Electronic Communications of the EASST
Volume 51 (2012)

Proceedings of the 5th International Workshop on Petri Nets, Graph Transformation and other Concurrency Formalisms (PNGT 2012)

Abstract Interleaving Semantics for Reconfigurable Petri Nets

Julia Padberg

14 pages

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

Abstract Interleaving Semantics for Reconfigurable Petri Nets

Julia Padberg

Hamburg University of Applied Sciences, Germany
julia.padberg@haw-hamburg.de

Abstract:

Reconfigurable Petri nets are Petri nets together with rules for the dynamic change of the nets. We employ them for the formal modeling in the context of the *Living Place Hamburg*, a smart home that is an urban apartment serving as a laboratory for investigating different areas of ambient intelligence. The interaction of the resident and the smart home is modeled using informal descriptions of scenarios. These scenarios provide the resident's procedures together with the smart home's support.

A case study using reconfigurable Petri nets for modeling these scenarios has required extensions of the theory and has clearly shown the need for an interleaving semantics for reconfigurable Petri nets. Scenarios are then given by nets, namely decorated place/transition nets that can be adapted to the evolving subgoals by applying rules that change the nets and hence the behavior of the smart home. Decorated place/transition nets are annotated place/transition nets with additional transition labels that may change when the transition is fired. To obtain such reconfigurable Petri nets we prove that decorated place/transition nets give rise to an $\mathscr{M}$-adhesive HLR category.

The abstract interleaving semantics we introduce is a graph with nodes that consist of an isomorphism class of the net structure and an isomorphism class of the current marking. Arcs between these nodes represent computation steps being either a transition firing or a direct transformation.

Keywords: Interleaving semantics, reconfigurable place/transition nets, net transformation

1 Introduction

Reconfigurable Petri nets (e.g. in [EP03, LO04, EHP+07, PEHP08]) consist of marked Petri nets, i.e. a net with a marking, and a set of rules whose application modifies the net's structure at runtime. Typical application areas are concerned with the modeling of dynamic structures, for example workflows in a dynamic infrastructure. Reconfigurable Petri nets have been studied since several years, but up to now there is no formal definition of their semantics. The advantages of formal semantics are well-known; the definition of a precise meaning and hence the possibilities for analysis and verification. Here we present an interleaving semantics that comprises the possible computations in a graph. The nodes represent the states, namely the net together with the current marking and the edges represent the computation steps, namely the firing of a transition or the change of the net by a net transformation.

In the application area we aim at, reconfigurable Petri nets are used for modeling scenarios from the *Living Place Hamburg* [LP112], a smart home serving as a laboratory various areas of IT-based urban living, see Section 2. These scenarios can evolve in very many different ways depending on the action of the resident and the smart home's reaction. To provide means for validation and for model checking we introduce an abstract interleaving semantics for reconfigurable Petri nets. This semantics abstracts from the actual identities of places and transitions and is given by the equivalence classes induced by net isomorphisms. The abstract reachability graph has nodes that are such isomorphism classes of nets. The nodes are connected by an edge if and only if a computation step (either firing a transition or applying a rule) is possible.

The paper is organized as follows: First we introduce the Living Place Hamburg and summarize the results from the case study in [Rei12]. Next we extend place/transition (P/T) nets to decorated P/T nets adding some annotations as names and capacities. Moreover, we motivate changing transition labels and extend the firing of a transition so that the labels may be changed. Nevertheless, this extension is conservative to the firing behavior. Then we prove that decorated P/T nets are an $\mathscr{M}$-adhesive HLR category and hence we can define reconfigurable Petri nets based on decorated P/T nets. In Section 4 we first motivate the use of isomorphism classes as an suitable abstraction. Subsequently, we define recursively the abstract reachability graph and show that it represents the behavior of a reconfigurable Petri net. Concluding remarks concern related and future work.

2 Modeling Scenarios of The Living Place Hamburg

The main application is the modeling and analysis of scenarios in a smart home. The Living Place Hamburg[1] at the University of Applied Sciences (HAW) Hamburg is a place for concepts of IT based modern living and is under constant development of a smart home since January 2009. The Living Place Hamburg is a laboratory for applied research in different areas of ambient intelligence as well as an opportunity for research collaborations between the university and companies. It covers different areas of IT-based urban living. The Living Place Hamburg is a loft style urban living apartment with dynamic mapping of functions to spaces according to the respective situation of the resident (e.g. bedroom, kitchen, living room), see Fig.1. The resident's behavior follows specific procedures that the smart home has to support. These procedures depend on the situation as well as the available sensor data and the resident's action. These procedures are captured informally as scenarios. In [Ten11] the so-called morning scenarios have been developed that describe the morning procedure for a working day or for a Sunday. These informal descriptions give a sequence of events that constitute a possible ordering of the activities that take place on a

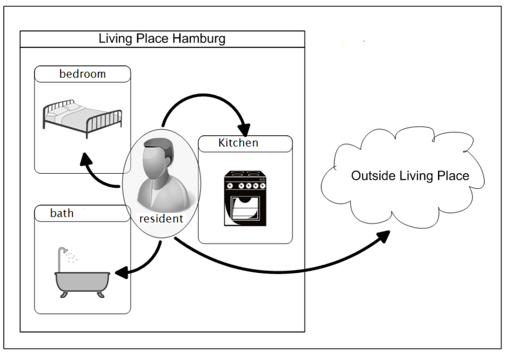


Figure 1: Action space of the Living Place's resident

[1] mainly funded by the Hamburg Ministry of Commerce and the Ministry of Science and Research

morning. The events and the ordering may differ
depending on the context, e.g. whether it is working day or a Sunday, whether there is enough time according to the schedule, on the weather and so on. The various sequences of events can be considered to be different scenarios that evolve according to the resident's behavior, sensor values and data dependent restrictions. These scenarios describe the anticipated procedures of the resident that the smart home should support. They are highly complex and hard to grasp in their unfolding.

