A "Symbolic" Representation of Object-Nets
(Extended Version)

Michael Köhler-Bußmeier[0000−0002−3074−4145] 1 and
Lorenzo Capra[0000−0002−1029−1169] 2

[1] University of Applied Science Hamburg
Berliner Tor 7, D-20099 Hamburg, Germany
michael.koehler-bussmeier@haw-hamburg.de
[2] Dipartimento di Informatica, Università degli Studi di Milano
Via Celoria 18, Milan, Italy
capra@di.unimi.it

Abstract. In this contribution we extend the concept of a Petri net morphism to Elementary Object Systems (Eos). Eos are a nets-within-nets formalism, i.e. we allow the tokens of a Petri net to be Petri nets again. This nested structure has the consequence that even systems defined by very small Petri nets have a quite huge reachability graph. In this contribution we use automorphism to describe symmetries of the Petri net topology. Since these symmetries carry over to markings as well this leads to a condensed state space, too.

Keywords: Automorphism, canonical representation, nets within nets, nets as tokens, state space reductions, symmetry

1 Exploiting Symmetry and Canonical Representations

In this paper we study Elementary Object Systems (Eos) [10] a Nets-within-Nets formalism as proposed by Valk [18], i.e., we allow the tokens of a Petri net to be Petri nets again. Due to the nesting structure many of the classical decision problems, like reachability and liveness, become undecidable for Eos.

From a complexity perspective we have studied these problems for *safe* Eos [12, 13, 10] where markings are restricted to *sets* (i.e., places are either marked or unmarked). More precisely: All problems that are expressible in LTL or CTL, which includes reachability and liveness, are PSpace-complete. This means that in terms of complexity theory safe Eos are no more complex than safe place transition nets (p/t nets). But, a look at the details shows a difference that is pratically relevant: For safe p/t nets it is known that whenever there are n places, then the number of reachable states is bounded by $O(2^n)$; but, for safe Eos the number of reachable states is in $O(2^{(n^2)})$ – a quite drastic increase. Therefore, our main goal is to derive a *condensed* state space for Eos, were 'condensed' is expressed as a factorisation modulo an equivalence.

In this contribution we extend the concept of a Petri net morphism to Elementary Object Systems (Eos). Eos are a nets-within-nets formalism. Here, we use automorphism to describe symmetries of the Petri net topology. Since these symmetries carry over to markings as well this leads to a condensed state space, too. In our approach these symmetries are introduced very naturally to the representation of the state space using *canonical representations* of markings.

The paper has the following structure. Section 2 introduces base nets-within-nets (EOS). In Section 3 we define a symbolic representation of the EOS structure. The work closes with a conclusion and outlook.

2 Nets-within-Nets, EOS

Object nets [9, 11, 10], follow the *nets-within-nets* paradigm as proposed by Valk [18]. Other approaches adapting the nets-within-nets approach are nested nets [14], mobile predicate/transition nets [19], PN^2 [8], hypernets [1], and adaptive workflow nets [15]. There are relationships to Rewritable Petri nets [2] and Reconfigurable Petri Nets [17]. Object Nets can be seen as the Petri net perspective on contextual change, in contrast to the Ambient Calculus [6] or the π-calculus [16], which form the process algebra perspective.

2.1 Petri Nets

The definition of Petri nets relies on the notion of multisets. A multiset $\mathbf{m}$ on the set D is a mapping $\mathbf{m} : D \to \mathbb{N}$. Multisets are generalisations of sets in the sense that every subset of D corresponds to a multiset $\mathbf{m}$ with $\mathbf{m}(d) \leq 1$ for all $d \in D$. The empty multiset $\mathbf{0}$ is defined as $\mathbf{0}(d) = 0$ for all $d \in D$. Multiset addition for $\mathbf{m}_1, \mathbf{m}_2 : D \to \mathbb{N}$ is defined component-wise: $(\mathbf{m}_1 + \mathbf{m}_2)(d) := \mathbf{m}_1(d) + \mathbf{m}_2(d)$. Multiset-difference $\mathbf{m}_1 - \mathbf{m}_2$ is defined by $(\mathbf{m}_1 - \mathbf{m}_2)(d) := \max(\mathbf{m}_1(d) - \mathbf{m}_2(d), 0)$. We use common notations for the cardinality of a multiset $|\mathbf{m}| := \sum_{d \in D} \mathbf{m}(d)$ and multiset ordering $\mathbf{m}_1 \leq \mathbf{m}_2$, where the partial order $\leq$ is defined by $\mathbf{m}_1 \leq \mathbf{m}_2 \iff \forall d \in D : \mathbf{m}_1(d) \leq \mathbf{m}_2(d)$.

A multiset $\mathbf{m}$ is finite if $|\mathbf{m}| < \infty$. The set of all finite multisets over the set D is denoted $MS(D)$. The set $MS(D)$ naturally forms a monoid with multiset addition $+$ and the empty multiset $\mathbf{0}$. Multisets can be identified with the commutative monoid structure $(MS(D), +, 0)$. Multisets are the free commutative monoid over D since every multiset has the unique representation in the form $\mathbf{m} = \sum_{d \in D} \mathbf{m}(d) \cdot d$, where $\mathbf{m}(d)$ denotes the multiplicity of d. Multisets can also be represented as a formal sum in the form $\mathbf{m} = \sum_{i=1}^{n} x_i$, where $x_i \in D$.

Any mapping $f : D \to D'$ is extended to a multiset homomorphism $f^\sharp : MS(D) \to MS(D')$: $f^\sharp \left(\sum_{i=1}^{n} x_i\right) = \sum_{i=1}^{n} f(x_i)$. This includes the special case $f^\sharp(\mathbf{0}) = \mathbf{0}$. We simply write f to denote the mapping $f^\sharp$. The notation is in accordance with the set-theoretic notation $f(A) = \{f(a) \mid a \in A\}$.

