Integrated Design and Process Technology, **IDPT-2007**
Printed in the United States of America, June, 2007

Flexible Modeling of Emergency Scenarios using Reconfigurable Systems

K. Hoffmann, H. Ehrig, J. Padberg

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

ABSTRACT

In emergency scenarios we can obtain a more effective coordination among team members constituting a mobile ad hoc network (MANET) through the use of reconfigurable systems. This means that cooperative work can be adequately modeled by low level and high level Petri nets with initial markings and the net structure can be adapted to new requirements of the environment during run time by a set of rules.

In this paper we give main requirements for flexible processes in MANETs and show how to realize them using the formal notions of reconfigurable systems. The main part presents a case study in the area of emergency management and demonstrates the advantages of our approach which allows the dynamic adaption of processes in mobile environments. In this context we also discuss the main results achieved for reconfigurable systems and outline some interesting aspects of future work.

I. INTRODUCTION

As the adaptation of systems to changing environments gets more and more important processes that can be modified at run time have become a significant topic in the recent years especially in the area of mobile ad hoc networks (MANETs). MANETs are networks of mobile devices that communicate with one another via wireless links without relying on an underlying infrastructure e.g. as in emergency/disaster scenarios where an effective coordination is crucial among teams and team members to stabilize the situation and reduce the probability of secondary damage as well as to provide emergency assistance for victims.

As noticed in the context of the research project WORKPAD[1] the situation in such scenarios is complicated by the fact that the common goal is reached by different teams belonging to different organizations. Moreover each team member should carry on specific activities while the different teams collaborate through the interleaving of all the different processes. Normally processes in mobile environments are not fixed once and for all at build time but constantly adapted at run time e.g. to predict situations of disconnection or to restructure specific parts and activities.

For the effective coordination among teams and team members a suitable process definition language is desirable that supports an adequate modeling of processes and their modifications. But as recognized e.g. in the context of the graduate school METRIK[2] the workflow oriented view on processes in emergency/disaster scenarios is a novel line of research and up to now there exists only a few approaches especially designed for such an application area.

In [2, 5, 6, 8, 15] the rule based approach of reconfigurable place/transition (P/T) systems is introduced, so that the modification of processes is realized at run time by a set of rules. The formalism of algebraic higher order systems follows the paradigm "nets and rules as tokens" and represents a meta model for reconfigurable P/T systems where process execution and process modification is distinguished by the use of specific transitions.

This paper is organized as follows: in Section II we give a characterization of main requirements for flexible processes in emergency/disaster scenarios in order to review the formal notions and results of reconfigurable systems in Section III and compare them with the listed requirements. To demonstrate the advantages of our approach we illustrate in Section IV reconfigurable systems by a case study in the area of

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

[1] www.workpad-project.eu
[2] metrik.informatik.hu-berlin.de/grk-wiki

pipeline emergencies. Finally in Section V we conclude with a discussion of some interesting aspects of future work.

II. Flexible Processes in MANETs

This section presents a characterization of main requirements for flexible processes in emergency/disaster scenarios. Based on the fundamental requirements for process definition languages called perspective in [25] these perspectives are improved to fit in our intended application area. Summarizing a process definition language should cover the process perspective, informational perspective, organizational perspective, functional perspective, and operational perspective [25].

The *process perspective* concentrates on the control flow, i.e. the start conditions and the order of activities that have to be executed. The Workflow Management Coalition[3] identifies some basic types of relationship between activities: sequential, parallel, conditional, and iterative routing. Following the approach in [12] in a completely decentralized system as in MANETs each activity could be in addition in one of the following states :

- Received: a start conditions has arrived from the previous team member and is waiting until all conditions are true and the current team member is available to start running it.
- Initiated: a new process instance has just started, this is where the team member starts it because all start conditions are true.
- Running: the team member is running the activity.
- Aborted: the team member failed to complete the activity either because the team member is disconnected or for any other reason.
- Completed: the team member completed the activity.
- On-Hold: the activity is completed but the next team member is not available yet to receive his/her start conditions.
- Rejected: the team member rejects to complete the given activity.

Moreover in a mobile environment movement activities concerning the network connectivity can be separated from activities concerning the intended process.

The *informational perspective* concentrates on the data flow, so that data dependencies between activities are characterized by input and output parameters. On the one hand control data is used for process management purposes and on the other hand production data subsumes information objects like documents, questionnaires and forms. In MANETs information about the geographic area is especially important e.g. to localize positions of team members or to predict situations of disconnection.

The *organizational perspective* is typically defined by roles, groups and other artifacts clarifying organizational issues. Because in emergency/disaster scenarios different teams belong to different organizations, the inter-organizational aspect should be respected. In addition, in MANETs the network topology typically represented as topology graphs [1] both influences and is influenced by the process.

The *functional perspective* prescribes the decomposition of a process into smaller units often represented by a hierarchical structure.

Finally, the *operational perspective* depends on the technical environment, so that elementary operations are performed by resources and applications. Based on the observation in [12] in a mobile environment the team member can be on line, i.e. he/she can receive new work, or off line, where the team member is not available to receive new work. In this case new activities may be on hold until the team member returns on line or even allocated to alternative team members. Team members before permanently leave may notify this otherwise the team leader may decide to treat any other team member failing to respond as permanent. For activities where the team member is temporarily off line, the execution of the process will continue, if possible. In this case when the team member returns some synchronization may be required or alternatively the execution will have to wait until the team member returns.

From a practical point of view processes in MANETs often have to be restructured e.g. because of unforeseen events or to maintain the network connectivity resulting in a highly dynamical modification of processes. In [25] three issues to dynamic change of processes are addressed. By *constrained flexibility* certain properties should be preserved during process adaptation while *instance change* refers to the modification of process instances at run time. Finally *instance migration* are based on simultaneous changes of both process schemes and process instances.

[3]www.wfmc.org

In addition dynamic changes are grouped into ad hoc changes, i.e. changes are responses to unforeseen exceptions, and pre-planned and evolutionary changes, i.e. changes are known at build time (see e.g [18, 25]). Besides others in [19, 21] a minimal set of change operations are characterized:

- inserting a new activity where also bridging actions may be used to keep network connectivity,
- removing an existing activity,
- modifying the order of activities, and
- modifying activity properties like data requirements, underlying applications, temporal constraints, resource allocation, or reassignment of activities from one team or member to another.

Processes have to be analysed (see e.g. [25]) for *verification* purposes, so that some form of correctness criteria, i.e. different properties on a syntactical and/or semantical level, has to be satisfied and can be checked. In contrast *validation* verifies processes with respect to the intended and typically informally formalised process and *performance analysis* is realized by simulating processes to detect e.g. potential deadlocks or livelocks.

.