Hence we aim at modeling those scenarios formally for the better understanding of the possible interaction of the smart home and its inhabitant. Petri nets in general are quite suitable for the modeling of predefined activities but usually lack the possibility of changing dynamically. Reconfigurable Petri nets provide both, subgoals of the scenarios can be modeled as workflows but the dynamic change of subgoals is modeled by transforming the Petri net itself. [Rei12] has shown that reconfigurable place/transition nets allow a suitable abstraction. There the scenarios describe the resident's procedures and the transformations describe the dynamic change of the infrastructure as the reactions to the resident's actions that both can be adequately captured.

To achieve such nets we introduce some additional decorations, capacities, names and the possibility to change the transition labels. This case study has been carried out without an explicit control structure for the application of the rules and the firing of transitions. We assume some decisions based on the sensor data, the current situation, and so on that may lead to any order of events. Causal dependencies need then to be expressed implicitly in the net structure or in the negative application conditions. Moreover, we need some trigger that relates firing and rule applications. So, we annotate the transitions with additional labels that may change when the transition fires. These decorations lead then to decorated place/transition nets.

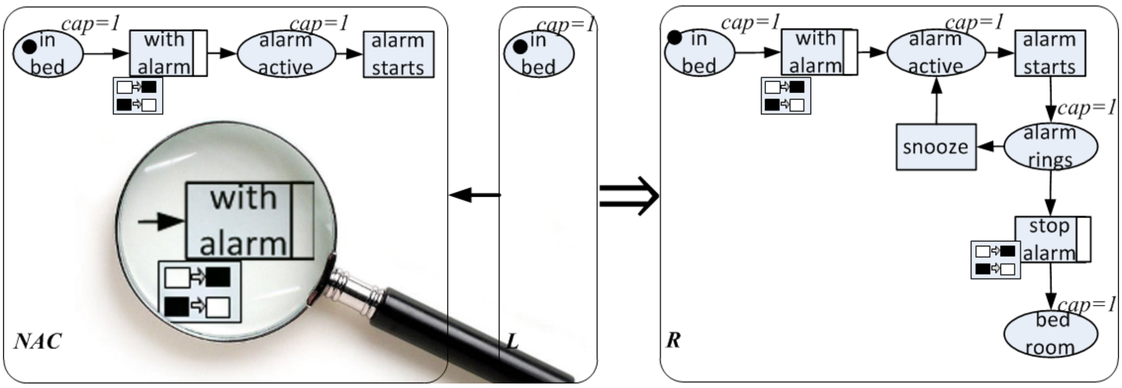


Figure 2: A rule of the "morning" scenarios

In Fig. 2 we see a rule from one of the morning scenarios, describing how the events concerning the alarm clock can evolve. Given that no other procedure with an alarm clock is running (that is represented by the NAC), then the net L describes that if the resident is in his/her bed the rule can be applied, yielding a procedure with an alarm clock that has a snooze function. The labels are given inside the places and transitions. All places have a capacity of 1 and the transitions `with alarm` and `stop alarm` have a changing label, that is either white or black and changes whenever the transition is fired. Inside the magnifying lens the transition label is depicted by the white bar left of the transition and the *rnw* function is depicted by a table mapping white to black and vice versa. These changing labels are used as a control structure for the rule application. Here the corresponding deleting rule removes the procedure for the alarm, if the resident has already got up. This rule may only be applied if the both transition labels are black, indicating that the alarm clock has been used.

The unfolding of these reconfigurable nets is quite complex and manifold. So, a formal semantics is clearly needed and a reachability graph can be the basis for further validation and analysis techniques. Reachability analysis in terms of a graph is the basic semantics for Petri nets, but for reconfigurable Petri nets we have to cope with the changing net structure (see Section 4). The abstract reachability graph allows checking boundedness and deadlocks, as well as liveness and reversibility. Recent investigations [Rei12] suggest that the scenarios even have a finite behavior, so the construction of a bounded reachability graph yields interesting possibilities for their analysis and verification based on model checking.

3 Decorated Place/Transition Nets are an $\mathscr{M}$-Adhesive HLR Category

Let us revisit the algebraic notion of Petri nets. A marked place/transition net is given by $N = (P, T, pre, post, M)$ with pre and post domain functions $pre, post : T \to P^{\oplus}$ and a current marking $M \in P^{\oplus}$, where $P^{\oplus}$ is the free commutative monoid over the set P of places. For $M_1, M_2 \in P^{\oplus}$ we have $M_1 \leq M_2$ if $M_1(p) \leq M_2(p)$ for all $p \in P$. A transition $t \in T$ is M-enabled for a marking $M \in P^{\oplus}$ if we have $pre(t) \leq M$, and in this case the follower marking M' is given by $M' = M \ominus pre(t) \oplus post(t)$ and $M[t\rangle M'$ is called firing step. Parallel firing of an firing vector $M[v\rangle M'$ can be computed using the pre and post domain functions $M' = M - pre^{\oplus}(v) + post^{\oplus}(v)$.