Definition 1. *A* p/t net *N is a tuple $N = (P, T, \mathbf{pre}, \mathbf{post})$, such that P is a set of places, T is a set of transitions, with $P \cap T = \emptyset$, and $\mathbf{pre}, \mathbf{post} : T \to MS(P)$ are the pre- and post-condition functions. A marking of N is a multiset of places: $\mathbf{m} \in MS(P)$. A p/t net with initial marking $\mathbf{m}_0$ is denoted $N = (P, T, \mathbf{pre}, \mathbf{post}, \mathbf{m}_0)$.*

We use the usual notation for nets such as ${}^\bullet x$ for the set of predecessors and $x^\bullet$ for the set of successors for a node $x \in (P \cup T)$. For $t \in T$ we have ${}^\bullet t = \{p \in P \mid \mathbf{pre}(t)(p) > 0\}$ and $t^\bullet = \{p \in P \mid \mathbf{post}(t)(p) > 0\}$. For $p \in P$ we have ${}^\bullet p = \{t \in T \mid \mathbf{post}(t)(p) > 0\}$ and $p^\bullet = \{t \in T \mid \mathbf{pre}(t)(p) > 0\}$.

A transition $t \in T$ of a p/t net N is enabled in marking $\mathbf{m}$ iff $\forall p \in P : \mathbf{m}(p) \geq \mathbf{pre}(t)(p)$ holds. The successor marking when firing t is $\mathbf{m}'(p) = \mathbf{m}(p) - \mathbf{pre}(t)(p) + \mathbf{post}(t)(p)$ for all $p \in P$. Using multiset notation enabling is expressed by $\mathbf{m} \geq \mathbf{pre}(t)$ and the successor marking is $\mathbf{m}' = \mathbf{m} - \mathbf{pre}(t) + \mathbf{post}(t)$. We denote the enabling of t in marking $\mathbf{m}$ by $\mathbf{m} \xrightarrow[N]{t}$. Firing of an enabled t is denoted by $\mathbf{m} \xrightarrow[N]{t} \mathbf{m}'$. The net N is omitted if it is clear from the context.

Firing is extended to sequences $w \in T^*$ in the obvious way: (i) $\mathbf{m} \xrightarrow{\epsilon} \mathbf{m}$; (ii) If $\mathbf{m} \xrightarrow{w} \mathbf{m}'$ and $\mathbf{m}' \xrightarrow{t} \mathbf{m}''$ hold, then we have $\mathbf{m} \xrightarrow{wt} \mathbf{m}''$. We write $\mathbf{m} \xrightarrow{*} \mathbf{m}'$ whenever there is some $w \in T^*$ such that $\mathbf{m} \xrightarrow{w} \mathbf{m}'$ holds. The set of reachable markings is $RS(\mathbf{m}_0) := \{\mathbf{m} \mid \exists w \in T^* : \mathbf{m}_0 \xrightarrow{w} \mathbf{m}\}$.

2.2 Elementary Object Systems

In the following we consider *Elementary Object System* (EOS) [10], which have a two-levelled structure. An elementary object system (EOS) is composed of a system net, which is a p/t net $\widehat{N} = (\widehat{P}, \widehat{T}, \mathbf{pre}, \mathbf{post})$ and a set of object nets $\mathcal{N} = \{N_1, \ldots, N_n\}$, which are p/t nets given as $N = (P_N, T_N, \mathbf{pre}_N, \mathbf{post}_N)$, where $N \in \mathcal{N}$. In extension we assume that all sets of nodes (places and transitions) are pairwise disjoint. Moreover we assume $\widehat{N} \notin \mathcal{N}$ and the existence of the object net $\bullet \in \mathcal{N}$, which has no places and no transitions and is used to model anonymous, so called black tokens.

Typing The system net places are typed by the mapping $d : \widehat{P} \to \mathcal{N}$ with the meaning, that a place $\widehat{p} \in \widehat{P}$ of the system net with $d(\widehat{p}) = N$ may contain only net-tokens of the object net type N.[3] No place of the system net is mapped to the system net itself since $\widehat{N} \notin \mathcal{N}$.

A typing is called *conservative* iff for each place $\widehat{p}$ in the preset of a system net transition $\widehat{t}$ such that $d(\widehat{p}) \neq \bullet$ there is place in the postset being of the same type: $(d({}^\bullet\widehat{t}) \cup \{\bullet\}) \subseteq (d(\widehat{t}^\bullet) \cup \{\bullet\})$. An EOS is *conservative* iff its typing d is.

An EOS is *p/t-like* iff it has only places for black tokens: $d(\widehat{P}) = \{\bullet\}$.

[3] In some sense, net-tokens are object nets with their own marking. However, net-tokens should not be considered as *instances* of an object net (as in object-oriented programming), since net-tokens do not have an identity. This is reflected by the fact that the firing rule joins and distributes the net-tokens' markings.

Instead, all net-tokes of an object net act as a *collective* entity, like a group. This group can be considered as an object with identy – an object with its state distributed over the net-tokens. For in in-depth discussion of this semantics cf. [18].

Nested Markings Since the tokens of an EOS are instances of object nets, a *marking* of an EOS is a *nested* multiset. A marking of a EOS OS is denoted $\mu = \sum_{k=1}^{|\mu|}(\widehat{p}_k, M_k)$, where $\widehat{p}_k$ is a place of the system net and M_k is the marking of a net-token of type $d(\widehat{p}_k)$. To emphasize nesting, the marks are also denoted as $\mu = \sum_{k=1}^{|\mu|} \widehat{p}_k[M_k]$. Markings of the form $\widehat{p}[\mathbf{0}]$ with $d(\widehat{p}) = \bullet$ are abbreviated as $\widehat{p}[]$.

