On the modularity in Petri Nets of Active Resources*

Vladimir A. Bashkin

Yaroslavl State University
Yaroslavl, 150000, Russia
email: bas@uniyar.ac.ru

Abstract. Petri Nets of Active Resources (AR-nets) represent a dual syntax of Petri nets with a single type of nodes (places and transitions are united) and two types of arcs (input and output arcs are separated). In AR-nets the same token may be considered as a passive resource (produced or consumed by agents) and an active agent (producing or consuming resources) at the same time.
It is shown that the homogeneous structure of nodes in AR-nets allows some specific modular modeling and transformation techniques. Properties of net partitions and reachability-equivalent module replacements are studied.

1 Introduction

Nowadays there exist a lot of Petri net modifications introducing different modular and/or hierarchical syntax. In particular, different high-level formalisms appeared to be quite effective and useful in practice [6, 14]. Many authors use algebraic approach to compositions and decompositions [3, 4], or apply some effective algebraic methods of modular verification [9]. Some models even allow a recursion [7, 12].

As a rule, in compositional Petri nets modules may be linked by synchronized transitions [5, 9, 12] or by common interface places [8]. This is a natural consequence of Petri net syntax. Indeed, the structure of a net is explicitly divided into two classes of elements: places and transitions. Places correspond to the passive component of the system (*state* or *resources*), transitions correspond to its active component (*actions* or *events* or *agents*).

However, explicit separation of nodes into places and transitions is not the only way of Petri net definition. There exists a number of equivalent formalisms with a different separation of elements. Some of them use the duality between places and transitions (the importance of studying this duality was mentioned by C.-A.Petri already in [13]).

In [11] K.Lautenbach introduced a notion of dual place/transition nets. In this formalism the transitions are also marked by special tokens called "t-tokens".

* This research is partially supported by Russian Fund for Basic Research (projects 09-01-00277, 11-01-00737) and Federal Program "Kadry" (project 02.740.11.0207).

The meaning of t-tokens is that they prevent transitions from being enabled. A transition carrying a t-token cannot be enabled by any marking of p-tokens. Place in the net can be enabled and fired in the dual way. Place firing transforms the marking of t-tokens, arcs for place firing are inverted. So the net can be dualized in the obvious way. K.Lautenbach in his work proposed dual P/T nets as a model of system fault propagation.

In [10] M.Köhler and H.Rölke introduced Super-Dual nets for modeling with dynamic refinement of events. In this formalism transitions are also marked by special tokens called "pokens", but these pokens *enable* transition firings. Places can also fire, but their firing use a special separate set of arcs called "glow relation" in contrast to common "flow relation". Super-Dual net can be dualized by interchanging places and transitions, tokens and pokens, flow arcs and glow arcs. In [10] it is proven that Super-Dual nets have the same expressive power as ordinary Petri nets.

In both dual P/T nets and Super-Dual nets duality is based on two types of elements of the system — resources and actions (places and transitions). These elements are represented in the net by vertices of a bipartite oriented graph. However, there is another (implicitly) divided set in every Petri net (and in every other bipartite oriented graph) — the set of arcs. It contains arcs of two crucially different types — input arcs from places to transitions remove tokens, output arcs from transitions to places produce tokens. The explicit separation of this notions allowed us to define an "orthogonal" syntax for Petri nets — Nets of Active Resources (AR-nets) [1].

A definition of an AR-net is a dualized definition of a Petri net. The set of arcs is explicitly transformed into two separate sets of *input arcs* and *output arcs*. The sets of transitions and places are united into a single set of *nodes*. Each node may contain *tokens*. A token in the node may fire, consuming some tokens through input arcs and producing some other tokens through output arcs. So a token simulates behaviour of both an active component (an agent) and a passive component (a resource) at the same time. Therefore the formalism is called "nets of active resources". AR-nets are well-suited for modeling systems with an explicit definition of an agent [2].

In this paper we study the compositional properties of AR-nets. A module is represented as a subnet defined by some subset of nodes. An interface of the module is a set of arcs linking its nodes with an outer subnet. A module may have four types of links: input, output, production and consumption. First two of them represent actions of the module itself, the other two represent actions of its neighbours. Hence syntactically a module with adjacent links can be treated as a node with adjacent arcs. This generalization is quite natural and does not affects the homogeneity of the graph of the net.

It is shown that a number of net properties may be inherited from the properties of modules of particular types. It is proven that any nested decomposition of the net can be transformed into an equivalent agent/resource decomposition. For a flat decomposition these kinds of transformations are applicable depending only on the chromatic number of the module linkage graph.

A problem of module equivalence w.r.t. system reachability is also defined and studied. A most general case of this equivalence is obviously undecidable. For a simple case of replacement a criterion of equivalence is given. It is shown how a module (with some additional requirements) can be replaced by a single node without affecting the global reachability relation.

The paper is organized as follows. In section 2 we give basic definitions and notations for nets of active resources. In section 3 an AR-module is introduced. We study different types of modules, their properties and methods of decompositions. In section 4 two notions of module equivalence are defined and studied. Section 5 contains some conclusions and directions for possible future work.