III. RECONFIGURABLE SYSTEMS

In this section we compare reconfigurable systems with the requirements listed in the last section and present the results achieved for reconfigurable place/transition (P/T) systems in [2, 5, 6, 8, 15].

A P/T system is a P/T net with an initial marking. P/T nets, P/T systems and their variants are an established process definition language (see e.g. [7, 23]) providing constructs of the process perspective. While P/T nets represent process schemes, P/T systems describe the behavior of process instances due to their initial markings. Activities are modeled by transitions while the control flow is reflected by arcs between places and transitions. Places can be seen as pre and post conditions for activities and source places with an empty pre domain can be used as start condition for the process. The Workflow Patterns Initiative[4] [24] presents a number of patterns for the relationship between activities following not only the basic types identified by the Workflow Management Coalition but also more advanced constructs.

[4]www.workflowpatterns.com

The concept of reconfigurable P/T systems was introduced for modeling changes of the net structure by rule based transformations while the system is kept running. For rule based transformations of P/T systems we use the framework of net transformations [3,5] following the double pushout (DPO) approach of graph transformation systems [20]. The basic idea behind net transformation is the stepwise development of P/T systems by given rules. Think of these rules as replacement systems where the left hand side is replaced by the right hand side while preserving a context. In reconfigurable P/T systems not only the follower marking can be computed but also the net structure can be changed by rule applications and we obtain new P/T systems that are more appropriate with respect to some requirements of the environment. In detail a reconfigurable P/T system $((PN_1, M_1), RULES)$ consists of a P/T system (PN_1, M_1), where PN_1 is a P/T net with initial marking M_1, and a set of rules $RULES$.

Rules and transformations in the DPO approach are based on morphisms preserving on the one hand firing steps and requiring on the other hand that the initial marking at corresponding places is increasing or even stronger. An application of a rule is called a transformation step and describes how an object is actually changed by the rule. In general a rule $prod = ((L, M_L) \xleftarrow{l} (K, M_K) \xrightarrow{r} (R, M_R))$ is given by three P/T systems called left hand side, interface and right hand side, respectively, and a span of two P/T morphisms l and r. We additionally need a match morphism $(L, M_L) \xrightarrow{m} (PN_1, M_1)$ that identifies the relevant parts of the left hand side (L, M_L) in the P/T system (PN_1, M_1). Now a direct transformation $(PN_1, M_1) \stackrel{(prod,m)}{\Longrightarrow} (PN_2, M_2)$ via $prod \in RULES$ and m can be constructed in two steps. We delete in a first step those elements from (PN_1, M_1) which are identified by the match m but not preserved by the interface (K, M_K) leading to the intermediate P/T system (PN_0, M_0). In a second step we glue together the P/T systems (PN_0, M_0) and (R, M_R) along the interface resulting in the new P/T system (PN_2, M_2).

The DPO approach does not allow the treatment of unmatched transitions at places which should be deleted. In this case the so called gluing condition forbids the application of rules. Furthermore items which are identified by a non injective match must be preserved by rule applications. Note that a positive check of the gluing condition makes sure that the intermediate P/T system is well defined.

The rule based approach of reconfigurable P/T systems supports dynamic changes in the sense that the concept of instance change is formalised by the application of appropriate rules realising the insertion of new activities, removing of existing activities or changing the order of activities. Because rules are fixed at build time the concept of reconfigurable P/T system supports pre-planned and evolutionary changes. To support constraint flexibility the set of rules can be restricted to property preserving rules [16], so that safety and liveness properties are preserved by rule applications.

The main result in [5] concerns the formal foundation for transformations of P/T systems based on the framework of adhesive high level replacement (HLR) systems [3, 4]. Adhesive HLR systems have been recently introduced as a new categorical framework for graph transformation in the DPO approach. They combine the well known framework of HLR systems with the framework of adhesive categories introduced in [13]. The main concept behind adhesive categories are the so called van Kampen squares. These ensure that pushouts along monomorphisms are stable under pullbacks and, vice versa, that pullbacks are stable under combined pushouts and pullbacks. Note that a pushout can be seen as a gluing construction of two objects over a specific interface, while a pullback is dual to a pushout in the sense that a pullback construction extracts the common part of two objects. In the case of adhesive HLR categories the class of all monomorphisms is replaced by a subclass of monomorphisms closed under composition and decomposition.

Within the framework of adhesive HLR systems there are many interesting results concerning the applicability of rules, the embedding and extension of transformations, parallel and sequential dependence and independence, and concurrency of rule applications. The concept of parallel independence states that two transformation steps are not in conflict while two consecutive transformation steps are sequentially independent if they are not causally dependent. Provided that the relevant conditions are satisfied two alternative transformation steps may be swapped and each of them can still be applied after the other has been performed. Since we have shown in [5] that P/T systems form a weak adhesive HLR category, we can apply these results to reconfigurable P/T systems.

Based on the observation of parallel and sequential independence of rule applications the main results in [6] deals with conflict situations between transformation and token firing. The traditional concurrency situation in P/T systems without capacities is that two transitions with overlapping pre domain are both enabled and together require more tokens than available in the current marking. As P/T systems can evolve in two different ways the notions of conflict and concurrency become more complex. Assume that a given P/T system represents a certain system state. The next evolution step can be obtained not only by token firing but also by the application of one of the rules available. Hence the question arises whether each of these evolution steps can be postponed after the realization of the other, yielding the same result, and if they can be performed in a different order without changing the result.

In [6] we have presented conditions for (co-)parallel and sequential independence and we have shown that in specific cases firing and transformation steps can be performed in any order, yielding the same result. We have correlated these conditions, i.e. that parallel independence implies sequential independence and, vice versa, sequential (coparallel) independence implies parallel and coparallel (parallel and sequential) independence. The advantage of the presented conditions is that they could be checked at a syntactical and local level instead of semantical and global one. Thus they are also applicable in the case of complex reconfigurable P/T systems.

In [8] we have introduced the paradigm "nets and rules as tokens" by a high level model with suitable data type part. The model called algebraic higher order (AHO) system exploits some form of control not only on rule application but also on token firing. In general an AHO system is defined by an algebraic high level net [14] with system places and rule places as for example shown in Fig. 1 where a marking can be given by suitable P/T systems and rules, respectively, on these places. For a detailed description of the data type part, i.e. the AHO SYSTEM-signature and corresponding algebra A, we refer to [8].

