Reference and Value Semantics Are Equivalent for Ordinary Object Petri Nets

Michael Köhler and Heiko Rölke

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

Abstract. The concept of mobile agents imposes a great security risk for information systems. In this paper we propose object nets as a specification formalism for multi-agent systems. Since the general formalism is Turing-powerful not every analysis method that is common for Petri net can be applied. So, we define the subclass of "ordinary" object nets that allows for the application of standard P/T-net techniques, i.e. the computation of boundedness, liveness etc.

1 Introduction

Object Petri nets following the *nets-within-nets* paradigm are a very powerful formalism to describe dynamic multi-levelled systems, e.g. mobile agent systems. It is well known that severe security problems arise in the context of mobile agents (cf. [5]). So, the use of formal methods to overcome security problems is necessary. In [10] the authors showed how the formalism of nets-within-nets can be used to model mobility, especially in the case of mobile agents. Mobile agents are developed using our architecture Mulan [9]. The Mulan-framework offers intuitive modelling even of large agent systems.[1] What was missing is the possibility to profit from analysing tools. This paper undertakes an attempt to build a conceptual background for the transformation of object net systems to P/T-nets. Several requirements have to be met for this transformation to succeed. These requirements are the subject of the following sections.

The need for analysis of object nets is paired with the choice of the firing rule for object nets: There exist two fundamental semantics (i.e. firing rules) for object nets introduced in [20], called reference and value semantics. The difference of reference and value semantics is the concept of "location" for net-tokens which is explicit for value but not for reference semantics, since it is unclear which reference can be considered as the location of a net-token.

As shown in [10] the concept of mobility cannot be expressed adequately by reference semantics due to the possibility of side-effects. Instead value semantics has to be applied. As shown in [12] the concept of locality makes value semantics richer

[1] Student projects created agent systems containing more than 200 different reference nets resulting in tens of thousands of net instances at run-time.

G. Ciardo and P. Darondeau (Eds.): ICATPN 2005, LNCS 3536, pp. 309–328, 2005.

than reference semantics – for example the reachablity problem becomes undecidable while boundedness remains decidable. However, the reference semantics can be simulated by a (larger) P/T net, so analysis methods can be applied directly.

Value semantics is adequate from a modelling point of view while reference semantics is adequate from an analytical point of view. In this paper we focus on a restricted class of object nets, the so called *ordinary* object nets. For this class of object nets it can be shown, that value semantics is as expressible as reference semantics. This is shown by providing a direct embedding and simulation of one semantics using the other one.

In this paper we study semantical aspects of the *nets-within-nets* paradigm.[2] The paradigm that allows nets as tokens was introduced by Rüdiger Valk in [19] and extended to the formalism of elementary object net systems in [20, 21] which allows to model a two-levelled system. The formalism has been extended in [11, 12] to an arbitrary nesting structure. A similar approach which allows nested Petri net structures is presented in [15]. For Hypernets [1] net-tokens are restricted to synchronisations of state machines. Reference nets [13] are a specialised nets-within-nets formalism based on reference semantics. Due to the nested structure of object nets the formalism is closely related to mobility calculi like the ambient calculus [4] or to formalism combining mobility calculi and Petri nets like [3].

The paper is structured as follows: Section 2 gives the formal definition of object-net systems. Firing is defined both for value and for reference semantics. Section 3 analyses *located* markings, i.e. markings that describe a unique relation of tokens and their locations: for each net-token there exists exactly one place containing it. Section 4 defines the subclass of *ordinary* object-net systems. It is shown that for this class of object nets all reachable markings are located. Using this result it is shown that reference and value semantics can simulate each other directly. In Section 5 we analyse the processes of ordinary object-net systems. It turns out that the firing relation, the mapping from a process to the original object net and the mappings from reference and value semantics are compatible with each other which results in a three-dimensional cube structure of embeddings. After having cleared the conceptual background we present a case study in Section 6 and present some analysis results. Finally, we give an outlook and conclusion.

2 Object-Net Systems

We define a generalised model of object-net systems, which drops the restriction of [20] to exactly two levels of nesting: Object-Net Systems (OS) are defined to give a precise definition of nets-within-nets using nested multi-set rewriting specifications.

[2] As [18] mentioned in the outlook it is a quite natural extension of algebraic Petri nets [16] to allow tokens to be *active* which is impossible for algebraic Petri nets. The canonic way for this extension is to consider nets as active tokens. These tokens are called net-tokens.

2.1 Informal Introduction to Object Nets

There exist two fundamental semantics for object nets introduced in [20], called reference and value semantics. The intuitive meaning of both semantics can be explained using the example Os given in Fig. 1. The example is known as the "α-centauri" example. The name is due to the interpretation that the net token describes a log which is copied and one copy remains on earth while the other one is sent to α-centauri. It then seems somehow counter-intuitive that for reference semantics the state change on α-centauri (the upper branch of the system net) becomes visible immediately on earth (the lower branch).

The arrow from the token on place s_1 expresses that the inner structure of the token is itself a net. The different levels in the object-net system are connected by channels. Transitions inscribed by corresponding channel expressions like on:ch() and :ch() must be fired synchronously. If there is more than one possible partner the choice is non-deterministic. In the Figure each transition pair (t_2, t_{11}) and (t_3, t_{12}) must fire synchronously.

For reference semantics (cf. Fig. 2) the place s_1 initially contains a reference to the object-net: $\mathbf{M} = s_1 + s_{11}$. Firing of t_1 duplicates this reference onto s_2 and s_3 resulting in $\mathbf{M}_1 = s_2 + s_3 + s_{11}$. This marking enables the transition pair (t_2, t_{11}) but not (t_3, t_{12}). The resulting marking is $\mathbf{M}_2 = s_4 + s_3 + s_{12}$. Since the effect in the object-net is visible in the whole system, the pair (t_3, t_{12}) is now enabled. Firing leads to $\mathbf{M}_3 = s_4 + s_5 + s_{13}$.

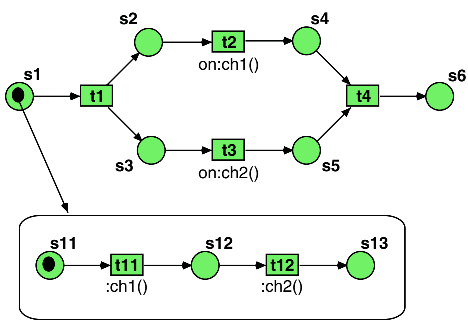

Fig. 1. An Os: The α-centauri example

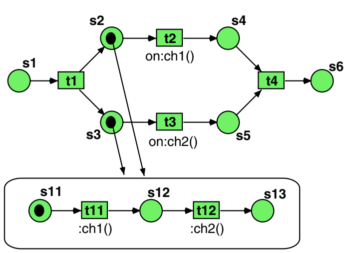

Fig. 2. Firing of transition t_1 w.r.t. reference semantics

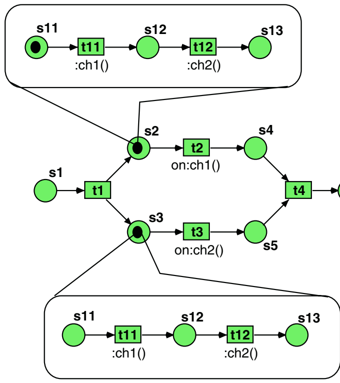

Fig. 3. Firing of transition t_1 w.r.t. value semantics

For value semantics (cf. Fig. 3) we have the *nested* multiset $\mathbf{M} = s_1[s_{11}]$ as the initial marking which corresponds to the initial marking $s_1 + s_{11}$ w.r.t. reference semantics. Firing of t_1 distributes the marking of the net-token. A possible distribution is the marking $\mathbf{M}_1 = s_2[s_{11}] + s_3[\mathbf{0}]$ (where $\mathbf{0}$ denotes the empty multiset) – corresponding to $s_2 + s_3 + s_{11}$ for reference semantics. This marking enables the transition pair (t_2, t_{11}) but not the pair (t_3, t_{12}). The resulting marking is $\mathbf{M}_2 = s_4[s_{12}] + s_3[\mathbf{0}]$. Since the effect in the object net is only local the pair (t_3, t_{12}) is not enabled. So $w = t_1(t_2, t_{11})(t_3, t_{12})$ is a possible firing sequence for reference but not for value semantics.

