Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science 242 (2009) 59–71
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Composition and Independence of High-Level Net Processes [1]

H. Ehrig, K. Hoffmann, K. Gabriel, J. Padberg

Institut für Softwaretechnik und Theoretische Informatik
Technische Universität Berlin
Germany

Abstract

Mobile ad-hoc networks (MANETs) are networks of mobile devices that communicate with each other via wireless links without relying on an underlying infrastructure. To model workflows in MANETs adequately a formal technique is given by algebraic higher-order nets. For this modeling technique we here present a high-level net process semantics and results concerning composition and independence. Based on the notion of processes for low-level Petri nets we analyse in this paper high-level net processes defining the non-sequential behaviour of high-level nets. In contrast to taking low-level processes of the well known flattening construction for high-level nets our concept of high-level net processes preserves the high-level structure. The main results are the composition, equivalence and independence of high-level net processes under suitable conditions. Independence means that they can be composed in any order leading to equivalent high-level net processes which especially have the same input/output behaviour. All concepts and results are explained with a running example of a mobile ad-hoc network in the area of a university campus.

Keywords: Algebraic models, algebraic high-level nets, behavioural semantics, high-level net processes, mobility, analysis of nets, composition of processes, equivalence and independence of processes.

1 Introduction

From an abstract point of view mobile ad-hoc networks (MANETs) consist of mobile nodes which communicate with each other independently from a stable infrastructure, while the topology of the network constantly changes depending on the current position of the nodes and their availability. In our research project *Formal Modeling and Analysis of Flexible Processes in Mobile Ad-hoc Networks* we develop the modeling technique of algebraic higher-order nets. This enables the modeling of flexible workflows in MANETs and supports changes of the network topology and the subsequent transformation of workflows. Algebraic higher-order (AHO) nets are Petri nets with complex tokens, especially reconfigurable place/transition (P/T) nets in

[1] This work has been partly funded by the research project forMA$_1$NET (see http://tfs.cs.tu-berlin.de/formalnet/) of the German Research Council.

doi:10.1016/j.entcs.2009.06.023

[6]. AHO-nets can be considered as a special case of algebraic high-level (AHL) nets. The main topic of this paper is to present a high-level process semantics for AHL-nets in general, where the example in Section 2 is given as a MANET and is modeled by an AHO-net.

For low-level Petri nets it is well known that processes are essential to capture their non-sequential truly concurrent behaviour (see e.g. [9,14,1,7,13]). Processes for high-level nets are often defined as processes of the low-level net which is obtained from flatting the high-level net. In [2,5] we have defined high-level net processes for high-level nets based on a suitable notion of high-level occurrence nets which are defined independently of the flattening construction. The flattening of a high-level occurrence net is in general not a low-level occurrence net due to so called assignment conflicts in the high-level net. The essential idea is to generalise the concept of occurrence nets from the low-level to the high-level case. This means that the net structure of a high-level occurrence net has similar properties like a low-level occurrence net, i.e. unitarity, conflict freeness, and acyclicity. But we have to abandon the idea that an occurrence net captures essentially one concurrent computation. Instead, a high-level occurrence net and a high-level process are intended to capture a set of different concurrent computations corresponding to different input parameters of the process. In fact, high-level processes can be considered to have a set of initial markings for the input places of the corresponding occurrence net, whereas there is only one implicit initial marking of the input places for low-level occurrence nets.

In this paper we extend the notion of high-level net processes with initial markings by a set of corresponding instantiations. An instantiation is a subnet of the flattening defining one concurrent computation of the process. The advantage is that we fix for a given initial marking a complete firing sequence where each transition fires exactly once. The main ideas and results in this paper concern the composition of high-level net processes. In general the composition of high-level net processes is not a high-level net process, because the composition may contain forward and/or backward conflicts and also the partial order might be violated. Thus we state suitable conditions, so that the composition of high-level processes leads to a high-level process. We introduce the concept of equivalence of high-level net processes, where the net structures of these high-level net processes might be different, but they have especially the same input/output behaviour. Hence their concurrent computations are compared in the sense that they start and end up with the same marking, but even corresponding dependent transitions may be fired in a different order. In this context the main problem solved in this paper is to analyse the independence of high-level net processes, i.e. under which condition high-level processes can be composed in any order leading to equivalent processes.

