Maintaining Consistency in Layered Architectures of Mobile Ad-hoc Networks

Julia Padberg, Kathrin Hoffmann*, Hartmut Ehrig,
Tony Modica, Enrico Biermann, Claudia Ermel

Institute for Software Technology and Theoretical Computer Science
Technical University Berlin, Germany

Abstract. In this paper we present a layered architecture for modeling workflows in **M**obile **A**d-Hoc **NET**works (MANETs) using algebraic higher order nets (AHO nets). MANETs are networks of mobile devices that communicate with each other via wireless links without relying on an underlying infrastructure, e.g. in emergency scenarios, where an effective coordination is crucial among team members, each of them equipped with hand-held devices.
Workflows in MANETs can be adequately modeled using a layered architecture, where the overall workflow, the team members' activities and the mobility issues are separated into three different layers, namely the workflow layer, the mobility layer and the team layer. Dividing the AHO net model into layers immediately rises the question of consistency. We suggest a formal notion of layer consistency that mainly requires that the team layer is given by the mapping of the individual member's activities to the gluing of the workflow and the mobility layer. The main results concern the maintenance of the layer consistency when changing the workflow layer, the mobility layer and the team layer independently.

1 Introduction

Mobile Ad-Hoc Networks (MANETs) consist of mobile nodes, communicating independently of a stable infrastructure. The network topology is changed continuously depending on the actual position and availability of the nodes. A typical example is a team of team members communicating using hand held devices and laptops as e.g. in the disaster recovery scenario in Sect. 3. Formal modeling of workflows in MANETs using algebraic higher order nets (AHO nets) has been first introduced in [1]. AHO nets can be considered as Petri nets with a higher order data type, and hence allow complex tokens, namely place/transition (P/T) nets as well as rules and net transformations for changing these nets. On this basis we present a layered architecture of the model that allows the separation of support activities concerning the network from activities concerning the intended workflow. This allows better and conciser models, since supporting the network

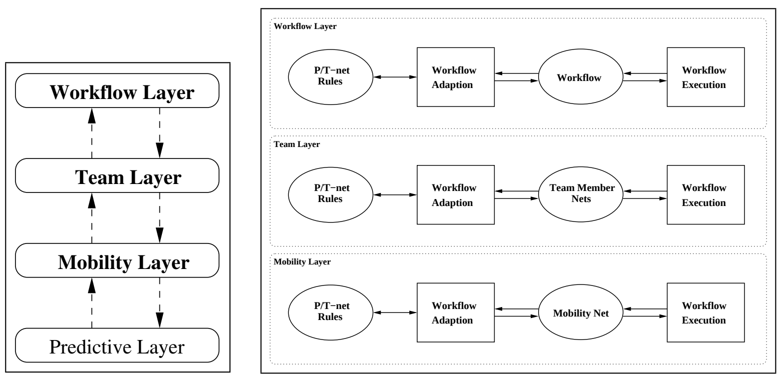
* This work has been partly funded by the research project forMA$_l$NET of the German research Council.

connectivity has a much finer granularity than the more or less fixed workflow execution. The layered architecture of AHO net models of workflows in MANETs distinguishes three layers, the workflow layer, the mobility layer and the team layer. The workflow layer describes the overall workflow that is to be achieved by the whole team. The mobility layer describes the workflows in order to maintain the MANETs connectivity. The team layer describes the individual activities of the team members. Moreover, we provide a set of rules in each layer for the transformation of corresponding P/T-nets expressing different system states. As we distinguish different layers in which transformations are applied independently, the question comes up how these layers fit together. Layer consistency means that these layers together form a valid AHO net model of workflows in MANETs. In a mobile setting it is not realistic to expect consistency at all moments, so there are different degrees of inconsistency that are feasible during maintenance of consistency. Consider the subsequent possibilities for maintaining consistency in a layered AHO net model of workflows in MANETs: *Checking consistency* means that in all states of the AHO net modeling the workflows in MANETs consistency can be checked. *Guaranteed consistency* is given if each state of the AHO net is a consistent one, that is the rules are only applied when the conditions that guarantee consistency are satisfied. *Backtracking consistency* is the possibility to reach an inconsistent state, and to have then the possibility to backtrack the transformations until a consistent state is reached. *Restoring consistency* is the possibility of inconsistent states in the AHO net, but with a "recipe" to fix them. (So, one could consider backtracking as a special case.) This recipe provides conditions for the application of the next transformations. The notion of consistency we present in this paper can be used for all four possibilities. Consistency maintenance depends on the precise AHO net model. More precisely the way consistency is maintained is given by the way rules are applied during the firing of the transitions of the AHO net model. Orthogonally, there are other notions of consistency that are relevant for an AHO net model of workflows in MANETs, e.g. is the intended workflow of the whole team covered by the individual activities of the team members. Another important consistency notion concerns the distributed behavior, in which way the behavior of each member is interrelated with the behavior of the other team members. In the conclusion we hint at the possible formalization of such a team work consistency or behavior consistency in our approach.

In Sect. 2 we discuss our approach to model workflows in MANETs using a layered architecture. Section 3 introduces an exemplary scenario of disaster management, which serves to illustrate the notions and results formalized in Sect. 4. Finally, we conclude discussing future work.

2 Layered Architectures of Mobile Ad-hoc Networks

In [1] a model for MANETs is described by a global workflow and its transformation by a global set of rules. Following the observation that a workflow in MANETs consists of different aspects we provide a layered architecture as depicted in Fig. 1.a to get a more adequate model. Thus it separates movement activities from general activities and it allows a local view of team members that is most important in such an unstable environment. From a practical point of view the MANET topology often has to be restructured to maintain the network connectivity resulting in a change of movement activities while general activities are more or less fixed during the workflow execution. Thus, the global workflow, based on a predictive layer, is separated into three different layers. Each of these layers is equipped with its own place/transition (P/T) nets and transformation rules. The advantage is that we exploit some form of control on rule application by assigning to a specific layer a set of rules. Under these restrictions transformations can be realized only in a specific layer of our model.



(a) Layers (b) Algebraic higher-order net

Fig. 1. Layered architecture for supporting cooperative work on MANETs

The predictive layer signals probable disconnections to the upper mobility layer. The predictive layer implements a probabilistic technique [2] that is able to predict whether in the next instant all devices will still be connected. The mobility layer summarizes movement activities of the involved team members and is in charge of managing those situations when a peer is going to disconnect. The team layer realizes the local view of team members onto the workflow and the mobility net. Here, a P/T-net describes those activities being relevant for one team member. Finally, the workflow layer represents in terms of a P/T-net[1] the cooperative work of the team but excludes movement activities.

