Petri nets and dynamic causality for service-oriented computations *

Giovanni Casu
Dipartimento di Matematica e Informatica,
Università di Cagliari, Via Ospedale 72
09124 Cagliari, Italy
giovanni.casu@gmail.com

G. Michele Pinna
Università di Cagliari, Cagliari, Italy,
Università di Cagliari, Via Ospedale 72
09124 Cagliari, Italy
gmpinna@unica.it

ABSTRACT

When dealing with service oriented computations the dependencies among the various distributed activities may be complex and difficult to represent *statically*. Recently Event Structures where the causality may change dynamically have been introduced and have been related with many other kind of Event Structures (with a particular focus on expressivity). In this paper we relate them to a kind of (labeled) Petri net which turns out to cover these new Event Structures. This relationship empowers the usage of all the available verification tools based on Petri nets, giving practical and usable means for the verification of the complex and distributed system whose behavior is modeled by this kind of event structures.

CCS Concepts

•Theory of computation → Distributed computing models; •Software and its engineering → Petri nets;

Keywords

Petri Nets; Contextual Petri Nets; Dynamic Event Structures

1. INTRODUCTION

Service oriented computations usually show a pattern of dependencies that may be difficult to represent with the usual notions of dependencies available for more static distributed scenarios, where the notions of event structures proposed in literature may suffice. To substantiate this claim we recall the relations among contract and event structures, leading to a notion of event structure with *circular causality* investigated in [6], or the notion of *lending nets* presented in [5] where circular causality is related to the possibility of executing a transition lending some tokens.

*This work has been partially supported by Aut. Reg. of Sardinia P.I.A. 2013 "NOMAD"

SAC 2017,April 03-07, 2017, Marrakech, Morocco

http://dx.doi.org/10.1145/3019612.3019806

Event structures are one of the main ways to represent computations in service oriented computations, thus the quest for event structures where the new kind of dependencies may be easily represented.

Consider a *seller* behaving as follows. When he receives a request for delivering a good, he sends a confirmation if the good is immediately available, but if it is not (hence the `not-available` event happens) he sends a confirmation only after being sure that the good is available in some other places and he may eventually get it. Furthermore he delivers the good only after receiving the payment unless the one requesting the good is a premium user. In this scenario the event `confirm` may happen after the event `request` has happened unless the `not-available` event has happened. In this case the event `confirm` *depends* not only from `request`, but also from the event `available` signalling that the good is somewhere available. The dependency pattern of `confirm` may *augment* in presence of a certain event (in this case `not-available`). The event `deliver` depends on the event `pay`, but if the event `premUser` is present then this dependency is dropped.

Event structures have generally a relation to represent the *dependencies* among events and another relation to stipulate which events cannot happen in the same computation under certain conditions. A computation is then a subset of events such that there is no pair of events in conflict and each event is such that all of its causes are contained in the configuration. The dependencies are often represented as a partial order or a relation that should be a partial order when confined to a configuration, like in flow event structures. At the beginning of the '90s the idea that a partial order was the unique way to represent dependencies has been somehow abandoned. *Bundle event structures* [13] have been introduced to give semantics to LOTOS [7] and may model *or*-causality and their generalization *dual event structures* [12] gain expressivity dropping the assumption that *or*-causality implies that either one or the other cause happens, but not both. These event structures have a more operational flavour with respect to *prime event structures* [18] *flow event structures* [8], *asymmetric event structures* [4] or *inhibitor event structures* [3], as the possibility of adding an event is prescribed by bundles that may be sets of events of any kind. This change in perspective has been driven by the observation that the same event (observable activity producing observable changes) could have totally different histories (pasts) and from these histories was not possible

to find a common pattern (see, for instance, the *possible events* in the *event automata* [16] or the notion of events in *causal automata*), and it has been further pursued with the notions of asymmetric and inhibitor event structures. However either these phenomena are due to the presence of *contexts* or *inhibitions*, that may be added or removed by the happening of events, or they are represented in a logical way, like in [11] or [17]. Event structure with *dynamic* causality are introduced in [1] by stipulating that the happening of certain events, called modifiers, may add or remove causal dependencies for a certain event, so that the phenomena of *shrinking* or *growing* causality may be captured. They compare their new notion of event structure to many other presented in literature with respect to their expressiveness.

This brand of event structures seems an interesting candidate to model some of the phenomena arising in a service oriented scenario. Still these event structures alone are of little use, and establishing a connection with Petri nets may ease the verification of properties of the computations modelled by these event structures.

The relationship among many kind event structures and suitable Petri nets have been extensively studied. In fact *prime event structures* and *causal nets* are strongly related, as shown in [15], where the relation between these two formalism has been investigated for the first time. Since then many authors have considered suitable classes of nets and have related them to suitable notions of event structures. We may recall, among others, the *flow nets* [8] and flow event structures, *contextual nets* [14] and asymmetric event structures, *inhibitor nets* and inhibitor event structure or the *1-occurrence net* and their corresponding notion of *event* and *configuration structures* [17]. The intuition behind all these approaches has always followed a common pattern: to each transition of the net it should be possible to associate an event in the corresponding event structure and from the net structure the relations among events may be deduced.

In many of the above mentioned approaches, with the exception of the case of 1-occurrence nets, nets are such that each execution gives all the relevant information about the dependencies in that execution. For instance, in flow nets it is required that each place is used only once in each computation, thus the dependency signalled by that place can be dug out, whereas inhibitor nets, when inhibitor and contextual arcs are forgotten, causal nets, and the (set of possible) execution(s) compatible with the usual dependencies can be again inferred.

In this paper we give a Petri nets counterpart to the notions of shrinking and growing causality that, as the example shows, can be one of the relevant ingredients for service oriented computations. The authors in [1] state that in the usual notion of event structures the dependencies have a *static* flavour, whereas, in the case of shrinking and growing causality, the dependencies to be dropped or added give a more *dynamic* account. However in the shrinking and growing causality relations it is somehow hardwired which dependencies are involved. Thus an event may have several *histories*, depending on the combination of modifiers that have happened so far. Our idea is rather simple. We introduce a transition (with the same label) for each possible history (similarly to what happens when considering labelled causal nets like the notion of *unfolding* [18] or *branching processes* [9] and we use contextual arcs to understand which is the set of modifiers that have happened so far. The kind of nets we devise should still have some characteristics similar to the ones of causal nets, hence we require that the restriction of the net to a specific subset of transitions (a computation) gives an acyclic net, *i.e.* a net where dependencies are easily understandable and conflicts are absent. We call this notion *unravel net*. The notion of unravel net with contextual arcs turns out to be powerful enough to represent shrinking and dynamic causality, but it should be stressed that we are actually giving an implementation on how the executions of the event structures may be performed. For this reason we focus on traces rather than on configurations.

By exploiting the relationship between dynamic event structures and labeled unravel nets with contextual arcs where the contextual arcs are not inhibitor ones, we remain in the classes of nets where many properties can be easily and efficiently verified.

2. PRELIMINARIES

We recall notations and notions we use in the paper. With $\mathbb{N}$ we denote the set of natural numbers. Let A be a set, a *multiset* of A is a function $m : A \to \mathbb{N}$ and the set of multisets of A is denoted by μA. The operations and relations on multisets, like inclusion ($\subseteq$), union ($+$) or difference ($-$), are the standard ones. If $m \in \mu A$, we denote by $[\![m]\!]$ the multiset defined as $[\![m]\!](a) = 1$ if $m(a) > 0$ and $[\![m]\!](a) = 0$ otherwise; and we use $[\![m]\!]$ also for denoting subset $\{a \in A \mid m(a) \geq 1\}$ of A. When a multiset m coincides with $[\![m]\!]$ then it is a set, and we will often confuse the multi set m with $[\![m]\!]$. The empty multiset (*i.e.* $[\![m]\!] = \emptyset$) is denoted with $\emptyset$.

Nets.