In the following we review the behavior of AHO systems according to [8]. With the symbol $Var(t)$ we

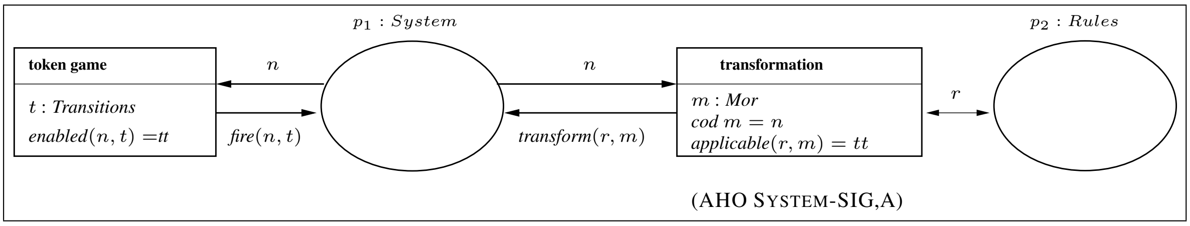

Fig. 1
ALGEBRAIC HIGHER ORDER SYSTEM

indicate the set of variables of a transition t, i.e. the set of all variables occurring in pre- and post domain and in the firing condition of t. The marking M determines the distribution of P/T systems and rules in an AHO system which are elements of a given higher order algebra A. Intuitively P/T systems and rules can be moved along AHO system arcs and can be modified during the firing of transitions. The follower marking is computed by the evaluation of net inscriptions in a variable assignment $v : Var(t) \rightarrow A$. The transition t is enabled in a marking M, if and only if (t, v) is consistent, that is if the evaluation of the firing condition is fulfilled. Then the follower marking after firing of transition t is defined by removing tokens corresponding to the net inscription in the pre domain of t and adding tokens corresponding to the net inscription in the post domain of t.

The transitions in the AHO system in Fig. 1 realize on the one hand firing steps and on the other hand transformation steps as indicated by the net inscriptions $fire(n, t)$ and $transform(r, m)$, respectively. To compute the follower marking of P/T systems we use the transition *token game* of the AHO system while the transition *transformation* is provided for changing the structure of P/T systems. In this way process execution and process modification is distinguished by these two transitions.

The pair (or sequence) of firing and transformation steps discussed in [6] is reflected by firing of the transitions one after the other in our AHO system. Thus these results are most important for the analysis of AHO systems.

Using P/T systems as tokens AHO systems focus on the process perspective. To integrate the informational perspective we can use high level nets as tokens themselves, i.e. the data type part is extended by algebraic high level nets and corresponding rules. Analogously the organizational and operational perspectives can be added following e.g. the approach in [22]. So activity properties like data requirements and the reassignment of activities from one team member to another can be modified by the applications of suitable rules. For the functional perspective the formalism of AHO systems can be adapted using the hierarchy concept of Coloured Petri Nets (see [10]).

To consider ad hoc changes of processes the modification of rule tokens requires an extension not only of the data type part but also of the net structure as introduced in [9], so that the definition of new rules by reusing existing rules is supported at run time by different operations like inheritance [17].

While the AHO system in Fig. 1 deals with one layer for reconfigurable P/T systems, in [15] we follow the observation that processes in MANETs consists of different aspects. Thus we separate movement activities from general activities and allow a local view of team members. This leads to an AHO system with different layers each of them equipped with its own P/T system and set of rules. Moreover the notion of layer consistent environment states that the views in each layer fit together realizing one form of instance migration. In [2] we extend this approach to allow the introduction of new team members by more advanced changes at each layer.

Because reconfigurable P/T systems and AHO systems are formalized on a rigorous mathematical foundation and have a clear formal semantics, several results as described above are provided to analyse systems in the sense of formal verification. These results present a line of research and there is a large amount of most interesting and relevant open questions directly related to the work presented here. We plan to develop a tool to support simulation and analysis aspects for our approach. For the application of net transformation rules this tool will provide an export to AGG[5], 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 [11], a tool infrastructure for Petri nets different net classes.

IV. Emergency/Disaster Scenario

In this section we illustrate the main idea of reconfigurable systems by a case study of a pipeline emergency scenario where an unknown source of a natural gas leak is detected in a residential area[6]: A postal worker delivering mail in a residential street smells a strong odor of gas. She immediately notifies the fire department. A single engine company is dispatched by the fire department with four firefighters leaded by one company officer. At the scene the postal worker meets the company officer and describes the problem. He calls the gas company and requests an additional law enforcement officers to control traffic into the area. While three firefighters evacuate the homes in the immediate area and afterwards deny entry to this area, another one reads the gas indicator and detects that the gas is highest in front of a home located on 114 Maple Street. After electricity and gas lines are shut off to each home the fire department stand by with fully charged hose lines and wait for the arrival of the gas company.

The cooperative process enacted by the firefighter company is depicted as P/T system (PN_1, M_1) in Fig. 2. To start the activities of the firefighter team the follower marking of the P/T system (PN_1, M_1) is computed by firing the *and-split*-transition and we obtain the new P/T system (PN_1, M_1') in Fig. 3.

Next we focus on dynamic changes while the process is running. The three firefighters responsible for the evacuation process need more detailed information how to proceed. So the company officer gives the instruction that first of all the residents are notified of the evacuation. Afterwards the firefighters should assist handicapped persons and guide all of them to the extend possible. To introduce the refinement of the *Evacuate homes*-transition into the P/T system (PN_1, M_1') we provide the rule $prod_{evacuate}$ in Fig. 4. The marking M_{L_1} of the P/T system in the left hand side of $prod_{evacuate}$ demands that the evacuation process is not yet started because there is one token in the pre domain of the *Evacuate homes*-transition. The application of the rule is given as follows: the match morphism m_1 is given by the obvious inclusion and identifies the relevant parts of the left hand side (L_1, M_{L_1}) of rule $prod_{evacuate}$ in (PN_1, M_1'); next, the *Evacuate homes*-transition is deleted and we obtain an intermediate P/T system (PN_0, M_0); then, the transitions *Notify residents*, *Assist handicapped persons* and *Guide persons* together with their (new) environment are added leading to the P/T system (PN_2, M_2) in Fig. 5. Thus we obtain the transformation step $(PN_1, M_1') \overset{(prod_{evacuate}, m_1)}{\Longrightarrow} (PN_2, M_2)$. Afterwards the firefighter company proceed with their activities and we obtain the P/T system (PN_2, M_2') in Fig. 6 by firing the corresponding transitions.

After the problem identification the odor of gas grows stronger and the firefighter takes an additional reading of the gas indicator and informs the company officer about the result, so that the company officer is able to determine if the atmosphere in the area is safe, unsafe, or dangerous. To extend our process by these additional activities we use the rule $prod_{analyse}$ in Fig. 7 where the marking M_{L_2} in the left hand side indicates that the problem location is identified. By the application of the rule we obtain the transformation step $(PN_2, M_2') \overset{(prod_{analyse}, m_2)}{\Longrightarrow} (PN_3, M_3)$ where the new P/T system (PN_3, M_3) is depicted in Fig. 8.