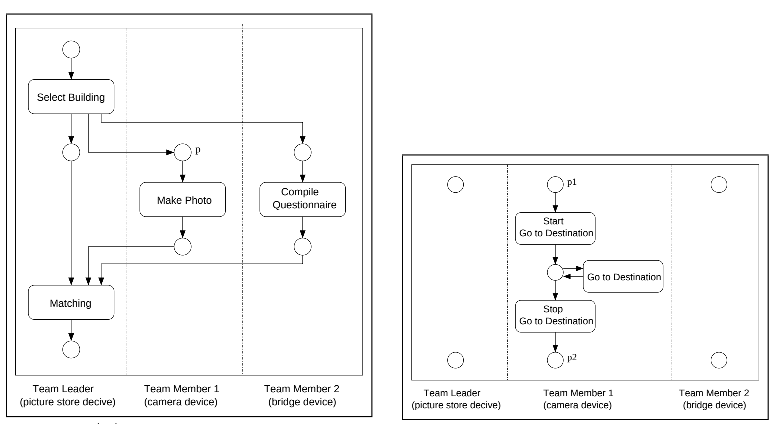
The layered architecture is formalized by a layered AHO net (see Fig. 1.b for a schematic view), so that rules in a certain layer are provided for transformations of corresponding P/T-nets, e.g., to react on some incoming events. In general, AHO nets [4] combine an algebraic data type part and Petri nets. For the purpose of this paper it suffices to consider the subsequent aspects of AHO nets: The integration of these approaches is achieved by the inscription of net elements with terms over the given data structure. Technically, the data type part of the AHO net in Fig. 1.b consists of P/T-nets, the well-known token

[1] Note that we have a P/T-net that describes the workflow, but this need not to be a workflow net in the sense [3], see Sect. 5 for a short discussion.

game, rules and rule-based transformation in the sense of the double pushout (DPO) approach [5], all of them are specified by appropriate sorts and operations. In this way, P/T-nets and rules can be used as tokens in our model, and the token game and rule-based transformations can be implemented in the net inscriptions. Moreover, places in the layered AHO net are either system or rule places, i.e. the state of our model is given by an appropriate marking consisting of P/T-net tokens and rule tokens. Rule tokens are static, i.e. these rules represented as tokens do not move and remain unchanged on the corresponding rule places (indicated by the double arrow). In short, firing a transition **Workflow Adaption** changes the structure of a corresponding P/T-net token according to an appropriate rule token (for details we refer to [1]). Specifically, the mobility layer is in charge of catching disconnection events incoming from the predictive layer and modifying the mobility net (e.g. adding a "Follow Member X" activity) by applying transformations rules.

3 Scenario: Emergency Management

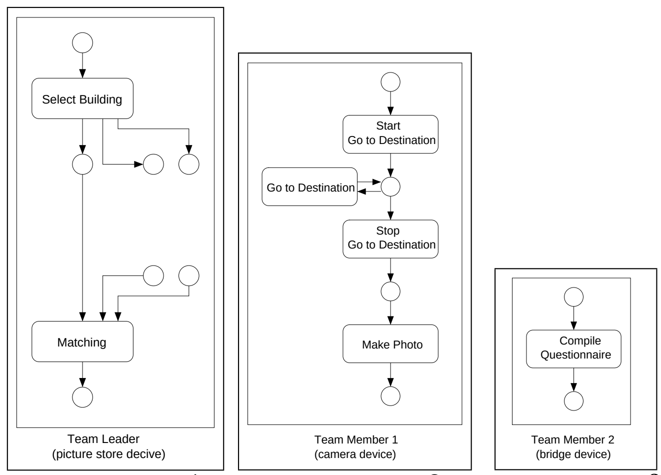
As a running example we use a scenario in archaeological disaster/recovery: after an earthquake, a team (led by a team leader) is equipped with mobile devices (laptops and PDAs) and sent to the affected area to evaluate the state of archaeological sites and the state of precarious buildings. The goal is to draw a situation map in order to schedule restructuring jobs. The team is considered as an overall MANET in which the team leader's device coordinates the other team member devices by providing suitable information (e.g. maps, sensible objects, etc.) and assigning activities. A typical cooperative process to be enacted by the team is shown in Fig. 2.a, where the team leader has to select a building based on previously stored details of the area while team member 1 could take some pictures of the precarious buildings and team member 2 (after a visual analysis of a building) could fill in some specific questionnaires. Finally, these results have to be analyzed by the team leader in order to schedule next activities.



(a) Workflow W_0 (b) Mobility net M_0

Fig. 2. P/T-nets in workflow and mobility layer

In a particular scenario the movement of the device equipped with the camera could result in a disconnection from the others. To maintain the network connectivity and ensuring a path among devices a layered architecture should be able to alert the mobility layer to select a possible “bridge” device (e.g., the one owned by team member 2) to follow the “going-out-of-range” camera device. In general this may result in a change of the MANET topology. Specifically, the current mobility net and the P/T-net of team member 2 have to be transformed in order to adapt it to the evolving network topology.



(a) P/T-net $\mathbf{t_0^1}$ (b) P/T-net $\mathbf{t_0^2}$ (c) P/T-net $\mathbf{t_0^3}$

Fig. 3. Team member nets in team layer

In the following we exemplarily present P/T-nets called token nets that are a possible marking of our AHO net model in Fig. 1.b. Fig. 2.a depicts the token net W_0 for the workflow layer that has to be cooperatively executed by the team, i.e. it represents the current marking of the place **Workflow** in Fig. 1.b. For the mobility layer in Fig. 2.b the token net M_0 is depicted (token net on the place **Mobility Net** in Fig. 1.b). Finally, in Fig. 3 there are three separate nets for the team layer showing the local view of each team member onto the workflow and the mobility net. Thus, the marking on the place **Team Member Nets** in our model consists of the token nets $\mathbf{t_0^1}$, $\mathbf{t_0^2}$ and $\mathbf{t_0^3}$. Here, the mobility aspect of team member 1 states that he/she has to go to the selected destination while the team leader and team member 2 stay put. Note that in general we consider the marking of the token nets. This requires switching from P/T-nets to P/T-systems so that firing a transition **Workflow Execution** in our model (see Fig. 1.b) computes the successor marking of a token net. But in this paper we prefer the notion of P/T-nets because our main results focus on the structure of token nets.

To maintain consistency in a layered architecture first of all the teamwork net T_0 (see Fig. 4) has to be produced by gluing the workflow W_0 and the mobility net M_0 (see Fig. 2). In more detail, the place $\mathbf{p}$ in the workflow W_0 is refined by the movement activities of team member 1. Moreover, the local view of each team member (see Fig. 3) is achieved by an inclusion into the teamwork net, called activity arrow, which realizes the relation of team members to their activities. Thus, we start with a consistent layer environment (for a detailed definition we refer to Subsec. 4.1).

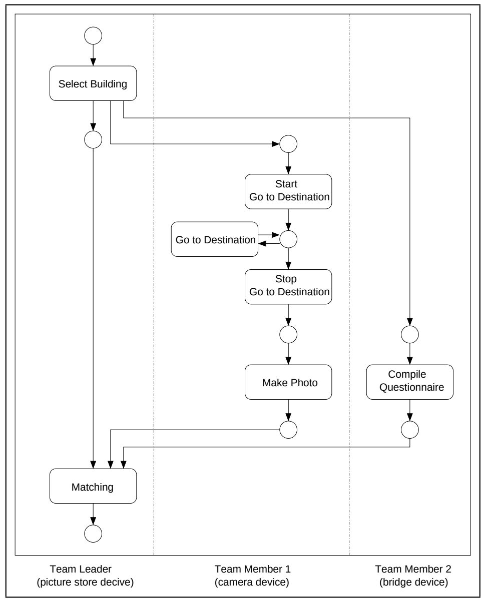
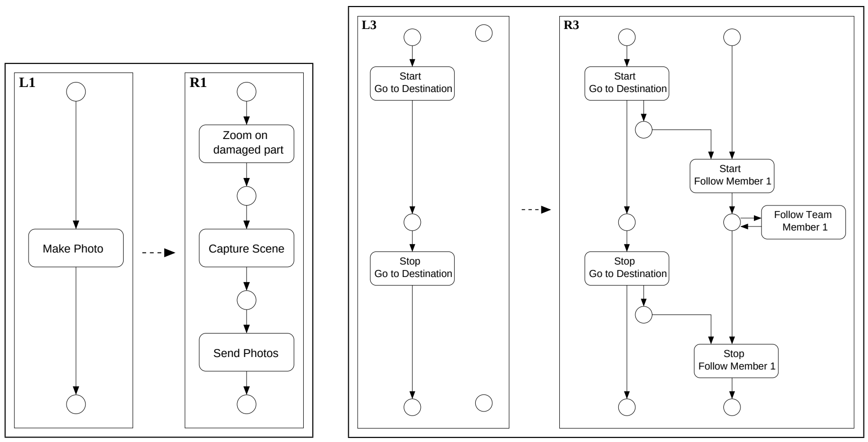


