

Structural and Dynamic Restrictions of Elementary Object Systems

Frank Heitmann and Michael Köhler-Bußmeier

University of Hamburg, Department for Informatics
Vogt-Kölln-Straße 30, D-22527 Hamburg
{heitmann,koehler}@informatik.uni-hamburg.de

Abstract. Elementary object systems (EOS for short) are Petri nets in which tokens may be Petri nets again. Originally proposed by Valk for a two levelled structure, the formalism was later generalised for arbitrary nesting structures.

However, even if restricted to a nesting depth of two, EOS are Turing-complete and thus many problems like reachability and liveness are undecidable for them. Nonetheless, since they are useful to model many practical applications a natural question is how to restrict the formalism in such a way, that the resulting restricted formalism is still helpful in a modelling context, but so that important verification problems like reachability become quickly decidable.

In the last years several structural and dynamic restrictions for EOS have therefore been investigated. These investigations have been central to the first author's recent PhD thesis and have been published in past CS&P conferences. In this paper we add several new results and present them together with the old in a unified fashion highlighting the central message of these investigations.

1 Introduction

Object Petri Nets are Petri Nets whose tokens may be Petri Nets again and thus may have an inner structure and activity. This approach is useful to model mobile systems and other systems arising in Computer Science which enjoy a certain nesting of structures (cf. [13] and [14]).

This approach, which is also called the *nets-within-nets* paradigm, was proposed by Valk [27, 28] for a two levelled structure and generalised in [15, 16] for arbitrary nesting structures. By now many related approaches like recursive nets [6], nested nets [24], adaptive workflow nets [25], AHO systems [11], PN² [10], Mobile Systems [23], and many others are known. Another line of research also dealing with nesting, but not in the field of Petri nets, is concerned with process calculi. Arguably most prominently there are the Ambient Calculus of Gordon and Cardelli [1] and the Seal Calculus [2] among many others. See [18] and [7] for a detailed discussion.

Unfortunately even if restricted to a two level structure as in elementary object systems, the formalism is Turing-complete. A natural question is how to restrict the formalism in such a way, that the resulting restricted formalism is still helpful in a modelling context, but so that important verification problems like reachability become quickly decidable. This “borderline” between modelling power and fast algorithms is in the case of p/t nets usually drawn at free choice Petri nets.

In the following we survey several structural restrictions for EOS and give results concerning the complexity of the reachability problem. Since we conclude, that even in very restricted cases the reachability problem becomes practically hard to decide, we then survey dynamic restrictions of EOS, most notably a safeness notion. Here, too, we focus on the complexity on the reachability problem.

The following section gives fundamental definitions of EOS. In Section 3 we survey results on structural restrictions and in Section 4 we survey results on dynamic restrictions. The paper ends with a summary of these results and a conclusion.

In the following we assume basic knowledge of Petri nets, see e.g. [26] and of EOS, see e.g. [18]. We do not define all notions rigorously here due to space restrictions, but all notions and an in-depth study can be found in [7].

2 Fundamentals

An elementary object system (EOS) is composed of a system net and a set of object nets where each of these nets is a p/t net. While the object nets use the usual black tokens, the system net’s places are marked with either black tokens or object nets. For this each place of the system net is typed with an object net with the meaning the only object nets of these type may rest on that place. Additionally each transition may be labelled with a channel. The meaning here is that transitions with the same label may only fire synchronously.

Definition 1 (EOS). *An elementary object system (EOS) is a tuple $OS = (\widehat{N}, \mathcal{N}, d, l)$ such that:*

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. $d : \widehat{P} \rightarrow \mathcal{N}$ is the typing of the system net places.
4. $l = (\widehat{l}, (l_N)_{N \in \mathcal{N}})$ is the labelling.

An EOS with initial marking is a tuple $OS = (\widehat{N}, \mathcal{N}, d, l, \mu_0)$ where $\mu_0 \in \mathcal{M}$ is the initial marking.

A system net transition may be labelled with a channel for each object net. Where an object net transition is only labelled with one channel. If for example a system net transition \widehat{t} is labelled with channel c_1 for the object net N_1 and with channel c_2 for object net N_2 , then \widehat{t} may only fire, if it is possible to *synchronously* fire a transition in N_1 which is labelled with channel c_1 and if it is possible to

synchronously fire a transition in N_2 which is labelled with channel c_2 . Firing may also happen system-autonomously (an unlabelled system net transition fires independently from any object net transition) or object-autonomously (firing of an unlabelled object net transition)

A formal treatment can be found in e.g. [9]. We only give an example here to illustrate the main points of the firing rule.