We recall the main definitions about labeled Petri nets with contextual arcs. We fix a set of *labels* L, that are used to label the transitions of a net. A *Petri net* with contextual arcs is a 5-tuple $N = \langle S, T, F, C, \mathsf{m} \rangle$, where S is a set of *places* and T is a set of *transitions* (with $S \cap T = \emptyset$), $F \subseteq (S \times T) \cup (T \times S)$ is the *flow* relation, $C \subseteq (T \times S)$ is the *context* relation and $\mathsf{m} \in \mu S$ is called the *initial marking*. We require that $C \cap F = \emptyset$, *i.e* if $(t, s) \in C$ then $(t, s) \notin F$ and if $(t, s) \in F$ then $(t, s) \notin C$. Petri nets are depicted as usual: places are circles, transitions are boxes, the flow relation is represented by arcs from x to y whenever $(x, y) \in F$ and the read arcs are depicted as a straight lines.

DEFINITION 1. *A* labeled *Petri net over* L *is the pair* $\mathsf{N} = (N, l)$ *where* $N = \langle S, T, F, C, \mathsf{m} \rangle$ *is a Petri net with contextual arcs and* $l \colon T \to \mathsf{L}$ *is a total labeling mapping.*

We consider only labeled nets, we write $\langle S, T, F, C, \mathsf{m}, l \rangle$ instead of $(\langle S, T, F, C, \mathsf{m} \rangle, l)$, and often confuse N with N. Given a net $N = \langle S, T, F, R, \mathsf{m}, l \rangle$ and $x \in S \cup T$, we define the following multisets (on S if $x \in T$ and on T is $x \in S$): ${}^\bullet x = F(-, x)$ and $x^\bullet = F(x, -)$. Given a transition t, its *context* is $\underline{t} = \{s \in S \mid (t, s) \in C\}$. Also $\underline{t}$ may be seen as a multiset on S: $\underline{t}(s) = 1$ whenever $(t, s) \in C$ and $\underline{t}(s) = 0$ otherwise. A transition $t \in T$ is enabled at a marking

$m \in \mu S$, denoted with $m\,[t\rangle$, whenever ${}^\bullet t \subseteq m$ and $\underline{t} \subseteq m$. A transition t enabled at a marking m can *fire* and its firing produces the marking $m' = m - {}^\bullet t + t^\bullet$. The firing of a transitions t at a marking m is denoted with $m\,[t\rangle\, m'$. We stress that the context is used only to check whether or not a transition is enabled at a given marking. We assume that each transition t of a net N is such that ${}^\bullet t + \underline{t} \neq \emptyset$ (which means that no transition may fire *spontaneously*).

Given a marking m, the *firing sequence* (fs) starting at m of the net $N = \langle S, T, F, C, \mathsf{m}, l\rangle$, is defined as usual: (a) m is a fs, and (b) if $m\,[t_1\rangle\, m_1 \cdots\ m_{n-1}\,[t_n\rangle\, m_n$ is a fs and $m_n\,[t\rangle\, m'$ then also $m\,[t_1\rangle\, m_1 \cdots m_{n-1}\,[t_n\rangle\, m_n\,[t\rangle\, m'$ is a fs. The set of firing sequences of a net N, starting at a marking m, is denoted with $\mathcal{R}^N_m$ and it is ranged over by σ, and we omit the index denoting the net when it is clear from the context. Given $\mathsf{fs}\ \sigma = m\,[t_1\rangle\, \sigma'\,[t_n\rangle\, m_n$, with $start(\sigma)$ we denote the marking m, with $lead(\sigma)$ the marking m_n and with $tail(\sigma)$ the $\mathsf{fs}\ \sigma'\,[t_n\rangle\, m_n$.

Given a net $N = \langle S, T, F, C, \mathsf{m}, l\rangle$, a marking m is *reachable* iff there exists a $\mathsf{fs}\ \sigma \in \mathcal{R}^N_{\mathsf{m}}$ such thats $lead(\sigma)$ is m; the set of reachable markings of N is $\mathcal{M}_N = \{m \in \mu S \mid \exists\, \sigma \in \mathcal{R}^N_{\mathsf{m}}.\ lead(\sigma) = m\}$. Given a firing sequence $\mathsf{fs}\ \sigma = m\,[t_1\rangle\, m_1 \cdots m_{n-1}\,[t_n\rangle\, m'$, with $X_\sigma = \sum_{i=1}^n \{t_i\}$ we denote the multiset of transitions associated to this fs. We call this multiset a *state* of the net. The set of states of a Petri nets is then $\mathsf{St}(N) = \{X_\sigma \in \mu T \mid \sigma \in \mathcal{R}^N_{\mathsf{m}}\}$. Observe that different firing sequences may give the same state.

A net is said *safe* whenever its places hold at most one token in all possible evolutions. Formally $\mathsf{N} = (N, l)$ is safe if each marking $m \in \mathcal{M}_N$ is such that $m = [\![m]\!]$. We recall now two notions we will use, namely the one of *subnet* and of *occurrence net*

Subnet and occurrence net: A subnet of a net is a net obtained restricting places and transitions, correspondingly the flow and the context relations F and C, and the initial marking. Let $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ be a Petri net and let $S' \subseteq S$ and $T' \subseteq T$. Then the subnet generated by T' and S' is the net $N|^{S'}_{T'} = \langle S', T', F', C', \mathsf{m}', l'\rangle$, where F' is the restriction of F to S' and T', C' is the restriction of C to S' and T', m' is the multiset on S' obtained by m restricting to places in S', and l' is the restriction of l to transitions in T'.

The notion of occurrence net we recall here is the one called 1-occurrence net of [17] and the intuition behind it is the following: regardless how tokens are produced or consumed, an occurrence net *guarantees* that each transition can *occur* only once. An *occurrence net* $O = \langle S, T, F, C, \mathsf{m}, l\rangle$ is a Petri net where each state is a set, i.e. $\forall X \in \mathsf{St}(O)$ it holds that $X = [\![X]\!]$. The consequence of this definition is that on occurrence nets it is easy to define a notion of *conflict* among transitions. In fact, two transitions t, t' are in conflict if they never appear together in the same state: $t \# t'$ iff $\forall X \in \mathsf{St}(O)$ it holds that $\{t, t'\} \not\subseteq X$.

Unravel Nets: We introduce now a version of the notion of occurrence where dependencies and conflicts among transitions may be observed in a more explicit way, which we call *unravel* nets. We say that a net $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ is *conflict-free* if $\forall s \in S$ it holds that $s^\bullet$ is at most a singleton, and it is acyclic if the reflexive and transitive closure of F is a partial order over $S \cup T$. As we are considering safe nets we will confuse multisets with sets.

Definition 2. *An* unravel net $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ *over a set of label* L *is a safe occurrence net such that (a) for each state* $X \in \mathsf{St}(N)$ *the net* $N|^{S'}_X$ *is acyclic and conflict-free, where* $S' = {}^\bullet X \cup X^\bullet \cup \underline{X}$*, (b) for all* $t_1, t_2 \in T$*, if* $t_1 \# t_2$ *and* $l(t_1) \neq l(t_2)$ *then for each* $t \in l^{-1}(l(t_1))$ *and for each* $t' \in l^{-1}(l(t_2))$ *it holds that* $t \# t'$*, (c) for each* $\mathsf{e} \in l(T)$ *there exists a unique place* s *such that* $\{s\} \subseteq \bigcap_{t \in l^{-1}(\mathsf{e})} {}^\bullet t$*,* ${}^\bullet s = \emptyset$ *and* $\mathsf{m}(s) = 1$*, and we denote that place with* ${}^\diamond\mathsf{e}$*, and (d) for each* $\mathsf{e} \in l(T)$ *there exists a unique place* s *such that* $\{s\} \subseteq \bigcap_{t \in l^{-1}(\mathsf{e})} t^\bullet$*,* $s^\bullet = \emptyset$*,* $\mathsf{m}(s) = 0$ *and* ${}^\bullet s = l^{-1}(\mathsf{e})$*, and we denote that place with* $\mathsf{e}^\diamond$*.*

Condition (a) states that the whole unravel net is not constrained to be either acyclic or conflict-free, but each of its executions gives an acyclic and conflict-free net. Condition (b) guarantees that if two transitions are in conflict and have different labels, then any pair of transitions with the same labels are in conflict as well. The other conditions (c) and (d) guarantee that two equally labeled transitions are in conflict, each transition can happen just once and equally labeled transitions have some common places in their preset. All equally labelled transitions share a place in the preset which is initially marked and that cannot be filled again, and all of them put in a place (where no other transition beside the ones bearing that specific label can put a token). This place may be redundant, unless it is connected to a transition with a contextual arc.