2.2 Petri Net Notations

A *P/T net structure* is a tuple $N = (P, T, W)$, such that: P is a finite set of places, T is a finite set of transitions, with $P \cap T = \emptyset$, and $W : ((P \times T) \cup (T \times P)) \to \mathbb{N}$ is the arc-weight function. A *marked P/T-net* $N = (P, T, W, \mathbf{M}_0)$ is an P/T net structure (P, T, W) together with an initial marking $\mathbf{M}_0 \in MS(P)$. The term *P/T net* is used both for the unmarked and the marked case. The flow relation is $F := \{(x, y) \mid W(x, y) > 0\}$. Given a net $P(N)$ denotes its places, $T(N)$ its transitions etc. N is called *ordinary* iff $W(x, y) \leq 1$ for all (x, y). For an ordinary P/T net the mapping W coincides with the characteristic function of the flow relation $\chi_F : ((P \times T) \cup (T \times P)) \to \{0, 1\}$. In the case of ordinary nets the relation F is also used to denote the arc weight W and vice versa.

A transition $t \in T$ of a P/T net $N = (P, T, W, \mathbf{M}_0)$ is *enabled* in marking $\mathbf{M}$ iff $\forall p \in P : \mathbf{M}(p) \geq W(p, t)$ holds. The successor marking when firing t is $\mathbf{M}'(p) = \mathbf{M}(p) - W(p, t) + W(t, p)$. The enablement of t in marking $\mathbf{M}$ is denoted by $\mathbf{M} \xrightarrow{t}$. Firing of t is denoted by $\mathbf{M} \xrightarrow{t} \mathbf{M}'$.

A *P/T net* is equivalently characterised as $N = (P, T, \mathbf{pre}, \mathbf{post}, \mathbf{M}_0)$ where the multi-set mappings $\mathbf{pre}, \mathbf{post} : T \to MS(P)$ are $\mathbf{pre}(t)(p) := W(p, t)$ and $\mathbf{post}(t)(p) := W(t, p)$. For notations cf. the appendix.

2.3 Nets as Tokens

An Object-Net System (Os) $OS = (\mathcal{N}, d, \Theta, \mathbf{M}_0)$ consists of a finite set $\mathcal{N}$ of pairwise disjoint P/T nets which includes the black token net $N_\bullet$ and the system-net N_{sn}. The black token net $N_\bullet$ is defined as the object-net with no places and no transitions: $P(N_\bullet) = T(N_\bullet) = \emptyset$. Let $P(OS)$ be the union of all components: $P(OS) := \bigcup_{N\in\mathcal{N}} P(N)$. Analogously for $T(OS)$, $F(OS)$, and $W(OS)$.

Markings are nested multi-sets. Tokens are described as pairs of the marked place p and the marking of $\mathbf{M}$ its net-token which is denoted as $p[\mathbf{M}]$ to emphasize the nesting. The place typing $d : P(OS) \to \mathcal{N}$ is used to define which net-tokens are allowed on a place. A black token is the special net-token $p[\mathbf{0}]$ which can be identified with p. The basic tokens are black tokens, higher-order tokens are generated using net-tokens.

$$\begin{aligned}\mathcal{P}_0(N) &:= \{p \mid p \in P(N) \wedge d(p) = N_\bullet\}\\ \mathcal{P}_{n+1}(N) &:= \{p[\mathbf{M}] \mid p \in P(N) \wedge \mathbf{M} \in MS(\mathcal{P}_n(d(p)))\}\end{aligned} \quad (1)$$

Define $\mathcal{P}(N) := \bigcup_{i=0}^{\infty} \mathcal{P}_n(N)$. Each mapping f defined on P can be extended to a mapping $f^\sharp$ on nested markings $\mathcal{P}$ by setting $f^\sharp(p[\mathbf{M}]) = f(p)[f^\sharp(\mathbf{M})]$. In the following $f^\sharp$ is also denoted as f.

A transition $t \in T(OS)$ may be synchronised with transitions of the net-tokens. The resulting synchronisations are nested transitions, i.e. trees:[3]

$$\begin{aligned}\mathcal{T}_0(N) &:= \{id\}\\ \mathcal{T}_{k+1}(N) &:= \{t[\theta_{N_1}, \ldots, \theta_{N_n}] \mid t \in T(N) \wedge \theta_{N_i} \in \textstyle\bigcup_{j=0}^{k} \mathcal{T}_j(N_i)\}\end{aligned} \quad (2)$$

Analogously to markings we identify the minimal synchronisation tree $t[\boldsymbol{id}] := t[id, \ldots, id]$ with the transition t itself. Here id is a "pseudo" transition with $\mathbf{pre}(id) = \mathbf{post}(id) = \mathbf{0}$ (see below). So, every node in the tree has the same degree of branching.

Define $\mathcal{T}(N) := \bigcup_{i=0}^{\infty} \mathcal{T}_n(N)$ and $\mathcal{T}(OS) := \mathcal{T}(\mathcal{N}) := \bigcup_{N\in\mathcal{N}} \mathcal{T}(N)$. A synchronisation structure $\Theta(OS)$ consits of a finite subset of $\mathcal{T}(\mathcal{N})$.

The nesting structure of markings is removed by $\mathrm{fl} : \mathcal{M}_v \to \mathcal{M}_r$ with $\mathcal{M}_r := MS(P(OS))$ where $\mathrm{fl}(p[\mathbf{M}])) := p + \mathrm{fl}(\mathbf{M})$.

The nesting structure of synchronisations is removed by $\mathrm{fl} : \mathcal{T}(OS) \to T(OS)$ where $\mathrm{fl}(t[\theta_1, \ldots, \theta_n]) := t + \mathrm{fl}(\theta_1) + \cdots + \mathrm{fl}(\theta_n)$ and $\mathrm{fl}(id) = \mathbf{0}$.

Let $\Theta \subseteq \mathcal{T}(N)$ be a set of synchronisations. To avoid cycles the set of synchronisations has to contain each transitions eaxactly once: $\mathrm{fl}(\Theta) = T(OS)$ (Note, that minimal synchronisation trees $t[\boldsymbol{id}]$ are allowed).

A marking is a multi-set of system-net tokens: $\mathbf{M} \in \mathcal{M}_v := MS(\mathcal{P}(N_{sn}))$.

Definition 1. *An* Object-Net System *is the tuple* $OS = (\mathcal{N}, d, \Theta, \mathbf{M}_0)$, *where*

- $\mathcal{N} = \{N_1, \ldots N_n\}$ *is a set of pairwise disjoint P/T-nets* $N_i = (P_i, T_i, W_i)$ *including the black token net* $N_\bullet$ *and the system-net* N_{sn}.
- $d : P \to \mathcal{N}$ *is the place typing.*

[3] In the graphical representation these trees are formalised by channel inscriptions.

– *$\Theta \subseteq \mathcal{T}(\mathcal{N})$ is a finite set of synchronisations with* $\text{fl}(\Theta) = T(OS)$.
– *The initial marking is* $\mathbf{M}_0 \in MS(\mathcal{P}(N_{sn}))$.

We generalise the notion of pre- and post-set for object nets by defining

$$
\begin{aligned}
{}^{(N)}t &:= {}^\bullet t \cap d^{-1}(N) = \{p \in {}^\bullet t \mid d(p) = N\} \\
t^{(N)} &:= t^\bullet \cap d^{-1}(N) = \{p \in t^\bullet \mid d(p) = N\}
\end{aligned}
$$

Definition 2. *For the synchronisation tree $t[\boldsymbol{\theta}] \in \Theta$ where $\boldsymbol{\theta} = (\theta_{N_1}, \ldots, \theta_{N_n})$ the firing rule is generated inductively from the firing rule of the subtrees θ_N (where the multi-set variables $\mathbf{X}_{p,t,i}$, $\mathbf{X}'_{p,t,i}$ describe the tokens that are transported and $\mathbf{Y}_{p,t,i}$, $\mathbf{Y}'_{p,t,i}$ describe the tokens that are used for synchronisation):*

$$
\begin{aligned}
&\sum_{p \in {}^\bullet t} \sum_{i=1}^{W(p,t)} p[\mathbf{X}_{p,t,i} + \mathbf{Y}_{p,t,i}] \xrightarrow{t[\boldsymbol{\theta}]} \sum_{p \in t^\bullet} \sum_{j=1}^{W(t,p)} p[\mathbf{X}'_{p,t,j} + \mathbf{Y}'_{p,t,j}] \\
&\text{if } \forall N \in \mathcal{N} : \textstyle\sum_{p \in {}^{(N)}t} \sum_{i=1}^{W(p,t)} \mathbf{X}_{p,t,i} = \sum_{p \in t^{(N)}} \sum_{j=1}^{W(t,p)} \mathbf{X}'_{p,t,j} \\
&\qquad \wedge \textstyle\sum_{p \in {}^{(N)}t} \sum_{i=1}^{W(p,t)} \mathbf{Y}_{p,t,i} = \mathbf{pre}(\theta_N) \\
&\qquad \wedge \textstyle\sum_{p \in t^{(N)}} \sum_{j=1}^{W(t,p)} \mathbf{Y}'_{p,t,j} = \mathbf{post}(\theta_N)
\end{aligned}
$$

For the minimal synchronisations $t[\boldsymbol{id}]$ this implies $\mathbf{Y}_{p,t,i} = \mathbf{Y}'_{p,t,i} = \mathbf{0}$.