Example 1. Figure 1 shows an EOS with the system net \hat{N} and the object nets $\mathcal{N} = \{N_1, N_2\}$. The system has four net-tokens: two on place p_1 and one on p_2 and p_3 each. The net-tokens on p_1 and p_2 share the same net structure, but have independent markings.

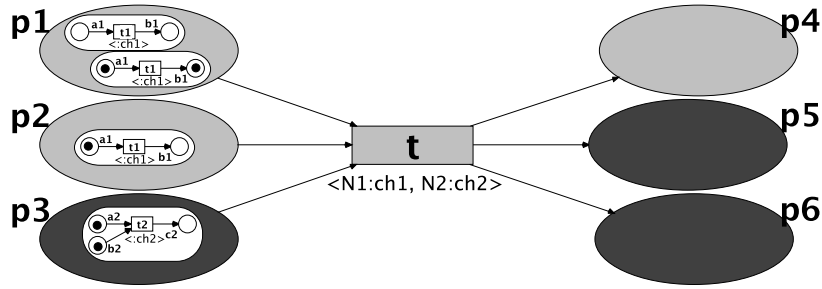


Fig. 1. An Elementary Object Net System

Formally we have the system net $\hat{N} = (\hat{P}, \hat{T}, \mathbf{pre}, \mathbf{post})$ with the places and transitions given by $\hat{P} = \{p_1, \dots, p_6\}$ and $\hat{T} = \{t\}$, the object net $N_1 = (P_1, T_1, \mathbf{pre}_1, \mathbf{post}_1)$ with $P_1 = \{a_1, b_1\}$ and $T_1 = \{t_1\}$ and the object net $N_2 = (P_2, T_2, \mathbf{pre}_2, \mathbf{post}_2)$ with $P_2 = \{a_2, b_2, c_2\}$ and $T_2 = \{t_2\}$. The typing is given by $d(p_1) = d(p_2) = d(p_4) = N_1$ and $d(p_3) = d(p_5) = d(p_6) = N_2$.

We have two channels ch_1 and ch_2 . The labelling function \hat{l} of the system net is defined by $\hat{l}(t)(N_1) = ch_1$ and $\hat{l}(t)(N_2) = ch_2$. The labelling l_{N_1} of the first object net is defined by setting $l_{N_1}(t_1) = ch_1$. Similarly, l_{N_2} is defined by $l_{N_2}(t_2) = ch_2$.

There is only one (synchronous) event: $\Theta = \Theta_l = \{t[N_1 \mapsto t_1, N_2 \mapsto t_2]\}$. The event is also written shortly as $t[t_1, t_2]$.

The initial marking μ has two net-tokens on p_1 , one on p_2 , and one on p_3 :

$$\mu = p_1[a_1 + b_1] + p_1[\mathbf{0}] + p_2[a_1] + p_3[a_2 + b_2]$$

Note that for Figure 1 the structure is the same for the three net-tokens on p_1 and p_2 but the net-tokens' markings are different.

The marking μ enables $t[N_1 \mapsto t_1, N_2 \mapsto t_2]$ in the mode (λ, ρ) , where

$$\begin{aligned} \mu &= p_1[\mathbf{0}] + p_1[a_1 + b_1] + p_2[a_1] + p_3[a_2 + b_2] = p_1[\mathbf{0}] + \lambda \\ \lambda &= p_1[a_1 + b_1] + p_2[a_1] + p_3[a_2 + b_2] \\ \rho &= p_4[a_1 + b_1 + b_1] + p_5[\mathbf{0}] + p_6[c_2] \end{aligned}$$

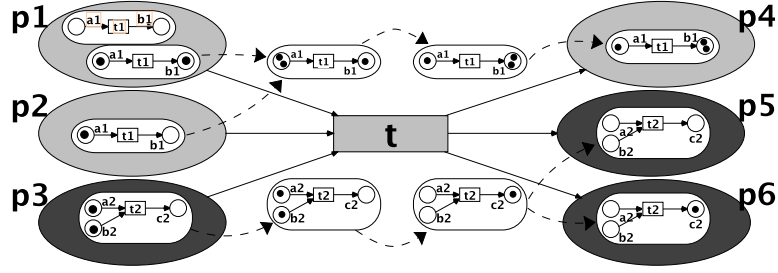


Fig. 2. The EOS of Figure 1 illustrating the projections $\Pi_N^2(\lambda)$ and $\Pi_N^2(\rho)$