The set of all markings which are syntactically consistent with the typing d is denoted $\mathcal{M}$, where $d^{-1}(N) \subseteq \widehat{P}$ is the set of system net places of the type N:

$$\mathcal{M} := MS\left(\bigcup\nolimits_{N\in\mathcal{N}} \left(d^{-1}(N) \times MS(P_N)\right)\right) \tag{1}$$

We define the partial order $\sqsubseteq$ on nested multisets by setting $\mu_1 \sqsubseteq \mu_2$ iff $\exists \mu : \mu_2 = \mu_1 + \mu$. A more liberal variant is the order $\preceq$ defined by:

$$\begin{aligned} \alpha \preceq \beta \quad \iff \quad & \alpha = \textstyle\sum_{i=1}^{m} \widehat{a}_i[A_i] \wedge \beta = \sum_{j=1}^{n} \widehat{b}_j[B_j] \wedge \\ & \exists \text{ injection } f : \{1, ..., m\} \to \{1, ..., n\} : \\ & \forall 1 \leq i \leq m : \widehat{a}_i = \widehat{b}_{f(i)} \wedge A_i \leq B_{f(i)} \end{aligned} \tag{2}$$

For $\alpha \preceq \beta$ the injection f generates a sub-marking of β which is denoted $f(\alpha) = \sum_{i=1}^{m} \widehat{a}_{f(i)}[A_{f(i)}]$. Note that $\alpha \sqsubseteq \beta$ is a special case of $\alpha \preceq \beta$, where $A_i \leq B_{f(i)}$ is restricted to $A_i = B_{f(i)}$.

Events Analogously to markings, which are nested multisets μ, the events of an EOS are also nested. An EOS allows three different kinds of events – as illustrated by the EOS in Fig. 1.

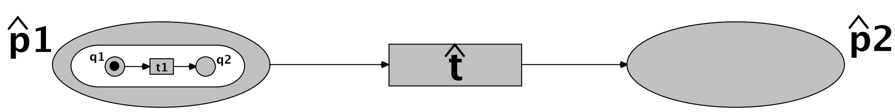

Fig. 1. An Elementary Object Net System (EOS)

1. System-autonomous: The system net transition $\widehat{t}$ fires autonomously which moves the net-token from $\widehat{p}_1$ to $\widehat{p}_2$ without changing its marking.
2. Object-autonomous: The object net fires transition t_1, which "moves" the black token from q_1 to q_2. The object net itself remains at its location $\widehat{p}_1$.
3. Synchronisation: The system net transition $\widehat{t}$ fires synchronously with t_1 in the object net. Whenever synchronisation is demanded, autonomous actions are forbidden.

The set of events is denoted Θ. Events are formalised as a pair $\widehat{\tau}[\vartheta]$, where $\widehat{\tau}$ is either the transition that fires in the system net or a special "idle" transition

$id_{\widehat{p}}$ (cf. below); and ϑ is a function such that $\vartheta(N)$ is the multiset of transitions, which have to fire synchronously with $\widehat{\tau}$, (i.e. for each object net $N \in \mathcal{N}$ we have $\vartheta(N) \in MS(T_N)$).[4]

In general $\widehat{\tau}[\vartheta]$ describes a synchronisation, but autonomous events are special subcases: Obviously, a system-autonomous event is the special case, where $\vartheta = \mathbf{0}$ with $\mathbf{0}(N) = \mathbf{0}$ for all object nets N. To describe an object-autonomous event we assume the set of *idle transitions* $\{id_{\widehat{p}} \mid \widehat{p} \in \widehat{P}\}$, where $id_{\widehat{p}}$ formalises object-autonomous firing on the place $\widehat{p}$:

1. Each idle transition $id_{\widehat{p}}$ has $\widehat{p}$ as a side condition: $\mathbf{pre}(id_{\widehat{p}}) = \mathbf{post}(id_{\widehat{p}}) := \widehat{p}$.
2. Each idle transition $id_{\widehat{p}}$ synchronises only with transitions from $N = d(\widehat{p})$:

$$\forall \widehat{\tau}[\vartheta] \in \Theta : \ \widehat{\tau} = id_{\widehat{p}} \quad \Longrightarrow \quad \forall N \in \mathcal{N} : (\vartheta(N) \neq \mathbf{0} \iff N = d(\widehat{p}))$$

Definition 2 (Elementary Object System, EOS). *An elementary object system (*Eos*) is a tuple $OS = (\widehat{N}, \mathcal{N}, d, \Theta)$, where:*

1. *$\widehat{N}$ is a p/t net, called the* system net*.*
2. *$\mathcal{N}$ is a finite set of disjoint p/t nets, called* object nets*.*
3. *$d : \widehat{P} \to \mathcal{N}$ is the* typing *of the system net places.*
4. *Θ is the set of* events*.*

An Eos *OS with initial marking μ_0 is a marked* Eos*. We use the term* Eos *both for marked and unmarked systems.*

Example 1. Figure 2 shows an Eos with the system net $\widehat{N}$ and the object nets $\mathcal{N} = \{N_1, N_2\}$. The system has four net-tokens: two on place $\widehat{p}_1$ and one on $\widehat{p}_2$ and $\widehat{p}_3$ each. (Please ignore the net-tokens above the transition and on the places $\widehat{p}_4$, $\widehat{p}_5$ and $\widehat{p}_6$ on the right; they are use below to illustrate the firing rule.) The net-tokens on $\widehat{p}_1$ and $\widehat{p}_2$ share the same net structure, but have independent markings.
The system net is $\widehat{N} = (\widehat{P}, \widehat{T}, \mathbf{pre}, \mathbf{post})$, where $\widehat{P} = \{\widehat{p}_1, \ldots, \widehat{p}_6\}$ and $\widehat{T} = \{\widehat{t}\}$.
One object net is $N_1 = (P_1, T_1, \mathbf{pre}_1, \mathbf{post}_1)$ with $P_1 = \{a_1, b_1\}$ and $T_1 = \{t_1\}$.
Another object net is $N_2 = (P_2, T_2, \mathbf{pre}_2, \mathbf{post}_2)$ with $P_2 = \{a_2, b_2, c_2\}$ and $T_2 = \{t_2\}$.
The typing is $d(\widehat{p}_1) = d(\widehat{p}_2) = d(\widehat{p}_4) = N_1$ and $d(\widehat{p}_3) = d(\widehat{p}_5) = d(\widehat{p}_6) = N_2$.
We have only one event: $\Theta = \{\widehat{t}[N_1 \mapsto t_1, N_2 \mapsto t_2]\}$.
The initial marking has two net-tokens on $\widehat{p}_1$, one on $\widehat{p}_2$, and one on $\widehat{p}_3$:

$$\mu = \widehat{p}_1[a_1 + b_1] + \widehat{p}_1[\mathbf{0}] + \widehat{p}_2[a_1] + \widehat{p}_3[a_2 + b_2]$$