Fig. 4. Teamwork net T_0

According to the requirements of our scenario the structure of the token nets in Figs. 2 and 3 has to be changed to react on incoming events, e.g. to avoid a "going-out-of-range"-situation. Thus, for each layer a specific transformation rule is depicted in Figs. 5 and 6. In general we consider the change of the net structure as rule-based transformation of P/T-nets. This theory is inspired by graph transformation systems [6], which have been generalized already to net transformation systems [5]. The existence of several consistency and compatibility results for net transformation systems is highly profitable for our purpose of maintaining consistency of workflows in MANETs. The basic idea behind net transformation systems is the stepwise development of P/T-nets by appropriate rules. Think of these rules as replacement systems, where the left-hand side of a rule is replaced by the right-hand side. In general, a transformation from a P/T-net N_0 to a P/T-net N_1 by a rule r is denoted by $N_0 \stackrel{r}{\Longrightarrow} N_1$.

In our example team member 1 has to refine his/her activity of making photos. For this reason the structure of the workflow W_0 in Fig. 2.a is changed using the rule r_{photo} in Fig. 5.a for the workflow layer resulting in the new workflow W_1 (see Fig. 16.a in App. B). Assume that the predictive layer signals a probable disconnection, while team member 1 is going to the previous selected destination. Here the rule r_{follow} in Fig. 5.b for the mobility layer maintains the network connectivity by adding movement activities for team member 2 to follow team member 1, i.e. $M_0 \overset{r_{follow}}{\Longrightarrow} M_1$, where M_1 is depicted in Fig. 16.b in App. B. Analogously, the net structure of the local view of team member 2 has to be adapted to include these movement activities. So, we provide the rule r_{m2} in Fig. 6 for the team layer to change the structure of token net $\mathbf{t_0^3}$, i.e. $\mathbf{t_0^3} \overset{r_{m2}}{\Longrightarrow} \mathbf{t_1^3}$ (see Fig. 17.b in App. B). Note that these rules are applied independently in each layer so that consistent transformations can not be guaranteed in general. But we present in the next section layer consistency conditions to maintain consistency of a layered architecture in MANETs, i.e. after the application of specific rules we have again a consistent layer environment.



(a) rule r_{photo} (b) rule r_{follow}

Fig. 5. P/T-net rules in workflow and mobility layer

First of all, the rule r_{photo} is compatible with place refinement because it preserves all involved places (Cond. 1 in Theorem 1). For the same reason, the rules r_{photo} and r_{follow} are independent of the interface given by the overlapping of the workflow W_0 and the mobility net M_0 (Cond. 2 in Theorem 1). Moreover, we obtain the parallel rule r (see Fig. 15 in App. B) consisting of both r_{photo} and r_{follow}. In a next step we focus on the rule r_{m2} in Fig. 6, which is compatible on the one hand with the parallel rule r, i.e. the reduction to those activities of rule r being relevant for team member 2 is equivalent to rule r_{m2} (Cond. 3 in Theorem 1); on the other hand the transformation $\mathbf{t_0^3} \overset{r_{m2}}{\Longrightarrow} \mathbf{t_1^3}$ is compatible with the transformation $T_0 \overset{r}{\Longrightarrow} T_1$, because there is a corresponding inclusion of the resulting token net $\mathbf{t_1^3}$ into the teamwork net T_1 in Fig. 18 in App. B (Cond. 4 in Theorem 1). So, we achieve again a consistent layer environment, i.e. the teamwork net T_1 is given by the gluing of the workflow W_1 and the mobility net M_1, and there are inclusions from the restructured local views of team members to the teamwork net T_1.

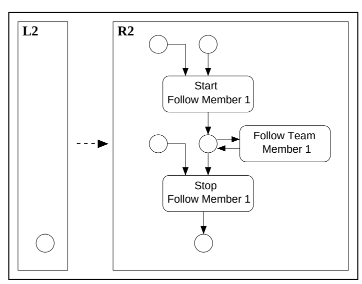


Fig. 6. Rule r_{m2} in team layer

4 Concepts and Results for Layer Consistency

In this section we discuss the basic concepts for maintaining consistency in our approach. Consistency is defined for the layered architecture of workflows in MANETs, that is the *workflow layer*, the *mobility layer* and the *team layer*. We present a notion of consistency, that relates the layers to the team members' activities. Moreover, as discussed in Sect. 3 we have rules and transformations for changes at the level of the workflow layer, of the mobility layer and for changing the individual activities of the team members. These rules and transformations allow the refinement of the workflow according to the imperatives of the network maintenance. To support the local views they have to be applied independently but must allow precise consistency maintenance. So, we give a precise definition of layer consistency and provide precise conditions that allow maintaining consistency. The main theorem states the conditions under which consistency can be maintained stepwise. This result can be extended, so that certain degrees of inconsistency are allowed, while restoring consistency is still possible. In Sect. 4.3 we pick up the discussion on maintaining consistency in view of the notions we present subsequently.

Here, we present these notions and results at a more informal level, but the notions are defined formally and the results have been proved mathematically (a condensed, categorical version can be found in App. A).

4.1 Consistent Layer Environment

Based on the layered architecture for MANETs we have for the *workflow layer* a P/T-net W, for the *mobility layer* a P/T-net M and for the *team work layer* for each team member a P/T-net $\mathbf{t^m}$. For each team member $m = 1, ..., n$ we provide a net $\mathbf{t^m}$ representing their individual activities as well as the relation to the activities of the whole team and rules changing these activities. Here, we assume merely that $\mathbf{t^m}$ are P/T-nets. Alternatively we could require workflow or process nets (see discussion in Section 5). The activities of the team members consist of parts concerning their workflow as well as parts concerning their mobility. Team members can change their team member nets according to specific rules. The main goal of our approach is modeling the changes that occur for reasons of the tasks to be achieved as well as the changes that are required because of the mobility issues. To that end we need the workflow W and rules r^W for transforming W, the mobility net M and rules r^M for transforming M as well as each team member's net $\mathbf{t^m}$ and rules $\mathbf{r^m}$ to transform these. These rules are given as net rules and transformations in the DPO approach [5] (see for example the P/T-net rules in Figs. 5 and 6 in Sect. 3). The nets W, M, and $\mathbf{t^m}$, as well as the rules r^W, r^M and $\mathbf{r^m}$ are the tokens in AHO net depicted in Fig. 1.b. Firing in this AHO net causes the transformation of nets in all three layers at the level of the tokens, i.e. the layer nets and their rules. Consistency

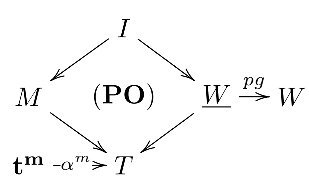


Fig. 7. Consistent layer environment

of such a layered AHO net means in a broad sense that the workflow W, the mobility net M and the individual team member net $\mathbf{t^m}$ of each team member have to be related as depicted in Fig. 7. The interface net is assumed to be fixed throughout this paper, but it is easy to adapt our constructions to changing the interface as well.