As equally labeled transitions are in conflict in an unravel net all the transitions in a firing sequence have different labels. The sequence of labels of a fs form a *trace* of the net. Formally, let $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ be an unravel net over a set of labels L. Let $\sigma \in \mathcal{R}^N_{\mathsf{m}}$ with $\sigma = \mathsf{m}\,[t_1\rangle\, m_1\,[t_2\rangle\, m_2 \cdots m_{n-1}\,[t_n\rangle\, m_n$. Then a *trace* of N is the sequence of labels $l(t_1 t_2 \cdots t_n)$ and it is denoted with $run(\sigma)$. The set of traces of a net is $\mathsf{Tr}(N) = \{w \in \mathsf{L}^* \mid \exists \sigma \in \mathcal{R}^N_{\mathsf{m}}.\ run(\sigma) = w\}$

Configurations are obtained forgetting the sequence, as each state is obtained by a firing sequence. Let $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ be an unravel net over a set of labels L. Let $X \in \mathsf{St}(N)$, then $\mathsf{X} = \{l(t) \mid t \in X\}$ is a *configuration* of N. The set of configuration of a net is denoted with $\mathsf{Conf}(N)$.

Event Structures.

Events structures usually model *concurrent systems* by defining relationships among events such as *causality* and *conflict*. Among all the possible kind of events structures, we focus on some kinds of event structures and to describe the states of these events structures, we will use event traces. This notion, besides some peculiar cases, suffices to retrieve the usual notion of configuration. Consider a set of events E and let $\rho = \mathsf{e}_1 \cdots \mathsf{e}_n$ be a sequence of distinct events in E (with overloading of notation with ϵ we denote the *empty* sequence). With $\overline{\rho}$ we denote the set $\{\mathsf{e}_1, \ldots, \mathsf{e}_n\}$. Clearly the set associated with the empty sequence is the empty set. Given a sequence ρ of events, its length is $|\overline{\rho}|$ and it is denoted with $len(\rho)$ for each $1 \leq i \leq len(\rho)$ with ρ_i we denote the sequence $\mathsf{e}_1 \cdots \mathsf{e}_i$, and with ρ_0 we denote the empty sequence.

Prime event structures: The notion of *prime event structure* we will use is the one of [1]. We consider only finite event structures, thus the set of events is always finite.

DEFINITION 3. *A* prime event structure (PES) *is a triple* $P = (\mathsf{E}, \rightarrow, \#)$*, where* E *is a set of* events *and* $\rightarrow$*,* $\#$ *are two binary relations on* E *called* causality relation *and* conflict relation *respectively, such that* $\#$ *is irreflexive and symmetric.*

This notion differ substantially from the one given in [18], as $\rightarrow$ should not be a partial order and $\#$ should not be inherited along the partial order, but both have the same configurations and traces, as we are considering finite event structures.

Given a PES $P = (\mathsf{E}, \rightarrow, \#)$ and an event $\mathsf{e} \in \mathsf{E}$ the set of the *immediate causes* of e is $\mathrm{ic}(\mathsf{e}) = \{\mathsf{e}' \mid \mathsf{e}' \rightarrow \mathsf{e}\}$. A sequence of events $\rho = \mathsf{e}_1 \cdots \mathsf{e}_n$ is a *trace* whenever $\overline{\rho}$ is *conflict free*, *i.e.* for each $\mathsf{e}_i, \mathsf{e}_j \in \overline{\rho}$. $\mathsf{e}_i \neq \mathsf{e}_j \Rightarrow \neg(\mathsf{e}_i \# \mathsf{e}_j)$, and for each $i \leq n$. $\mathrm{ic}(\mathsf{e}_i) \subseteq \overline{\rho_{i-1}}$. The configuration associated with ρ is $\overline{\rho}$.

We now review some kinds of event structure where causality may change.

Shrinking Event Structures: Given a PES it is possible to add a suitable relation to model the removal of causal dependencies [1]. The idea is to introduce a ternary relation stipulating that the happening of a specific event (the *modifier*) allows to drop a specific cause for another event (the *target*).

DEFINITION 4. *A* Shrinking Causality Event Structure (SES) *is the quadruple* $\gamma = (\mathsf{E}, \rightarrow, \#, \triangleright)$ *where* $(\mathsf{E}, \rightarrow, \#)$ *is a* PES *and* $\triangleright \subseteq \mathsf{E} \times \mathsf{E} \times \mathsf{E}$ *is the* shrinking causality *relation such that* $[\mathsf{e} \rightarrow \mathsf{e}''] \triangleright \mathsf{e}'$ *implies* $\mathsf{e} \rightarrow \mathsf{e}''$ *for all* $\mathsf{e}, \mathsf{e}', \mathsf{e}'' \in \mathsf{E}$.

The only requirement is that to have a cause to drop for an event, this cause should be hardwired in the causality relation $\rightarrow$. For $[\mathsf{c} \rightarrow \mathsf{t}] \triangleright \mathsf{m}$ we call m the modifier, t the target and c the *contribution*, and we denote with $[\mathsf{c} \rightarrow \mathsf{t}]\triangleright$ the set of all modifiers dropping $\mathsf{c} \rightarrow \mathsf{t}$. The set of dropped causes of an event can be defined w.r.to a specific history, *i.e.* a set of events, by the function $\mathrm{dc}\colon \mathbf{2}^{\mathsf{E}} \times \mathsf{E} \rightarrow \mathbf{2}^{\mathsf{E}}$, which is defined as follows: $\mathrm{dc}(H, \mathsf{e}) = \{\mathsf{e}' \mid \exists \mathsf{d} \in H.\ [\mathsf{e}' \rightarrow \mathsf{e}] \triangleright \mathsf{d}\}$. As for PES, we have a function $\mathrm{ic}\colon \mathsf{E} \rightarrow \mathbf{2}^{\mathsf{E}}$ defined in the same way, *i.e* $\mathrm{ic}(\mathsf{e}) = \{\mathsf{e}' \mid \mathsf{e}' \rightarrow \mathsf{e}\}$, and we say that $\mathrm{ic}(\mathsf{e})$ are the *initial* causes of the event e.

Traces for a SES are defined as follows. Let $\gamma = (\mathsf{E}, \rightarrow, \#, \triangleright)$ be a SES. A *trace* of γ is a sequence of distinct events $\rho = \mathsf{e}_1 \cdots \mathsf{e}_n$ with $\overline{\rho} \subseteq \mathsf{E}$ such that (a) $\overline{\rho}$ is conflict-free and (b) $\forall 1 \leq i \leq \mathit{len}(\rho).\ (\mathrm{ic}(\mathsf{e}_i) \setminus \mathrm{dc}(\overline{\rho_{i-1}}, \mathsf{e}_i)) \subseteq \overline{\rho_{i-1}}$. The relevant difference with the notion of trace for PES is just that in the enabling of each event, depending on the modifiers happened so far (hence the role of the history), the set of the immediate causes may shrink. Consider the following SES, with just three events a, b and c, with $\mathsf{b} \rightarrow \mathsf{c}$ and $[\mathsf{b} \rightarrow \mathsf{c}] \triangleright \mathsf{a}$. a is the modifier for the target c and its happening has the effect that the cause b may be dropped. The maximal traces are: acb, abc, bca and bac. It is worth to notice that in all but the trace bca the event c has no predecessor.

Growing Event Structures: Opposite to shrinking causality there is the notion of growing causality [1], in which a dependency between two events is added after the execution of a third event.

DEFINITION 5. *A* growing causality event structure (GES) *is the quadruple* $\delta = (\mathsf{E}, \rightarrow, \#, \blacktriangleright)$ *where* $(\mathsf{E}, \rightarrow, \#)$ *is a* PES *and* $\blacktriangleright \subseteq \mathsf{E} \times \mathsf{E} \times \mathsf{E}$ *is the* growing causality *relation such that* $\forall \mathsf{e}, \mathsf{e}', \mathsf{e}'' \in \mathsf{E}.\ \mathsf{e}' \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}''] \Longrightarrow \neg(\mathsf{e} \rightarrow \mathsf{e}'')$.

