Mobile Object Net Systems: Petri Nets as Active Tokens

Michael Köhler

University of Hamburg, Department of Computer Science
koehler@informatik.uni-hamburg.de

Abstract. In this work we present the model of "mobile object net systems" – an algebraic formalisation of the "nets within nets"-paradigm, which is well suited to express the dynamics of open, mobile systems, since it allows tokens to be active. The algebraic theory of "nets within nets" covers an integrated view of the two major topics concurrency and locality, which are central in the area of mobile computing. As a main result of this contribution, we derive an algebraic model for "nets within nets" in the "Petri nets are monoids" style.

Keywords: concurrency, mobility, nets within nets, mobile object net systems, Petri nets as monoids

1 Introduction

The general context of this presentation is the formalisation of mobility in terms of the "nets within nets" approach by Valk [Val91,Val98,Val00a], where the tokens of a Petri net are also Petri nets, i.e. tokens are entities with inner actions. Valk studies the special situation where one net – the "object net" – is the token of another net – called "system net". So, a two level hierarchy is described.

This contribution defines the model of *"mobile object net systems"*, which generalises the model of [Val98,Val00a] by allowing an arbitrary number of net types and an arbitrary nesting level of nets. The nesting of token-nets into other nets naturally describes the concept of a location, which is essential for mobility.

Monoidal categories in the "Petri nets are monoids"-style of [MM90,MMS97] are the fundamental descriptions of Petri nets (cf. [Mes92,SMÖ01]). In this contribution it is shown, that the semantics of "nets within nets" can be characterised as a monoidal category. This is an interesting result, since it makes clear, that the structure of hierarchical transitions can be treated the same way as for place/transition nets. So, analysis methods for place/transition nets are applicable. The existence of categorial characterisation is not so obvious, since the firing of transitions at different levels may interfere with each other. For example, the action of a token-net can interfere with the action that moves the token-net itself.

The paradigm of nets within nets has been adopted by other approaches, for example the formalism of nested Petri nets [Lom99]. The relationship of the nets within nets formalism introduced by Valk to linear logic has been discussed in [Far99,Far00]. The "nets within nets" approach has been implemented in the tool Renew by [KWD03] based on [Kum98,Kum01].

The concept of a Petri net as a token is related to mobility-calculi like [CGG99] or [VC98], since they both integrate the idea of a location, where

computation takes place.[1] Mobility can easily be expressed by the "nets within nets"-paradigm: one net serves as the environment, another net as the mobile entity (cf. [KR02b]).

This presentation is structured as follows: Section 2 gives a short introduction to the model of mobile object net systems. Section 3 describes the algebraic viewpoint on Petri nets. In section 4 the structure of object nets and mobile object net systems is given. The central part of this work is presented in section 5, where the semantics of mobile object net systems is defined. As a main result we obtain a clear description of concurrency for object net systems in accordance with the "Petri nets are monoids" style. Thus, object net systems are described by a monoidal category. The work closes with a conclusion.

2 Mobility and Nets within Nets

The model of "mobile object net systems" (MONS) is based on the paradigm of "nets within nets" due to Valk [Val98,Val00a]. The paradigm formalises the aspect that tokens of a Petri net can again be nets. Taking this as a view point it is possible to model hierarchical structures in an elegant way. In this context the modelling of mobility can be achieved very naturally: A mobile entity is described by a Petri net which itself is a token of another Petri net describing the surrounding system.

Nets within nets can be investigated from two points of view: reference and value semantics [Val00b]. For *reference semantics* net tokens are considered as references to the net itself. The same reference can be used as a marking for more than one place. For *value semantics* each net token is considered as a value, that describes the marking of the net token, which includes also all markings of net tokens at deeper nesting levels.

Which semantics applies best to mobility? Reference semantics assumes a global name space, since net tokens can be referenced anywhere in the system. The change of a marking in one net token has an effect to all places in the system, where the net token is referenced. This situation is analysed in [Val00b] for the *α-centauri example*: The net token represents a log book, which is copied. One copy is send to α-centauri, one remains on earth. Changes to the log book on α-centauri are visible on earth and vice versa. So, reference semantics does not apply to systems with distributed localities.

Value semantics describes net tokens as values, so side effects are impossible. But, another difficulty arises. Consider the α-centauri example: The net token representing the log book on α-centauri is completely independent form the copy on earth. Inconsistent evolution of the log books cannot be detected before the books are recombined again. Moreover, in general the marking of the net token does not contain enough information to detect inconsistencies in all cases, so for the general case, the processes of the net tokens have to be recorded. This leads to the *process semantics*, where net tokens are represented by their processes. When copies of net tokens are recombined, their processes are combined to their least upper bound.

[1] This is different to the π-calculus of [MPW92], where the concept of a location is not formalised explicitly, but only implicitly by exchanging names. So, nets within nets are quite different from mobile Petri nets [AB96], which are based on the π-calculus.

If the least upper bound does not exist due to inconsistent process evolution an error has occurred. But in fact, the error has occurred at the point, when a conflict in the net tokens has been resolved differently. So, value and process semantics do not match the expectations for mobility.

Our conclusion of this discussion is, that the value semantics has to be adjusted for mobility. The problem is due to the definition, that net tokens are copied on multiple outgoing arcs. If net tokens are copied inconsistency is possible. Also, by duplicating tokens a second problem arises: Tokens representing resources leads to the situation, that the same resource can be used more than once by different copies at different locations.

The *mobility semantics* presented in the following overcomes the problem by *distributing* the marking of a net token to the copies on outgoing arcs rather than copying them. Each token of the net token is assigned to exactly one copy, so resources are used at most once and inconsistent evolution of net processes is impossible.[2] The resulting model is called "*mobile object net systems*" (MONS).

For mobile scenarios mobility arises recursively: robots can move in a ship, which crosses the ocean. So, as an extension of [Val98,Val00a] we allow an unlimited nesting structure for MONS, so the tokens of a net token can be net tokens again and so on.

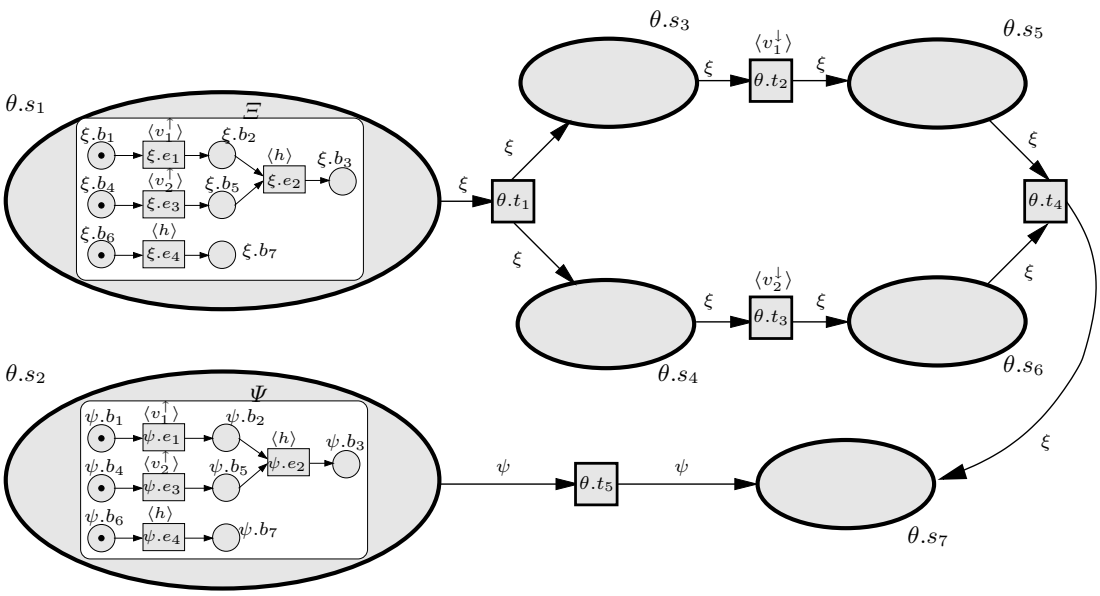

Fig. 1. Example scenario

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". The places $\theta.s_1$ and $\theta.s_2$ of the system net named

[2] Another approach is to ignore the aspect of replicated data. Then net tokens that are duplicated are treaten as different object. This viewpoint is considered in [Lom99].

