Data and Process Resonance
Identifier Soundness for Models of Information Systems

Jan Martijn E. M. van der Werf[1(✉)], Andrey Rivkin[2], Artem Polyvyanyy[3], and Marco Montali[2]

[1] Utrecht University, Princetonplein 5, 3584 Utrecht, CC, The Netherlands
j.m.e.m.vanderwerf@uu.nl
[2] Free University of Bozen-Bolzano, piazza Domenicani 3, 39100 Bolzano, Italy
{rivkin,montali}@inf.unibz.it
[3] The University of Melbourne, Parkville, VIC 3010, Australia
artem.polyvyanyy@unimelb.edu.au

Abstract. A model of an information system describes its processes and how these processes manipulate data objects. Object-aware extensions of Petri nets focus on modeling the life-cycle of objects and their interactions. In this paper, we focus on Petri nets with identifiers, where identifiers are used to refer to objects. These objects should "behave" well in the system from inception to termination. We formalize this intuition in the notion of *identifier soundness*, and show that although this property is undecidable in general, useful subclasses exist that guarantee identifier soundness by construction.

Keywords: Information System · Verification · Data and Processes

1 Introduction

Petri nets are widely used to describe distributed systems capable of expanding their resources indefinitely [26]. A *Petri net* describes passive and active components of a system, modeled as places and transitions, respectively. The active components of a Petri net communicate asynchronously with each other via local interfaces. Thus, state changes in a Petri net system have local causes and effects and are modeled as tokens consumed, produced, or transferred by the transitions of the system. A *token* is often used to denote an *object* in the physical world the system manipulates or a *condition* that can cause a state change in the system.

Petri nets with identifiers extend classical Petri nets to provide formal means to relate tokens to objects. Every token in such a Petri net is associated with a vector of identifiers, where each identifier uniquely identifies a data object. Consequently, active components of a Petri net with identifiers model how groups of objects, either envisioned or those existing in the physical world, can be consumed, produced, or transferred by the system.

It is often desirable that modeled systems are correct. Many criteria have been devised for assessing the correctness of systems captured as Petri nets. Those

L. Bernardinello and L. Petrucci (Eds.): PETRI NETS 2022, LNCS 13288, pp. 369–392, 2022.
https://doi.org/10.1007/978-3-031-06653-5_19

criteria target models of systems that use tokens to represent conditions that control state changes. In other words, they can be used to verify the correctness of processes the systems can support and not of the object manipulations carried out within those processes. Such widely-used criteria include boundedness [18], liveness [12], and soundness [1]. The latter one, for instance, ensures that a system modeled as a *workflow net*, a special type of a Petri net used to encode workflow at an organization, has a terminal state that can be distinguished from other states of the modeled system, the system can always reach the terminal state, and every transition of the system can in principle be enabled and, thus, be executed by the system.

Real-world systems, such as information systems [25], are characterized by processes that manipulate objects. For instance, an online retailer system manipulates products, invoices, and customer records. However, correctness criteria that address both aspects, that is, the processes and data, are understood less well. Hence, the paper at hand to address the gap.

In this paper, we propose a correctness criterion for Petri nets with identifiers that combines the checks of the soundness of the system's processes with the soundness of object manipulations within those processes. Intuitively, objects of a specific type are correctly manipulated by the system if every object instance of that type, characterized by a unique identifier, can "leave" the system, that is, a dedicated transition of the system can consume it, and once that happens, no references to that object instance remain in the system. When a system achieves this harmony for its processes and all data object types, we say that the system is *identifier sound*, or, alternatively, that the data and processes of the system are in *resonance*. Specifically, this paper makes these contributions:

- It motivates and defines the notion of *identifier soundness* for checking correctness of data object manipulations in processes of a system; and
- It discusses aspects related to *decidability of identifier soundness* in the general case and for certain restricted, but still useful, classes of systems.

The paper proceeds as follows. The next section introduces concepts and notions required to support subsequent discussions. Section 3 introduces typed Petri nets with identifiers, a model for modeling distributed systems whose state is defined by objects the system manipulates. Section 4 presents the notion of identifier soundness, including a proof that the notion is in general undecidable. Section 5 discusses several classes of systems for which identifier soundness is guaranteed by construction. Finally, the paper concludes with a discussion of related work (Sect. 6) and conclusions (Sect. 7).

2 Preliminaries

Let S and T be sets. The powerset of S is denoted by $\wp(S) = \{S' \mid S' \subseteq S\}$ and $|S|$ denotes the cardinality of S. Given a relation $R \subseteq S \times T$, its range is defined by $\text{RNG}(R) = \{y \in T \mid \exists x \in S : (x, y) \in R\}$. A *multiset* m over S is a mapping of the form $m : S \to \mathbb{N}$, where $\mathbb{N} = \{0, 1, 2, \ldots\}$ denotes the set of

natural numbers. For $s \in S$, $m(s) \in \mathbb{N}$ denotes the number of times s appears in the multiset. We write s^n if $m(s) = n$. For $x \notin S$, we assume $m(x) = 0$. We use $S^{\oplus}$ to denote the set of all finite multisets over S and $\emptyset$ to denote the *empty multiset*. The support of $m \in S^{\oplus}$ is the set of elements that appear in m at least once: $supp(m) = \{s \in S \mid m(s) > 0\}$. Given two multisets m_1 and m_2 over S: *(i)* $m_1 \subseteq m_2$ (resp., $m_1 \subset m_2$) iff $m_1(s) \leq m_2(s)$ (resp., $m_1(s) < m_2(s)$) for each $s \in S$; *(ii)* $(m_1 + m_2)(s) = m_1(s) + m_2(s)$ for each $s \in S$; *(iii)* if $m_1 \subseteq m_2$, $(m_2 - m_1)(s) = m_2(s) - m_1(s)$ for each $s \in S$; and *(iv)* $|m| = \sum_{s \in S} m(s)$. A *sequence* over S of length $n \in \mathbb{N}$ is a function $\sigma : \{1, \ldots, n\} \to S$. If $n > 0$ and $\sigma(i) = a_i$, for $1 \leq i \leq n$, we write $\sigma = \langle a_1, \ldots, a_n \rangle$. The length of a sequence σ is denoted by $|\sigma|$. The sequence of length 0 is called the *empty sequence*, and is denoted by ϵ. The set of all finite sequences over S is denoted by S^*. We write $a \in \sigma$ if there is $1 \leq i \leq |\sigma|$ such that $\sigma(i) = a$. Projection of sequences on a set T is defined inductively by $\epsilon_{|T} = \epsilon$, $(\langle a \rangle \cdot \sigma)_{|T} = \langle a \rangle \cdot \sigma_{|T}$ if $a \in T$ and $\langle a \rangle \cdot \sigma_{|T} = \sigma_{|T}$ otherwise, where $\cdot$ is the sequence concatenation operator. Renaming a sequence with an injective function $r : S \to T$ is defined inductively by $\rho_r(\epsilon) = \epsilon$, and $\rho_r(\langle a \rangle \cdot \sigma) = \langle r(a) \rangle \cdot \rho_r(\sigma)$. Renaming is extended to multisets of sequences as follows: given a multiset $m \in (S^*)^{\oplus}$, we define $\rho_r(m) = \sum_{\sigma \in supp(m)} \sigma(m) \cdot \rho_r(\sigma)$. For example, $\rho_{\{x \mapsto a, y \mapsto b\}}([\langle x, y \rangle^3]) = [\langle a, b \rangle^3]$.

Labeled Transition Systems. To model the behavior of a system, we use *labeled transition systems*. Given a finite set A of (action) labels, a *(labeled) transition system* (LTS) over A is a tuple $\Gamma = (S, A, s_0, \rightarrow)$, where S is a (possibly infinite) set of *states*, s_0 is the *initial state* and $\rightarrow \subseteq (S \times (A \cup \{\tau\}) \times S)$ is the *transition relation*, where $\tau \notin A$ denotes the silent action [11]. In what follows, we write $s \xrightarrow{a} s'$ for $(s, a, s') \in \rightarrow$. Let $r : A \to (A' \cup \{\tau\})$ be an injective, total function. Renaming Γ with r is defined as $\rho_r(\Gamma) = (S, A', s_0, \rightarrow')$ with $(s, r(a), s') \in \rightarrow'$ iff $(s, a, s') \in \rightarrow$. Given a set T, hiding is defined as $\hat{\mathtt{H}}_T(\Gamma) = \rho_h(\Gamma)$ with $h : A \to A \cup \{\tau\}$ such that $h(t) = \tau$ if $t \in T$ and $h(t) = t$ otherwise. Given $a \in A$, $p \mathrel{-\!\!\stackrel{a}{\!\!\rightarrow}} q$ denotes a *weak transition relation* that is defined as follows: *(i)* $p \mathrel{-\!\!\stackrel{a}{\!\!\rightarrow}} q$ iff $p(\xrightarrow{\tau})^* q_1 \xrightarrow{a} q_2(\xrightarrow{\tau})^* q$; *(ii)* $p \mathrel{-\!\!\stackrel{\tau}{\!\!\rightarrow}} q$ iff $p(\xrightarrow{\tau})^* q$. Here, $(\xrightarrow{\tau})^*$ denotes the reflexive and transitive closure of $\xrightarrow{\tau}$.