The paper is organised as follows. In Section 2 we exemplarily explain the concepts and results of this paper using a mobile ad-hoc network in the area of a university campus. In Section 3 on the one hand we review the notions for high-level net processes and on the other hand we introduce the new notion of high-level net processes with instantiations. In Section 4 we present our main results concerning

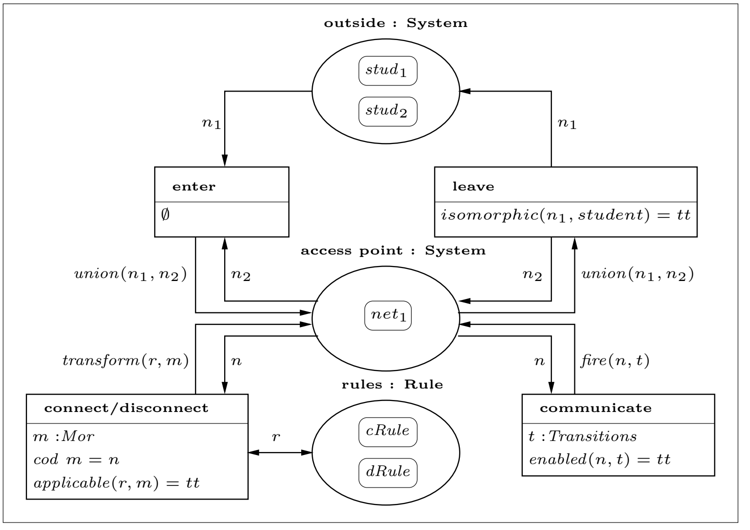


Fig. 1. AHO-net AN_{Campus}

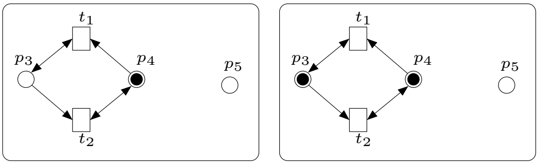
the composition, equivalence and independence of high-level net processes. Due to space limitation the definitions and theorems are given on an informal level, while the details can be found in [4]. Finally we conclude with related work and some interesting aspects of future work in Section 5.

2 Mobile Ad-Hoc Network on University Campus

In this section we introduce a simple example of a wireless network on a university campus and illustrate thereby the concepts in the following sections. As modeling technique we use algebraic higher-order (AHO) nets. AHO-nets are Petri nets with complex tokens, namely place/transition (P/T) nets and rules to support changes of the network topology. With the specific data type part in [10] they can be considered as a special case of algebraic high-level nets.

The example models a network, where students can exchange their messages. For this reason two different locations are represented by the places *outside* and *access point* in the AHO-net AN_{Campus} in Fig. 1. The marking of the AHO-net shows the distribution of the students at different places. Initially there are two students outside the campus and three additional students are on the campus represented by the tokens $stud_1$, $stud_2$ and net_1 in Fig. 1. The mobility aspect of the students is modeled by transitions termed *enter* and *leave* in Fig. 1, while the static structure of the wireless network is changed by rule-based transformations using the rules *cRule* and *dRule*. Moreover the transition *communicate* realises the well known token game.

Subsequently we concentrate on the behaviour of the transitions *communicate* and *connect/disconnect*. On the left hand side of Fig. 2 the P/T-net net_1 of the current network is depicted, where two students, represented by the places p_3 and p_4, respectively, had established a communication structure to exchange messages, while student p_5 is disconnected. The P/T-net net_1 is the token on the place *access*



(a) net_1 (b) net'_1

Fig. 2. Net tokens

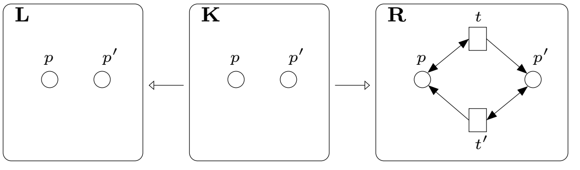
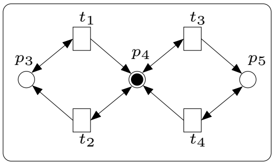


Fig. 3. Rule token *cRule*

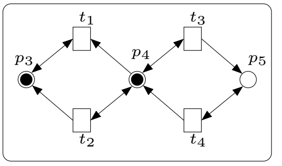
point in Fig. 1. To start the communication we use the transition *communicate* of the AHO-net in Fig. 1. First we give an assignment v_1 of the variables n and t in the environment of this transition and assign the network net_1 to the variable n and the transition t_2 to the variable t. The firing condition checks that the student p_4 is able to send a message. This is modeled by an abstract black token on the place p_4. The evaluation of the net inscription $fire(n, t)$ realises the well-known token game by computing the follower marking of the P/T-net and so we obtain the new P/T-net net'_1 depicted on the right hand side of Fig. 2, where the student p_3 has received the message.