The net-tokens' markings are added by the projections Π_N^2 resulting in the markings $\Pi_N^2(\lambda)$. The sub-synchronisation generate $\Pi_N^2(\rho)$. (The results are shown above and below the transition t in Figure 2.) After the synchronisation we obtain the successor marking μ' with net-tokens on p_4 , p_5 , and p_6 as shown in Figure 2:

$$\begin{aligned}\mu' &= (\mu - \lambda) + \rho = p_1[\mathbf{0}] + \rho \\ &= p_1[\mathbf{0}] + p_4[a_1 + b_1 + b_1] + p_5[\mathbf{0}] + p_6[c_2]\end{aligned}$$

For general EOS the following theorem holds due to Köhler-Bußmeier [12].

Theorem 1 (Köhler 2007). *EOS can simulate 2-counter machines. Important problems like reachability and liveness are thus undecidable.*

In the following two sections we focus on introducing structural and dynamic restrictions that result in a decidable reachability problem.

3 Structural Restrictions

The main reason why EOS are Turing-complete is a null-test that is possible due to the firing rule. In Figure 3 the system net transition \hat{t}_1 is not able to fire: the object net in the preset has a token on place a_1 . This token would have to be distributed among the places of the same object net type in the postset of \hat{t}_1 , but there is no such place. The transition \hat{t}_2 to the right may fire. The object net's marking is $\mathbf{0}$ and so there are no tokens that need to be distributed.

Conservative EOS avoid this by demanding that each object net type that appears in the preset of a system net transition also appears in the postset of that transition.

Definition 2 (Conservative Eos). *A typing is called conservative iff*

$$(d(\bullet\hat{t}) \cup \{\bullet\}) \subseteq (d(\hat{t}\bullet) \cup \{\bullet\}),$$

i.e. each object net type that appears in the preset of \hat{t} also appears in its postset. An EOS is conservative iff its typing d is.

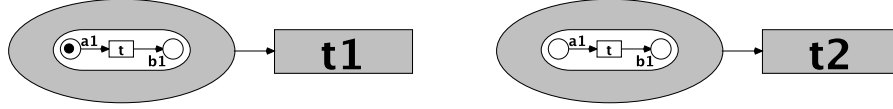


Fig. 3. The transition \hat{t}_1 is disabled, \hat{t}_2 is enabled.

While boundedness and coverability become decidable for conservative EOS, reachability and liveness remain undecidable. This was proven in [12] and [21].

Theorem 2. *For conservative EOS boundedness and coverability are decidable, while reachability and liveness are undecidable.*

The idea in the definition of conservative EOS can be strengthened further by demanding that each object net type appears exactly once in the preset and the postset of a system net transition or does not appear at all. This leads to *generalised state machines* (GSM) which are suitable to model many practical applications, since each object net can be seen as a physical entity.

Definition 3 (Generalised State Machines). *Let $G = (\hat{N}, \mathcal{N}, d, l, \mu_0)$ be an EOS. G is a generalised state machine (GSM) iff for all $N \in \mathcal{N} \setminus \{N_\bullet\}$*

1. $\forall \hat{t} \in \hat{T} : |\{\hat{p} \in \bullet \hat{t} \mid d(\hat{p}) = N\}| = |\{\hat{p} \in \hat{t} \bullet \mid d(\hat{p}) = N\}| \leq 1$
2. $\sum_{\hat{p} \in \hat{P}, d(\hat{p})=N} \Pi^1(\mu_0)(\hat{p}) \leq 1$

holds.

In [17]) it was shown that for each GSM a p/t net, called reference net, can be easily constructed from which decidability results follow (see [12] and [17]).

Theorem 3. *The reachability problem is decidable for generalised state machines.*

While this is a first positive result, the set of transitions is defined by the set of events of the GSM and the size of this set can become exponential in the size of the GSM.