Definition 1 (Strong and weak bisimulation). *Let $\Gamma_1 = (S_1, A, s_{01}, \rightarrow_1)$ and $\Gamma_2 = (S_2, A, s_{02}, \rightarrow_2)$ be two LTSs. A relation $R \subseteq (S_1 \times S_2)$ is called a* strong simulation*, denoted as $\Gamma_1 \prec_R \Gamma_2$, if for every pair $(p, q) \in R$ and $a \in A \cup \{\tau\}$, it holds that if $p \xrightarrow{a}_1 p'$, then there exists $q' \in S_2$ such that $q \xrightarrow{a}_2 q'$ and $(p', q') \in R$. Relation R is a* weak simulation*, denoted by $\Gamma_1 \preccurlyeq_R \Gamma_2$, iff for every pair $(p, q) \in R$ and $a \in A \cup \{\tau\}$ it holds that if $p \xrightarrow{a}_1 p'$, then either $a = \tau$ and $(p', q) \in R$, or there exists $q' \in S_2$ such that $q \mathrel{-\!\!\stackrel{a}{\!\!\rightarrow}}_2 q'$ and $(p', q') \in R$.*

R is called a strong (weak) bisimulation*, denoted by $\Gamma_1 \sim_R \Gamma_2$ ($\Gamma_1 \approx_R \Gamma_2$) if both $\Gamma_1 \prec \Gamma_2$ ($\Gamma_1 \preccurlyeq_R \Gamma_2$) and $\Gamma_2 \prec_{R^{-1}} \Gamma_1$ ($\Gamma_2 \preccurlyeq_{R^{-1}} \Gamma_1$). The relation is called* rooted *iff $(s_{01}, s_{02}) \in R$. A rooted relation is indicated with a superscript ^{r}.* ◁

Petri Nets. A weighted Petri net is a 4-tuple (P, T, F, W) where P and T are two disjoint sets of *places* and *transitions*, respectively, $F \subseteq ((P \times T) \cup (T \times P))$

is the *flow relation*, and $W : F \to \mathbb{N}^+$ is a *weight function*. For $x \in P \cup T$, we write ${}^\bullet x = \{y \mid (y,x) \in F\}$ to denote the *preset* of x and $x^\bullet = \{y \mid (x,y) \in F\}$ to denote the *postset* of x. We lift the notation of preset and postset to sets element-wise. If for a Petri net no weight function is defined, we assume $W(f) = 1$ for all $f \in F$. A *marking* of N is a multiset $m \in P^\oplus$, where $m(p)$ denotes the number of *tokens* in place $p \in P$. If $m(p) > 0$, place p is called *marked* in marking m. A *marked Petri net* is a tuple (N, m) with N a weighted Petri net with marking m. A transition $t \in T$ is enabled in (N, m), denoted by $(N, m)[t\rangle$ iff $W((p,t)) \leq m(p)$ for all $p \in {}^\bullet t$. An enabled transition can *fire*, resulting in marking m' iff $m'(p) + W((p,t)) = m(p) + W((t,p))$, for all $p \in P$, and is denoted by $(N, m)[t\rangle(N, m')$. We lift the notation of firings to sequences. A sequence $\sigma \in T^*$ is a *firing sequence* iff $\sigma = \epsilon$, or markings $m_0, \ldots, m_n$ exist such that $(N, m_{i-1})[\sigma(i)\rangle(N, m_i)$ for $1 \leq i \leq |\sigma| = n$, and is denoted by $(N, m_0)[\sigma\rangle(N, m_n)$. If the context is clear, we omit the weighted Petri net N. The set of reachable markings of (N, m) is defined by $\mathcal{R}(N, m) = \{m' \mid \exists \sigma \in T^* : m[\sigma\rangle m'\}$. The semantics of a marked Petri net (N, m) with $N = (P, T, F, W)$ is defined by the LTS $\Gamma_{N,m} = (P^\oplus, T, m_0, \to)$ with $(m, t, m') \in \to$ iff $m[t\rangle m'$.

Workflow Nets. A *workflow net* (WF-net for short) is a tuple $N = (P, T, F, W, in, out)$ such that: *(i)* (P, T, F, W) is a weighted Petri net; *(ii)* $in, out \in P$ are the source and sink place, respectively, with ${}^\bullet in = out^\bullet = \emptyset$; *(iii)* every node in $P \cup T$ is on a directed path from in to out. N is called *k-sound* for some $k \in \mathbb{N}$ iff *(i)* it is proper completing, i.e., for all reachable markings $m \in \mathcal{R}(N, [in^k])$, if $[out^k] \subseteq m$, then $m = [out^k]$; *(ii)* it is weakly terminating, i.e., for any reachable marking $m \in \mathcal{R}(N, [in^k])$, the final marking is reachable, i.e., $[out^k] \in \mathcal{R}(N, m)$; and *(iii)* it is quasi-live, i.e., for all transitions $t \in T$, there is a marking $m \in \mathcal{R}(N, [in])$ such that $m[t\rangle$. The net is called *sound* if it is 1-sound. If it is k-sound for all $k \in \mathbb{N}$, it is called *generalized sound* [15].

3 Typed Petri Nets with Identifiers

Processes and data are highly intertwined: processes manipulate data objects. These manipulations can be complex and involve multiple objects. As an example, consider a retailer shop with three types of objects: *products* that are sold through the shop, and *customers* that can order these products, which is supported through an *order* process. Here, object relations can be many-to-many: e.g., a product can be ordered for many customers and the same customer can order many products. Relations can also be one-to-many, e.g., an order is always for a single customer, but a customer can have many orders. In addition, objects may have their own life cycle, which can be considered to be a process itself. For example, a product may temporarily be unavailable, or customers may be blocked by the shop, disallowing them to order products.

Different approaches have been studied to model and analyse such models that combine objects and processes. For example, data-aware Proclets [7] allow to describe the behavior of individual artifacts and their interactions. Another

approach is followed in ν-PN [28], in which tokens can carry a single identifier [27]. These identifiers can be used to reference entities in an information model. However, referencing a fact composed of multiple entities is not possible in ν-PNs. In this paper, we study *typed Petri nets with identifiers* (t-PNIDs), which build upon ν-PNs [28] by extending tokens to carry vectors of identifiers [25,31]. Vectors, represented by multisets, have the advantage that a single token can represent multiple objects or entities at the same time, such as for which customer an order is. Identifiers are typed, i.e., the countable, infinite set of identifiers is partitioned into a set of types, such that each type contains a countable, infinite set of identifiers. Variables are typed as well and can only refer to identifiers of the associated type.

Definition 2 (Identifier Types). *Let $\mathcal{I}$, Λ, and $\mathcal{V}$ denote countable, infinite sets of identifiers, type labels, and variables, respectively. We define:*

- *the* domain assignment *function $I : \Lambda \to \wp(\mathcal{I})$, such that $I(\lambda_1)$ is an infinite set, and $I(\lambda_1) \cap I(\lambda_2) \neq \emptyset$ implies $\lambda_1 = \lambda_2$ for all $\lambda_1, \lambda_2 \in \Lambda$;*
- *the* id typing *function $type_I : \mathcal{I} \to \Lambda$ s.t. if $type_I(\mathtt{id}) = \lambda$, then $\mathtt{id} \in I(\lambda)$;*
- *a* variable typing *function $type_{\mathcal{V}} : \mathcal{V} \to \Lambda$, prescribing that $x \in \mathcal{V}$ can be substituted only by values from $I(type_{\mathcal{V}}(x))$.*

When clear from the context, we omit the subscripts of type. ◁

In a t-PNID, each place is annotated with a label, called the *place type*. A place type is a vector of types, indicating types of identifier tokens the place can carry. A place with an empty place type, represented by the empty vector, is a classical Petri net place carrying indistinguishable (black) tokens. Each arc is inscribed with a multiset of vectors of identifiers, such that the type of each variable coincides with the place types. This allows to model situations in which a transition may require multiple tokens with different identifiers from the same place.

Definition 3 (Petri net with identifiers). *A* Typed Petri net with identifiers *(t-PNID) N is a tuple (P, T, F, α, β), where:*

- *(P, T, F) is a Petri net;*
- *$\alpha : P \to \Lambda^*$ is the* place typing function*;*
- *$\beta : F \to (\mathcal{V}^*)^{\oplus}$ defines for each flow a multiset of* variable vectors *such that $\alpha(p) = type(x)$ for any $x \in supp(\beta((p,t)))$ and $type(y) = \alpha(p')$ for any $y \in supp(\beta((t,p')))$ where $t \in T$, $p \in {}^\bullet t$, $p' \in t^\bullet$;*

Figure 1 shows a t-PNID, N_{rs}, of the retailer shop. Each place is colored according to its identifier type. In N_{rs}, places *product* and *unavailable product* are annotated with a vector $\langle product \rangle$, i.e., these places contain tokens that carry only a single identifier of type *product*. Places *customer* and *blocked customer* have type $\langle customer \rangle$. All other places, except for place p, are labeled with type $\langle order \rangle$. Place p maintains the relation between orders and customers, and is typed $\langle order, customer \rangle$, i.e., tokens in this place are identifier vectors of size 2. N_{rs} uses three variables: x for *product*, y for *order* and z for *customer*.

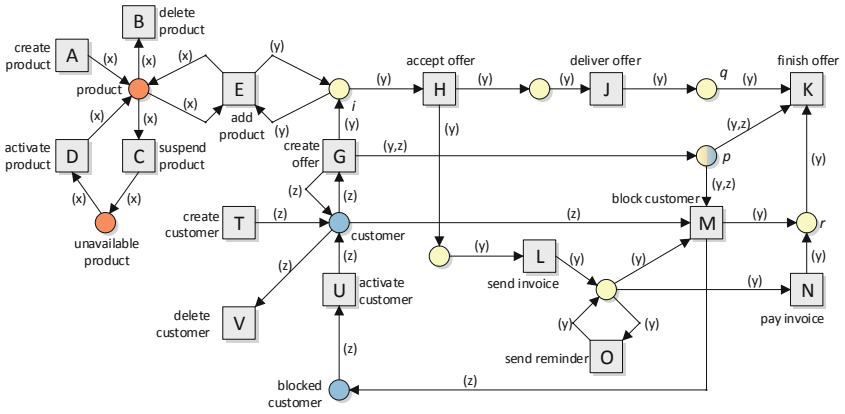


Fig. 1. t-PNID for the retailer shop with types products, customers and orders. Each place is colored according to its type. Place p carries pairs of identifiers: an order and a customer.

A marking of a t-PNID is the configuration of tokens over the set of places. Each token in a place should be of the correct type, i.e., the vector of identifiers carried by a token in a place should match the corresponding place type. All possible vectors of identifiers a place may carry is defined by the set $\mathtt{C}(p)$.

Definition 4 (Marking). *Given a t-PNID* $N = (P, T, F, \alpha, \beta)$*, and place* $p \in P$*, its* id set *is* $\mathtt{C}(p) = \prod_{1 \leq i \leq |\alpha(p)|} I(\alpha(p)(i))$*. A* marking *is a function* $m \in \mathbb{M}(M)$*, with* $\mathbb{M}(M) = P \to (\Lambda^*)^{\oplus}$*, such that* $m(p) \in C(p)^{\oplus}$*, for each place* $p \in P$*. The set of identifiers used in* M *is denoted by* $Id(M) = \bigcup_{p \in P} \text{RNG}(supp(M(p)))$*. The pair* (N, M) *is called a* marked t-PNID.

To define the semantics of a t-PNID, the variables need to be valuated with identifiers. In Fig. 1, transition G uses variable y to create an identifier of type *order*, whereas transition K uses the same variable to remove an identifier from the marking.