The marking $\mathbf{M}$ can be fired by $\theta \in \Theta$ to $\mathbf{M}'$ iff $\mathbf{pre}(\theta)$ is a subterm of $\mathbf{M}$ and $\mathbf{M}'$ is obtained from $\mathbf{M}$ by substituting $\mathbf{pre}(\theta)$ with $\mathbf{post}(\theta)$. Firing is denoted by $\mathbf{M} \xrightarrow{\theta} \mathbf{M}'$.

This firing relation formalises value semantics. For reference semantics the "flat" version of *OS* is needed (for more details cf. [12]).

Definition 3. *The* underlying *P/T net of $OS = (\mathcal{N}, d, \Theta, \mathbf{M}_0)$ is defined as:*

$$\text{fl}(OS) := (P(OS), \Theta, \mathbf{pre}^{\text{fl}}, \mathbf{post}^{\text{fl}}, \text{fl}(\mathbf{M}_0))$$

where $\mathbf{pre}^{\text{fl}}(\theta) := \mathbf{pre}(\text{fl}(\theta))$ and $\mathbf{post}^{\text{fl}}(\theta) := \mathbf{post}(\text{fl}(\theta))$.

Reference semantics is obtained by forgetting the nesting structure:

Theorem 1. *For an* Os *OS the mapping* fl *provides a direct embedding of the value semantics, i.e. every firing w.r.t. value semantics is possible w.r.t. reference semantics:*

$$
\begin{array}{ccc}
\mathbf{M}_v & \xrightarrow[OS]{\theta} & \mathbf{M}'_v \\
\text{fl}\downarrow & & \downarrow\text{fl} \\
\text{fl}(\mathbf{M}_v) & \xrightarrow[\text{fl}(OS)]{\theta} & \text{fl}(\mathbf{M}'_v)
\end{array}
$$

Proof. Since $\mathbf{pre}^{\text{fl}}(\theta) := \mathbf{pre}(\text{fl}(\theta))$ and $\mathbf{post}^{\text{fl}}(\theta) := \mathbf{post}(\text{fl}(\theta))$ it is sufficient to show $\text{fl}(\mathbf{pre}(\theta)) = \mathbf{pre}^{\text{fl}}(\theta)$ and $\text{fl}(\mathbf{post}(\theta)) = \mathbf{post}(\text{fl}(\theta))$. This is shown by induction over the synchronisation tree.

– For the minimal synchronisation tree we have (since $\sum_{p\in{}^\bullet t} \mathbf{X}_{p,t,i} = \sum_{p\in t^\bullet} (\mathbf{X}'_{p,t,j})$):

$$\begin{aligned}\mathrm{fl}(\mathbf{pre}(t[\boldsymbol{id}])) &= \mathrm{fl}(\textstyle\sum_{p\in{}^\bullet t}\sum_{i=1}^{W(p,t))} p[\mathbf{X}_{p,t,i}])\\ &= \textstyle\sum_{p\in{}^\bullet t}\sum_{i=1}^{W(p,t))} \mathrm{fl}(p[\mathbf{X}_{p,t,i}])\\ &= \textstyle\sum_{p\in{}^\bullet t}\sum_{i=1}^{W(p,t))} p + \mathrm{fl}(\mathbf{X}_{p,t,i}) = \mathbf{pre}(\mathrm{fl}(t[\boldsymbol{id}]))\end{aligned}$$

$$\begin{aligned}\mathbf{post}(t[\boldsymbol{id}]) &= \textstyle\sum_{p\in t^\bullet}\sum_{j=1}^{W(t,p))} p + \mathrm{fl}(\mathbf{X}'_{p,t,j})\\ &= \textstyle\sum_{p\in t^\bullet}\sum_{j=1}^{W(t,p))} \mathrm{fl}(p[\mathbf{X}'_{p,t,j}])\\ &= \mathrm{fl}(\textstyle\sum_{p\in t^\bullet}\sum_{j=1}^{W(t,p))} p[\mathbf{X}'_{p,t,j}]) = \mathrm{fl}(\mathbf{post}(t))\end{aligned}$$

– By assumption we have $\mathrm{fl}(\sum_{p\in{}^{(N)}t}\sum_{i=1}^{W(p,t))} \mathbf{Y}_{p,t,i}) = \mathbf{pre}(\mathrm{fl}(\theta_N))$ and also $\mathrm{fl}(\sum_{p\in t^{(N)}}\sum_{j=1}^{W(t,p))} \mathbf{Y}'_{p,t,j}) = \mathbf{post}(\mathrm{fl}(\theta_N))$. Induction on $\theta = t[\boldsymbol{\theta}]$:

$$\begin{aligned}\mathrm{fl}(\mathbf{pre}(t[\boldsymbol{\theta}])) &= \mathrm{fl}(\textstyle\sum_{p\in{}^\bullet t}\sum_{i=1}^{W(p,t))} p[\mathbf{X}_{p,t,i} + \mathbf{Y}_{p,t,i}])\\ &= \textstyle\sum_{p\in{}^\bullet t}\sum_{i=1}^{W(p,t))} \mathrm{fl}(p[\mathbf{X}_{p,t,i} + \mathbf{Y}_{p,t,i}])\\ &= \textstyle\sum_{p\in{}^\bullet t}\sum_{i=1}^{W(p,t))} p + \mathrm{fl}(\mathbf{X}_{p,t,i}) + \mathrm{fl}(\mathbf{Y}_{p,t,i})\\ &= \textstyle\sum_{p\in{}^\bullet t}\sum_{i=1}^{W(p,t))} p + \mathrm{fl}(\mathbf{X}_{p,t,i}) + \sum_{N\in\mathcal{N}} \mathbf{pre}(\mathrm{fl}(\theta_N))\\ &= \mathbf{pre}(\mathrm{fl}(t[\boldsymbol{\theta}]))\end{aligned}$$

$$\begin{aligned}\mathrm{fl}(\mathbf{post}(t[\boldsymbol{\theta}])) &= \textstyle\sum_{p\in t^\bullet}\sum_{j=1}^{W(t,p))} p + \mathrm{fl}(\mathbf{X}'_{p,t,j}) + \sum_{N\in\mathcal{N}} \mathbf{post}(\mathrm{fl}(\theta_N))\\ &= \textstyle\sum_{p\in t^\bullet}\sum_{j=1}^{W(t,p))} p + \mathrm{fl}(\mathbf{X}'_{p,t,j}) + \mathrm{fl}(\mathbf{Y}'_{p,t,j})\\ &= \textstyle\sum_{p\in t^\bullet}\sum_{j=1}^{W(t,p))} \mathrm{fl}(p[\mathbf{X}'_{p,t,j} + \mathbf{Y}'_{p,t,j}])\\ &= \mathrm{fl}(\textstyle\sum_{p\in t^\bullet}\sum_{j=1}^{W(t,p))} p[\mathbf{X}'_{p,t,j} + \mathbf{Y}'_{p,t,j}])\\ &= \mathrm{fl}(\mathbf{post}(t[\boldsymbol{\theta}]))\end{aligned}$$

This proves the embedding. □

The converse (every firing w.r.t. reference semantics is possible w.r.t. value semantics) of Theorem 1, however, does not hold in the general case as seen for the α-centauri example.

3 Located Markings

In the following we consider markings where the localisation of tokens coincide with the type structure induced by the mapping d.

Definition 4. *A marking* $\mathbf{M}_r \in \mathcal{M}_r$ *is* located *iff for each net* N *there exists exactly one place containing* N *and no place contains the net* N_{sn}*:*

$$\forall N \in \mathcal{N} \setminus \{N_{sn}, N_\bullet\} : \sum_{p\in d^{-1}(N)} |\mathbf{M}_r(p)| = 1 \wedge \sum_{p\in d^{-1}(N_{sn})} |\mathbf{M}_r(p)| = 0$$

A marking $\mathbf{M}_v \in \mathcal{M}_v$ *is located iff* $\mathrm{fl}(\mathbf{M}_v)$ *is located.*

Therefore for all $N \in \mathcal{N} \setminus \{N_{sn}, N_\bullet\}$ the uniquely defined place for which $|\mathbf{M}(p)| = 1$ holds is denoted by $p(N)$.

Note, that if $\mathbf{M}_v$ is located, then there cannot be any recursive nesting, since otherwise there is more than one place containing a net token of type N. Thus, the location of each token is uniquely determined.

