

Maintaining Consistency in Layered Architectures of Mobile Ad-Hoc Networks

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

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

Abstract. In this paper we present a layered architecture for modeling workflows in **Mobile Ad-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 requiring 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 Section 2. Formal modeling of workflows in MANETS using algebraic higher order nets (AHO nets) has been first introduced in [5]. AHO nets are Petri nets with 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 yields better and conciser models, since supporting the network connectivity has a much finer granularity than the more

* This work has been partly funded by the research project forMA₁NET (see <http://tfs.cs.tu-berlin.de/formalnet/>) of the German research Council.

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, backtracking could be considered 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. the intended workflow of the whole team is covered by the individual activities of the team members. Another important consistency notion concerns the distributed behavior that means 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.

The formal approach presented in this paper was developed in strong collaboration with some research projects¹ where an adaptive workflow management system for MANETS, specifically targeted to emergency scenarios, is partly realized resp. going to be implemented. Section 2 introduces an exemplary scenario of disaster management to illustrate our notions and results, while in Section 3 we discuss our approach to model workflows in MANETS using a layered architecture. The formalization to maintain consistency in layered architectures can be found in Section 4. Finally, we discuss future work.

¹ MOBIDIS - <http://www.dis.uniroma1.it/pub/mecella/projects/MobiDIS>, MAIS - <http://www.mais-project.it>, IST FP6 WORKPAD - <http://www.workpad-project.eu/>

2 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. 1(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.

In the following we exemplarily present P/T-nets called token nets for our scenario. As described above, Fig. 1(a) presents the workflow W_0 that has to be cooperatively executed by the team. The dashed lines are an additional information illustrating the relation among tasks and team members and are not a part of the P/T-net itself. There is a corresponding workflow \underline{W}_0 where the place \mathbf{p} is represented by two places $\mathbf{p1}$ and $\mathbf{p2}$ (and similar place \mathbf{p}') to integrate movement activities. In Fig. 1(b) the token net M_0 presents the mobility aspect of team member 1 stating that he/she has to go to the selected destination while team member 2 stays put. Finally, in Fig. 2 there are three separate nets for the team layer showing the local view of each team member onto the workflow and the mobility net.