Note that for Figure 2 the structure is the same for the three net-tokens on $\widehat{p}_1$ and $\widehat{p}_2$ but the net-token markings are different.

[4] In the graphical representation the events are generated by transition inscriptions. For each object net $N \in \mathcal{N}$ a system net transition $\widehat{t}$ is labelled with a multiset of channels $\widehat{l}(\widehat{t})(N) = ch_1 + \cdots + ch_n$, depicted as $\langle N{:}ch_1, N{:}ch_2, \ldots \rangle$. Similarily, an object net transition t may be labelled with a channel $l_N(t) = ch$ – depicted as $\langle {:}ch \rangle$ whenever there is such a label. We obtain an event $\widehat{t}[\vartheta]$ by setting $\vartheta(N) := t_1 + \cdots + t_n$ to be any transition multiset such that the labels match: $l_N(t_1) + \cdots + l_N(t_n) = \widehat{l}(\widehat{t})(N)$.

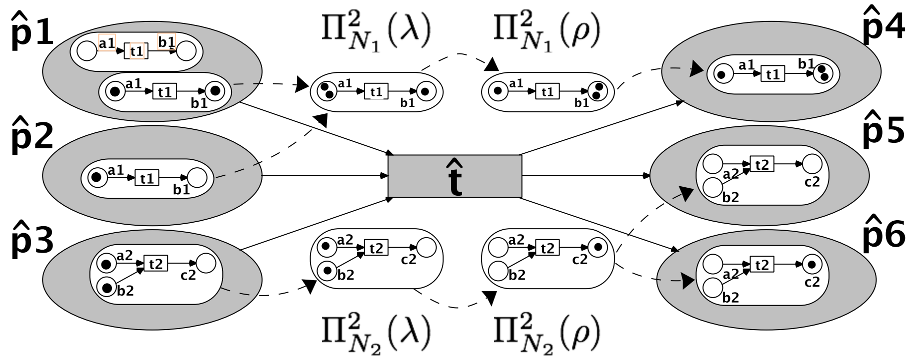

Fig. 2. An Example EOS illustrating the firing of $\widehat{t}[N_1 \mapsto t_1, N_2 \mapsto t_2]$

Firing Rule The projection Π^1 on the first component abstracts from the sub-structure of all net-tokens for a marking of an EOS:

$$\Pi^1 \left(\sum\nolimits_{k=1}^{n} \widehat{p}_k[M_k]\right) := \sum\nolimits_{k=1}^{n} \widehat{p}_k \tag{3}$$

The projection Π_N^2 on the second component is the sum of all net-token markings M_k of the type $N \in \mathcal{N}$, ignoring their local distribution within the system net:

$$\Pi_N^2 \left(\sum\nolimits_{k=1}^{n} \widehat{p}_k[M_k]\right) := \sum\nolimits_{k=1}^{n} \mathbf{1}_N(\widehat{p}_k) \cdot M_k \tag{4}$$

where the indicator function $\mathbf{1}_N : \widehat{P} \to \{0,1\}$ is $\mathbf{1}_N(\widehat{p}) = 1$ iff $d(\widehat{p}) = N$. Note that $\Pi_N^2(\mu)$ results in a marking of the object net N.

A system event $\widehat{\tau}[\vartheta]$ removes net-tokens together with their individual internal markings. Firing the event replaces a nested multiset $\lambda \in \mathcal{M}$ that is part of the current marking μ, i.e. $\lambda \sqsubseteq \mu$, by the nested multiset ρ. Therefore the successor marking is $\mu' := (\mu - \lambda) + \rho$. The enabling condition is expressed by the *enabling predicate* ϕ_{OS} (or just ϕ whenever OS is clear from the context):

$$\begin{aligned}\phi(\widehat{\tau}[\vartheta], \lambda, \rho) \iff\ & \Pi^1(\lambda) = \mathbf{pre}(\widehat{\tau}) \wedge \Pi^1(\rho) = \mathbf{post}(\widehat{\tau}) \wedge \\ & \forall N \in \mathcal{N} : \Pi_N^2(\lambda) \geq \mathbf{pre}_N(\vartheta(N)) \wedge \\ & \forall N \in \mathcal{N} : \Pi_N^2(\rho) = \Pi_N^2(\lambda) - \mathbf{pre}_N(\vartheta(N)) + \mathbf{post}_N(\vartheta(N))\end{aligned} \tag{5}$$

With $\widehat{M} = \Pi^1(\lambda)$ and $\widehat{M'} = \Pi^1(\rho)$ as well as $M_N = \Pi_N^2(\lambda)$ and $M'_N = \Pi_N^2(\rho)$ for all $N \in \mathcal{N}$ the predicate ϕ has the following meaning:

1. The first conjunct expresses that the system net multiset $\widehat{M}$ corresponds to the pre-condition of the system net transition $\widehat{\tau}$, i.e. $\widehat{M} = \mathbf{pre}(\widehat{\tau})$.
2. In turn, a multiset $\widehat{M'}$ is produced, that corresponds to the post-set of $\widehat{\tau}$.
3. A multi-set $\vartheta(N)$ of object net transitions is enabled if the sum M_N of the net-token markings (of type N) enable it, i.e. $M_N \geq \mathbf{pre}_N(\vartheta(N))$.