Definition 5. *The* localisation $\mathrm{lc}(\mathbf{M})$ *is defined recursively starting with the system net:* $\mathrm{lc}(\mathbf{M}) := \mathrm{lc}_M(\mathbf{M}_{|P(N_{sn})})$ *where*

$$\mathrm{lc}_M(p) := \begin{cases} p[\mathrm{lc}_{\mathbf{M}}(\mathbf{M}_{|P(N_{d(p)})})], & \text{if } d(p) \neq N_\bullet \\ p, & \text{otherwise} \end{cases}$$

For events $t \in T$ *we define* $\mathrm{lc}(t) = t$.

Theorem 2. *For located markings the mapping* lc *is inverse to* fl.

1. *If* $\mathbf{M}_r \in \mathcal{M}_r$ *is located, then we have* $\mathrm{fl}(\mathrm{lc}(\mathbf{M}_r)) = \mathbf{M}_r$.
2. *If* $\mathbf{M}_v \in \mathcal{M}_v$ *is located, then we have* $\mathrm{lc}(\mathrm{fl}(\mathbf{M}_v)) = \mathbf{M}_v$.

Proof. 1. Let $\mathbf{M}_r \in \mathcal{M}_r$. Define the relation $R_{\mathbf{M}_r} \subseteq (\mathcal{N} \setminus \{N_\bullet\})^2$ by

$$(N_1, N_2) \in R_{\mathbf{M}_r} \iff \exists p_1 \in M : p_1 \in P(N_1) \wedge d(p_1) = N_2$$

If $\mathbf{M}_r$ is located, then the place p_1 such that $d(p_1) = N_2$ with $N_2 \in \mathcal{N} \setminus \{N_{sn}, N_\bullet\}$ is uniquely defined, i.e. it is $p_1 = p(N_2)$. So, it follows that $R_{\mathbf{M}_r}$ describes a tree with the system net N_{sn} as the root node (since $\sum_{p \in d^{-1}(N_{sn})} |\mathbf{M}_r(p)| = 0$).
It is easy to see from the definition of lc that the marking is nested along the relation R_M, i.e. all markings of the nets $N' \in (N\, R_{\mathbf{M}_r}\text{-})$ on paths from the root of the tree are located by $\mathrm{lc}_M(\mathbf{M}_{|P(N)})$:

$$\mathrm{fl}(\mathrm{lc}_M(\mathbf{M}_{|P(N)})) = \sum_{N' \in (N\, R_{\mathbf{M}_r}\text{-})} \mathbf{M}_{|P(N')}$$

For the whole marking it follows that:

$$\begin{aligned} \mathrm{fl}(\mathrm{lc}(\mathbf{M})) &= \mathrm{lc}_M(\mathbf{M}_{|P(N_{sn})}) \\ &= \mathrm{fl}\left(\sum\nolimits_{N' \in (N_{sn}\, R_{\mathbf{M}_r}\text{-})} \mathbf{M}_{|P(N')}\right) \\ &= \mathbf{M}_{|P(N_{sn})} + \mathbf{M}_{|P(N_\bullet)} + \cdots + \mathbf{M}_{|P(N_n)} \\ &= M \end{aligned}$$

Note, that $\mathbf{M}_{P(N_\bullet)} = \mathbf{0}$ since $P(N_\bullet) = \emptyset$.

2. Let $\mathbf{M}_v \in \mathcal{M}_r$. Define the relation $R_{\mathbf{M}_v} \subseteq (\mathcal{N} \setminus \{N_\bullet\})^2$ by

$$(N_1, N_2) \in R_{\mathbf{M}_v} \iff \exists \text{ subterm } (p_1, \mathbf{M}_1) \text{ of } \mathbf{M}_v : p_1 \in P(N_1) \wedge d(p_1) = N_2$$

it is easy to see, that if $\mathbf{M}_v$ is located, then the relations $R_{\mathbf{M}_r}$ and $R_{\mathbf{M}_v}$ are equal, so lc just reconstructs $\mathbf{M}_v$, i.e. $\mathrm{lc}(\mathrm{fl}(\mathbf{M}_v)) = \mathbf{M}_v$.

□

4 Ordinary Object-Net Systems

As we have seen for the α-centauri example, the converse of Theorem 1 does not hold in general. It will be shown, that for the case of so called *ordinary* ONS the opposite direction can also be proved.

Definition 6. *Let OS be an* Os*. A transitions t is* simple *iff*

$$\forall N \in \mathcal{N} \setminus \{N_\bullet\} : |^{(N)}t| = |t^{(N)}| \leq 1$$

OS is ordinary *iff all its object nets are ordinary, all transitions are simple and the initial marking* $\mathbf{M}_0$ *is located.*

Consider $t[\boldsymbol{\theta}] \in \Theta$. For ordinary Os all arc weights $W(x, y) = 1$ iff $(x, y) \in F$. Additionally, if $|^{(N)}t| > 0$ there is exactly one place $p_N \in {}^\bullet t$ such that $d(p) = N$ and one place $p'_N \in t^\bullet$ such that $d(p') = N$. So, the variable $\mathbf{X}_{p,t}$ can be denoted as $\mathbf{X}_{d(p)}$ (and similar for $\mathbf{X}'_{t,p}$ etc.). Due to this one-to-one correspondence the representation can be simplified further:

– Value semantics: For the synchronisation tree $t[\boldsymbol{\theta}] \in \mathcal{T}_{n+1}(OS)$ the firing rule is generated from the firing rule of the θ_N:

$$\sum_{p \in {}^\bullet t} p[\mathbf{X}_{d(p)} + \mathbf{pre}(\theta_N)] \xrightarrow{t[\boldsymbol{\theta}]} \sum_{p \in t^\bullet} p[\mathbf{X}'_{d(p)} + \mathbf{post}(\theta_N)]$$

For a minimal synchronisation tree this further simplifies to:

$$\sum_{p \in {}^\bullet t} p[\mathbf{X}_{d(p)}] \xrightarrow{t[\boldsymbol{id}]} \sum_{p \in t^\bullet} p[\mathbf{X}'_{d(p)}]$$

– Reference semantics: For the synchronisation tree $t[\boldsymbol{\theta}] \in \mathcal{T}_{n+1}(OS)$:

$$\sum_{p \in {}^\bullet t} p + \mathbf{pre}(\mathrm{fl}(\theta_{d(p)})) \xrightarrow{\mathrm{fl}(t[\boldsymbol{\theta}])} \sum_{p \in t^\bullet} p + \mathbf{post}(\mathrm{fl}(\theta_{d(p)}))$$

For the minimal synchronisation tree this simplifies to:

$$\sum_{p \in {}^\bullet t} p \xrightarrow{\mathrm{fl}(t[\boldsymbol{id}])} \sum_{p \in t^\bullet} p$$

Theorem 3. *If OS is an ordinary* Os*, then all reachable markings are located and all places p with* $d(p) \neq N_\bullet$ *are 1-safe.*

Proof. The initial marking is located by definition. If $\mathbf{M}_r \xrightarrow{t} \mathbf{M}'_r$ then there is exactly one location for N (since $\sum_{p \in d^{-1}(N)} |\mathbf{M}_r(p)| = 1$ for all $N \in \mathcal{N} \setminus \{N_{sn}, N_\bullet\}$), which is either untouched or relocated, since all t are simple.

If $\mathbf{M}_r$ is located, then all places p with $d(p) \neq N_\bullet$ are marked with at most one net-token. Since every reachable marking of a simple Os is located these places are 1-safe. □

Using Theorem 3 we know that all reachable markings are located, we can conclude from Theorem 2:

$$\begin{array}{ll} \forall \mathbf{M}_r \in R(\mathrm{fl}(OS)) : & \mathrm{fl}(\mathrm{lc}(\mathbf{M}_r)) = \mathbf{M}_r \\ \forall \mathbf{M}_v \in R(OS) : & \mathrm{lc}(\mathrm{fl}(\mathbf{M}_v)) = \mathbf{M}_v \end{array}$$

Theorem 4. *For ordinary* Os *OS the mapping* lc *provides a direct embedding of the reference semantics. If* $\mathbf{M}_r$ *is located, then:*

$$\begin{array}{ccc} \mathbf{M}_r & \xrightarrow[\mathrm{fl}(OS)]{\theta} & \mathbf{M}'_r \\ \mathrm{lc}\downarrow & & \downarrow\mathrm{lc} \\ \mathrm{lc}(\mathbf{M}_r) & \xrightarrow[OS]{\theta} & \mathrm{lc}(\mathbf{M}'_r) \end{array}$$

Proof. Induction over the synchronisation tree $t[\boldsymbol{\theta}]$.

