Electronic Communications of the EASST
Volume 40 (2011)

Proceedings of the
4th International Workshop on Petri Nets and Graph Transformation
(PNGT 2010)

Functorial Analysis of Algebraic Higher-Order Net Systems with Applications to Mobile Ad-Hoc Networks

Ulrike Golas, Kathrin Hoffmann, Hartmut Ehrig,
Alexander Rein and Julia Padberg

20 pages

Guest Editors: Claudia Ermel, Kathrin Hoffmann
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Functorial Analysis of Algebraic Higher-Order Net Systems with Applications to Mobile Ad-Hoc Networks*

Ulrike Golas[1], Kathrin Hoffmann[23], Hartmut Ehrig[2], Alexander Rein[2] and Julia Padberg[3]

[1] golas@zib.de
Konrad-Zuse-Zentrum für Informationstechnik Berlin, Germany
[2] hoffmann|ehrig|alex@cs.tu-berlin.de
Technische Universität Berlin, Germany
[3] padberg@informatik.haw-hamburg.de
Hochschule für Angewandte Wissenschaften Hamburg, Germany

Abstract: Algebraic higher-order (AHO) net systems are Petri nets with place/transition systems, i.e. place/transition nets with initial markings, and rules as tokens. In several applications, however, there is the need for explicit data modeling. The main idea of this paper is to introduce AHO net systems with high-level net systems and corresponding rules as tokens. We relate them to AHO net systems with low-level net systems as tokens and analyze the firing and transformation properties of the corresponding net class transformation defined as functors between the corresponding categories of AHO net systems.

All concepts and results are explained with an example in the application area of mobile ad-hoc networks. From an abstract point of view, mobile ad-hoc networks consist of mobile nodes which communicate with each other independent of a stable infrastructure, while the topology of the network constantly changes depending on the current position of the nodes and their availability. To ensure satisfactory team cooperation in workflows of mobile ad-hoc networks we use the modeling technique of AHO net systems.

Keywords: algebraic higher-order nets, mobile ad-hoc networks, skeleton functor

1 Introduction and Related Work

Place/transition (P/T) systems and their variants are an established process definition language for the representation, validation and verification of workflow procedures (see, e. g., [vdA03] for an overview), where P/T nets represent process schemes and P/T systems describe the behavior of process instances due to their initial markings. The paradigm of *nets as tokens* has been introduced by Valk [Val98] where so-called object nets are token within another net, called a system net. In elementary object systems, object nets can move through a system net and interact

* This work has been partly funded by the research project forMA$_\ell$NET (see http://tfs.cs.tu-berlin.de/formalnet/) of the German Research Council.

with both the system net and other object nets. This allows to change the marking of the object nets, but not their net structure.

In [HME05, EHP+07], the concept of reconfigurable place/transition net systems has been introduced which is most important to model changes of the net structure while the system is kept running. In detail, a reconfigurable P/T net system consists of a P/T net with marking and a set of rules. In these nets, not only the follower marking can be computed but also the structure can be changed by rule application to obtain a new P/T net system that is more appropriate with respect to some requirements of the environment. For rule-based transformations of P/T net systems we use the framework of net transformations [EEPT06] following the double-pushout (DPO) approach of graph transformation systems and the notation of Petri nets as monoids [MM90]. The basic idea behind net transformations is the stepwise development of P/T net systems by given rules. One may think of these rules as replacement systems where the left-hand side is replaced by the right-hand side while preserving a context.

In low-level Petri nets, the tokens are indistinguishable. The integration of Petri nets with data type descriptions has led to high-level nets as powerful specification techniques like algebraic high-level (AHL) nets [PER95], which can also be transformed within the DPO approach [Pra08]. In [HME05], we have introduced the paradigm nets and rules as tokens by a high-level model with a suitable data type part. The model called algebraic higher-order (AHO) net system exploits some form of control not only on the rule application but also on token firing. An AHO net system is defined by an AHL net system with net places and rule places, where the marking is given by suitable low-level net systems or rules, respectively, on these places. As shown in [BRHM06], this paradigm has been very useful to model applications in the area of mobile ad-hoc networks.

Mobile ad-hoc networks (MANETs) consist of mobile nodes forwarding data to other nodes based on the network connectivity independent from a stable infrastructure. The constant change of the network's topology depends on the current position of the nodes and their availability. A typical example of a complex application is a team communicating using hand-held devices and laptops as in emergency scenarios [PHE+07]. In such a scenario, each team member performs specific activities while different teams collaborate through the interleaving of all the different workflows. Normally, workflows in mobile environments are not fixed once and for all at design time but are constantly adapted at run time predicting disconnections or reorganizing activities. This requires on the one hand a suitable description of the distributed workflows and on the other hand expressive techniques for the adaption.

Research on MANETs [AZ03] has focused mainly on the infrastructure at the lower levels of the ISO/OSI-standards. To apply MANETs in larger operations it is necessary to abstract from the network layer. In [RMPM03], an interface for network services that can be used by applications abstracting from the underlying protocols is suggested. In contrast to approaches using models mainly for the network we propose modeling the application in terms of workflows, such that the adaption of workflows to accommodate the requirements in an ad-hoc setting are met. Our experience with the case study in [PEH07] has clearly shown the need to integrate data on the level of workflows. The main idea of this paper is to introduce AHO net systems with high-level net systems and corresponding rules as tokens. We relate them to AHO net systems with low-level net systems and rules as tokens, and analyze the firing and transformation properties of the corresponding net class transformation defined as functor between corresponding categories of AHO net systems. This functor is based on the corresponding functor from high-level to low-level nets [Urb03]. All concepts and results are explained with an example in the area of mobile ad-hoc networks. In contrast to [PEH07], where we have used merely low-level net systems, we present now a pipeline emergency scenario where we use data dependent workflows.

This paper is organized as follows: In Section 2, we introduce as motivating example a pipeline emergency scenario using data dependent workflows. In Section 3, we define AHO net systems with AHL net systems and rules as tokens as well as their firing behavior and transformations. In Section 4, we present the skeleton functor transforming AHO net systems with high-level tokens to such with low-level tokens and show that enabling and firing is preserved. Finally, in Section 5, we give a conclusion and an outlook to future work.

2 Motivation: Emergency Scenario

In emergency scenarios, we can obtain an effective coordination among team members constituting a mobile ad-hoc network through the use of net system and rule tokens. In this way,

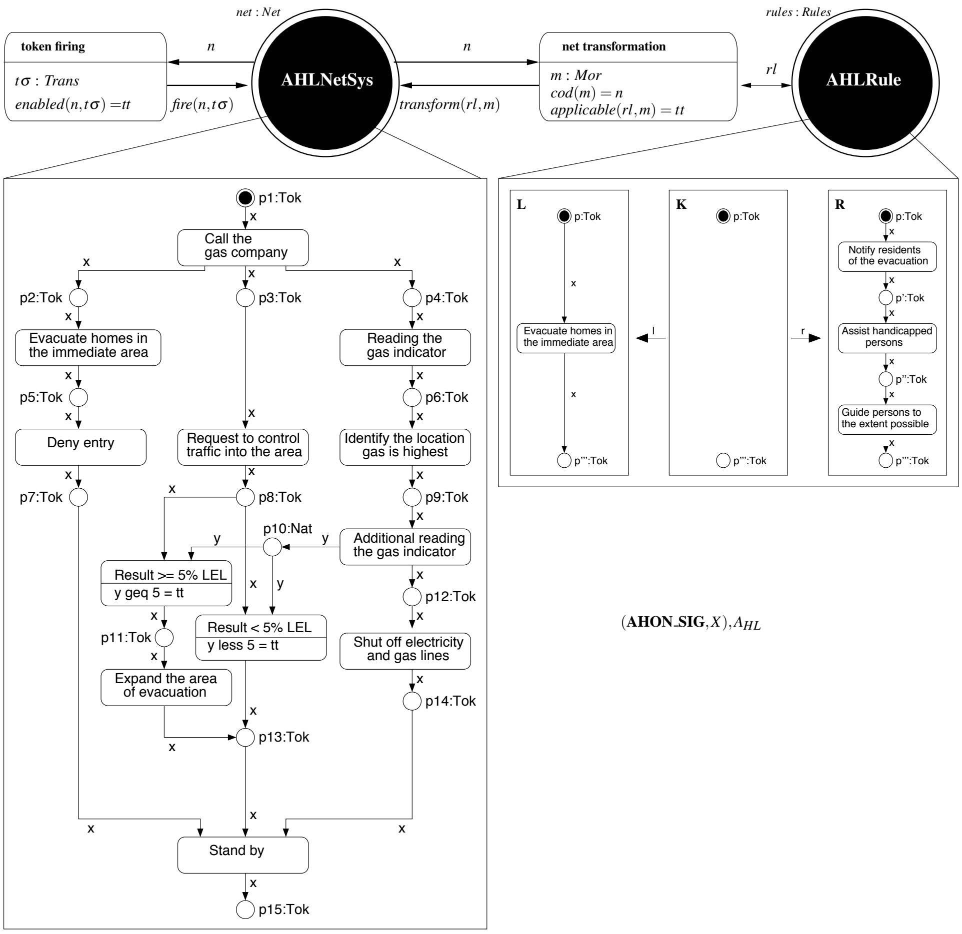

Figure 1: Algebraic higher-order net system for emergency scenario

cooperative work can be adequately modeled by high-level Petri nets with initial markings. The net structure can be adapted to new requirements of the environment during run time by a set of rules, i.e. token firing and net transformation can be interleaved with each other. In contrast to [PEH07], where we have used merely low-level net systems, we present a pipeline emergency scenario based on data dependent workflows. This means that we use AHL net systems and corresponding rules as tokens.