To provide the technical basis for the modeling of scenarios we need place/transition nets that have the following additional decorations: capacities, names for transitions as well as places and transition labels that can be changed by firing that transition. These transition labels may change when the transition fires. This new feature is important for the application of a rule after a transition has already fired (see the discussion of Fig. 2 in Section 2) and cannot be modeled without changing the labels. Considering the tokens in the post place of the transition does not work, because these tokens may be consumed as well. The extension to changing labels is conservative with respect to Petri nets as it does not alter the net's behavior, but it is crucial for the control of rule application and transition firing.

Morphisms of decorated place/transition nets are given as a pair of mappings for the places and the transitions, so that the structure and the decoration is preserved and the marking may be mapped strict, yielding an $\mathscr{M}$-adhesive high-level replacement (HLR) category (see Lemma 1). HLR categories and systems are an adequate framework for several kinds of transformation systems based on the double pushout approach. Such categories ensure that the numerous results for $\mathscr{M}$-HLR systems (e.g. [EEPT06, EGH+12]) also hold for decorated place/transition nets.

Definition 1 (Decorated place/transition net) A decorated place/transition net (dPTnet) is a marked P/T net $N = (P, T, pre, post, M)$ together with

- a capacity as a function $cap : P \to \mathbb{N}_+^\omega$
- A_P, A_T name spaces with $pname : P \to A_P$ and $tname : T \to A_T$
- the function $tlb : T \to W$ mapping transitions to transition labels W and
- the function $rnw : T \to END$ where END is a set containing some endomorphisms on W, so that $rnw(t) : W \to W$ is the function that renews the transition label.

Remark: If a partition on the transition labels is desired, then we use the family $(W_t)_{t \in T}$ of changing labels with $W := \biguplus_{t \in T} W_t$. Then the functions $tlb : T \to W$ and $rnw : T \to END$ are given so that $tlb(t) \in W_t$ and END is a set containing endomorphisms $f \uplus \biguplus_{x \in T \setminus \{t\}} id_{W_x} : W \to W$ with $f = rnw(t) : W_t \to W_t$.

The firing of these nets is the usual for place/transition nets except for changing the transition labels. In Fig. 2 the rnw function is represented as a table, where the two colors black and white are related. There the transition `with alarm` is mapped to the label "white" by tlb. The firing of the transition `with alarm` yields then a new labeling function tlb' that maps the transition to the label "black". This new labeling function is computed using the renew function rnw, stating that $tlb'(\texttt{with alarm}) = rnw \circ tlb(\texttt{with alarm}) = rnw(\text{white}) = \text{black}$.

Moreover, this extension works for parallel firing as well.

Definition 2 (Changing Labels by Parallel Firing) Given a transitions vector $v = \sum_{t \in T} k_t \cdot t$ then the label is renewed by firing $tlb[v\rangle tlb'$ and for each $t \in T$ the transition label $tlb' : T \to W$ is defined by:

$$tlb'(t) = rnw(t)^{k_t} \circ tlb(t)$$

As a purely mathematical example, consider natural numbers as the changing labels with $tlb(t_1) = 1$, $tlb(t_2) = 2$, and $tlb(t_3) = 3$. Then each transition is assigned a function on the natural numbers $rnw(t_i) : \mathbb{N} \to \mathbb{N}$ for $1 \le i \le 3$; for example $rnw(t_1) = id_{\mathbb{N}}$, $rnw(t_2)(n) = n+1$, and $rnw(t_3)(n) = n^2$. When firing a transition vector - say $t_1 + 2t_3$ - the corresponding renew functions are applied to the transition labels and the renewed label is given by $tlb[t_1 + 2t_3\rangle tlb'$. The follower marking m' is computed as usual $m[t_1 + 2t_3\rangle m'$, hence we have a conservative extension. The renewed transition label $tlb' : T \to \mathbb{N}$ is then defined by

$$\begin{aligned} tlb'(t_1) = \;& rnw(t_1) \circ tlb(t_1) = rnw(t_1)(1) = id(1) = 1 \\ tlb'(t_2) = \;& rnw(t_2)^0 \circ tlb(t_2) = id \circ tlb(t_2) = 2 \\ tlb'(t_3) = \;& rnw(t_3)^2 \circ tlb(t_3) = (rnw(t_3) \circ rnw(t_3))(3) = ((3)^2)^2 = 81 \end{aligned}$$

In order to define rules and transformations for decorated place/transition nets we introduce morphisms that map transitions to transitions by f_T and places to places by f_P. The later is extended to linear sums by $f_P^\oplus$. These morphisms preserve firing steps by Condition (1) and all annotations by Condition (2-4) below. Since Condition (4) preserves the transition labels, these labels only can be changed by firing the corresponding transition, but not by transformations.

Additionally, these morphisms require that the marking at corresponding places is not decreased (Condition (5)). For strict morphisms, in addition injectivity and the preservation of markings is required (Condition (6)).