– If $\mathbf{M}_r \xrightarrow{t[\boldsymbol{id}]} \mathbf{M}'_r$ then by monotonicity we can add $\mathrm{fl}(\mathbf{X}_{d(p)})$ with $\mathbf{X}_{d(p)} = \mathrm{lc}(\mathbf{M}_{r|P(d(p))})$.

$$\mathbf{M}_r + \sum_{p \in {}^\bullet t} \mathrm{fl}(\mathbf{X}_{d(p)}) \xrightarrow{t[\boldsymbol{id}]} \mathbf{M}'_r + \sum_{p \in {}^\bullet t} \mathrm{fl}(\mathbf{X}'_{d(p)})$$

Since $\sum_{p \in {}^\bullet t} \mathbf{X}_{d(p)} = \sum_{p \in t^\bullet} \mathbf{X}'_{d(p)} =: X$ the basic tree is:

$$\begin{array}{cl} = & \mathrm{lc}(\mathbf{pre}^{\mathrm{fl}}(t + id_{\mathrm{fl}(\mathbf{X})})) \\ = & \mathrm{lc}(\sum_{p \in {}^\bullet t} p + \mathrm{fl}(\mathbf{X}_{d(p)})) \\ = & \sum_{p \in {}^\bullet t} p[\mathbf{X}_{d(p)}]) \\ \xrightarrow{t[\boldsymbol{id}]} & \sum_{p \in t^\bullet} p[\mathbf{X}'_{d(p)}]) \\ = & \mathrm{lc}(\sum_{p \in t^\bullet} p + \mathrm{fl}(\mathbf{X}'_{d(p)})) \\ = & \mathrm{lc}(\mathbf{post}^{\mathrm{fl}}(t + id_{\mathrm{fl}(\mathbf{X})})) \end{array}$$

– Induction: We add $\mathrm{fl}(\mathbf{Y}_{d(p)})$ with $\mathbf{Y}_{d(p)} = \mathbf{pre}(\theta_N)$ and $\mathrm{fl}(\mathbf{X}_{d(p)})$ with $\mathbf{X}_{d(p)} = \mathrm{lc}(\mathbf{M}_{r|P(d(p))}) - \mathbf{Y}_{d(p)}$. Note, that $\mathrm{lc}(\mathbf{M}_{r|P(d(p))}) \geq \mathbf{Y}_{d(p)}$ since $\theta_{d(p)}$ is activated. Let $\mathbf{Y}'_N = \mathbf{post}(\theta_N)$.

By assumption $\mathbf{pre}(\theta_N) \xrightarrow{\theta_N} \mathbf{post}(\theta_N)$:

$$\begin{array}{l} = \mathrm{lc}(\mathbf{pre}^{\mathrm{fl}}(t[\theta_{N_1}, \ldots, \theta_{N_n}] + id_{\mathrm{fl}(\mathbf{X})})) \\ = \mathrm{lc}(\sum_{p \in {}^\bullet t} p + \mathrm{fl}(\mathbf{X}_{d(p)}) + \mathrm{fl}(\mathbf{Y}_{d(p)})) \\ = \sum_{p \in {}^\bullet t} p[\mathbf{X}_{d(p)} + \mathbf{Y}_{d(p)}]) \\ \xrightarrow{t[\boldsymbol{\theta}]} \sum_{p \in t^\bullet} p[\mathbf{X}'_{d(p)} + \mathbf{Y}'_{d(p)}] \\ = \mathrm{lc}(\sum_{p \in t^\bullet} p + \mathrm{fl}(\mathbf{X}'_{d(p)}) + \mathrm{fl}(\mathbf{Y}'_{d(p)}) \\ = \mathrm{lc}(\mathbf{post}^{\mathrm{fl}}(t[\theta_{N_1}, \ldots, \theta_{N_n}] + id_{\mathrm{fl}(\mathbf{X})})) \end{array}$$

This shows the property. □

$$
\begin{array}{ccc}
\mathbf{M}_v & \xrightarrow[OS]{\theta} & \mathbf{M}'_v \\
{\scriptstyle \text{fl}}\downarrow & & \downarrow{\scriptstyle \text{fl}} \\
\text{fl}(\mathbf{M}_v) & \xrightarrow[\text{fl}(OS)]{\theta} & \text{fl}(\mathbf{M}'_v) \\
{\scriptstyle \text{lc}}\downarrow & & \downarrow{\scriptstyle \text{lc}} \\
\text{lc}(\text{fl}(\mathbf{M}_v)) & \xrightarrow[OS]{\theta} & \text{lc}(\text{fl}(\mathbf{M}'_v)) = \mathbf{M}'_v
\end{array}
\qquad
\begin{array}{ccc}
\mathbf{M}_r & \xrightarrow[\text{fl}(OS)]{\theta} & \mathbf{M}'_r \\
{\scriptstyle \text{lc}}\downarrow & & \downarrow{\scriptstyle \text{lc}} \\
\text{lc}(\mathbf{M}_r) & \xrightarrow[OS]{\theta} & \text{lc}(\mathbf{M}'_r) \\
{\scriptstyle \text{fl}}\downarrow & & \downarrow{\scriptstyle \text{fl}} \\
\text{fl}(\text{lc}(\mathbf{M}_r)) & \xrightarrow[\text{fl}(OS)]{\theta} & \text{fl}(\text{lc}(\mathbf{M}'_r)) = \mathbf{M}'_r
\end{array}
$$

Fig. 4. Embeddings extended to an Simulation

Theorem 5. *If OS is a ordinary* Os*, then reference and value semantics can simulate each other directly.*

Proof. Composition of the two diagrams in Theorem 1 and 4 is shown in Fig. 4. Both diagrams can be further reduced to the following two simulations:

$$
\begin{array}{ccc}
\mathbf{M}_v & \xrightarrow[OS]{\theta} & \text{lc}(\text{fl}(\mathbf{M}'_v)) = \mathbf{M}'_v \\
{\scriptstyle \text{fl}}\downarrow & & \uparrow{\scriptstyle \text{lc}} \\
\text{fl}(\mathbf{M}_v) & \xrightarrow[\text{fl}(OS)]{\theta} & \text{fl}(\mathbf{M}'_v)
\end{array}
\qquad
\begin{array}{ccc}
\mathbf{M}_r & \xrightarrow[\text{fl}(OS)]{\theta} & \text{fl}(\text{lc}(\mathbf{M}'_r)) = \mathbf{M}'_r \\
{\scriptstyle \text{lc}}\downarrow & & \uparrow{\scriptstyle \text{fl}} \\
\text{lc}(\mathbf{M}_r) & \xrightarrow[OS]{\theta} & \text{lc}(\mathbf{M}'_r)
\end{array}
$$

So, the embeddings in Theorem 1 and 4 also imply a direct simulation. □

5 Processes of Ordinary Object-Net Systems

In [8] we have given a characterisation of those processes of the reference semantics that can be simulated by the value semantics for the general case. For ordinary object-net systems we know due to Theorem 5 that there is a one-to-one correspondence of reference and value semantics. In the following we will show that this correspondence carries over for processes.

5.1 Basic Definitions

Petri net processes (cf. [6, 2]) describe the behaviour of Petri nets. Processes are themselves Petri nets from the class of *causal nets*, where no branching is allowed for the places. A *run* of a net N is defined as a causal net R with a pair of mappings $\phi = (\phi^P : B \to P, \phi^T : E \to T)$. Extending ϕ^P and ϕ^T to multisets, the run is associated to the net, by requiring the commutativity expressed by: $\phi^P({}^\bullet e) = \mathbf{pre}(\phi^T(e))$ and $\phi^P(e^\bullet) = \mathbf{post}(\phi^T(e))$. That is, ϕ preserves the localities of transitions.[4]

[4] Alternatively, a process (R, ϕ) can be constructed from the possible firings, i.e. the enabling of transitions, of the net N by adding transitions according to the enabling condition of the net N. The starting point is given by the initial marking.

Definition 7. *Let $N = (P, T, W, \mathbf{M}_0)$ be a P/T net and $R = (B, E, \lessdot)$ a causal net. Furthermore let $\phi = (\phi^P : B \to P, \phi^T : E \to T)$ be a pair of mappings. Then (R, ϕ) is a* process *of N if the following conditions hold:*

1. *Preservation of the flow relation: $x \lessdot y \Longrightarrow \phi(x)\ F\ \phi(y)$.*
2. *Representation of the initial marking $\mathbf{M}_0$ by ${}^\circ R$: $\phi^P({}^\circ R) = \mathbf{M}_0$.*
3. *Compatibility of ϕ with the arc-weight function:*
   $\phi^P({}^\bullet e) = \mathbf{pre}(\phi^T(e))$ and $\phi^P(e^\bullet) = \mathbf{post}(\phi^T(e))$.
4. *Representability of R as the limit of finite processes.*

For a run (R, ϕ) of a Petri net N the symmetric and reflexive relations **li** and **co** are defined by $\mathbf{li} := (< \cup <^{-1} \cup id_A)$ and $\mathbf{co} := (\bar{\mathbf{li}} \cup id_A)$. A ken with respect to **li** is often called a *line*, while a ken with respect to **co** is called a *cut*. If $C \in \text{KEN}(<)$ and $C \subseteq P$ then C is called a P-cut of R.