Here the requirement is that a dependency can be added iff it is not hardwired in $\rightarrow$. For $\mathsf{m} \blacktriangleright [\mathsf{c} \rightarrow \mathsf{t}]$ we call m the *modifier*, t the *target* and c the *contribution*, and we denote with $\blacktriangleright [\mathsf{c} \rightarrow \mathsf{t}]$ the set of all modifiers adding $\mathsf{c} \rightarrow \mathsf{t}$. The set of added causes of an event has to be defined w.r.to a specific history, *i.e.* a set of events, and it is done using the function $\mathrm{ac}\colon \mathbf{2}^{\mathsf{E}} \times \mathsf{E} \rightarrow \mathbf{2}^{\mathsf{E}}$, which is defined as follows: $\mathrm{ac}(H, \mathsf{e}) = \{\mathsf{e}' \mid \exists \mathsf{d} \in H.\ \mathsf{d} \blacktriangleright [\mathsf{e}' \rightarrow \mathsf{e}]\}$. The initial causes of an event are defined as usual.

Let $\delta = (\mathsf{E}, \rightarrow, \#, \blacktriangleright)$ be a GES. A *trace* of δ is a sequence of distinct events of E $\rho = \mathsf{e}_1 \cdots \mathsf{e}_n$ such that (a) $\overline{\rho}$ is conflict-free and (b) $\forall 1 \leq i \leq \mathit{len}(\rho).\ (\mathrm{ic}(\mathsf{e}_i) \cup \mathrm{ac}(\overline{\rho_{i-1}}, \mathsf{e}_i)) \subseteq \overline{\rho_{i-1}}$.

Consider just three events a, b and c, and the unique non empty relation is $\mathsf{a} \blacktriangleright [\mathsf{b} \rightarrow \mathsf{c}]$. a is the modifier for the target c and its happening has the effect that the cause b should be added. The maximal traces are: abc, bac, cab and cba.

Dynamic causality event structure: In [1] it is shown that neither the shrinking causality relation can be expressed with the growing causality relation or vice versa, hence SES and GES are incomparable. Thus the two notions of shrinking and growing causality can be put together in a definition.

DEFINITION 6. *A* dynamic causality event structure (DCES) *is a 5-tuple* $\Sigma = (\mathsf{E}, \rightarrow, \#, \triangleright, \blacktriangleright)$*, where* $(\mathsf{E}, \rightarrow, \#)$ *is a* PES $\triangleright \subseteq \mathsf{E} \times \mathsf{E} \times \mathsf{E}$ *is the* shrinking causality *relation, and* $\blacktriangleright \subseteq \mathsf{E} \times \mathsf{E} \times \mathsf{E}$ *is the* growing causality *relation, and are such that for all* $\mathsf{e}, \mathsf{e}', \mathsf{e}'' \in \mathsf{E}$ *(a)* $[\mathsf{e} \rightarrow \mathsf{e}''] \triangleright \mathsf{e}' \wedge \nexists \mathsf{m} \in \mathsf{E}.\ \mathsf{m} \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}''] \Longrightarrow \mathsf{e} \rightarrow \mathsf{e}''$*, (b)* $\mathsf{e}' \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}''] \wedge \nexists \mathsf{m} \in \mathsf{E}.[\mathsf{e} \rightarrow \mathsf{e}''] \triangleright \mathsf{m} \Longrightarrow \neg(\mathsf{e} \rightarrow \mathsf{e}'')$*, and (c)* $\forall \mathsf{e}, \mathsf{e}' \in \mathsf{E}.\ \nexists \mathsf{m}, \mathsf{a} \in \mathsf{E}.\ \mathsf{a} \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}'] \triangleright \mathsf{m}$.

For detailed comments on this definition we refer to [1], it should be observed, however, that the definition we consider here is slightly less general of the one presented there, as we add a further condition, the last one, which is defined in [2] and does not allow that the same contribution can be added and removed by two different modifiers. These are called in [2] *single state dynamic causality event structures* and rule out the fact that some causality (or absence of) depends on the order of modifiers. Conditions (a) and (b) simply rephrase the conditions under which the shrinking and growing relations are defined: in the case of the shrinking relation the dependency should be present and in the case of the growing not; condition (c) says that if a dependency is added then it cannot be removed.

The traces are defined accordingly. Let $\Sigma = (\mathsf{E}, \rightarrow, \#, \triangleright, \blacktriangleright)$ be a DCES. A *trace* of Σ is a sequence of distinct events $\rho = \mathsf{e}_1 \cdots \mathsf{e}_n$ with $\overline{\rho} \subseteq \mathsf{E}$ such that (a) $\overline{\rho}$ is conflict-free and (b) $\forall 1 \leq i \leq \mathit{len}(\rho).\ ((\mathrm{ic}(\mathsf{e}_i) \cup \mathrm{ac}(\overline{\rho_{i-1}}, \mathsf{e}_i)) \setminus \mathrm{dc}(\overline{\rho_{i-1}}, \mathsf{e}_i)) \subseteq \overline{\rho_{i-1}}$.

Consider the set of events $\{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}\}$, with $\mathsf{b} \rightarrow \mathsf{c}$, $[\mathsf{b} \rightarrow \mathsf{c}]\triangleright\mathsf{a}$ and $\mathsf{d} \blacktriangleright [\mathsf{e} \rightarrow \mathsf{c}]$. a and d are the modifiers for the target c, the happening of a has the effect that the cause b may be dropped, and the one of d that the cause e should be added for c. If the prefix of the trace is bc (the target c is executed before of one of its modifiers a and d) then the final part of the trace is any combination of a, e and d. If the modifier

a is executed before c then we have the traces ac followed by any combination of the remaining three events, abcde and abced and finally if the two modifiers happen before c we have adecb or adebc and similarly we have daecb and daebc. Clearly, after the happening of the two modifiers in any order, it not prescribed that c happens immediately, as it may happen after b which is not any longer a cause for it.

We recast in this setting the example in the introduction.

EXAMPLE 1. *The events of the seller example illustrated in the introduction are* c *corresponding to* `confirm`, r *to* `request`, n *to* `not-available`, a *to* `available` *which means available somewhere else,* p *to* `pay`, d *to* `deliver`, i *that means that the good is present in stock and finally* u *to* `premUser`. *The event structure has just a pair of events in conflict, namely* i *and* n *and the various relations are: the causal relation is* $\mathsf{r} \to \mathsf{i}, \mathsf{r} \to \mathsf{a}, \mathsf{i} \to \mathsf{c}, \mathsf{p} \to \mathsf{d}, \mathsf{c} \to \mathsf{d}$ *and* $\mathsf{u} \to \mathsf{d}$, *the shrinking relation is* $[\mathsf{p} \to \mathsf{s}] \rhd \mathsf{u}$, $[\mathsf{i} \to \mathsf{c}] \rhd \mathsf{n}$ *and* $[\mathsf{u} \to \mathsf{d}] \rhd \mathsf{p}$ *and finally the growing relation is* $\mathsf{n} \blacktriangleright [\mathsf{a} \to \mathsf{c}]$. *Once that the event* r *happens then there are two possibilities: either the good is present (*i*) or it is not (*n*). In the former situation the confirm message can be issued (*c*) and after that the good can be delivered, either after a payment* p *or because the one issuing the request is a premium user (*u*). A trace could be* ricpd *or* ricud. *A premium user can also pay, but this has no influence on the delivering of the good. In the latter situation (the good is not in stock) the* c *depends on the availability of the good at some place (*$\mathsf{n} \blacktriangleright [\mathsf{a} \to \mathsf{c}]$*) and it should not depend any longer from* i *(*$[\mathsf{i} \to \mathsf{c}] \rhd \mathsf{n}$*). A trace is* rnacpd.

3. UNRAVEL NETS AND EVENT STRUCTURES

In this section we relate unravel nets and event structures. We first characterize some unravel nets and then we show how to associate an events structure, and then we will directly consider DCES and construct a suitable unravel net.

From unravel nets to event structures.

Given an unravel net $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ over a set of labels L, a transition $t' \in T$, with $\mathsf{Pred}(t')$ we denote the set $\{t \in T \mid t^\bullet \cap {}^\bullet t' \neq \emptyset\}$. Two different labels e and e′ in $l(T)$ are said to be in conflict, denoted $\mathsf{e} \bowtie \mathsf{e}'$ iff $\exists t \in l^{-1}(\mathsf{e})$ and $\exists t' \in l^{-1}(\mathsf{e}')$ and $t \# t'$. Observe that due to condition (b) of Def. 2 this is well defined.