Definition 3 (Morphisms between decorated place/transition nets) Given two decorated place/transition nets $N_i = (P_i, T_i, pre_i, post_i, M_i, cap_i, pname_i, tname_i, tlb_i, rnw_i)$ for $i = 1,2$ then $f : N_1 \to N_2$ is given by $f = (f_P : P_1 \to P_2, f_T : T_1 \to T_2)$ and the following equations hold:

1. $pre_2 \circ f_T = f_P^{\oplus} \circ pre_1$ and $post_2 \circ f_T = f_P^{\oplus} \circ post_1$
2. $cap_1 = cap_2 \circ f_p$
3. $pname_1 = pname_2 \circ f_P$
4. $tname_1 = tname_2 \circ f_T$ and $tlb_1 = tlb_2 \circ f_T$ and $rnw_1 = rnw_2 \circ f_T$
5. $M_1(p) \leq M_2(f_P(p))$ for all $p \in P_1$

Moreover, the morphism f is called strict

6. if both f_P and f_T are injective and $M_1(p) = M_2(f_P(p))$ holds for all $p \in P_1$.

Decorated place/transition nets together with the above morphisms yield the category **decoPT**.

$\mathscr{M}$-adhesive HLR systems can be considered as a unifying framework for graph and Petri net transformations and allow a uniform description of the different notion and results based on a class $\mathscr{M}$ of specific monomorphisms. Next we show that decorated place/transition nets yield an $\mathscr{M}$-adhesive HLR category for $\mathscr{M}$ being the class of strict morphisms. Hence we obtain all the well-known results, as transformation, local confluence and parallelism, application conditions, amalgamation and so on.

Lemma 1 *The category* **decoPT** *of decorated place/transition nets is an $\mathscr{M}$-adhesive HLR category.*

Proof. The proof applies the construction for weak adhesive HLR categories (see Theorem 1 in [PEL08]):
Constructing the category **decoPT** using comma categories, we use the identity functor ID : **Sets** $\to$ **Sets** and the functors $F_i : \mathbf{PT}_i \to$ **Sets** yielding either the place set P or the transition set T where the categories $\mathbf{PT}_i$ for $0 \leq i \leq 4$ arise from the stepwise construction with $\mathbf{PT} =: \mathbf{PT}_0$. The category of place/transition nets (with markings) is a weak adhesive HLR category (see [Pra08]): Then the comma category $\mathbf{PT_1} := CommCat(F_0, ID, \{pname\})$ yields the category of place/transition nets with places names and is a weak adhesive HLR category as F_0 preserves pushouts and ID pullbacks.

For the capacities we use the constant functor G_N : **Sets** $\to$ **Sets** yielding $\mathbb{N}_+^{\omega}$ the set of positive numbers together with ω. $\mathbf{PT_2} := CommCat(F_1, G_N, \{cap\})$ yields the category of place/transition nets with places names and capacities and is a weak adhesive HLR category as F_1 preserves pushouts and obviously G_N preserves pullbacks.

Analogously, we achieve the weak adhesive HLR category $\mathbf{PT_3} := CommCat(F_2, ID, \{tname\})$ with $F_2 : \mathbf{PT}_2 \to \mathbf{Sets}$ yielding the transition set T.

For the changing labels we need the category of sets with endomorphisms, that is a functor category $\mathbf{End} \cong [\mathbf{Loop}, \mathbf{Sets}]$ for the small category $\mathbf{Loop}$:  The objects are sets with one function on the set, $(W, f : W \to W)$ and morphisms are functions between the sets compatible with the function, i.e. $g : (W, f : W \to W) \to (W', f' : W' \to W')$ with $g \circ f = f' \circ g$. We then have $G_{tlb} : \mathbf{End} \to \mathbf{Sets}$ yielding the set W. G_{tlb} preserves pullbacks as functor categories preserve completeness and the limits are constructed "componentwise". With $F_3 : \mathbf{PT}_3 \to \mathbf{Sets}$ yielding the transition set T we have $\mathbf{PT}_4 = CommCat(F_3, G_{tlb}; \{tlb\})$ we have a weak adhesive HLR category for place/transition nets with transition labels.

Last, we employ the functor $G_{rnw} : \mathbf{End} \to \mathbf{Sets}$ yielding the singleton set $\{f\}$ containing f. This functor preserves pullbacks, because singleton sets are isomorphic in $\mathbf{Sets}$. Hence we obtain $\mathbf{decoPT} \cong Commcat(F_4, G_{rnw}, \{rnw\})$ with $F_4 : \mathbf{PT}_4 \to \mathbf{Sets}$ yielding the transition set as a weak adhesive HLR category. Hence, we have an $\mathscr{M}$-adhesive HLR category, see [EGH10]. □

Based on this fact we can define reconfigurable decorated place/transition nets, so that powerful notions and results for rules and transformations are already available, e.g. in [EEPT06, EGH+12].

Definition 4 (Reconfigurable Nets) A reconfigurable decorated place/transition net $RN = (N, \mathscr{R})$ is given by an decorated $N = (P, T, pre, post, M, cap, pname, tname, tlb, rnw)$ and a set of rules $\mathscr{R}$, where rules $r \in \mathscr{R}$ are given by $r = (NACS, L \leftarrow K \to R)$ with $K \to L, K \to R \in \mathscr{M}$.