2 Preliminaries

Let S be a finite set. A *multiset* M over a set S is a mapping $M : S \to Nat$, where *Nat* is the set of natural numbers (including zero), i. e. a multiset may contain several copies of the same element.

For two multisets M, M' we write $M \subseteq M'$ iff $\forall s \in S : M(s) \leq M'(s)$ (the inclusion relation). The sum and the union of two multisets M and M' are defined as usual: $\forall s \in S : (M+M')(s) = M(s)+M'(s),\ M \cup M'(s) = max(M(s), M'(s))$.

By $\mathcal{M}(S)$ we denote the set of all finite multisets over S.

For a multiset $M \in \mathcal{M}(S)$ and a subset $S' \subseteq S$ denote a *projection* $M[S'] \in \mathcal{M}(S')$ of M onto S' as follows: $\forall s \in S' : M[S'](s) = M(s)$.

Similarly, for a binary relation $R \subseteq \mathcal{M}(S) \times \mathcal{M}(S)$ a *projection* $R[S'] \subseteq \mathcal{M}(S') \times \mathcal{M}(S')$ is defined as follows: $\forall s_1, s_2 \in S'\ (s_1, s_2) \in R[S'] \Leftrightarrow (s_1, s_2) \in R$.

Definition 1. *[1] A* net of active resources *is a tuple $N = (V, I, O)$, where*

- *V is a finite set of* resource nodes *(*vertices*);*
- *$I \subseteq \mathcal{M}(V \times V)$ is a* consumption relation *(*input arcs*);*
- *$O \subseteq \mathcal{M}(V \times V)$ is a* production relation *(*output arcs*).*

In graphic form the nodes are represented by circles, the consumption relation by dotted arrows and the production relation by solid arrows.

A *marked net of active resources* is a pair (N, M_0) where N is an AR-net and $M_0 \in \mathcal{M}(V)$ is its *initial marking*.

As usual, pictorially the marking is denoted by black dots.

For a node $v \in V$ by $I(\bullet, v), O(v, \bullet), I(v, \bullet)$ and $O(\bullet, v)$ denote the multisets of nodes of *preconditions*, *postconditions*, *consumers* and *producers*: $\forall w \in V$

$$I(\bullet, v)(w) =_{\text{def}} I(w, v); \quad O(v, \bullet)(w) =_{\text{def}} O(v, w);$$
$$I(v, \bullet)(w) =_{\text{def}} I(v, w); \quad O(\bullet, v)(w) =_{\text{def}} O(w, v).$$

Definition 2. *A node $v \in V$ is* active *in a marking M iff*

- *$M(v) > 0$ (the node v is not empty);*
- *$I(\bullet, v) \subseteq M$ (there are enough tokens in all its input nodes).*

An active node v may fire *yielding a new marking M' s.t.*

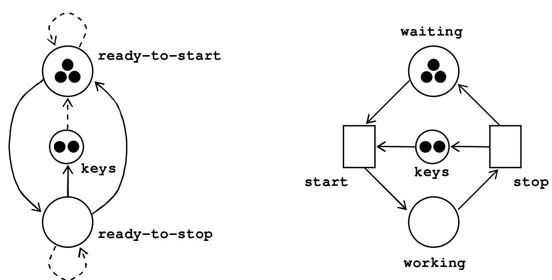
$$M' =_{def} M - I(\bullet, v) + O(v, \bullet) \text{ (denoted } M \xrightarrow{v} M'\text{).}$$

Some natural notions:

Let $i \in I$ and $i = (v_1, v_2)$. Then the arc i is called an *input* arc for the node v_2 and a *consuming* arc for the node v_1. A token in the node v_1 may be *consumed* through the arc i, a token in the node v_2 can *consume* through the arc i.

Let $o \in O$ and $o = (v_1, v_2)$. Then the arc o is called an *output* arc for the node v_1 and a *producing* arc for the node v_2. A token in the node v_1 can *produce* through the arc o, a token in the node v_2 may be *produced* through the arc o.

It is impossible to define consuming output and producing input. The token may be producing, consuming, produced and consumed at the same time (through different incident arcs). It can be even self-copied (by producing loop) or self-consumed (by consuming loop).



(a) AR-net (b) Petri net

Fig. 1. A model of mutex semaphore with three processes and two resources (AR-model and an equivalent Petri net model).

An example of an AR-net is given on Fig. 1(a). Here we model a system, containing three processes that request two shared resources. Processes are modeled by tokens in the nodes `ready-to-start` and `ready-to-stop`, access keys – by tokens in the node `keys`. The model guarantees that at most two processes can work with resources at the same time. An example of a sequence of firings is given on Fig. 2 (here r_1 denotes `ready-to-start` and r_2 denotes `ready-to-stop`).