Assume the student p_5 wants to enter the network in order to communicate with the other students. Formally, we apply the rule *cRule* in Fig. 3 that is a token on place *rules* in Fig. 1. In general a rule $r = (L \leftarrow K \rightarrow R)$ is given by three P/T-nets called left-hand side, interface, and right-hand side respectively and the application of a rule describes the replacement of the left-hand side by the right-hand side preserving the interface. The connection between the student p_4 and p_5 is established by firing the transition *connect/disconnect* in the AHO-net in Fig. 1 using the following assignment of the variables n, r and m given in the net inscriptions of this transition: $v'_2(n) = net'_1$, $v'_2(r) = cRule$ and $v'_2(m) = g$, where g is a P/T-net morphism which identifies the left hand side of the rule *cRule* in the network net'_1. In our case the match g maps p to p_4 and p' to p_5. The firing conditions of the transition *connect/disconnect* makes sure that on the one hand the rule is applied to the P/T-net net'_1 and on the other hand the rule is applicable with match g to this P/T-net. Finally we evaluate the term *transform(r,m)* yielding the direct transformation leading to the P/T-net net'_2 on the right hand side in Fig. 4. The effect of firing the transition *connect/disconnect* in the AHO-net in Fig. 1 with assignments of variables as discussed above is the removal of the P/T-net net'_1 from place *access point* and adding the P/T-net net'_2 to the place *access point*.

Vice versa student p_5 can enter the network net_1 by the application of the rule *cRule* to the network net_1 resulting in the network net_2 on the left hand side of Fig. 4 and afterwards students p_3 and p_4 start their communication leading to net net'_2



(a) net_2



(b) net_2'

Fig. 4. Net tokens after rule application

in Fig. 4. Formally this is achieved by firing the corresponding transitions in the AHO-net in Fig. 1 in opposite order with suitable variable assignments v_2 and v_1'.

Summarising, we have explained two different firing sequences of the AHO-net in Fig. 1. The first one starts with the token firing of net_1 leading to the P/T-net net_1' (see Fig. 2) before student p_5 enters the network (see right hand side of Fig. 4). The second one begins by introducing student p_5 into the network net_1 resulting in the network net_2 (see left hand side of Fig. 4) before students p_3 and p_4 exchange the message (see right hand side of Fig. 4).

Similar to processes for low-level nets we want to consider now processes for AHL-nets of which AHO-nets are a special case. These AHL-processes are based on AHL-occurrence nets. In fact the two firing sequences considered above correspond to different AHL-occurrence nets. An AHL-occurrence net is similar to a low-level occurrence net concerning unitarity, conflict freeness, and acyclicity. However, in contrast to a low-level occurrence net an AHL-occurrence net realises more than one concurrent computation depending on different initial markings and variable assignments. So we consider AHL-occurrence nets with a set of initial markings of the input places and corresponding instantiations of places and transitions by data and consistent variable assignments, respectively. For details see Section 3.

In our example we get the two AHL-occurrence nets K and K' on the left hand sides of Fig. 5 and Fig. 6 where the initial marking of the input places is given by the P/T-net net_1 and the rule *cRule*. The corresponding instantiations L_{init} and $L_{init'}$ on the right hand sides of Fig. 5 and Fig. 6 fix the two different firing sequences described above. Note that the AHL-occurrence nets K and K' have the same input and output places. But due to the firing of the transitions *communicate* and *connect/disconnect* in opposite order we use the different variable evaluations v_1 and v_2' in L_{init} and v_2 and v_1' in $L_{init'}$. Nevertheless the two different firing sequences end up with the same marking of the output places where the student p_5 is connected to the other students and the student p_3 received the message from student p_4 as depicted in the P/T-net net_2' on the left hand side of Fig. 4. We show in Section 4 that there are basic AHL-occurrence nets K_1 and K_2, such that K and K' can be obtained as composition in different order of K_1 and K_2. This allows considering the corresponding processes of K and K' with instantiations as equivalent processes of the AHO-net AN_{Campus} in Fig. 1.