Definition 5 (Variable sets). *Given a t-PNID* $N = (P, T, F, \alpha, \beta)$*,* $t \in T$ *and* $\lambda \in \Lambda$*, we define the following sets of variables:*

- input variables *as* $In(t) = \bigcup_{x \in \beta((p,t)), p \in {}^\bullet t} \text{RNG}(supp(x))$*;*
- output variables *as* $Out(t) = \bigcup_{x \in \beta((t,p)), p \in t^\bullet} \text{RNG}(supp(x))$*;*
- variables *as* $Var(t) = In(t) \cup Out(t)$*;*
- emitting variables *as* $Emit(t) = Out(t) \setminus In(t)$*;*
- collecting variables *as* $Collect(t) = In(t) \setminus Out(t)$*;*
- emitting transitions *as* $E_N(\lambda) = \{t \mid \exists x \in Emit(t) \wedge type(x) = \lambda\}$*;*
- collecting transitions *as* $C_N(\lambda) = \{t \mid \exists x \in Collect(t) \wedge type(x) = \lambda\}$*;*
- types in N *as* $type(N) = \{\vec{\lambda} \mid \exists p \in P : \vec{\lambda} \in \alpha(p)\}$*.* ◁

As customary in colored Petri nets, the firing of a transition requires a *binding* that valuates variables to identifiers. The binding is used to inject new fresh data into the net via variables that emit identifiers, i.e., via variables that appear only on the output arcs of that transition. We require bindings to be an injection, i.e., no two variables within a binding may refer to the same identifier. Note that in this definition, freshness of identifiers is local to the marking, i.e., disappeared identifiers may be reused, as it does not hamper the semantics of the t-PNID. Our semantics allow the use of well-ordered sets of identifiers, such as the natural numbers, as used in [25,27] to ensure that identifiers are globally new. Here we assume local freshness over global freshness.

Definition 6 (Firing rule). *Given a marked t-PNID* (N, M) *with* $N = (P, T, F, \alpha, \beta)$*, a* binding *for transition* $t \in T$ *is an injective function* $\psi : \mathcal{V} \to \mathcal{I}$ *such that* $type(v) = type(\psi(v))$ *and* $\sigma(v) \notin Id(M)$ *iff* $v \in Emit(t)$*. Transition* t *is* enabled *in* (N, M) *under binding* ψ*, denoted by* $(N, M)[t, \psi\rangle$ *iff* $\rho_\psi(\beta(p, t)) \leq M(p)$ *for all* $p \in {}^\bullet t$*. Its firing results in marking* M'*, denoted by* $(N, M)[t, \psi\rangle(N, M')$*, such that* $M'(p) + \rho_\psi(\beta(p, t)) = M(p) + \rho_\psi(\beta(t, p))$*.* ◁

Again, the firing rule is inductively extended to sequences $\eta \in (T \times (\mathcal{V} \to \mathcal{I}))^*$. A marking M' is *reachable* from M if there exists $\eta \in (T \times (\mathcal{V} \to \mathcal{I}))^*$ s.t. $M[\eta\rangle M'$. We denote with $\mathcal{R}(N, M)$ the set of all markings reachable from M for (N, M).

The execution semantics of a t-PNID is defined as an LTS that accounts for all possible executions starting from a given initial marking.

Definition 7. *Given a marked t-PNID* (N, M_0) *with* $N = (P, T, F, \alpha, \beta)$*, its induced transition system is* $\Gamma_{N,M_0} = (\mathbb{M}(N), (T \times (\mathcal{V} \to \mathcal{I})), M_0, \to)$ *with* $M \xrightarrow{(t,\sigma)} M'$ *iff* $M[t, \sigma\rangle M'$*.*

t-PNIDs are a vector-based extension of ν-PNs. In other words, a ν-PN can be translated into a strongly bisimilar t-PNID with a single type, and all place types are of length of at most 1.

Lemma 1. *For any* ν*-PN there exists a single-typed t-PNID such that the two nets are strongly rooted bisimilar.* ◁

As a result, decidability of reachability for ν-PNs transfers to t-PNIDs [28].

Proposition 1. *Reachability is undecidable for t-PNIDs.* ◁

4 Correctness Criteria for t-PNIDs

Many criteria have been devised for assessing the correctness of systems captured as Petri nets. Traditionally, Petri net-based criteria focus on the correctness of processes the systems can support. Enriching the formalism with ability to capture object manipulation while keeping analyzability is a delicate balancing

act. Therefore, object manipulations can only be captured if these are reflected in the token game of the net.

For t-PNIDs, correctness criteria can be categorized as system-level and as object-level. Criteria on the system-level focus on traditional Petri net-based criteria to assess the system as a whole, whereas criteria on object-level address correctness of individual objects represented by identifiers.

4.1 Correctness Criteria on System-Level

System-level properties address the overall behavior of the system. For example, liveness is a typical system-level property. It expresses that any transition is always eventually enabled again. As such, a live system guarantees that its activities cannot eventually become unavailable and never recover again.

Definition 8 (Liveness). *A marked t-PNID* $((P, T, F, \alpha, \beta), M_0)$ *is* live *iff for every marking* $M \in \mathcal{R}(N, M_0)$ *and every transition* $t \in T$*, there exist a marking* $M' \in \mathcal{R}(N, M)$ *and a binding* $\psi : \mathcal{V} \to \mathcal{I}$ *such that* $M'[t, \psi\rangle$. ◁

Boundedness expresses that the reachability graph of a Petri net is finite, i.e., that there are finitely many tokens in the system. Thus, it is a typical system-level property. Many systems have a dynamic number of simultaneously active objects. Designers often do not want to limit themselves on the maximum number of active objects. Consequently, many systems are unbounded by design. Similar to ν-PN, we differentiate between various types of boundedness. *Boundedness* expresses that the number of tokens in any reachable place does not exceed a given bound, whereas *width-boundedness* expresses that the modeled system has a bound on the number of simultaneously active objects. Notice that a width-bounded net may be unbounded if it contains infinitely many tokens referring to finitely many available objects.

Definition 9 (Bounded, width-bounded). *Let* $((P, T, F, \alpha, \beta), M_0)$ *be a marked t-PNID. A place* $p \in P$ *is called:*

- bounded *if there is* $k \in \mathbb{N}$ *such that* $|M(p)| \leq k$ *for all* $M \in \mathcal{R}(N, M_0)$*;*
- width-bounded *if there is* $k \in \mathbb{N}$ *such that* $|Id(M)| \leq k$ *for all* $M \in \mathcal{R}(N, M_0)$*;*

If all places in (N, M_0) *are (width-) bounded,* (N, M_0) *is called (width-) bounded.* ◁

4.2 Correctness Criteria on Object-Level

An object-level property assesses the correctness of individual objects. In t-PNIDs, identifiers can be seen as references to objects: if two tokens carry the same identifier, they refer to the same object. The projection of an identifier on the complete reachability graph of a t-PNID represents the life-cycle of an identifier. Whereas boundedness of a t-PNID implies that states in its reachability graph of the whole system are bounded, *depth-boundedness* expresses that for

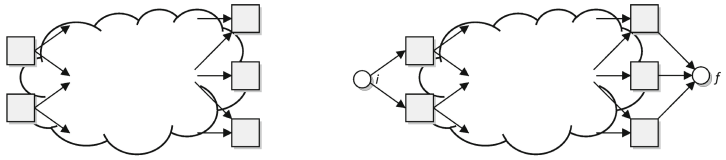


Fig. 2. A transition-bordered WF-Net and its closure for soundness [17].

each identifier the number of tokens carrying that identifier is bounded. In other words, if a t-PNID is depth-bounded, the complete system may be unbounded, but the life-cycle of each individual identifier is finite.

Definition 10 (Depth-boundedness). *Let* $((P, T, F, \alpha, \beta), M_0)$ *be a marked t-PNID. A place* $p \in P$ *is called* depth-bounded *if there is* $k \in \mathbb{N}$ *such that* $M(p)(\vec{\mathtt{id}}) \leq k$ *for all* $\vec{\mathtt{id}} \in \mathcal{I}^*$, $M \in \mathcal{R}(N, M_0)$, *and* $\vec{\mathtt{id}} \in \mathtt{C}(p)$ *with* $\mathtt{id} \in \vec{\mathtt{id}}$. *If all places are depth-bounded,* (N, M_0) *is called depth-bounded.* ◁

Depth-boundedness is undecidable for ν-PNs [28] and thus also for t-PNIDs.

Proposition 2. *Depth-boundedness is undecidable for t-PNIDs.* ◁

Each type has a life-cycle. Intuitively, an object of a given type "enters" the system via an emitter that creates a unique identifier that refers to the object. The identifier remains in the system, until the object "leaves" the system by firing a collecting transition (that binds to the identifier and consumes it). Hence, once that transition fires, there should be no remaining tokens referring to the removed object. The process of a type is a model that describes all possible paths allowed for a type. It can be represented as a transition-bordered WF-net [17]. Instead of a sink and source place, a transition-bordered WF-net has transitions that represent the start and finish of a process. A transition-bordered WF-net is sound, if its closure is sound. As shown in Fig. 2, its closure is constructed by creating a new source place i s.t. each emitting transition consumes from i, and a new sink place f s.t. each collecting transition produces in f. Consider in t-PNID N_{rs} of Fig 1, identifier type *order*. Its life cycle starts with transition G. Transitions K and V are two transitions that may remove the last reference to an *order*. Soundness of a transition-bordered WF-net would require that firing transition K or transition V would result in the final marking. In the remainder of this section, we develop this intuition into the concept of identifier soundness.

Soundness constitutes three properties: proper completion, weak termination and quasi-liveness. Similarly to [15], we focus on the first two properties. *Proper completion* states that if a marking covers the final marking, it is the final marking. In other words, as soon as a token is produced in the final place, all other places are empty. Following the idea of transition-bordered WF-nets, identifiers should have a similar behavioral property: once an identifier is consumed by a collector, the identifier should be removed from the marking.

Definition 11 (Proper type completion). *Given type $\lambda \in \Lambda$, a marked t-PNID (N, M_0) is called* proper λ-completing *iff for all $t \in C_N(\lambda)$, bindings $\psi : \mathcal{V} \to \mathcal{I}$ and markings $M, M' \in \mathcal{R}(N, M_0)$, if $M[t, \psi\rangle M'$, then for all identifiers $\mathtt{id} \in \text{RNG}(\psi|_{Collect(t)}) \cap Id(M)$ and $type(\mathtt{id}) = \lambda$, it holds that $\mathtt{id} \notin Id(M')$.*[1] ◁

As an example, consider t-PNID N_{rs} in Fig. 1. For type *customer*, we have $C_{N_{rs}}(customer) = \{K, V\}$. In the current – empty – marking, transition T is enabled with binding $\psi = \{y \mapsto \mathtt{o}, z \mapsto \mathtt{c}\}$, which results in marking M with $M(customer) = [\mathtt{c}]$. Next, transitions G, H, J, L and N can fire, all using the same binding, producing marking M' with $M'(p) = [\mathtt{o}, \mathtt{c}]$, $M'(customer) = [\mathtt{c}]$ and $M'(q) = M'(r) = [\mathtt{c}]$. Hence, transition K is enabled with binding ψ. However, firing K with ψ results in marking M'' with $M''(customer) = [\mathtt{c}]$, while $\psi(z) = \mathtt{c}$. Hence, N_{rs} is not properly *customer*-completing.

Weak termination for a WF-net signifies that from any reachable marking, the final marking can be reached. Translated to identifiers, it should always eventually be possible to remove an identifier from a marking.