5.2 Processes of Ordinary Object Nets

The definition of of an object net process is based on the net $\text{fl}(OS)$ defined in Def. 3.

Definition 8. *Let $OS = (\mathcal{N}, d, \Theta, \mathbf{M}_0)$ be an* Os*. The pair (R, ϕ) is a process of OS iff it is a process of $\text{fl}(OS)$.*

Define the set of elements of a P-cut C belonging to a net type N:

$$B_N(C) := C \cap \phi^{-1}(d^{-1}(N)) = \{b \in C \mid d(\phi(b)) = N\}$$

Analogously to Theorem 3 we obtain that all reachable P-cuts are located:

Lemma 1. *Let (R, ϕ) be a process of an ordinary* Os *OS. For each P-cuts C of R there is exactly one element $b \in C$ carrying a net-token of type N.*

$$\forall N \in \mathcal{N} : |B_N(C)| = 1$$

The uniquely defined element of $B_N(C)$ is denoted by $b(N)$.

Similarly to Def. 5 we define a localisation of P-cuts resulting in a nested structure. The restructuring also extends to events, where each e is mapped to an nested event $e^t[\boldsymbol{\epsilon}]$ where ϵ is a nested structure of events which mimics the structure of $\phi(e) = t[\boldsymbol{\theta}]$.

Definition 9. *For a P-cut C of a process R, we define the* localisation $\text{lc}(C)$ *of C as $\text{lc}(C) := \text{lc}_C(B_{N_{sn}}(C))$ where*

$$\text{lc}_C(b) := \begin{cases} b[\text{lc}(B_{d(\phi(b))}(C))], & \text{if } d(\phi(b)) \neq N_\bullet \\ b, & \text{otherwise} \end{cases}$$

For events $e \in E$ we define $\text{lc}(e) := f_e(\phi(e))$ where

$$f_e(t[\boldsymbol{\theta}]) := e^t[f_e(\boldsymbol{\theta})]$$

The process mapping is extended to nested sets by defining $\phi(b[X]) := \phi(b)[\phi(X)]$. Then the localisation commutes with the process map ϕ.

Theorem 6. *Let (R, ϕ) be a process of of an ordinary* Os *OS. The process map ϕ commutes with* lc. *For each P-cut C of R we have:*

$$\phi(\mathrm{lc}_C(C)) = \mathrm{lc}_{\phi(C)}(\phi(C))$$

Proof. Induction base for $N_\bullet$:

$$\phi(\mathrm{lc}_C(B_{N_\bullet}(C)) = \phi(B_{N_\bullet}(C)) = \mathrm{lc}_{\phi(C)}(\phi(B_{N_\bullet}(C))$$

Induction step:

$$\begin{aligned}
\phi(\mathrm{lc}_C(B)) &= \phi\Big(\textstyle\sum_{b\in B}(b, \mathrm{lc}_C(B_{d(\phi(b))}(C)))\Big) \\
&= \textstyle\sum_{b\in B} \phi(b)[\phi(\mathrm{lc}_C(B_{d(\phi(b))}(C))] \\
&= \textstyle\sum_{b\in B} \phi(b)[\mathrm{lc}_{\phi(C)}\big(\phi(B_{d(\phi(b))}(C))] \\
&= \textstyle\sum_{b\in B} \phi(b)[\mathrm{lc}_{\phi(C)}\big(\phi(C \cap \phi^{-1}(d^{-1}(N)))] \\
&= \textstyle\sum_{b\in B} \phi(b)[\mathrm{lc}_{\phi(C)}\big(\phi(C) \cap d^{-1}(N)] \\
&= \textstyle\sum_{b\in B} \phi(b)[\mathrm{lc}_{\phi(C)}(\phi(C)_{|P(N)})] \\
&= \mathrm{lc}_{\phi(C)}(\phi(B))
\end{aligned}$$

This proves the commutativity. □

Theorem 7. *Let (R, ϕ) a process of an ordinary* Os *OS. Then we have for all P-cuts C and C' of R:*

$$\begin{array}{ccc}
C & \xrightarrow[R]{e} & C' \\
\phi\downarrow & & \downarrow\phi \\
\phi(C) & \xrightarrow[\mathrm{fl}(OS)]{\phi(e)} & \phi(C') \\
\mathrm{lc}\downarrow & & \downarrow\mathrm{lc} \\
\mathrm{lc}(\phi(C)) & \xrightarrow[OS]{\phi(e)} & \mathrm{lc}(\phi(C'))
\end{array}$$

Proof. By definition R is a process of $\mathrm{fl}(OS)$, which shows the first embedding via ϕ. Theorem 4 shows that every step for an ordinary Os can be simulated via lc which is the second embedding. □

The map lc associates with each process R a nested process which is an object net system: "The semantics of an object net is an object net."

Definition 10. *Let (R, ϕ) with $R = (B, E, \lessdot)$ be a process of the* Os *$OS = (\mathcal{N}, d, \Theta, \mathbf{M}_0)$. Define the* located process $(\mathrm{lc}(R), \phi_R)$ *by the* Os

$$\mathrm{lc}(R) = (\mathcal{N}_R, d_R, \Theta_R, \mathbf{M}_R)$$

where $\mathcal{N}_R = \{R_N \mid N \in \mathcal{N}\}$ with

$$\begin{aligned} B(R_N) &= B \cap \phi^{-1}(P(N)) \\ E(R_N) &= \{e^t \mid t \in \mathrm{fl}(\phi(e)) \wedge t \in T(N)\} \\ F(R_N) &= \lessdot_{|(B(R_N)\times E(R_N)) \cup (E(R_N)\times B(R_N))} \end{aligned}$$

and $d_R(b) = d(\phi(b))$*,* $\Theta_R(b) = E$*, and* $\mathbf{M}_R(b) = \mathrm{lc}({}^\circ R)$*. The process mapping is defined by* $\phi_R(b) = \phi(b)$ *and* $\phi_R(e^t) = \phi(e)$*.*

Analogously to the previous Theorem we obtain the following embedding when the application order of lc and ϕ is switched.

Theorem 8. *Let* $OS = (\mathcal{N}, d, \Theta, \mathbf{M}_0)$ *be an ordinary* Os *and* (R, ϕ) *a process of* OS*. Then we have for all* P*-cuts* C *and* C' *of* R*:*

$$\begin{array}{ccc} C & \xrightarrow[R]{e} & C' \\ \mathrm{lc}\downarrow & & \downarrow\mathrm{lc} \\ \mathrm{lc}(C) & \xrightarrow[\mathrm{lc}(R)]{\mathrm{lc}(e)} & \mathrm{lc}(C') \\ \phi_R\downarrow & & \downarrow\phi_R \\ \phi_R(\mathrm{lc}(C)) & \xrightarrow[OS]{\phi_R(\mathrm{lc}(e))} & \phi_R(\mathrm{lc}(C')) \end{array}$$

Proof. Let $C \xrightarrow[R]{e} C'$ and $\phi(e) = t[\boldsymbol{\theta}]$ then by Def. 9 $\mathrm{lc}(e) = e^t[\boldsymbol{\epsilon}]$. It is easy to see that by the construction in Def. 10 the event $e^t[\boldsymbol{\epsilon}]$ is enabled in $\mathrm{lc}(R)$ for the nested cut $\mathrm{lc}(C)$: $\mathrm{lc}(C) \xrightarrow[\mathrm{lc}(R)]{\mathrm{lc}(e)} \mathrm{lc}(C')$.

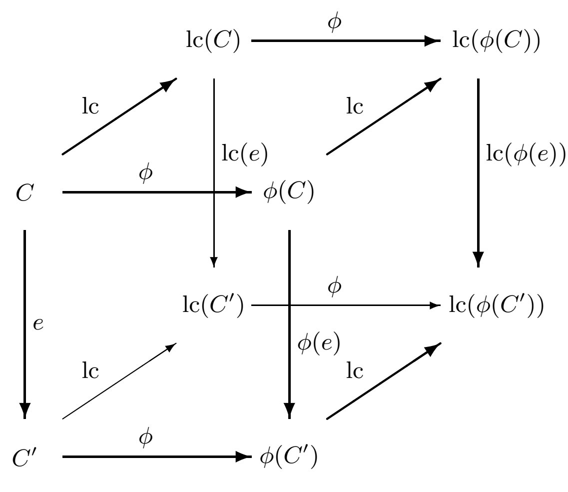

Fig. 5. Process embeddings

Since ϕ_R equals ϕ on places, we have $\phi_R(\mathrm{lc}(C)) = \phi(\mathrm{lc}(C))$. Using Theorem 6 and 7 we know, that $\phi_R(\mathrm{lc}(C)) = \mathrm{lc}(\phi(C))$ can be rewritten by $\phi(e)$.