On Fig. 1(b) an equivalent Petri net is also presented. Note the difference between two nets. In AR-net the more simple node structure and the more complex arc structure allowed us to use the same node of the graph as a model for both place and transition. For example, the node `ready-to-start` is a replacement for both place `waiting` and transition `start`. Its tokens produce other tokens (in `ready-to-stop`) and are produced by other tokens (by `ready-to-stop`). They also consume other tokens (from `keys`) and are self-consumed (through the loop).

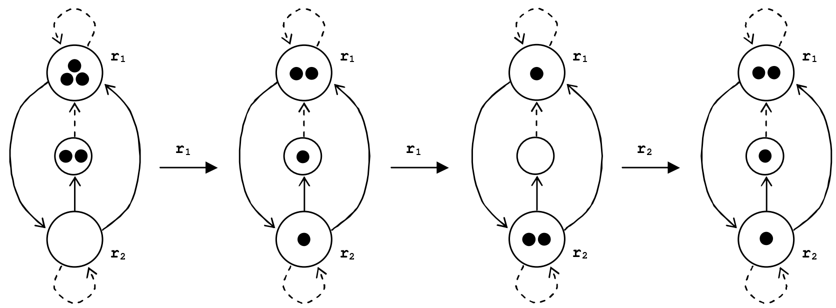


Fig. 2. A sequence of node firings in AR-net.

The notion of firing is extended to sequences in the standard way: for $\sigma \in V^*$ s.t. $\sigma = \sigma' v$ with $v \in V$ we say that $M \xrightarrow{\sigma} M'$ iff $M \xrightarrow{\sigma'} M'' \xrightarrow{v} M'$ for some M''.

A set of reachable markings is defined as follows:

$$\mathcal{R}(N, M_0) =_{\text{def}} \{M \in \mathcal{M}(V) \mid \exists \sigma \in T^* \ : \ M_0 \xrightarrow{\sigma} M\}.$$

A node v is *live* in a marked net (N, M_0) iff for any $M \in \mathcal{R}(N, M_0)$ there exists $M' \in \mathcal{R}(N, M)$ such that v is active in M'. A marked net is *live* iff all its nodes are live.

The reachability relation is defined as follows:

$$\mathcal{R}each(N, M_0) =_{\text{def}} \{(M, M') \in \mathcal{M}(V) \times \mathcal{M}(V) \mid \exists \sigma, \sigma' \in T^* \ : \ M_0 \xrightarrow{\sigma} M \xrightarrow{\sigma'} M'\}.$$

The syntax of AR-nets substantially differs from the syntax of Petri nets. It may be considered dual: instead of two types of vertices and a single type of arcs we use a single type of vertices and two types of arcs. However, AR-nets define the same class of systems and hence represent yet another variant of Petri net formalism:

Theorem 1. *[1] Nets of active resources are equivalent to Petri nets.*[1]

The proof of AR→PN transformation is based on the method, proposed in [10] for Super Dual nets. A node of the AR-net is replaced in the Petri net by a pair (place,transition) (as depicted in Fig. 3). This "flattening" allows to obtain a Petri net with the same reachability relation without any additional loops in the reachability graph.

3 Modular nets

Let $N = (V, I, O)$ be an AR-net. A *module* μ of the net N is defined by some subset of nodes $V_\mu \subseteq V$ (considered as internal nodes of the module).

For a module μ of N denote:

[1] For each AR-net there exists a Petri net with the same reachability and vice versa.

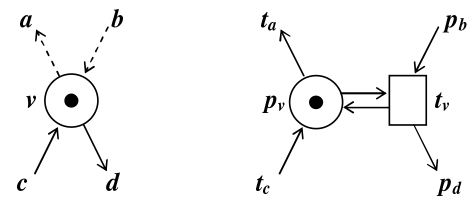


Fig. 3. Transformation of an AR-node into an equivalent Petri net.

- $I_\mu = \{(v, v') \in I \mid v, v' \in V_\mu\}$ – internal input arcs;
- $O_\mu = \{(v, v') \in O \mid v, v' \in V_\mu\}$ – internal output arcs;
- $N_\mu = (V_\mu, I_\mu, O_\mu)$ – a net of the module μ;
- $A^i_\mu = \{(v, v') \in I \mid v \in (V \setminus V_\mu), v' \in V_\mu\}$ – input links;
- $A^o_\mu = \{(v, v') \in O \mid v \in V_\mu, v' \in (V \setminus V_\mu)\}$ – output links;
- $R^i_\mu = \{(v, v') \in I \mid v \in V_\mu, v' \in (V \setminus V_\mu)\}$ – consuming links;
- $R^o_\mu = \{(v, v') \in O \mid v \in (V \setminus V_\mu), v' \in V_\mu\}$ – producing links.

Informally, A-links represents the observable activity of the module, R-links describe its role as a resource.

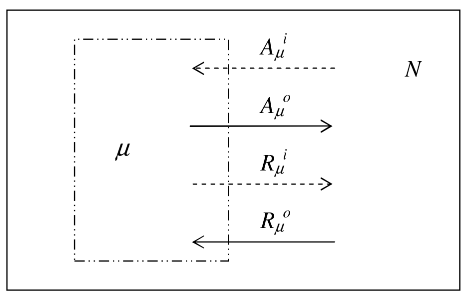


Fig. 4. Four link types in modular AR-nets.