Lemma 1 ([8], [9]). *Let $|T| := \max\{|T_N| \mid N \in \mathcal{N}\}$ then $|\Theta| \leq |\hat{T}| \cdot |T|^{|\mathcal{N}|}$.*

Given a GSM it might thus be very expensive to construct its reference net. This exponential blow up stems from the fact that in a GSM a $m : n$ -synchronisation between the system net and the object nets exists, i.e. if there are m system net transitions labelled with channel c for object net N_1 and if in N_1 n transitions are labelled with channel c then each of the m system net transitions may fire synchronously with each of the n object net transitions, resulting in $m \cdot n$ different events.

To prevent this, we introduced deterministic GSMs and EOS in [8] (see also [9]).

Definition 4 (Deterministic and Strongly Deterministic Eos). A EOS OS is called deterministic if for each $N \in \mathcal{N}$ and every two transitions $t, t' \in T_N$, $t \neq t'$ with $l_N(t) \neq \tau \neq l_N(t')$, $l_N(t) \neq l_N(t')$ holds, i.e. if the labels for all all $t \in T_N$ with $l_N(t) \neq \tau$ are pairwise different.

OS is strongly deterministic if OS is deterministic and additionally for all \hat{t} and N with $\hat{l}(\hat{t})(N) \neq \tau$ the labels $\hat{l}(\hat{t})(N)$ are pairwise different.

Thus, in a deterministic EOS or GSM each channel is used at most once in each object net (resulting in a $m : 1$ -synchronisation). In a strongly deterministic EOS or GSM each channel is additionally used at most once in the system net (resulting in a $1 : 1$ -synchronisation).

For EOS the definition of determinism does not significantly reduce the power of the formalisms introduced so far, namely of EOS or conservative EOS.

Theorem 4 ([7]). *The reachability problem for strongly deterministic, deterministic and general EOS is undecidable - even if the EOS is conservative.*

However, for GSMs the size of the events and thus the size of the reference net is reduced considerably.

Lemma 2 ([7]). *Let $G = (\hat{N}, \mathcal{N}, d, l, \mu_0)$ be a deterministic or strongly deterministic GSM, then $|\Theta|$ is bounded above by $|\hat{T}| + \sum_{N \in \mathcal{N}} |T_N|$.*

However, strongly deterministic GSMs can still simulate p/t nets and thus the reachability problem remains EXPSPACE-hard.

Theorem 5 ([7]). *Every p/t net system \mathcal{N} can be simulated by a strongly deterministic GSM $G_{\mathcal{N}}$.*

Corollary 1 ([7]). *The reachability problem for strongly deterministic, deterministic and general GSMs is EXPSPACE-hard.*

Due to this results further structural restrictions are necessary to reduce the complexity of the reachability problem. Several structural restrictions known from p/t nets are carried over to GSMs and investigated in [8], [9], [21], and [7].

The results are summarised in Table 1. ttGSMs, ppGSMs, ptGSMs, and tpGSMs are GSMs where the system net and/or the object nets are restricted to be T-nets or P-nets. In ptGSMs the system net is a P-net and the object nets are T-nets and in tpGSMs the system net is a T-net and the object nets are P-nets. Despite being rather simple in the case of p/t nets, it is evident in Table 1, that the reachability problem becomes hard in the case of object nets due to the synchronisation between the system net and the object nets. Most notably in the case of ppGSMs where all participating nets are P-nets and thus similar to finite automata, the reachability problem is already PSPACE-complete.

acGSMs are GSMs where all nets are acyclic and fcGSMs are GSMs where all nets are free-choice nets. For cfGSMs the definition of conflict-freedom has to be adapted, because it is not enough to demand a similar structural restrictions as for p/t nets if one wants to structurally rule out conflicts (cf. [8] and [7]).

Table 1. Complexity of the reachability problem for various formalisms.

	strongly deterministic	deterministic	general
ttGSM	P	?	?
ppGSM	PSPACE-complete	PSPACE-complete	PSPACE-complete
ptGSM	NP-hard	PSPACE-hard	PSPACE-hard
tpGSM	?	?	?
acGSM	NP-hard	NP-hard	NP-hard
cfGSM	NP-complete	NP-hard	NP-hard
fcGSM	EXPSpace-hard	EXPSpace-hard	EXPSpace-hard
GSM	EXPSpace-hard	EXPSpace-hard	EXPSpace-hard
cEOS	undecidable	undecidable	undecidable
EOS	undecidable	undecidable	undecidable

Definition 5 (Conflict-Free GSMs ([8])). A GSM $OS = (\widehat{N}, \mathcal{N}, d, l, \mu_0)$ is conflict-free if