Based on the additional results of the gas indicator the company officer analyses that the atmosphere in this area is over the lower explosive limit and thereby more dangerous than expected. He determines that the best course of action is to call for additional resources to maintain the isolation perimeter and expand the area of evacuation as a precaution. So, in a next step the follower marking of the P/T system (PN_3, M_3) is computed by firing the *Additional reading*- and *Analyse*-transitions leading to the P/T system (PN_3, M_3') in Fig. 9. Afterwards the rule $prod_{expand}$ depicted in Fig. 10 is applied to the P/T system (PN_3, M_3') resulting in the new P/T system (PN_4, M_4) in Fig. 11.

Summarizing at the beginning our reconfigurable P/T system $((PN_1, M_1), \{prod_{evacuate}, prod_{analyse}, prod_{expand}\})$ consists of the P/T system (PN_1, M_1) in Fig. 2 and the set of rules depicted in Figs. 4, 7 and 10. Let the reconfigurable P/T system be the initial

[5] tfs.cs.tu-berlin.de/agg
[6] www.pipelineemergencies.com

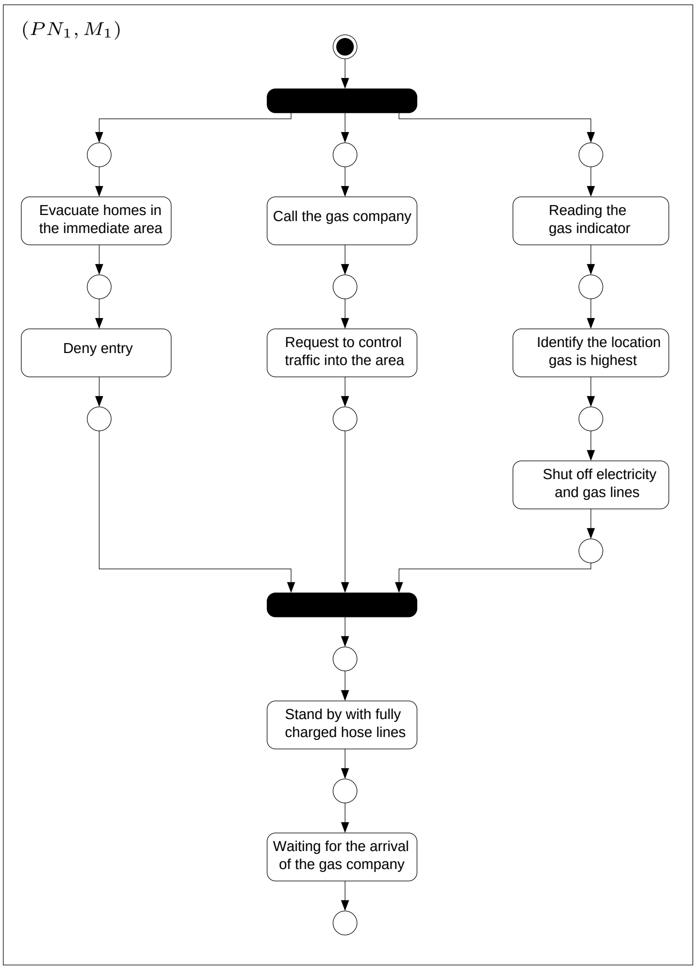

Fig. 2
PROCESS (PN_1, M_1)

marking of the AHO system in Fig. 1, i.e. the P/T system (PN_1, M_1) is on the place p_1 while the marking of the place p_2 is given by the set of rules. To compute the follower marking of the P/T system we use the transition *token game* of the AHO system. First the variable n is assigned to the P/T system (PN_1, M_1) and the variable t to the *and-split*-transition that is enabled, so that the firing condition is fulfilled. Due to the evaluation of the term $fire(n, t)$ we obtain the new P/T system (PN_1, M_1') in Fig. 2.

For changing the structure of P/T systems the transition *transformation* is provided in Fig. 1. Again we have to give an assignment v for the variables of this transition, i.e. variables n, m and r, where $v(n) = (PN_1, M_1')$, $v(m) = m_1$ is a suitable match morphism and $v(r) = prod_{evacuate}$ (see Fig. 4). The firing condition *cod* $m = n$ ensures that the codomain of the match morphism is equal to (PN_1, M_1') while the second condition $applicable(r, m)$ checks the gluing condition, i.e. if the rule $prod_{evacuate}$ is applicable with match m_1. Afterwards the transformation step is computed by the evaluation of the net inscription $transform(r, m)$ and the effect of firing the transition *transformation* is the removal of the P/T system (PN_1, M_1') from place p_1 and adding the P/T system (PN_2, M_2) in Fig. 5 to it.

Analogously we proceed with the computation of the follower markings and dynamic adaption of our process as described above. After several firing steps of the transitions *token game* and *transformation* we obtain the reconfigurable P/T system $((PN_4, M_4), \{prod_{evacuate}, prod_{analyse}, prod_{expand}\})$

consisting of the P/T system (PN_4, M_4) (see Fig. 11) and the original set of rules.

To analyse the reconfigurable P/T systems we apply the results presented in [6] and described in the previous section. For example the transformation step $(PN_1, M_1') \overset{(prod_{evacuate}, m_1)}{\Longrightarrow} (PN_2, M_2)$ is parallel independent of the firing step given by the *Reading gas indicator*-transition because the transition is not deleted by the transformation step and the marking of the P/T system (PN_1, M_1') is unchanged by the application of the rule $prod_{evacuate}$. Moreover the pair of transformation and firing steps is sequentially independent because the *Reading gas indicator*-transition is not created by the transformation step. Thus the pair of steps may be swapped and each of them can be applied after the other has been performed leading to the same result.

In the context of our AHO system in Fig. 1 this observation is reflected by an independent firing of the transitions *token game* and *transformation*, i.e. the sequential firing of these transitions leading to the same result independent of the order these transitions are fired.

The pair of consecutive steps given by firing the *and-split*-transition in (PN_1, M_1) and the transformation $(PN_1, M_1') \overset{(prod_{evacuate}, m_1)}{\Longrightarrow} (PN_2, M_2)$ is sequentially dependent because the marking of the left hand side of $prod_{evacuate}$ demands a token in the pre domain of the *Evacuate homes*-transition.

Further situations of independent and dependent firing and transformation steps are illustrated in Fig. 12 where, however, the traditional concurrency situation of transitions and transformations, respectively, is not shown. Note that e.g. the two consecutive transformations $(PN_1, M_1^2) \overset{(prod_{evacuate}, m_1)}{\Longrightarrow} (PN_2, M_2^2)$ and $(PN_2, M_2^2) \overset{(prod_{analyse}, m_2)}{\Longrightarrow} (PN_3, M_3^2)$ are sequentially independent because the overlapping of the right hand side of $prod_{evacuate}$ and the left hand side of $prod_{analyse}$ in (PN_2, M_2^2) is included in the intersection of the interfaces.