θ are each marked with one net token of the same net type named Ξ and Ψ. 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 The places of the system net model agent platforms and transitions model mobility between them.

Object and system nets synchronise via *channels*. A channel l is denoted as transition inscription of the form $\langle l \rangle$. Transitions without an inscription can fire autonomously and concurrently to other autonomous transitions.

The transition $\theta.t_1$ distributes the net token Ξ to the places $\theta.s_3$ and $\theta.s_4$. The two copies have their own markings, where it is assumed that the sum of the markings in the copies equals the original marking of Ξ. So, a mobile agent is able to duplicate, but the copies have to share the original resources. For this example assume the object net token on $\xi.b_1$ is assigned to the copy on $\theta.s_3$, while the tokens on $\xi.b_4$ and $\xi.b_6$ are assigned to the copy on $\theta.s_4$.

For this distribution two *vertical* synchronisations are possible: $\theta.t_2$ synchronises with $\xi.e_1$ over the channel v_1 and $\theta.t_3$ synchronises with $\xi.e_1$ over the channel v_2 The synchronised firing of $\theta.t_2$ and $\xi.e_1$ moves the first copy of the net token Ξ from $\theta.s_3$ to $\theta.s_5$ and inside the net token the token on $\xi.b_1$ is fired to $\xi.b_2$. Analogously, the second copy is transported from $\theta.s_4$ to $\theta.s_6$ and inside the net token the token on $\xi.b_4$ is fired to $\xi.b_5$.

Transition $\theta.t_4$ recombines the two copies of the net token Ξ by recombining their markings, resulting in the marking $\xi.b_2 + \xi.b_5 + \xi.b_6$. The recombined net token is placed on $\theta.s_7$. Independently, transition $\theta.t_5$ transports the net token Ψ from $\theta.s_2$ to $\theta.s_7$.

The two net tokens Ξ and Ψ are now lying on the same place $\theta.s_7$. On this place they can synchronise *horizontally* over the channel h: the synchronous firing of $\xi.e_2$ and $\psi.e_4$ fires $\xi.b_2 + \xi.b_5$ to $\xi.b_3$ and $\psi.b_6$ to $\psi.b_7$.

3 Algebra of Petri Nets

Following the work of [MM90] we describe Petri nets in terms of their monoidal structure. This structure is build by multi-sets of places. In general, multi-sets over a set A are denoted as $MS(A)$. Together with multi-set addition $+$ and the neutral element 0 the structure $(MS(A), +, 0)$ forms a monoid. We assume the usual notions for multi-sets: Let M, M' be multi-sets, then $|M|$ denotes the cardinality of M. The usual ordering on multi-sets is denoted by $M < M'$. Also, all operators for sets are extended in the usual way. In the case of finite multi-sets, a multi-set M is denoted as a formal sum: $M = \sum_{i \in I} a_i$ or short: $M = \sum_I a_i$ where $a_i \in A$. Multi-sets are assumed to be finite in the following.

Definition 1. *A* Petri net *is a tuple $N = (S, T, \partial_0, \partial_1, m_0)$, where S is a set of places, T is a disjoint set of transitions and $\partial_0, \partial_1 : T \to MS(S)$ are mappings from transitions to their domain and their co-domain. The initial marking $m_0 \in MS(S)$ is a multi set.*

Given a set of labels L, a labelled Petri net *is the tuple $N = (S, T, \partial_0, \partial_1, l, m_0)$ where $l : T \to L$ is the labelling function.*

Given a labelled Petri net $N = (S, T, \partial_0, \partial_1, l, m_0)$, the components are denoted as $S(N)$, $T(N)$, $\partial_0(N)$, $\partial_1(N)$, $l(N)$, and $m_0(N)$. A transition of a Petri

net is denoted as $t : \sum_I s_i \to \sum_J s'_j$ for short, iff $\partial_0(t) = \sum_I s_i$ and $\partial_1(t) = \sum_J s'_j$ hold.

The semantics of a Petri net can be described by the structure $U(T)$, which describes sequential as well as concurrent actions (cf. [MM90]). Sequence is described by the operator ; and concurrency by $\oplus$. In the following definition of the structure $U(T)$ the mappings ∂_0 and ∂_1 are lifted from T to $U(T)$.

Definition 2. *Let $N = (S, T, \partial_0, \partial_1, m_0)$ be a Petri net. The structure $U(T)$ is generated by the following rules:*

(Identity-steps)

$$\frac{m \in MS(S)}{id_m : m \to m \in U(T)}$$

(Generator)

$$\frac{t : \partial_0(t) \to \partial_1(t) \in T}{t : \partial_0(t) \to \partial_1(t) \in U(T)}$$

(Sequence)

$$\frac{u_1, u_2 \in U(T) \quad \text{with} \quad \partial_1(u_1) = \partial_0(u_2)}{(u_1; u_2) : \partial_0(u_1) \to \partial_1(u_2) \in U(T)}$$

(Concurrency)

$$\frac{u_i : \qquad \partial_0(u_i) \to \partial_1(u_i) \in U(T), \quad i \in \{1, 2\}}{(u_1 \oplus u_2) : \partial_0(u_1) + \partial_0(u_2) \to \partial_1(u_1) + \partial_1(u_2) \in U(T)}$$

For all $u_1, u_2, u_3 \in U(T)$ and A, B,C,A',B',$C' \in MS(S)$ the following equations hold:

(Associativity)

$$(u_1 \oplus u_2) \oplus u_3 = u_1 \oplus (u_2 \oplus u_3)$$
$$(u_1; u_2); u_3 = u_1; (u_2; u_3)$$

(Commutativity)

$$u_1 \oplus u_2 = u_2 \oplus u_1$$

(Identity) *For all $u \in U(T)$ the the operator ; has the left identity $id_{\partial_0(u)}$ and the right identity $id_{\partial_1(u)}$. The operator $\oplus$ has the identity element id_0:*

$$u; id_{\partial_1(u)} = id_{\partial_0(u)}; u = u$$
$$u \oplus id_0 = \ id_0 \oplus u \ = u$$

(Distributivity of + and $\oplus$)

$$id_{A+B} = id_A \oplus id_B$$

(Distributivity of ; and $\oplus$) *For all $u_1 : A \to B$, $u_2 : A' \to B'$, $u_3 : B \to C$ and $u_4 : B' \to C'$:*

$$(u_1 \oplus u_2); (u_3 \oplus u_4) = (u_1; u_3) \oplus (u_2; u_4)$$

The identity-steps together with the addition rules for concurrency create the system states and steps. The usual definition of a step $m_0 \xrightarrow{t} m_1$ of transition t from a state m_0 to m_1 is possible iff $\exists m : m_i = m + \partial_i(t)$, $i \in \{0, 1\}$ holds. This is expressed by the identities:

$$\frac{\begin{array}{rc} t: & \partial_0(t) \to \partial_1(t) \\ id_m: & m \to m \end{array}}{(id_m \oplus t) : m + \partial_0(t) \to m + \partial_1(t)}$$

The structure $U(T)$ forms the morphism of a monoidal category[3].

Definition 3. *Let $N = (S, T, \partial_0, \partial_1, m_0)$ be a Petri net. The generated category $\mathcal{N}$ is defined as:*

$$\mathcal{N} = (MS(S), U(T), \partial_0, \partial_1, ;, id_m),$$

For coloured nets this notation can be extended. Each place s has a *coloured token* $c \in d(s) \in \mathcal{C}$ assigned to it by the mapping $d : S \to \mathcal{C}$. Each *colour* $C \in \mathcal{C}$ is a set. The set of *marked places* is:

$$\hat{S} := \{(s, c) \mid s \in S, c \in d(s)\} \tag{1}$$

For the sake of simplicity we present the formalism for object nets that are arc constant coloured Petri nets, i.e nets with variable-free arc inscriptions. Arc constant nets are obtained by substituting a transition t by its binding set. Variables are introduced into the formalism the usual way.