Since $\phi_R(\mathrm{lc}(e)) = \phi_R(f_e(\phi(e))) = \phi_R(e^t[\boldsymbol{\epsilon}]) = t[\boldsymbol{\theta}] = \phi(e)$ holds, we have $\phi_R(\mathrm{lc}(C)) \xrightarrow[OS]{\phi_R(\mathrm{lc}(e))} \phi_R(\mathrm{lc}(C'))$ for the object-net system OS. □

The cube in Figure 5 summarises all the embeddings of Theorem 7 and 8. The vertical dimension illustrates the firing steps, the dimension from left to right illustrates the mapping from the process to the object system, the dimension from the front to the back illustrates the relation of reference and value semantics.

6 The Household Robot Example, Revisited

In [10] the authors showed how the formalism of nets-within-nets can be used to model mobility, especially in the case of mobile agents. A case study was presented that models a mobile household robot. We adapt this case study for a first approach on how to analyse agent systems. To reach this aim the overall system architecture is simplified while the ideas are still visible. Going along with a better tool support for the analysis we will switch back to the original model.

The household is represented by the system net in Figure 6.[5] The household consists of several rooms (locations): hall, living room, kitchen, next room, and the front yard (dark places). Each room offers special services to the robot: it can fetch coffee in the kitchen, serve it in the living room, fetch mail in the front yard, open and close the door in the hall, and so on (light transitions). The possible movements from one location to another are displayed as dark transitions. Note that moving from room to room is not symmetric in this scenario. For example it is not possible to move directly from the kitchen to the next room. Service transitions are supplemented with additional information (service state/buffer, light places) showing for instance if new mail has arrived, coffee is available and so on. Extraneous actions not accessible for the robot are displayed as thin-lined transitions: arrival of new mail, new assignments for the robot etc.

The door of the house is used to show another possibility of viewing special parts of the system: the state of the door (open/closed) is modelled directly. This system state does not belong to a single service (as for example the state of the mailbox), but is queried by a couple of service transitions including the movements into and out of the house.

This model of the household is filled with life by implementing an appropriate robot agent and defining the desired services for the platforms. The behaviour modelling for this kind of agents has been introduced in [9]. While the Petri net model of the household hides some details – namely the transition inscriptions

[5] The use of colour greatly supports the differentiation of different types of places, transitions, or arcs. Unfortunately this is – even in the adapted form of the figures – not so obvious in a black and white representation.

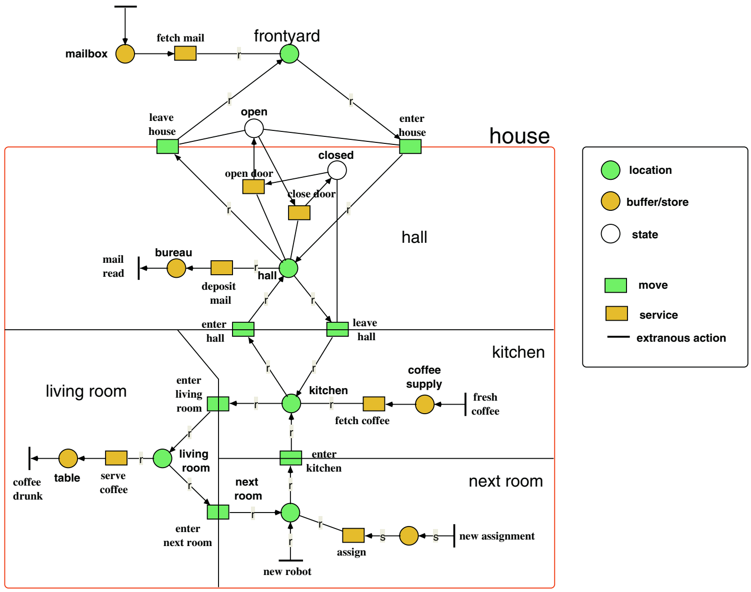

Fig. 6. Household System Net

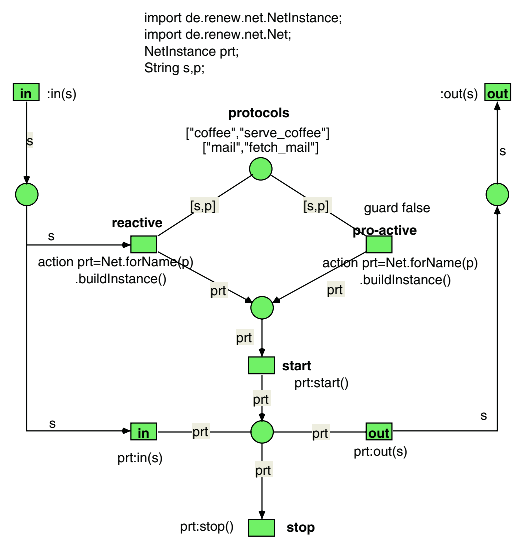

Fig. 7. The robot

e.g. for moving around – the nets for the robot (Fig. 7) and one of its plans (Fig. 8) are presented in full detail using the syntax of Renew [14].

Figure 7 shows the interface net of the robot implemented as a Mulan agent. This kind of agents is explained in [9]. The Figure shows a simplified version still capable of autonomous, pro- or reactive and reconfigurable behaviour. What is

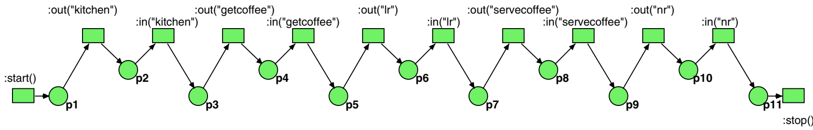

Fig. 8. Plan for serving coffee

omitted in this publication is the platform net, the layer between the overall system (household) and the agent (robot). It is necessary only if one is interested in a dynamically changing environment. Leaving it out we get an architecture of three layers: system - agent - behaviour protocol (plan).

Figure 8 presents a behaviour protocol of the robot. This protocol only consists of sequential actions, therefore we call it a *plan*. Protocols tell the agent what to do and when to do it. Protocol nets can be generated at run-time, which can not be shown here. The plan for serving coffee instruments the robot to move to the kitchen, fetch the coffee, move to the living room, serve the coffee and move back to the next room. Having done this the plan stops.

We have analysed the household system where the house net has been restricted to the kitchen, the next and the living room, since the yard and the hall are not relevant in the restricted scenario. The transitions new assignment, fresh coffee, and coffee drunk have been fused to generate some kind of loop at the system net level. For this example an analysis with INA [17] (integrated in the PEP tool [7]) shows that the simulating P/T net is ordinary, extended simple, bounded, reversible and live. It has no transitions without any pre- or without any post-place. The system is covered by semi-positive P-invariants and thus structurally bounded. It is covered by a semi-positive T-invariant containing each transition once. There are 99 minimal deadlocks. All nonempty traps are initially marked. The system is state-machine decomposable.

Since the system is bounded we know that the design relies on only finitely many resources and since it is live we know e.g. that the robot can offer its service regularly.

7 Conclusion

In this presentation we have introduced the subclass of ordinary object-net systems. This subclass is of special importance because its structure guarantees that each marking is located. We have shown that this implies further, that reference and value semantics can simulate each other directly for this subclass. The structural simulation is also compatible with the concept of a Petri net process – illustrated by the cube in Figure 5. Due to this one-to-one correspondence a formal analysis based on standard tools is possible. Structural and dynamic properties were checked for our household/robot example using the tool INA. Current work is undertaken to investigate extensions of the formalism to allow for high-level concepts as arc inscriptions, bindings etc.