V. Conclusion

In this paper we have given main requirements for flexible processes in emergency/disaster scenarios in order to show that most of them are realized by reconfigurable systems, a rule based formalism based on the one hand on low level and high level Petri nets with a suitable marking and on the other hand on the categorical framework of weak adhesive high level replacement systems. As future work, it would be important to investigate and verify additional requirements necessary for flexible processes in emergency/disaster scenarios and mobile environments.

The main part of this paper presents the case study in the area of pipeline emergencies where dynamic changes of the process are realised at run time by rule applications to express the refinement and insertion of activities. Note that our processes focus on the intended activities and exclude movement activities because the network connectivity is assured due to the limited perimeter of the affected area and the use of cell phones and radio devices. Nevertheless, the scenario could be extended in such a way that the problem is located beyond the range of these equipment and several team members have to follow other ones to avoid a situation of disconnection.

One aspect of future work is integration of the informational and organizational perspectives into our formalism because within our case study these aspects become most relevant. In fact process modifications in our case study depend on the exchange of messages and data concerning a detailed instruction of the evacuation process, the results of reading the gas indicator and the final analysis of these results by the company officer. In addition the processes enacted by the gas company and the law enforcement officer have to be taken into account, so that the different teams collaborate through the interleaving of all the different processes to achieve the common goal.

References

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

[2] E. Biermann, K. Hoffmann, and J. Padberg. Layered Architecture Consistency for MANETs: Introducing New Team Members. In *Proc. Integrated Design and Process Technology (IDPT)*, 2007. accepted.

[3] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation*. EATCS Monographs in Theoretical Computer Science. Springer, 2006.

[4] H. Ehrig, A. Habel, J. Padberg, and U. Prange. Adhesive High-Level Replacement Systems: A New Categorical Framework for Graph Transformation. *Fundamenta Informaticae*, 74(1):1–29, 2006.

[5] H. Ehrig, K. Hoffmann, U. Prange, and J. Padberg. Formal Foundation for the Reconfiguration of Nets. Technical Report 2007-01, Technical University Berlin, Fak. IV, 2007.

[6] H. Ehrig, J. Padberg K. Hoffmann, U. Prange, and C. Ermel.

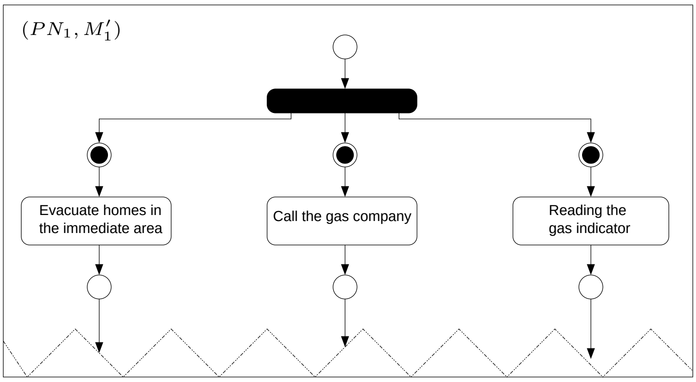

Fig. 3
RELEVANT PART OF PROCESS (PN_1, M_1')

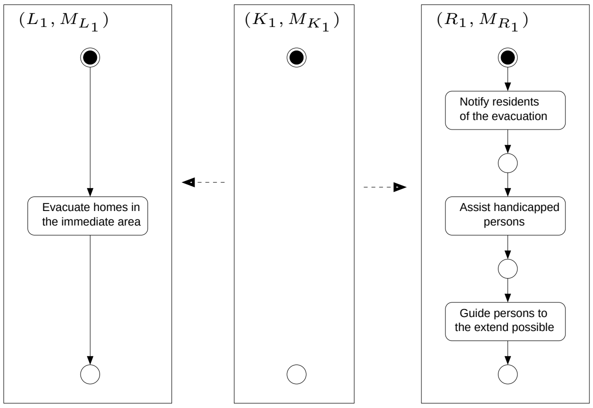

Fig. 4
RULE $prod_{evacuate}$

Independence of Net Transformations and Token Firing in Reconfigurable Place/Transition Systems. In *Proc. Application and Theory of Petri Nets (ATPN)*, 2007. accepted.

[7] C.A. Ellis. Information Control Nets: A Mathematical Model of Office Information Flow. In *Proc. Simulation, Measurement and Modelling of Computer Systems*, pages 225–240. ACM Press, 1979.

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

[9] K. Hoffmann, F. Parisi-Presicce, and T. Mossakowski. Higher-Order Nets for Mobile Policies. In *Workshop on Petri Nets and Graph Transformation (PNGT)*, volume 127 of *Electronic Notes in Theoretical Computer Science*, pages 87–105. Elsvier, 2005.

[10] K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. EATCS Monographs in Theoretical Computer Science. Springer, 1996.

[11] 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.

[12] E. Kyriacou, G. Fakas, and V. Pavlaki. A Completely Decentralized Workflow Management System for the Support of Emergency Telemedicine and Patient Monitoring. In *Proc. IEEE EMBS Annual International Conference*, 2006.

[13] S. Lack and P. Sobocinski. Adhesive and Quasiadhesive Categories. *Theoretical Informatics and Applications*, 39(5):511–546, 2005.

[14] J. Padberg, H. Ehrig, and L. Ribeiro. Algebraic high-level net transformation systems. *Mathematical Structures in Computer Science*, 5:217–256, 1995.

[15] J. Padberg, K. Hoffmann, H. Ehrig, T. Modica, E. Biermann, and C. Ermel. Maintaining Consistency in Layered Architectures of Mobile Ad-hoc Networks. In *Proc. Fundamental Approaches to Software Engineering (FASE)*, 2007. accepted.

[16] J. Padberg and M. Urbasek. 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*, pages 161–196. Springer, 2003.