Definition 4. *Let $\mathcal{C}$ be a colour set. An* arc constant coloured Petri net *is a tuple*

$$N = (S, T, d, \partial_0, \partial_1, m_0),$$

where S is a set of places, T is a disjoint set of transitions, $d : S \to \mathcal{C}$ is the colour mapping, and $\partial_0, \partial_1 : T \to MS(\hat{S})$ are mappings from transitions to their domain and their co-domain. The initial marking $m_0 \in MS(\hat{S})$ is a multi set over $\hat{S}$.

The tuple $N = (S, T, d, \partial_0, \partial_1, l, m_0)$ with $l : T \to L$ is called a labelled, arc constant coloured Petri net.

Analogous to Petri nets, the category $\mathcal{N} = (MS(\hat{S}), U(T), \partial_0, \partial_1, ;, id_m)$ describes the behaviour of a coloured net N.

[3] For a general introduction into monoidal categories cf. [ML71] or [BW90].

4 Mobile Object Net Systems

Mobile Object net systems consist of a set of object nets, where one object net can be regarded as the place colour for other nets. To avoid a circular definition an mobile object net system is based on a set $\{D_{N_1} \ldots, D_{N_n}\}$ of net *descriptors* which corresponds to the set of object nets. We assume the *primitive descriptor* $D_{N_*} \notin \{D_{N_1} \ldots, D_{N_n}\}$, which is used to model black tokens, in the set of descriptor. The descriptor domain is:

$$\mathbf{D} = \{D_{N_1} \ldots, D_{N_n}\} \cup \{D_{N_*}\}$$

Colours Net instances as the tokens of an mobile object net system belong to a colour set. All object net instances have an *identity*, which is build by a net descriptor $D \in \mathbf{D}$ and a natural number $q \in \mathbb{N}$. The *net colour* set $\mathbf{C}(D)$ of a net descriptor D is defined as:

$$\mathbf{C}(D) := \begin{cases} \{(D, 0)\}, & \text{if } D = D_{N_*} \\ \{(D, q) \mid q \in \mathbb{N}\}, & \text{if } D \neq D_{N_*} \end{cases} \tag{2}$$

Let $\mathbf{C} := \bigcup_{D \in \mathbf{D}} \mathbf{C}(D)$ be the union of all net colours. The set $\mathbf{C}(D_{N_*}) = \{(D_{N_*}, 0)\}$ is a singleton. The instance for black tokens $*$ can be expressed as the only element of D_{N_*}:

$$* := (D_{N_*}, 0)$$

Formalising the "nets within nets"-paradigm we have nets as colours of places. For object net systems we have the colour set:

$$\mathcal{C}_{ONS} := \{\mathbf{C}(D_{N_1}), \ldots, \mathbf{C}(D_{N_n}), \mathbf{C}(D_{N_*})\}$$

The places s are mapped by $d : S \to \mathcal{C}_{ONS}$ to one net colour $\mathbf{C}(D)$.

Channels Agent net synchronises via channels. Channels are either for horizontal or for vertical synchronisation. The channel set for horizontal synchronisation is H. The vertical channel set is $V^{\uparrow} \cup V^{\downarrow}$, where $V^{\uparrow} := \{v^{\uparrow} \mid v \in V\}$ and $V^{\downarrow} := \{v^{\downarrow} \mid v \in V\}$ for a set of vertical base channels V. A label set L is called a *channel set* if it is the union of pairwise disjoint sets $H, V^{\uparrow}, V^{\downarrow}$:

$$L = H \cup V^{\uparrow} \cup V^{\downarrow}$$

The labelling function $l : T \to MS(L)$ maps a transition to a multi-set of channels. Using multi-sets of labels allows to describe multiple synchronisation.[4] The neutral element 0 means that no synchronisation is needed.

Definition 5. *Let* $\mathbf{D}$ *be a set of net descriptor generating the colour set* $\mathcal{C}_{ONS}$ *and let* L *be a channel set. An* object net

$$N = (S, T, d, \partial_0, \partial_1, l, m_0)$$

is a labelled, arc-constant coloured Petri net for $\mathcal{C}_{ONS}$*, where* $l : T \to MS(L)$ *is a channel mapping.*

The *primitive object net* N_*, which is the sheme for black tokens, has no places or transitions:

$$N_* := (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$$

Let $\mathcal{N} = \{N_1, \dots N_n\} \cup \{N_*\}$ be a set of object nets, which is in one-to-one correspondence to the descriptor set $\mathbf{D} = \{D_{N_1}, \dots D_{N_n}\} \cup \{D_{N_*}\}$.

In the definition of markings Infinite cycles have to be prevented by an partial order on the object nets.[5]

Definition 6. *An object net system ONS is* grounded *if a partial order $<$ on $\mathcal{N}$ exists such $s \in m_0(N) \wedge d(s) = \mathbf{C}(N')$ implies $N > N'$ for all nets $N, N' \in \mathcal{N}$.*

One net $R \in \mathcal{N}$ – the *root net* – acts a the starting point for the initial state of the system. For uniformity the net R is embedded in an additional place s_0, which must be different from all other net elements. The place s_0 is called the *environment place*. The colour mapping d is extended for the place s_0 by the definition $d(s_0) = \mathbf{C}(D_R)$.

Definition 7. *A* mobile object net system *is the tuple*

$$MONS = (\mathbf{D}, L, \mathcal{N}, N_*, R, s_0)$$

1. $\mathbf{D} = \{D_{N_1}, \dots D_{N_n}\} \cup \{D_{N_*}\}$ *is the set of descriptors, which includes the name of the black token net D_{N_*}.*
2. $L = H \cup V^{\uparrow} \cup V^{\downarrow}$ *is a channel set.*
3. $\mathcal{N} = \{N_1, \dots N_n\} \cup \{N_*\}$ *is the grounded set of object nets, corresponding one-to-one to $\mathbf{D}$. The set of places $S(N)$ and transitions $T(N)$ must be pairwise disjoint for all $N \in \mathcal{N}$.*
4. $R \in \mathcal{N}$ *is the root net.*
5. s_0 *is the environment place with $s_0 \notin T \cup \bigcup_{N \in \mathcal{N}} S(N)$.*

where the MONS has the derived elements:

$$\begin{array}{ll} S := \{s_0\} \cup \bigcup_{N \in \mathcal{N}} S(N) & d := (s_0, \mathbf{C}(D_R)) \cup \bigcup_{N \in \mathcal{N}} d(N) \\ T := \bigcup_{N \in \mathcal{N}} T(N) & \partial_{0,1} := \bigcup_{N \in \mathcal{N}} \partial_{0,1}(N) \\ & l := \bigcup_{N \in \mathcal{N}} l(N) \end{array}$$

For the net domain $\mathcal{N} = \{N\} \cup \{N_*\}$ and $L = \emptyset$ a mobile object net system is equivalent to a normal Petri net.

Example For the introducing example in Figure 1 we have the descriptors: $\mathbf{D} = \{D_{N_1}, D_{N_2}, D_{N_*}\}$. The channel set is $L = \{h, v_1^{\downarrow}, v_1^{\uparrow}, v_2^{\downarrow}, v_2^{\uparrow}\}$. The starting net is N_1 the environment is s_0.

The system net N_1 is defined by its components: $S(N_1) = \{s_1, \dots, s_7\}$, $T(N_1) = \{t_1, \dots, t_5\}$ with $t_1 : (s_1, \xi) \to (s_3, \xi) + (s_4, \xi)$, $t_2 : (s_3, \xi) \to (s_5, \xi)$, $t_3 : (s_4, \xi) \to (s_6, \xi)$, $t_4 : (s_5, \xi) + (s_6, \xi) \to (s_7, \xi)$, and $t_5 : (s_2, \psi) \to (s_7, \psi)$. The

[4] The Renew-tool restricts synchronisation to $|l(t)(h)| = 0, h \in H$ and $|l(t)(v^{\uparrow})| \leq 1, v^{\uparrow} \in V^{\uparrow}$. With this restrictions synchronisation can be be constructed in polynomial time (cf. [Kum98]).

