1 Introduction: Nets within Nets

In this presentation the decidability of the reachability and the boundedness problem of “object net systems” [Val98] is studied. Object net systems formalise the aspect that tokens of a Petri net can be nets again. Taking this point of view it is possible to model e.g. mobility very naturally: A mobile entity is described by a Petri net which is a token of another Petri net describing the whole surrounding system (cf. [KMR03]).

To give an example we consider a situation in Figure 1 where we have a two-level hierarchy. The net token is then called the “object net”, the surrounding net is called the “system net”. An intuitive interpretation of this model is a scenario, where each object net models a mobile agent and the system-net models the agent system. In this example the agent \( a \) wants to travel from New York to Europe and will not come back until he has visited at least three European cities. Initially the agent is a net token on the place \texttt{New York} – indicated by the \texttt{ZOOM}. The places in the system-net describe the cities of the scenario, the transitions movements between them.

Object and system-nets synchronise via channels. The channels are denoted as transition inscriptions of the form \texttt{a:visit\_city} in the system-net and \texttt{:visit\_city} in the object net. The asymmetry is due to the fact, that the object net (the agent \( a \)) is known in the system-net but not vice versa. Transitions with corresponding channels must fire synchronously. Transitions without an inscription can fire autonomously and concurrently to other enabled transitions.

Let us look at an example process of the system in Figure 1: The first firing step is a synchronous firing of the transitions \texttt{fly to Europe} and the transition \( v \) (“visit”) wrt. the channel \texttt{visit\_city}. As a result one black token is generated on the place \texttt{cnt} (the counter) inside the object net and the whole object net token is located in \texttt{London}. Then the agent moves to \texttt{Madrid}, generating a second token on the place \texttt{cnt}. This time

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{example_diagram}
\caption{A mobile agent as a net token}
\end{figure}
the agent cannot fly back to New York, since he has only two tokens on the place cnt, while three are needed to activate the channel leave_Europe. So, the transitions fly back and I ("leave") are not activated. The sole possibility is to travel to Berlin, to Rome, and afterwards to Madrid again. Now, the agent can fly back, since he has five tokens on the place cnt.

In this example the net token has only been transported, since each transition in the system-net has exactly one input and one output place. For transitions with multiple outgoing arcs multiple copies of the net token are created where the marking of the original net token is distributed to the copies.

Related models of this "nets within nets" model are Nested Petri Nets [Lom00], Linear Logic Object Petri Nets [Far99], Reference Nets [Kum02] and Mobile Object Petri Net Systems [KR03]. Furthermore, there is a close connection to mobility calculi, like the ambient calculus [CGG99].

2 Formal Definition of Object Petri Nets

The simplest model of "nets within nets" are "Unary Elementary Object Systems" (Eos) defined by [Val98]. They are called "elementary" since the nesting hierarchy is limited to depth of two. On the top level there is one so called system-net which has one instance (but different copies) of the object net – therefore the term unary. Here, we give a generalised version of Eos, since Place/Transition nets (short: P/T-nets) are considered (instead of EN systems as in [Val98]).

In the following multisets over \( P \) are denoted as \( MS(P) \), multiset addition is denoted by \( M_1 + M_2 \), and the empty multiset is \( 0 \). A P/T-net is denoted by \( N = (P,T,\text{pre},\text{post},M_0) \) where \( P \) is a finite set of places; \( T \) a finite set of transitions disjoint from \( P \); \( \text{pre}, \text{post} : T \rightarrow MS(P) \) are the pre- and post-condition functions; and \( M_0 \in MS(P) \) is the initial marking of \( N \). For Eos we use the notation \( x \) for elements of the whole object net system, \( \bar{x} \) for elements of the object net, and \( x \) for elements of the system net.

**Definition 1.** An unary elementary object system is a tuple \( OS = (N,\bar{N},\rho) \), such that:

- The system net \( N = (P,T,\text{pre},\text{post},M_0) \) is a P/T-net with \( |M_0| = 1 \) and \( |T|, |\rho| > 0 \) for all \( t \in T \).
- The object net \( \bar{N} = (\bar{P},\bar{T},\overline{\text{pre}},\overline{\text{post}},\bar{M}_0) \) is a P/T-net with \( (P \cup T) \cap (\bar{P} \cup \bar{T}) = \emptyset \).
- \( \rho \subseteq T \times \bar{T} \) is the interaction relation.