More precisely, a *consistent layer environment* according to the layers in Fig. 1.b is given for the team members' nets $\mathbf{t^1}, ..., \mathbf{t^n}$, the workflow W and the mobility net M if the following conditions are satisfied:

- There is the fixed interface net I included in M and W, so that T is the gluing of M and W along I, written $T = M +_I W$.
- There are activity arrows for each team member $\mathbf{t^1} \stackrel{\alpha^1}{\to} T, ..., \mathbf{t^n} \stackrel{\alpha^n}{\to} T$ that are net morphism (see App. A) relating a team member's activities – given by the net $\mathbf{t^m}$ – to the teamwork net T.
- In order to have refinement of places in W with subnets of M we allow replacing W by $\underline{W} \stackrel{pg}{\to} W$, where pg is a place gluing morphism (bijective on transitions and surjective on places; see App. A).

The nets $W, (\mathbf{t^1}, ..., \mathbf{t^n}), M$ and T correspond in our example to the nets given in Figs. 2, 3 and 4, respectively. $\underline{W}$ is obtained from W by splitting **p** in Fig. 2.a into two places **p1** and **p2** that are unconnected. I consists among others of the places **p1** and **p2** included in $\underline{W}$ and also in M in Fig. 2.b where the places **p1** and **p2** are the entry and exit places, respectively. Analogously the other places in I are included in the activity nets of the team leader and team member 2 in Fig. 2.b.

4.2 Transformations at different layers

As mentioned before we want to model changes using rules and transformations at the different layers we have. The transformation of the mobility net M, the workflow W and the team members' activities $\mathbf{t^m}$ is achieved using net transformations as illustrated in Section 3. For more details on net transformation; see [5].

Example 1. Starting at a consistent layer environment firing of the AHO net transitions **Workflow Adaption** in Fig. 1.b yields various transformations in the different layers. So, at the level of the tokens (i.e. nets and rules) we have then e.g. the situation depicted in Fig 8: There are rules in the mobility layer, in the workflow layer and three rules in the team layer that have been applied, yielding the following transformations $M_0 \stackrel{r^M}{\Longrightarrow} M_1$, $W_0 \stackrel{r^W}{\Longrightarrow} W_1$ as well as rules for each team member $\mathbf{t^1_0} \stackrel{\mathbf{r^1}}{\Longrightarrow} \mathbf{t^1_1}$, $\mathbf{t^2_0} \stackrel{\mathbf{r^2}}{\Longrightarrow} \mathbf{t^2_1}$ and $\mathbf{t^3_0} \stackrel{\mathbf{r^3}}{\Longrightarrow} \mathbf{t^3_1}$. This is the situation as discussed in Sect. 3 with the team members' nets $\mathbf{t^1_0}$, $\mathbf{t^2_0}$ and $\mathbf{t^3_0}$.

According to the discussion in Section 1 we now need conditions that allow maintaining consistency. We have to obtain the teamwork net that integrates the changes induced by the transformations above. The results for net transformations (see [5]) yield a variety of independence conditions for the sequential, parallel application of rules and for the compatibility with pushouts. Subsequently we develop the conditions for maintaining layer consistency based on transformations at the mobility and the workflow layer. Later in Cor. 1 we assume not only transformations, but transformation sequences.

Let there be the transformations $W_0 \stackrel{r^W}{\Longrightarrow} W_1$ and $M_0 \stackrel{r^M}{\Longrightarrow} M_1$. We first need to ensure *compatibility with the place refinement* (Cond. 1). That means the transformation using the rule r^W is compatible with the pg-morphisms in the following sense: A rule r^W may be applied to W_0 only if it is applicable to $\underline{W}_0$ and applicable to W_0 via $\underline{W}_0$ (in the sense that we have Fig. 14 in App. A), leading to transformations $\underline{W}_0 \stackrel{r^W}{\Longrightarrow} \underline{W}_1$ and $W_0 \stackrel{r^W}{\Longrightarrow} W_1$ together with a place gluing morphism $pg_1 : \underline{W}_1 \to W_1$ (see Fig. 9).

Next, we apply the Union Theorem for net transformations (see App. A). Provided the *preservation of the interface* I (Cond. 2), that is the applications of the rules r^W and r^M are independent of I then there is the parallel rule $r = r^W + r^M$, so that the application of r to the teamwork net T_0 yields the transformation $T_0 \stackrel{r}{\Longrightarrow} T_1$, with $T_1 = M_1 +_I \underline{W}_1$. So, the first step to the next consistent layer environment is achieved.

Now we restrict the transformation $T_0 \stackrel{r}{\Longrightarrow} T_1$ to the transformations $\mathbf{t_0^m} \stackrel{\mathbf{r^m}}{\Longrightarrow} \mathbf{t_1^m}$ for each team member $m = 1, ..., n$. Since the team members' activities are represented by activity arrows, the rules have to be compatible with arrows. The *existence of activity rules* (Cond. 3) ensures that for each team member the rule $r = (L \leftarrow K \rightarrow R)$ is restricted to an activity rule $\mathbf{r^m} = (\mathbf{L^m} \leftarrow \mathbf{K^m} \rightarrow \mathbf{R^m})$, where $\mathbf{K^m}$ has to be the pullback (roughly an intersection) of $\mathbf{L^m}$ and K as well as the pullback of $\mathbf{R^m}$ and K (see App. A).

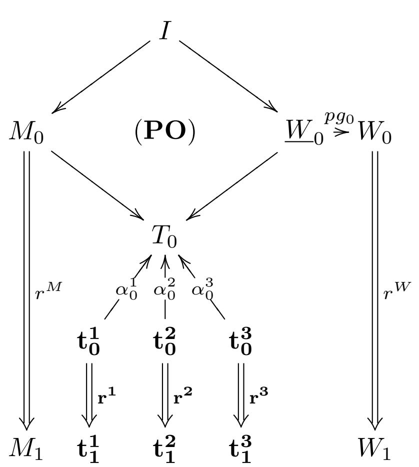


Fig. 8. State after some transformations

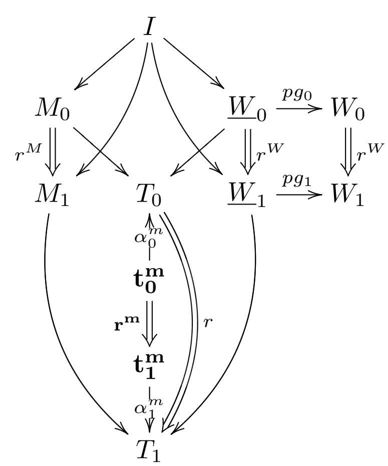


Fig. 9. A new consistent layer environment for $m = 1, 2, 3$

Moreover, each activity rule $\mathbf{r^m}$ has to be the reduction of the corresponding rule r to that part being relevant for the team member m. Provided the *conformance of activity rules and team member nets* (Cond. 4) that means $\mathbf{L^m}$ is additionally the pullback of $\mathbf{t_0^m}$ and L, the application of an activity rule $\mathbf{r^m}$ to a team member net $\mathbf{t_0^m}$ yields the transformation $\mathbf{t_0^m} \stackrel{\mathbf{r^m}}{\Longrightarrow} \mathbf{t_1^m}$.