For a marked net (N, M_0) and a module μ a marked net of the module $(N_\mu, (M_0)_\mu)$ is defined straightforwardly: $(M_0)_\mu =_{\text{def}} M_0[V_\mu]$.

Define also *a complement* $\overline{\mu}$ of the module μ as a module, defined by a subset of nodes $V \setminus V_\mu$. A complement of the module may be considered as *a system subnet* of the net.

A well-known model of dining philosophers is given on Fig. 5. For the sake of simplicity we consider only two participants. A module is defined representing the first philosopher. Note that it has only input and output links (elements of A^i_μ and A^o_μ), so the module may be considered as a pure agent.

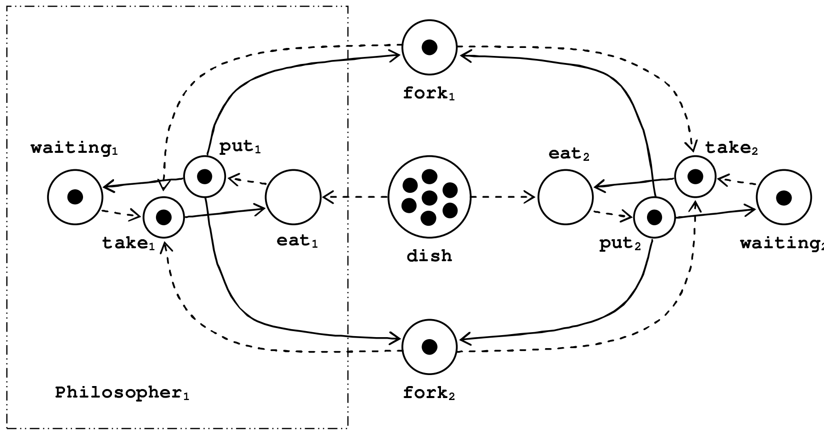


Fig. 5. Two dining philosophers.

A module in an AR-net has almost the same external appearance as a single node: it may consume and produce resources of other modules, and its own resources may be consumed and produced by other modules. Moreover, the relations between modules are naturally denoted by the same constructive elements as at the underlying level of nodes: input and output arcs (links). Hence, the induced hierarchical syntax is quite compact.

Modules having not all four types of links are of a particular interest. We will call a module μ an A-module (resp. R-module) if it has only A-links (resp. R-links). For example, the `philosopher`$_1$ is an A-module. Modules with a more restricted interfaces will be denoted using appropriate superscripts: for example, A^{i}R^o-module has only input and producing links.

Any AR-net may be considered as a composition of modules of different types (Fig. 6). Here are several trivial properties of some of these types:

Proposition 1. *Let (N, M_0) be a marked net and μ be a module of N. Then*

1. $(N_\mu, (M_0)_\mu)$ *is unbounded and μ is an $A^o R$-module* $\Rightarrow (N, M_0)$ *is unbounded;*
2. $(N_\mu, (M_0)_\mu)$ *is not live and μ is an AR^i-module* $\Rightarrow (N, M_0)$ *is not live;*
3. $(N_\mu, (M_0)_\mu)$ *is live and μ is an A^o-module* $\Rightarrow (N_{\overline{\mu}}, (M_0)_{\overline{\mu}})$ *is unbounded.*

Proof. 1. Since there are no input links, the behavior of the active nodes of the module does not depend on the marking of the system part of the net. Therefore we can take an unbounded run of the module as an unbounded run of the whole net.
2. The AR^i-module cannot obtain any additional tokens from the outside (there are no producing links). So its nodes are not live in the whole net too.
3. Obviously, the live module with only output external links sends the unbounded number of tokens to the outside.

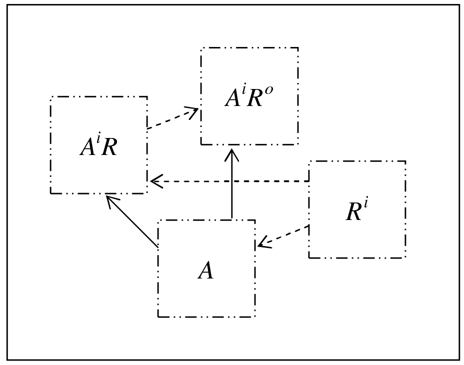


Fig. 6. Links define the role of the module in the system.