In this section, we introduce our emergency scenario and illustrate the main idea of algebraic higher-order (AHO) net systems using high-level net system and rule tokens, while the detailed definitions can be found in Section 3. According to the pipeline emergency scenario[1], a natural gas leak of unknown source is detected in a residential area. At the scene, the company officer calls the gas company and requests an additional law enforcement officer to control traffic into the area. Upon the arrival of the gas company, the firefighters evacuate the homes in the immediate area, subsequently deny entry to this area and finally stand by with fully charged hose lines. In fact, the firefighter company as well as the gas company collaborate through the exchange of messages to achieve the common goal.

In Figure 1, the cooperative workflow enacted by the firefighter company is depicted in the algebraic high-level (AHL) net system $AHLNetSys = (AHLNet, M)$ with AHL net $AHLNet$ and marking M as token on the place net. Such an AHL net system token is called object net system. This object net system is coordinated by an AHO net system at the system level consisting of the places and transitions in the upper row of Figure 1 and the data type part $(\mathbf{AHON\_SIG}, X, A_{HL})$ (see Section 3). The AHO net system consists of two places net holding AHL net systems and $rules$ holding transformation rules. It allows on the one hand to trigger firing steps and on the other hand to apply rule based transformation steps for the object net system firing the transitions *token firing* and *net transformation* as indicated by the net inscriptions $fire(n, t\sigma)$ and $transform(rl, m)$, respectively.

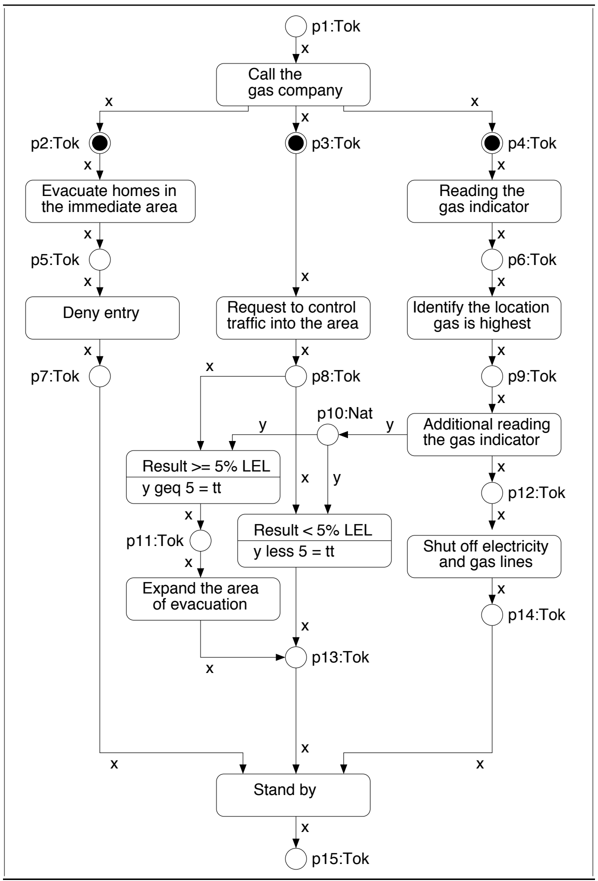

Figure 2: $AHLNetSys'$

The execution of the workflow is controlled by firing the transitions on the system level. To start the activities of the firefighter team the transition *Call the gas company* is fired at the object level using the transition *token firing* at the system level. The result

[1] Pipeline Emergencies Home Page http://pipeline.mindgrabmedia.com

is the new AHL net system $AHLNetSys' = (AHLNet, M')$ shown in Figure 2.

Next we focus on the dynamic changes of the workflow at run time. The firefighters responsible for the evacuation process need more detailed information how to proceed. To introduce the refinement of the *Evacuate homes*-transition into the AHL net system $AHLNetSys'$ we provide the rule $AHLRule$ in Figure 1.

The rule $AHLRule = (L \xleftarrow{l} K \xrightarrow{r} R)$ consists of the three AHL net systems L, K, and R, called left-hand side, interface, and right-hand side, respectively. The marking of the AHL net system L demands that the evacuation process is not yet started because there is one token in the pre domain of the *Evacuate homes*-transition.

To apply the rule $AHLRule$ to the object net $AHLNetSys$, the transition *net transformation* is fired at the system level. The application of the rule refining the *Evacuate homes*-transition is achieved as follows: First, the match morphism m identifies the relevant parts of the left hand side L of the rule $AHLRule$ in the AHL net system $AHLNetSys'$. As a next step, the *Evacuate homes*-transition is deleted and we obtain an intermediate AHL net system. Afterwards, the transitions *Notify residents*, *Assist handicapped persons*, and *Guide persons* together with their (new) environments are added to the intermediate system leading to the new AHL net system $AHLNetSys''$ shown in Figure 3. Thus we obtain the rule based transformation $AHLNetSys' \Longrightarrow AHLNetSys''$ via $(AHLRule, m)$ and the result of firing the transition *net transformation* at the system level is the replacement of the AHL net system $AHLNetSys'$ by $AHLNetSys''$ on the place *net*.

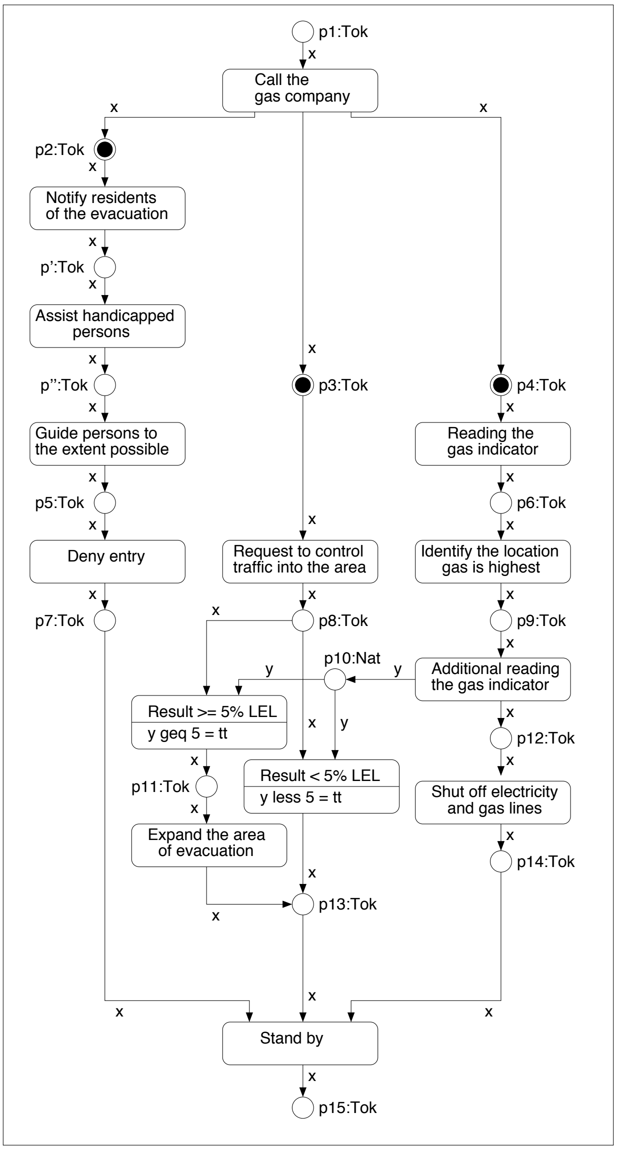

Figure 3: $AHLNetSys''$

Afterwards, the firefighter company proceeds with its activities. The gas company has entered the area. After the identification of the problem the odor of gas grows stronger and the gas company personnel take an additional reading of the gas indicator. Immediately afterwards the gas company personnel inform the company officer about the lower explosive limit (LEL) such that he is able to determine the isolation perimeter. Note that these activities depend on the exchange of messages and data concerning the result of reading the gas indicator and the final analysis of these results by the company officer.

In this paper, we restrict the initial marking of the AHL net system $AHLNetSys$ to one object net system and one corresponding rule to help the reader focus on the main concepts. In our example, only a predefined refinement of the object net system is applicable. Such a situation may be useful for modeling purposes or to show properties of a small object net, which is then transformed using rules that preserve these properties. But in general, different rules are available which are applicable depending on the state of the object net. Their application can be further restricted by application conditions on the rule level or additional firing conditions for the transitions on the system level. Moreover, new rules may be added at runtime which allow to handle unforeseen events. For a full description of the emergency scenario, more object net systems and rules should be defined as, e. g., those given in [BRHM06, PHE+07].