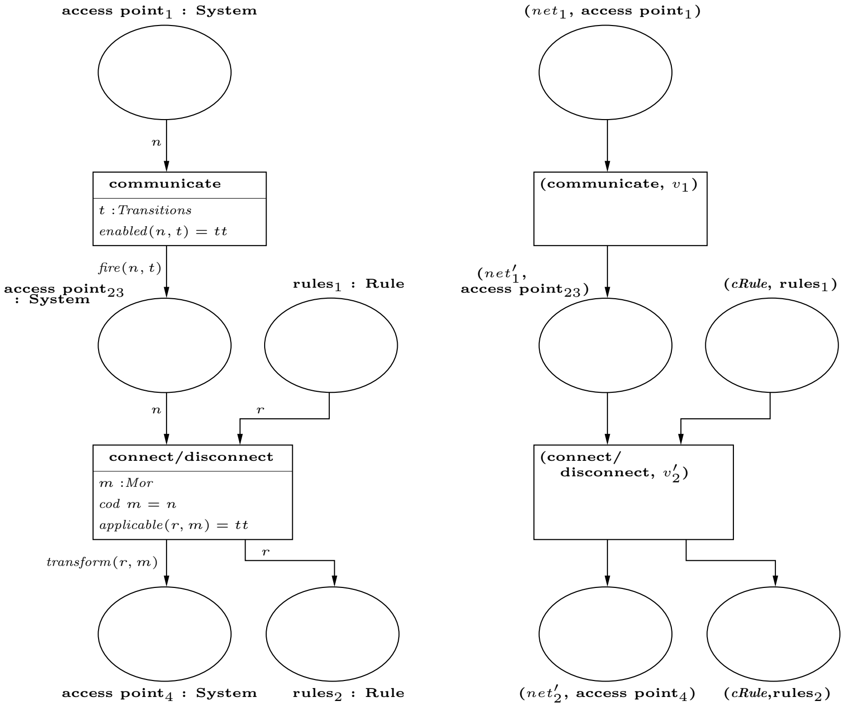


Fig. 5. AHL-occurrence net K with instantiation L_{init}

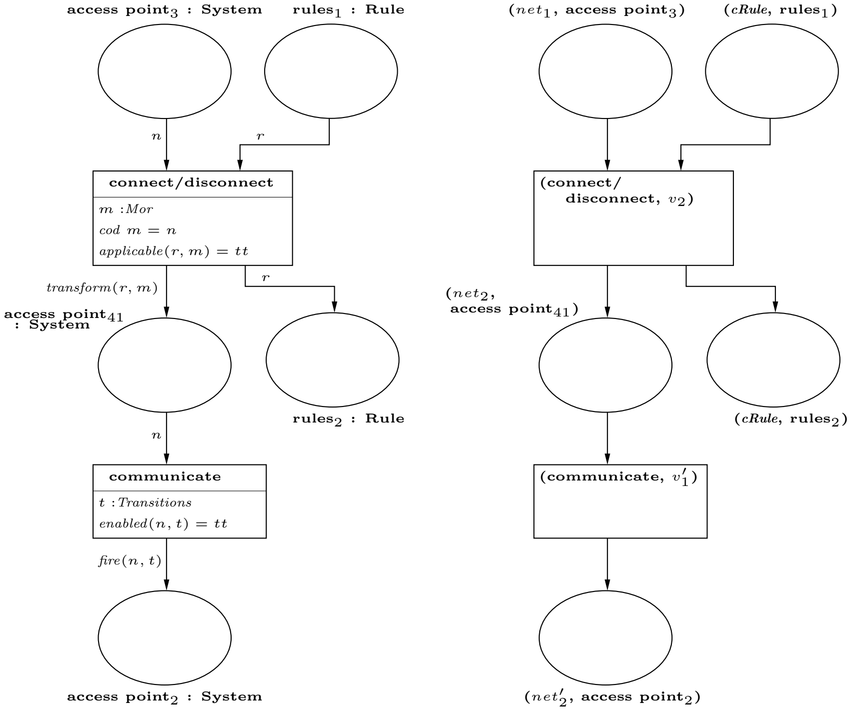


Fig. 6. AHL-occurrence net K' with instantiation $L_{init'}$

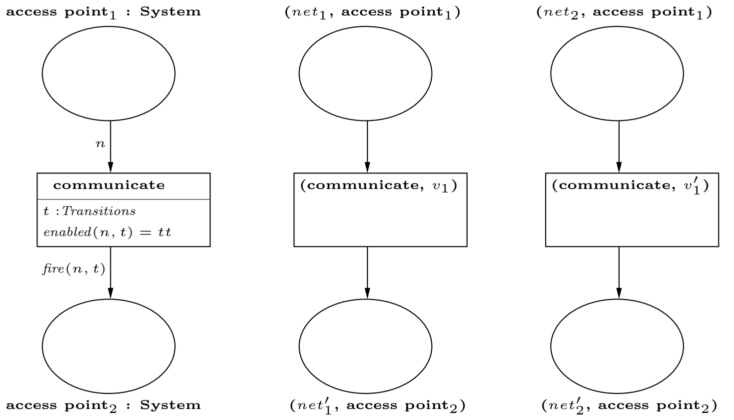


Fig. 7. AHL-occurrence net K_1 with instantiations L_{init_1} and $L_{init'_1}$

3 Algebraic High-Level Net Processes

In this section we review algebraic high-level nets and give a definition of high-level processes [2,5] based on high-level occurrence nets. Moreover we extend this definition by a suitable notation of instantiations for each initial marking.