4. The firing of $\widehat{\tau}[\vartheta]$ must also obey the *object marking distribution condition*: $M'_N = M_N - \mathbf{pre}_N(\vartheta(N)) + \mathbf{post}_N(\vartheta(N))$, where $\mathbf{post}_N(\vartheta(N)) - \mathbf{pre}_N(\vartheta(N))$ is the effect of the object net's transitions on the net-tokens.

Note that conditions 1. and 2. assure that only net-tokens relevant for the firing are included in λ and ρ. Conditions 3. and 4. allow for additional tokens in the net-tokens.

For system-autonomous events $\widehat{t}[\mathbf{0}]$ the enabling predicate ϕ can be simplified further. We have $\mathbf{pre}_N(\mathbf{0}(N)) = \mathbf{post}_N(\mathbf{0}(N)) = \mathbf{0}$. This ensures $\Pi^2_N(\lambda) = \Pi^2_N(\rho)$, i.e. the sum of markings in the copies of a net-token is preserved w.r.t. each type N. This condition ensures the existence of linear invariance properties

Analogously, for an object-autonomous event we have an idle-transition $\widehat{\tau} = id_{\widehat{p}}$ for the system net and the first and the second conjunct is: $\Pi^1(\lambda) = \mathbf{pre}(id_{\widehat{p}}) = \widehat{p} = \mathbf{post}(id_{\widehat{p}}) = \Pi^1(\rho)$. So, there is an addend $\lambda = \widehat{p}[M]$ in μ with $d(\widehat{p}) = N$ and M enables $\vartheta(N)$.

Definition 3 (Firing Rule). *Let OS be an* EOS *and $\mu, \mu' \in \mathcal{M}$ markings. The event $\widehat{\tau}[\vartheta]$ is enabled in μ for the mode $(\lambda, \rho) \in \mathcal{M}^2$ iff $\lambda \sqsubseteq \mu \wedge \phi(\widehat{\tau}[\vartheta], \lambda, \rho)$ holds.*

An event $\widehat{\tau}[\vartheta]$ that is enabled in μ for the mode (λ, ρ) can fire: $\mu \xrightarrow[OS]{\widehat{\tau}[\vartheta](\lambda,\rho)} \mu'$. The resulting successor marking is defined as $\mu' = \mu - \lambda + \rho$.

We write $\mu \xrightarrow[OS]{\widehat{\tau}[\vartheta]} \mu'$ whenever $\mu \xrightarrow[OS]{\widehat{\tau}[\vartheta](\lambda,\rho)} \mu'$ for some mode (λ, ρ).

Note that the firing rule makes no a-priori assumptions about how to distribute the object net markings onto the generated net-tokens. Therefore we need the mode (λ, ρ) to formulate the firing of $\widehat{\tau}[\vartheta]$ in a functional way.

Example 2. Consider the EOS of Figure 2 again. The current marking μ of the EOS enables $\widehat{t}[N_1 \mapsto t_1, N_2 \mapsto t_2]$ in the mode (λ, ρ), where

$$\begin{aligned}
\mu &= \widehat{p}_1[\mathbf{0}] + \widehat{p}_1[a_1 + b_1] + \widehat{p}_2[a_1] + \widehat{p}_3[a_2 + b_2] = \widehat{p}_1[\mathbf{0}] + \lambda \\
\lambda &= \widehat{p}_1[a_1 + b_1] + \widehat{p}_2[a_1] + \widehat{p}_3[a_2 + b_2] \\
\rho &= \widehat{p}_4[a_1 + b_1 + b_1] + \widehat{p}_5[\mathbf{0}] + \widehat{p}_6[c_2]
\end{aligned}$$

The net-token markings are added by the projections Π^2_N resulting in the markings $\Pi^2_N(\lambda)$. The sub-synchronisation generates $\Pi^2_N(\rho)$. (The results are shown above and below the transition $\widehat{t}$.) After the synchronisation we obtain the successor marking μ' with net-tokens on $\widehat{p}_4$, $\widehat{p}_5$, and $\widehat{p}_6$ as shown in Figure 2:

$$\begin{aligned}
\mu' &= (\mu - \lambda) + \rho = \widehat{p}_1[\mathbf{0}] + \rho \\
&= \widehat{p}_1[\mathbf{0}] + \widehat{p}_4[a_1 + b_1 + b_1] + \widehat{p}_5[\mathbf{0}] + \widehat{p}_6[c_2]
\end{aligned}$$

Note, that we have only presented one mode (λ, ρ) and that other modes are possible, too.

3 Eos-Automorphism and Canonical Representation

The pseudo-symbolic representation of EOS that we will introduce relies on established concepts like graph (auto)morphism and canonical representative. In what follows, we implicitly employ multiset homomorphism.

Definition 4. *Given* p/t nets N *and* N'*, a morphism between* N *and* N' *is a pair* $\varphi = (\varphi_t, \varphi_p)$ *of bijective maps* $\varphi_t : T_N \to T_{N'}, \varphi_p : P_N \to P_{N'}$ *such that:*

$$\forall t \in T_N : \ \mathbf{pre}_{N'}(\varphi_t(t)) = \varphi_p(\mathbf{pre}_N(t)) \wedge \mathbf{post}_{N'}(\varphi_t(t)) = \varphi_p(\mathbf{post}_N(t))$$

We use the notation $\varphi : N \to N'$*, and* $N \cong N'$ *means there exists* $\varphi : N \to N'$*. A morphism between marked* p/t nets $(N, \mathbf{m})$ *and* $(N', \mathbf{m}')$ *is a* p/t *morphism* $\varphi : N \to N'$ *such that* $\varphi_p(\mathbf{m}) = \mathbf{m}'$*.*
(We extend the notation introduced above to marked nets' morphism.)