Among all 11 possible types of modules pure A- and R-modules (*agents* and *resources*) are the most important. Any interface between two modules may be transformed into an equivalent[2] A/R interface with one module being an agent, and the other being a resource. Consider a simple procedure transforming input and output links into producing and consuming links:

Lemma 1. *Let* (N, M_0) *be a marked AR-net and* $v \in V$ *be a node such that* $I(\bullet, v) \neq \varnothing$ *or* $O(v, \bullet) \neq \varnothing$ *(*v is active: it can consume or produce tokens*).*

Let N' *be a net, constructed from* N *by removing all arcs, participating in* $I(\bullet, v)$ *and* $O(v, \bullet)$ *(*v became passive*), and adding a new node* v_t *with* $I(v_t, \bullet) = O(\bullet, v_t) = \varnothing$ *(*v_t cannot be produced or consumed*),* $I(\bullet, v_t) = I(\bullet, v) \cup \{(v, v_t)\}$*,* $O(v_t, \bullet) = O(v, \bullet) \cup \{(v_t, v)\}$ *(*v_t simulate in N' the firing of v in N*).*

Let M_0' *be a marking of* N' *such that* $M_0'[V] = M_0$*,* $M_0'(v_t) = 1$*. Then*

$$\mathcal{R}each(N, M_0) = \mathcal{R}each(N', M_0') \cap (V \times V) \text{ and } \forall M' \in \mathcal{R}(N', M_0') \quad M'(v_t) = 1.$$

Proof. The illustration of such a transformation is given on Fig. 7.

The proof is straightforward – all possible links of a node are considered. Actually, we just separated active and passive properties of node v (just like in Fig. 3). The new node v_t is a *transition*: it behaves completely like an ordinary Petri net transition. Similarly, the node v in N' is a Petri net *place*.

The restructuring, described in Lemma 1, extends the set of nodes by a transition, *always* marked by a single token. So we do not take this node into account when considering the reachability set of the new net.

Corollary 1. *1) Any module of an AR-net may be transformed into R-module without changing the reachability set of the net;*
2) Any module of an AR-net may be transformed into A-module without changing the reachability set of the net.

[2] (w.r.t. reachability)

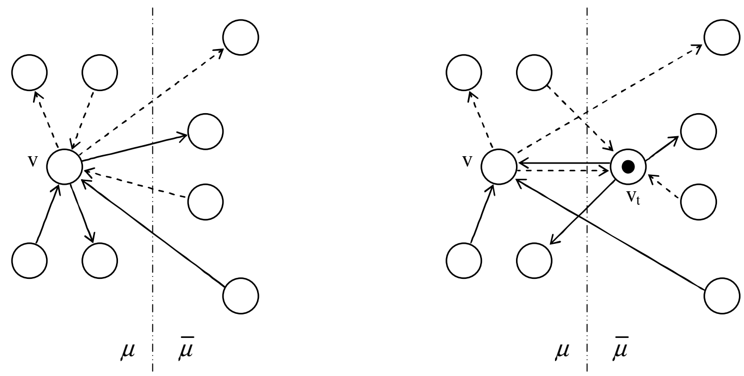


Fig. 7. Transformation of a node into a pair (place,transition).

Proof. (1) Any active node v of the module, having an external link, is replaced by a pair (v, v_t) of a place and a transition. The new transitions are put outside of the module (Fig. 7).

(2) A dual transformation: active nodes of the system part are transformed, transitions are put into the module.

So the interface of any module can be simplified to R-interface or A-interface. Of course, doing this we change the structure of the net (v_t is added). However, this modification is local and does not affect the "inner" part of the module.

In practice A- and R-modules may be considered as "active" and "passive" parts of the system ("control" and "data"). Corollary 1 states that the separation is "relative": we can easily modify an agent to be a resource and vice versa. This duality is quite trivial in modular AR-nets.

Definition 3. *A* flat modularization *Ω of a net N is a partition of V into non-intersecting modules $\{\mu_1, \ldots, \mu_n\}$.*

A flat modularization is called a flat A/R-modularization *iff every module is either A-module or R-module.*

Corollary 2. *Let $\Omega = \{\mu_1, \ldots, \mu_n\}$ be a flat modularization of N. Let G be a graph with vertices from Ω, such that two vertices μ_i and μ_j are connected in G iff there is an arc between modules μ_i and μ_j in N.*

The net N may be transformed into an equivalent (w.r.t. reachability) net N' such that Ω is an A/R-modularization of N' iff the chromatic number of G is 2.

Proof. Modules of one color are transformed into A-modules, of another – into R-modules.

Corollary 2 states that any two-color partition of the net can be transformed to an active/passive partition. Hence we can easely identify control and data structures, corresponding to the given partition: control modules sharing the

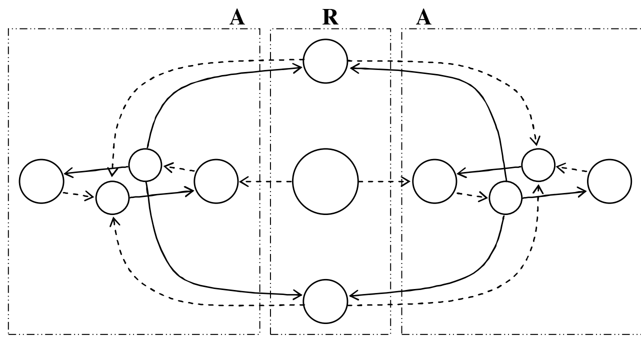


Fig. 8. Flat A/R-modularization.

same data, data modules sharing the same control, chains of linked modules etc. Moreover, control and data subnets are dualizable (Corollary 1).