3 AHO Net Systems with High- and Low-Level Tokens

In this section, we introduce our formal framework for modeling data-dependent workflows in mobile ad-hoc networks as motivated by the emergency scenario in Section 2. For this purpose, we introduce algebraic higher-order (AHO) net systems which allow algebraic high-level (AHL) net systems and rules as tokens. This extends our approach in [HME05] where we used AHO net systems with P/T systems and rules as tokens. AHL net systems resp. P/T systems are nets together with initial markings. These and the corresponding rules are used as tokens to model the interaction of token firing and net transformations.

3.1 Review of the Categories PTSys and AHLNetSys

In the following, we review the category **PTSys** of P/T systems [HME05] and the category **AHLNetSys** of AHL net systems [Pra08], which are used as tokens for the corresponding AHO net systems. We use the notation of Petri nets as monoids [MM90], where a P/T net is defined by $PTNet = (P,T, pre, post)$ with pre and post domain functions $pre, post : T \to P^{\oplus}$. A P/T system is given by $PTSys = (PTNet,M)$ with a P/T net $PTNet$ and a marking $M \in P^{\oplus}$, where $P^{\oplus}$ is the free commutative monoid over the set P of places with binary operation $\oplus$. Moreover, each function $f : A \to B$ can be extended to a monoid homomorphism $f^{\oplus} : A^{\oplus} \to B^{\oplus}$ with $f^{\oplus}(\sum_{a\in A} k_a a) = \sum_{a\in A} k_a f(a)$.

Note that M can also be considered as a function $M : P \to \mathbb{N}$ where only for a finite set $P' \subseteq P$ we have $M(p) \geq 1$ with $p \in P'$. We can switch between these notations by defining $\sum_{p\in P} M(p)\cdot p = M \in P^{\oplus}$ and $M(p) = a_p$ for $M = \sum_{p\in P} a_p p \in P^{\oplus}$. Moreover, for $M_1,M_2 \in P^{\oplus}$ we have $M_1 \leq M_2$ if $M_1(p) \leq M_2(p)$ for all $p \in P$. Note that the inverse $\ominus$ of $\oplus$ is only defined in $M_1 \ominus M_2$ if $M_2 \leq M_1$.

Definition 1 (Category **PTSys** of P/T Systems) Given P/T systems $PTSys_i = (PTNet_i,M_i)$ with $PTNet_i = (P_i,T_i, pre_i, post_i)$ for $i = 1,2$, a P/T system morphism $f : PTSys_1 \to PTSys_2$ is given by $f = (f_P, f_T)$ with functions $f_P : P_1 \to P_2$ and $f_T : T_1 \to T_2$ satisfying

(1) $f_P^{\oplus} \circ pre_1 = pre_2 \circ f_T$ and $f_P^{\oplus} \circ post_1 = post_2 \circ f_T$
(2) $M_1(p) \leq M_2(f_P(p))$ for all $p \in P_1$

Moreover, f is called strict if f_P and f_T are injective and

(3) $M_1(p) = M_2(f_P(p))$ for all $p \in P_1$.

The category defined by P/T systems and P/T system morphisms is denoted by **PTSys** where the composition and identities are defined componentwise for places and transitions.

P/T systems are called low-level nets due to the fact that the tokens are indistinguishable. In contrast, a high-level net may hold different kinds of tokens and the firing behavior depends on the token kind. In AHL systems, the tokens are data elements and thus distinguishable.

Definition 2 (Category **AHLNetSys** of AHL Net Systems) An algebraic high-level (AHL) net system $AHLNetSys = (AHLNet, M)$ with

$$AHLNet = ((SIG, X), P, T, pre, post, cond, type, A)$$

consists of an algebraic signature $SIG = (sorts, opns)$ with sorts and operation symbols, additional variables X, sets of places P and transitions T, pre and post domain functions $pre, post : T \to (T_{SIG}(X) \otimes P)^{\oplus}$, firing conditions $cond : T \to \mathscr{P}_{fin}(Eqns(SIG;X))$, a typing of places $type : P \to sorts$, a SIG-algebra A, and an initial marking $M \in CP^{\oplus}$. $T_{SIG}(X)$ is the term algebra with variables over X, $T_{SIG}(X) \otimes P = \{(term, p) | term \in T_{SIG}(X)_{type(p)}, p \in P\}$, $Eqns(SIG;X)$ are all equations over the signature SIG with variables X, and $CP = A \otimes P = \{(a, p) | a \in A_{type(p)}, p \in P\}$.
Given AHL net systems $AHLNetSys_i = (AHLNet_i, M_i)$ with $AHLNet_i = ((SIG, X), P_i, T_i, pre_i, post_i, cond_i, type_i, A)$ for $i = 1, 2$, an AHL net system morphism $f : AHLNetSys_1 \to AHLNetSys_2$ is given by $f = (f_P, f_T)$ with functions $f_P : P_1 \to P_2$ and $f_T : T_1 \to T_2$ satisfying

(1) $(id \otimes f_P)^{\oplus} \circ pre_1 = pre_2 \circ f_T$ and $(id \otimes f_P)^{\oplus} \circ post_1 = post_2 \circ f_T$,
(2) $cond_2 \circ f_T = cond_1$
(3) $type_2 \circ f_P = type_1$
(4) $\forall a \in A_{type_1(p_1)}, p_1 \in P_1 : M_1(a, p_1) \leq M_2(a, f_p(p_1))$

Moreover, f is called strict if f_P and f_T are injective and

(5) $M_1(a, p_1) = M_2(a, f_P(p_1))$ for all $a \in A_{type_1(p_1)}$ and $p_1 \in P_1$.

The category defined by AHL net systems and AHL net system morphisms is denoted by **AHLNetSys** where the composition and identities are defined componentwise for places and transitions.

For the firing of a transition in an AHL net system, we first have to find a suitable variable assignment such that for the term inscription on the pre arc, tokens on the corresponding predomain places are selected satisfying the firing condition.

Definition 3 (Firing Behavior of AHL Net Systems) Given an AHL net system $AHLNetSys = (AHLNet, M)$ with $AHLNet = ((SIG, X), P, T, pre, post, cond, type, A)$, the set of variables $Var(t) \subseteq X$ of a transition $t \in T$ are the variables of the net inscriptions in $pre(t), post(t)$, and $cond(t)$.

Let $\sigma : Var(t) \to A$ be a variable assignment with term evaluation $\sigma^{\#} : T_{SIG}(Var(t)) \to A$, then (t, σ) is a consistent transition assignment iff $cond(t)$ is valid in A under σ. The set CT of consistent transition assignments is defined by $CT = \{(t, \sigma) | (t, \sigma) \text{ consistent transition assignment}\}$.

A transition $t \in T$ is enabled in M under σ iff $(t, \sigma) \in CT$ and $pre_A(t, \sigma) \leq M$, where $pre_A : CT \to CP^{\oplus}$ is defined for $pre(t) = \sum_{i=1}^{n}(term_i, p_i)$ by $pre_A(t, \sigma) = \sum_{i=1}^{n}(\sigma^{\#}(term_i), p_i)$, and similarly for $post_A(t, \sigma)$. Then the follower marking is computed by $M' = M \ominus pre_A(t, \sigma) \oplus post_A(t, \sigma)$.

Intuitively, a pushout in the categories **PTSys** and **AHLNetSys** means the gluing of two net systems along an interface system. The pushout object is constructed componentwise for transitions and places in the category **SET** with corresponding pre and post domain functions and initial markings.

Definition 4 (Pushouts in the category **PTSys**) The pushout $(PN_2, M_2) \xrightarrow{n} (PN_3, M_3) \xleftarrow{g} (PN_1, M_1)$ of the P/T system morphisms $m : (PN_0, M_0) \to (PN_1, M_1)$ and $f : (PN_0, M_0) \to (PN_2, M_2)$, where m is strict, can be constructed as pushout in **PTNet** [EP04], i.e. component-wise for places and transitions. The marking M_3 is defined by

1. $\forall p_1 \in P_1 \setminus m(P_0) : M_3(g_P(p_1)) = M_1(p_1)$
2. $\forall p_2 \in P_2 \setminus f(P_0) : M_3(n_P(p_2)) = M_2(p_2)$
3. $\forall p_0 \in P_0 : M_3(n_P \circ f_P(p_0)) = M_2(f_P(p_0))$

for P_i being the sets of places of PN_i for $i = 0, 1, 2, 3$.