[5] The Renew-tool forbids net tokens in the initial marking. Net tokens have to be constructed at run-time. So, each object net system is obviously grounded.

domain $d(s) = D_{N_2}$ for all $s \in S(N)$. The labelling is $l(t_1) = l(t_4) = l(t_5) = 0$, $l(t_2) = v_1^{\downarrow}$, and $l(t_3) = v_2^{\downarrow}$. The initial marking is $m_0(N_1) = (s_1, \xi) + (s_2, \psi)$.

The object net N_2 is defined by its components: $S(N_2) = \{b_1, \ldots, b_7\}, T(N_2) = \{e_1, e_2, e_3, e_4\}$ with $e_1 : (b_1, *) \rightarrow (b_2, *)$, $e_2 : (b_2, *) + (b_5, *) \rightarrow (b_3, *)$, $e_3 : (b_4, *) \rightarrow (b_5, *)$, and $e_4 : (b_6, *) \rightarrow (b_7, *)$. The domain $d(s) = D_{N_*}$ for all $s \in S(N)$. The labelling is $l(e_1) = v_1^{\uparrow}$, $l(e_3) = v_2^{\uparrow}$, and $l(e_2) = l(e_4) = h$. The initial marking is $m_0(N_2) = (b_1, *) + (b_4, *) + (b_6, *)$. $\diamond$

5 Semantics of Mobile Object Net Systems

In the following the semantics of a mobile object net system $MONS$ is defined formally. The static description of object nets is extended for object net instances. Nets within nets need a formalism that is self reflective – so, it is not natural to use high-level nets for the formalisation of nets within nets.[6] We use concurrent rewriting systems [Mes92] instead, which reflect the algebraic structure of net systems.

5.1 Net Tokens as Instances

Each place $s \in S(N)$ of a net N corresponds to several places $\theta.s \in \mathcal{S}(N)$ in the instances of N. The tokens are instances of object nets generated by the operator $\_.\_ : \mathbf{C} \times S \rightarrow \mathcal{S}$. The set of *instantiated places* $\mathcal{S}$ of a net N is:

$$\mathcal{S}(N) := \{\theta.s \mid \theta \in \mathbf{C}(D_N), s \in S(N)\} \tag{3}$$

The union $\mathcal{S} := \bigcup_{N \in \mathcal{N}} \mathcal{S}(N)$ defines all instantiated places. The notation is lifted to multi-sets: $\theta.(\sum_{i \in I} s_i) = \sum_{i \in I} \theta.s_i$.

In the following nets as tokens are defined. Object net tokens are object net instance systems (ONIS). They are generated by the operator $[\_]\_ : MS(\hat{\mathcal{S}}) \times \mathcal{S} \rightarrow ONIS$. The net tokens generate the marked places $\hat{\mathcal{S}}(N)$

Note, that the definitions are of the markings $\hat{\mathcal{S}}(N)$ and the set object net instances $ONIS(N)$ are defined mutually dependent, due to the recursive structure of the model. Infinite cycles are prevented since the object net system is assumed to be grounded.

The set of *object net instance systems* $ONIS$ is defined as $ONIS(N) := \bigcup_{n=1}^{\infty} ONIS_n(N)$, and the set of *instantiated atomar markings* $\hat{\mathcal{S}}(N)$ is defined as $\hat{\mathcal{S}}(N) := \bigcup_{n=1}^{\infty} \hat{\mathcal{S}}_n(N)$, where:

$$\begin{aligned} ONIS_0(N) &:= \emptyset \\ ONIS_{n+1}(N) &:= \{\, [\theta.m]_\theta \mid \theta.m \in MS(\hat{\mathcal{S}}_{n+1}(N)\} \\ \hat{\mathcal{S}}_{n+1}(N) &:= \{(\theta.s, \Xi) \mid \theta.s \in \mathcal{S}(N), \Xi \in ONIS_n(N'), d(s) = \mathbf{C}(D_{N'})\} \end{aligned} \tag{4}$$

Note, that exept for $m = 0$ the identity θ is redundant in $[\theta.m]_\theta$, so the subskript is usually omitted.

[6] Tokens of high-level nets are data types, while nets are not. So, we have a hierarchy of the concept "net" over the concept "data type". For the paradigm of nets within nets we do not have this hierarchy, since net tokens and data types are objects with the same qualities.

The set of instantiated markings of the environment place s_0 is:[7]

$$\hat{s}_0 := \{(*.s_0, \Xi) \mid \Xi \in ONIS(N), d(s_0) = \mathbf{C}(D_N)\} \tag{5}$$

For the uniformity of the model, the elements of s_0 are considered as special marked places. For the union of all nets we obtain:

$$\hat{\mathcal{S}} = \hat{s}_0 \cup \bigcup_{N \in \mathcal{N}} \hat{\mathcal{S}}(N) \tag{6}$$

5.2 Initial Marking

For an object net system $MONS$ the initial marking m_0 is constructed recursively, starting at the top-level net R. In general the calculation of the initial marking may not terminate due to cyclic dependencies. Let $RN(N)$ be set of reachable nets wrt. the initial marking $m_0(N)$:

$$RN(N) := \{N\} \cup \bigcup_{\substack{s \in m_0(N) \\ d(s) = \mathbf{C}(N')}} RN(N')$$

Cycles can be prevented if the set $RN(R)$ of nets reachable from R can be partially ordered. The initial marking of a mobile object net system $MONS$ can be calculated since the $MONS$ is grounded. The initial marking m_0 is constructed by the definition:

$$m_0 := \text{init}((*.s_0, (D_R, 0)) \tag{7}$$

where the init mapping is lifted to multi sets and defined as:

$$\text{init}((\theta.s, \xi)) := \begin{cases} (\theta.s, [\text{init}(\xi.m_0(N))]), & \text{if } d(s) = \mathbf{C}(D_N), D_N \in \mathbf{D} \setminus \{D_{N_*}\} \\ (\theta.s, \bullet), & \text{if } d(s) = \mathbf{C}(D_{N_*}) \end{cases}$$

Example For the introducing example in Figure 1 N_1 is the starting net located in the environment s_0.

$$m_0 = \text{init}((*.s_0, \theta)) = (*.s_0, [\text{init}(\theta.m_0(N_1))]) \quad \text{where } \theta = (D_{N_1}, 0)$$

The initial marking of N_1 is $m_0(N_1) = (s_1, \xi) + (s_2, \psi)$:

$$\text{init}((\theta.m_0(N_1))) = \text{init}(\theta.((s_1, \xi) + (s_2, \psi))) = \text{init}((\theta.s_1, \xi)) + \text{init}((\theta.s_2, \psi))$$

For $(\theta.s_1, \xi)$ we obtain:

$$\text{init}((\theta.s_1, \xi)) = (\theta.s_1, [\text{init}(\xi.m_0(N_2))])$$

The initial marking of N_2 is $m_0(N_2) = (b_1, *) + (b_4, *) + (b_6, *)$:

$$\begin{aligned} \text{init}(\theta.m_0(N_2)) &= \text{init}((\xi.b_1, *)) + \text{init}((\xi.b_4, *)) + \text{init}((\xi.b_6, *)) \\ &= (\xi.b_1, \bullet) + (\xi.b_4, \bullet) + (\xi.b_6, \bullet) \end{aligned}$$

So, we obtain

$$\text{init}((\theta.s_1, \xi)) = (\theta.s_1, [\,(\xi.b_1, \bullet) + (\xi.b_4, \bullet) + (\xi.b_6, \bullet)\,])$$

[7] To avoid the introduction of an extra instance for s_0 the identity $*$ is used.