References

1. M. A. Bednarczyk, L. Bernardinello, W. Pawlowski, and L. Pomello. Modelling mobility with Petri hypernets. In J. L. Fiadeiro, P. D. Mosses, and F. Orejas, editors, *Recent Trends in Algebraic Development Techniques (WADT 2004)*, Lecture Notes in Computer Science. Springer-Verlag, 2004.
2. E. Best and C. Fernández. *Nonsequential processes: a Petri net view.* Springer-Verlag, 1988.
3. N. Busi. Mobile nets. *Formal Methods for Open Object-Based Distributed Systems*, pages 51–66, 1999.
4. L. Cardelli, A. D. Gordon, and G. Ghelli. Mobility types for mobile ambients. In *Proceedings of the Conference on Automata, Languages, and Programming (ICALP'99)*, volume 1644 of *Lecture Notes in Computer Science*, pages 230–239. Springer-Verlag, 1999.
5. D. M. Chess. Security issues in mobile code systems. In G. Vigna, editor, *Mobile Agents and Security*, volume 1419 of *Lecture Notes in Computer Science*, pages 1–17. Springer-Verlag, 1998.
6. U. Goltz and W. Reisig. The non-sequential behaviour of Petri nets. *Information and Control*, 57:125–147, 1983.
7. B. Grahlmann and E. Best. PEP — more than a Petri net tool. In T. Margaria and B. Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 1055 of *Lecture Notes in Computer Science*, pages 397–401. Springer-Verlag, 1996.
8. M. Köhler and B. Farwer. Mobile object-net systems and their processes. *Fundamenta Informaticae*, 59:113–129, 2004.
9. M. Köhler, D. Moldt, and H. Rölke. Modeling the behaviour of Petri net agents. In J. M. Colom and M. Koutny, editors, *International Conference on Application and Theory of Petri Nets*, volume 2075 of *Lecture Notes in Computer Science*, pages 224–241. Springer-Verlag, 2001.
10. M. Köhler, D. Moldt, and H. Rölke. Modelling mobility and mobile agents using nets within nets. In W. v. d. Aalst and E. Best, editors, *International Conference on Application and Theory of Petri Nets 2003*, volume 2679 of *Lecture Notes in Computer Science*, pages 121–140. Springer-Verlag, 2003.
11. M. Köhler and H. Rölke. Concurrency for mobile object-net systems. *Fundamenta Informaticae*, 54(2-3), 2003.
12. M. Köhler and H. Rölke. Properties of Object Petri Nets. In J. Cortadella and W. Reisig, editors, *International Conference on Application and Theory of Petri Nets 2004*, volume 3099 of *Lecture Notes in Computer Science*, pages 278–297. Springer-Verlag, 2004.
13. O. Kummer. *Referenznetze*. Logos Verlag, 2002.
14. O. Kummer, F. Wienberg, M. Duvigneau, J. Schumacher, M. Köhler, D. Moldt, H. Rölke, and R. Valk. An extensible editor and simulation engine for Petri nets: Renew. In J. Cortadella and W. Reisig, editors, *International Conference on Application and Theory of Petri Nets 2004*, volume 3099 of *Lecture Notes in Computer Science*, pages 484 – 493. Springer-Verlag, 2004.
15. I. A. Lomazova. Nested Petri nets – a formalism for specification of multi-agent distributed systems. *Fundamenta Informaticae*, 43(1-4):195–214, 2000.
16. W. Reisig. Petri nets and algebraic specifications. *Theoretical Computer Science*, 80:1–34, 1991.
17. S. Roch and P. Starke. INA – Integrierter Netz-Analysator Version 1.7. Handbuch, Humboldt-University, Institute of Computer Science, Berlin, 1997.

18. M.-O. Stehr, J. Meseguer, and P. C. Ölveczky. Rewriting logic as a unifying framework for Petri nets. In H. Ehrig, G. Juhas, J. Padberg, and G. Rozenberg, editors, *Unifying Petri Nets*, Lecture Notes in Computer Science (Advances in Petri Nets). Springer-Verlag, December 2001.
19. R. 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.
20. R. Valk. Petri nets as token objects: An introduction to elementary object nets. In J. Desel and M. Silva, editors, *Application and Theory of Petri Nets*, volume 1420 of *Lecture Notes in Computer Science*, pages 1–25, 1998.
21. R. Valk. Object Petri nets: Using the nets-within-nets paradigm. In J. Desel, W. Reisig, and G. Rozenberg, editors, *Advanced Course on Petri Nets 2003*, volume 3098 of *Lecture Notes in Computer Science*, pages 819–848. Springer-Verlag, 2003.

A Notations and Basic Definitions

Let $R \subseteq A \times B$ be a relation. A pair $(a,b) \in R$ will also be denoted $a\,R\,b$ in infix notation. The identity relation is defined as $id_A := \{(a,a) \mid a \in A\}$. For $a \in A$ and $b \in B$ define the domain of an element $b \in B$ by $(\_R\,b) := \{a \mid (a,b) \in R\}$ and its co-domain by $(a\,R\_) := \{b \mid (a,b) \in R\}$. We generalise the notion of domain and co-domain to sets $C \subseteq A$ and $D \subseteq B$ by $(C\,R\_) := \{b \mid \exists a \in C : (a,b) \in R\}$ and $(\_R\,D) := \{a \mid \exists b \in D : (a,b) \in R\}$.

Let $R \subseteq A \times A$ be a symmetric and reflexive relation. The set $K \subseteq A$ is a *clique* with respect to R iff all pairs of its elements are in the relation, i.e. for all $x, y \in K$ we have $(x,y) \in R$. A maximal clique is called a *ken* and the set of all kens of R is denoted by $\textsc{ken}(R)$.

The definition of Petri nets relies on the notion of multisets. A multiset on the set D is a mapping $\mathbf{A} : D \to \mathbb{N}$. Multisets are generalisations of sets in the sense that every subset of D corresponds to a multiset $\mathbf{A}$ with $\mathbf{A}(x) \leq 1$ for all $x \in D$. The empty multiset $\mathbf{0}$ is defined as $\mathbf{0}(x) = 0$ for all $x \in D$. The cardinality is $|\mathbf{A}| := \sum_{x \in D} \mathbf{A}(x)$. A multiset $\mathbf{A}$ is called *finite* iff $|\mathbf{A}| < \infty$. The multiset sum $\mathbf{A} + \mathbf{B}$ is defined as $(\mathbf{A} + \mathbf{B})(x) := \mathbf{A}(x) + \mathbf{B}(x)$ the difference $\mathbf{A} - \mathbf{B}$ by $(\mathbf{A} - \mathbf{B})(x) := \max(\mathbf{A}(x) - \mathbf{B}(x), 0)$. Equality $\mathbf{A} = \mathbf{B}$ is defined element-wise: $\forall x \in D : \mathbf{A}(x) = \mathbf{B}(x)$. Multisets are partially ordered: $\mathbf{A} \leq \mathbf{B} \iff \forall x \in D : \mathbf{A}(x) \leq \mathbf{B}(x)$ The strict order $\mathbf{A} < \mathbf{B}$ holds iff $\mathbf{A} \leq \mathbf{B}$ and $\mathbf{A} \neq \mathbf{B}$. The notation is overloaded, being used for sets as well as multisets. The meaning will be apparent from its use.

Any mapping $f : D \to D'$ can be generalised to a mapping $f : MS(D) \to MS(D')$ on multisets:

$$f\left(\sum_{i=1}^{n} a_i\right) = \sum_{i=1}^{n} f(a_i)$$

This includes the special case $f(\mathbf{0}) = \mathbf{0}$. These definitions are in accordance with the set-theoretic notation $f(A) = \{f(a) \mid a \in A\}$.

The set of all finite multisets over the set D is denoted $MS(D)$. A multiset $\mathbf{A}$ can be considered as the formal sum $\mathbf{A} = \sum_{x \in D} \mathbf{A}(x) \cdot x$. Finite multisets are the freely generated commutative monoid $(MS(D), +, 0)$. If the set D is finite, then a multiset $\mathbf{A} \in MS(D)$ can be represented equivalently as a vector $\mathbf{A} \in \mathbb{N}^{|D|}$.

$N = (P, T, F)$ is a *Petri net* iff the set of places P and the set of transitions T are disjoint, i.e. $P \cap T = \emptyset$, $F \subseteq (P \times T \cup T \times P)$ is the flow relation. Some commonly used notations for Petri nets are ${}^\bullet y := (\_F\, y)$ for the *preset* and $y^\bullet := (y\, F\_)$ for the *postset* of a net element y. The set of minimal elements of a net N is denoted ${}^\circ N := \{x \in P \cup T \mid {}^\bullet x = \emptyset\}$, the set of maximal elements is $N^\circ := \{x \in P \cup T \mid x^\bullet = \emptyset\}$.

A finitely branching Petri net $N = (B, E, \lessdot)$ is an *causal net* iff the transitive closure $\lessdot^+$ of the flow is acyclic and $|{}^\bullet b| \leq 1$ and $|b^\bullet| \leq 1$ holds for all $b \in B$. For a causal net $N = (B, E, \lessdot)$ we define the order $<$ on the net elements $(B \cup E)$ by $< := \lessdot^+$.