Definition 5 (Pushouts in the category **AHLNetSys**) The pushout $(AHLNet_2, M_2) \xrightarrow{n} (AHLNet_3, M_3) \xleftarrow{g} (AHLNet_1, M_1)$ of the AHL net system morphisms $m : (AHLNet_0, M_0) \to (AHLNet_1, M_1)$ and $f : (AHLNet_0, M_0) \to (AHLNet_2, M_2)$, where m is strict, can be constructed as pushout in **AHLNet** [PER95], i.e. componentwise for places and transitions. The marking M_3 is defined by

1. $\forall p_1 \in P_1 \setminus m(P_0), a \in A_{type_1(p_1)} : M_3(a, g_P(p_1)) = M_1(a, p_1)$
2. $\forall p_2 \in P_2 \setminus f(P_0), a \in A_{type_2(p_2)} : M_3(a, n_P(p_2)) = M_2(a, p_2)$
3. $\forall p_0 \in P_0, a \in A_{type_0(p_0)} : M_3(a, n_P \circ f_P(p_0)) = M_2(a, f_P(p_0))$

for P_i being the places of $AHLNet_i$ for $i = 0, 1, 2, 3$.

Remark 1 *Note that Items 2 and 3 combined are equivalent to* $\forall p_2 \in P_2, a \in A_{type_2(p_2)} : M_3(a, n_P(p_2)) = M_2(a, p_2)$.

3.2 AHO Net Systems with AHL Net Systems as Tokens

An algebraic higher-order (AHO) net system can be considered as an algebraic high-level (AHL) net system modeling the system level. The main difference is that an AHO net system contains not an algebra, but a class algebra, i.e. an algebra with classes instead of sets as base sets. This specific class algebra A defines net systems and rules as tokens at the object level. The class algebra A includes domains like A_{Net}, which is the class of all net systems of a specific kind of low-level or high-level net systems and cannot be restricted to be a set. In the following, we introduce AHO net systems which have AHL net systems and corresponding rules as tokens.

First we introduce an algebraic signature for the class algebra of AHO net systems which can be used for low-level and high-level net systems as tokens, the enabling and firing of transitions as well as the applicability of rules, and the rule based transformation of net systems.

Definition 6 (Signature for AHO Net Systems) The signature for algebraic higher order (AHO) net systems with net systems and rules as tokens is given by:

AHON_SIG =

sorts: $Net, Trans, Bool, Mor, Rules$

opns: $undefined : \to Net,$
$tt, ff : \to Bool,$
$enabled : Net \times Trans \to Bool,$
$fire : Net \times Trans \to Net,$
$applicable : Rules \times Mor \to Bool,$
$transform : Rules \times Mor \to Net$
$cod : Mor \to Net$

Note that the signature **AHON_SIG** does not fix the type of net systems which may be used as tokens. This is achieved by the corresponding class algebras A_{HL} for AHL net systems and A_{PT} for P/T systems as tokens, which are both class algebras for the signature **AHON_SIG**.

For A_{HL}, $(A_{HL})_{Net}$ is the class of all AHL net systems $AHLNetSys$ including a special *undefined* element specified by a constant. In order to model the consistent transition assignments of AHL net systems we define the domain $(A_{HL})_{Trans}$ to be the class **Sets** of all sets. Note, however, that only sets s of the form $s = \{(t,\sigma)\}$ with $(t,\sigma) \in CT_{AHL}$ are relevant, where CT_{AHL} is the set of all consistent transition assignments, $t \in T_{AHL}$ is a transition and $\sigma : Var(t) \to D_{AHL}$ a corresponding variable assignment of some AHL net system $AHLNetSys = ((SIG_{AHL}, X_{AHL}), P_{AHL}, T_{AHL}, pre_{AHL}, post_{AHL}, cond_{AHL}, type_{AHL}, D_{AHL}, M_{AHL})$. For the definition of the operations $enabled_{HL}$ and $fire_{HL}$ we use the extended pre and $post$ domain functions $pre_{D_{AHL}}$ and $post_{D_{AHL}}$ (see Definition 3).

In order to model the applicability of rules and rule based transformations of AHL net systems [Pra08] we use the operations $applicable_{HL}$ and $transform_{HL}$ defined on the domains $(A_{HL})_{Mor}$ of all **AHLNetSys**-morphisms and $(A_{HL})_{Rules}$ of all rules $rl = (L \xleftarrow{l} K \xrightarrow{r} R)$ for AHL net systems. A rule rl is applicable to $AHLNetSys'$ via match morphism $m : AHLNetSys \to AHLNetSys'$ if $L = AHLNetSys$ and m satisfies the gluing condition w.r.t. rl, which means that we have a pushout complement $AHLNetSys_0$ in (1) below. In this case, we can construct the pushout (2) leading to $AHLNetSys''$. The double pushout (1+2) defines the rule based transformation

$AHLNetSys' \Rightarrow AHLNetSys''$ via (rl,m). Note that due to the fact that pushouts are only unique up to isomorphism, we have to fix a concrete construction for the definition of $transform_{HL}$ selecting a specific pushout for each transformation according the one in Definition 5.

$$\begin{array}{ccccc} L & \xleftarrow{l} & K & \xrightarrow{r} & R \\ {\scriptstyle m}\downarrow & (1) & \downarrow & (2) & \downarrow \\ AHLNetSys' & \longleftarrow & AHLNetSys_0 & \longrightarrow & AHLNetSys'' \end{array}$$

Definition 7 (Class Algebra A_{HL} for AHL Net Systems as Tokens) Given the signature **AHON_SIG**, the class algebra A_{HL} for AHL net systems and corresponding rules as tokens is given by

$A_{HL} = ((A_{HL})_{Net}, (A_{HL})_{Trans}, (A_{HL})_{Bool}, (A_{HL})_{Mor}, (A_{HL})_{Rules},$
$\quad undefined_{HL}, tt_{HL}, ff_{HL}, enabled_{HL}, fire_{HL}, applicable_{HL}, transform_{HL})$

with

$(A_{HL})_{Net} = \{AHLNetSys \mid AHLNetSys \text{ is an AHL net system}\} \cup \{undef\}$
the class of all AHL net systems including undefined,
$(A_{HL})_{Trans} =$ **Sets** the class of all sets,
$(A_{HL})_{Bool} = \{true, false\}$,
$(A_{HL})_{Mor} = \{m : AHLNetSys \to AHLNetSys' \mid m$ **AHLNetSys**-morphism$\}$,
$(A_{HL})_{Rules} = \{rl = (L \xleftarrow{l} K \xrightarrow{r} R) \mid l, r$ strict **AHLNetSys**-morphisms$\}$,
$undefined_{HL} = undef$, $tt_{HL} = true$, $ff_{HL} = false$,
$enabled_{HL} : (A_{HL})_{Net} \times (A_{HL})_{Trans} \to (A_{HL})_{Bool}$ defined for $s \in (A_{HL})_{Trans}$ by $enabled_{HL}(undef, s) = false$ and for $AHLNetSys = ((SIG_{AHL}, X_{AHL}), P_{AHL}, T_{AHL}, pre_{AHL}, post_{AHL}, cond_{AHL}, type_{AHL}, D_{AHL}, M_{AHL})$ by

$$enabled_{HL}(AHLNetSys, s) = \begin{cases} true & \text{if } \exists t, \sigma : s = \{(t,\sigma)\}, t \in T_{AHL}, (t,\sigma) \in CT_{AHL} \\ & pre_{D_{AHL}}(t,\sigma) \le M_{AHL} \\ false & \text{else} \end{cases}$$

$fire_{HL} : (A_{HL})_{Net} \times (A_{HL})_{Trans} \to (A_{HL})_{Net}$ with $fire_{HL}(undef, s) = undef$ and for $AHLNetSys = (AHLNet, M_{AHL})$ with

$$fire_{HL}(AHLNetSys, s) = \begin{cases} (AHLNet, M'_{AHL}) & \text{if } enabled_{HL}(AHLNetSys, s) = true, s = \{(t,\sigma)\}, \\ & M'_{AHL} = M_{AHL} \ominus pre_{D_{AHL}}(t,\sigma) \oplus post_{D_{AHL}}(t,\sigma) \\ undef & \text{else} \end{cases}$$

$applicable_{HL} : (A_{HL})_{Rules} \times (A_{HL})_{Mor} \to (A_{HL})_{Bool}$ with

$$applicable_{HL}(rl, m) = \begin{cases} true & \text{if } L = AHLNetSys \wedge m \text{ satisfies gluing condition w.r.t. } rl \\ false & else \end{cases}$$

where $rl = (L \xleftarrow{l} K \xrightarrow{r} R)$ and $m : AHLNetSys \to AHLNetSys'$

$transform_{HL} : (A_{HL})_{Rules} \times (A_{HL})_{Mor} \to (A_{HL})_{Net}$ with

$$transform_{HL}(rl,m) = \begin{cases} AHLNetSys'' & \text{if } applicable_{HL}(rl,m) = true \text{ and} \\ & AHLNetSys' \overset{(rl,m)}{\Longrightarrow} AHLNetSys'' \\ undef & \text{else} \end{cases}$$

$cod_{HL} : (A_{HL})_{Mor} \to (A_{HL})_{Net}$ with

$$cod_{HL}(m) = AHLNetSys' \text{ for } m : AHLNetSys \to AHLNetSys'.$$

Now we are able to define AHO net systems with AHL net systems and rules as tokens and the corresponding category $\mathbf{AHON_{AHL}}$. An AHO net system is defined using a "class algebra based" AHL net system, i.e. an AHL net system $AHON_{AHL}$ in the sense of Definition 2, where the algebra A is allowed to be a class algebra like A_{HL} in Definition 7. But note that the carrier set $(A_{HL})_{Net}$ does not include class based AHL net systems like itself to avoid set theoretical problems of self inclusion.