Then we can state our first main result, that provides the conditions for stepwise consistency maintenance.

Theorem 1 (Stepwise Consistency Maintenance). *Given a consistent layer environment $T_0 = M_0 +_I \underline{W}_0$ with the place gluing $\underline{W}_0 \stackrel{pg_0}{\to} W_0$ and the activity arrows $\mathbf{t_0^m} \stackrel{\alpha_0^m}{\to} T_0$ for each member $m = 1, ..., n$, then the transformations $W_0 \stackrel{r^W}{\Longrightarrow} W_1$, $M_0 \stackrel{r^M}{\Longrightarrow} M_1$ and the transformations $\mathbf{t_0^m} \stackrel{\mathbf{r^m}}{\Longrightarrow} \mathbf{t_1^m}$ yield again a consistent layer environment $T_1 = M_1 +_I \underline{W}_1$ with the place gluing $\underline{W}_1 \stackrel{pg_1}{\to} W_1$ and the activity arrows $\mathbf{t_1^m} \stackrel{\alpha_1^t}{\to} T_1$ for each m, provided the* layer consistency conditions *hold:*

1. *compatibility with the place refinement, i. e. the rule r^W is compatible with the morphism pg,*
2. *preservation of the interface I, i.e. the application of the rules r^W and r^M are independent of I,*
3. *existence of activity rules, i.e. for each m there are activity rules $\mathbf{r^m}$ over the parallel rule $r = r^W + r^M$ and*
4. *conformance of activity rules and team member nets, i.e. $\mathbf{t_0^m} \stackrel{\mathbf{r^m}}{\Longrightarrow} \mathbf{t_1^m}$ is compatible with $T_0 \stackrel{r}{\Longrightarrow} T_1$.*

Proof. see App. A

Example 2. Considering the example in Section 3, outlined in Fig. 8 we have the following situation: There is a rule $r = r^W + r^M$, r^W is compatible with place refinement, the application of the rules and $\mathbf{r^m}$ is an activity rule over r. Then the compatibility of r^W with place refinement yields the transformations $\underline{W}_0 \stackrel{r^W}{\Longrightarrow} \underline{W}_1$ and the place-gluing morphism $\underline{W}_1 \stackrel{pg_1}{\to} W_1$. Using the Union Theorem we have the pushout $T_1 = \underline{W}_1 +_I M_1$ and the construction of the activity rule yields the activity arrow $\mathbf{t_1^m} \stackrel{\alpha_1^m}{\to} T_1$. So, we obtain the consistent layer environment depicted in Fig 9. If we allow transformation sequences instead of transformation steps in Theorem 1 we may obtain inconsistent states. For recovery of consistency then we need additional conditions. At the different layers the application of the rules needs to be checked with respect to the last known consistent state, because there cannot be made assumptions on the actual state of the layers. Technically this can be achieved using parallel independent rules, where the independence is considered with respect to the last known consistent state. The subsequent corollary states that restoring consistency under these conditions is achieved using Theorem 1 twice.

Corollary 1 (Restoring Consistency). *Given a consistent layer environment, shortly* $(\mathbf{t_0^m} \overset{\alpha_0^m}{\to} T_0 = M_0 +_I \underline{W}_0 \overset{pg_0}{\to} W_0)$ *then the transformation sequences* $M_0 \overset{*}{\Longrightarrow} M_{n_M}$ *via* r_i^M *and* $W_0 \overset{*}{\Longrightarrow} W_{n_W}$ *via* r_j^W *and the transformation steps* $\mathbf{t_0^m} \overset{\mathbf{r^m}}{\Longrightarrow} \mathbf{t_1^m}$ *lead to the possibly inconsistent state depicted in Fig. 10.*

An intermediate layer consistent state $(\mathbf{t_1^m} \overset{\alpha_1^m}{\to} T_1 = M +_I \underline{W} \overset{pg}{\to} W)$ *can be constructed (see Fig. 11) provided we have*

- *that all rules* r_i^M *and* r_j^W *are parallel independent of* M_0 *and* W_0*, respectively*
- *the parallel rules* r^M *and* $\bar{r}^M$ *for the decomposition of the transformation sequence* $(M_0 \overset{*}{\Longrightarrow} M_{n_M}) = (M_0 \overset{r^M}{\Longrightarrow} M \overset{\bar{r}^M}{\Longrightarrow} M_{n_M})$,
- *the parallel rules* r^W *and* $\bar{r}^W$ *for the decomposition of the transformation sequence* $(W_0 \overset{*}{\Longrightarrow} W_{n_W}) = (W_0 \overset{r^W}{\Longrightarrow} W \overset{\bar{r}^W}{\Longrightarrow} W_{n_W})$ *and*
- *that the layer consistency conditions in Theorem 1 hold for the parallel rules* r^W *and* r^M.

The next consistent layer environment $\mathbf{t_2^m} \overset{\alpha_2^m}{\to} T_2 = M_{n_M} +_I \underline{W}_{n_W} \overset{pg_{n_W}}{\to} W_{n_W}$ *can be constructed as depicted in Fig. 11 provided there are activity rules* $\mathbf{\bar{r}^m}$ *so that the layer consistency conditions in Theorem 1 hold again for the parallel rules* $\bar{r}^W$ *and* $\bar{r}^M$.

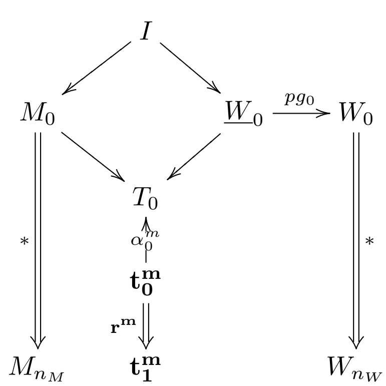


Fig. 10. Possibly inconsistent

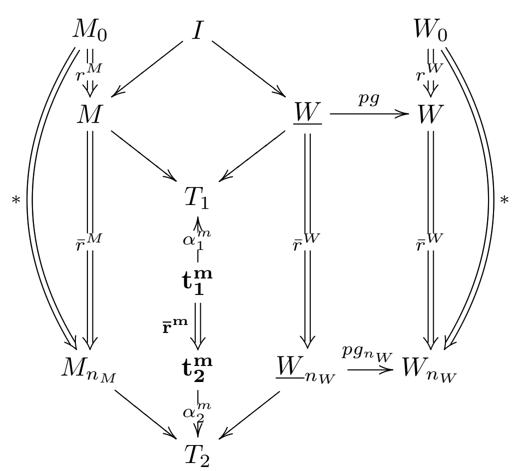


Fig. 11. Restoring the next consistent layer environment

4.3 Maintaining Consistency

The notions and results we have introduced above concern the fundamental understanding of consistency in MANETs. As mentioned in the introduction other notions of consistency are possible and desirable. The AHO net model given in Fig. 1.b merely presents the rough structure but abstracts especially from the details of the firing conditions. The exact formulation of the firing conditions models the way the rules are applied in the different layers. Hence the formulation of the firing conditions of the AHO net constitutes the way consistency is dealt with. The discussion below abstracts from realization issues, as e.g. the complexity of the task to find morphisms between nets. Considering the possibilities discussed in the introduction we have:

- *Checking consistency:* The AHO net in Fig. 1.b allows the application of arbitrary rules and it can be checked for a consistent layer environment. Since we have a formal definition of consistency, it can be checked whether a certain state of an AHO net model for MANETs is a consistent layer environment. There need to be the fixed interface I, the token nets M and W on the places **Mobility Net** and **Workflow**, respectively and the token nets $\mathbf{t^m}$ for each team member m on the place **Team Member Nets**, so that they present a consistent layer environment. This means there are nets T and $\underline{W}$, so that there is a place gluing morphism $\underline{W} \to W$, T is the gluing of M and $\underline{W}$ along I and there are m activity arrows $\mathbf{t^m} \stackrel{\alpha^m}{\to} T$.
- *Guaranteed consistency:* Theorem 1 ensures transformations so that each state is consistent. Then the AHO net in Fig. 1.b may allow only the application of rules that satisfy these conditions. Moreover, the parallel firing of the transitions in the different layers has to be ensured to have consistency in each state.
- *Backtracking:* Since all rules are symmetric (as one of the characteristics of the DPO approach) the inverse rules can be applied in the inverse order. Then the AHO net in Fig. 1.b may allow the application of arbitrary rules, but requires a storage of the transformations. Then an explicit backtracking can be achieved by firing the transitions in the AHO net but using only the inverse rules.
- *Restoring consistency:* Corollary 1 gives conditions for restoring consistency. Then the AHO net in Fig. 1.b may allow only the application of rules that satisfy these conditions. An explicit restoration is possible using the transformations constructed in the corollary. Note that here we merely treat transformation sequences for the mobility and the workflow layer. Restoring consistency after transformation sequences at the team layer is very closely related to the question of team work consistency and hence not treated here (see Sect. 5 for a short discussion).

5 Conclusion

The use of a layered architecture for modeling workflows in MANETs has the advantage of separating different views with different granularity, but rises the question of consistency immediately. In this paper, we have presented the notion of layer consistent environment stating that the views in the workflow layer, the mobility layer and the team layer fit together. Since the main modeling advantage of AHO nets is the possibility to model net transformations we have introduced maintenance means for the AHO net for workflows in MANETs that take changes modeled by net transformation into account.

Related work on distribution of workflows in a possibly mobile setting can be found e.g. in [7–9] where a unique workflow is divided on the one hand in different autonomous workflows and on the other hand the resulting workflows are adapted by using inheritance resp. graph rules. In contrast we present a layered architecture, where a global workflow and its transformation are separated into three different parts, each of them relevant for a specific aspect of workflows in MANETs.

Outlook In this paper we present the first results of a larger research activity[2] concerning formal modeling and analysis of MANETs. So, there is a large amount of most interesting and relevant open questions. The subsequent issues concern questions directly related to the work presented here:

Behaviour of token nets The behaviour of the token nets has been treated in previous papers [1] and has be deliberately excluded here. The nets in the different layers have their own behaviour that is executed by firing the corresponding transitions in the AHO net (see Fig. 1.b). This directly leads to a most challenging consistency issue, namely how are the individual processes related to each other. A very elegant solution would be to use the theory for open nets [10].

Team work consistency Other relevant notions of consistency concern e.g. the consistency between each team member's activities and the complete teamwork. It should be ensured, that the team members' activities together cover the complete team work. This can be realized in our categorical approach using a given topology graph to glue the team members' nets together, then team consistency is given if this gluing corresponds to the teamwork net T. Then again, team consistency has to be maintained during transformations in the different layers.

Restriction of activities In this paper we have used arbitrary P/T-nets without further restrictions for modeling the layers as well as the team members' activities. Nevertheless, syntactic restrictions, e.g restricting the team members' activities to (non)-deterministic processes as well as semantic restrictions, e.g. using the approach of workflow nets in the sense of [3] for all involved nets may be useful. The restriction of the P/T-nets in the different layers requires some additional treatment. To restrict team members' activities to (non)-deterministic processes the approach to the categorical formulation of processes of (open) nets in [10] can be adopted successfully. The team members' activities are then given by a process of the teamwork net. The technical constructions we presented in this paper are compatible with the process notions, mainly since the projection of processes along injections are given by pullbacks as well.

Property preserving rules Especially in the area of workflow modeling properties like safety and liveness are of importance. In [11,12] inheritance preserving rules and property preserving rules, respectively, are formalized, so that restructuring of workflows preserves properties. Thus, another interesting aspect of future work is to study an integration of preserving rules into the AHO net in

[2] The research project *Formal modeling and analysis of flexible processes in mobile ad-hoc networks* (forMA$_1$NET) of the German research Council.

Fig. 1.b. To do that, on the one hand the set of token rules would have to be restricted to these kinds of rules and on the other hand the firing conditions would have to be adequately specified.

References

1. Bottoni, P., De Rosa, F., Hoffmann, K., Mecella, M.: Applying Algebraic Approaches for Modeling Workflows and their Transformations in Mobile Networks. Mobile Information Systems **2**(1) (2006) 51–76
2. De Rosa, F., Malizia, A., Mecella, M.: Disconnection Prediction in Mobile Ad hoc Networks for Supporting Cooperative Work. IEEE Persavive Computing **4**(3) (2005)
3. van der Aalst, W.M.P., van Hee, K.: Workflow Management: Models, Methods, and Systems. MIT Press (2001)
4. Hoffmann, K.: Formal Approach and Applications of Algebraic Higher Order Nets. PhD thesis, Technical University Berlin (2006)
5. Ehrig, H., Padberg, J.: Graph grammars and Petri net transformations. In: Lectures on Concurrency and Petri Nets Special Issue Advanced Course PNT. Volume 3098 of Lecture Notes in Computer Science., Springer (2004) 496–536
6. Rozenberg, G.: Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations. World Scientific (1997)
7. van der Aalst, W.M.P., Weske, M.: The P2P Approach to Interorganizational Workflows. In: Proc. of 13th Intern. Conf. on Advanced Information Systems Engineering. Volume 2068 of Lecture Notes in Computer Science., Springer (2001) 140–156
8. Baresi, L., Maurino, A., Modafferi, S.: Workflow partitioning in mobile information systems. In: IFIP TC 8 Working Conference on Mobile Information Systems. (2004) 93–106
9. Maurino, A., Modafferi, S.: Partitioning rules for orchestrating mobile information systems. Personal and Ubiquitous Computing **9**(5) (2005) 291–300
10. Baldan, P., Corradini, A., Ehrig, H., Heckel, R.: Compositional Semantics of Open Petri Nets based on Deterministic processes. MSCS **15**(1) (2005) 1–35
11. van der Aalst, W.M.P., Basten, T.: Inheritance of workflows: an approach to tackling problems related to change. Theor. Comput. Sci. **270**(1-2) (2002) 125–203
12. Padberg, J., Urbasek, M.: Rule-Based Refinement of Petri Nets: A Survey. In: Advances in Petri nets: Petri Net Technologies for Modeling Communication Based Systems. Volume 2472 of Lecture Notes in Computer Science., Springer (2003) 161–196
13. Meseguer, J., Montanari, U.: Petri Nets are Monoids. Information and Computation **88**(2) (1990) 105–155
14. Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of Algebraic Graph Transformation. EATCS Monographs in Theoretical Computer Science. Springer (2006)
15. Biermann, E., Modica, T., Hoffmann, K.: Categorical foundation for layer consistency in AHO-net models supporting workflow management in MANETs. Technical Report 2006/13, TU Berlin, Fak. IV (2006)

A Formal Foundation

We use the algebraic notion of P/T-nets as introduced in [13]. A P/T-net is given by the set of transitions and the set of places and the pre- and post-domain function. $N = T \overset{pre}{\underset{post}{\rightrightarrows}} P^{\oplus}$, where $P^{\oplus}$ is the free commutative monoid over P, or the set of finite multisets over P. Net morphisms (f_P, f_T) map places to places and transitions to transitions, so that $pre_2 \circ f_T = f_P^{\oplus} \circ pre_1$ and $post_2 \circ f_T = f_P^{\oplus} \circ post_1$ where we use the homomorphism $f_P^{\oplus}$ generated over the set of places P. They preserve firing and they yield the category **PT** that is cocomplete and together with the class of injective net morphisms $\mathcal{M}$ a weak adhesive HLR category (see [14]). Place gluing morphisms are net morphisms (f_P, f_T), so that the mapping of the transitions is bijective, the mapping of the places is surjective.