An application of a rule $r = (NACS, L \leftarrow K \to R)$ (with negative application conditions $NACS$) is called a transformation step and describes how a net is changed by the rule. A rule in the DPO approach is given by three nets called left hand side L, interface K and right hand side R, respectively, and a span of two strict net morphisms $K \to L$ and $K \to R$. Additionally an occurrence morphism $o : L \to N$ is required that identifies the relevant parts of the left hand side in the given net N. Then a transformation step $N \stackrel{(r,o)}{\Longrightarrow}$ via rule r can be constructed in two steps, provided that the negative application conditions $(L \xrightarrow{n} NAC) \in NACS$ hold. Given a rule with a occurrence $o : L \to N$, then the negative application conditions hold if and only if there does not exist a morphism $q : NAC \to N$ with $q \circ n = o$ for any $NAC \in NACS$. The gluing condition has to be satisfied in order to apply a rule at a given occurrence. Its satisfaction requires that nothing id identified by the occurrence o that is deleted and that nothing is deleted that leads to incomplete nets. It is a sufficient condition for the existence and uniqueness of the so-called pushout complement which is needed for the first step in a transformation. In this case, we obtain a net M leading to a step $N \stackrel{(r,o)}{\Longrightarrow} M$ consisting of the following pushout (1) and (2) in Fig. 3.

$$\begin{tikzcd}
NAC \arrow[dr, bend right, "q"'] & L \arrow[l, "n"'] \arrow[d, "o"] \arrow[l, phantom, shift right=4, "\neg\exists"] \arrow[dr, phantom, "(1)"] & K \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "(2)"] & R \arrow[d] \\
 & N & D \arrow[l] \arrow[r] & M
\end{tikzcd}$$

Figure 3: Transformation of a net

4 Reachability Graph for Reconfigurable Petri Nets

First we give an intuitive idea of a transition system that describes firing transitions as well as applying rules for the dynamic change. Since the decorations of the decorated place/transition nets are not relevant for the following consideration, let us investigate the simple place/transition net N together with one rule r in Fig. 4. Starting with the given net it may fire or it may be transformed using the given rule. Each of the nets that can be reached by firing a transition or by applying the rule should be represented in the reachability graph.

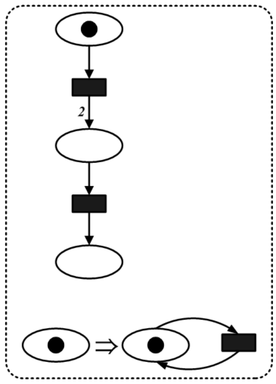


Figure 4: $RN = (N, \{r\})$

Obviously, these nets are given in an abstract way, not considering the exact identities of the places and transitions. The example in Fig. 5 gives already an intuitive idea of the interleaving semantics. We have a graph whose nodes are denoting the system's state. Both transition firing and application of a rule are considered to be computation steps and are represented by arcs in that graph. States can no longer be denoted by mere markings of net, as the net may have been changed using transformations.

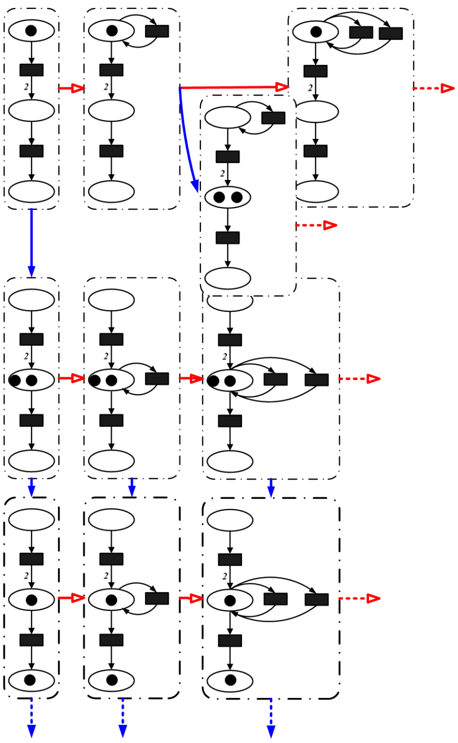
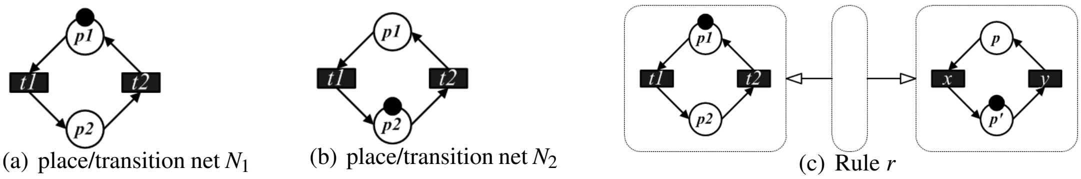


Figure 5: Abstract reachability graph of RN

So, we use the net together with the current marking as a state description. Hence, for each net that is reachable by the transformations its reachability graph is a subgraph of the interleaving semantics of the reconfigurable net. Using isomorphism classes of nets is natural, because all diagrams of nets are only defined up to isomorphism as the set of places and the set of transitions is given without their identities as elements of a set. Concrete nets, that are the nets with the identities of places and transitions are not suitable as we then obtain for each transformation step infinitely many resulting nets. The result of a transformation step is defined only up to isomorphism, since the underlying categorical pushout construction is also only up to isomorphism.