We start characterizing unravel nets representing event structures where no modifier is present.

DEFINITION 7. *Let $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ be an unravel net over the set of labels* L. *We say that N is* simple *whenever l is injective, C is empty and for each $s \in S$. $|{}^\bullet s| \leq 1$.*

To a simple unravel net a PES can be easily associated. The intuition is the expected one: to each place s in the preset of a transition t labeled with an event e we associate a causal dependency for e, namely with the unique event associated with the transition ${}^\bullet s$.

PROPOSITION 1. *Let $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ be a simple unravel net, then $\mathcal{E}_{si}(N) = (l(T), \to, \#)$ is a* PES, *where the conflict relation is defined as follows: $\mathsf{e} \# \mathsf{e}'$ iff $\mathsf{e} \bowtie \mathsf{e}'$, and the causal relation is defined as follows: for each $\mathsf{e} \in l(T)$, for each $s \in {}^\bullet l^{-1}(\mathsf{e})$ we have $l({}^\bullet s) \to \mathsf{e}$.*

The following theorem states that a simple net and the corresponding PES have the same traces.

THEOREM 1. *Let $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ be a simple unravel net and let $\mathcal{E}_{si}(N)$ be the associated* PES. *Then w is a trace of N iff w is a trace of $\mathcal{E}_{si}(N)$.*

We introduce now the notion of *stable* unravel net. The intuition behind this definition is the following: contextual conditions can be removed (they are initially marked places, and there is no incoming arc connected to these conditions) or they can be established and they last *forever* (the places have no outgoing arcs). Given a transition t such that $\underline{t} \neq \emptyset$, with $\mathsf{Modifiers}(t)$ we denote the set of labels associated with the contextual arcs and it is equal to $\{l(s^\bullet) \mid s \in \underline{t}\} \cup \{l({}^\bullet s) \mid s \in \underline{t}\}$.

DEFINITION 8. *Let $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ be an unravel net over the set of labels* L. *We say that N is* stable *whenever (i) C is not empty, (ii) for all $t \in T$ $\underline{t} \neq \emptyset \;\Rightarrow\; \forall s \in \underline{t}.\ {}^\bullet s = \emptyset \;\vee\; s^\bullet = \emptyset$, (iii) for all $\mathsf{e} \in l(T)$, for each marking $m \in \mathcal{M}_N$, for all $t, t' \in l^{-1}(\mathsf{e})$ with $t \neq t'$ if $m\,[t\rangle$ then $\neg(m\,[t'\rangle)$, and (iv) for all $\mathsf{e} \in l(T)$, for all $t, t' \in l^{-1}(\mathsf{e})$ it holds that $\emptyset \neq \underline{t} \neq \underline{t'} \neq \emptyset$ and $\mathsf{Modifiers}(t) = \mathsf{Modifiers}(t')$.*

In stable unravel nets contextual arcs are used to signal modifiers to a label, by allowing to trigger the proper instance of that label (the one corresponding to the proper combination of triggers), whereas in the classical theory of contextual nets ([14] or [4]) they are used to either enlarge concurrency or to model asymmetric conflicts. The final conditions are introduced to assure that equally labeled transitions cannot be simultaneously enabled at any marking and have always a non empty context. Stability is not enough to capture shrinking or growing causality. In fact we have to identify, among the various conflicting transitions that are equally labeled which is the one representing the occurrence of the event without that any modifier has happened. Once that we have found that the transition representing the happening without any modifier, we may find out whether the modifier is a shrinking or growing (or both, but for different labels).

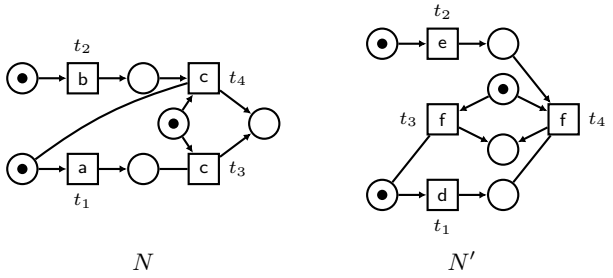

Figure 1: Two well formed unravel nets

DEFINITION 9. *Let $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ be a stable unravel net over the set of labels* L. *We say that N is* well

formed *(wfn) iff, for all* $\mathsf{e} \in l(T)$*, one of the following holds: if* $|l^{-1}(\mathsf{e})| > 1$ *then* $\exists!\ t \in l^{-1}(\mathsf{e})$ *such that* $\forall s \in \underline{t}.\ {}^\bullet s = \emptyset$ *or if* $l^{-1}(\mathsf{e}) = \{t\}$ *then* $\underline{t} = \emptyset$.

This definition guarantees that among the transitions bearing the same label there is one that may happen before that all the modifiers have happened. The second condition says that if there is just one instance of a given label (event) then there is no modifier for that event.

Due to the two definitions above it is possible to introduce the following set of labels. Given a wfn $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ and a label $\mathsf{e} \in l(T)$, with $\mathsf{StableC}(\mathsf{e}) = l(\bigcup_{s \in {}^\bullet t} {}^\bullet s)$, where t is the unique transition in $l^{-1}(\mathsf{e})$ such that either $\forall s \in \underline{t}.\ {}^\bullet s = \emptyset$ or $\underline{t} = \emptyset$, we denote the set of *stable* causes of e.

Consider the stable N net in Fig. 1, it is clearly a well formed, the set of modifier for c contains just a and $\mathsf{StableC}(\mathsf{c}) = \{\mathsf{b}\}$. Instead, considering the wfn N' in Fig. 1, the set of modifiers for f contains just d but $\mathsf{StableC}(\mathsf{f}) = \emptyset$.

The following definition captures what is the usage of contextual arcs in the case of removing causes.

Definition 10. *Let* $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ *be a wfn over the set of labels* L *and let* $\mathsf{e}', \mathsf{e} \in l(T)$ *such that* $\mathsf{e}' \neq \mathsf{e}$*. We say that* e' *is a* shrinking modifier *for* e *iff* $\exists t, t' \in l^{-1}(\mathsf{e})$ *such that* $t \neq t'$, $({}^\diamond\mathsf{e}', t) \in C$ *and* $(\mathsf{e}'^\diamond, t') \in C$ *and there exists* $\mathsf{e}'' \in \mathsf{StableC}(\mathsf{e})$ *such that* $\mathsf{e}'' \notin l(\bigcup_{s \in {}^\bullet t'} {}^\bullet s)$*. The set of shrinking modifiers for* e *is denoted with* $\mathsf{ShrMod}(\mathsf{e})$.

When the set $\mathsf{ShrMod}(\mathsf{e})$ is not empty, one has to identify what are the dependencies that are dropped. These are captured as follows. Consider $\mathsf{e}' \in \mathsf{ShrMod}(\mathsf{e})$, then there exists $t \in l^{-1}(\mathsf{e})$ such that $({}^\diamond\mathsf{e}', t) \in C$. The elements that are dropped are $\mathsf{Drop}^{\mathsf{e}}_{\mathsf{e}'} = \{l(\bar{t}) \mid \exists ({}^\diamond\mathsf{e}', t), (\mathsf{e}'^\diamond, t') \in C.\ l(t) = l(t') = \mathsf{e} \ \wedge\ \bar{t} \in {}^\bullet({}^\bullet t)\} \cap \mathsf{StableC}(\mathsf{e})$.

Similarly we can define the growing modifier, using the contextual conditions as well.

Definition 11. *Let* $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ *be a wfn over the set of labels* L *and let* $\mathsf{e}', \mathsf{e} \in l(T)$ *such that* $\mathsf{e}' \neq \mathsf{e}$*. We say that* e' *is a* growing modifier *for* e *iff* $\exists t, t' \in l^{-1}(\mathsf{e})$ *and* $\hat{t} \in l^{-1}(\mathsf{e}')$ *such that* $t \neq t'$, $({}^\diamond\mathsf{e}', t) \in C$ *and* $(\mathsf{e}'^\diamond, t') \in C$ *there exists* $\mathsf{e}'' \in l(\bigcup_{s \in {}^\bullet t'} {}^\bullet s)$ *such that* $\mathsf{e}'' \notin \mathsf{StableC}(\mathsf{e})$*. The set of growing modifiers for* e *is denoted with* $\mathsf{GroMod}(\mathsf{e})$.