A morphism $\varphi : N \to N$ *is referred to as an automorphism:* φ_p, φ_t *are permutations. The markings* $\mathbf{m}_1$ *and* $\mathbf{m}_2$ *of* N *are said equivalent if and only if there exists a morphism* $\varphi : (N, \mathbf{m}_1) \to (N, \mathbf{m}_2)$*. We denote this by* $\mathbf{m}_1 \cong \mathbf{m}_2$*. Furthermore,* $\cong$ *establishes an equivalence relation on the set of markings of* N*.*

A *p/t* morphism maintains the firing rule. Let $\varphi : N \to N'$.

$$\mathbf{m}_1 \xrightarrow[N]{t} \mathbf{m}_2 \Longrightarrow \varphi_p(\mathbf{m}_1) \xrightarrow[N']{\varphi_t(t)} \varphi_p(\mathbf{m}_2) \tag{6}$$

The equivalence relation $\cong$ on the markings of a *p/t net* N is thus a congruence when considering transition firing.

3.1 Eos-Automorphism

We provide a natural extension of automorphisms to EOS, which considers that EOS components are p/t nets.

Definition 5. *Let* $OS = (\widehat{N}, \mathcal{N}, d, \Theta)$ *be an* EOS *(Def. 2).*
An EOS-automorphism φ_{OS} *is a collection of* p/t *automorphisms*

$$\varphi_{OS} = \left(\varphi_{\widehat{N}} : \widehat{N} \to \widehat{N},\ (\varphi_N : N \to N)_{N \in \mathcal{N}}\right)$$

that preserve d *and* Θ*:*

1. $\forall \widehat{p} \in P_{\widehat{N}} : \ d(\varphi_{\widehat{N}}(\widehat{p})) = d(\widehat{p})$
2. $\forall \widehat{\tau}[\vartheta] \in \Theta : \ \exists \vartheta' : \varphi_{\widehat{N}}(\widehat{\tau})[\vartheta'] \in \Theta \ \wedge \forall N \in \mathcal{N} : \vartheta'(N) = \varphi_N(\vartheta(N))$

Observe that the second condition in Definition 5 implicitly establishes a permutation within the events in Θ.

We naturally extend the concept of EOS-automorphism to *marked* EOS.

Definition 6. *An automorphism between* (OS, μ) *and* (OS, μ') *is an* Eos-*automorphism* φ_{OS} *such that:*
If $\mu = \sum_{k=1}^{|\mu|} \widehat{p}_k[M_k]$ *then* $\mu' = \varphi_{OS}(\mu) := \sum_{k=1}^{|\mu|} \varphi_{\widehat{N}}(\widehat{p}_k)\big[\varphi_{d(\widehat{p}_k)}(M_k)\big]$.

The markings μ_1 *and* μ_2 *of* OS *are said to be* equivalent *if and only if there exists* $\varphi_{OS} : (OS, \mu_1) \to (OS, \mu_2)$. *We write (abusing notation)* $\mu_1 \cong \mu_2$.

Again, $\cong$ *defines an equivalence relation on the set of markings of* OS.

The analogy of p/t automorphism and Eos-automorphism also holds for the Eos firing rule (cf. Def. 3). Observe that an automorphism φ_{OS} maintains the firing predicate (5).

Lemma 1. *The firing predicate* ϕ *is invariant w.r.t. an automorphism* φ_{OS}*:*

$$\phi(\widehat{\tau}[\vartheta], \lambda, \rho) \implies \phi(\varphi_{OS}(\widehat{\tau}[\vartheta]), \varphi_{OS}(\lambda), \varphi_{OS}(\rho)) \tag{7}$$

Here, $\varphi_{OS}(.)$ *is the congruent homomorphic application of* φ_{OS} *components to* Eos *events and markings.*

Proof. We check the properties of ϕ as defined by (5). By assumption we have $\Pi^1(\lambda) = \mathbf{pre}(\widehat{\tau})$. This implies $\Pi^1(\varphi_{OS}(\lambda)) \cong \Pi^1(\lambda) = \mathbf{pre}(\widehat{\tau}) \cong \mathbf{pre}(\varphi_{OS}(\widehat{\tau}))$. Analogously, we obtain $\Pi^1(\varphi_{OS}(\rho)) = \mathbf{post}(\varphi_{OS}(\widehat{\tau}))$. For the object-nets we have for the preset:

$$\Pi^2_N(\varphi_{OS}(\lambda)) \cong \Pi^2_N(\lambda) \geq \mathbf{pre}_N(\vartheta(N)) \cong \mathbf{pre}_N(\varphi_{OS}(\vartheta(N)))$$

and for the generated net-tokens in the postset we have:

$$\begin{aligned}
\Pi^2_N(\varphi_{OS}(\rho)) &\cong \Pi^2_N(\rho) \\
&= \Pi^2_N(\lambda) \quad - \mathbf{pre}_N(\vartheta(N)) \quad + \mathbf{post}_N(\vartheta(N)) \\
&\cong \Pi^2_N(\varphi_{OS}(\lambda)) - \mathbf{pre}_N(\varphi_{OS}(\vartheta(N))) + \mathbf{post}_N(\varphi_{OS}(\vartheta(N))) \qquad \square
\end{aligned}$$

From this Lemma we immediately obtain the invariance of the firing rule.