For the abstraction from the identity of places and transitions we introduce standard isomorphisms allowing only one isomorphism for each pair of net structures. Using these standard isomorphisms avoids the following problem with plain isomorphisms classes: Consider the example in Figure 6. The sequential firing of the transitions in net N_1 in Fig. 6(a) leads to $N_1[t_1\rangle N_2[t_2\rangle N_1$. Obviously, the nets N_1 and N_2 are isomorphic, but they have to be differentiated in order to describe the firing adequately. Using plain isomorphism classes nets N_1 and N_2 are within the same isomorphism class $[N_1]$ and the firing would then lead to $[N_1][t1\rangle[N_1][t2\rangle[N_1]$. This is not an adequate abstraction of the firing behavior. Moreover, considering the rule r in Fig. 6(c) it is clear, that the resulting net is isomorphic to both nets N_1 and N_2. But to ensure the firing behavior only one possibility may be considered, so for the reachability graph we have to determine which. In our example using standard isomorphisms on the net structure ensures that N_1 and N_2 do not belong to the same isomorphism class. Only the identity on the net structure is allowed and for the identical mapping of places the markings of N_1 and N_2 are different. For the rule r standard isomorphisms determine, which of the two possibilities is to be taken. For the resulting net there is a standard isomorphism either to net N_1 or to net N_2, but not to both. So, plain isomorphism classes do not distinguish nets precisely enough, but standard isomorphisms do.



(a) place/transition net N_1 (b) place/transition net N_2 (c) Rule r

Figure 6: The need for standard isomorphisms

The problem is closely related to the construction of abstract derivations in [CMR+97], where standard isomorphism have been used between graphs. In [CMR+97] abstract models of computation for a (graph) grammar are defined that take care of isomorphisms and shift equivalence leading to an abstract concurrent semantics. Standard isomorphisms are used because classes the composition of abstract computation fails using plain isomorphisms. The reason is that the relation between two isomorphic graphs has to be determined and may not be changed within one computation. This need for fixing the isomorphisms is the same as for the abstract reachability graph.

So, for the isomorphism classes we need to separate the net's marking and the net's structure, that is the net without its current marking.

Definition 5 (Net structure and net marking) Given a decorated place/transition net $N = (P, T, pre, post, M, cap, pname, tname, tlb, rnw)$ its net structure is denoted by $ns(N) = (P, T, pre, post, cap, pname, tname, tlb, rnw)$ and its marking by $mark(N) = M$.

Hence, we use the equivalence on the net structure induced by isomorphisms and we employ the notion of standard isomorphisms (as in [CMR+97]). Here, we allow for isomorphic net structures only one fixed isomorphism in between.

Definition 6 (Standard isomorphism) A family $s = \{s(NS_1, NS_2) \mid NS_1 \cong NS_2\}$ of standard isomorphisms is indexed by pairs of isomorphic net structures $NS_i = (P_i, T_i, pre_i, post_i, cap_i, pname_i, tname_i, tlb_i, rnw_i)$ for $i = 1, 2$ satisfying the following conditions:

- $s(NS_1, NS_2) : NS_1 \to NS_2$
- $s(NS, NS) = id_{NS}$

- $s(NS_1, NS_2) \circ s(NS_0, NS_1) = s(NS_0, NS_2)$

An abstract net structure is given by the set of all isomorphic net structures, so we refrain from considering the identities of places and transitions.

Definition 7 (Abstract net structure) Given a net structure $NS = (P, T, pre, post, cap, pname, tname, tlb, rnw)$ then the abstract net structure $[NS]$ is the equivalence class induced by isomorphisms.

Using equivalence classes of net structures leads to an explicit treatment of the markings. Two markings are equivalent if and only if they belong to isomorphic net structures and are then mapped onto another by the standard isomorphism.

Definition 8 (Abstract Marking) Given markings $M_i \in P_i^{\oplus}$ with $i = 1,2$, then $M_1 \sim M_2$ iff there are correspondig net structures $NS_i = (P_i, T_i, pre_i, post_i, cap_i, pname_i, tname_i, tlb_i, rnw_i)$, so that the standard isomorphisms $s : NS_1 \to NS_2$ maps the markings onto each other $s_P^{\oplus}(M_1) = M_2$. An abstract marking $[M_1]$ is the equivalence class induced by $\sim$.

Obviously, this relation is an equivalence relation. The abstract reachability graph of a reconfigurable decorated place/transition net $R = (N, \mathscr{R})$ is given by the abstract net structures and the abstract markings that are related by computation steps, i.e the firing of a transition or the application of a rule.

Definition 9 (Abstract reachability graph) For a reconfigurable decorated place/transition net $R = (N, \mathscr{R})$ its abstract reachability graph $AR = (V, E)$ with $E \subseteq V \times V$ is the smallest graph satisfying the following conditions:

1. $([ns(N)], [mark(N)]) \in V$

2. If $([\widehat{NS}], [\widehat{M}]) \in V$ and for some net N' with its net structure $ns(N') \in [\widehat{NS}]$ and its marking $M' = mark(N') \in [\widehat{M}]$ there exists $t \in T$ with $M'[t\rangle M''$,
then $([\widehat{NS}], [M'']) \in V$ and $\langle([\widehat{NS}], [\widehat{M}]), ([\widehat{NS}], [M''])\rangle \in E$.