Again when the set $\mathsf{GroMod}(\mathsf{e})$ is not empty, one has to identify what are the dependencies that are added. Consider again $\mathsf{e}' \in \mathsf{GroMod}(\mathsf{e})$, again there exists $t \in l^{-1}(\mathsf{e})$ such that $({}^\diamond\mathsf{e}', t) \in C$. The elements that are added are $\mathsf{Add}^{\mathsf{e}}_{\mathsf{e}'} = \{l(\bar{t}) \mid \exists ({}^\diamond\mathsf{e}', t), (\mathsf{e}'^\diamond, t') \in C.\ l(t) = l(t') = \mathsf{e} \ \wedge\ \bar{t} \in {}^\bullet({}^\bullet t)\} \setminus \mathsf{StableC}(\mathsf{e})$.

Consider again the nets of the Fig. 1. The first one has a shrinking modifier a for the label c ($\mathsf{ShrMod}(\mathsf{c}) = \{\mathsf{a}\}$) and $\mathsf{Drop}^{\mathsf{c}}_{\mathsf{a}} = \{\mathsf{b}\}$, whereas the second one has a growing modifier d for the label f ($\mathsf{GroMod}(\mathsf{f}) = \{\mathsf{d}\}$) and $\mathsf{Add}^{\mathsf{f}}_{\mathsf{d}} = \{\mathsf{e}\}$. In the case of the wfn in Fig. 2 we have that the label d has two modifiers: e and f, where e is both shrinking and growing, whereas f is only shrinking.

The following two propositions characterize event structures associated with wfns where only one kind of modifier is present, either shrinking or growing.

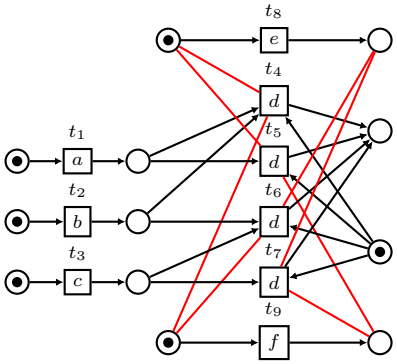

Figure 2: A wfn such that $\mathsf{ShrMod}(\mathsf{d}) \neq \emptyset$ and $\mathsf{GroMod}(\mathsf{d}) \neq \emptyset$. Contextual arcs are drawn in red to avoid confusion.

Proposition 2. *Let* $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ *be a wfn such that* $\bigcup_{\mathsf{e} \in l(T)} \mathsf{ShrMod}(\mathsf{e}) \neq \emptyset$ *and* $\bigcup_{\mathsf{e} \in l(T)} \mathsf{GroMod}(\mathsf{e}) = \emptyset$*, then* $\mathcal{E}_{sh}(N) = (l(T), \#, \rightarrow, \rhd)$ *is a* SES*, where the conflict relation is defined as follows:* $\mathsf{e} \ \#\ \mathsf{e}'$ *iff* $\mathsf{e} \bowtie \mathsf{e}'$*, the causality relation is defined as follows: for each* $\mathsf{e} \in l(T)$*, for each* $s \in {}^\bullet l^{-1}(\mathsf{e})$*, for each* $t \in {}^\bullet s$ *we have* $l(t) \rightarrow \mathsf{e}$*, and the shrinking causality relation is defined as follows: for all* $\mathsf{e} \in l(T)$*, for each* $\mathsf{e}' \in \mathsf{ShrMod}(\mathsf{e})$ *define* $[\mathsf{e}'' \rightarrow \mathsf{e}] \rhd \mathsf{e}'$ *where* $\mathsf{e}'' \in \mathsf{Drop}^{\mathsf{e}}_{\mathsf{e}'}$.

Proposition 3. *Let* $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ *be a wfn such that* $\bigcup_{\mathsf{e} \in l(T)} \mathsf{ShrMod}(\mathsf{e}) = \emptyset$ *and* $\bigcup_{\mathsf{e} \in l(T)} \mathsf{GroMod}(\mathsf{e}) \neq \emptyset$*, then* $\mathcal{E}_{gr}(N) = (l(T), \#, \rightarrow, \blacktriangleright)$ *is a* GES*, where the conflict relation is defined as follows:* $\mathsf{e} \ \#\ \mathsf{e}'$ *iff* $\mathsf{e} \bowtie \mathsf{e}'$*, the causality relation is defined as follows: for each* $\mathsf{e} \in l(T)$*, for each* $s \in {}^\bullet l^{-1}(\mathsf{e})$*, for each* $t \in {}^\bullet s$ *we have* $l(t) \rightarrow \mathsf{e}$ *provided that for all* $\mathsf{e}' \in \mathsf{GroMod}(\mathsf{e})$ *it holds that* $l(t) \notin \mathsf{Add}^{\mathsf{e}}_{\mathsf{e}'}$*, and the growing causality relation is defined as follows: for all* $\mathsf{e} \in l(T)$*, for each* $\mathsf{e}' \in \mathsf{GroMod}(\mathsf{e})$ *define* $\mathsf{e}'' \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}']$ *where* $\mathsf{e}'' \in \mathsf{Add}^{\mathsf{e}}_{\mathsf{e}'}$.

The shrinking event structure associated with N in Fig. 1 is $[\mathsf{b} \rightarrow \mathsf{c}] \rhd \mathsf{a}$, whereas the growing event structure associated with N' is $\mathsf{d} \blacktriangleright [\mathsf{e} \rightarrow \mathsf{f}]$

As before we have the following two theorems.

Theorem 2. *Let* $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ *be a wfn such that* $\bigcup_{\mathsf{e} \in l(T)} \mathsf{ShrMod}(\mathsf{e}) \neq \emptyset$ *and* $\bigcup_{\mathsf{e} \in l(T)} \mathsf{GroMod}(\mathsf{e}) = \emptyset$*, and let* $\mathcal{E}_{sh}(N) = (l(T), \#, \rightarrow, \rhd)$ *be the associated* SES*. Then* w *is a trace of* N *iff* w *is a trace of* $\mathcal{E}_{sh}(N)$.

Theorem 3. *Let* $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ *be a wfn such that* $\bigcup_{\mathsf{e} \in l(T)} \mathsf{ShrMod}(\mathsf{e}) = \emptyset$ *and* $\bigcup_{\mathsf{e} \in l(T)} \mathsf{GroMod}(\mathsf{e}) \neq \emptyset$*, and let* $\mathcal{E}_{gr}(N) = (l(T), \#, \rightarrow, \blacktriangleright)$ *be the associated* GES*. Then* w *is a trace of* N *iff* w *is a trace of* $\mathcal{E}_{gr}(N)$.

We can now put together what we have seen up to now obtaining a DCES.

Proposition 4. *Let* $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ *be a wfn such that if* $\mathsf{e}' \in \left(\bigcup_{\mathsf{e} \in l(T)} \mathsf{ShrMod}(\mathsf{e}) \cap \bigcup_{\mathsf{e} \in l(T)} \mathsf{GroMod}(\mathsf{e}) \neq \emptyset\right)$ *it holds that* $\mathsf{Drop}^{\mathsf{e}}_{\mathsf{e}'} \cap \mathsf{Add}^{\mathsf{e}}_{\mathsf{e}'} = \emptyset$*, then* $\mathcal{E}_{sg}(N) = (l(T), \#, \rightarrow, \rhd, \blacktriangleright)$ *is a* DCES*, where the conflict relation is defined as*

follows: $\mathsf{e} \# \mathsf{e}'$ *iff* $\mathsf{e} \bowtie \mathsf{e}'$*, the causality relation is defined as follows: for each* $\mathsf{e} \in l(T)$*, for each* $s \in {}^\bullet l^{-1}(\mathsf{e})$*, for each* $t \in {}^\bullet s$ *we have* $l(t) \to \mathsf{e}$ *provided that for all* $\mathsf{e}' \in \mathsf{GroMod}(\mathsf{e})$ *it holds that* $l(t) \notin \mathsf{Add}^{\mathsf{e}}_{\mathsf{e}'}$*, the shrinking causality relation is defined as follows: for all* $\mathsf{e} \in l(T)$*, for each* $\mathsf{e}' \in \mathsf{ShrMod}(\mathsf{e})$ *define* $[\mathsf{e}'' \to \mathsf{e}] \rhd \mathsf{e}'$ *where* $\mathsf{e}'' \in \mathsf{Drop}^{\mathsf{e}}_{\mathsf{e}'}$*, and the growing causality relation is defined as follows: for all* $\mathsf{e} \in l(T)$*, for each* $\mathsf{e}' \in \mathsf{GroMod}(\mathsf{e})$ *define* $\mathsf{e}'' \blacktriangleright [\mathsf{e} \to \mathsf{e}']$ *where* $\mathsf{e}'' \in \mathsf{Add}^{\mathsf{e}}_{\mathsf{e}'}$*.*