We use the algebraic notion of place/transition nets as in [12]. A place/transition (P/T) net $N = (P, T, pre, post)$ is given by the set of places P, the set of transitions T, and two mappings $pre, post : T \to P^{\oplus}$, the pre-domain and the post-domain, where $P^{\oplus}$ is the free commutative monoid over P that can also be considered as the set of finite multisets over P. Then we use simple homomorphisms that are generated over the set of places. These morphisms map places to places and transitions to transitions. A P/T-net morphism $f : N_1 \to N_2$ between two P/T-nets N_1 and N_2 is given by $f = (f_P, f_T)$ with functions $f_P : P_1 \to P_2$ and $f_T : T_1 \to T_2$ preserving the pre-domain as well as the post-domain of a transition. Examples of P/T nets with markings are given in Fig. 2 and Fig. 4.

An algebraic high-level (AHL) net [2,5] is essentially a P/T-net together with a suitable data type part given by an an algebraic specification and a corresponding algebra. An AHL-net morphism $f : AN_1 \to AN_2$ between two AHL-nets AN_1 and AN_2 is more or less analogously defined as a P/T-net morphism but in addition the arc inscriptions and firing conditions have to be preserved. An example of an AHL-net is given in Fig. 1. The AHO-net AN_{Campus} is a special case of an AHL-net with specific data type part defining P/T-nets and rules. For details on the signature *HLRN-System-SIG* and algebra A we refer to [10].

Now we introduce high-level occurrence nets and high-level net processes according to [2,5], called AHL-occurrence net and AHL-process respectively. The net structure of a high-level occurrence net has similar properties like a low-level occurrence net. An AHL-occurrence net K is an AHL-net such that the pre- and post domain of its transitions are sets rather than multisets and the arc-inscriptions are unary. Moreover there are no forward and backward conflicts, the partial order given by the flow relation is irreflexive and for each element in the partial order the set of its predecessors is finite.

In contrast to low-level occurrence nets a high-level occurrence net captures a set of different concurrent computations due to different initial markings. In fact, high-level occurrence nets have a set of initial markings for the input places, whereas there is only one implicit initial marking of the input places for low-level occurrence nets. The notion of high-level net processes generalises the one of low-level net processes. An AHL-process of an AHL-net AN is an AHL-net morphism $p : K \to AN$ where K is an AHL-occurrence net described above. Examples of high-level and low-level occurrence nets are given by K and K' (resp. L_{init} and $L_{init'}$) in Fig. 5 and Fig. 6.

Because in general there exist different meaningful markings of an AHL-occurrence net K, we extend this notion by a set of initial markings $INIT$ of the input places of K and a set of corresponding instantiations INS for each initial marking. An instantiation defines one concurrent execution of a marked high-level occurrence net. In more detail an instantiation is a subnet of the flattening of the AHL-occurrence net corresponding to the initial marking. The flattening $Flat(AN)$ of an AHL-net AN results in a corresponding low-level net N, where the data type part (SIG, A) and the firing behaviour of the AHL-net AN is encoded in the sets of places and transitions of N. Thus the flattening $Flat(AN)$ leads to an infinite P/T-net N if the algebra A is infinite. In contrast the skeleton $Skel(AN)$ of an AHL-net AN is a low-level net N' preserving the net structure of the AHL-net but dropping the net inscriptions. While there is a bijective correspondence between firing sequences of the AHL-net and firing sequences of its flattening, each firing of the AHL-net implies a firing of the skeleton, but not vice versa. In [2,5] it is shown that for a marked AHL-occurrence net there exists a complete firing sequence if and only if there exists an instantiation which net structure is isomorphic to the AHL-occurrence net and has the initial marking of the AHL-occurrence net as input places.

Note that in general for a given initial marking of an AHL-occurrence net there exists more than one instantiation. Thus different firing sequences result in different markings of the output places of the AHL-occurrence net. For this reason we fix exactly one instantiation for a given initial marking, i.e. one concurrent execution of the marked AHL-occurrence net. Thus an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$ is given by an AHL-occurrence net K, a set of initial markings $INIT$ and a set of corresponding instantiations INS. An instantiated AHL-process of an AHL-net AN is defined by KI together with an AHL-net morphism $mp : K \to AN$.

As an example the AHL-occurrence net with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ is depicted in Fig. 7 according to the discussion in Section 2. The AHL-occurrence net K_1 is the AHL-net on the left hand side of Fig. 7. There are two different initial markings, i.e the set of initial markings is defined by $INIT_1 = \{(net_1, \mathit{access\ point}_1), (net_2, \mathit{access\ point}_1)\}$ and the set of the two instantiations on the right hand side of Fig. 7 by $INS_1 = \{L_{init_1}, L_{init'_1}\}$.