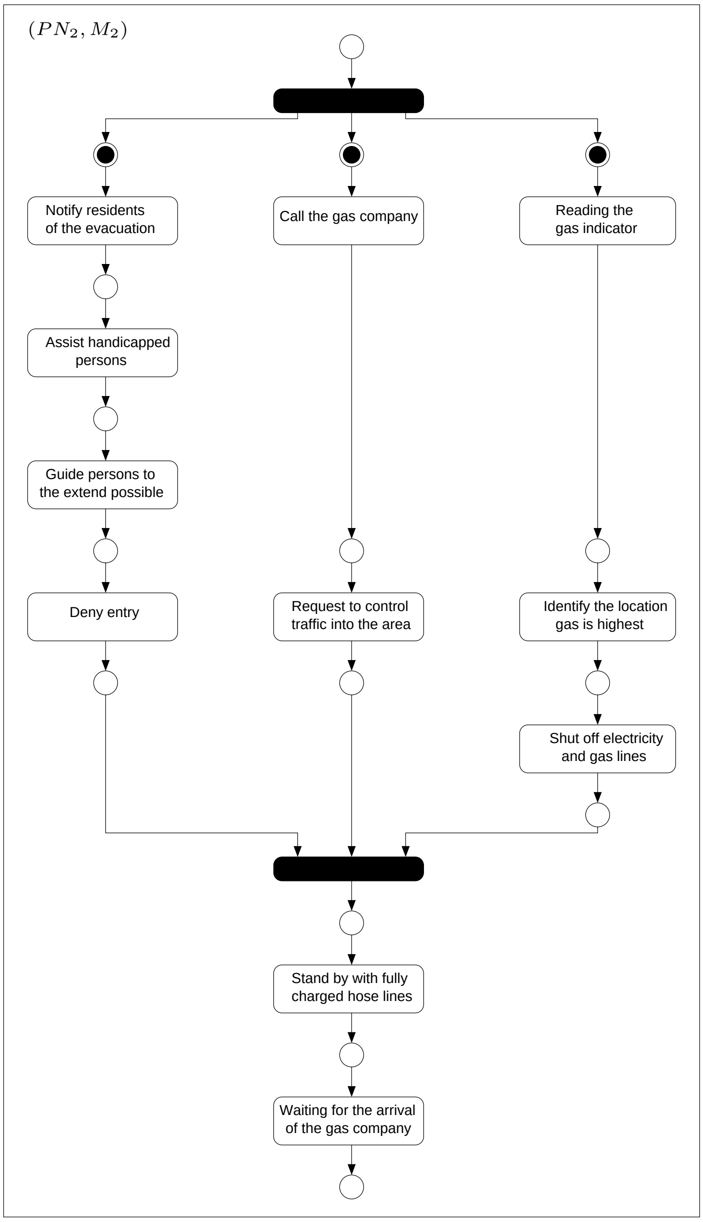

Fig. 5
PROCESS (PN_2, M_2)

[17] F. Parisi-Presicce. On modifying high level replacement systems. *Electronic Notes in Theoretical Computer Science*, 44(2), 2001.

[18] S. Rinderle, M. Reichert, and P. Dadam. Correctness criteria for dynamic changes in workflow systems - a survey. *Data Knowl. Eng.*, 50(1):9–34, 2004.

[19] F. De Rosa. *Adaptive process management in mobile and dynamic scenarios*. PhD thesis, SAPIENZA - Universita di Roma, Department of Computer Science, 2007.

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

[21] S. Sadiq, O. Marjanovic, and M. Orlowska. Managing change and time in dynamic workflow processes. *Journal of Cooperative Information Systems*, 9(12), 2000.

[22] W. M. P. van der Aalst and M. Weske. The P2P Approach to Interorganizational Workflows. In *Proc. Advanced Information Systems Engineering (CAiSE)*, volume 2068 of *LNCS*, pages 140–156. Springer, 2001.

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

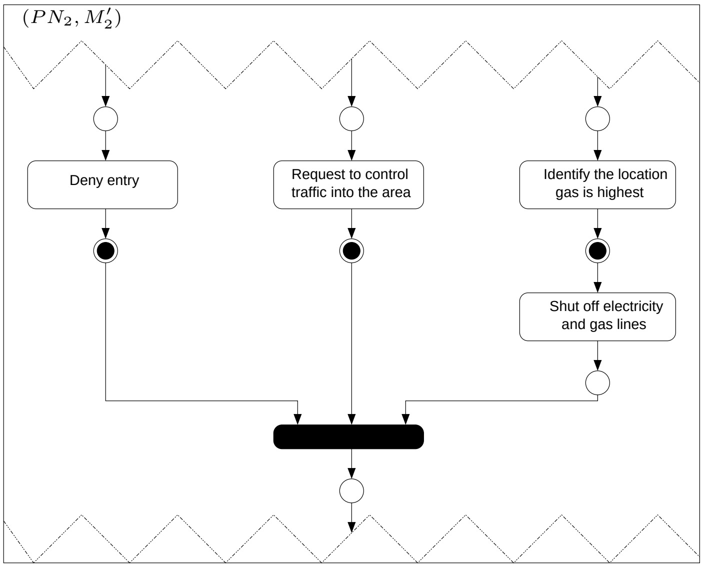

Fig. 6
RELEVANT PART OF PROCESS (PN_2, M_2')

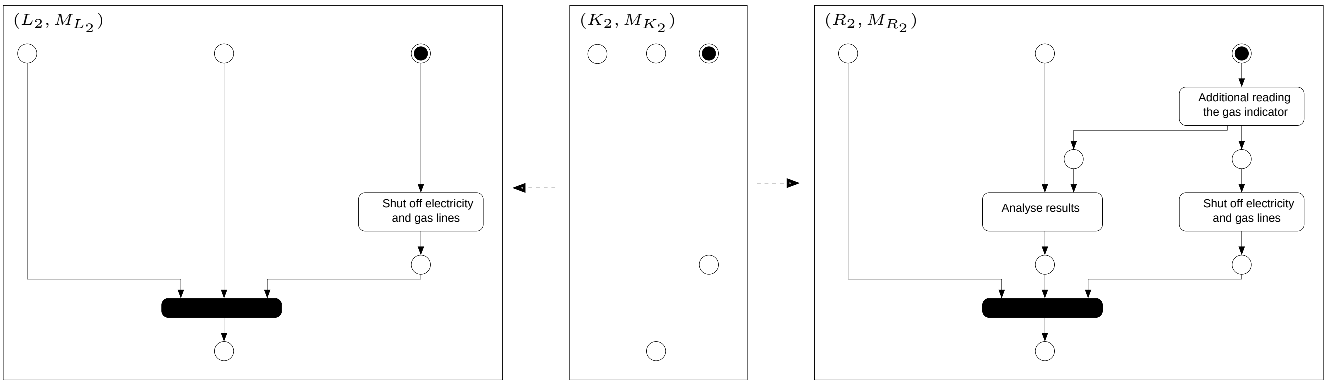

Fig. 7
RULE $prod_{analyse}$

[24] W.M.P. Van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. In *Proc. Cooperative Information Systems (CoopIS)*, volume 1901 of *LNCS*, pages 18–29. Springer, 2000.

[25] W.M.P. van der Aalst, M. Weske, and G. Wirtz. Advanced Topics in Workflow Management: Issues, Requirements, and Solutions. *Journal of Integrated Design and Process Science*, 7(3), 2003.

