi_1(q)) \mid \exists t, t' \in C \,.\, q \in t'^{\bullet} \,\wedge\, \lambda(t') \rightsquigarrow \lambda t \,\wedge\, t' < t \leq p\}$$

A prefix U of $\mathcal{U}(\mathbf{N})$ is complete for i-marking reachability *(*i-complete*), when for any configuration $C \in \mathcal{C}(\mathcal{U}(\mathbf{N}))$ there is $C' \in \mathcal{C}(U)$ such that $\mathsf{M}_i^*(C) = \mathsf{M}_i^*(C')$.*

Intuitively, whenever $\delta(p, q)$ the token in p absorbs the token in q to its level, if they are used in the same pre-set.

It can be proved that an i-complete prefix U of $\mathcal{U}(\mathbf{N})$ includes sufficient information for deciding whether $\mathbf{N}$ contains a weak intransitive causal place. This fact, with Theorem 4 and Proposition 6, implies that one can check BINI for a net system on an i-complete prefix of the unfolding of its causal reduct.

Corollary 3 (BINI on i-complete prefixes). *Let $\mathbf{N}$ be a safe net system and let U be a i-complete prefix of $\mathcal{U}(\gamma(\mathbf{N}))$. Then $\mathbf{N}$ is not BINI iff there exists in $U^{\rhd}$ a condition b and events h', l' such that $b \in {}^{\bullet}l' \cap h'^{+}$, $\lambda h' \not\rightsquigarrow \lambda l'$, and furthermore $\forall b' \in {}^{\bullet}l' \,.\, \neg(b' \delta_{[l')} b)$.*

In words, an interference is witnessed by an event l' that uses a token b of a non accessible level such that b is not absorbed. As in the transitive case, this result is used for designing an algorithm that checks BINI on safe net systems.

5 The tool MultiUBIC

The unfolding-based algorithms outlined in the previous sections are implemented in MultiUBIC [14]. It extends a previous tool UBIC, which was limited to two level security domains (possibly with downgrading). MultiUBIC inputs a security policy (transitive or intransitive) and a safe net system, and it checks whether BNDC (transitive policies) or BINI (intransitive policies) is satisfied.

Compared to PNSC [21] and ANICA [22], "interleaving competitors" based on the work [5], MultiUBIC inherits the good performance of its ancestor UBIC: the use of a partial order semantics leads to a gain of efficiency especially for highly concurrent systems, where the state explosion problem is more serious.

The verification of multi-level security policies can be reduced to a number of checks in a two-level setting (possibly with downgrading, in the intransitive case). MultiUBIC comes equipped with facilities for performing such reduction. The definition of BNDC suggests that such reduction can be expensive, since the two-levels problems arise from partitions of the security domain whose number can be exponential in the number of levels. For net systems it can be actually shown that we can limit to a linear number of two-level checks, one for each level (see Proposition 1 for the transitive case and Proposition 5 for the intransitive case). Still, some preliminary experiments reveal that solving directly the original multi-level problem, typically provides a linear gain of efficiency at the price of an increase of memory usage. The performances of MultiUBIC can degrade for net systems where a relevant number of places have input transitions of different levels, a fact that potentially causes an exponential blow of the number of enriched markings. A precise characterization of this pathological situations is under investigation. Due to space limitations, a presentation of the experimental results and a more extensive discussion are deferred to the full version.

6 Conclusions

We studied non-interference in a multilevel setting, for transitive and intransitive security domains, focusing on Petri nets. Generalizing [7,11], we showed that Bisimilarity-based Non-Deducibility on Composition (BNDC) and its intransitive extension BINI [6], admit a causal characterizations in the unfolding of safe net systems. This led to verification algorithms for BNDC and BINI on safe net systems with multilevel policies, implemented in the tool MultiUBIC.

Causal semantics have been used in [23] for deducing the occurrence of non-observable transitions in the diagnosis of discrete event systems. There is a clear conceptual relation relation between diagnosability properties and non-interference, despite the fact that the former are trace-based while our non-interference is bisimulation-based. The work on intransitive non-interference in [24], that relies on automata models and language theory could be helpful for establishing a formal relation.

In the setting of Petri nets other classes of information flow properties have been studied, like opacity properties [25] (which include non-interference) and selective non-interference [26]. Exploring the use of causal semantics in this general setting appears as an interesting and challenging venue of future research.

A huge literature exists on non-interference for various formalisms, including process calculi and imperative languages (see, e.g., [2,27] for surveys). Fruitful connections could emerge investigating a causal characterizations of non-interference in these settings, possibly through encodings into Petri nets.

We also plan to consider formalizations of non-interference obtained from the classical ones, by replacing interleaving observational semantics with true-concurrent ones [28]. The higher distinguishing power of such semantics could allow to identify new forms of interference which cannot be captured in an interleaving setting. Interesting reflections in this directions are reported in [29].