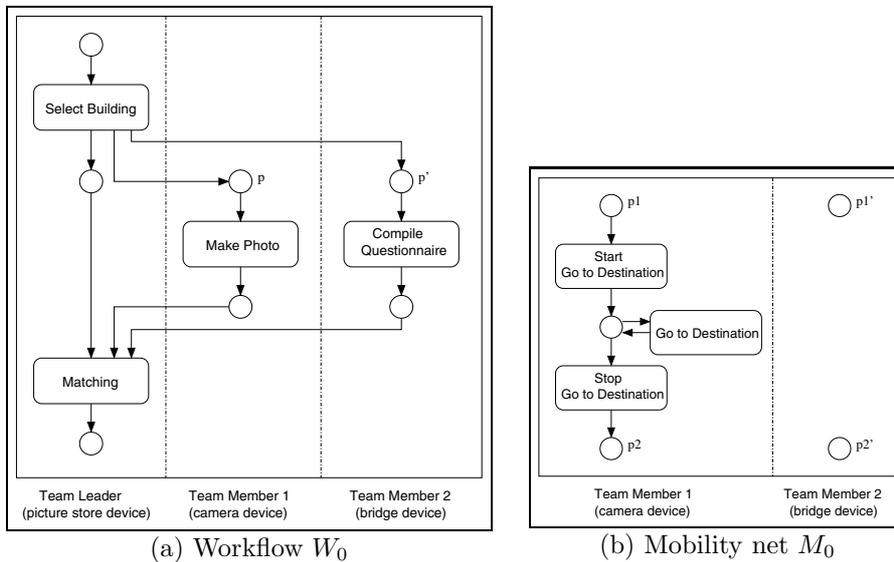


Fig. 1. P/T-nets in the workflow and the mobility layer

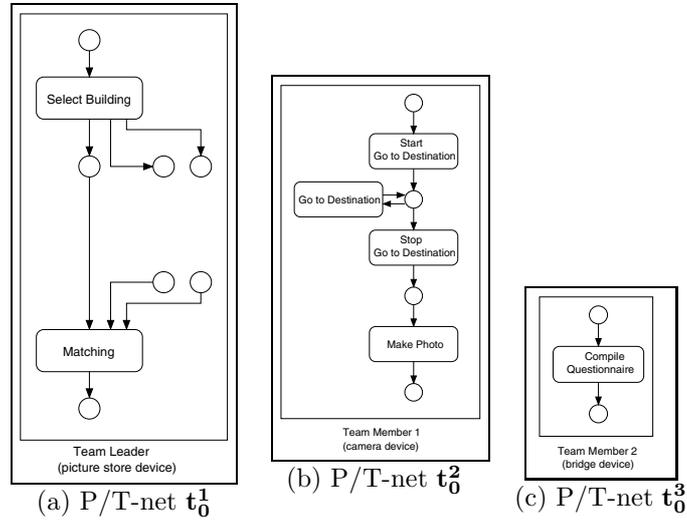


Fig. 2. Team member nets in the team layer

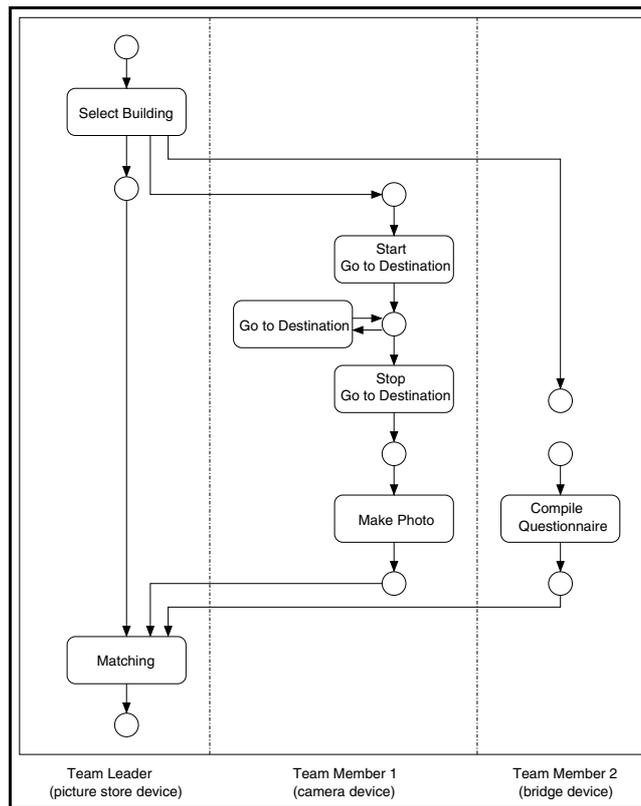


Fig. 3. Teamwork net T_0

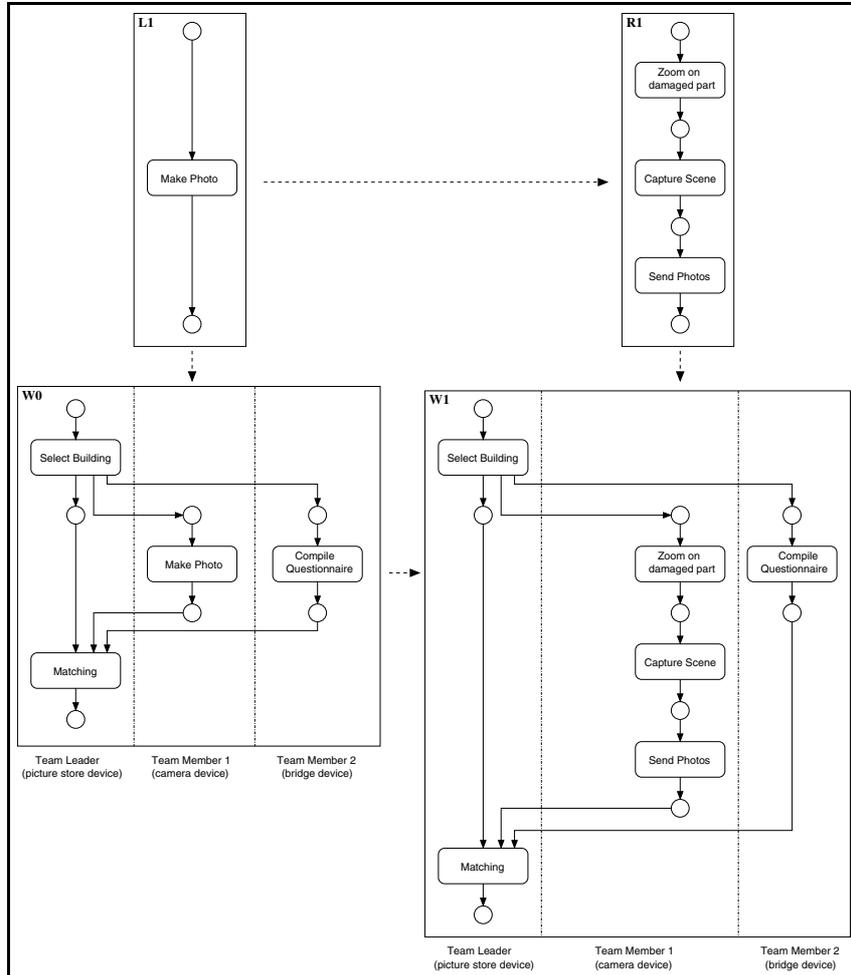


Fig. 4. Rule r_{photo} in the workflow layer and its application

To maintain consistency in a layered architecture first of all the teamwork net T_0 (see Fig. 3) has to be produced by gluing the workflow W_0 and the mobility net M_0 (see Fig. 1). 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. 2) is achieved by an inclusion into the teamwork net, called activity arrow that realizes the relation of team members to their activities. Thus, we start with a consistent layer environment (see Section 4).

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 ensure 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.

Thus, according to the requirements of our scenario, the structure of the token nets in Figs. 1 and 2 has to be changed to react to incoming events, e.g. to avoid a "going-out-of-range"-situation. 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 [12] that were generalized to net transformation systems [7]. The existence of several consistency and compatibility results for net transformation systems is highly profitable for 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. A transformation from a P/T-net N_0 to a P/T-net N_1 by a rule r is denoted by $N_0 \xrightarrow{r} N_1$.