The instantiated AHL-process is the AHL-occurrence net with instantiations KI_1 together with the AHL-net morphism $mp_1 : K_1 \to AN_{Campus}$. The morphism mp_1 consists of the inclusion of the transition *communicate*, while the places *access*

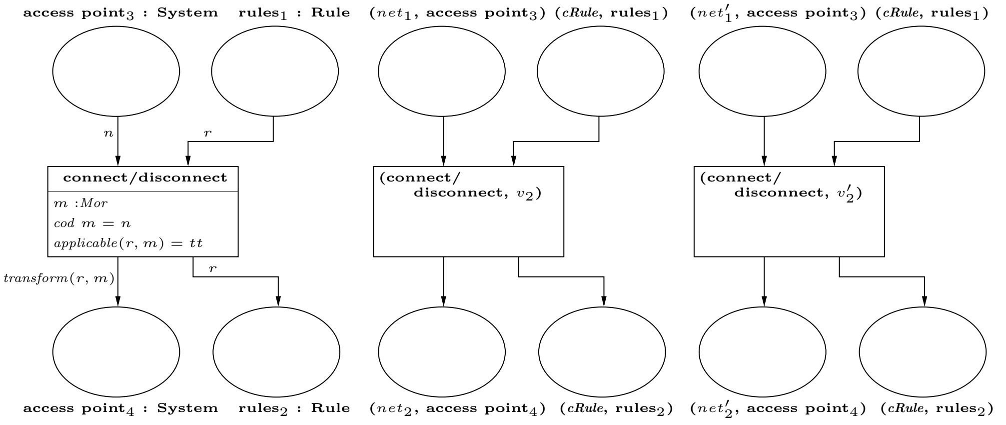


Fig. 8. AHL-occurrence net K_2 with instantiations L_{init_2} and $L_{init'_2}$

point$_1$ and *access point*$_2$ are mapped to the place *access point* of the AHL-net AN_{Campus} in Fig. 1.

Further examples are given in Fig. 5 and Fig. 6, where we have the AHL-occurrence net K with one instantiation $KI = (K, \{init\}, \{L_{init}\})$ and the AHL-occurrence net K' with instantiation KI' together with corresponding morphisms $mp : K \to AN_{Campus}$ and $mp' : K' \to AN_{Campus}$.

4 Composition, Equivalence and Independence of Algebraic High-Level Net Processes

In this section we define the composition of AHL-occurrence nets and AHL-processes with instantiations and introduce the concept of equivalence and independence of high-level net processes. The main result states that two independent high-level net processes can be composed in any order leading to equivalent high-level net processes which especially have the same input/output behaviour. For the detailed theorems and corresponding proofs we refer to [4].

The composition of two AHL-occurrence nets K_1 and K_2 is defined by merging some of the output places of K_1 with some of the input places of K_2, so that the result of the composition is an AHL-occurrence net. In general this is not necessarily true, because the result of gluing two high-level occurrence nets arbitrarily may contain forward and/or backward conflicts and may violate the partial order.

Result 1 (Composition of AHL-Occurrence Nets) *The composition of two AHL-occurrence nets K_1 and K_2 given by merging some of the output places of K_1 with some of the input places of K_2 results in an AHL-occurrence net K.*

As mentioned above instantiations define one concurrent execution of a marked AHL-occurrence net. To generalise the composition given above to the composition of instantiations we have to check that the data elements of the merged output places of K_1 and input places of K_2 are coincident in the corresponding instantiations. In this case the composition of some of the instantiations of KI_1 with some of the instantiations of KI_2 leads to suitable instantiations of the AHL-occurrence net K

that is the result of the composition of the two AHL-occurrence nets K_1 and K_2.

The AHL-occurrence net with instantiations $KI_2 = (K_2, INIT_2, INS_2)$ is given in Fig. 8. The sequential composition of K_1 (see Fig. 7) and K_2 is defined by merging the output place *access point*$_2$ of K_1 and the input place *access point*$_3$ of K_2 leading to the AHL-occurrence net K (see Fig. 5). The corresponding instantiations L_{init_1} in Fig. 7 and $L_{init'_2}$ in Fig. 8 can be composed analogously to the instantiation L_{init} in Fig. 5. Note that L_{init_1} and $L_{init'_2}$ are composable, because they have the same data element net'_1 in the output and input place, respectively.

Result 2 (Composition of AHL-Occurrence Nets with Instantiations) *The composition of two AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and $KI_2 = (K_2, INIT_2, INS_2)$ with composable K_1, K_2 and INS_1, INS_2, respectively, is an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$, where K is the composition of K_1 and K_2 and INS is the corresponding composition of INS_1 and INS_2. The set of initial markings $INIT$ is derived by the input places of the instantiations in INS.*

Given the two basic AHL-occurrence nets with instantiations KI_1 and KI_2, the composition of KI_1 and KI_2 results in the AHL-occurrence net with instantiation KI (see Fig. 5), while the opposite composition of KI_2 and KI_1 is the AHL-occurrence net with instantiation KI' (see Fig. 6).

The following result generalizes the composition to AHL-processes with instantiations where in addition the AHL-net morphisms have to be taken into account.