Definition 8 (Category $\mathbf{AHON_{AHL}}$ of AHO Net Systems) Given the signature $\mathbf{AHON\_SIG}$ (see Definition 6) and the class algebra A_{HL} (see Definition 7), an AHO net system $AHON_{AHL}$ – with AHL net systems and rules as tokens – is a (class algebra based) AHL net system

$$AHON_{AHL} = ((\mathbf{AHON\_SIG}, X), P, T, pre, post, cond, type, A_{HL}, M_{HL})$$

where the signature $\mathbf{AHON\_SIG}$ with sufficiently large family of sets X of variables and algebra A_{HL} are fixed. The initial marking $M_{HL} \in (A_{HL} \otimes P)^{\oplus}$ can be represented by $M_{HL} : A_{HL} \otimes P \to \mathbb{N}$.

A morphism $f : AHON_{1_{AHL}} \to AHON_{2_{AHL}}$ of AHO net systems is an **AHLNetSys**-morphism. This means that for AHO net systems $AHON_{i_{AHL}} = ((\mathbf{AHON\_SIG}, X), P_i, T_i, pre_i, post_i, cond_i, type_i, A_{HL}, M_{i_{HL}})$ for $i = 1,2$, we have $f = (f_P : P_1 \to P_2, f_T : T_1 \to T_2)$ with

$$\begin{tikzcd}
 & T_1 \arrow[dl, "cond_1"'] \arrow[dd, "f_T"] \arrow[r, shift left, "pre_1"] \arrow[r, shift right, "post_1"'] & (T_{\mathbf{AHON_SIG}}(X) \otimes P_1)^{\oplus} \arrow[dd, "(id \otimes f_P)^{\oplus}"] \\
\mathscr{P}_{fin}(Eqns(\mathbf{AHON_SIG}, X)) \arrow[r, phantom, "(=)"] & {} \arrow[r, phantom, "(=)"] & {} \\
 & T_2 \arrow[ul, "cond_2"] \arrow[r, shift left, "pre_2"] \arrow[r, shift right, "post_2"'] & (T_{\mathbf{AHON_SIG}}(X) \otimes P_2)^{\oplus}
\end{tikzcd}$$

$$\begin{tikzcd}
P_1 \arrow[dr, "type_1"] \arrow[dd, "f_P"'] & \\
{} \arrow[r, phantom, "(=)"] & sorts(\mathbf{AHON_SIG}) \\
P_2 \arrow[ur, "type_2"'] &
\end{tikzcd}$$

and $M_{1_{HL}}(a, p_1) \le M_{2_{AHL}}(a, f_P(p_1))$ for all $p_1 \in P_1$ and $a \in (A_{HL})_{type_1(p_1)}$.

The category $\mathbf{AHON_{AHL}}$ of AHO net systems based on the class algebra A_{HL} consists of all AHO net systems as objects and AHO net system morphisms as morphisms.

Example 1 (AHO Net System for Emergency Scenario) *The AHO net system* $AHON_{AHL} = ((\mathbf{AHON\_SIG}, X), P, T, pre, post, cond, type, A_{HL}, M_{HL})$ *in* Figure 1 *consists of*

- *the signature* **AHON_SIG** *(see Definition 6), the* **AHON_SIG***-algebra* A_{HL} *(see Definition 7), and variables* $X_{Net} = \{n\}$, $X_{Trans} = \{t\sigma\}$, $X_{Bool} = \varnothing$, $X_{Mor} = \{m\}$, $X_{Rules} = \{rl\}$,
- *the set of places* $P = \{net, rules\}$ *with* $type(net) = Net$ *and* $type(rules) = Rules$,
- *the set of transitions* $T = \{token\,firing, net\,transformation\}$ *with pre and post domain and firing conditions as shown in Figure 1, and*
- *the initial marking* $M_{HL} = (AHLNetSys, net) \oplus (AHLRule, rules)$, *where* $AHLNetSys$ *and* $AHLRule$ *are shown as marking on the places* net *and* $rules$*, respectively. The data type part of the AHL net systems* $AHLNetSys$ *and* L, K *and* R *in* $AHLRule$ *is given by the signature and corresponding algebra of natural numbers and boolean values together with a sort* Tok *for black tokens.*

As shown in [Pra08] for algebraic high-level net systems with fixed data type, the category $\mathbf{AHON_{AHL}}$ is a weak adhesive HLR category where pushouts are constructed componentwise, because the fact that A_{HL} is a class algebra instead of a classical algebra does not affect these properties.

3.3 AHO Net Systems with P/T Systems as Tokens

In order to define a net class transformation from AHO net systems with high-level net system and rules as tokens to those with low-level net systems and rules as tokens we briefly introduce the second kind of AHO net system. In fact, we can use the same signature **AHON_SIG** (see Definition 6) but use a different class algebra A_{PT} for P/T systems and corresponding rules as tokens.

$$\begin{aligned} A_{PT} = (&(A_{PT})_{Net}, (A_{PT})_{Trans}, (A_{PT})_{Bool}, (A_{PT})_{Mor}, (A_{PT})_{Rules}, \\ &undef, true, false, enabled_{PT}, fire_{PT}, applicable_{PT}, transform_{PT}) \end{aligned}$$

The main difference is that $(A_{PT})_{Net}$ is the class of all P/T systems (including a constant for *undefined*) and $(A_{PT})_{Mor}$, $(A_{PT})_{Rules}$ as well as $applicable_{PT}$ and $transform_{PT}$ are based on **PTSys**-morphisms instead of **AHLNetSys**-morphisms (see Definition 7). The operations $enabled_{PT}$ and $fire_{PT}$ are defined for P/T systems $PTSys = (PTNet, M_N) = (P_N, T_N, pre_N, post_N, M_N)$ and a set $s \in (A_{PT})_{Trans}$ by

$enabled_{PT} : (A_{PT})_{Net} \times (A_{PT})_{Trans} \to (A_{PT})_{Bool}$ with
$enabled_{PT}(undef, s) = false$,

$$enabled_{PT}(PTSys, s) = \begin{cases} true & \text{if } \exists t \in T_N : s = \{t\} \text{ and } pre_N(t) \leq M_N \\ false & \text{else} \end{cases}$$

$fire_{PT} : (A_{PT})_{Net} \times (A_{PT})_{Trans} \to (A_{PT})_{Net}$ with
$fire_{PT}(undef, s) = undef$ and

$$fire_{PT}(PTSys, s) = \begin{cases} (PTNet, M'_N) & \text{if } enabled_{PT}(PTSys, s) = true \text{ with } s = \{t\} \text{ and} \\ & M'_N = M_N \ominus pre_N(t) \oplus post_N(t) \\ undef & \text{else} \end{cases}$$

This class algebra leads to AHO net systems $AHON_{PT}$ with P/T systems and rules as tokens and to the corresponding category $\mathbf{AHON_{PT}}$. Note that our class algebra A_{PT} is more general than the one in [HME05], where $(A_{PT})_{Net}$ is restricted to all P/T systems which are subsystems of a given super net system.

Definition 9 (Category $\mathbf{AHON_{PT}}$ of AHO Net Systems) Given the signature **AHON_SIG** (see Definition 6) and the class algebra A_{PT}, an AHO net system $AHON_{PT}$ with P/T systems and rules as tokens is a (class algebra based) AHL net system $AHON_{PT} = ((\mathbf{AHON\_SIG}, X), P, T, pre, post, cond, type, A_{PT}, M_{PT})$ with fixed $(\mathbf{AHON\_SIG}, X)$ and A_{PT}.

Together with corresponding AHL net system morphisms we obtain the category $\mathbf{AHON_{PT}}$ of AHO net systems based on class algebra A_{PT}.

Example 2 *An example of an AHO net system with P/T systems and rules as tokens is depicted in Figure 4, which is analogously defined to the AHO net system $AHON_{AHL}$ in Figure 1, but the* **AHON_SIG***-algebra is given by A_{PT} and the initial marking by $M_{PT} = (PTSys, net) \oplus (PTRule, rules)$, where $PTSys$ is a P/T system and $PTRule$ a P/T system rule.*

4 Net Class Transformation for AHO Net Systems

Since several net classes form different categories, net class transformations are expressed by functors. Well-known examples of net class transformations are the concepts of *flattening* and *skeleton* as introduced in [Urb03]. In this section, the extended skeleton of AHO net systems is introduced. It is expressed by the functor $F_{Skel} : \mathbf{AHON_{AHL}} \to \mathbf{AHON_{PT}}$ based on the skeleton functor $Skel : \mathbf{AHLNetSys} \to \mathbf{PTSys}$. Both functors are defined and some interesting properties of the functor F_{Skel} are verified.