Analogously, for ψ:

$$\text{init}((\theta.s_2, \psi)) = (\theta.s_2, [\,(\psi.b_1, \bullet) + (\psi.b_4, *)) + (\psi.b_6, \bullet)\,])$$

All in all the initial marking m_0 is:

$$\begin{aligned} m_0 = (*.s_0, [\ & (\theta.s_1, [\,(\xi.b_1, \bullet) + (\xi.b_4, \bullet) + (\xi.b_6, \bullet)\,]) \\ & + (\theta.s_2, [\,(\psi.b_1, \bullet) + (\psi.b_4, *)) + (\psi.b_6, \bullet)\,])\,]) \end{aligned}$$

◇

5.3 Firing Rule

The objects m of the rewriting system are multi sets of net tokens: $m \in MS(\hat{\mathcal{S}})$. Each transition $t \in T(N)$ generates a family of transition instances $\theta.t$, $\theta \in \mathbf{C}(D_N))$ which describe the semantics of the MONS. For the firing rule of a transition t it is sufficient to consider a partition wrt. to the net colours, since the general firing rule is just the sum of all partitions. Each multi set $m \in MS(\hat{S})$ can be partitionated wrt. the nets ($\mathcal{N}$) and the net colours ($\mathbf{C}$), where $I_{N,\xi_{N,q}}$ is a index set and $\xi_{N,q} = (D_N, q)$:

$$m = \sum_{N \in \mathcal{N}} \sum_{\xi_{N,q} \in \mathbf{C}(D_N)} \sum_{i \in I_{N,\xi_{N,q}}} (s_{N,\xi_{N,q},i}, \xi_{N,q})$$

A transition $t : \sum_{i \in I}(s_i, \xi_i) \to \sum_{j \in J}(s'_j, \xi'_j)$ can be partionated the same way:

$$\begin{aligned} & t : \textstyle\sum_{N \in \mathcal{N}} \sum_{\xi_{N,q} \in \mathbf{C}(D_N)} \sum_{i \in I_{N,\xi_{N,q}}} (s_{N,\xi_{N,q},i}, \xi_{N,q}) \\ & \to \textstyle\sum_{N \in \mathcal{N}} \sum_{\xi_{N,q} \in \mathbf{C}(D_N)} \sum_{j \in J_{N,\xi_{N,q}}} (s'_{N,\xi_{N,q},j}, \xi_{N,q}) \end{aligned} \tag{8}$$

For such a partition the *transition part* $t_{|\xi_{N,q}}$ wrt. one net instance $\xi_{N,q}$ is:

$$t_{|\xi_{N,q}} : \sum_{i \in I_{N,\xi_{N,q}}} (s_{N,\xi_{N,q},i}, \xi_{N,q}) \to \sum_{j \in J_{N,\xi_{N,q}}} (s'_{N,\xi_{N,q},j}, \xi_{N,q}) \tag{9}$$

Transition parts describe the behaviour for one net token $\xi_{N,q}$. Each transition part $t_{|\xi_{N,q}}$ is mapped to the family of instantiated transitions $\theta.t_{|\xi_{N,q}}$. The transition t itself is mapped to the sum of all instantiated transition parts:

$$\boxed{\theta.t : \textstyle\sum_{N \in \mathcal{N}} \sum_{\xi_{N,q} \in \mathbf{C}(D_N)} \partial_0(\theta.t_{|\xi_{N,q}}) \to \sum_{N \in \mathcal{N}} \sum_{\xi_{N,q} \in \mathbf{C}(D_N)} \partial_1(\theta.t_{|\xi_{N,q}})} \tag{10}$$

The synchronisation info remains the same for all instances: $l(\theta.t) := l(t)$.

First we define the basic rules for transitions t and second the set of synchronised transitions. Two cases have two be considered: Transitions with empty domain, which create new net instances, and transitions with a non empty domain.

Creation New nets are created by the transition $t_{|\xi_{N,q}}$ if $N \neq N_*$ and $t_{|\xi_{N,q}}$ has an empty domain ($|\partial_0(t_{|\xi_{N,q}})| = 0$) and a non empty codomain ($|\partial_1(t_{|\xi_{N,q}})| > 0$):

$$\boxed{\begin{array}{c} t \in T(N') \qquad \theta \in \mathbf{C}(D_{N'}) \qquad |J_{N,\xi_{N,q}}| > 0 \\ (\theta.s'_j, \Xi) = \mathrm{init}(\theta.s'_{N,\xi_{N,q},j}, \xi_{N,q}) \\ t_{|\xi_{N,q}} : 0 \to \sum_{j \in J_{N,\xi_{N,q}}} (s'_{N,\xi_{N,q},j}, \xi_{N,q}) \\ \hline \theta.t_{|\xi_{N,q}} : 0 \to \sum_{j \in J_{N,\xi_{N,q}}} (\theta.s'_{N,\xi_{N,q},j}, \Xi) \end{array}} \tag{11}$$

Since value semantics is considered, each place $\theta.s'_j$ is marked with its own net token Ξ. For the transition $t_{|\xi_{N,q}}$ where $N = N_*$ only black tokens $\bullet$ are created. The creation of net tokens is well defined since init cannot be invoked in a cyclic way due to the groundedness of the object net system.

Autonomous Action For all non-creation cases ($|\partial_0(t_{|\xi_{N,q}})| > 0$ and $|\partial_1(t_{|\xi_{N,q}})| \geq 0$) existing net tokens are moved around. This also includes the deletion of net tokens. If one net tokens is "sent" to many places the marking is distributed to the copies. This is expressed by the family of predicates $\mathit{distr}_n, n \in \mathbb{N}, n > 0$:

$$\mathit{distr}_n(\xi.m_1, \ldots, \xi.m_n, \xi.m) \iff \sum_{i=1}^{n} m_i = m$$

For $n = 0$ the predicate $\mathit{distr}_0(\xi.m)$ is always true. This special case is used for destruction of net tokens.

Let ξ be one net instance of an transition part. All in-going net instances $[\xi.m_i]$ are combined to $[\xi.m]$, so $\mathit{distr}_{|I_{N,\xi_{N,q}}|}(\xi.m_1, \ldots, \xi.m_n, \xi.m)$ must hold. All out-going instances $[\xi.m'_j]$ share the marking $[\xi.m]$, so analogously the predicate $\mathit{distr}_{|J_{N,\xi_{N,q}}|}(\xi.m'_1, \ldots, \xi.m'_{n'}, \xi.m)$ must hold. The rule for autonomous firing is:

$$\boxed{\begin{array}{c} t \in T(N') \qquad \theta \in \mathbf{C}(D_{N'}) \qquad |\partial_0(t_{|\xi_{N,q}})| > 0 \\ n = |I_{N,\xi_{N,q}}| \qquad \mathit{distr}_n(\xi_{N,q}.m_1, \ldots, \xi_{N,q}.m_n, \xi_{N,q}.m) \\ n' = |J_{N,\xi_{N,q}}| \qquad \mathit{distr}_{n'}(\xi_{N,q}.m'_1, \ldots, \xi_{N,q}.m'_{n'}, \xi_{N,q}.m) \\ t_{|\xi_{N,q}} : \qquad \sum_{i \in I_{N,\xi_{N,q}}} (s_{N,\xi_{N,q},i}, \xi_{N,q}) \to \sum_{j \in J_{N,\xi_{N,q}}} (s'_{N,\xi_{N,q},j}, \xi_{N,q}) \\ \hline \theta.t_{|\xi_{N,q}} : \sum_{i \in I_{N,\xi_{N,q}}} (\theta.s_{N,\xi_{N,q},i}, [\xi_{N,q}.m_i]) \to \sum_{j \in J_{N,\xi_{N,q}}} (\theta.s'_{N,\xi_{N,q},j}, [\xi_{N,q}.m'_j]) \end{array}} \tag{12}$$

This rule is quite general, since it allows any assignment of markings to the copies. A restriction to a special form of distribution can be formulated by adding a guard to the transition t.