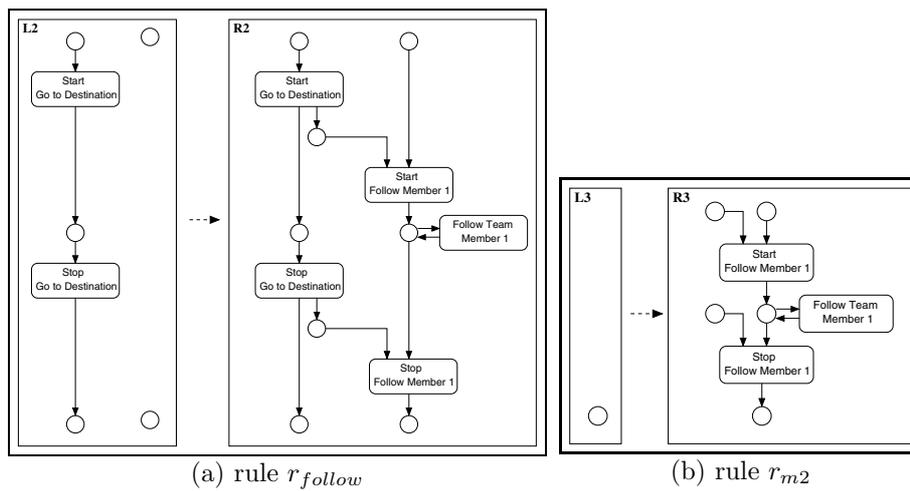


Fig. 5. Rules for the mobility and the team layer

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. 1(a) is changed using the rule r_{photo} depicted in the upper row in Fig. 4 resulting in the new workflow W_1 in Fig. 4. Assume a probable disconnection while team member 1 is going to the previously selected destination. Here the rule r_{follow} in Fig. 5(a) maintains the network connectivity by adding movement activities for team member 2 to follow team member 1, i.e. $M_0 \xrightarrow{r_{follow}} M_1$. 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. 5(b) for the team layer to change the structure of the token net \mathbf{t}_0^3 (see Fig. 2(c)), i.e. $\mathbf{t}_0^3 \xrightarrow{r_{m2}} \mathbf{t}_1^3$. Note that these rules are applied independently so that consistent transformations cannot be guaranteed

in general. But we present in Section 4 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.

3 Layered Architectures of Mobile Ad-Hoc Networks

In [5] 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. 6(a) to get a more adequate model. We separate movement activities from general activities and allow 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 P/T-nets and transformation rules. The advantage is that we exploit some form of control on rule application by assigning a set of rules to a specific layer. Under these restrictions transformations can be realized in a specific layer of our model.

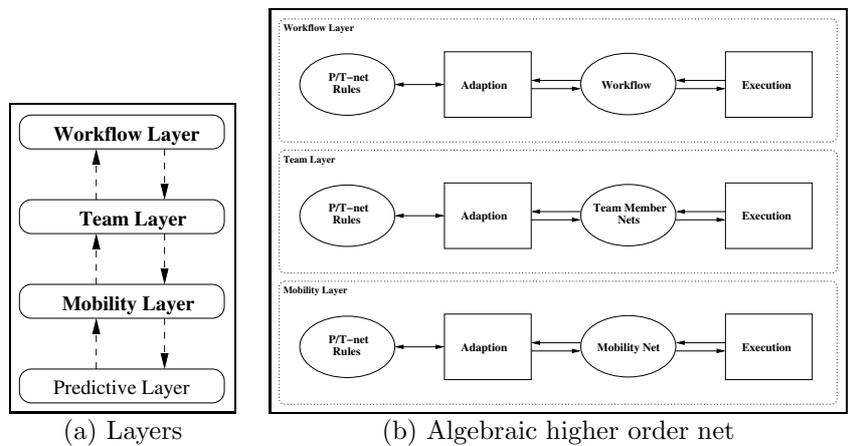


Fig. 6. Layered architecture for supporting cooperative work on MANETs

The predictive layer signals probable disconnections to the upper mobility layer. The predictive layer implements a probabilistic technique [6] 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-net² the cooperative work of the team but excludes movement activities.

The layered architecture is formalized by a layered AHO net (see Fig. 6(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 to some incoming events. In general, AHO nets [8] combine an algebraic data type part and Petri nets by the inscription of net elements with terms over the given data structure. Technically, the data type part of the AHO net in Fig. 6(b) consists of P/T-nets, the well-known token game, rules and rule-based transformation in the sense of the double pushout (DPO) approach [7], where all of them 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 token nets and token rules. Token rules are static, i.e. 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 **Adaption** changes the structure of a corresponding token net according to an appropriate token rule (for details we refer to [8]). 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 transformation rules.