4.1 Extended Skeleton Functor for AHO Net Systems

The functor $F_{Skel} : \mathbf{AHON_{AHL}} \to \mathbf{AHON_{PT}}$ extends the skeleton functor $Skel : \mathbf{AHLNetSys} \to \mathbf{PTSys}$ (see [Urb03]) that "forgets" the data type of an AHL net system but preserves the token count on the places and the multiplicity of the edges.

Definition 10 (Functor $Skel$) Given the AHL net system $AHLNetSys = ((SIG, X), P, T, pre, post, cond, type, A, M)$ and a morphism $f = (f_P, f_T) : AHLNetSys \to AHLNetSys'$, the skeleton functor $Skel : \mathbf{AHLNetSys} \to \mathbf{PTSys}$ is defined by

$$Skel(AHLNetSys) = PTSys = (P, T, pre', post', M')$$

$$Skel(f) = f' : Skel(AHLNetSys) \to Skel(AHLNetSys')$$

with $pre'(t) = \pi_{P^\oplus}(pre(t))$, $post'(t) = \pi_{P^\oplus}(post(t))$ for all $t \in T$, where $\pi_{P^\oplus} : (T_{SIG}(X) \otimes P)^\oplus \to P^\oplus$ is a projection, $M'(p) = \sum_{a \in A_{type(p)}} M(a, p)$, and $f'_P(p) = f_P(p)$ and $f'_T(t) = f_T(t)$ for all $p \in P$ and $t \in T$.

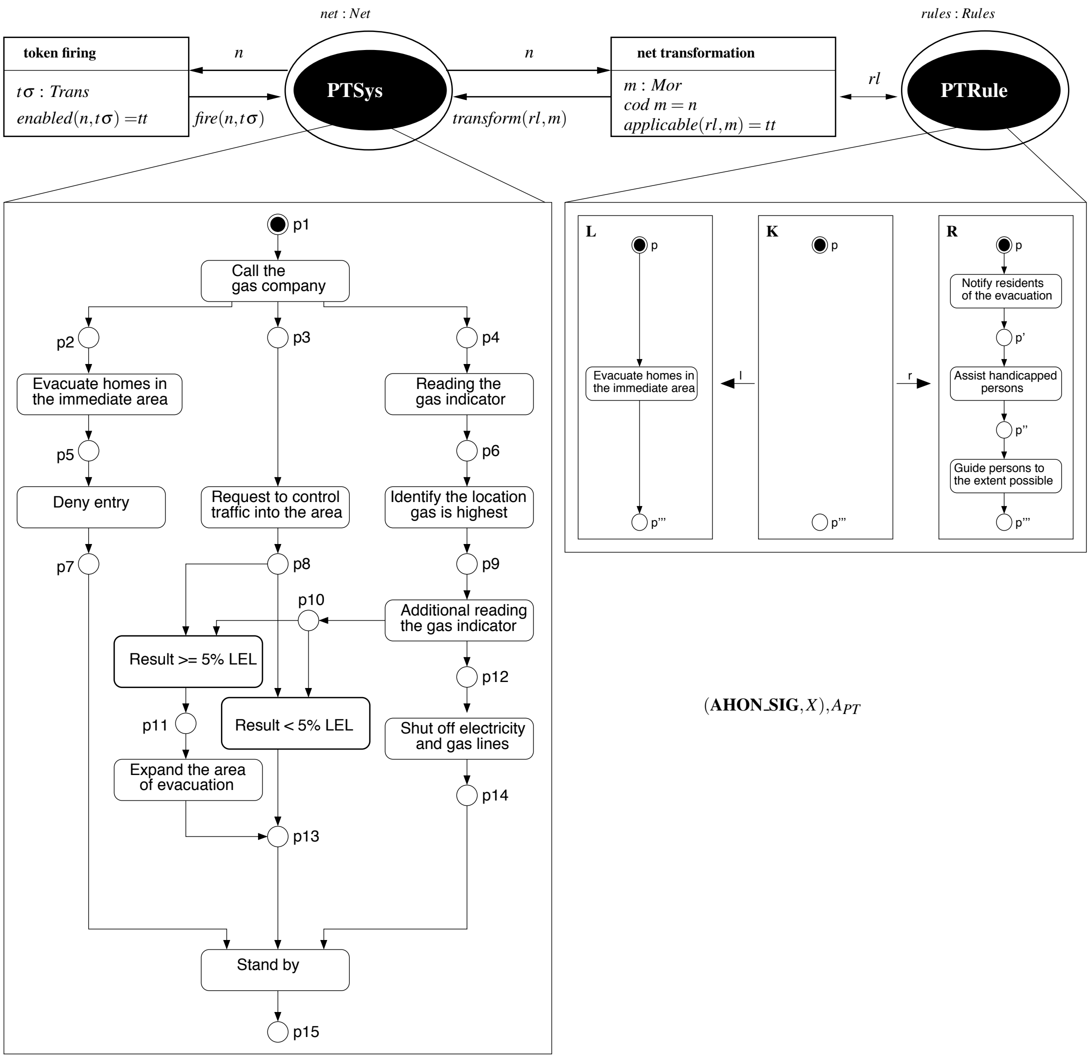

Figure 4: Algebraic higher-order net system $AHON_{PT}$

Remark 2 *The skeleton functor $Skel : AHLNetSys \to PTSys$ preserves pushouts, double-pushout (DPO) transformations, and also enabling and firing of transitions.*

The extended skeleton functor F_{Skel} for AHO net systems "forgets" the data type part of the object net systems using the skeleton functor $Skel$. Moreover, this functor replaces the **AHON_SIG**-algebra A_{HL} for AHL net systems and the corresponding rules by the algebra A_{PT} for P/T systems and the corresponding skeleton rules.

Definition 11 (Functor F_{Skel}) Given the AHO net system $AHON_{AHL} = ((\mathbf{AHON\_SIG}, X), P, T, pre, post, cond, type, A_{HL}, M_{HL})$ with fixed class algebra A_{HL} (see Definition 7) and $M_{HL} : A_{HL} \otimes P \to \mathbb{N}$ then the extended skeleton functor $F_{Skel} : \mathbf{AHON_{AHL}} \to \mathbf{AHON_{PT}}$ is given by

$$F_{Skel}(AHON_{AHL}) = AHON_{PT} = ((\mathbf{AHON\_SIG}, X), P, T, pre, post, cond, type, A_{PT}, M_{PT})$$

with fixed class algebra A_{PT} (see Subsection 3.3) and $M_{PT} : A_{PT} \otimes P \to \mathbb{N}$ defined by

$$(*)\ M_{PT}(a_{PT}, p) = \sum_{a_{HL} \in (A_{HL})_{type(p)}} \ \sum_{S_{type(p)}(a_{HL}) = a_{PT}} M_{HL}(a_{HL}, p)$$

for $a_{PT} \in A_{PT}$, $a_{HL} \in A_{HL}$, and $S : A_{HL} \to A_{PT}$ defined as family $S = [S_s : (A_{HL})_s \to (A_{PT})_s]_{s \in Y}$ for $Y = \{Net, Trans, Bool, Mor, Rules\}$ is given for $a \in (A_{HL})_s$ by

$$S_s(a) = \begin{cases} Skel(a) & \text{for } s \in \{Net, Mor, Rules\} \\ undef & \text{for } s = Net, a = undef \\ \{t\} & \text{for } s = Trans \text{ and } a = \{(t, \sigma)\} \text{ with } t \in T \\ a & \text{for } (s = Trans \text{ and } a \neq \{(t, \sigma)\} \text{ with } t \in T), \text{ or } s = Bool \end{cases}$$

For a morphism $f = (f_P, f_T) : AHON_{AHL} \to AHON'_{AHL}$ of AHO net systems it is defined by $F_{Skel}(f) = f' : AHON_{PT} \to AHON'_{PT}$ with $f' = (f_P, f_T)$.

Remark 3 *Note that $S : A_{HL} \to A_{PT}$ defined above by the family $[S_s]_{s \in Y}$ is not an $AHON_{SIG}$-homomorphism since it is not compatible with the enable-operations as shown in the introduction of the Subsection 4.2.*

In the following theorem, we show that F_{Skel} is a functor and preserves pushouts along strict morphisms and transformations.

Theorem 1

1. $F_{Skel} : \mathbf{AHON_{AHL}} \to \mathbf{AHON_{PT}}$ *is functor.*

2. F_{Skel} *preserves pushouts along strict morphisms and double pushout transformations.*

Proof.