Definition 12 (Weak type termination). *Given type $\lambda \in \Lambda$, a marked t-PNID (N, M_0) is called* weakly λ-terminating *iff for every $M \in \mathcal{R}(N, M_0)$ and identifier $\mathtt{id} \in I(\lambda)$ such that $\mathtt{id} \in Id(M)$, there exists a marking $M' \in \mathcal{R}(N, M)$ with $\mathtt{id} \notin Id(M')$.* ◁

Identifier soundness combines the two properties of proper type completion and weak type termination: the former ensures that as soon a collector fires for an identifier, the identifier is removed, whereas the latter ensures that it is always eventually possible to remove that identifier.

Definition 13. *A marked t-PNID (N, M_0) is* λ-sound *iff it properly λ-completes and weakly λ-terminates. It is* identifier sound *iff it is λ-sound for every $\lambda \in type(N)$.* ◁

There are two interesting observations that one can make about the identifier soundness property. First, identifier soundness does not imply soundness in the classical sense: any classical net N without types, i.e., $type(N) = \emptyset$, is identifier sound, independently of the properties of N. Second, identifier soundness implies depth-boundedness. In other words, if a t-PNID is identifier sound for all types, it cannot accumulate infinitely many tokens carrying the same identifier.

Lemma 2. *If a t-PNID (N, M_0) is identifier sound, then it is depth-bounded.* ◁

Proof. Suppose that (N, M_0) is identifier sound, but not depth-bounded. Then, at least for one place $p \in P$ and identifier $\mathtt{id} \in \mathtt{C}(p)$ of type $\vec{\lambda}$ there exists an infinite sequence of increasing markings M_i, all reachable in (N, M_0), such that $M_i(p)(\mathtt{id}) < M_{i+1}(p)(\mathtt{id})$. Assume $\lambda \in \vec{\lambda}$. As (N, M_0) weakly λ-terminates, there exists $M' \in \mathcal{R}(N, M)$ such that $\mathtt{u} \notin Id(M')$, where $type(\mathtt{u}) = \lambda$. This means that the above sequence cannot exist as all constituents of $\mathtt{id}$ must be eventually removed. Hence, (N, M_0) is depth-bounded. ⊣

As identifier soundness relies on reachability, it is undecidable.

[1] Here, we constrain ψ only to objects of type λ that are only consumed.

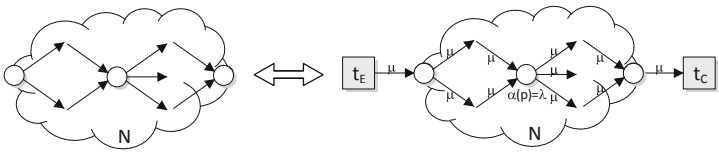


Fig. 3. EC-closure of a WF-net N.

Theorem 1. *Identifier soundness is undecidable for t-PNIDs.* ◁

Proof. Let (N, M_0) be a marked t-PNID. By Definition 13, we need to show that it properly completes and weakly terminates. Since the latter requires a reachability test, it is undecidable by Proposition 1. ⊣

The above theorem also naturally follows from the fact that all non-trivial decision problems are undecidable for Petri nets in which tokens carry pairs of data values (taken from unordered domains) and in which element-wise equality comparisons are allowed over such pairs in transition guards [19].

5 Correctness by Construction

As shown in the previous section, identifier soundness is undecidable. However, we are still interested in ensuring correctness criteria over the modeled system. In this section, we propose a structural approach to taming the undecidability and study sub-classes of t-PNIDs that are identifier sound by construction.

5.1 EC-Closed Workflow Nets

WF-nets are widely used to model business processes. The initial place of the WF-net signifies the start of a *case*, the final place represents the goal state, i.e., the process case completion. A firing sequence from initial state to final state represents the activities that are performed for a single case. Thus, a WF-net describes all possible sequences of a single case. Process engines, like Yasper [14] simulate the execution of multiple cases in parallel by coloring the tokens with the case identifier (a similar idea is used for resource-constrained WF-net variants of ν-PNs in [23]). In other words, they label each place with a case type, and inscribe each arc with a variable. To execute it, the WF-net is closed with an emitter and a collector, as shown in Fig. 3. We generalize this idea to any place label, i.e., any finite sequence of types may be used to represent a case.

Definition 14 (EC-Closure). *Given a WF-net N, place type $\vec{\lambda} \in \Lambda^*$ and a variable vector $\vec{v} \in \mathcal{V}^*$ such that type$(\vec{v}) = \vec{\lambda}$. Its* EC-closure *is a t-PNID $\mathcal{W}(N, \lambda, \vec{v}) = (P_N, T_N \cup \{t_E, t_C\}, F_N \cup \{(t_E, in), (out, t_C)\}, \alpha, \beta)$, with:*

– $\alpha(p) = \vec{\lambda}$ *for all places* $p \in P_N$*;*
– $\beta(f) = \vec{v}^{W(f)}$ *for all flows* $f \in F_N$*, and* $\beta((t_e, in)) = \beta((out, t_c)) = [\vec{v}]$*;* ◁

The EC-closure of a WF-net describes all cases that run simultaneously at any given time. In other words, any reachable marking of the EC-closure is the "sum" of all simultaneous cases. Lemma 3 formalizes this idea by establishing weak bisimulation between the projection on a single case and the original net.

Lemma 3 (Weak bisimulation for each identifier). *Let* N *be a WF-net,* $\vec{\lambda} \in \Lambda^*$ *be a place type and* $\vec{v} \in \mathcal{V}^*$ *be a variable vector s.t.* $type(\vec{v}) = \vec{\lambda}$*. Then, for any* $\mathtt{id} \in \mathcal{I}^{|\vec{\lambda}|}$*,* $\rho_r(\Gamma_{\mathcal{W}(N,\vec{\lambda},\vec{v}),\emptyset}) \approx \Gamma_{N,[in]}$ *with* $r(t,\psi) = r(t)$*, if* $\psi(\vec{v}) = \mathtt{id}$*, and* $r((t,\psi)) = \tau$*, otherwise.* ◁

Proof. Define $R = \{(M, m) \mid \forall p \in P : M(p)(a) = m(p)\}$. We need to show that R is a weak bisimulation. ($\Rightarrow$) Let M, M' and m be such markings that $(M, m) \in R$ and $M[t, \psi\rangle M'$, with $t \in T$ and $\psi : \mathcal{V} \to \mathcal{I}$. By Definition 14, $\psi(\vec{v}) = u$, for some $u \in \mathcal{I}^{|\vec{\lambda}|}$. From the firing rule, we obtain $M'(p) + [u^{W((p,t))}] = M(p) + [u^{W((t,p))}]$, for any $p \in P$. If $u \neq \mathtt{id}$, then $r(t, \psi) = \tau$, and $(M', m) \in R$. If $u = \mathtt{id}$, there exists such marking m' that $m[t\rangle m'$ (since $m(p) = M(p)(\mathtt{id})$ and thus $m(p) \geq W((p,t))$) and $m'(p) + W((p,t)) = M(p)(\mathtt{id}) + W((t,p))$. Then, by construction, $m'(p) = M'(p)(id)$ and $(M', m') \in R$.
($\Leftarrow$) By analogy with the previous argument. ⊣

A natural consequence of this weak bisimulation result is that any EC-closure of a WF-net is identifier sound if and only if the underlying WF-net is sound.

Theorem 2. *Given a WF-Net* N*, if* N *is sound, then* $\mathcal{W}(N, \vec{\lambda}, \vec{v})$ *is identifier sound and live, for any place type* $\vec{\lambda} \in \Lambda^*$ *and variable vector* $\vec{v} \in \mathcal{V}^*$ *with* $type(\vec{v}) = \vec{\lambda}$*.* ◁

Proof. Let $\mathcal{W}(N, \vec{\lambda}, \vec{v}) = (P, T, F, \alpha, \beta)$. By definition of $\mathcal{W}$, $Collect(t) = \emptyset$ for any transition $t \in T \setminus \{t_C\}$. Hence, only transition t_C can remove identifiers, and thus, by construction, $\mathcal{W}$ is properly type completing on all $\lambda \in \vec{\lambda}$.

Next, we need to show that M is weakly type terminating for all types $\lambda \in \vec{\lambda}$. Let $M \in \mathcal{R}(\mathcal{W}, \emptyset)$, with firing sequence $\eta \in (T \times (\mathcal{V} \to \mathcal{I}))^*$, i.e., $M_0[\eta\rangle M$. Let $\mathtt{id} \in \mathtt{C}(p)$ such that $M(p)(\mathtt{id}) > 0$ for some $p \in P$. We then construct a sequence ω by stripping the bindings from ηs.t. it contains only transitions of T. Then, using Lemma 3, we get that $[in][\psi\rangle m$, with $m(p) = M(p)(\mathtt{id})$. Since N is sound, there exists a firing sequence ω' such that $m[\omega'\rangle[out]$. Again by Lemma 3, a firing sequence η' exists such that $M[\eta'\rangle M'$ and $(M', [out]) \in \mathcal{R}(\mathcal{W}, \emptyset)|_{\mathtt{id}}$, where $\mathcal{R}(\mathcal{W}, \emptyset)|_{\mathtt{id}}$ is the set of all reachable markings containing $\mathtt{id}$. Hence, if $M'(p)(\mathtt{id}) > 0$, then $p = out$. Thus, transition t_C is enabled with some binding ψ such that $\psi(\vec{v}) = \mathtt{id}$, and a marking M'' exists such that $M'[t_c, \psi\rangle M''$, which removes all identifiers in $\mathtt{id}$ from M'. Hence, $\mathcal{W}$ is identifier sound.

As transition t_e is always enabled, any transition is live, since N is quasi live. Hence $\mathcal{W}(N, \vec{\lambda}, \vec{v})$ is live. ⊣

5.2 Typed Jackson Nets

A well-studied class of processes that guarantee soundness are block-structured nets. Examples include Process Trees [20], Refined Process Structure Trees [30] and Jackson Nets [13]. Each of the techniques have a set of rules in common from which a class of nets can be constructed that guarantees properties like soundness. In this section, we introduce Typed Jackson Nets (t-JNs), extending the ideas of Jackson Nets [13,17] to t-PNIDs, that guarantee both identifier soundness and liveness. The six reduction rules presented by Murata in [24] form the basis of this class of nets. The rules for t-JNs are depicted in Fig. 4.

Rule 1: **Place Expansion.** The first rule is based on *fusion of a series of places*. As shown in Fig. 4a, a single place p is replaced by two places p_i and p_f that are connected via transition t. All transitions that originally produced in p, produce in p_i in the place expansion, and similarly, the transitions that consumed from place p, now consume from place p_f. In fact, transition t can be seen as a transfer transition: it needs to move tokens from place p_i to place p_f, before the original process can continue. This is also reflected in the labeling of the places: both places have the same place type, and all arcs of transition t are inscripted with $[\vec{\mu}]$, i.e., only consuming and producing a single token in a firing.