The P/T-nets presented in Section 2 are possible markings of our AHO net model in Fig. 6(b). Fig. 1(a) depicts the token net W_0 for the workflow layer, i.e. it represents the current marking of the place **Workflow** in Fig. 6(b). For the mobility layer in Fig. 1(b) the token net M_0 is depicted (token net on the place **Mobility Net** in Fig. 6(b)). Finally, the team member nets in Fig. 2 are a marking on the place **Team Member Nets** in our model. 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 **Execution** in our model (see Fig. 6(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. Analogously, for each layer a specific transformation rule is depicted in Figs. 4 and 5.

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 Section 2 we have rules and transformations for changes at the level of the workflow layer, of the mobility layer and for

² Note that we have a P/T-net that describes the workflow, but this needs not be a workflow net in the sense of [14].

changing the individual activities of the team members. These rules and transformations allow the refinement of the token nets 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 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 Subsection 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 proven mathematically in [4].

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 $m = 1, \dots, n$ a P/T-net \mathbf{t}^m representing their individual activities. Moreover, we have 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. 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 this 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 [7] (see for example the P/T-net rules in Figs. 4 and 5 in Section 2). 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 the AHO net depicted in Fig. 6(b). Firing in this AHO net causes the transformation of the nets W , M and \mathbf{t}^m . Consistency of such a layered AHO net means in a broad sense that W , M and \mathbf{t}^m have to be related as depicted in Fig. 7. The interface net I is assumed to be fixed throughout this paper, but it is straightforward to adapt our constructions to changing the interface as well.

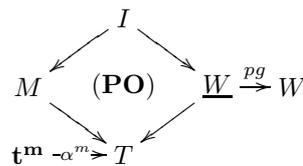


Fig. 7. Consistent layer environment

Definition 1. A consistent layer environment according to the layers in Fig. 6(b) is given for the the workflow W , the mobility net M and team members' nets $\mathbf{t}^1, \dots, \mathbf{t}^n$ if the following conditions are satisfied:

1. In order to have refinement of places in W with subnets of M we allow replacing W by $\underline{W} \xrightarrow{pg} W$, where pg is a place gluing morphism (bijective on transitions and surjective on places).

2. There is the fixed interface net I included in M and \underline{W} , so that a teamwork net T is obtained by the gluing of M and \underline{W} along I , written $T \stackrel{=}{=} M +_I \underline{W}$.
3. There are activity arrows for each team member $\mathbf{t}^1 \xrightarrow{\alpha_1} T, \dots, \mathbf{t}^n \xrightarrow{\alpha_n} T$ that are net morphisms relating a team member's activities – given by the net \mathbf{t}^m – to the teamwork net T .

The nets $W, M, (\mathbf{t}^1, \dots, \mathbf{t}^n)$ and T correspond in our example to the nets given in Figs. 1, 2 and 3, respectively. \underline{W}_0 is obtained from W_0 by splitting \mathbf{p} in Fig. 1(a) into two places $\mathbf{p1}$ and $\mathbf{p2}$ that are unconnected (and similar \mathbf{p}' into $\mathbf{p1}'$ and $\mathbf{p2}'$). I consists of the places $\mathbf{p1}, \mathbf{p2}, \mathbf{p1}'$ and $\mathbf{p2}'$ included in \underline{W}_0 and also in M_0 in Fig. 1(b) where these places are the entry and exit places.

4.2 Transformations at Different Layers

As mentioned before we want to model changes using transformation rules 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 (see [7]) as illustrated in Section 2.

Example 1. Starting at a consistent layer environment firing of the AHO net transitions **Adaption** in Fig. 6(b) yields various transformations in the different layers. So, at the level of the tokens (i.e. nets and rules) we then have 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 $M_0 \xrightarrow{r^M} M_1, W_0 \xrightarrow{r^W} W_1$ as well as rules for each team member $\mathbf{t}_0^1 \xrightarrow{r^1} \mathbf{t}_1^1, \mathbf{t}_0^2 \xrightarrow{r^2} \mathbf{t}_1^2$ and $\mathbf{t}_0^3 \xrightarrow{r^3} \mathbf{t}_1^3$. This is the situation as discussed in Section 2 if we assume additional rules for the team leader and team member 1.