Definition 4. *A* nested modularization *Ω of a net N is a partition of V into a module μ and a system part $\overline{\mu}$, where μ may also be modularized.*

A nested modularization is called a nested A/R-modularization / A-modularization / R-modularization *iff every module is either A- or R-module / A-module / R-module.*

Corollary 3. *For any nested modularization Ω of N this net may be transformed into an equivalent (w.r.t. reachability) net N' such that Ω is a nested α-modularization of N' ($\alpha \in \{A/R, A, R\}$).*

Proof. Straightforward. The transformation (if required) must be started from the innermost module.

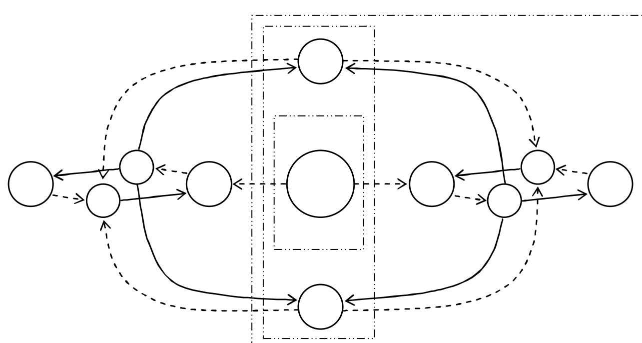


Fig. 9. Nested R-modularization.

The nested modularization allows us to construct a hierarchical structure with any kind of inter-level communication. The combinations of flat and nested modularizations are also possible.

A/R-modularizations are interesting because they allow to incorporate a natural hierarchy into the set of modules. Such hierarchy may have many applications in extended formalisms. For example, active modules may be considered responsible for intermodular data transfer. Another example is security: the R-module is able to hide the exact moments of its transition firings from the agent (the A-module), because agent observes only the already changed state of the resource.

4 Modular reachability

In this section we consider equivalent modules. The key problem is to check whether a particular module can be replaced by another one without harming the behaviour of the whole system. We study the equality of reachability relations of two nets – a fundamental behavioural equivalence, which is stronger than language equivalence and bisimulation.

Definition 5. *Consider AR-nets N_1 and N_2 and modules μ_1 and μ_2 of N_1 and N_2 respectively, such that $(N_1)_{\overline{\mu}_1} = (N_2)_{\overline{\mu}_2} = N_{sys}$ for some AR-net $N_{sys} = (V_{sys}, I_{sys}, O_{sys})$ (a same system net). Consider markings M_1, M_2 and M_{sys} of μ_1, μ_2 and N_{sys} respectively.*

Marked modules (μ_1, M_1) and (μ_2, M_2) are called equivalent w.r.t. system reachability *for a marked system net (N_{sys}, M_{sys}) (SR-equivalent for short) iff*

$$\mathcal{R}each(N_1, M_1 + M_{sys})[V_{sys}] = \mathcal{R}each(N_2, M_2 + M_{sys})[V_{sys}].$$

Informally, two SR-equivalent modules have the same effect on the system part of the net. They can be replaced by each other without harming the system's reachability set.

An example is given on Fig. 10. The module `Philosopher'` is SR-equivalent to the module $\texttt{Philosopher}_1$, shown on Fig. 5. Note that these modules are different: `Philosopher'` has additional state `ready`, in which it has both forks but is not eating.

Theorem 2. *SR-equivalence is undecidable for general AR-nets.*

Proof. Follows from the undecidability of R-equivalence for general Petri nets (a problem of deciding whether two nets have the same reachability set). Indeed, one can put all "agent nodes" ("transitions") of compared nets into corresponding modules (and all "resource nodes" aka "places" into system nets) and try to check their SR-equivalence.

Note that in the proof we used active modules (A-modules). Hence SR-equivalence is undecidable even for A-modules. It may be interesting to study

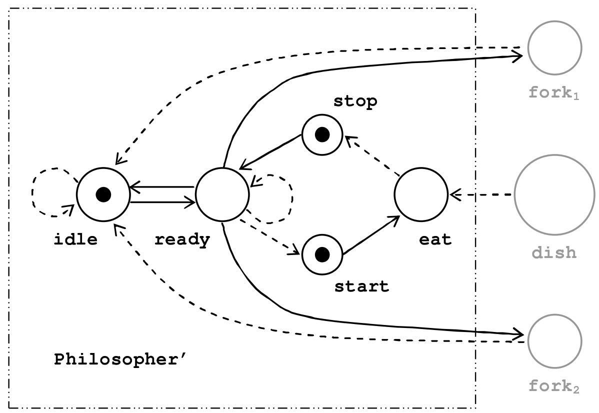


Fig. 10. An SR-equivalent philosopher.

the equivalence for other specific types of modules, in particular, for R-modules. We believe that it is undecidable as well.