1. From the definition, it is clear that F_{Skel} is compatible with composition and identities. Mainly, we have to show that it is well-defined for the marking, i.e. for $f : AHON1_{AHL} \to AHON2_{AHL}$ in $\mathbf{AHON_{AHL}}$ with $M1_{HL}(a_{HL}, p) \leq M2_{HL}(a_{HL}, f_P(p))$ we have to show that $M1_{PT}(a_{PT}, p) \leq M2_{PT}(a_{PT}, f_P(p))$ for $F_{Skel}(f) : F_{Skel}(AHON1_{AHL}) \to F_{Skel}(AHON2_{AHL})$ in $\mathbf{AHON_{PT}}$. Using (*) in Definition 11 in Section 4 we have that

$$\begin{aligned} M1_{PT}(a_{PT}, p) &= \sum_{S(a_{HL}) = a_{PT}} M1_{HL}(a_{HL}, p) \\ &\leq \sum_{S(a_{HL}) = a_{PT}} M2_{HL}(a_{HL}, f_P(p)) = M2_{PT}(a_{PT}, f_P(p)) \end{aligned}$$

2. First, we show that F_{Skel} preserves pushouts along strict morphisms. Let $AHON_{i_{AHL}} = (AN_{i_{AHL}}, M_{i_{HL}})$ for $i = 0,1,2,3$ and (1) be a pushout in $\mathbf{AHON_{AHL}}$ (see Definition 5) along the strict morphism f_1.

$$\begin{tikzcd} (AN_{0_{AHL}}, M_{0_{HL}}) \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & (AN_{1_{AHL}}, M_{1_{HL}}) \arrow[d, "g_1"] \\ (AN_{2_{AHL}}, M_{2_{HL}}) \arrow[r, "g_2"] & (AN_{3_{AHL}}, M_{3_{HL}}) \end{tikzcd} \quad \begin{tikzcd} (AN_{0_{PT}}, M_{0_{PT}}) \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(2)"] & (AN_{1_{PT}}, M_{1_{PT}}) \arrow[d, "g_1"] \\ (AN_{2_{PT}}, M_{2_{PT}}) \arrow[r, "g_2"] & (AN_{3_{PT}}, M_{3_{PT}}) \end{tikzcd}$$

Applying F_{Skel} we obtain $AHON_{i_{PT}} = (AN_{i_{PT}}, M_{i_{PT}})$ in (2) with $AN_{i_{PT}}$ obtained from $AN_{i_{AHL}}$ by replacing A_{HL} by A_{PT}. We have to show that (2) is a pushout in $\mathbf{AHON_{PT}}$(see Definition 4). According to the pushout construction in **AHLNetSys** (see Definition 5) along the strict morphism f_1 in (1) we have that $M_{i_{HL}}$ $(i = 0,1,2,3)$ with

$$M_{3_{HL}}(a_{HL}, g_1(p_1)) = M_{1_{HL}}(a_{HL}, p_1) \text{ for } p_1 \in P_1 \backslash f_1(P_0)$$

and

$$M_{3_{HL}}(a_{HL}, g_2(p_2)) = M_{2_{HL}}(a_{HL}, p_2) \text{ for } p_2 \in P_2.$$

By definition of F_{Skel} we have

$$M_{i_{PT}}(a_{PT}, p) = \sum_{S(a_{HL}) = a_{PT}} M_{i_{HL}}(a_{HL}, p)$$

for i=0,1,2, 3, where P_i are the places of $AN_{i_{HL}}$ resp. $AN_{i_{PT}}$. It remains to show that

$$\begin{aligned} & M_{3_{PT}}(a_{PT}, g_1(p_1)) = M_{1_{PT}}(a_{PT}, p_1) \text{ for } p_1 \in P_1 \backslash f_1(P_0) \\ \text{and} \quad & M_{3_{PT}}(a_{PT}, g_2(p_2)) = M_{2_{PT}}(a_{PT}, p_2) \text{ for } p_2 \in P_2. \end{aligned}$$

For $p_1 \in P_1 \backslash f_1(P_0)$, we have that

$$\begin{aligned} M_{3_{PT}}(a_{PT}, g_1(p_1)) &= \sum_{S(a_{HL}) = a_{PT}} M_{3_{HL}}(a_{HL}, g_1(p_1)) \\ &= \sum_{S(a_{HL}) = a_{PT}} M_{1_{HL}}(a_{HL}, p_1) = M_{1_{PT}}(a_{PT}, p_1) \end{aligned}$$

Similarly, $M_{3_{PT}}(a_{PT}, g_2(p_2)) = M_{2_{PT}}(a_{PT}, p_2)$ for all $p_2 \in P_2$.

Hence, (2) is a pushout in $\mathbf{AHON_{PT}}$ which implies that F_{Skel} preserves pushouts along strict morphisms. This also implies that F_{Skel} preserves double pushout transformations.

□

4.2 Firing Properties of the Net Class Transformation

In this subsection, we analyze under which conditions the net class transformation F_{Skel} : $\mathbf{AHON_{AHL}} \to \mathbf{AHON_{PT}}$ preserves enabling and firing of transitions at the system level.

As explained above, $h : A_{HL} \to A_{PT}$ based on the skeleton functor $Skel$: **AHLNetSys** $\to$ **PTSys** is not a homomorphism because this functor preserves the firing behavior, but in general it does not reflect enabling and firing of transitions. This means that $enabled_{HL}(AHLNetSys, \{(t,\sigma)\}) = b$ implies $enabled_{PT}(Skel(AHLNetSys)), \{t\}) = b$ for $b = true$ but not for $b = false$.

Definition 12 (Firing Properties of F_{Skel}) Given an AHO net system $AHON_{AHL} = ((\mathbf{AHON\_SIG}, X), P, T, pre, post, cond, type, A_{HL}, M_{HL})$ with $F_{Skel}(AHON_{AHL}) = AHON_{PT}$ we say that F_{Skel} preserves

1. consistent transition assignments, if $\forall t \in T, \sigma : Var(t) \to A_{HL}$:

$$(t,\sigma) \in CT_{AHL} \Rightarrow (t,\sigma') \in CT_{PT} \text{ for } \sigma' = S \circ \sigma$$

2. enabling, if $\forall (t,\sigma) \in CT_{AHL}$:

$$pre_{A_{HL}}(t,\sigma) \leq M_{HL} \Rightarrow pre_{A_{PT}}(t,\sigma') \leq M_{PT} = (S \otimes id_P)^{\oplus}(M_{HL})$$

3. firing, if for M_{HL} enabled under $(t,\sigma) \in CT_{AHL}$:

$$\begin{array}{rl} & M'_{HL} = M_{HL} \ominus pre_{A_{HL}}(t,\sigma) \oplus post_{A_{HL}}(t,\sigma) \\ \Rightarrow & M'_{PT} = (S \otimes id_P)(M'_{HL}) = M_{PT} \ominus pre_{A_{PT}}(t,\sigma') \oplus post_{A_{PT}}(t,\sigma') \end{array}$$

Remark 4 *In general, the functor $F_{Skel} : AHON_{AHL} \to AHON_{PT}$ does not preserve consistent transition assignments, especially if $cond(t)$ includes an equation of the form $enabled(n,t\sigma) = ff$ (see introduction of Subsection 4.2). But F_{Skel} preserves consistent transition assignments if $cond(t)$ includes only special equations of the form $enabled(n,t\sigma) = tt$, $fire(n,t\sigma) = n'$, $applicable(rl,m) = tt$, $transform(rl,n) = n'$, and $cod(m) = n$. In this case, Skel preserves enabling, firing, pushouts, DPO-transformations (see Remark 2), and codomains.*

Theorem 2 (Firing Properties of F_{Skel}) *For an AHO net system $AHON_{AHL}$, F_{Skel} :* $\mathbf{AHON_{AHL}} \to \mathbf{AHON_{PT}}$ *preserves enabling and firing if it preserves consistent transition assignments.*

Proof. Suppose that the arc inscriptions of $AHON_{AHL}$ are only variables or sums of variables. If this is not the case, we modify the AHO net system by introducing for each term tm in the arc inscriptions a new variable x and an equation $x = tm$ for the corresponding transition.

For all $t \in T$, $\sigma : Var(t) \to A_{HL}$ we have that $(t,\sigma) \in CT_{AHL}$ implies $(t,\sigma') \in CT_{PT}$ for $\sigma' = S \circ \sigma$.

1. *F_{Skel} preserves enabling.* For $(t,\sigma) \in CT_{AHL}$ and $pre_{AHL}(t,\sigma) \leq M_{HL}$ we have that

$$pre_{A_{PT}}(t,\sigma') \overset{(**)}{=} (S \otimes id_P)(pre_{A_{HL}}(t,\sigma)) \leq (S \otimes id_P)(M_{HL}) = M_{PT}$$

where (**) holds because $pre(t) = \sum_{i=1}^{n}(term_i, p_i)$ and $\sigma : Var(t) \to A_{HL}$ implies

$$pre_{A_{HL}}(t,\sigma) = \sum_{i=1}^{n}(\sigma^{\#}(term_i), p_i)$$

and

$$\begin{aligned} pre_{A_{PT}}(t,\sigma') &= \sum_{i=1}^{n}(\sigma'^{\#}(term_i), p_i) \\ &= (S \otimes id_P)(\sum_{i=1}^{n}(\sigma^{\#}(term_i), p_i)) \\ &= (S \otimes id_P)(pre_{A_{PT}}(t,\sigma)) \end{aligned}$$

Note that $\sigma' = S \circ \sigma$ implies $\sigma'^{\#}(term_i) = S(\sigma^{\#}(term_i))$ because by assumption the arc inscriptions $term_i$ are only variables. Otherwise we would need $S : A_{HL} \to A_{PT}$ to be an **AHON_SIG**-homomorphism which is not true in general.