Results for net transformations: The Church-Rosser Theorem states a local confluence in the sense of term rewriting. The Parallelism Theorem states that sequential or parallel independent transformations can be carried out either in arbitrary sequential order or in parallel, i.e. sequentially independent $N \overset{r_1}{\Longrightarrow} N_1$ and $N \overset{r_2}{\Longrightarrow} N_2$ yield $N_2 \overset{r_1}{\Longrightarrow} M$ and $N_1 \overset{r_2}{\Longrightarrow} M$ or $N \overset{r_1+r_2}{\Longrightarrow} M$. The Union Theorem states the compatibility of union and net transformations, where the union is given in terms of a pushout over a common interface net. Given a union $N_1 +_I N_2 = N$ and net transformations $N_1 \overset{r_1}{\Longrightarrow} M_1$ and $N_2 \overset{r_2}{\Longrightarrow} M_2$ then we have a parallel rule $r_1 + r_2$ and a parallel net transformation $N \overset{r_1+r_2}{\Longrightarrow} M$. $M = M_1 +_I M_2$ is then the union of M_1 and M_2 with the shared interface I, provided that the given net transformations preserve the interface I.

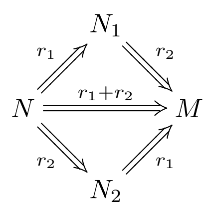


Fig. 12. Parallelism

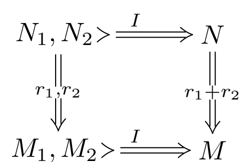


Fig. 13. Union

Proof of Theorem 1 (for details see [15]) : Cond. 1 allows constructing $\underline{W}_0 \overset{r^W}{\Longrightarrow} \underline{W}_1$, since the rule r^W is compatible with place refinement given by $pg_0 : \underline{W}_0 \to W_0$, and is applicable to $\underline{W}_0$ and applicable to W_0 via $\underline{W}_0$, so we have the adjacent diagram consisting of four pushouts. This definition is well-defined as place-gluing morphisms are pushout stable along injective net morphisms. Component wise construction of the pushout yields again an bijective transition mapping and a surjective place mapping.

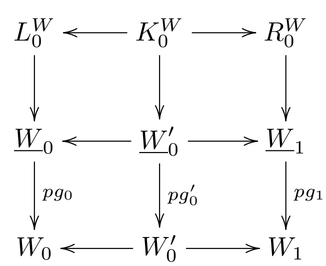
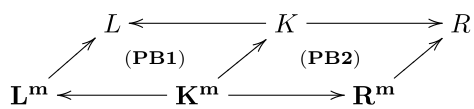


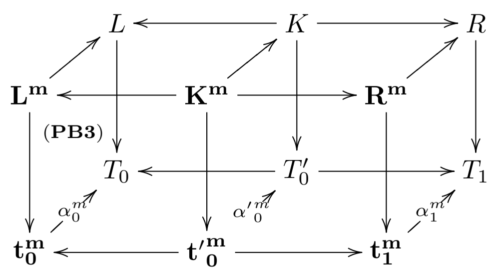
Fig. 14. Transformation step compatible with place refinement

Cond. 2 the independence of r^W and r^M of I means there are net morphisms a_W and a_M so that $I \overset{a_M}{\to} M_0' \to M_0 = I \to M_0$ and $I \overset{a_W}{\to} \underline{W}_0' \to \underline{W}_0 = I \to \underline{W}_0$.

Then the Union Theorem states that for transformations $M_0 \overset{r^M}{\Longrightarrow} M_1$ and $\underline{W}_0 \overset{r^W}{\Longrightarrow} \underline{W}_1$ there is the parallel rule $r = (r^M + r^W)$ together with the transformation $T_0 \overset{r}{\Longrightarrow} T_1$ so that we have the following pushout $T_1 = \underline{W}_{n_W} +_I M_{n_M}$. Since $I \overset{a_W}{\to} \underline{W}'_0 \to \underline{W}_0$ and $\underline{W}'_0 \to \underline{W}_0$ are injective, so a_W is injective as well. The same for a_M.

For each team member m the rule $r = (L \leftarrow K \rightarrow R)$ can be restricted to an activity rule $\mathbf{r^m} = (\mathbf{L^m} \leftarrow \mathbf{K^m} \rightarrow \mathbf{R^m})$, since we have the pullbacks (**PB1**) and (**PB2**) (due to Cond. 3). The application of an activity rule to an activity arrow $\mathbf{t_0^m} \overset{\alpha_0^m}{\to} T_0$ is based on the transformation of the underlying net $T_0 \overset{r}{\Longrightarrow} T_1$ given by the double pushout, i.e. the back squares below. Additionally the pullback (**PB7**) is given due to Cond. 4. Due to VK property of the weak adhesive HLR-category **PT** we then obtain the following diagram, where the front and the back are pushouts and the top, bottom and side squares are pullbacks.





The activity arrow $\mathbf{t'^m_0} \overset{\alpha'^m_0}{\to} T'_0$ is obtained by pullback construction, so the left bottom square is a pullback leading to the induced morphism $\mathbf{K^m} \to \mathbf{t'^m_0}$ so that all squares in the left cube become pullbacks by pullback -composition and -decomposition. Using the VK-property in the left cube with the back square being a pushout and all side squares being pullbacks we obtain that the left front square is a pushout as well.

$\mathbf{R^m} \to \mathbf{t_1^m} \leftarrow \mathbf{t'^m_0}$ is constructed as pushout over $\mathbf{R^m} \leftarrow \mathbf{K^m} \to \mathbf{t'^m_0}$. The activity arrow $\alpha_1^m : \mathbf{t_1^m} \to T_1$ is a induced pushout morphism. In the right cube due to the VK-property using the front and back squares being pushouts and the top and left side square being pullbacks we get that the right floor square and the right side square are pullbacks. And so we have the activity transformation $\mathbf{t_0^m} \overset{\mathbf{r^m}}{\Longrightarrow} \mathbf{t_1^m}$ due to the front pushouts above together with the activity arrow $\mathbf{t_1^m} \overset{\alpha_1^m}{\to} T_1$. $\checkmark$