The sum of all transition parts (10) allows the simultaneous creation, deletion and movement of different net tokens in one single firing step. The closure of T wrt. the rules (11), and (12) is denoted as $BVAL(T)$, the basic transitions of value semantics.

Synchronous Action The transitions of $BVAL(T)$ still have the need for synchronisation. A transition $\theta.t \in BVAL(T)$ with $l(t) \in H$ requests a horizontal synchronisation, a transition t with $l(\theta.t) \in V^{\downarrow} \cup V^{\uparrow}$ requests a vertical synchronisation and a transition t with $l(\theta.t) = 0$ is free of synchronisation.

Horizontal synchronisation Horizontal synchronisation of two transitions $\theta_1.t_1$ and $\theta_2.t_2$ is possible if the labels match: $h \in l(\theta_1.t_1)$, $h \in l(\theta_2.t_2)$ for $h \in H$, and if the two object net instances θ_1 and θ_2 are placed on the same surrounding place $\theta.s$.

$$\frac{\begin{array}{c}\theta_1.t_1, \theta_2.t_2 \in BVAL(T) \qquad \theta.s \in \mathcal{S} \\ h \in l(\theta_1.t_1) \quad h \in l(\theta_2.t_2) \quad h \in H\end{array}}{\begin{array}{r}(\theta_1.t_1||_h\theta_2.t_2) : (\theta.s, [\theta_1.m_1 + \partial_0(\theta_1.t_1)]) + (\theta.s, [\theta_2.m_2 + \partial_0(\theta_2.t_2)]) \\ \rightarrow (\theta.s, [\theta_1.m_1 + \partial_1(\theta_1.t_1)]) + (\theta.s, [\theta_2.m_2 + \partial_1(\theta_2.t_2)])\end{array}} \tag{13}$$

The transition $(\theta_1.t_1||_h\theta_2.t_2)$ does not need synchronisation on the channel h, so h is subtracted:

$$l(\theta_1.t_1||_h\theta_2.t_2) := l(\theta_1.t_1) - h + l(\theta_2.t_2) - h$$

Vertical synchronisation In the case of vertical synchronisation one transition $\theta_1.t_1$ can make a call to the object net θ_2, if θ_2 marks the domain of $\theta_1.t_1$: $|\partial_0(\theta_1.t_{1|\theta_2})| > 0$ (line 2 and 3). The domain $\partial_0(\theta_2.t_2)$ of the object net transitions has to be distributed over the in-coming net tokens and the codomain $\partial_1(\theta_2.t_2)$ is distributed over the out-going ones (line 4 and 5). Also the labels of $\theta_1.t_1$ and $\theta_2.t_2$ must match: $v^{\downarrow} \in l(\theta_1.t_1)$ and $v^{\uparrow} \in l(\theta_2.t_2)$ (line 6).

$$\frac{\begin{array}{c}\theta_1.t_1, \theta_2.t_2 \in BVAL(T) \\ \theta_1.t_{1|\theta_2} : \sum_{i\in I}(\theta_1.s_i, [\theta_2.m_i]) \rightarrow \sum_{j\in J}(\theta_1.s'_j, [\theta_2.m'_j]) \quad |I| > 0 \\ \theta_1.t_1 : A + \partial_0(\theta_1.t_{1|\theta_2}) \rightarrow A' + \partial_1(\theta_1.t_{1|\theta_2}) \\ distr_{|I|}(\theta_2.B_1, \ldots, \theta_2.B_{|I|}, \partial_0(\theta_2.t_2)) \\ distr_{|J|}(\theta_2.B'_1, \ldots, \theta_2.B'_{|J|}, \partial_1(\theta_2.t_2)) \\ v^{\downarrow} \in l(\theta_1.t_1) \quad v^{\uparrow} \in l(\theta_2.t_2) \quad v \in V\end{array}}{\begin{array}{r}(\theta_1.t_1 /\!\!/_v \theta_2.t_2) : A + \sum_{i\in I}(\theta_1.s_i, [\theta_2.m_i + \theta_2.B_i]) \\ \rightarrow A' + \sum_{j\in J}(\theta_1.s'_j, [\theta_2.m'_j + \theta_2.B'_j])\end{array}} \tag{14}$$

Also the transition $(\theta_1.t_1 /\!\!/_v \theta_2.t_2)$ does not need synchronisation on v:

$$l(\theta_1.t_1 /\!\!/_v \theta_2.t_2) := l(\theta_1.t_1) - v^{\downarrow} + l(\theta_2.t_2) - v^{\uparrow}$$

The rules (13) and (14) generated the set $SVAL(T)$ of partially synchronised transitions. Transitions are rewriting rules in the semantics of the object net system if they are free of synchronisation. The set of fully synchronised transitions $VAL(T)$ is:

$$VAL(T) := \{t \in SVAL(T) \mid l(t) = 0\} \tag{15}$$

Example For the introducing example in Figure 1 the initial marking m_0 is given as $m_0 = (*.s_0, [\,(\theta.s_1, \Xi_0) + (\theta.s_2, \Psi_0)\,])$ where $\Xi_0 := [\,(\xi.b_1, \bullet) + (\xi.b_4, \bullet) + (\xi.b_6, \bullet)\,]$ and $\Psi_0 := [\,(\psi.b_1, \bullet) + (\psi.b_4, \bullet) + (\psi.b_6, \bullet)\,]$.

One transition instance of t_1 created by (11) is:

$$\begin{array}{l}\theta.t_1 : (\theta.s_1, [\,(\xi.b_1, \bullet) + (\xi.b_4, \bullet) + (\xi.b_6, \bullet)\,]) \\ \quad \rightarrow (\theta.s_3, [\,(\xi.b_1, \bullet)\,]) + (\theta.s_4, [\,(\xi.b_4, \bullet) + (\xi.b_6, \bullet)\,])\end{array}$$

One instance of t_5 is:

$$\theta.t_5 : (\theta.s_2, \Psi_0) \rightarrow (\theta.s_7, \Psi_0)$$

The concurrent firing $\theta.t_1 \oplus \theta.t_5$ rewrites m_0 to the marking:

$$m_1 = (*.s_0, [\,(\theta.s_3, [\,(\xi.b_1, \bullet)\,]) + (\theta.s_4, [\,(\xi.b_4, \bullet) + (\xi.b_6, \bullet)\,]) + (\theta.s_7, \Psi_0)\,])$$

The vertical synchronisation of $\theta.t_2 : (\theta.s_3, [(\psi.b_1, \bullet)]) \rightarrow (\theta.s_5, [(\psi.b_1, \bullet)])$ and $\xi.e_1 : (\xi.e_1, \bullet) \rightarrow (\xi.e_2, \bullet)$ over the channel v_1 is expressed by the rule:

$$(\theta.t_2 /\!\!/_{v_1} \xi.e_1) : (\theta.s_3, [\,(\psi.b_1, \bullet)\,]) \rightarrow (\theta.s_5, [\,(\psi.b_2, \bullet)\,])$$

Analogous the vertical synchronisation of $\theta.t_3 : (\theta.s_3, [(\psi.b_4, \bullet) + (\psi.b_6, \bullet)] \rightarrow (\theta.s_4, [(\psi.b_4, \bullet) + (\psi.b_6, \bullet)])$ and $\xi.e_3 : (\xi.e_4, \bullet) \rightarrow (\xi.e_5, \bullet)$ over the channel v_2:

$$(\theta.t_3 /\!\!/_{v_2} \xi.e_3) : (\theta.s_4, [\,(\psi.b_4, \bullet) + (\psi.b_6, \bullet)\,]) \rightarrow (\theta.s_6, [\,(\psi.b_5, \bullet) + (\psi.b_6, \bullet)\,])$$

The concurrent rewrite $(\theta.t_2 /\!\!/_{v_1} \xi.e_1) \oplus (\theta.t_3 /\!\!/_{v_2} \xi.e_3)$ on m_1 leads to the marking:

$$m_2 = (*.s_0, [\,(\theta.s_5, [\,(\xi.b_2, \bullet)\,]) + (\theta.s_6, [\,(\xi.b_5, \bullet) + (\xi.b_6, \bullet)\,]) + (\theta.s_7, \Psi_0)\,])$$