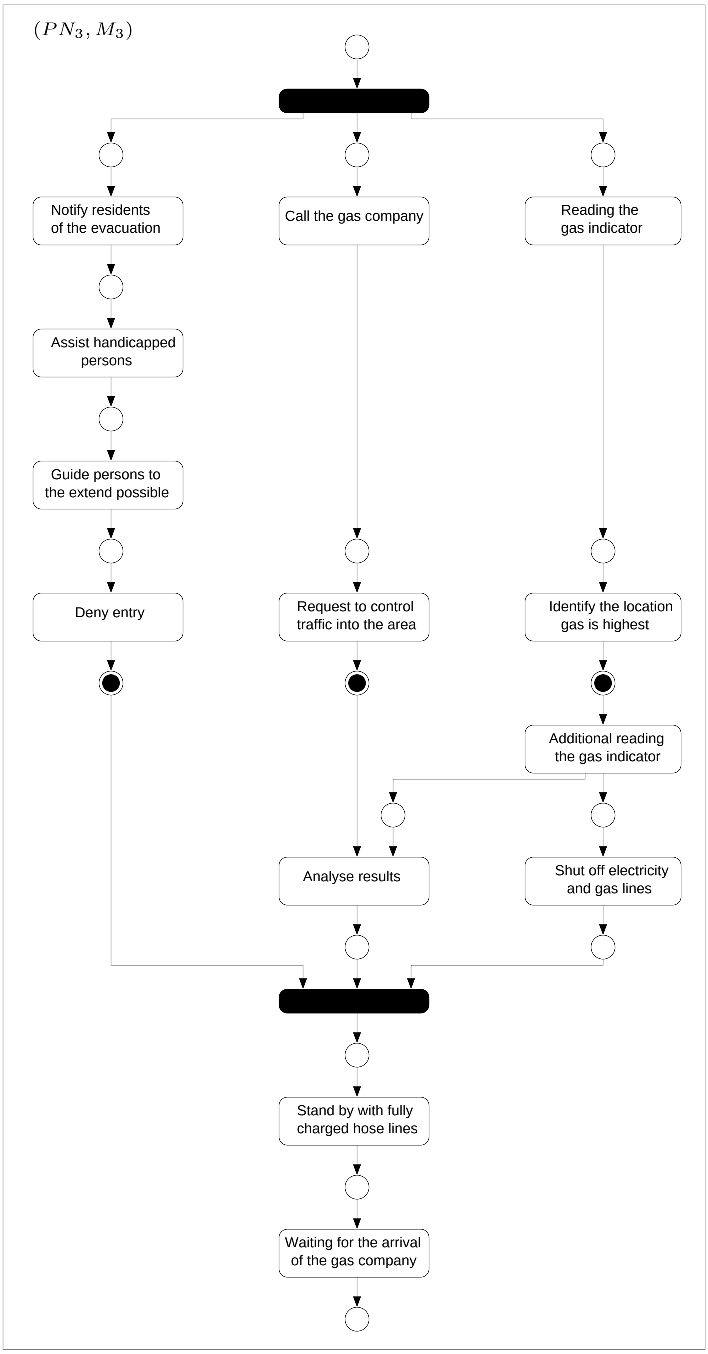

Fig. 8
PROCESS (PN_3, M_3)

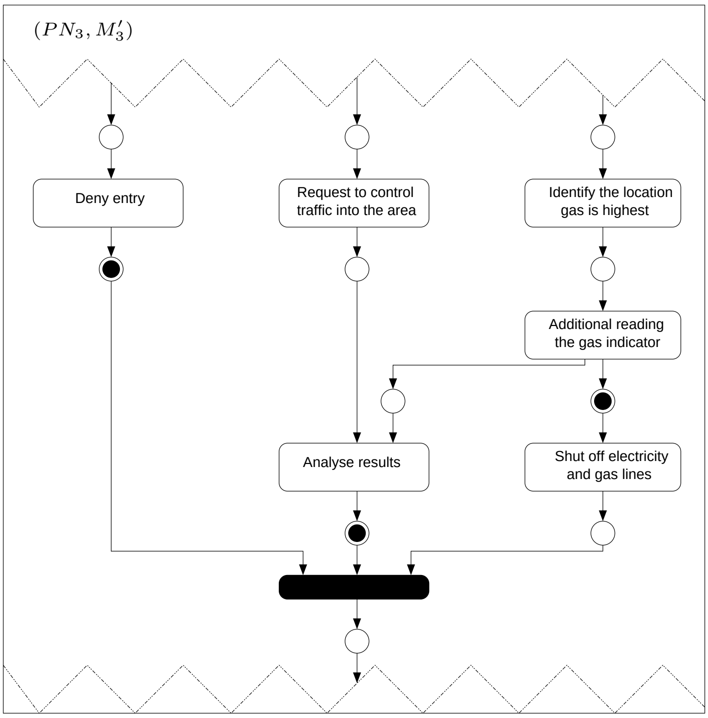

Fig. 9
RELEVANT PART OF PROCESS (PN_3, M_3')