1. $\forall N \in \mathcal{N} \cup \{\widehat{N}\} \forall p \in \widehat{P} \cup P_N : |p^\bullet| > 1 \Rightarrow p^\bullet \subseteq \bullet p$
2. $\forall N \in \mathcal{N} \forall p \in P_N : (\exists t \in p^\bullet \exists \widehat{t}_1, \widehat{t}_2 \in \widehat{T} \exists \widehat{p} \in \widehat{P} \exists c \in C : \widehat{t}_1 \neq \widehat{t}_2 \wedge \widehat{p} \in \bullet \widehat{t}_1 \cap \bullet \widehat{t}_2 \wedge d(\widehat{p}) = N \wedge l_N(t) = \widehat{l}(\widehat{t}_1)(N) = \widehat{l}(\widehat{t}_2)(N) = c) \Rightarrow p \in t^\bullet$

holds. We also say that OS is a cfGSM.

Proofs for the results in Table 1 can be found in [7] with pointers to the literature where the results were first proven.

Most notably in Table 1 are the results for ppGSMs and for fcGSMs. They show that even very strong structural restrictions as in the case of ppGSMs lead to an already hard to solve reachability problem and that more openly structural restrictions as in the case of fcGSMs where the formalism would be suitable for modelling purposes lead to complexity results that render algorithms practically unusable. Thus other restrictions than structural restrictions are necessary if one aims at solving the reachability problem quickly.

4 Dynamics Restrictions

In [7] unary and persistent EOS are introduced where in unary EOS in each reachable marking at most one event is enabled and in persistent EOS conflicts are dynamically ruled out. For both formalisms, however, the reachability problem remains undecidable. These dynamical restrictions are thus not strong enough for our purpose.

In general, the state space of an EOS is of infinite size which is a source for undecidability results or strong lower complexity bounds. In [19] we therefore introduced four different notions of safeness for EOS, safe(1), safe(2), safe(3), and safe(4), to adapt the notion of safeness for p/t nets to EOS. For p/t nets 1-safeness guarantees not only finiteness of the state space size but also that each reachable marking can be seen as a set. This set idea is adapted in [19]. Furthermore, a safe(4) EOS is also a safe(3) EOS which is in turn a safe(2) EOS

and so on. Positive results concerning the solvability of the reachability problem thus carry over from $\text{safe}(i)$ to $\text{safe}(i + 1)$ and negative results carry over from $\text{safe}(i + 1)$ to $\text{safe}(i)$.

However, despite the fact that the markings are sets, $\text{safe}(1)$ and $\text{safe}(2)$ EOS still have an infinite state space and the reachability problem remains undecidable for them even if the EOS is additionally structurally restricted.

Theorem 6 ([19], [7]). *The reachability problem is undecidable for $\text{safe}(1)$ or $\text{safe}(2)$ EOS - even in the case of strongly deterministic and conservative EOS.*

In the following we concentrate on $\text{safe}(3)$ EOS which have a finite state space.

Definition 6 (Safeness). *An EOS OS is $\text{safe}(3)$ or simply safe iff for all reachable markings there is at most one token on each system net place and each net-token is safe:*

$$\forall \mu \in RS(OS) : \forall \hat{p} \in \hat{P} : \Pi^1(\mu)(\hat{p}) \leq 1 \wedge \\ \forall N \in \mathcal{N} : \forall p \in P_N : \forall \hat{p}[M] \leq \mu : M(p) \leq 1$$

Theorem 7 ([19], [7]). *If an EOS is $\text{safe}(3)$ or $\text{safe}(4)$, then its set of reachable markings is finite.*

Indeed, an upper bound for the state space size is given in [7]. Let $k := |\hat{P}|$ and $l := \max\{|P_N| \mid N \in \mathcal{N}\}$, then there are at most $(1 + 2^l)^k$ different markings.

For safe EOS two very strong results can be shown. Not only are reachability and liveness decidable, but every property that can be expressed in the temporal logics LTL or CTL can be decided in polynomial space in the size of the EOS and the formula. The problems are PSPACE-hard follows directly from similar results for safe p/t nets (see e.g. [5]). It is thus surprising that this bound can also be met from above in the case of safe EOS despite their quite huge state space.