The two copies of Ξ_0 are recombined by the rule for t_4:

$$\begin{aligned} &\theta.t_4 : (\theta.s_5, [\,(\xi.b_2, \bullet)\,]) + (\theta.s_6, [\,(\xi.b_5, \bullet) + (\xi.b_6, \bullet)\,]) \\ &\quad \rightarrow (\theta.s_7, [\,(\xi.b_2, \bullet) + (\xi.b_5, \bullet) + (\xi.b_6, \bullet)\,]) \end{aligned}$$

A rewrite result in the marking:

$$m_3 = (*.s_0, [\,(\theta.s_7, [\,(\xi.b_2, \bullet) + (\xi.b_5, \bullet) + (\xi.b_6, \bullet)\,]) + (\theta.s_7, \Psi_0)\,])$$

In marking m_3 the two net tokens can synchronise horizontally in the place $\theta.s_7$ by the synchronous action of $\xi.e_2 : (\xi.b_2, \bullet) + (\xi.b_5, \bullet) \rightarrow (\xi.b_3, \bullet)$ and $\psi.e_4 : (\psi.b_6, \bullet) \rightarrow (\psi.b_7, \bullet)$ over the channel h:

$$\begin{aligned} (\xi.t_1 ||_h \psi.t_2) : &(\theta.s_7, [\,(\xi.b_2, \bullet) + (\xi.b_5, \bullet) + (\xi.b_6, \bullet)\,]) \\ &+ (\theta.s_7, [\,(\psi.b_1, \bullet) + (\psi.b_4, \bullet) + (\psi.b_6, \bullet)\,]) \\ \rightarrow &(\theta.s_7, [\,(\xi.b_3, \bullet) + (\xi.b_6, \bullet)\,]) + (\theta.s_7, [\,(\psi.b_1, \bullet) + (\psi.b_4, \bullet) + (\psi.b_7, \bullet)\,]) \end{aligned}$$

Applied to m_3 this leads to the marking:

$$m_4 = (*.s_0, [\,(\theta.s_7, [\,(\xi.b_3, \bullet) + (\xi.b_6, \bullet)\,]) + (\theta.s_7, [\,(\psi.b_1, \bullet) + (\psi.b_4, \bullet) + (\psi.b_7, \bullet)\,])\,])$$

$\diamond$

5.4 The Semantics of Mobile Object Net Systems

The semantics of the of mobile object net systems can be expresses by the the set of marked places $\hat{\mathcal{S}}$ and the transition set $VAL(T)$. So, $U(VAL(T))$ – according to definition 2 – ist the morphism structure of the category $\mathcal{MONS}$, which describes the behaviour of object net system under the value semantics. The elements of $\mathcal{MONS}$ are multi-set markings of net tokens $MS(\hat{\mathcal{S}})$.

Definition 8. *Let $MONS = (\mathbf{D}, L, \mathcal{N}, N_*, R, s_0)$ be an mobile object net system wrt. definition 7. The semantics of $MONS$ is defined as the category:*

$$\mathcal{MONS} = (MS(\hat{\mathcal{S}}), U(VAL(T)), \partial_0, \partial_1, ;, id_m)$$

The initial state m_0 *by the initialisation of the starting net R – assuming that $MONS$ is grounded:*

$$m_0 = \mathrm{init}((*.s_0, (D_R, 0)))$$

5.5 Value vs. Reference

The rule for autonomous firing (12) includes a special case, which relates the value semantics with the reference semantics as in [KR02a]. A quite interesting distribution of $[\xi.m]$ to $[\xi.m'_1], \ldots, [\xi.m'_{|J_{N,\xi_{N,q}}|}]$ is the one where $m'_1 = m$ and $m'_2 = \cdots = m'_{n'} = 0$. The marking $[0]_\xi$ naturally describes a *reference* to the net token, since the *name* ξ is distributed, but not the *marking*. Since the net token carries the empty marking no synchronisation can take place with the token $[0]_\xi$.

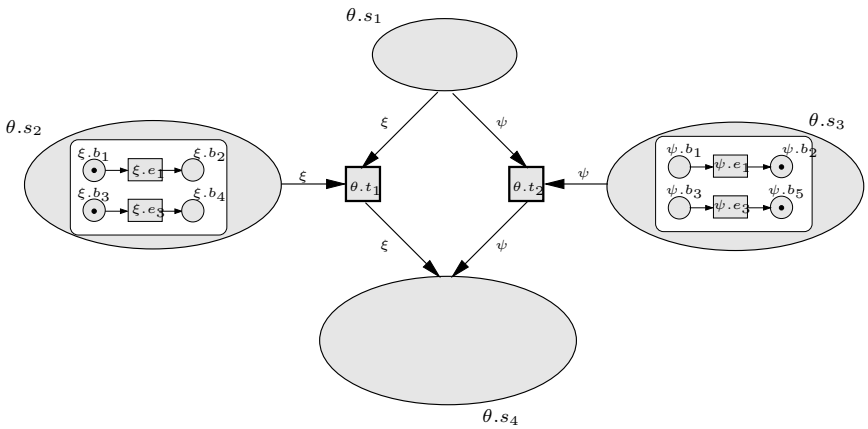

Fig. 2. A conflict

Nevertheless the knowledge of the name is desirable, since $[0]_\xi$ can be used as a selector. Figure 2 shows an conflict between the transitions $\theta.t_1$ and $\theta.t_2$ which is resolved by the marking on $\theta.s_1$. Assume that $\theta.s_1$ is marked only with a selector, either $[0]_\xi$ or $[0]_\psi$ while the "real" net tokens $\Xi = [m]_\xi$ and $\Psi = [m']_\psi$ are placed on $\theta.s_2$ and $\theta.s_3$. The concrete marking of $\theta.s_1$ then resolves the conflict and places the corresponding "real" net token – either Ξ or Ψ – on place $\theta.s_4$. This selection mechanism has been possible by the firing rule (12) which combines $[0]_\xi$ and $[m]_\xi$ to $[m]_\xi$, which is again the "real" net token.

The distinction between the aggregation of an object and the reference towards it is a common design pattern in software engineering (cf. [WMB99]). The formalism of MONS show that in general aggregation and reference are no antithetic concepts. Aggregation is the basic concept, which occurs at different granularity. Aggregation forms a continuum, which is expressed by the predicate $distr_n(\xi.m_1, \ldots, \xi.m_n, \xi.m)$. Reference is just a special case of aggregation.

6 Relationship to Nets within Nets

The model of mobile object net systems is a generalisation of the basic model of "unary elementary object net system" (UEOS) of Valk [Val91,Val98,Val00a]. The model of unary elementary object net system is the simplest "nets within nets" formalism, so it is well suited to act as an starting point for the analysis of the relationship of the reference and the value oriented semantics (cf. [Val00b]). The model of UEOS can be seen as a restricted form of mobile object net systems:

1. Exactly three net types are present: the system net N_S, the object net N_O, and the black token net N_*, so $\mathbf{D} = \{N_S, N_O, N_*\}$. The starting net is N_S.
2. The object nets N_S and N_O are elementary nets [Thi87], so places are safe and the domain and codomain maps to sets rather to multi sets: $|\partial_{0,1}(t)(x)| \leq 1$. Places of the system net N_S contain net tokens derived from the object net N_O, places of the object net N_O contain black tokens: $d(s) = \mathbf{C}(D_{N_O})$ for all $s \in S(N_S)$ and $d(s) = \mathbf{C}(D_{N_*})$ for all $s \in S(N_O)$.
3. The UEOS is grounded since all nets are implicitly ordered $N_S > N_O > N_*$ – this is also expressed by the terms "system net" and "object net".
4. Creation of net tokens by a creation transition (11) is forbidden. Initially, there is exactly one object net token: $|m_0(N_S)| = 1$. Since creation is impossible this net token is the only instance, so for the partition (8) there is at most one non-empty set $I_{N,\xi_{N,q}}$ and one non-empty set $J_{N,\xi_{N,q}}$ – the marking and the transitions are *unary*.
5. The label set is restricted to vertical synchronisation for a UEOS: $H = \emptyset$, since there is at most one net token on each place. Synchronisation is formalised by an relation $\rho \subseteq T(N_S) \times T(N_O)$. Let $t\rho\cdot := \{e \mid (t,e) \in \rho\}$ and $\cdot\rho e := \{t \mid (t,e) \in \rho\}$. A transition $t \in T(N_S)$ must synchronise with one member of $t\rho\cdot$ if $t\rho\cdot \neq \emptyset$, a transition $e \in T(N_O)$ must synchronise with a member of $\cdot\rho e$ if $\cdot\rho e \neq$.
   So, synchronisation of t generally describes a synchronisation conflict between the transitions $t\rho\cdot = \{e_1, \ldots, e_n\}$. By replacing each t with the set $\{t_{e_1}, \ldots, t_{e_n}\}$ where $\partial_0(t_{e_i}) = \partial_0(t)$, $\partial_1(t_{e_i}) = \partial_1(t)$, and $(t_{e_i}, e_i) \in \rho$ the conflict is expressed by the conflict between the t_{e_i} and we obtain the normal form where $|t\rho \cdot| \leq 1$ holds. Analogous for $\cdot\rho e$. This normal form describes the set of vertical channels: $V = \rho$.
6. For a UEOS tokens are considered as pure values, so the net marking is copied directly, while for mobile object net systems the marking is shared among the copies. So, for a UEOS different copies can resolve conflicts in different ways. This inconsistent evolution of the distributed object net tokens raises the need for the inspection of object net processes [Val98,Val00a]. Net processes contain the information to detect inconsistent evolution of copies in the systems.
   The inconsistency is prevented in MONS since for object net copies the marking is shared and not copied.

7 Conclusion

In this article we have presented the formalism of mobile object net system. Mobile object net system follow the paradigm of nets within nets. From a theoretical point of view the characterisation of the semantics of mobile object net systems as a monoidal category leads to insights to the interplay of locality and concurrency. The formalisation is fundamentally based on both of these concepts. As a main result we achieved a description of concurrency in object net system the same way as for ordinary Petri nets by the $\oplus$ operator. This has been made possible by the analysis of the algebraic nature of net tokens, developed in this contribution.

In forthcoming work the formal characterisation of nets within nets of the distribution relation $distr_n$ has to be investigated: The model of mobile object nets can be considered as the general framework which is customised for each the application field. Each field defines distribution constraints for the system net and/or the object net.

This work is integrated within the general research on the semantics of distributed mobile entities. Nets within nets are an attractive approach for an intuitive representation of mobile entities (cf. [KR02b]). Models of mobile systems can be implemented using our tool Renew [KWD03]. The formal results of this analysis can directly be integrated in the design of the Petri net based multi agent system architecture Mulan [KMR01], designed with Renew. Mobility of Mulan-agents are expressed by a special constraint mode the general framework of mobile object nets.

References

[AB96] Andrea Asperti and Nadia Busi. Mobile Petri nets. Technical report, Department of computer science, University of Bologna, TR UBLCS-96-10, 1996.

[BW90] Michael Barr and Charles Wells. *Category theory for computing science*. Prentice Hall, 1990.

[CGG99] Luca Cardelli, Andrew D. Gordon, and Giorgio 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.

[Far99] Berndt Farwer. A linear logic view of object Petri nets. *Fundamenta Informaticae*, 37(3):225–246, 1999.

[Far00] Berndt Farwer. *Linear Logic Based Calculi for Object Petri Nets*. Logos Verlag, Berlin, 2000.

[KMR01] Michael Köhler, Daniel Moldt, and Heiko Rölke. Modeling the behaviour of Petri net agents. In J. M. Colom and M. Koutny, editors, *Proceedings of the 22st Conference on Application and Theory of Petri Nets*, volume 2075 of *LNCS*, pages 224–241. Springer-Verlag, June 2001.

[KR02a] Michael Köhler and Heiko Rölke. Mobile object net systems: Concurrency and mobility. In H.-D. Burkhard, L. Czaja, G. Lindemann, A. Skowron, and P. Starke, editors, *Proceedings of the International Workshop on Concurrency, Specification, and Programming (CS&P 2002)*, 2002.

[KR02b] Michael Köhler and Heiko Rölke. Modelling mobility and mobile agents using nets within nets. In D. Moldt, editor, *Proceedings of the Workshop on Modelling of Object, Components, and Agents (MOCA'02)*. Computer Science Department, Aarhus University, 2002.

[Kum98] Olaf Kummer. Simulating synchronous channels and net instances. In J. Desel, P. Kemper, E. Kindler, and A. Oberweis, editors, *Forschungsbericht Nr. 694: 5. Workshop Algorithmen und Werkzeuge für Petrinetze*, pages 73–78. Universität Dortmund, Fachbereich Informatik, 1998.

[Kum01] Olaf Kummer. Introduction to Petri nets and reference nets. *Sozionik-aktuell*, 1, 2001.

[KWD03] Olaf Kummer, Frank Wienberg, and Michael Duvigneau. *The Reference Net Workshop (Renew)*. Universität Hamburg, 1998-2003. http://www.renew.de.

[Lom99] Irina A. Lomazova. Nested Petri nets – a formalism for specification of multi-agent distributed systems. In Burkhard, H.-D., Czaja, L., Nguyen, H.-S., and Starke, P., editors, *Proceedings of the CSP'99 Workshop, Warsaw, 28-30 September 1999*, pages 127–140, 1999.

[Mes92] José Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96:73–155, 1992.

[ML71] Saunders Mac Lane. *Categories for the working mathematician*. Springer-Verlag, 1971.

[MM90] José Meseguer and Ugo Montanari. Petri nets are monoids. *Information and Computation*, 88(2):105–155, October 1990.

[MMS97] José Meseguer, Ugo Montanari, and Vladimiro Sassone. On the semantics of place/transition Petri nets. *Mathematical Structures in Computer Science*, 7(4):359–397, 1997.

[MPW92] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, parts 1-2. *Information and computation*, 100(1):1–77, 1992.

[SMÖ01] 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.

[Thi87] P. Thiagarajan. Elementary net systems. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Central Models and Their Properties*, volume 254 of *Lecture Notes in Computer Science*, pages 26–59. Springer-Verlag, 1987.

[Val91] Rüdiger Valk. Modelling concurrency by task/flow EN systems. In *Proceedings 3rd Workshop on Concurrency and Compositionality*, number 19 in GMD-Studien, St. Augustin, Bonn, 1991. Gesellschaft für Mathematik und Datenverarbeitung.

[Val98] Rüdiger Valk. Petri nets as token objects: An introduction to elementary object nets. In Jörg Desel and Manuel Silva, editors, *Application and Theory of Petri Nets*, volume 1420 of *Lecture Notes in Computer Science*, pages 1–25, 1998.

[Val00a] Rüdiger Valk. Concurrency in communicating object Petri nets. In G. Agha, F. De Cindio, and G. Rozenberg, editors, *Concurrent Object-Oriented Programming and Petri Nets*, volume 2001 of *Lecture Notes in Computer Science*. Springer-Verlag, 2000.

[Val00b] Rüdiger Valk. Relating different semantics for object Petri nets. Technical Report FBI-HH-B-226/00, Universität Hamburg, Fachbereich Informatik, 2000.

[VC98] Jan Vitek and Giuseppe Castagna. Seal: A framework for secure mobile computations. In *ICCL Workshop: Internet Programming Languages*, pages 47–77, 1998.

[WMB99] Axel Wienberg, Florian Matthes, and Marko Boger. Modeling dynamic software components with UML. In Robert France and Bernhard Rumpe, editors, *UML'99 - The Unified Modeling Language. Beyond the Standard. Second International Conference, Fort Collins, CO, USA, October 28-30. 1999, Proceedings*, volume 1723 of *LNCS*, pages 204–219. Springer, 1999.