According to the discussion in Section 1, we now need conditions that allow to maintain consistency. We have to obtain the teamwork net that integrates the changes induced by the transformations above. The results for net transformations yield a variety of independence conditions for the sequential and parallel application of rules and for the compatibility with pushouts (see [7]). Subsequently we develop the conditions for maintaining layer consistency based on transformations at the mobility and the workflow layer. Later, in Corollary 1, we assume not only transformation steps, but transformation sequences.

Let there be the transformations $W_0 \xrightarrow{r^W} W_1$ and $M_0 \xrightarrow{r^M} M_1$. We first need to ensure *compatibility with place refinement*. This means that the rule r^W is also applicable to \underline{W}_0 and there exists a place-gluing morphism $pg_1 : \underline{W}_1 \rightarrow W_1$, such that the diagram (1) in Fig. 9 commutes.

Provided the *preservation of the interface I* , that is, the applications of the rules r^W and r^M are independent of I , 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 $T_0 \xrightarrow{r} T_1$, 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 $T_0 \xrightarrow{r} T_1$ to the transformations $\mathbf{t}_0^m \xrightarrow{r^m} \mathbf{t}_1^m$ for each team member $m = 1, \dots, n$. Since

the team members' activities are represented by activity arrows, the rules have to be compatible with arrows. The *existence of activity rules* 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 .

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 . The *conformance of activity rules and team member nets* means that \mathbf{L}^m is additionally the pullback of \mathbf{t}_0^m and L , and the application of an activity rule \mathbf{r}^m to a team member net \mathbf{t}_0^m yields the transformation $\mathbf{t}_0^m \xrightarrow{\mathbf{r}^m} \mathbf{t}_1^m$.

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

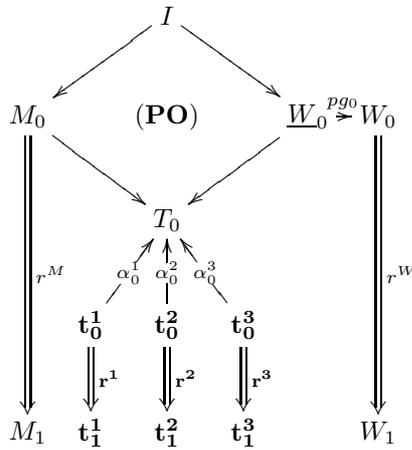


Fig. 8. State after some transformations

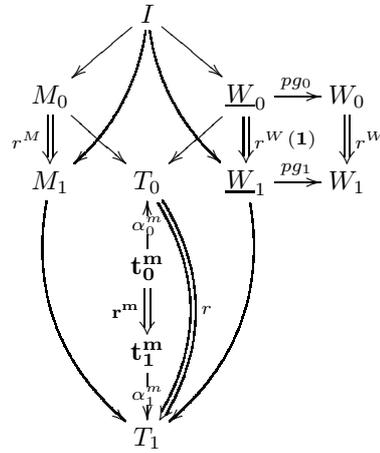


Fig. 9. A new consistent layer environment for $m = 1, 2, 3$

Theorem 1 (Stepwise Consistency Maintenance)

Given a consistent layer environment $T_0 = M_0 +_I \underline{W}_0$ with the place gluing $\underline{W}_0 \xrightarrow{p_{g_0}} W_0$ and the activity arrows $\mathbf{t}_0^m \xrightarrow{\alpha_0^m} T_0$ for each member $m = 1, \dots, n$, then the transformations $W_0 \xrightarrow{r^W} W_1$, $M_0 \xrightarrow{r^M} M_1$ and the transformations $\mathbf{t}_0^m \xrightarrow{\mathbf{r}^m} \mathbf{t}_1^m$ yield again a consistent layer environment $T_1 = M_1 +_I \underline{W}_1$ with the place gluing $\underline{W}_1 \xrightarrow{p_{g_1}} W_1$ and the activity arrows $\mathbf{t}_1^m \xrightarrow{\alpha_1^m} T_1$ for each m , provided the layer consistency conditions hold:

1. compatibility with place refinement, i.e. the rule r^W is compatible with the morphism p_{g_0} ,
2. preservation of the interface I , i.e. the application of the rules r^W and r^M is 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. $\mathbf{t}_0^m \xrightarrow{r^m} \mathbf{t}_1^m$ is compatible with $T_0 \xrightarrow{r} T_1$.*

Proof. It is shown in [4] that given the four layer consistency conditions above, the properties for a consistent layer environment are fulfilled: existence of place gluing morphism, compatibility of parallel rule application and gluing of W and M , and existence of activity arrows for each team member net.

Example 2. Considering the example in Section 2, outlined in Fig. 8 we have the following situation: The rule r_{photo} is compatible with place refinement because it preserves all involved places. For the same reason, the rules r_{photo} and r_{follow} are independent of the interface given by the overlapping of the workflow \underline{W}_0 and the mobility net M_0 and we obtain the parallel rule r consisting of both r_{photo} and r_{follow} .

In a next step we focus on the rule r_{m2} in Fig. 5(b) that 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} ; on the other hand the transformation $\mathbf{t}_0^3 \xrightarrow{r_{m2}} \mathbf{t}_1^3$ is compatible with the transformation $T_0 \xrightarrow{r} T_1$, because there is a corresponding inclusion of the resulting token net \mathbf{t}_1^3 into the teamwork net T_1 . Thus, we have the pushout $T_1 = M_1 +_I \underline{W}_1$ and the construction of the activity rule for each team member yields the activity arrows $\mathbf{t}_1^m \xrightarrow{\alpha_1^m} T_1$. So, we obtain the consistent layer environment depicted in Fig 9, where $r^W = r_{photo}$ and $r^M = r_{follow}$.

If we allow transformation sequences instead of transformation steps in Theorem 1 we may obtain inconsistent states. For recovery of consistency we need additional conditions. At the different layers rule application need 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 $(\mathbf{t}_0^m \xrightarrow{\alpha_0^m} T_0 = M_0 +_I \underline{W}_0 \xrightarrow{pg_0} W_0)$, and transformation sequences $M_0 \xrightarrow{*} M_{n_M}$ via r_i^M and $W_0 \xrightarrow{*} W_{n_W}$ via r_j^W and the transformation steps $\mathbf{t}_0^m \xrightarrow{r^m} \mathbf{t}_1^m$ leading to a possibly inconsistent state (see Fig. 10). It is possible to get an intermediate consistent layer environment $(\mathbf{t}_1^m \xrightarrow{\alpha_1^m} T_1 = M +_I \underline{W} \xrightarrow{pg} W)$ and a next consistent layer environment $\mathbf{t}_2^m \xrightarrow{\alpha_2^m} T_2 = M_{n_M} +_I \underline{W}_{n_W} \xrightarrow{pg_{n_W}} W_{n_W}$ (see Fig. 11) if the following conditions hold:*

1. *The transformation sequence $M_0 \xrightarrow{*} M_{n_M}$ can be decomposed, such that $(M_0 \xrightarrow{*} M_{n_M}) = (M_0 \xrightarrow{r^M} M \xrightarrow{\bar{r}^M} M_{n_M})$ for suitable rules r^M and \bar{r}^M .*
2. *The transformation sequence $W_0 \xrightarrow{*} W_{n_W}$ can be decomposed, such that $(W_0 \xrightarrow{*} W_{n_W}) = (W_0 \xrightarrow{r^W} W \xrightarrow{\bar{r}^W} W_{n_W})$ for suitable rules r^W and \bar{r}^W .*

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 inverse order. Then the AHO net in Fig. 6(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. 6(b) may allow only rule application which 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 Section 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 immediately rises the question of consistency. 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 [15, 3, 10] 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.