3. If $([\widehat{NS}], [\widehat{M}]) \in V$ and for some net N' with its net structure $ns(N') \in [\widehat{NS}]$ and its marking $mark(N') \in [\widehat{M}]$ there is a transformation step $N' \stackrel{(r,o)}{\Longrightarrow} N''$ for a rule $r = (L \leftarrow K \rightarrow R) \in \mathscr{R}$ and an occurrence $o : L \to N'$,
then $([ns(N'')], [mark(N'')]) \in V$ and $\langle([\widehat{NS}], [\widehat{M}]), ([ns(N'')], [mark(N'')])\rangle \in E$.

The construction of the abstract reachability graph adequately describes the interleaving of a reconfigurable decorated place/transition net since we can prove that for each computation step (i.e. a possible firing or transformation) there is exactly one edge in the reachability graph (see Lemma 2). Moreover, for each transition step, that yields a different marking, there are two distinguished abstract net classes (see Lemma 3). So, the problem stated above for plain isomorphisms classes does not occur for this construction.

Lemma 2 (*AR* is well-defined) *In the abstract reachability graph AR there is a computation step* $(v_1, v_2) \in E$ *with* $v_i = ([\widehat{NS_i}], [\widehat{M_i}])$ *for* $i = 1,2$ *if and only if for each net* N_1 *with* $ns(N_1) \in [\widehat{NS_1}]$ *and* $mark(N_1) \in [\widehat{M_1}]$ *:*

- *there is a transition* $t \in T$*, such that* $mark(N_1)[t\rangle M''$ *and* $ns(N_1) \in [\widehat{NS_2}]$ *and* $M'' \in [\widehat{M_2}]$
- *or there is a rule* $r = (L \leftarrow K \rightarrow R) \in \mathscr{R}$ *and a occurrence* $o : L \rightarrow N_1$*, so that* $N_1 \stackrel{(r,o)}{\Longrightarrow} N_2$ *and* $ns(N_2) \in [\widehat{NS_2}]$ *and* $mark(N_2) \in [\widehat{M_2}]$

Proof. By induction on the number of computation steps and using the facts that net morphisms preserve the firing and transformations are compatible with isomorphisms we have:
Given a computation step $\langle([\widehat{NS_1}], [\widehat{M_1}]), ([\widehat{NS_2}], [\widehat{M_2}])\rangle \in E$, then for each net N_1 with $ns(N_1) \in [\widehat{NS_1}]$ and $mark(N_1) = M_1 \in [\widehat{M_1}]$ there is by construction

- a net N' with $ns(N') \in [\widehat{NS_1}]$ and its marking $M' = mark(N') \in [\widehat{M_1}]$ so that there exists $t \in T$ with $M'[t\rangle M''$.

  Then we have the standard isomorphism $s : ns(N') \rightarrow ns(N_1)$ and $M' \sim M_1$, so in N_1 we have $M_1 = s_P^{\oplus}(M')$ and $s_P^{\oplus}(M')[s_T(t)\rangle s_P^{\oplus}(M'')$, because net morphisms preserve firing. And obviously, $s_P^{\oplus}(M'') \in [\widehat{M_2}]$.

- Or there is some net N' with its net structure $ns(N') \in [\widehat{NS_1}]$ and its marking $mark(N') \in [\widehat{M_1}]$ so that there is a transformation step $N' \stackrel{(r,o')}{\Longrightarrow} N''$ for a rule $r = (L \leftarrow K \rightarrow R) \in \mathscr{R}$ and an occurrence $o' : L \rightarrow N'$. Then we have $L \stackrel{o'}{\rightarrow} N' \stackrel{s}{\rightarrow} N_1$ for some standard isomorphisms $s : ns(N') \rightarrow ns(N_1)$ and hence for $o : L \rightarrow N_1$ with $o := s \circ o'$ there is the transformation step $N_1 \stackrel{(r,o)}{\Longrightarrow} N_2$ with $ns(N_2) \in [\widehat{N_2}]$ and $mark(N_2) \in [\widehat{M_2}]$ as transformation steps are compatible with isomorphisms.

The only if part is due to the construction. □

The construction of the abstract reachability graph ensures that nets having the same net structure but different markings cannot represented by one node.

Lemma 3 (*AR* preserves firing) *Given a net N with* $M[t\rangle M'$*, then* $([N],[M]) = ([N],[M'])$ *if and only if* $M = M'$*.*

Proof. Since the standard isomorphism on N is the identity we have $([N],[M]) = ([N],[M'])$ iff $[M] = [M']$ iff $id^{\oplus}(M) = M'$ iff $M = M'$. □

5 Conclusion

Related work: Any approach to extend Petri net semantics to reconfigurable Petri nets has to deal with a changing net structure that is not taken into account by the given semantics. There are various approaches to the semantics of Petri nets as well as to the semantics of graph transformations. But the combination of both as required by reconfigurable Petri nets, does not immediately yield a semantics. Interleaving semantics in terms of a reachability graph are well-known since the beginnings of Petri nets, e.g. [Rei82]. Other approaches to Petri nets semantics based on partial order semantics [MR95, BP96] might be related to graph processes as in [CMR96].

For a concurrent semantics that is based on the transformations we could investigate Petri nets with individual tokens (PTI nets). These can be considered as graph transformation systems (see [Mod12]). PTI nets can be mapped by a collection construction to the category of place/transition nets with a marking as an element of the free commutative monoid over the set of places. Transition firing can then be simulated by transformation steps. But translating nets with individual tokens to a graph transformation system does not yield immediately reconfigurable nets with individual tokens. If the net structure is changed, these changes need to be integrated to the net's behavior using multi-amalgamation. Then would be feasible to use directly the graph transformation approach in [CMR+97].