Consider a more restricted case of a module. Let both modules have the same interface, i.e. the same set of links and adjacent nodes in the system net: $\forall v \in V_{sys}$

$$\sum_{v_1 \in V_{\mu_1}} A^o_{\mu_1}(v_1, v) = \sum_{v_2 \in V_{\mu_2}} A^o_{\mu_2}(v_2, v); \quad \sum_{v_1 \in V_{\mu_1}} R^i_{\mu_1}(v_1, v) = \sum_{v_2 \in V_{\mu_2}} R^i_{\mu_2}(v_2, v);$$
$$\sum_{v_1 \in V_{\mu_1}} A^i_{\mu_1}(v, v_1) = \sum_{v_2 \in V_{\mu_2}} A^i_{\mu_2}(v, v_2); \quad \sum_{v_1 \in V_{\mu_1}} R^o_{\mu_1}(v, v_1) = \sum_{v_2 \in V_{\mu_2}} R^o_{\mu_2}(v, v_2).$$

We will say that a module is *compatible* with a system net (and vice versa) iff the net contains nodes required by all links of the module.

Definition 6. *Marked modules (μ_1, M_1) and (μ_2, M_2), having the same interface, are called* universally equivalent w.r.t. system reachability *(USR-equivalent for short) iff they are SR-equivalent for any compatible marked system net.*

The USR-equivalence of two modules means that they produce equal sets of markings on passive interface nodes (by active agent links) and obey the same sets of restrictions and commands, coming from active interface nodes (by passive resource links).

A USR-equivalence is a restriction of an SR-equivalence. However, we believe that it is also undecidable.

Consider one of the simplest nontrivial module replacement – let the first module (denoted by μ) be a general AR-net (with some restrictions) and the second one (denoted by ν) be a single node.

Theorem 3. *Let (μ, M) be a marked module s.t.*

1. *All active interface nodes of module μ have the same multisets of active links:*

$$\exists A^i, A^o \in \mathcal{M}(V_{sys}) \quad \forall v \in V_\mu$$
$$(A^i_\mu(\bullet, v) = A^o_\mu(v, \bullet) = \varnothing) \ \vee \ (A^i_\mu(\bullet, v) = A^i \ \wedge \ A^o_\mu(v, \bullet) = A^o).$$

2. *All active interface nodes of system net perform operations on the whole numbers of the same multiset of internal nodes:*

$$\exists R^{io} \in \mathcal{M}(V_\mu) \quad \forall v \in V_{sys} \quad \exists k^i(v), k^o(v) \in Nat$$
$$(R^i_\mu(\bullet, v) = k^i(v) \times R^{io} \ \wedge \ R^o_\mu(v, \bullet) = k^o(v) \times R^{io}).$$

3. *All active internal nodes of the module, affecting nodes of R^{io}, perform the same operation on the whole numbers of R^{io}'s:*

$$\exists k^c, k^p \in Nat \quad \forall v \in V_\mu \quad (A^i_\mu(\bullet, v) \cap R^{io} \neq \varnothing \ \vee \ A^o_\mu(v, \bullet) \cap R^{io} \neq \varnothing) \quad \Rightarrow$$
$$\big(\exists X, Y \in \mathcal{M}(V_\mu) \quad (X \cap R^{io} = Y \cap R^{io} = \varnothing) \ \wedge$$
$$(A^i_\mu(\bullet, v) = k^c \times R^{io} + X) \ \wedge \ (A^o_\mu(v, \bullet) = k^p \times R^{io} + Y)\big).$$

4. *The initial marking M may be decomposed as $M = M' + m \times R^{io}$, where $m \in Nat$ and $M' \cap R^{io} = \varnothing$.*
5. *The marked internal net $(N_\mu, \overline{M})$ is live, were $\overline{M}$ denotes a marking, produced from M by emptying all passive interface nodes of μ :*

$$\forall v \in V_\mu \quad \overline{M}(v) =_{def} \begin{cases} 0 & \textit{if} \ \ (A^i_\mu(\bullet, v) \cup A^o_\mu(v, \bullet)) \neq \varnothing; \\ M(v) & \textit{otherwise.} \end{cases}$$

Then (μ, M) is USR-equivalent to a marked single-node module (ν, M_ν), where

- $V_\nu = \{w\}$;
- $I_\nu(w, w) = k^c$; $O_\nu(w, w) = k^p$;
- $A^i_\nu(\bullet, w) = A^i$; $A^o_\nu(w, \bullet) = A^o$;
- $\forall v \in V_{sys} \quad R^i_\nu(w, v) = k^i(v)$, $R^o_\nu(v, w) = k^o(v)$;
- $M_\nu(w) = m$.

Proof. The proof is technical. It is based on the fact that the liveness of the "unmarked" module implies the liveness of any node of the module in any bigger initial marking (such a marking may be obtained by input from the system). The system does not depend on the actual behaviour of the internal net (note that the effect of all interface firings is the same). It is enough to know that any firing is eventually possible.

An example of correct replacement is given on Fig. 11.

Here the module `Procedure` (live and unbounded) models some computation, that may be performed by one of three computers, initially positioned in the node `idle`. The service `loader` starts the computation, loading one of the computers with an input data. The computer performs calculations (may be, infinitely) and sometimes produces the results (to `output`). It also may be unloaded by another service `unloader`.