Theorem 8 ([19], [7]). *Given a $\text{safe}(3)$ or $\text{safe}(4)$ EOS OS and an LTL formula ϕ , checking whether OS satisfies ϕ can be done in polynomial space in the size of OS and ϕ , that is, there is a polynomial p , independent of OS and ϕ , such that the algorithm uses $O(p(|OS| + |\phi|))$ space.*

Corollary 2. *The reachability problem for safe EOS is PSPACE-complete.*

Theorem 9 ([20], [7]). *Given a $\text{safe}(3)$ or $\text{safe}(4)$ EOS OS and a CTL formula ϕ checking whether OS satisfies ϕ can be done in $O(|OS|^4 \cdot |\phi|)$ space.*

Corollary 3. *The liveness problem for safe EOS is PSPACE-complete.*

Both model checking algorithms are an adaptation of a technique from Esparza for 1-safe p/t nets [5]. The LTL model checking algorithms additionally use techniques from Vardi from automata theory [29].

The proofs of the above theorems are very involved. Detailed discussions and proofs can be found in [19], [20], and [7].

In addition to the results above it can also be decided in polynomial space if an EOS is $\text{safe}(3)$, which is helpful from a modelling point of view.

Theorem 10 ([7]). *Given an EOS OS it is PSPACE-complete to decide if OS is safe.*

Safe EOS can in addition be structurally restricted. In some cases as in the case of conflict-free EOS this reduces the complexity of the reachability problem (cf. Table 2). However, the formalisms where this happens seem to be too restricted to be useful in a modelling context.

Table 2. Complexity of the reachability problem for *safe* EOS with further structural restrictions.

	strongly deterministic	deterministic	general
ttGSM	P	PSPACE	PSPACE
ppGSM	PSPACE-complete	PSPACE-complete	PSPACE-complete
ptGSM	NP-hard, PSPACE	PSPACE-complete	PSPACE-complete
tpGSM	PSPACE	PSPACE	PSPACE
acGSM	PSPACE	PSPACE	PSPACE
cfGSM	P	PSPACE	PSPACE
fcGSM	PSPACE-complete	PSPACE-complete	PSPACE-complete
GSM	PSPACE-complete	PSPACE-complete	PSPACE-complete
cEOS	PSPACE-complete	PSPACE-complete	PSPACE-complete
EOS	PSPACE-complete	PSPACE-complete	PSPACE-complete

A discussion of structural restrictions of safe(1) and safe(2) EOS can be found in [7].

5 Conclusion and Outlook

In summary we have investigated formalisms which are useful to model mobility, interaction, and nesting of structures. We then focused on object nets, but it is possible to adapt these results for other formalisms mentioned in the introduction. Since in their general form object nets are Turing-complete it was our goal to restrict the formalism of elementary object systems such that modelling capabilities mostly remain and interesting applications can still be conveniently modelled, and also such that properties can be automatically and quickly verified.

To achieve this goal we have restricted the formalism structurally and dynamical. We introduced restrictions natural for object nets like determinism and strong determinism and carried over restrictions known from p/t nets to object nets. We then focused on the complexity of the reachability problem to evaluate the formalisms. The results are summarised in Table 1 and 2.

As a conclusion structural restrictions alone are not enough even if the possibility to synchronise is additionally restricted. The restriction to safe(3) EOS, however, allows for a quick verification of important properties and is also still useful from a modelling point of view.

In [7] EOS and GSMs are furthermore generalised to an arbitrary but fixed nesting depth $k > 2$. In this case a safeness definition can also be introduced which allows to carry over the results for CTL and LTL model checking. These problems are then solvable in polynomial space, too, albeit the polynomial worsens.

Also in [7] object net systems are introduced, which allow a vertical transport of net tokens, i.e. a transport of net tokens between nesting levels. For this systems different safeness definitions are introduced in [7]. In particular, it is not enough to demand that on each place resides at most one (net) token. Due to the arbitrary nesting depth, the state space might still be infinite. However, for the strongest safeness definition for object net systems in [7], which among other things does not allow the creation and destruction of net tokens, again PSPACE-completeness results for LTL- and CTL-model-checking can be established.

Open question in the context presented here are, obviously, to match upper and lower bounds in the tables above. While this might be interesting from a theoretical point of view, the effect for the question tackled here are limited. The open cases are for formalisms which are structurally restricted in such a way, that using them in a modelling context is doubtful. Thus even if the reachability problem is solvable more quickly than PSPACE the formalisms will not be practically useful.