References

1. Goguen, J.A., Meseguer, J.: Security policies and security models. In: Proceedings of the Symposium on Security and Privacy. IEEE Computer Society (1982) 11–20
2. Focardi, R., Gorrieri, R.: Classification of security properties (Part I: Information flow). In: Proceedings of FOSAD'00. Springer (2001) 331–396
3. Ryan, P., Schneider, Y.: Process algebra and non-interference. Journal of Computer Security **9**(1/2) (2001) 75–103
4. Mantel, H.: Possibilistic definitions of security - an assembly kit. In: Proceedings of CSFW'00. IEEE Computer Society (2000) 185–199
5. Busi, N., Gorrieri, R.: Structural non-interference in elementary and trace nets. Mathematical Structures in Computer Science **19**(6) (2009) 1065–1090
6. Best, E., Darondeau, P., Gorrieri, R.: On the decidability of non interference over unbounded Petri nets. In Chatzikokolakis, K., Cortier, V., eds.: Proceedings of SecCo'10. Volume 51 of EPTCS. Open Publishing Association (2010) 16–33
7. Baldan, P., Carraro, A.: A causal view on non-intereference. Fundamenta Informaticae **140**(1) (2015) 1–38

8. McCullough, D.: Noninterference and the composability of security properties. In: Symposium on Security and Privacy. IEEE Computer Society (1988) 178–186
9. Wittbold, J., Johnson, D.: Information flow in nondeterministic systems. In: Symposium on Security and Privacy. IEEE Computer Society (1990) 148–161
10. Nielsen, M., Plotkin, G., Winskel, G.: Petri nets, event structures and domains, part 1. Theoretical Computer Science **13** (1981) 85–108
11. Baldan, P., Burato, F., Carraro, A.: Intransitive non-interference by unfolding. In Lanese, I., Madelaine, E., eds.: Proceedings of FACS'14. Volume 8997 of LNCS. Springer (2014) 269–287
12. Denning, D.E.: A lattice model of secure information flow. Communication of the ACM **19**(5) (1976) 236–243
13. Rushby, J.M.: Design and verification of secure systems. In: Proceedings of SOSP'81. ACM (1981) 12–21
14. Beggiato, A.: MultiUBIC. `https://github.com/AlessandroBeggiato/MultiUbic/releases`
15. Service Technology: ANICA: Automated Non-Interference Check Assistant. `http://service-technology.org/anica`
16. Gorrieri, R., Vernali, M.: On intransitive non-interference in some models of concurrency. In Aldini, A., Gorrieri, R., eds.: Proceedings of FOSAD'11. Volume 6858 of LNCS. Springer (2011) 125–151
17. Esparza, J., Heljanko, K.: Unfoldings - A Partial order Approach to Model Checking. EACTS Monographs in Theoretical Computer Science. Springer (2008)
18. McMillan, K.L.: A technique of state space search based on unfolding. Form. Methods Syst. Des. **6**(1) (1995) 45–65
19. Rushby, J.: Noninterference, transitivity, and channel-control security policies. Technical report (Dec 1992)
20. Khomenko, V., Koutny, M., Vogler, W.: Canonical prefixes of Petri net unfoldings. Acta Informatica **40** (2003) 95–118
21. Frau, S., Gorrieri, R., Ferigato, C.: Petri net security checker: Structural non-interference at work. In Degano, P., Guttman, J. Martinelli, F., eds.: Proceedings of FAST'08. Volume 5491 of LNCS. Springer (2008) 210–225
22. Accorsi, R., Lehmann, A.: Automatic information flow analysis of business process models. In Barros, A., Gal, A., Kindler, E., eds.: Proceedings of BPM'12. Volume 7481 of LNCS. Springer (2012) 172–187
23. Haar, S.: Types of asynchronous diagnosability and the reveals-relation in occurrence nets. IEEE Transactions on Automatic Control **55**(10) (2010) 2310–2320
24. Hadj-Alouane, B.N., Lafrance, S., Lin, F., Mullins, J., Yeddes, M.M.: On the verification of intransitive noninterference in multilevel security. IEEE Transactions on Systems, Man, and Cybernetics, Part B **35**(5) (2005) 948–958
25. Bryans, J., Koutny, M., Ryan, P.: Modelling dynamic opacity using Petri nets with silent actions. In Dimitrakos, T., Martinelli, F., eds.: Proceedings of FAST'05. Volume 173 of LNCS. Springer (2005) 159–172
26. Best, E., Darondeau, P.: Deciding selective declassification of Petri nets. In: POST'12. Volume 7215 of LNCS. Springer (2012) 290–308
27. Mantel, H., Sands, D.: Controlled declassification based on intransitive noninterference. In: APLAS'04. (2004) 129–145
28. van Glabbeek, R., Goltz, U.: Refinement of actions and equivalence notions for concurrent systems. Acta Informatica **37**(4/5) (2001) 229–327
29. Fröschle, S.: Causality, behavioural equivalences, and the security of cyberphysical systems. In Meyer, R., Platzer, A., Wehrheim, H., eds.: Correct System Design. Volume 9360 of LNCS. Springer (2015) 83–98