A marking of \( OS \) is a multiset \( M \in M := MS(P \times MS(\bar{P})) \). The initial marking of an Eos \( OS \) is \( M_0 = (p,\bar{M}_0) \) for \( M_0 = p \).

By using projections on the first or last component of an Eos marking \( M \), it is possible to compare object system markings. The projection \( \Pi^1(M) \) on the first component abstracts away the substructure of a net token, while the projection \( \Pi^2(M) \) on the second component can be used as the abstract marking of the net tokens without considering their local distribution within the system net.

The Eos firing rule is defined for three cases: system-autonomous firing, object-autonomous firing, and synchronised firing. The set of synchronising transitions in the system net is \( T_\rho := \{ \rho \bar{T} \} \). The set of synchronisation free transitions is \( T_\rho := T \setminus T_\rho \). Analogously in the object net: \( \bar{T}_\rho := (T \rho) \) and \( \bar{T}_\rho := \bar{T} \setminus \bar{T}_\rho \). A transition \( \tau \) can only fire autonomously if there exists no synchronisation partner in \( \rho \), i.e. if \( \tau \in (T \rho \cup \bar{T}_\rho) \). Otherwise if \( \tau \in \rho \), only synchronous firing is possible.

The autonomous firing of a system net transition \( t \) removes net tokens in the pre-conditions together with their individual internal markings. Since the markings of Eos are higher-order multisets, we have to consider terms \( \text{PRE} \in M \) that correspond to the preset of \( t \) in their first component: \( \Pi^1(\text{PRE}) = \text{pre}(t) \). In turn, a multiset \( \text{POST} \in M \) is produced, that corresponds with the post-set of \( t \) in its first component. Thus, the
successor marking is $M' = M - \text{PRE} + \text{POST}$, in analogy to the successor marking $M' = M - \text{pre}(t) + \text{post}(t)$ of P/T-nets. The firing of $t$ must also obey the object marking distribution condition $\Pi^2(\text{PRE}) = \Pi^2(\text{POST})$, ensuring that the sum of markings in the copies of a net token is preserved.

An object-net transition $t$ is enabled autonomously if in $M$ there is an addend $(p, \hat{M})$ in the sum and $\hat{M}$ enables $t$. For synchronous firing, a combination of both is required.

**Definition 2.** Let $M \in \mathcal{M}$ be a marking of an Eos $OS = (N, \hat{N}, \rho)$:

1. System-autonomous firing: For $t \in T_p$ define $M \xrightarrow{t} M'$ iff $\exists \text{PRE}, \text{POST} \in \mathcal{M}$:
   $$\Pi^1(\text{PRE}) = \text{pre}(t) \text{ and } \Pi^1(\text{POST}) = \text{post}(t) \text{ and } \Pi^2(\text{PRE}) = \Pi^2(\text{POST}) \text{ and } M' = M - \text{PRE} + \text{POST}.$$  

2. Synchronous firing: Let $(t, \hat{t}) \in \rho$ be a pair of transitions. We define: $M \xrightarrow{(t, \hat{t})} M' \text{ iff } \exists \text{PRE}, \text{POST} \in \mathcal{M}$:
   $$\Pi^1(\text{PRE}) = \text{pre}(t) \text{ and } \Pi^1(\text{POST}) = \text{post}(t) \text{ and } \Pi^2(\text{PRE}) = \hat{\text{pre}}(\hat{t}) \text{ and } \Pi^2(\text{POST}) = \hat{\text{post}}(\hat{t}) \text{ and } M' = M - \text{PRE} + \text{POST}.$$  

3. Object-autonomous firing: For $\hat{t} \in \hat{T}_p$ define $M \xrightarrow{\hat{t}} M'$ iff $\exists \text{PRE}, \text{POST} \in \mathcal{M}$:
   $$\exists p \in P : \Pi^1(\text{PRE}) = \Pi^1(\text{POST}) = p \text{ and } \Pi^2(\text{PRE}) = \hat{\text{pre}}(\hat{t}) \text{ and } \Pi^2(\text{POST}) = \hat{\text{post}}(\hat{t}) \text{ and } M' = M - \text{PRE} + \text{POST}.$$  

By using a degenerated Eos, which has neither places nor transitions for the object net, it can easily be shown that Eos are a canonical extension of P/T-nets with respect to firing sequences.