Proposition 1. *The firing rule is invariant w.r.t. an automorphism* φ_{OS}*:*

$$\mu \xrightarrow[OS]{\widehat{\tau}[\vartheta](\lambda,\rho)} \mu' \implies \varphi_{OS}(\mu) \xrightarrow[OS]{\varphi_{OS}(\widehat{\tau}[\vartheta])(\varphi_{OS}(\lambda),\varphi_{OS}(\rho))} \varphi_{OS}(\mu') \tag{8}$$

Proof. We have to check the firing rule of Def. 3. The event $\varphi_{OS}(\widehat{\tau}[\vartheta])$ is enabled in the mode $(\varphi_{OS}(\lambda), \varphi_{OS}(\rho))$ since $\varphi_{OS}(\lambda) \cong \lambda \sqsubseteq \mu \cong \varphi_{OS}(\mu))$ and $\phi(\varphi_{OS}(\widehat{\tau}[\vartheta]), \varphi_{OS}(\lambda), \varphi_{OS}(\rho))$ holds, too. The successor marking is $\varphi_{OS}(\mu') = \varphi_{OS}(\mu) - \varphi_{OS}(\lambda) + \varphi_{OS}(\rho)) \cong \mu - \lambda + \rho = \mu'$. $\square$

Example 3. In the following we give an example to illustrate our approach. Our Eos models a kitchen with two working stations S_1 and S_2, i.e., places where different activities can be executed. The whole scenario can be seen as a metaphor for flexible manufacturing systems, where the kitchen is the plant and the recipe is the workflow.

We have a cook (corresponding to a robot) executing a simple recipe: First split eggs (action a); then, independently mix egg yolks with sugar (action b)

and mix egg white with wine (action c); and, finally fill the white creme on top of pudding (action d). In process algebraic notation the recipe is denoted as $a; (b\|c); d$, where $\_;\_$ denotes sequential composition and $\_\|\_$ denotes an and-split (parallel execution).

The corresponding Eos is shown in Figure 3. The system net has two places S_1 and S_2 for the kitchen stations. Each station has side transitions to execute those actions that are possible at each station. The cook/recipe can move freely between the two stations. We have only one object net that models the recipe. In the initial marking the recipe starts at station S_1. The symmetry of the locations is captured by the automorphism: $\varphi_{OS}(S_1) = S_2$, $\varphi_{OS}(p_1) = p_2$, and $\varphi_{OS}(p_3) = p_4$.

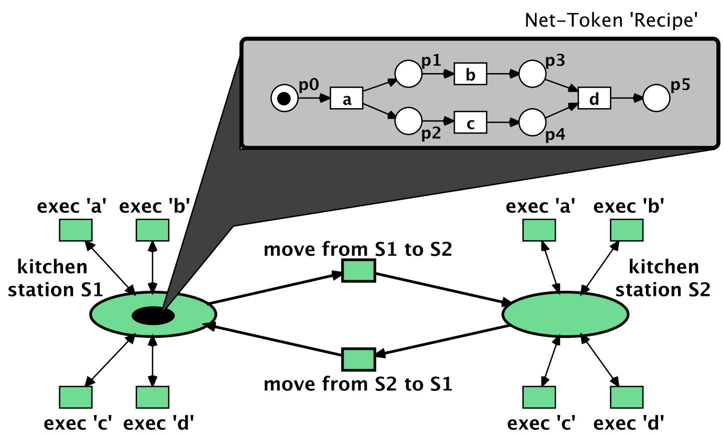

Fig. 3. Eos: A Kitchen with a Recipe as a Net-Token

The resulting state space mainly describes the following execution: The recipe moves from kitchen station S_1 where a is possible to station S_2, where it is possible to execute both b and c (in any order). Finally, the recipe moves back to station S_1 to execute d. In between of all these steps the recipe may moves 'erratically' between the two locations without making any progress to the recipe.

3.2 Canonical Representations of Eos

Our basic idea is to provide some pseudo symbolic state representation based on automorphism. We use canonical representatives, i.e., a most minimal representation of an equivalence class (corresponding to an automorphism group) considering a total order on nodes: e.g., we assume that for any p/t net N there are two bijections $P_N \to \{1, \ldots, |P_N|\}$ and $T_N \to \{1, \ldots, |T_N|\}$. We derive the Eos-automorphism from the canonical representative of p/t components (net-tokens and system-net).

Example 4. Let us illustrate the main concepts using Figure 3 (here $\widehat{p}_1$ means S_1, $\widehat{p}_2$ means S_2). The following markings:

$$\mu = \widehat{p}_1[p_1 + p_4] + \widehat{p}_1[p_2 + p_3] + \widehat{p}_2[p_2 + p_3]$$

$$\mu' = \widehat{p}_1[p_2 + p_3] + \widehat{p}_2[p_1 + p_4] + \widehat{p}_2[p_2 + p_3]$$

are actually automorphic, their canonical representative is (we use weights for clarity)

$$\mu_c = \widehat{p}_1[p_1 + p_4] + 2 \cdot \widehat{p}_2[p_1 + p_4]$$

In this example, the state aggregation obtained through EOS automorphism and canonization grows exponentially with the quantity of net-tokens.

We integrate the formalization of canonization for EOS along the results for Rewritable Petri Nets as given in previous work [5, 3]. There we used the well established `Maude` tool [7] to define canonical representation of *Rewritable p/t* Nets. This formalization utilizes a multiset-based algebraic definition of mutable *p/t* nets. Whereas the method detailed in [3] is generic (similar to other graph canonization techniques), the one outlined in [5] is specific to modular *p/t* nets created and modified using selected composition operators and a structured node labelling, making it significantly more efficient for symmetric models.

The two approaches can be seamlessly incorporated into the EOS formalism within `Maude` [4], which uses a consistent representation of *p/t* terms.

In theory, it is known that generating canonical representative is at least as complex as the graph automorphism problem (quasi polynomial). We hope for a more efficient canonization since for EOS the *p/t* structure doesn't change: the canonical representative of a marking must retain the net structure, which is a major source for efficiency. While the exact effect on efficiency is still being studied, the application of the method [5] appears to be promising, when EOS components are built modularly.

Observe that the proposed canonization approach is pseudo-symbolic, because from a canonical representative, we might reach equivalent markings through equivalent instances. To approach entirely symbolic, canonical representative of firing modes should be calculated (using a uniform technique).