We have presented an abstract interleaving semantics for reconfigurable Petri nets that is based on abstract net structures. This abstract reachability graph represents the behavior of reconfigurable Petri nets as each possible computation step is represented. But from the practical point of view the abstraction to isomorphism classes has an quite unpleasant consequence. The construction of the *AR* needs to check for isomorphic net structures. But searching for graph isomorphisms can be quite expensive. So, it is most important to use the results from [PEHP08, HEH10] making use of the necessary and sufficient condition for parallel independence of transition firing and transformations.

Future work is to implement the abstract reachability graph for the tool *ReConNet* that is developed as a student's project at the HAW Hamburg (see [EHOP12]). The main task is to compute the dependencies during the construction of the reachability graph. Since the reachability graphs of the Living Place's scenarios are most probably bounded considering model checking is then very promising.

Bibliography

[BP96] N. Busi, G. M. Pinna. Non Sequential Semantics for Contextual P/T Nets. In Billington and Reisig (eds.), *Application and Theory of Petri Nets*. Lecture Notes in Computer Science 1091, pp. 113–132. Springer, 1996.

[CMR96] A. Corradini, U. Montanari, F. Rossi. Graph Processes. *Fundam. Inform.* 26(3/4):241–265, 1996.

[CMR+97] A. Corradini, U. Montanari, F. Rossi, H. Ehrig, R. Heckel, M. Löwe. Algebraic Approaches to Graph Transformation - Part I: Basic Concepts and Double Pushout Approach. In *Handbook of Graph Grammars*. Pp. 163–246. World Scientific, 1997.

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

[EGH10] H. Ehrig, U. Golas, F. Hermann. Categorical Frameworks for Graph Transformation and HLR Systems Based on the DPO Approach. *Bulletin of the EATCS* 102:111–121, 2010.

[EGH+12] H. Ehrig, U. Golas, A. Habel, L. Lambers, F. Orejas. M-Adhesive Transformation Systems with Nested Application Conditions. Part 2: Embedding, Critical Pairs and Local Confluence. *Fundam. Inform.* 118(1-2):35–63, 2012.

[EHOP12] M. Ede, K. Hoffmann, G. Oelker, J. Padberg. ReConNet: A Tool for Modeling and Simulating with Reconfigurable Place/Transition Nets. In Krause and Westfechtel (eds.), *Proc. Seventh International Workshop on Graph-Based Tools (GraBaTs'12)*. Electronic Communications of the EASST, 2012. accepted.

[EHP+07] H. Ehrig, K. Hoffmann, J. Padberg, U. Prange, C. Ermel. Independence of Net Transformations and Token Firing in Reconfigurable Place/Transition Systems. In Kleijn and Yakovlev (eds.), *ICATPN*. Lecture Notes in Computer Science 4546, pp. 104–123. Springer, 2007.

[EP03] H. Ehrig, J. Padberg. Graph Grammars and Petri Net Transformations. In Desel et al. (eds.), *Lectures on Concurrency and Petri Nets*. Lecture Notes in Computer Science 3098, pp. 496–536. Springer, 2003.

[HEH10] K. Hoffmann, H. Ehrig, F. Hermann. Flexible Independence of Net Transformations and Token Firing in the Cospan DPO Approach. In *Proceedings of the 3rd International Symposium of Multiagent Systems (MAS), Robotics and Cybernetics: Theory and Practice, 2009*. IIAS, 2010.

[LO04] M. Llorens, J. Oliver. Structural and Dynamic Changes in Concurrent Systems: Reconfigurable Petri Nets. *IEEE Trans. Computers* 53(9):1147–1158, 2004.

[LP112] The Living Place Hamburg. 2012. http://livingplace.informatik.haw-hamburg.de/blog/.

[Mod12] T. Modica. *Formal Modeling, Simulation, and Validation of Communication Platforms*. PhD thesis, Technische Universität Berlin, 2012.

[MR95] U. Montanari, F. Rossi. Contextual Nets. *Acta Inf.* 32(6):545–596, 1995.

[PEHP08] U. Prange, H. Ehrig, K. Hoffmann, J. Padberg. Transformations in Reconfigurable Place/Transition Systems. In Degano et al. (eds.), *Concurrency, Graphs and Models*. Lecture Notes in Computer Science 5065, pp. 96–113. Springer, 2008.

[PEL08] U. Prange, H. Ehrig, L. Lambers. Construction and Properties of Adhesive and Weak Adhesive High-Level Replacement Categories. *Applied Categorical Structures* 16(3):365–388, 2008.

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

[Rei82] W. Reisig. *Petrinetze, Eine Einführung*. Springer, 1982.

[Rei12] F. Reiter. Modellierung und Analyse von Szenarien des Living Place mit rekonfigurierbaren Petrinetzen. Bachelor Thesis, Hochschule für Angewandte Wissenschaften Hamburg, 2012.
http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/reiter.pdf

[Ten11] S. Tennstedt. Szenarien basierte Interpretationen eines Smart-Homes. Master Thesis, Hochschule für Angewandte Wissenschaften Hamburg, 2011.
http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/tennstedt.pdf