B P/T-nets and rules

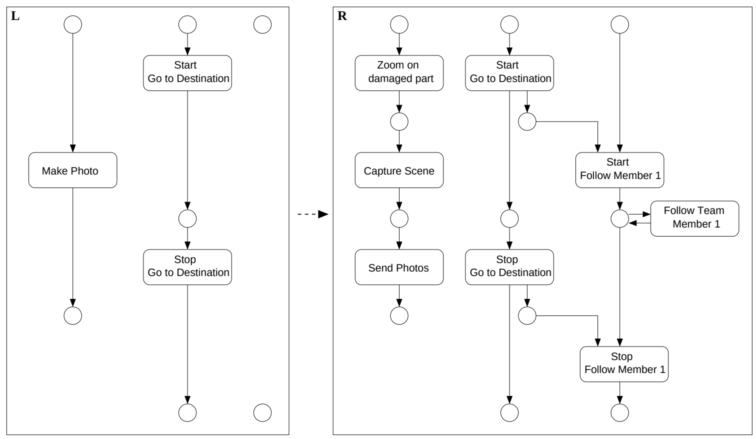
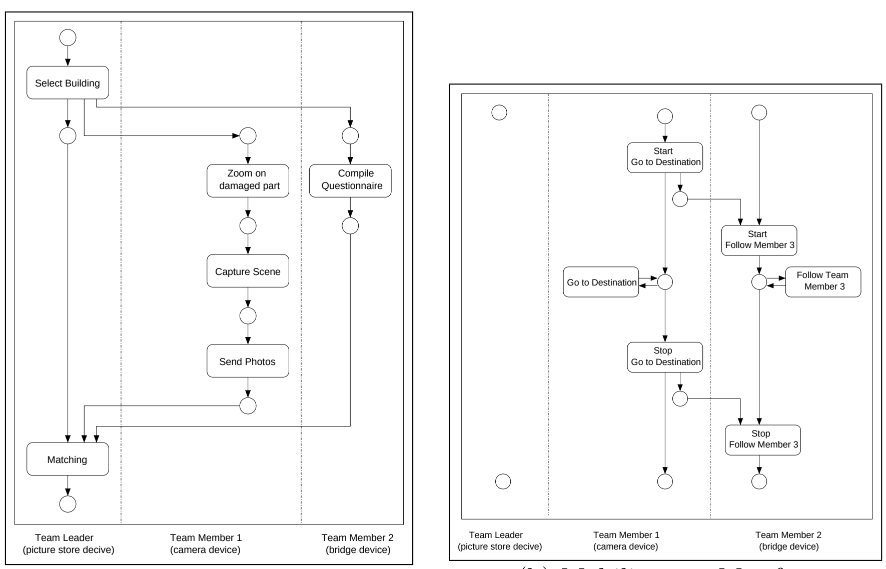


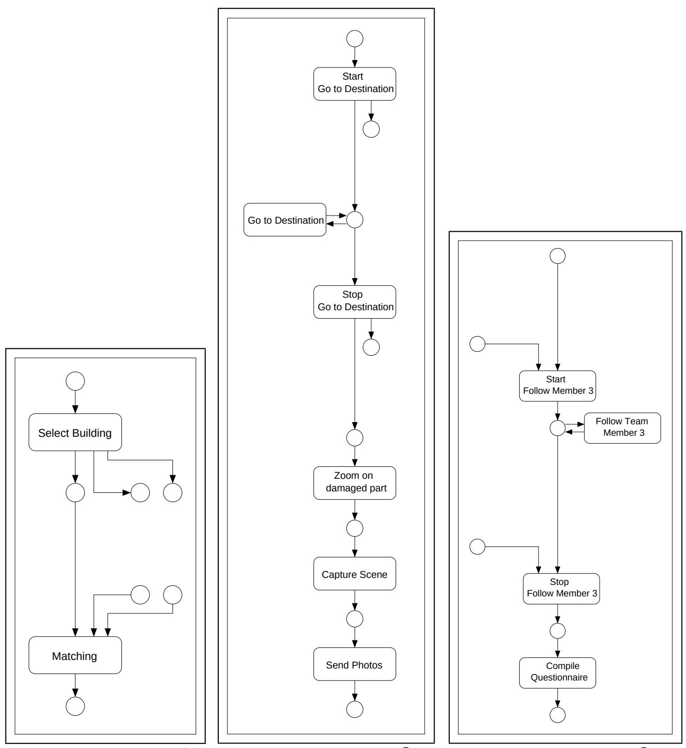
Fig. 15. Parallel rule $r = r_{photo} + r_{follow}$



(a) Workflow W_1 after application of rule r_{photo}

(b) Mobility net M_1 after application of rule r_{follow}

Fig. 16. P/T-nets in workflow and mobility layer after rule application



(a) P/T-net $\mathbf{t_1^1}$ (b) P/T-net $\mathbf{t_1^2}$ (c) P/T-net $\mathbf{t_1^3}$

Fig. 17. Team member nets in team layer after rule application

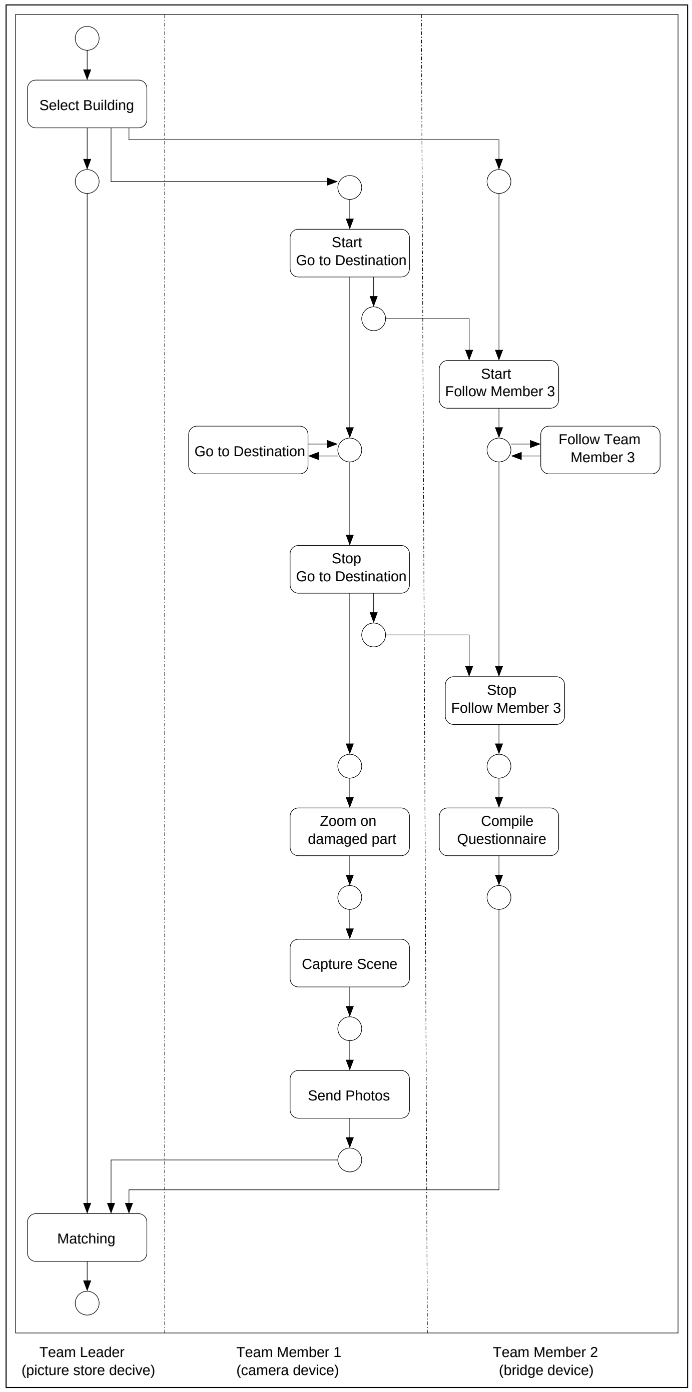


Fig. 18. Teamwork net T_1 after application of rules r_{photo} and r_{follow}