We finally like to mention that in [10] we have considered another 'in-built' symmetry of EOS. The nested multisets α and β that coincide in their projections give rise to the so-called *projection equivalence*:

$$\alpha \cong \beta :\Longleftrightarrow \Pi^1(\alpha) = \Pi^1(\beta) \wedge \forall N \in \mathcal{N} : \Pi^2_N(\alpha) = \Pi^2_N(\beta) \tag{9}$$

Lemma 2 ([10], Lem. 3.1). *The enabling predicate is invariant with respect to projection:* $\phi(\widehat{\tau}[\vartheta], \lambda, \rho) \iff (\forall \lambda', \rho' : \lambda' \cong \lambda \wedge \rho' \cong \rho \Longrightarrow \phi(\widehat{\tau}[\vartheta], \lambda', \rho'))$

This equivalence will give rise to additional state space reductions.

4 Conclusion

In this paper, we have extended the concepts of automorphisms for Nets-within-Nets, here: Eos. In combination with canonical forms for representing the markings we have obtained a potential significant reduction of the size of the state space. We have integrated the approach in a `Maude` encoding of Eos.

In future work, we plan to investigate the degree of state space reduction for a broader set of case studies. We further hypothesize that the automorphism concept for Eos can be readily extended to Object-nets with any level of net and channel nesting, allowing net-tokens to flow between different levels. We also aim to explore the practicality of less restrictive partial symmetry concepts.

References

1. Bednarczyk, M.A., Bernardinello, L., Pawlowski, W., Pomello, L.: Modelling mobility with Petri hypernets. In: Fiadeiro, J.L., Mosses, P.D., Orejas, F. (eds.) Recent Trends in Algebraic Development Techniques (WADT 2004). Lecture Notes in Computer Science, vol. 3423, pp. 28–44. Springer-Verlag (2004)
2. Capra, L.: A maude implementation of rewritable petri nets: a feasible model for dynamically reconfigurable systems. In: Gleirscher, M., Pol, J.v.d., Woodcock, J. (eds.) Proceedings First Workshop on Applicable Formal Methods. Electronic Proceedings in Theoretical Computer Science, vol. 349, pp. 31–49. Open Publishing Association (2021)
3. Capra, L.: Canonization of reconfigurable pt nets in Maude. In: Lin, A.W., Zetzsche, G., Potapov, I. (eds.) Reachability Problems. pp. 160–177. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-19135-0_11
4. Capra, L., Köhler-Bußmeier, M.: Modelling adaptive systems with nets-within-nets in maude. In: Proceedings of the 18th International Conference on Evaluation of Novel Approaches to Software Engineering - ENASE, Prague, Czech Republic. pp. 487–496. INSTICC, SciTePress (2023). https://doi.org/10.5220/0011860000003464
5. Capra, L., Köhler-Bußmeier, M.: Modular rewritable Petri nets: an efficient model for dynamic distributed systems. Theoretical Computer Science **990**, 114397 (2024). https://doi.org/10.1016/j.tcs.2024.114397
6. Cardelli, L., Gordon, A.D., Ghelli, G.: Mobility types for mobile ambients. In: Proceedings of the Conference on Automata, Languages, and Programming (ICALP'99). Lecture Notes in Computer Science, vol. 1644, pp. 230–239. Springer-Verlag (1999)
7. Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcott, C.L. (eds.): All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic, Lecture Notes in Computer Science, vol. 4350. Springer-Verlag (2007)
8. Hiraishi, K.: PN^2: An elementary model for design and analysis of multi-agent systems. In: Arbab, F., Talcott, C.L. (eds.) Coordination Models and Languages, COORDINATION 2002. Lecture Notes in Computer Science, vol. 2315, pp. 220–235. Springer-Verlag (2002)
9. Köhler, M., Rölke, H.: Properties of Object Petri Nets. In: Cortadella, J., Reisig, W. (eds.) International Conference on Application and Theory of Petri Nets 2004. Lecture Notes in Computer Science, vol. 3099, pp. 278–297. Springer-Verlag (2004)

10. Köhler-Bußmeier, M.: A survey on decidability results for elementary object systems. Fundamenta Informaticae **130**(1), 99–123 (2014)
11. Köhler-Bußmeier, M., Heitmann, F.: On the expressiveness of communication channels for object nets. Fundamenta Informaticae **93**(1-3), 205–219 (2009)
12. Köhler-Bußmeier, M., Heitmann, F.: Safeness for object nets. Fundamenta Informaticae **101**(1-2), 29–43 (2010)
13. Köhler-Bußmeier, M., Heitmann, F.: Liveness of safe object nets. Fundamenta Informaticae **112**(1), 73–87 (2011)
14. Lomazova, I.A.: Nested Petri nets – a formalism for specification of multi-agent distributed systems. Fundamenta Informaticae **43**(1-4), 195–214 (2000)
15. Lomazova, I.A., van Hee, K.M., Oanea, O., Serebrenik, A., Sidorova, N., Voorhoeve, M.: Nested nets for adaptive systems. In: Application and Theory of Petri Nets and Other Models of Concurrency. pp. 241–260. Lecture Notes in Computer Science, Springer-Verlag (2006)
16. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, parts 1-2. Information and computation **100**(1), 1–77 (1992)
17. Padberg, J., Kahloul, L.: Overview of reconfigurable Petri nets. In: Heckel, R., Taentzer, G. (eds.) Graph Transformation, Specifications, and Nets, vol. 10800, pp. 201–222. Springer-Verlag (2018)
18. Valk, R.: Object Petri nets: Using the nets-within-nets paradigm. In: Desel, J., Reisig, W., Rozenberg, G. (eds.) Advanced Course on Petri Nets 2003. Lecture Notes in Computer Science, vol. 3098, pp. 819–848. Springer-Verlag (2003)
19. Xu, D., Deng, Y.: Modeling mobile agent systems with high level Petri nets. In: IEEE International Conference on Systems, Man, and Cybernetics'2000 (2000)