Another open question concerns the borderline between safe(1) and safe(2) EOS and the borderline between safe(3) and safe(4) EOS. While it is known that e.g. for safe(3) and safe(4) EOS the reachability problem is PSPACE-complete, the polynomial will almost surely be smaller for safe(4) EOS. Again this question is more from a theoretical nature.

A question with much practical relevance is how to improve the LTL- and CTL-model-checking algorithms uses so far in this context. In [7] the algorithm are rather direct and not optimised. It is to be expected that these algorithms can be improved. Afterwards, it would be nice to implement these algorithms in a tool which can then be used by modellers. We believe that such a tool might be very useful in practice, since many applications can be modelled more intuitively with object nets and such a tool would allow to verify properties of the model automatically and with modest time and space requirements.

Questions not tackled so far which would open up whole new directions are *compositionality* and *adaptivity*. The goal in compositionality is to find properties and restrictions such that properties of the whole system can than be deduced from properties of components treated in isolation. This might reduce the complexity considerably. First results concerning compositionality with regard to nested nets are published in [3] and [4].

Adaptivity means to introduce formalisms which do not only allow a transport of net tokens as presented here, but also to manipulated these net tokens during run-time. First formalisms which allow this are adaptive workflow nets [25] and higher order nets [22].

This two questions, how to introduce compositionality and adaptivity for object nets, are the questions we want to tackle in the future. We believe that

they will be of high practical relevance for run-time analysis of systems in general and workflows in particular.

References

1. Luca Cardelli and Andrew D. Gordon. Mobile ambients. *Theoretical Computer Science*, 240:177–213, 2000.
2. G. Castagna, J. Vitek, and F. Zappa Nardelli. The seal calculus. *Information and Computation*, 201:1–54, 2005.
3. Leonid W. Dworzański and Irina A. Lomazova. On compositionality of boundedness and liveness for nested Petri nets. In Marcin Szczuka, Ludwik Czaja, Andrzej Skowron, and Magdalena Kacprzak, editors, *Concurrency, Specification and Programming (CS&P 2011), Proceedings*, Pułtusk, Poland, 2011. Białystok University of Technology.
4. Leonid W. Dworzański and Irina A. Lomazova. On compositionality of boundedness and liveness for nested Petri nets. *Fundamenta Informaticae*, 120(3–4):275–293, 2012.
5. Javier Esparza. Decidability and complexity of petri net problems – an introduction. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets*, volume 1491 of *Lecture Notes in Computer Science*, pages 374–428. Springer-Verlag, 1998.
6. Serge Haddad and Denis Poitrenaud. Theoretical aspects of recursive Petri nets. In S. Donatelli and J. Kleijn, editors, *Application and Theory of Petri Nets*, volume 1639 of *Lecture Notes in Computer Science*, pages 228–247. Springer-Verlag, 1999.
7. Frank Heitmann. *Algorithms and Hardness Results for Object Nets*. PhD thesis, University of Hamburg, 2013.
8. Frank Heitmann and Michael Köhler-Bußmeier. On defining conflict-freedom for object nets. In B. Farwer and M. Köhler-Bußmeier, editors, *Proceedings of the Second International Workshop on Logic, Agents, and Mobility (LAM 2011)*, 2011.
9. Frank Heitmann and Michael Köhler-Bußmeier. P- and t-systems in the nets-within-nets-formalism. In Serge Haddad and Lucia Pomello, editors, *Application and Theory of Petri Nets. 33rd International Conference, PETRI NETS 2012. Hamburg, Germany, June 2012. Proceedings*, volume 7347 of *Lecture Notes in Computer Science*, pages 368–387. Springer-Verlag, 2012.
10. Kunihiro Hiraishi. PN²: An elementary model for design and analysis of multi-agent systems. In Farhad Arbab and Carolyn L. Talcott, editors, *Coordination Models and Languages, COORDINATION 2002*, volume 2315 of *Lecture Notes in Computer Science*, pages 220–235. Springer-Verlag, 2002.
11. Kathrin Hoffmann, Hartmut Ehrig, and Till Mossakowski. High-level nets with nets and rules as tokens. In *Application and Theory of Petri Nets and Other Models of Concurrency*, volume 3536 of *Lecture Notes in Computer Science*, pages 268 – 288. Springer-Verlag, 2005.
12. Michael Köhler. The reachability problem for object nets. *Fundamenta Informaticae*, 79(3-4):401 – 413, 2007.
13. Michael Köhler, Daniel Moldt, and Heiko Rölke. Modeling the behaviour of Petri net agents. In J. M. Colom and M. Koutny, editors, *Application and Theory of Petri Nets*, volume 2075 of *Lecture Notes in Computer Science*, pages 224–241. Springer-Verlag, 2001.