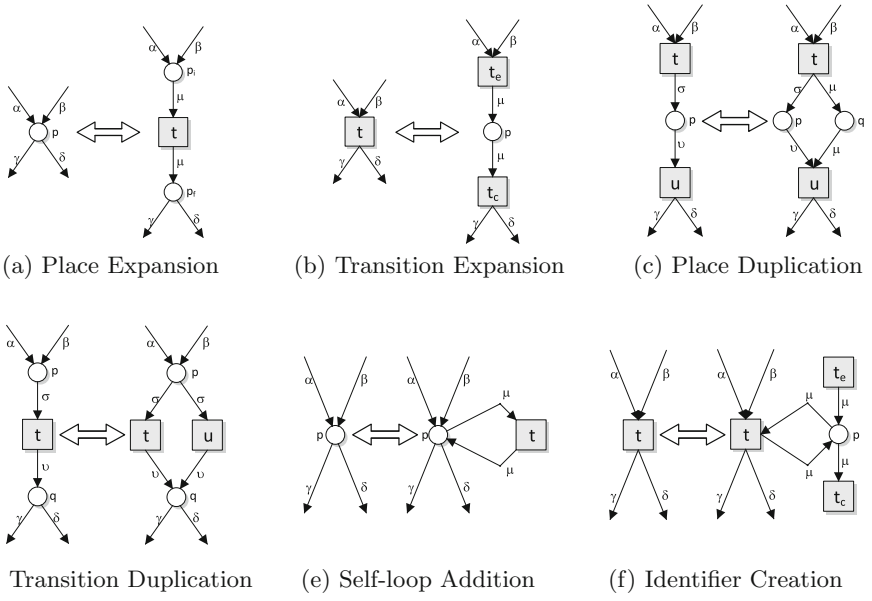
Definition 15 (Place expansion). *Let (N, M) be a marked t-PNID with $N = (P, T, F, \alpha, \beta)$, $p \in P$ be a place and $\vec{\mu} \in \mathcal{V}^*$ be a variable vector s.t. type$(\vec{\mu}) = \alpha(p)$. The* place expanded t-PNID *is defined by $R_{p,\vec{\mu}}(N, M) = ((P', T', F', \alpha', \beta'), M')$, where:*

- $P' = (P \setminus \{p\}) \cup \{p_i, p_f\}$ *with* $p_i, p_f \notin P$; *and* $T' = T \cup \{t\}$ *with* $t \notin T$;
- $F' = (F \setminus (((\{p\} \times p^\bullet) \cup ({}^\bullet t \times \{p\})) \cup ({}^\bullet p \times \{p_i\}) \cup \{(p_i, t), (t, p_f)\} \cup (\{p_f\} \times p^\bullet)$;
- $\alpha'(q) = \alpha(p)$, *if* $q \in \{p_i, p_f\}$, *and* $\alpha'(q) = \alpha(q)$, *otherwise.*
- $\beta'(f) = [\vec{\mu}]$, *if* $f \in \{(p_i, t), (t, p_f)\}$, $\beta'((u, p_i)) = \beta((u, p))$, *if* $u \in {}^\bullet p$, $\beta'((p_f, u)) = \beta((p, u))$, *if* $u \in p^\bullet$, *and* $\beta'(f) = \beta(f)$, *otherwise.*
- $M'(q) = M(q)$ *for all* $q \in P \setminus \{p\}$, $M'(p_f) = 0$, *and* $M'(p_i) = M(p)$. ◁

Inscription $\vec{\mu}$ cannot alter the vector identifier on the tokens, as the type of $\vec{\mu}$ should correspond to both place types $\alpha(p)$ and $\alpha(q)$. Hence, the transition is enabled with the same bindings as any other transition that consumes a token from place p, modulo variable renaming. As such, transition t only "transfers" tokens from place p_i to place p_f. Hence, as the next lemma shows, place expansion yields a weakly bisimilar t-PNID.

Lemma 4. *Let (N, M_0) be a marked t-PNID with $N = (P, T, F, \alpha, \beta)$, $p \in P$ be a place to expand and $\vec{\mu} \in \mathcal{V}^*$ be a variable vector. Then $\Gamma_{N,M_0} \approx^r \hat{\mathrm{H}}_{\{t\}}(\Gamma_{R_{p,\vec{\mu}}(N,M_0)})$, with transition t added by $R_{p,\vec{\mu}}$.* ◁

Proof. Let $(N', M_0') = R_{p,\mu}(N, M_0)$. We define $Q \subseteq \mathbb{M}(N) \times \mathbb{M}(N')$ such that $M(q) = M'(q)$ for all places $q \in P \setminus \{p\}$ and $M'(p_i) + M'(p_f) = M(p)$. Then $(M_0, M_0') \in Q$, hence the relation is rooted.



(a) Place Expansion (b) Transition Expansion (c) Place Duplication

(d) Transition Duplication (e) Self-loop Addition (f) Identifier Creation

Fig. 4. Construction rules of the typed Jackson Nets.

($\Rightarrow$) Let $(M, M') \in Q$ and $M[u, \psi\rangle \bar{M}$, where u is as in Definition 15. We need to show that there exists marking $\bar{M}'$ such that $M' \xrightarrow{(t,\psi)} \bar{M}'$ and $(\bar{M}, \bar{M}') \in Q$.

Suppose $p \notin {}^\bullet u$. Then $M'(q) = M(q)$ and $M(q) \geq \rho_\psi(\beta((p, u)))$ (note that $\rho_\psi(\beta((p, u))) = \rho_\psi(\beta'((p, u))))$. By the firing rule, a marking $\bar{M}'$ exists with $M'[u, \psi\rangle \bar{M}'$, $\bar{M}(q) = \bar{M}'(q)$ for all $q \in P'$. Thus, $(\bar{M}, \bar{M}') \in Q$. Suppose $p \in {}^\bullet u$. Then $\rho_\psi(\beta((p_f, u))) \leq M(p) = M'(p_i) + M'(p_f)$. If $\rho_\psi(\beta(p_f, u))) \leq M'(p_f)$, then transition u is enabled, and a marking $\bar{M}'$ exists with $M'[u, \psi\rangle \bar{M}'$ and $(M', \bar{M}') \in Q$. Otherwise, $\rho_\psi(\beta(p_f, u))) \leq M'(p_i)$. Construct a binding ψ' by letting $\psi'(\mu(i)) = \psi(\beta(p, u)(i))$, for all $1 \leq i \leq |\mu|$. Then, $\rho_{\psi'}(\mu) = \rho_\psi(\beta(p, u))$, and transition t is enabled with binding ψ'. Hence, a marking exists M'' with $M'[t, \psi'\rangle M''$ and $\rho_\psi(\beta((p', u))) \leq M''(p')$. Then $(M, M'') \in Q$ and t is labeled τ in $\hat{\mathfrak{h}}_{\{t\}}(R_{(p,\vec{\mu})}(N))$, and the first case applies on M''. In all cases, $M' \xrightarrow{(t,\psi)} \bar{M}'$.
($\Leftarrow$) By analogy with the previous argument.

Rule 2: **Transition Expansion.** The second rule is transition expansion, which corresponds to Murata's *fusion of series transitions*. As shown in Fig. 4b, transition t is divided into two transitions, t_c that consumes the tokens, and a second transition t_p that produces the tokens. The two transitions are connected with a single, fresh place p. This place can have any type, as long as it does not hamper firing the post transition t_p, i.e., place p should ensure that all variables consumed by t_c, and that are required by t_e are passed. Transition t_c is allowed to emit new identifiers, as long as these are not already produced by t_p.

Definition 16 (Transition expansion). *Let (N, M) be a marked t-PNID with $N = (P, T, F, \alpha, \beta)$, let $t \in T$, and let $\lambda \subseteq \Lambda^*$ and $\mu \in (\mathcal{V} \setminus Emit(t))^*$ such that $type(x) \in \lambda$ and $x \in \mu$, for all $x \in In(t)$, and $type(\mu) = \lambda$. The* transition expanded t-PNID *is defined by $R_{t,\lambda,\mu}(N, M) = ((P', T', F', \alpha', \beta'), M)$, where:*

- $P' = P \cup \{p\}$ *with* $p \notin P$; *and* $T' = (T \setminus \{t\}) \cup \{t_e, t_c\}$ *with* $t_e, t_c \notin T$;
- $F' = (F \setminus (({}^\bullet t \times \{t\}) \cup (\{t\} \times t^\bullet))) \cup ({}^\bullet t \times \{t_e\}) \cup \{(t_e, p), (p, t_c)\} \cup (\{t_c\} \times t^\bullet)$;
- $\alpha'(p) = \lambda$ *and* $\alpha'(q) = \alpha(q)$ *for all* $q \in P$;
- $\beta'(f) = [\mu]$ *if* $f \in \{(t_e, p), (p, t_c)\}$, $\beta'((q, t_e)) = \beta((q, t))$ *for* $q \in {}^\bullet t$, $\beta'((t_c, q)) = \beta((t, q))$ *for* $q \in t^\bullet$, *and* $\beta'(f) = \beta(f)$ *otherwise.* ◁

Transition t_e is allowed to introduce new variables, but key is that inscription μ contains all input variables of transition t. Consequently, μ encodes the binding of transition t. We use this to prove weak bisimulation between a t-PNID and it transition expanded net. The idea behind the simulation relation Q is that the firing of t_e is postponed until t_c fires. In other words, Q encodes that tokens remain in place q until transition t_c fires.

Lemma 5. *Given marked t-PNID (N, M_0) with $N = (P, T, F, \alpha, \beta)$, transition $t \in T$, $\lambda \in \Lambda^*$ and $\mu \in \mathcal{V}^*$. Let t_e, t_c be the transitions added by the expansion. Then $\Gamma_{N,M_0} \approx^r \rho_r(\Gamma_{R_{t,\lambda,\mu}(N,M_0)})$ with $r = \{(t_e, \tau), (t_c, t)\}$.* ◁

Proof. Let $N' = R_{t,\lambda,\mu}(N)$. Define relation $Q \subseteq \mathbb{M}(N) \times \mathbb{M}(N')$ such that $M(q) = M'(q)$ for all places $q \in P \setminus {}^\bullet t$ and $M(q) = M'(q) + \sum_{b \in supp(M'(p))} M'(p)(b) \cdot \rho_{\mu(b)}\beta((q, t))$, where $\mu(b)$ is a shorthand for the binding $\psi : \mathcal{V} \to \mathcal{I}$ with $\psi(x) = b(i)$ iff $\mu(i) = x$ for all $1 \leq i \leq |\mu|$. Then $(M_0, M_0) \in Q$.
($\Rightarrow$) Follows directly from the firing rule, and the construction of μ.
($\Leftarrow$) Let $(M, M') \in Q$ and $M'[u, \psi\rangle \bar{M}'$. We need to show a marking $\bar{M}$ exists such that $M \xrightarrow{(t,\psi)} \bar{M}$ and $(\bar{M}, \bar{M}') \in Q$. If $t_e \neq u \neq t_c$, the statement holds by definition of the firing rule. Suppose $u = t_e$, i.e., $r(u) = \tau$. Hence, we need to show that $(M, \bar{M}') \in Q$. Let $q \in {}^\bullet t$. Since $(M, M') \in Q$, we have $M(q) = M'(q) + \sum_{b \in supp(M'(p))} M'(p)(b) \cdot \rho_{\mu(b)}\beta((q, t))$. By the firing rule, we have $\bar{M}'(p) = M'(p) + [\rho_\psi(\mu)]$ and $M'(q) = \bar{M}'(q) + \rho_\psi(\beta((q, t)))$. By construction, ρ_ψ and $\rho_{\mu([\rho_\psi(\mu)])}$ are identical functions. Rewriting gives $M(q) = \bar{M}'(q) + \sum_{b \in supp(\bar{M}'(p))} M'(p)(b) \cdot \rho_{\mu(b)}\beta((q, t))$, and thus $(M, \bar{M}') \in Q$.