2. *F_{Skel} preserves firing.*

$$\begin{aligned} M'_{PT} &= (S \otimes id_P)(M'_{HL}) \\ &= (S \otimes id_P)(M_{HL} \ominus pre_{A_{HL}}(t,\sigma) \oplus post_{A_{HL}}(t,\sigma)) \\ &= (S \otimes id_P)(M_{HL}) \ominus (S \otimes id_P)(pre_{A_{HL}}(t,\sigma)) \oplus (S \otimes id_P)(post_{A_{HL}}(t,\sigma)) \\ &= M_{PT} \ominus pre_{A_{PT}}(t,\sigma') \oplus post_{A_{PT}}(t,\sigma') \end{aligned}$$

□

Example 3 *The result of the extended skeleton functor F_{Skel} applied to the AHO net system $AHON_{AHL}$ in Figure 1 is the AHO net system $AHON_{PT} = ((\textbf{AHON\_SIG}, X), P, T, pre, post, cond, type, A_{PT}, M_{PT})$ with P/T system and rule tokens in Figure 4.*

The firing conditions of $AHON_{AHL}$ in Figure 1 consist only of equations of the form $enabled(n, t\sigma) = tt$, $applicable(rl, m) = tt$, and $cod(m) = n$ so that consistent transition assignments are preserved by the extended skeleton functor (see Remark 4). Moreover, by Remark 4 we obtain new equations of the form $fire(n, t\sigma) = n'$ and $transform(rl, n) = n'$ if we replace the arc inscriptions $fire(n, t\sigma)$ and $transform(rl, n)$ by n' and introduce the new equations as conditions for the transitions token firing and net transformation, respectively. According to Remark 4, F_{Skel} still preserves consistent transition assignments including also there new equations. Hence, we can assume without loss of generality that the arc inscriptions of $AHON_{AHL}$ in Figure 1 on the system level are only variables. Thus, due to Theorem 2, the enabling and firing of $AHON_{AHL}$ in Figure 1 is preserved by F_{Skel}.

In more detail, consider the consistent transition assignment $(token\ firing, \sigma)$ with $\sigma(n) = AHLNetSys$ and $\sigma(t\sigma)$ be the consistent transition assignment of the object net system $AHLNetSys$ given by the Call the gas company-transition and corresponding variable assignment and M'_{HL} be the follower marking of $AHON_{AHL}$ (see Figure 2). Then $(token\ firing, \sigma')$ with

$\sigma'(n) = Skel(AHLNetSys) = PTSys$, $\sigma'(t\sigma) = Call\ the\ gas\ company$ and $PTSys = (PTNet, M)$ *is a consistent transition assignment of the AHO net system* $AHON_{PT}$ *(see Figure 4). Moreover, the transition token firing is enabled in* $AHON_{PT}$ *under* σ' *and the follower marking is defined by* $M'_{PT} = (PTSys', net) \oplus (PTRule, rules)$ *with* $PTSys' = (PTNet, M')$*, where* M' *is given by the marking of the three places in the post domain of the Call the gas company-transition.*

5 Conclusion and Future Work

In this paper, we have introduced AHO net systems, a special kind of algebraic high-level nets with net systems and corresponding rules as tokens. Moreover, we have analyzed the skeleton functor $F_{Skel} : \mathbf{AHON_{AHL}} \to \mathbf{AHON_{PT}}$ relating AHO net systems based on AHL net systems to AHO net systems based on P/T nets.

In [Urb03], net class transformations have been introduced to allow a model developer to change the underlying Petri net class during the modeling step. This concept has been applied on classes of marked place/transition nets and algebraic high-level nets. In addition to the functor $Skel : \mathbf{AHLNetSys} \to \mathbf{PTSys}$ we should analyze the functors $Flat : \mathbf{AHLNetSys} \to \mathbf{PTSys}$ and $Data : \mathbf{PTSys} \to \mathbf{AHLNetSys}$ as given in [Urb03] leading to corresponding functors $F_{Flat} : \mathbf{AHON_{AHL}} \to \mathbf{AHON_{PT}}$ and $F_{Data} : \mathbf{AHON_{PT}} \to \mathbf{AHON_{AHL}}$.

The functor $Flat : \mathbf{AHLNetSys} \to \mathbf{PTSys}$ relates AHL net systems to P/T systems as well, but in contrast to *Skel* it preserves the data type information as each possible marking of the AHL net system is considered to be a place in the P/T system. The data type is flattened into the set of places, hence the name. The functor $Data : \mathbf{PTSys} \to \mathbf{AHLNetSys}$ lifts a P/T system to an AHL net system by providing a trivial data type that describes the black tokens. This functor preserves the net structure and adds a one-sorted specification and an algebra where the carrier set has only one element, that is the black token. Similar to [Urb03], the corresponding functors $F_{Flat} : \mathbf{AHON_{AHL}} \to \mathbf{AHON_{PT}}$ and $F_{Data} : \mathbf{AHON_{PT}} \to \mathbf{AHON_{AHL}}$ formalize net class transformations at the level of tokens of an AHO net system. These net class transformations allow changes of the underlying Petri net class at the token level.

The signature and algebra in this paper are tailored to our needs to represent MANETs with AHO net systems. But the underlying principles including the analysis using the skeleton functor can be transferred to other high-level systems based on additional sorts and operations. A more thorough analysis concerning the properties that are required for the algebra would be an interesting line of future work.

Bibliography

[vdA03] W. van der Aalst. The Application of Petri Nets to Workflow Management. *Journal of Circuits, Systems and Computers* 8(1):21–66, 2003.

[AZ03] D. Agrawal, Q. Zeng. *Introduction to Wireless and Mobile Systems*. Thomson Brooks/Cole, 2003.

[BRHM06] P. Bottoni, F. Rosa, K. Hoffmann, M. Mecella. Applying Algebraic Approaches for Modeling Workflows and their Transformations in Mobile Networks. *Journal of Mobile Information Systems* 2(1):51–76, 2006.

[EEPT06] H. Ehrig, K. Ehrig, U. Prange, G. Taentzer. *Fundamentals of Algebraic Graph Transformation*. EATCS Monographs. Springer, 2006.

[EHP+07] H. Ehrig, K. Hoffmann, J. Padberg, U. Prange, C. Ermel. Independence of Net Transformations and Token Firing in Reconfigurable Place/transition Systems. In *Proc. of ATPN 2007*. LNCS 4546, pp. 104–123. Springer, 2007.

[EP04] H. Ehrig, J. Padberg. Graph Grammars and Petri Net Transformations. In *Lectures on Concurrency and Petri Nets Special Issue Advanced Course PNT*. LNCS 3098, pp. 496–536. Springer, 2004.

[HME05] K. Hoffmann, T. Mossakowski, H. Ehrig. High-Level Nets with Nets and Rules as Tokens. In *Proc. of ATPN 2005*. LNCS 3536, pp. 268–288. Springer, 2005.

[MM90] J. Meseguer, U. Montanari. Petri Nets are Monoids. *Information and Computation* 88(2):105–155, 1990.

[PEH07] J. Padberg, H. Ehrig, K. Hoffmann. Formal Modeling and Analysis of Flexible Processes in Mobile Ad-Hoc Networks. *Bulletin of the EATCS* 91:128–132, 2007.

[PER95] J. Padberg, H. Ehrig, L. Ribeiro. Algebraic High-Level Net Transformation Systems. *MSCS* 5:217–256, 1995.

[PHE+07] J. Padberg, K. Hoffmann, H. Ehrig, T. Modica, E. Biermann, C. Ermel. Maintaining Consistency in Layered Architectures of Mobile Ad-Hoc Networks. In *Proc. of FASE 2007*. LNCS 4422, pp. 383–397. Springer, 2007.

[Pra08] U. Prange. Towards Algebraic High-Level Systems as Weak Adhesive HLR Categories. *ENTCS* 203(6):67–88, 2008.

[RMPM03] F. Rosa, V. Martino, L. Paglione, M. Mecella. Mobile Adaptive Information Systems on MANET: What We Need as Basic Layer? In *Proc. of MMIS 2003*. 2003.

[Urb03] M. Urbášek. *Categorical Net Transformations for Petri Net Technology*. PhD thesis, TU Berlin, 2003.

[Val98] R. Valk. Petri Nets as Token Objects: An Introduction to Elementary Object Nets. In *Proc. of ATPN 1998*. LNCS 1420, pp. 1–25. Springer, 1998.