The external behaviour of module `Procedure` is relatively simple, so it can be replaced by a single node.

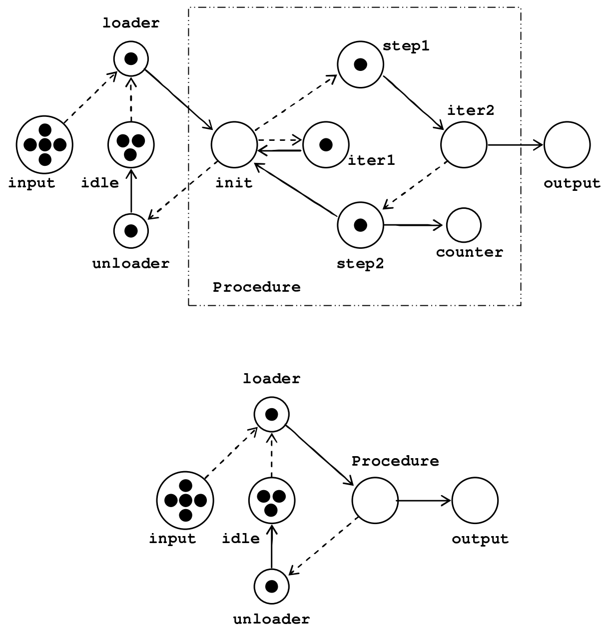


Fig. 11. A module, replaced by a USR-equivalent node.

5 Conclusion and Future work

We tried to identify the distinctive features of AR-nets, that would possibly allow them to be a successful base of some modular (and/or hierarchical) formalism. We also performed some simple analyses of expressiveness of specific modular constructs and decidability of basic module equivalences. It is also shown that modular AR-nets may be a convenient modeling tool.

Since AR-nets are expressively equivalent to general Petri nets, all the results, mentioned in the paper, can be applied to a standard Petri net syntax (in terms of places and transitions). However, in contrast to ordinary Petri nets, the set of nodes is homogeneous here and hence the syntax of module seems quite compact and natural.

We also believe that our work shows the opportunities provided by "coloured" arcs in Petri nets. Two-coloured arcs allowed us to remove the partition of nodes into places and transitions. Obviously, the process of generalization can go further, to a more complex/useful arcs/relations.

The possible directions of a future research in the area of modular AR-nets are: the decidability of certain equivalences of modules; the problem of finding the most effective (smallest/the least connected) decomposition of a given net; the refinement of nodes; the algebraic manipulations with AR-nets and modules; the synchronous compositions (requires additional constructs); etc.

References

1. V.A. Bashkin. Nets of active resources for distributed systems modeling. *Joint Bulletin of NCC&IIS, Comp. Science*. Novosibirsk. 2008. V.28. P.43–54.
2. V.A. Bashkin. Formalization of semantics of systems with unreliable agents by means of nets of active resources. *Programming and Computer Software*, 2010, Vol.36, No.4, P.187–196.
3. E. Best, R. Devillers, M. Koutny. Petri Net Algebra. *EATCS Monographs on TCS*. Springer, Berlin, 2001.
4. E. Best, W. Frączak, R.P. Hopkins, H. Klaudel, E. Pelz. M-nets: an algebra of high level Petri nets, with an application to the semantics of concurrent programming languages. *Acta Inf.*, 1998. Vol.35. P.813–857.
5. S. Christensen, L. Petrucci. Modular analysis of Petri nets. *The Computer Journal*, 2000. Vol.43(3). P.224–242.
6. K. Jensen. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Springer, 1994.
7. S. Haddad, D. Poitrenand. Theoretical aspects of recursive Petri nets. In *Proc. of ATPN'99*. LNCS 1639. Springer, 1999. P.228–247.
8. E. Kindler. A compositional partial order semantics for Petri net components. In *Proc. of ATPN'1997*. LNCS 1248. Springer, 1997. P.235–252.
9. K. Klai, S. Haddad, J.-M. Ilié. Modular Verification of Petri Nets Properties: A Structure-Based Approach. In *Proc. of FORTE'2005*. LNCS 3731. Springer, 2005. P.189–203.
10. M. Köhler, H. Rölke. Super-Dual Nets. In *Proc. of CS&P'2005*. P.271–280.
11. K. Lautenbach. Duality of Marked Place/Transition Nets. Universitat Koblenz-Landau, Institut fur Informatik, Research Report 18, 2003.
12. I.A. Lomazova. Nested Petri nets - a Formalism for Specification and Verification of Multi-Agent Distributed Systems. *Fundamenta Informaticae*. 2000. V.43. P.195–214.
13. C.-A. Petri. “Forgotten Topics” of Net Theory. In *Proc. of ATPN'1987*. P.500–514.
14. R. Valk. Petri Nets as Token Objects: An Introduction to Elementary Object Nets. LNCS 1420. Springer, 1998. P.1–25.