In this paper we present the first results of a line of research³ concerning formal modeling and analysis of MANETS. So, there is a large amount of most interesting and relevant open questions directly related to the work presented here. The behavior of token nets has been treated in previous papers [5] and has been excluded here deliberately. Nets in different layers have their own behavior that is executed by firing the corresponding transitions in the AHO net (see Fig. 6(b)). This directly leads to a challenging consistency issue, namely how the individual processes relate to each other. A solution would be to use the theory for open nets [2]. Other relevant notions of consistency concern e.g. the consistency between each team member's activities and the complete team work. It should be ensured

³ The research project *Formal modeling and analysis of flexible processes in mobile ad-hoc networks* (forMA₁NET) of the German research Council.

that the team members' activities together cover the complete team work. Again, team consistency has to be maintained during transformations in the different layers. Especially in the area of workflow modeling, properties like safety and liveness are of importance. In [13, 11] 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 Fig. 6(b).

We plan to develop a tool for our approach. For the application of net transformation rules, this tool will provide an export to AGG [1], a graph transformation engine as well as a tool for the analysis of graph transformation properties like termination and rule independence. Furthermore, the token net properties could be analyzed using the Petri Net Kernel [9], a tool infrastructure for Petri nets different net classes.

References

1. AGG Homepage. <http://tfs.cs.tu-berlin.de/agg>.
2. P. Baldan, A. Corradini, H. Ehrig, and R. Heckel. Compositional Semantics of Open Petri Nets Based on Deterministic Processes. *MSCS*, 15(1):1–35, 2005.
3. L. Baresi, A. Maurino, and S. Modafferi. Workflow partitioning in mobile information systems. volume 158 of *IFIP International Federation for Information Processing*, pages 93–106, 2005.
4. E. Biermann, T. Modica, and K. Hoffmann. Categorical Foundation for Layer Consistency in AHO-Net Models Supporting Workflow Management in MANETs. Technical Report 2006/13, TU Berlin, Fak. IV, 2006.
5. P. Bottoni, F. De Rosa, K. Hoffmann, and M. Mecella. Applying Algebraic Approaches for Modeling Workflows and their Transformations in Mobile Networks. *MIS*, 2(1):51–76, 2006.
6. F. De Rosa, A. Malizia, and M. Mecella. Disconnection Prediction in Mobile Ad hoc Networks for Supporting Cooperative Work. *IEEE Pervasive Comp.*, 4(3):62–70, 2005.
7. H. Ehrig and J. Padberg. Graph Grammars and Petri Net Transformations. volume 3098 of *LNCS*, pages 496–536. Springer, 2004.
8. K. Hoffmann, T. Mossakowski, and H. Ehrig. High-Level Nets with Nets and Rules as Tokens. volume 3536 of *LNCS*, pages 268–288. Springer, 2005.
9. E. Kindler and M. Weber. The Petri Net Kernel - An Infrastructure for Building Petri Net Tools. *Software Tools for Technology Transfer*, 3(4):486–497, 2001.
10. A. Maurino and S. Modafferi. Partitioning Rules for Orchestrating Mobile Information Systems. *Personal and Ubiquitous Computing*, 9(5):291–300, 2005.
11. J. Padberg and M. Urbasek. Rule-Based Refinement of Petri Nets: A Survey. volume 2472 of *LNCS*, pages 161–196. Springer, 2003.
12. G. Rozenberg. *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*. World Scientific, 1997.
13. W. M. P. van der Aalst and T. Basten. Inheritance of Workflows: An Approach to Tackling Problems Related to Change. *TCS*, 270(1-2):125–203, 2002.
14. W. M. P. van der Aalst and K. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT Press, 2002.
15. W. M. P. van der Aalst and M. Weske. The P2P Approach to Interorganizational Workflows. volume 2068 of *LNCS*, pages 140–156. Springer, 2001.