Consider the net in Fig. 2. The sets of modifiers for d are $\mathsf{ShrMod}(\mathsf{d}) = \{\mathsf{e}, \mathsf{f}\}$ and $\mathsf{GroMod}(\mathsf{d}) = \{\mathsf{e}\}$, $\mathsf{Drop}^{\mathsf{d}}_{\mathsf{e}} = \{\mathsf{a}\}, \mathsf{Drop}^{\mathsf{d}}_{\mathsf{f}} = \{\mathsf{b}\}$ and $\mathsf{Add}^{\mathsf{d}}_{\mathsf{e}} = \{\mathsf{c}\}$ and finally $\mathsf{StableC}(\mathsf{d}) = \emptyset$. The associated DCES has the following enabling relation: $\mathsf{a} \to \mathsf{d}$ and $\mathsf{b} \to \mathsf{d}$, the conflict relation is empty, the shrinking causality relation is $[\mathsf{a} \to \mathsf{d}] \rhd \mathsf{e}$ and $[\mathsf{b} \to \mathsf{d}] \rhd \mathsf{f}$ and the growing causality relation is $\mathsf{e} \blacktriangleright [\mathsf{c} \to \mathsf{d}]$.

THEOREM 4. *Let* $N = \langle S, T, F, C, \mathsf{m}, l\rangle$ *be a* wfn *such that if* $\mathsf{e}' \in \big(\bigcup_{\mathsf{e}\in l(T)} \mathsf{ShrMod}(\mathsf{e}) \cap \bigcup_{\mathsf{e}\in l(T)} \mathsf{GroMod}(\mathsf{e}) \neq \emptyset\big)$ *it holds that* $\mathsf{Drop}^{\mathsf{e}}_{\mathsf{e}'} \cap \mathsf{Add}^{\mathsf{e}}_{\mathsf{e}'} = \emptyset$*, and let* $\mathcal{E}_{sg}(N) = (l(T), \#, \to, \rhd, \blacktriangleright)$ *be the associated* DCES*. Then* w *is a trace of* N *iff* w *is a trace of* $\mathcal{E}_{sg}(N)$*.*

From DCES *to unravel nets.*

The idea here to associate to each event of the event structure a set of equally labeled transitions, each of them representing one of the possible *enabling conditions* of the event.

We start defining a number of sets we will use in the following. Consider a DCES $\Sigma = (\mathsf{E}, \#, \to, \rhd, \blacktriangleright)$ then, for each $\mathsf{e} \in \mathsf{E}$

$\mathsf{Before}(\mathsf{e}) = \{\mathsf{e}' \mid \mathsf{e}' \to \mathsf{e} \ \vee \ \mathsf{e}'' \blacktriangleright [\mathsf{e}' \to \mathsf{e}]\}$ is the set of events that causally are before e, also *potentially*, in the case of growing causality,

$\mathsf{After}(\mathsf{e}) = \{\mathsf{e}' \mid \mathsf{e} \to \mathsf{e}' \ \vee \ \mathsf{e}'' \blacktriangleright [\mathsf{e} \to \mathsf{e}']\}$ is the set of events that causally are after e, also *potentially*, in the case of growing causality,

$\mathsf{ModShr}(\mathsf{e}) = \{\mathsf{e}' \mid [\mathsf{e}'' \to \mathsf{e}] \rhd \mathsf{e}'\}$ is the set of shrinking modifiers for a certain event e,

$\mathsf{ModGro}(\mathsf{e}) = \{\mathsf{e}' \mid \mathsf{e}' \blacktriangleright [\mathsf{e}'' \to \mathsf{e}]\}$ is the set of growing modifiers for a certain event e,

$\mathsf{Less}(\mathsf{e}, X) = \{\mathsf{e}' \mid \exists \mathsf{e}'' \in X.\ [\mathsf{e}' \to \mathsf{e}] \rhd \mathsf{e}''\}$ is the set of causes of e that may be removed due to a modifier in the set X,

$\mathsf{More}(\mathsf{e}, X) = \{\mathsf{e}' \mid \exists \mathsf{e}'' \in X.\ \mathsf{e}'' \blacktriangleright [\mathsf{e}' \to \mathsf{e}]\}$ is the set of causes of e that may be added due to a modifier in the set X

and $\mathsf{Solid}(\mathsf{e}) = \{\mathsf{e}' \mid \mathsf{e}' \to \mathsf{e}\}$.

For each event e and each subset of modifiers $\mathsf{ModShr}(\mathsf{e}) \cup \mathsf{ModGro}(\mathsf{e})$ we will introduce a transition labeled with e. Among all the transitions with the same labels the one that should be executed is the one where all and only the modifiers in the subset have been previously executed.

PROPOSITION 5. *Let* $\Sigma = (\mathsf{E}, \#, \to, \rhd, \blacktriangleright)$ *be a* DCES*. Then to* Σ *we associate the* wfn $\mathcal{N}(\Sigma) = \langle S, T, F, C, \mathsf{m}, l\rangle$ *where*

$S = (\mathsf{E} \times \{*\}) \cup (\mathsf{E} \times \{\circ\}) \cup \{(\mathsf{e}, \mathsf{e}', \to) \mid \mathsf{e}' \in \mathsf{After}(\mathsf{e})\} \cup \{(\mathsf{e}, \mathsf{e}', \to) \mid \mathsf{e} \in \mathsf{Before}(\mathsf{e}')\} \cup \{(\{\mathsf{e}, \mathsf{e}'\}, \#) \mid \mathsf{e} \# \mathsf{e}'\}$,

$T = \{(\mathsf{e}, X) \mid \mathsf{e} \in \mathsf{E} \ \wedge \ X \subseteq \mathsf{ModShr}(\mathsf{e}) \cup \mathsf{ModGro}(\mathsf{e})\}$,

$(s, t) \in F$ *whenever one of the following holds: (a)* $s = (\mathsf{e}, *)$ *and* $t = (\mathsf{e}, X)$*, for any* X*, (b)* $s = (\mathsf{e}', \mathsf{e}, \to)$ *and* $t = (\mathsf{e}, X)$ *with either* $\mathsf{e}' \in \mathsf{Solid}(\mathsf{e}) \setminus \mathsf{Less}(\mathsf{e}, X \cap \mathsf{ModShr}(\mathsf{e}))$ *or* $\mathsf{e}' \in \mathsf{Solid}(\mathsf{e}) \setminus \mathsf{More}(\mathsf{e}, X \cap \mathsf{ModGro}(\mathsf{e}))$*, (c)* $s = (Y, \#)$*, with* $t = (\mathsf{e}, X)$ *and* $\mathsf{e} \in Y$*, for any* X*,*

$(t, s) \in F$ *whenever one of the following holds: (a)* $s = (\mathsf{e}, \circ)$ *and* $t = (\mathsf{e}, X)$*, for any* X*, (b)* $s = (\mathsf{e}, \mathsf{e}', \to)$ *and* $t = (\mathsf{e}, X)$*, for any* X*,*

$(s, t) \in C$ *whenever (a)* $s = (\mathsf{e}', *)$ *and* $t = (\mathsf{e}, \emptyset)$*, for all* $\mathsf{e}' \in \mathsf{ModGro}(\mathsf{e}) \cup \mathsf{ModShr}(\mathsf{e}) \neq \emptyset$*, (b)* $s = (\mathsf{e}', *)$ *and* $t = (\mathsf{e}, X)$ *if* $\mathsf{e}' \notin X \cap \mathsf{ModShr}(\mathsf{e})$*, (c)* $s = (\mathsf{e}', \circ)$ *and* $t = (\mathsf{e}, X)$*, if* $\mathsf{e}' \in X \cap \mathsf{ModGro}(\mathsf{e})$*,*

$\mathsf{m}(s) = 1$ *iff* s *is equal to* $(\mathsf{e}, *)$ *or to* $(\mathsf{e}, \mathsf{e}', \#)$ *and it is equal to* 0 *otherwise, and*

$l(\mathsf{e}, X) = \mathsf{e}$.