### 3 Decidability Issues

In the following some decidability questions related to Petri nets are studied for Eos, especially the reachability and the boundedness problem is considered.

It is a well known fact that P/T-nets are not Turing powerful, since the reachability problem, i.e. the question whether a given marking is reachable from the initial one, is decidable [May81], whereas the halting problem for Turing machines is not. The reachability problem is undecidable for Eos, since Eos can simulate nets with transfer arcs. Transfer arcs (defined in [Cia94]) are used to transfer all tokens of one place to another place. Note, that transfer arcs are related to reset arcs which removes all tokens from one place.

**Proposition 1.** The reachability problem for Eos is undecidable.

**Proof.** This can be seen by giving a simulation of Petri nets with transfer arcs using Eos and the fact that the reachability problem for Petri nets with at least two transfer arcs is undecidable (cf. Theorem 11 in [DFS98]). In the simulation the system net describes the original Petri net $N$ and the object net on $N$ acts as a container for the tokens. Whenever a token is added to a place $p$ in $N$ the system net removes the object net on $p$, adds one token inside using the synchronisation mechanism, and puts the object net back on $p$. Whenever the whole marking of the place $p$ is transferred to $p'$ in $N$ the object net token $\hat{N}$ is moved as a whole from $p$ to $p'$ together with its marking $\hat{M}$. 

On the other hand Eos are computational weaker than Petri nets with reset arcs (i.e. arcs that removes all tokens from a place), since it can be shown that boundedness is a decidable property of Eos, while boundedness is undecidable for Nets with at least two reset arcs [DFS98, Theorem 8]. The boundedness of an Eos can be decided using the coverability graph construction of [KM69] which has been generalised in [FS01]. To construct the coverability graph we are looking for a marking sequence $m_0 \rightarrow m \rightarrow m'$ where $m < m'$ holds, i.e. $m'$ covers $m$ on the places from a non-empty set $U \subseteq P$: $m(p) < m'(p)$ for all $p \in U$ and $m(p) = m'(p)$ for all $p \in P \setminus U$. Since Petri nets enjoy the property of monotonicity (i.e. if $m_1 < m_2$ and $m_1 \rightarrow m'_1$ then there exists
a sequence \( m_2 \prec m'_2 \) with \( m'_1 \prec m'_2 \) this sequence can be repeated infinitely often showing that any \( n \in \mathbb{N} \) can be covered by the marking \( m(p) \) of places \( p \in U \), i.e. the place \( p \) is unbounded.

To apply this technique for Eos the partial order \( \leq \) on multisets can be extended to a partial order on nested multisets. Let \( M, M' \in \mathcal{M} \) be two nested multisets. Define \( M \preceq M' \) iff there exists a total and injective mapping \( f \) from \( M = \sum_{i=1}^{n} (p_i, M_i) \) to \( M' = \sum_{j=1}^{n} (p'_j, M'_j) \) with \( f((p_i, M_i)) = (p'_j, M'_j) \) implying \( p_i = p'_j \) and \( M_i \leq M'_j \) (Note, that \( M \preceq M' \) implies \( II^1(M) \preceq II^1(M') \)).

**Proposition 2.** The boundedness problem for Eos is decidable.

**Proof.** A partial order \( \leq \) has the property of strict transitive compatibility, iff \( m_1 \prec m_2 \) and \( m_1 \rightarrow m'_1 \) then there exists a sequence \( m_2 \prec m'_2 \) with \( m'_1 \prec m'_2 \). Generalising the result of [KM69] it is shown in [FS01], that the boundedness problem is decidable iff \( \leq \) is a decidable, strict partial order and the set of successor markings is decidable. Obviously \( \leq \) is decidable and strict transitive compatible and the set of successors is effectively constructible. \( \square \)

## 4 Conclusion

In this contribution we presented decidability results for “nets within nets” formalisms, i.e. formalisms that allow for Petri nets as tokens. The analysis of decidable properties has shown that Eos are more than just a convenient representation of another – possibly larger – Petri net model – since the reachability problem is undecidable for Eos. So, Eos are a real extension with greater computational power. Nevertheless, interesting questions – like boundedness – remain decidable, making Eos weaker than Petri nets with reset arcs.

## References