14. Michael Köhler, Daniel Moldt, and Heiko Rölke. Modelling mobility and mobile agents using nets within nets. In W. v. d. Aalst and E. Best, editors, *Application and Theory of Petri Nets*, volume 2679 of *Lecture Notes in Computer Science*, pages 121–140. Springer-Verlag, 2003.
15. Michael Köhler and Heiko Rölke. Concurrency for mobile object-net systems. *Fundamenta Informaticae*, 54(2-3), 2003.
16. Michael Köhler and Heiko Rölke. Properties of Object Petri Nets. In J. Cortadella and W. Reisig, editors, *Application and Theory of Petri Nets*, volume 3099 of *Lecture Notes in Computer Science*, pages 278–297. Springer-Verlag, 2004.
17. Michael Köhler and Heiko Rölke. Reference and value semantics are equivalent for ordinary object petri nets. In G. Ciardo and P. Darondeau, editors, *Application and Theory of Petri Nets*, volume 3536 of *Lecture Notes in Computer Science*, pages 309–328. Springer-Verlag, 2005.
18. Michael Köhler-Bußmeier. Decidability results for elementary object systems. Report of the department of informatics, Universität Hamburg, Fachbereich Informatik, 2011.
19. Michael Köhler-Bußmeier and Frank Heitmann. Safeness for object nets. *Fundamenta Informaticae*, 101(1-2):29–43, 2010.
20. Michael Köhler-Bußmeier and Frank Heitmann. Liveness of safe object nets. *Fundamenta Informaticae*, 112(1):73–87, 2011.
21. Michael Köhler-Bußmeier and Frank Heitmann. Conservative elementary object systems. *Fundamenta Informaticae*, 120(3–4):325–339, 2012.
22. Michael Köhler-Bußmeier and Frank Heitmann. Complexity results for elementary hornets. In José-Manuel Colom and Jörg Desel, editors, *Application and Theory of Petri Nets and Concurrency. 34th International Conference, PETRI NETS 2013. Milan, Italy, June 24-28, 2013. Proceedings*, volume 7927 of *Lecture Notes in Computer Science*, pages 150–169. Springer-Verlag, 2013.
23. Charles Lakos. A Petri net view of mobility. In *Formal Techniques for Networked and Distributed Systems (FORTE 2005)*, volume 3731 of *Lecture Notes in Computer Science*, pages 174–188. Springer-Verlag, 2005.
24. Irina A. Lomazova. Nested Petri nets – a formalism for specification of multi-agent distributed systems. *Fundamenta Informaticae*, 43(1-4):195–214, 2000.
25. Irina A. Lomazova, Kees M. van Hee, Olivia Oanea, Alexander Serebrenik, Natalia Sidorova, and Marc Voorhoeve. Nested nets for adaptive systems. In *Petri Nets and Other Models of Concurrency - ICATPN 2006. 27th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency, Turku, Finland, June 26-30, 2006. Proceedings*, volume 4024 of *Lecture Notes in Computer Science*, pages 241–260. Springer-Verlag, 2006.
26. Wolfgang Reisig and Grzegorz Rozenberg, editors. *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.
27. Rüdiger Valk. Modelling concurrency by task/flow EN systems. In *3rd Workshop on Concurrency and Compositionality*, number 191 in GMD-Studien, St. Augustin, Bonn, 1991. Gesellschaft für Mathematik und Datenverarbeitung.
28. Rüdiger Valk. Object Petri nets: Using the nets-within-nets paradigm. In Jörg Desel, Wolfgang Reisig, and Grzegorz Rozenberg, editors, *Advanced Course on Petri Nets 2003*, volume 3098 of *Lecture Notes in Computer Science*, pages 819–848. Springer-Verlag, 2003.
29. Moshe Vardi. An automata-theoretic approach to linear temporal logic. In F. Moller and G. Birtwistle, editors, *Logics for Concurrency: Structure versus Automata*, volume 1043 of *Lecture Notes in Computer Science*, pages 238–266. Springer-Verlag, 1996.