Result 3 (Composition of AHL-Processes with Instantiations) *Let $KI_1 = (K_1, INIT_1, INS_1)$ and $KI_2 = (K_2, INIT_2, INS_2)$ be two AHL-occurrence nets, such that $KI = (K, INIT, INS)$ is the result of their composition. Let KI_1 together with the AHL-net morphism $mp_1 : K_1 \to AN$ and KI_2 together with the AHL-net morphism $mp_2 : K_2 \to AN$ be two instantiated AHL-processes of the AHL-net AN. If the merged output places of K_1 and input places of K_2 are mapped by mp_1 and mp_2 to the same places in AN then there is one and only one AHL-net morphism $mp : K \to AN$, and KI together with the AHL-net morphism mp is an instantiated AHL-process of the AHL-net AN.*

Because for low-level occurrence nets the input/output behaviour is fixed by the net structure, two low-level occurrence nets are considered to be equivalent if they are isormorphic. For high-level occurrence nets the input/output behaviour additionally depends on the marking of their input places and on corresponding variable assignments. Hence we introduce the equivalence of two AHL-processes with instantiations, where the net structures of equivalent AHL-processes may be different, but they have the same input/output behaviour.

In more detail the AHL-occurrence nets have (up to renaming) the same sets of transitions and places and their instantiations are equivalent, i.e. there exist corresponding instantiations with the same input/output behaviour. In this case specific firing sequences of equivalent AHL-processes are comparable in the sense that they start and end up with the same data elements as marking of their input places and output places, respectively, but in general the corresponding transitions may be fired in a different order.

The AHL-processes with instantiations $KI = (K, \{init\}, \{L_{init}\})$ in Fig. 5 and $KI' = (K, \{init'\}, \{L_{init'}\})$ in Fig. 6 together with the AHL-net morphisms $mp : K \to AN_{Campus}$ and $mp' : K \to AN_{Campus}$ are equivalent. There are bijections between their transitions and places, respectively, which are not isomorphisms. The bijection of places is defined by mapping the input places of K to the input places of K' (and analogously the output places) and the place *access point*$_{23}$ of KI to the place *access point*$_{41}$ of K. Moreover the instantiations L_{init} in Fig. 5 and $L_{init'}$ in Fig. 6 are equivalent, because they have the same input and output places up to renaming.

The main result in this context are suitable conditions s.t. AHL-net processes with instantiation can be composed in any order leading to equivalent high-level net processes. Here we use especially the assumption that the instantiations are consistent, i.e. there is a close relation between their input and output places. Given the AHL-process with instantiations KI together with $mp : K \to AN$ and KI' together with $mp' : K' \to AN$ as results of the composition and opposite composition of KI_1 with $mp_1 : K_1 \to AN$ and KI_2 with $mp_2 : K_2 \to AN$. Now the question arises if KI with mp and KI' with mp' are equivalent processes.

In order to obtain equivalent processes we check that the instantiations INS_1 and INS_2 are consistent, i.e. they can be composed in any order leading to instantiations with the same input/output behaviour. Thus equivalence of KI and KI' intuitively means that the AHL-processes KI_1 and KI_2 with consistent instantiations can be considered to be independent, because the composition in each order leads to equivalent processes.

As an example let KI_1 and KI_2 be the two instantiated AHL-processes as described above. Their sets of instantiations INS_1 and INS_2 are consistent, because the composition of the instantiations L_{init_1} (see Fig. 7) and $L_{init'_2}$ (see Fig. 8) leads to the instantiation L_{init} (see Fig. 5) and the composition of the instantiations L_{init_2} and $L_{init'_1}$ leads to the instantiation $L_{init'}$ (see Fig. 6). Thus, we state the following main result.

Main Result (Equivalence and Independence of AHL-Processes) *Given an AHL-net AN and AHL-occurrence nets $KI_1 = (K_1, INIT_1, INS_1)$ and $KI_2 = (K_2, INIT_2, INS_2)$, which are composable in both directions, with consistent instantiations and AHL-net morphisms $mp_1 : K_1 \to AN$ and $mp_2 : K_2 \to AN$. Then we have instantiated AHL-processes $KI = (K, INIT, INS)$ with $mp : K \to AN$ and $KI' = (K', INIT', INS')$ with $mp' : K' \to AN$ defined by the composition of KI_1 and KI_2 in both directions. Moreover both are equivalent processes of AN, provided that mp_1 and mp_2 are compatible with the compositions. Under these conditions KI_1 and KI_2 are called independent w.r.t. the given composition in both directions.*

Applying this main result to the AHL-net AN_{Campus} in Fig. 1 we have: The two basic instantiated processes defined by KI_1 in Fig. 7 and KI_2 in Fig. 8 are composable with consistent instantiations and the composition in both directions leads to equivalent instantiated processes defined by KI in Fig. 5 and KI' in Fig. 6. Hence the processes defined by KI_1 and KI_2 are independent.