Suppose $u = t_c$, i.e., $r(u) = t$ and $[\rho_\psi(\mu)] \leq M'(p)$. Let $q \in {}^\bullet t$. Then $M(q) = M'(q) + \sum_{b \in supp(M'(p))} M'(p)(b) \cdot \rho_{\mu(b)}\beta((q, t))$. Since $\bar{M}'(p) + [\rho_\psi(\mu)] = M'(p)$ and $\rho_\psi(\beta((q, u))) = \rho_{\mu([\rho_\psi(\mu)])}(\beta((q, u)))$, we obtain $M(q) = M'(q) + \left(\sum_{b \in supp(\bar{M}'(p))} \bar{M}'(p)(b) \cdot \rho_{\mu(b)}\beta((q, t))\right) + \rho_\psi\beta((q, t))$. Hence, a marking $\bar{M}$ exists such that $M[t, \psi\rangle \bar{M}$. Rewriting gives $(\bar{M}, \bar{M}') \in Q$. ⊣

Rule 3: Place Duplication. Whereas the previous two rules only introduced ways to create sequences, the third rule introduces parallelism by duplicating a place, as shown in Fig. 4c. It is based on the *fusion of parallel transitions* reduction rule of Murata. For t-PNIDs, duplicating a place has an additional advantage: as all information required for passing the identifiers is already guaranteed, the duplicated place can have any place type.

Definition 17 (Duplicate place). *Let (N, M) be a marked t-PNID with $N = (P, T, F, \alpha, \beta)$, let $p \in P$, such that $M(p) = \emptyset$, and some transitions $t, u \in T$ exist with ${}^\bullet p = \{t\}$ and $p^\bullet = \{u\}$. Let $\lambda \in \Lambda^*$ and $\mu \in (\mathcal{V} \setminus Emit(u))^*$ such that $type(\mu) = \lambda$. Its* duplicated place t-PNID *is defined by $D_{p,\lambda,\mu}(N, M) = ((P', T, F', \alpha', \beta'), M)$, where:*

- *$P' = P \cup \{q\}$, with $q \notin P$, and $F' = F \cup \{(t, q), (q, u)\}$;*
- *$\alpha' = \alpha \cup \{q \mapsto \lambda\}$ and $\beta' = \beta \cup \{(t, q) \mapsto [\mu], (q, u) \mapsto [\mu]\}$.* ◁

As the duplicated place cannot hamper the firing of any transition, all behavior is preserved by a strong bisimulation on the identity mapping.

Lemma 6. *Given a marked t-PNID (N, M_0) with $N = (P, T, F, \alpha, \beta)$, place $p \in P$, $\lambda \in \Lambda^*$ and $\mu \in \mathcal{V}^*$. Then $\Gamma_{N,M_0} \sim^r \Gamma_{D_{p,\lambda,\mu}(N,M_0)}$.* ◁

Proof. Let $(N', M_0') = D_{p,\lambda,\mu}(N)$. Define relation $Q \subseteq \mathbb{M}(N) \times \mathbb{M}(N')$ such that $(M, M') \in Q$ iff $M(p) = M'(p)$ for all places $p \in P$. The bisimulation relation trivially follows from the firing rule. ⊣

Rule 4: **Transition Duplication.** As already recognized by Berthelot [6], if two transitions have an identical preset and postset, one of these transitions can be removed while preserving liveness and boundedness. Murata's fusion of parallel places is a special case of this rule, requiring that the preset and postset are singletons. For t-JNs, this results in the duplicate transition rule: any transition may be duplicated, as shown in Fig. 4d.

Definition 18 (Duplicate place). *Let (N, M) be a marked t-PNID with $N = (P, T, F, \alpha, \beta)$, and let $t \in T$ such that some places $p, q \in P$ exist with ${}^\bullet t = \{p\}$ and $t^\bullet = \{q\}$. Its* duplicated transition t-PNID *is defined by $D_t(N, M) = ((P, T', F', \alpha, \beta'), M)$, where:*

- *$T' = T \cup \{u\}$, with $t' \notin T$, and $F' = F \cup \{(p, u), (u, q)\}$;*
- *$\beta'((p, u)) = \beta((p, t))$, $\beta((u, q)) = \beta((t, q))$ and $\beta'(f) = \beta(f)$ for all $f \in F$.* ◁

As the above rule only duplicates $t \in T$, the identity relation on markings is a strong rooted bisimulation. The proof is straightforward from the definition.

Lemma 7. *Given a marked t-PNID (N, M_0) with $N = (P, T, F, \alpha, \beta)$, and transition $t \in T$. Then $\Gamma_{N,M_0} \sim^r \rho_{\{(u,t)\}}(\Gamma_{D_t(N,M_0)})$.* ◁

Proof. Let $(N', M_0') = D_t(N)$. Define relation $Q \subseteq \mathbb{M}(N) \times \mathbb{M}(N')$ such that $(M, M') \in Q$ iff $M(p) = M'(p)$ for all places $p \in P$. The bisimulation relation trivially follows from the firing rule. ⊣

Rule 5: **Adding Identity Transitions.** In [6], Berthelot classified a transition t with an identical preset and postset, i.e., ${}^\bullet t = t^\bullet$ as irrelevant, as its firing does not change the marking. The reduction rule *elimination of self-loop transitions* is a special case, as Murata required these sets to be singletons. For t-JNs, adding a self-loop transition is the fifth rule, as shown in Fig. 4e.

Definition 19 (Self-loop addition). *Let $(N, M$ be a marked t-PNID with $N = (P, T, F, \alpha, \beta)$, and let $p \in P$. Its* Self-loop Added t-PNID *is defined by $A_p(N, M) = ((P, T', F', \alpha, \beta'), M)$, where:*

- $T' = T \cup \{t\}$, *with* $t \notin T$, *and* $F' = F \cup \{(p, t), (t, p)\}$;
- $\beta'((p, t')) = \beta'((p, t')) = [\vec{\mu}]$ *with* $\vec{\mu} \in \mathcal{V}^*$ *such that* $type(\mu) = \alpha(p)$, *and* $\beta'(f) = \beta(f)$ *otherwise.* ◁

Similar to the duplicate transition rule, the self-loop addition rule does not introduce new behavior, except for silent self-loops. Hence, the identity relation on markings is a weak rooted bisimulation.

Lemma 8. *Given a marked t-PNID (N, M_0) with $N = (P, T, F, \alpha, \beta)$, and place $p \in P$. Then $\Gamma_{N,M_0} \approx^r \hat{\mathfrak{H}}_{\{t\}}(\Gamma_{A_p(N,M_0)})$ with t the added self-loop transition.* ◁

Proof. Let $(N', M_0') = A_p(N, M_0)$. Define relation $Q \subseteq \mathbb{M}(N) \times \mathbb{M}(N')$ such that $(M, M') \in Q$ iff $M(p) = M'(p)$ for all places $p \in P$. The bisimulation relation trivially follows from the firing rule. ⊣

Rule 6: **Identifier Introduction.** The first five rules preserve the criteria of block-structured WF-nets. Murata's *elimination of self-loop places* states that adding or removing a marked place with identical preset and postset does preserve liveness and boundedness. This rule is often used to introduce a fixed resource to a net, i.e., the number of resources is determined in the initial marking. Instead, identifier introduction adds dynamic resources, as shown in Fig. 4f: transition t_e emits new identifiers as its inscription uses only "new" variables (i.e., those that have not been used in the net), and place p works like a storage of the available resources, which can be removed by firing transition t_c.

Definition 20 (Identifier Introduction). *Let (N, M) be a marked t-PNID with $N = (P, T, F, \alpha, \beta)$, let $t \in T$, let $\vec{\lambda} \in (\Lambda \setminus type(N))^*$ and $\vec{\mu} \in \mathcal{V}^*$ such that $type(\vec{\mu}) = \vec{\lambda}$. The* Identifier introducing t-PNID *is defined by $A_{t,\vec{\lambda},\vec{\mu}}(N, M) = ((P', T', F', \alpha', \beta'), M)$, where:*

- $P' = P' \cup \{p\}$ *and* $T' = T \cup \{t_e, t_c\}$, *for* $p \notin P$ *and* $t_e, t_c \notin T$, *and* $F' = F \cup \{(p, t), (t, p), (t_e, p), (p, t_c)\}$;
- $\alpha' = \alpha \cup \{p \mapsto \vec{\lambda}\}$ *and* $\beta' = \beta \cup \{(p, t) \mapsto [\vec{\mu}], (t, p) \mapsto [\vec{\mu}], (t_e, p) \mapsto [\vec{\mu}], (p, t_c) \mapsto [\vec{\mu}]\}$; ◁

Lemma 9. *Given a marked t-PNID (N, M_0) with $N = (P, T, F, \alpha, \beta)$, transition $t \in T$, $\vec{\lambda} \in \Lambda^*$ and $\vec{\mu} \in (\mathcal{V} \setminus Var(t))^*$. Then $\Gamma_{N,M_0} \approx^r \hat{\mathfrak{H}}_{\{t_e,t_c\}}(\Gamma_{A_{(p)}(N,M_0)})$ with t_e, t_c the added transitions.* ◁

Proof. Let $N' = (P', T', F', \alpha', \beta')$. Define $Q \subseteq \mathbb{M}(N) \times \mathbb{M}(N')$ such that $(M, M') \in Q$ iff $M(p) = M'(p)$ for all $p \in P$.
($\Rightarrow$) Suppose $M[u, \psi\rangle \bar{M}'$. If $t \neq u$, the statement directly follows from the firing rule. If $t = u$, then a marking M'' and binding ψ' exists such that $M'[t_e, \psi'\rangle M''$. Then $M'(p) > \emptyset$, $(M, M'') \in Q$, and $M''[t, \psi\rangle$. Hence, markings $\bar{M}''$ and $\bar{M}'$ exist such that $M''[t, \psi\rangle \bar{M}''[t_c, \psi'\rangle \bar{M}'$, and $(M', \bar{M}''), (M', \bar{M}') \in Q$.
($\Leftarrow$) Follows directly from the firing rule. ⊣

As shown in [28], unbounded places are width-bounded, i.e., can contain an infinite number of identifiers, or depth-bounded, i.e., for each identifier, the number of tokens carrying that identifier is bounded, or both. The place added by the identifier creation rule is by definition width-unbounded, as it has an empty preset. However, it is identifier sound, and thus depth-bounded, as shown in the next lemma.