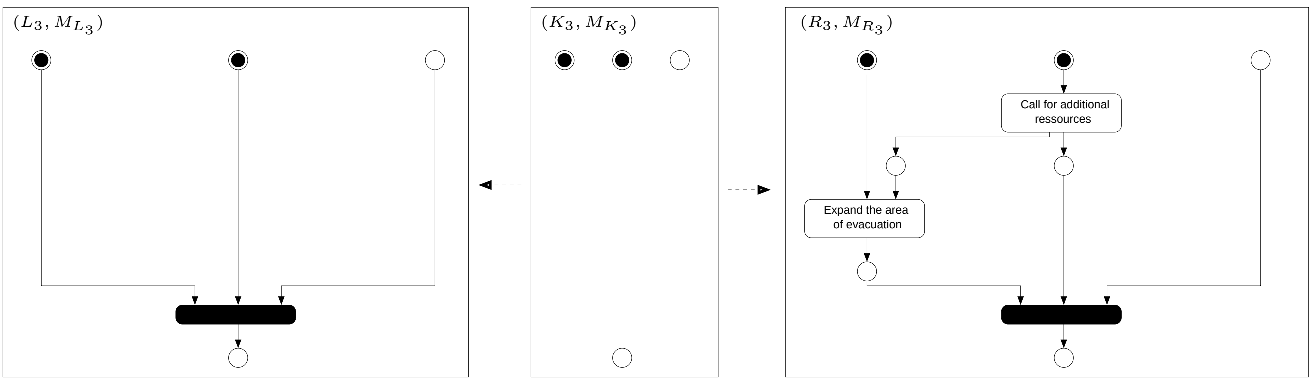

Fig. 10
RULE $prod_{expand}$

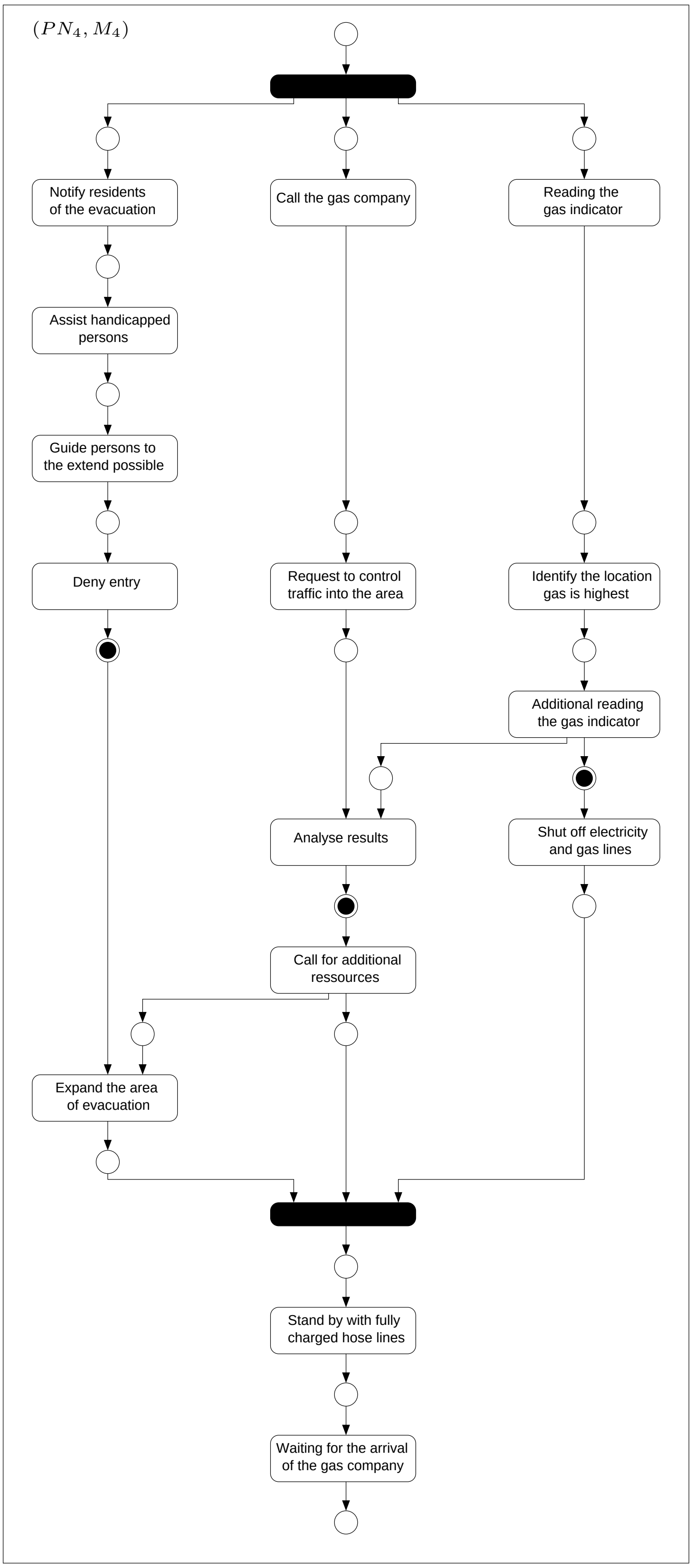

Fig. 11
PROCESS (PN_4, M_4)

$$
\begin{tikzcd}
(PN_1, M_1) \arrow[d, "\textit{and-split}"'] \arrow[dr, dotted, dash, "dependent"] & & & \\
(PN_1, M_1') \arrow[r, Rightarrow, "(prod_{evacuate}, m_1)"] \arrow[d, "\textit{Reading gas indicator}"'] & (PN_2, M_2) \arrow[d, "\textit{Reading gas indicator}"] & & \\
(PN_2, M_1^1) \arrow[r, Rightarrow, "(prod_{evacuate}, m_1)"] \arrow[d, "\textit{Identify the location}"'] & (PN_2, M_2^1) \arrow[d, "\textit{Identify the location}"] \arrow[dr, dotted, dash, "dependent"] & & \\
(PN_1, M_1^2) \arrow[r, Rightarrow, "(prod_{evacuate}, m_1)"] \arrow[d, "\textit{Call the gas company}"'] & (PN_2, M_2^2) \arrow[r, Rightarrow, "(prod_{analyse}, m_2)"] \arrow[d, "\textit{Call the gas company}"] & (PN_3, M_3^2) \arrow[d, "\textit{Call the gas company}"] & \\
(PN_1, M_1^3) \arrow[r, Rightarrow, "(prod_{evacuate}, m_1)"] \arrow[d, "\textit{Request to control traffic}"'] & (PN_2, M_2^3) \arrow[r, Rightarrow, "(prod_{analyse}, m_2)"] \arrow[d, "\textit{Request to control traffic}"] & (PN_3, M_3^3) \arrow[d, "\textit{Request to control traffic}"] & \\
(PN_1, M_1^4) \arrow[r, Rightarrow, "(prod_{evacuate}, m_1)"] \arrow[dr, dotted, dash, "dependent"'] & (PN_2, M_2^4) \arrow[r, Rightarrow, "(prod_{analyse}, m_2)"] \arrow[d, "\textit{Notify residents}"'] & (PN_3, M_3^4) \arrow[d, "\textit{Notify residents}"] & \\
& (PN_2, M_2^5) \arrow[r, Rightarrow, "(prod_{analyse}, m_2)"] \arrow[d, "\textit{Assist handicapped persons}"'] & (PN_3, M_3^5) \arrow[d, "\textit{Assist handicapped persons}"] & \\
& (PN_2, M_2^6) \arrow[r, Rightarrow, "(prod_{analyse}, m_2)"] \arrow[d, "\textit{Guide persons}"'] & (PN_3, M_3^6) \arrow[d, "\textit{Guide persons}"] & \\
& (PN_2, M_2^7) \arrow[r, Rightarrow, "(prod_{analyse}, m_2)"] \arrow[d, "\textit{Deny entry}"'] & (PN_3, M_3^7) \arrow[d, "\textit{Deny entry}"] & \\
& (PN_2, M_2') \arrow[r, Rightarrow, "(prod_{analyse}, m_2)"] \arrow[dr, dotted, dash, "dependent"'] & (PN_3, M_3) \arrow[d, "\textit{Additional reading}"'] & \\
& & (PN_3, M_3^1) \arrow[d, "\textit{Analyse results}"'] \arrow[dr, dotted, dash, "dependent"] & \\
& & (PN_3, M_3') \arrow[r, Rightarrow, "(prod_{expand}, m_3)"] & (PN_4, M_4)
\end{tikzcd}
$$

Fig. 12

INDEPENDENCE AND DEPENDENCE OF FIRING AND TRANSFORMATION STEPS