5 Conclusion and Related Work

In this paper we have presented main results of a line of research concerning the modeling and analysis of high-level net processes. Based on the notions of high-level net processes with initial markings in [2,5] we have introduced high-level net processes with instantiations. As main results we have presented conditions for the composition and independence of high-level net processes. Under these conditions the composition of two high-level net processes leads again to a high-level net process and they can be composed in any order leading to equivalent processes. In this case the two high-level net processes are called independent.

In [8,11] the semantics of object Petri nets is defined by a suitable extension of low-level processes. Object Petri nets are high-level nets with P/T-systems as tokens. A process of an object Petri net is given by a pair of processes, a high-level net process containing low-level processes of the corresponding P/T-systems. In contrast the approach presented in this paper extends the notion of high-level net processes for algebraic high-level nets. The token structure of an algebraic high-level net is defined in its data type part that is not restricted to P/T-systems but we also use rules as tokens. Thus low-level processes of P/T-systems as tokens are not considered.

In the example of a wireless network on a university campus (see Section 2) the dynamicity of the communication structure is captured by net transformations, i.e. changes of the network topology are modeled by the application of corresponding rules. While these rules focus on modifications of the net structure, an interesting aspect of future work will be to investigate the concept of broad- and multicasting using rule-based transformations. For this reason rules to modify the marking of an AHO-net have to be introduced, so that a message can simultaneously be sent to a specific number of receivers.

Our main result of independence of high-level net processes is inspired by the results of local Church-Rosser for graph resp. net transformation [15,3], where under suitable conditions transformation steps can be performed in any order leading to the same result. In [6] we have transferred these results, so that net transformations and token firing can be executed in arbitrary order provided that certain conditions are satisfied. Further ongoing work concerns the correspondence between these different concepts of independence in more detail and transfer these results to high-level net processes.

References

[1] Degano, P., J. Meseguer and U. Montanari, *Axiomatizing Net Computations and Processes*, in: *Proc. on Logic in Computer Science (LICS)* (1989), pp. 175–185.

[2] Ehrig, H., *Behaviour and Instantiation of High-Level Petri Net Processes*, Fundamenta Informaticae **65** (2005), pp. 211–247.

[3] Ehrig, H., K. Ehrig, U. Prange and G. Taentzer, "Fundamentals of Algebraic Graph Transformation," EATCS Monographs in TCS, Springer, 2006.

[4] Ehrig, H., K. Gabriel, J. Padberg and K. Hoffmann, *Composition and Independence of High-Level Net Processes*, Technical report, TU Berlin, Fak. IV (2008), (see http://tfs.cs.tu-berlin.de/formalnet/results.html).

[5] Ehrig, H., K. Hoffmann, J. Padberg, P. Baldan and R. Heckel, *High-level net processes*, in: *Formal and Natural Computing*, LNCS **2300** (2002), pp. 191–219.

[6] Ehrig, H., K. Hoffmann, J. Padberg, U. Prange and C. Ermel, *Independence of net transformations and token firing in reconfigurable place/transition systems*, in: *Proc. Application and Theory of Petri Nets (ATPN)*, LNCS **4546** (2007), pp. 104–123.

[7] Engelfriet, J., *Branching Processes of Petri Nets*, Acta Informatica **28** (1991), pp. 575–591.

[8] Farwer, B. and M. Köhler, *Mobile Object-Net Systems and their Processes*, Fundam. Inform. **60** (2004), pp. 113–129.

[9] Goltz, U. and W. Reisig, *The Non-sequential Behavior of Petri Nets*, Information and Control **57** (1983), pp. 125–147.

[10] Hoffmann, K., T. Mossakowski and H. Ehrig, *High-Level Nets with Nets and Rules as Tokens*, in: *Proc. Application and Theory of Petri Nets (ATPN)*, LNCS **3536**, Springer, 2005 pp. 268–288.

[11] Köhler, M. and H. Rölke, *Reference and Value Semantics Are Equivalent for Ordinary Object Petri Nets*, in: *Proc. Application and Theory of Petri Nets (ATPN)*, LNCS **3536** (2005), pp. 309–328.

[12] Meseguer, J. and U. Montanari, *Petri Nets Are Monoids*, Information and Computation **88** (1990), pp. 105–155.

[13] Meseguer, J., U. Montanari and V. Sassone, *On the Semantics of Place/Transition Petri Nets*, Mathematical Structures in Computer Science **7** (1997), pp. 359–397.

[14] Rozenberg, G., *Behaviour of Elementary Net Systems*, in: *Petri Nets: Central Models and Their Properties, Advances in Petri Nets*, LNCS **254** (1987), pp. 60–94.

[15] Rozenberg, G., "Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations," World Scientific, 1997.