Lemma 10. *Given a marked t-PNID* (N, M) *with* $N = (P, T, F, \alpha, \beta)$*. Then* $A_{t,\vec{\lambda},\vec{\mu}}(N, M)$ *is identifier sound iff* (N, M) *is identifier sound.* ◁

Proof. Let $(N', M') = A_{t,\vec{\lambda},\vec{\mu}}(N, M)$, and let $p \in P' \setminus P$. Let $\vec{\lambda} \in type(N')$. If $\vec{\lambda} \in type(N)$, it is $\vec{\lambda}$-sound by Lemma 9. Suppose $\vec{\lambda} \notin type(N)$. Then $E_{N'}(\lambda) = \{t_e\}$ and $C_{N'}(\lambda) = \{t_c\}$. Let $\mathtt{a} \in Id(M)$ such that $type(\mathtt{a}) = \vec{\lambda}$. Then an $\vec{id} \in \mathtt{C}(p)$ exists with $\mathtt{a} \in \vec{id}$ and $M(p)(\vec{id}) = 1$. Hence, a binding ψ exists such that $\psi(\vec{\mu}) = \vec{id}$, and transition t_c is enabled with ψ. Let M' be a marking such that $M[t_c, \psi\rangle M'$. Then $\mathtt{a} \notin Id(M')$, which proves the statement. ⊣

Any net that can be reduced to a net with a single transition using these rules is called a typed Jackson Net (t-JN).

Definition 21. *The class of* typed Jackson Nets $\mathcal{T}$ *is inductively defined by:*

- $((\emptyset, \{t\}, \emptyset, \emptyset, \emptyset), \emptyset) \in \mathcal{T}$;
- *if* $(N, M) \in \mathcal{T}$, *then* $R_{p,\vec{\mu}}(N, M) \in \mathcal{T}$;
- *if* $(N, M) \in \mathcal{T}$, *then* $R_{t,\vec{\lambda},\vec{\mu}}(N, M) \in \mathcal{T}$;
- *if* $(N, M) \in \mathcal{T}$, *then* $D_{p,\vec{\lambda},\vec{\mu}}(N, M_0) \in \mathcal{T}$;
- *if* $(N, M) \in \mathcal{T}$, *then* $D_t(N, M) \in \mathcal{T}$;
- *if* $(N, M) \in \mathcal{T}$, *then* $A_p(N, M) \in \mathcal{T}$;
- *if* $(N, M) \in \mathcal{T}$, *then* $A_{t,\vec{\lambda},\vec{\mu}}(N, M) \in \mathcal{T}$. ◁

As any t-JN reduces to a single transition, and each construction rule goes hand in hand with a bisimulation relation, any liveness property is preserved. Consequently, any t-JN is identifier sound and live.

Theorem 3. *Any typed Jackson Net is identifier sound and live.* ◁

Proof. We prove the statement by induction on the structure of t-JNs. The statement holds trivially for the initial net, $((\emptyset, \{t\}, \emptyset, \emptyset, \emptyset), \emptyset)$. Suppose $(N, M) \in \mathcal{T}$. Then for each of the rules, the statement follows directly from the respective bisimulation relations of Lemma 4–9, and the result of Lemma 10. ⊣

To solve the problem of the running example, several solutions exist. One solution is shown in Fig. 5. In this example, the net is a t-JN: starting from transition G, a self loop is added (transition G). The transition is then expanded with transition Z. Place p is then duplicated to create the subnet R. In this way, subnet R only knows of type *offer*, as the connection with *customer* is stored in place p. Consequently, the net is identifier sound, and live.

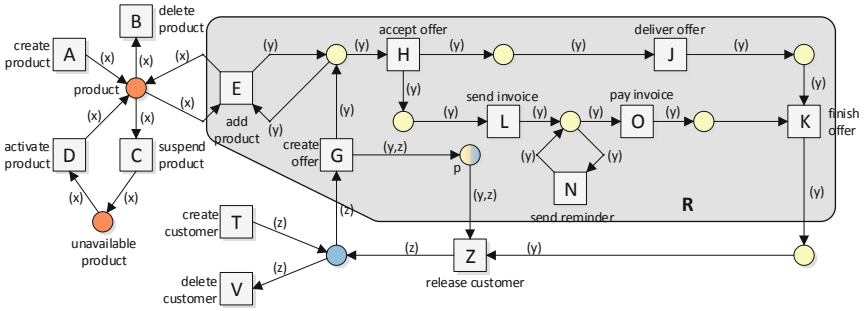


Fig. 5. The example of the retailer shop as a typed Jackson Net.

5.3 Workflow Refinement

A well-known refinement rule is workflow refinement [15]. In a WF-net, any place may be refined with a generalized sound WF-net. If the original net is sound, then the refined net is sound as well. In this section, we present a similar refinement rule. Given a t-PNID, any place may be refined by a generalized sound WF-net. In the refinement, each place is labeled with the place type of the refined place, and all arcs in the WF-net are inscribed with the same variable vector.

Definition 22 (Workflow refinement). *Let* $L = (P_L, T_L, F_L, \alpha_L, \beta_L)$*, be a t-PNID,* $p \in P_L$ *a place, and* $N = (P_N, T_N, F_N, W_N, in_N, out_N)$ *a WF-net.* Workflow refinement *is defined by* $L \oplus_p N = (P, T, F, \alpha, \beta)$*, where:*

- $P = (P_L \setminus \{p\}) \cup P_N$ *and* $T = T_L \cup T_N$*;*
- $F = (F_N \cap ((P \times T) \cup (T \times P))) \cup F_L \cup \{(t, in_N) \mid t \in {}^\bullet p\} \cup \{(out, t) \mid t \in p^\bullet\}$*;*
- $\alpha(q) = \alpha_N(q)$ *for* $q \in P_L \setminus \{p\}$*, and* $\alpha(q) = \alpha(p)$ *for* $q \in P_N$*;*
- $\beta(f) = \beta_L(f)$ *for* $f \in F_L$*,* $\beta(f) = [\vec{\mu}]^{(W(f))}$*for* $f \in F_N$ *and* $type(\vec{\mu}) = \alpha(p)$*,* $\beta((t, in)) = \beta((t, p))$ *for* $t \in {}^\bullet t$ *and* $\beta((out, t)) = \beta((p, t))$ *for* $t \in t^\bullet$*.* ◁

Generalized soundness of a WF-Net ensures that any number of tokens in the initial place are "transferred" to the final place. As shown in Sect. 5.1, the EC-closure of a sound WF-net is identifier sound and live. A similar approach is taken to show that the refinement is weakly bisimilar to the original net. Analogously to [15], the bisimulation relation is the identity relation, except for place p. The relation maps all possible token configurations of place p to any reachable marking in the WF-net, given p's token configuration.

Lemma 11. *Let* $L = (P_L, T_L, F_L, \alpha_L, \beta_L)$ *be a t-PNID with initial marking* M_0*, let* $p \in P_L$ *be a place s.t.* $M_0(p) = \emptyset$*, and let* $N = (P_N, T_N, F_N, W_N, in_N, out_N)$ *be a WF-net. If* N *is generalized sound, then* $\Gamma_L \approx^r \hat{\mathbb{H}}_{T_N}(\Gamma_{L \oplus_p N})$*.* ◁

Proof. For simplicity, we start by defining a type extension of N as a t-PNID $N' = (P_N, T_N, F_N, \alpha, \beta)$, where $type(\vec{v}) = \vec{\lambda}$, $\alpha(p) = \vec{\lambda}$ for all places $p \in P_N$, and $\beta(f) = \vec{v}^{W(f)}$ for all $f \in F_N$, and $\beta((t_e, in)) = \beta((out, t_c)) = [\vec{v}]$.

To prove bisimilarity, we define $R = \{(M, M' + m) \mid M \in \mathcal{R}(L, M_0), M' \in \mathcal{A}, m \in \mathcal{B}\}$ where $\mathcal{A} = \{M' \mid M', M \in \mathcal{R}(L, M_0), M'(p) = \emptyset, \forall q \in P_L \setminus \{p\} : M'(q) = M(q)\}$ and $\mathcal{B} = \{m \mid m \in \mathcal{R}(N', [in]), [in] = M(p), \forall M \in \mathcal{R}(L, M_0)\}$. ($\Rightarrow$) Let $(M, M' + m) \in R$ and $M[t, \psi\rangle \bar{M}$. We need to show that exists $\bar{M}'$ and $\bar{m}$ such that $(M' + m) \xrightarrow{(t,\psi)} (\bar{M}' + \bar{m})$ and $(\bar{M}, \bar{M}' + \bar{m}) \in R$. If $t \notin {}^\bullet p$ (or $p \notin {}^\bullet t$), then $M'(q) = M(q)$ for all $q \in P_L \setminus p$. Thus t is also enabled in $M'(q)$ and $M(q) \geq \beta((q, t))$. Then by the firing rule there exists $\bar{M}'$ such that $(M' + m)[t, \psi\rangle(\bar{M}' + m)$ and $\bar{M}(q) = \bar{M}'(q)$ for all $q \in P_L$. Thus, $(M', \bar{M}' + m) \in R$. If $t \in {}^\bullet p$, then, by construction, $M'(q) = M(q)$ for all $q \in P_L \setminus p$. Thus, using the same reasoning as above, $(M' + m)[t, \psi\rangle(\bar{M}' + \bar{m})$ for all $q \in P_L \setminus p$ and $\bar{m}(in) = M'(p)$. Thus, $(M', \bar{M}' + \bar{m}) \in R$.

Now, assume that $p \in {}^\bullet t$ and $\rho_\psi(\beta((p, t))) = \mathtt{id}$. Given that N is generalized sound and by applying Lemma 3, there exists a firing sequence η for N' that carries identifier $\mathtt{id}$ to *out*. This means that, by construction, $M'(q) = M(q)$, for all $q \in P_L$, and $m(out)(\mathtt{id}) = M(p)(\mathtt{id})$. Hence, t is enabled in $(M' + m)$ under binding ψ' that differs from ψ everywhere but on place *out*. By the firing rule, there exists $(\bar{M}' + \bar{m})$ s.t. $(M' + m)[t, \psi'\rangle(\bar{M}' + \bar{m})$ and $(M, (\bar{M}' + \bar{m})) \in R$. ($\Leftarrow$) By analogy with the previous argument.