Each target of modifiers has a number of different *implementations* depending on the set of modifiers that has happened. The flow and contextual relation are defined easily using the sets of labels defined before Prop. 5.

Consider the DCES where the causal relation is $\mathsf{a} \to \mathsf{d}$ and $\mathsf{b} \to \mathsf{d}$, the empty conflict relation, the shrinking causality relation is $[\mathsf{a} \to \mathsf{d}] \rhd \mathsf{e}$ and $[\mathsf{b} \to \mathsf{d}] \rhd \mathsf{f}$ and the growing causality relation is $\mathsf{e} \blacktriangleright [\mathsf{c} \to \mathsf{d}]$. It is straightforward to check that the synthesized wfn is the net in Fig. 2.

THEOREM 5. *Let* $\Sigma = (\mathsf{E}, \#, \to, \rhd, \blacktriangleright)$ *be a* DCES*, and let* $\mathcal{N}(\Sigma) = \langle S, T, F, C, \mathsf{m}, l\rangle$ *be the associated unravel net. Then* w *is a trace of* Σ *iff* w *is a trace of* $\mathcal{N}(\Sigma)$*.*

Observe that $\mathcal{E}_{sg}(\mathcal{N}(\Sigma))$ gives indeed Σ but in general, starting from a wfn N satisfying all the condition in Prop. 4, N if different from $\mathcal{N}(\mathcal{E}_{sg}(N))$.

EXAMPLE 2. *Consider the seller described in the introduction and formalized as* DCES Σ *in Ex. 1. The associated net* $\mathcal{N}(\Sigma)$ *is the one depicted in Fig. 3. The event* s *has four instances, as it has two modifiers (*p *and* u*), hence the four transitions are* $(\mathsf{s}, \emptyset), (\mathsf{s}, \{\mathsf{p}\}), (\mathsf{s}, \{\mathsf{u}\})$ *and* $(\mathsf{s}, \{\mathsf{p}, \mathsf{u}\})$*, whereas the event* c *has just two instances as there is just a modifier acting as a modifier letting the dependencies grow and shrink at the same time.*

4. CONCLUSION

In this paper we propose a suitable implementation of dynamic causality in Petri nets. This kind of causality is somehow observed at the level of traces of the net and it relies on having, for the same observable action, a different transition coding the causes really used. To activate the proper transition we use contextual arcs connected to the pre and post places of the various modifiers.

We depart from the usual way of relating nets to event structures, as usually there is an one to one correspondence between events and transitions. However we do believe that our approach is correct even from the point of view of the Petri nets theory. In fact the possibility of changing the

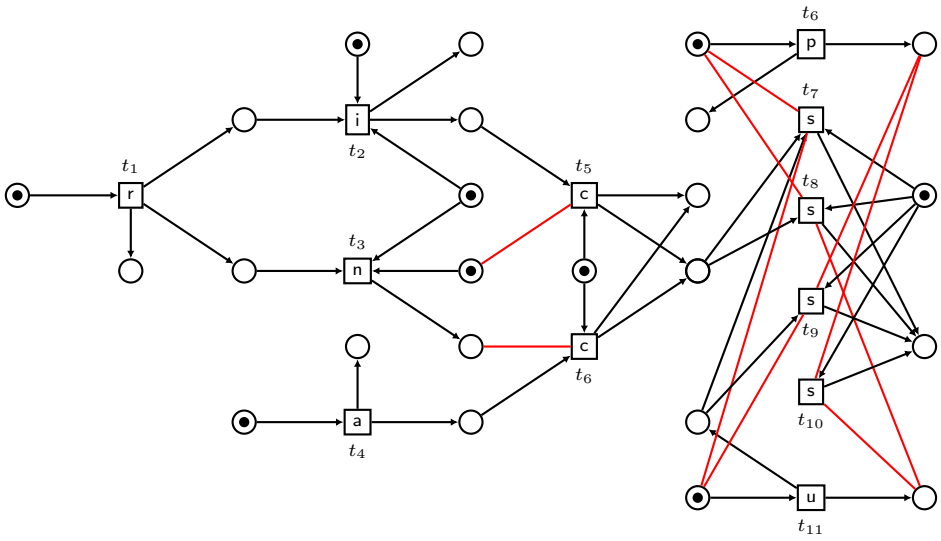

Figure 3: The unravel net of the seller in example 1

causes of a given event resembles what happens in Predicate/Transition Petri nets where the actual execution of a transition depends on the satisfaction of a given predicate (see [10] for an account on this kind of nets), and these nets can be flattened to ordinary nets, provided that a number of duplications occurs.

With our proposal we believe we have done a first step toward the settling of a proper relationship between Petri nets and dynamic causality event structures, which can be fruitfully used in the service oriented scenario.

Many issues have to be investigated, for instance whether it would be possible to overcome the limitation stating that for a given contributor and target it is not possible to have a growing and a shrinking modifier, the net we obtain may be unnecessarily redundant, or what would happen when not only dynamic *standard* causality is considered, but also *dynamic* non standard causality like the circular one which has a relevant part in contract oriented computations.

Acknowledgments. The authors wish to thank the reviewers for their useful comments and suggestions.

5. REFERENCES

[1] Y. Arbach, D. Karcher, K. Peters, and U. Nestmann. Dynamic causality in event structures. In *FORTE 2015*, LNCS 9039, 2015.

[2] Y. Arbach, D. Karcher, K. Peters, and U. Nestmann. Dynamic causality in event structures (technical report). *CoRR*, abs/1504.00512, 2015.

[3] P. Baldan, N. Busi, A. Corradini, and G. Pinna. Domain and event structure semantics for Petri nets with read and inhibitor arcs. *Theoretical Computer Science*, 323(1-3), 2004.

[4] P. Baldan, A. Corradini, and U. Montanari. Contextual Petri nets, asymmetric event structures and processes. *Information and Computation*, 171(1), 2001.

[5] M. Bartoletti, T. Cimoli, and G. M. Pinna. Lending Petri nets. *Science of Computer Programming*, 112, 2015.

[6] M. Bartoletti, T. Cimoli, G. M. Pinna, and R. Zunino. Circular causality in event structures. *Fundamenta Informaticae*, 134(3-4), 2014.

[7] T. Bolognesi and E. Brinksma. Introduction to the ISO specification language LOTOS. *Computer Networks*, 14, 1987.

[8] G. Boudol. Flow Event Structures and Flow Nets. *Semantics of Systems of Concurrent Processes*, LNCS 469, 1990.

[9] J. Engelfriet. Branching processes of Petri nets. *Acta Informatica*, 28(6), 1991.

[10] H. J. Genrich and K. Lautenbach. System modelling with high-level Petri nets. *Theoretical Computer Science*, 13, 1981.

[11] J. Gunawardena. Causal automata. *Theoretical Computer Science*, 101(2), 1992.

[12] J. Katoen. *Quantitative and Qualitative Extensions of Event Structures*. PhD thesis, Enschede: Centre for Telematics and Information Technology, 1996.

[13] R. Langerak. Bundle event structures: A non-interleaving semantics for lotos. *FORTE '92*, 1993.

[14] U. Montanari and F. Rossi. Contextual nets. *Acta Informatica*, 32(6), 1995.

[15] M. Nielsen, G. Plotkin, and G. Winskel. Petri Nets, Event Structures and Domains, Part 1. *Theoretical Computer Science*, 13, 1981.

[16] G. M. Pinna and A. Poigné. On the nature of events: another perspective in concurrency. *Theoretical Computer Science*, 138(2), 1995.

[17] R. J. van Glabbeek and G. D. Plotkin. Configuration structures, event structures and Petri nets. *Theoretical Computer Science*, 410(41), 2009.

[18] G. Winskel. Event Structures. *Petri Nets: Central Models and Their Properties*, LNCS 255, 1987.