As a consequence of the bisimulation relation, the refinement is identifier sound and live if the original net is identifier sound.

Theorem 4. *Let (L, M) be a marked t-PNID and N be a generalized sound WF net. Then (L, M) is identifier sound and live iff $(L \oplus N, M)$ is identifier sound and live.* ◁

The refinement rule allows to combine the approaches discussed in this section. For example, a designer can first design a net using the construction rules of Sect. 5.2, and then design generalized WF-nets for specific places. In this way, the construction rules and refinement rules ensure that the designer can model systems where data and processes are in resonance.

6 Related Work

This work belongs to the line of research that aims at augmenting pure control-flow description of processes with data, and study formal properties of the resulting, integrated models. When doing so, it becomes natural to move from case-centric process models whose analysis focuses on the evolution of a single instance in isolation, to so-called *object-centric process models* where multiple related instances of the same or different processes co-evolve. This is relevant for process modeling, analysis, and mining [2].

Different approaches to capture the control-flow backbone of object-oriented processes have been studied in literature, including declarative [4] and database-centric models [22]. In this work, we follow the Petri net tradition, which comes with three different strategies to tackle object-centric processes.

A first strategy is to represent objects implicitly. The most prominent example in this vein is constituted by proclets [7]. Here, each object type comes with a Petri net specifying its life cycle. Special ports, annotated with multiplicity constraints, are used to express generation and synchronization points in the process. Correctness analysis of proclets is an open research topic.

A second strategy is to represent objects explicitly. Models adopting this strategy are typically extensions of ν-PNs [28], building on their ability to generate (fresh) object identifiers and express guarded transitions relating multiple objects at once. The ISML approach [25] equips Petri nets with identifiers (PNIDs) [16] with the ability of manipulating populations of objects defining the extensional level of an ORM data model. For such models, correctness properties are assessed by imposing that the overall set of object identifiers is finite, and fixed a-priori. Catalog-nets [10] extend PNIDs with the ability of querying a read-only database containing background information. Decidability and other meta-properties, as well as actual algorithms for verification based on SMT model-checking, are given for safety properties, whereas (data-aware) soundness can only be assessed for state-bounded systems [5,22].

The third, final strategy for modeling object-centric processes with Petri nets is to rely on models that highlight how multiple objects of different types may flow through shared transitions, without considering object identifier values. This approach is followed in [3], where object-centric nets are extracted from event logs, where logged events might come with sets of object identifiers. Soundness for this model is studied in [21].

The approach studied in this paper focuses on the essence of Petri net-based object-centric processes adopting the explicit approach, that is, grounded on PNIDs. We provide, for the first time, a notion of *identifier soundness* that conceptually captures the intended evolution of objects within a net, show that such a property is undecidable to check in general, and provide a pattern-based construction technique that guarantees to produce identifier-sound models.

7 Conclusions

Achieving harmony in models that describe how processes data objects manipulate is challenging. In this paper, we use typed Petri nets with Identifiers (t-PNIDs) to model these complex interactions of multiple objects, referred through their identifiers. We propose identifier soundness as a correctness criterion that conceptually captures the expected evolution of each object. Identifier soundness is, in general, undecidable for t-PNIDs. For two subclasses, we show that identifier soundness is guaranteed and that the overall model remains live.

Many systems allow for a dynamic number of simultaneously active objects. In theory, this number can be infinite, and thus such models become width-unbounded. However, for many systems, there is a natural upper bound, which can be either assumed or guaranteed with different modeling techniques (such as multiplicity upper bounds on objects [22] or resources [23,29]. One can extend t-PNIDs by enriching objects with attributes over different data types, similar

as it is done in data modeling and knowledge graphs. This calls for combining the techniques studied in this paper with data abstraction techniques used to deal with numerical data types, possibly equipped with arithmetics [8,9].

We plan to provide tool support for designers of such systems. Although many correctness criteria are undecidable, this does not mean designers should be left in the dark. Since the ISM-suite [31] already allows to model t-PNIDs, we intend to work on extending it with verification techniques to support the modeler in designing systems where processes and data are in resonance.

References

1. van der Aalst, W.M.P.: Verification of workflow nets. In: Azéma, P., Balbo, G. (eds.) ICATPN 1997. LNCS, vol. 1248, pp. 407–426. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63139-9_48
2. van der Aalst, W.M.P.: Object-centric process mining: dealing with divergence and convergence in event data. In: Ölveczky, P.C., Salaün, G. (eds.) SEFM 2019. LNCS, vol. 11724, pp. 3–25. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30446-1_1
3. van der Aalst, W.M.P., Berti, A.: Discovering object-centric petri nets. Fundam. Informaticae **175**(1–4), 1–40 (2020)
4. Artale, A., Kovtunova, A., Montali, M., van der Aalst, W.M.P.: Modeling and reasoning over declarative data-aware processes with object-centric behavioral constraints. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol. 11675, pp. 139–156. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_11
5. Bagheri Hariri, B., Calvanese, D., Montali, M., Deutsch, A.: State-boundedness in data-aware dynamic systems. In: Proceedings of KR 2014. AAAI Press (2014)
6. Berthelot, G.: Verification de Reseaux de Petri. Ph.D. thesis, Université Pierre et Marie Curie (Paris) (1978)
7. Fahland, Dirk: Describing behavior of processes with many-to-many interactions. In: Donatelli, Susanna, Haar, Stefan (eds.) PETRI NETS 2019. LNCS, vol. 11522, pp. 3–24. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21571-2_1
8. Felli, P., de Leoni, M., Montali, M.: Soundness verification of decision-aware process models with variable-to-variable conditions. In: Proceedings of ACSD 2019, pp. 82–91. IEEE (2019)
9. Felli, P., Montali, M., Winkler, S.: Linear-time verification of data-aware dynamic systems with arithmetic. In: Proceedings of AAAI 2022. AAAI Press (2022)
10. Ghilardi, S., Gianola, A., Montali, M., Rivkin, A.: Petri nets with parameterised data. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 55–74. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_4
11. Glabbeek, R.: The linear time - branching time spectrum II: the semantics of sequential systems with silent moves. In: CONCUR 1993. LNCS, vol. 715, pp. 66–81. Springer (1993). https://doi.org/10.1007/3-540-57208-2_6
12. Hack, M.: The recursive equivalence of the reachability problem and the liveness problem for Petri nets and vector addition systems. In: Proceedings of SWAT 1974. pp. 156–164. IEEE Computer Society (1974)

13. van Hee, K.M., Hidders, J., Houben, G.J., Paredaens, J., Thiran, P.: On the relationship between workflow models and document types. Inf. Syst. **34**(1), 178–208 (2009)
14. van Hee, K.M., Oanea, O., Post, R., Somers, L.J., van der Werf, J.M.E.M.: Yasper: a tool for workflow modeling and analysis. In: Proceedings of ACSD 2006, pp. 279–282. IEEE (2006)
15. van Hee, K., Sidorova, N., Voorhoeve, M.: Soundness and separability of workflow nets in the stepwise refinement approach. In: van der Aalst, W.M.P., Best, E. (eds.) ICATPN 2003. LNCS, vol. 2679, pp. 337–356. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-44919-1_22
16. van Hee, K.M., Sidorova, N., Voorhoeve, M., van der Werf, J.M.E.M.: Generation of database transactions with petri nets. Fundam. Inform. **93**(1–3), 171–184 (2009)
17. van Hee, K.M., Sidorova, N., van der Werf, J.M.: Business process modeling using petri nets. In: Jensen, K., van der Aalst, W.M.P., Balbo, G., Koutny, M., Wolf, K. (eds.) Transactions on Petri Nets and Other Models of Concurrency VII. LNCS, vol. 7480, pp. 116–161. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38143-0_4
18. Karp, R.M., Miller, R.E.: Parallel program schemata. J. Comput. Syst. Sci. **3**(2), 147–195 (1969)
19. Lasota, S.: Decidability border for petri nets with data: WQO dichotomy conjecture. In: Kordon, F., Moldt, D. (eds.) PETRI NETS 2016. LNCS, vol. 9698, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39086-4_3
20. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs - a constructive approach. In: Colom, J.-M., Desel, J. (eds.) PETRI NETS 2013. LNCS, vol. 7927, pp. 311–329. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38697-8_17
21. Lomazova, I.A., Mitsyuk, A.A., Rivkin, A.: Soundness in object-centric workflow Petri nets. CoRR abs/2112.14994 (2021)
22. Montali, M., Calvanese, D.: Soundness of data-aware, case-centric processes. Int. J. Softw. Tools for Technol. Transf. **18**(5), 535–558 (2016). https://doi.org/10.1007/s10009-016-0417-2
23. Montali, M., Rivkin, A.: Model checking Petri nets with names using data-centric dynamic systems. Formal Aspects Comput. **28**(4), 615–641 (2016). https://doi.org/10.1007/s00165-016-0370-6
24. Murata, T.: Petri nets: properties, analysis and applications. Proc. IEEE **77**(4), 541–580 (1989)
25. Polyvyanyy, A., van der Werf, J.M.E.M., Overbeek, S., Brouwers, R.: Information systems modeling: language, verification, and tool support. In: Giorgini, P., Weber, B. (eds.) CAiSE 2019. LNCS, vol. 11483, pp. 194–212. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_13
26. Reisig, W.: Understanding Petri Nets - Modeling Techniques, Analysis Methods. Case Studies. Springer, Berlin Heidelberg (2013). https://doi.org/10.1007/978-3-642-33278-4
27. Rosa-Velardo, F., Alonso, O.M., de Frutos-Escrig, D.: Mobile synchronizing petri nets: a choreographic approach for coordination in Ubiquitous systems. Electron. Notes Theor. Comput. Sci. **150**(1), 103–126 (2006)
28. Rosa-Velardo, F., de Frutos-Escrig, D.: Decidability and complexity of Petri nets with unordered data. Theor. Comput. Sci. **412**(34), 4439–4451 (2011)
29. Sidorova, N., Stahl, C.: Soundness for resource-constrained workflow nets is decidable. IEEE Trans. Syst. Man Cybern. Syst. **43**(3), 724–729 (2013)

30. Weidlich, M., Polyvyanyy, A., Mendling, J., Weske, M.: Causal behavioural profiles - efficient computation, applications, and evaluation. Fundam. Inform. **113**(3–4), 399–435 (2011)
31. van der Werf, J.M.E.M., Polyvyanyy, A.: The information systems modeling suite. In: Janicki, R., Sidorova, N., Chatain, T. (eds.) PETRI NETS 2020. LNCS, vol. 12152, pp. 414–425. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51831-8_22