Programming and Computer Software, Vol. 27, No. 6, 2001, pp. 309–319.

Compositional Petri Net Approach to the Development of Concurrent and Distributed Systems

N. A. Anisimov, E. A. Golenkov, and D. I. Kharitonov

Institute for Automation and Control Processes, Far Eastern Division, Russian Academy of Sciences, ul. Radio 5, Vladivostok, 690041 Russia
e-mail: demiurg@iacp.vl.ru

Received November 10, 2000

Abstract—In the paper, a formal model based on Petri nets is proposed in the context of a compositional approach to the development and analysis of complex concurrent and distributed systems. Mutlilabels of Petri nets are introduced allowing labeling a transition not only with a single symbol, but also with a multiset of symbols. Operations on multilabeled Petri nets—parallel composition and restriction—are defined. A definition of a Petri net entity is given based on the notion of multilabels. A Petri net entity is a Petri net with a set of multilabels, where each multilabel is regarded as an access point of the entity. The operation of entity composition is introduced. Equivalence of entities is defined based on bisimulation equivalence of Petri nets. It is shown that the equivalence relation is congruent with respect to entity composition. It is also demonstrated that the composition operation is commutative and associative.

Key words: concurrent systems, distributed systems, Petri nets, Petri net entity, compositionality.

1. INTRODUCTION

It is well known that the formal approach is essential for the development of concurrent and distributed systems. Yet, a formal model can be used in practice only if it is *concurrent*, *structured*, and *abstract* [28]. Unfortunately, none of the existing formalisms has all these properties in full measure.

Indeed, although concurrent algebraic languages such as CCS [27], TCSP [5], and ACP [14] and their application-oriented versions LOTOS [18], occam [23], and PSF [26] are abstract and structured, they are not appropriate for representing concurrency. In particular, they treat concurrency of two events as interleaving, i.e., sequential firing of all possible event sequences. For instance, concurrency of events *a* and *b* is defined as firing of either the sequence *ab* or the sequence *ba*; this is definitely a limited understanding of concurrency, because it does not allow for simultaneous firing of events.

Another group of formalisms, which is based on Petri net theory [3, 30], makes it possible to represent concurrency in a more appropriate and natural way. However, these formalisms are neither abstract nor structured, which hinders their application to a wide range of practical problems. In recent years, serious researches have been conducted in order to overcome these drawbacks. Some of them are focused on replacing the interleaving semantics of algebraic languages with the semantics of the so-called "true" concurrency based on net models. Such attempts were made for CCS [22, 32], TCSP [28, 32], ACP [21], as well as for the LOTOS [11] and occam [23] languages. It should be noted that these approaches faced serious difficulties, because abstract concurrent algebraic languages had been initially designed without net semantics in mind. Among other things, they all are as expressive as the Turing machine, whereas Petri nets are less expressive [3]. This disagreement stimulates search for subsets of abstract languages the semantics of which could be expressed in terms of Petri nets [11, 22].

Another approach to the same problem is to make Petri nets structured and abstract without regard to any languages. Works in this direction try to enrich compositional capacities of Petri nets by defining the notion of interface for a Petri net and a set of composition rules [15]. Several papers employed places for such an interface [19, 25, 33], and others employed transitions [12, 34, 35]. Composition of nets was performed by merging places or transitions respectively. The major advances in this direction were made in the framework of the Caliban project and the earlier DEMON project [31] of the European Program Esprit. The result of these researches is the Petri Box Calculus (PBC) discussed in [17].

The PBC model, called *Petri box*, is essentially a Petri net with an interface defined as a labeling function. This function maps a net transition to a multiset of elementary actions. The operator of net synchronization uses this information to generate a set of synchronization transitions by merging communicating transitions. It is assumed that an elementary action is the name of a communication channel, which may have parameters.

The next step along this line is further structuring of events, which makes it possible to determine communication directions more explicitly. This approach allows for the nature of distributed systems, where the developer has to specify system units and their interaction. The notion of the Petri net entity, which is a generalization of labeled Petri nets, was introduced within this framework. The single labeling function is replaced by several ones called access points. Each access point maps a transition to a multiset of names, where each name is treated as an elementary operation of communication with other entities. As compared to PBC, the communication directions are determined explicitly, which results in splitting a single labeling function into several ones.

The Petri net entity (without a multilabel) was first introduced in [1] and elaborated in [7, 8]. However, in these papers, the emphasis was on application of the formalism to design of computer network protocols. This paper is devoted to further development of the technique of Petri net entities, its formal definition, and its generalization to multilabeled entities.

2. MULTILABELING PETRI NETS

Let $A = \{a_1, a_2, \dots, a_k\}$ be some set. A multiset over the set A is a function $\mu: A \longrightarrow \{0, 1, 2, \dots\}$ associating each element of the set A with a nonnegative integer. Sometimes, it is convenient to represent a multiset over the set A as a formal sum $n_1a_1 + n_2a_2 + \dots + n_ka_k$ or $\sum n_i a_i$, where $n_i = \mu(a_i)$ is the number of occurrences of $a \in A$ in the multiset. As a rule, members of the sum with $a_i = 0$ are dropped. The union and intersection of two multisets $\mu_1 = n_1a_1 + \dots + n_ka_k$ and $\mu_2 = m_1a_1 + \dots + m_ka_k$ over the set A are defined as $\mu_1 + \mu_2 = (n_1 + m_1)a_1 + \dots + (n_k + m_k)a_k$ and $\mu_1 - \mu_2 = (n_1 - m_1)a_1 + \dots + (n_k - m_k)a_k$, respectively; intersection is defined only when $n_i - m_i \geq 0$ for all $1 \leq i \leq k$. We write $\mu_1 \leq \mu_2$ if $n_i \leq m_i$ for any $1 \leq i \leq k$, and we write $\mu_1 < \mu_2$ if $\mu_1 \leq \mu_2$ and $\mu_1 \neq \mu_2$. If $n_i = 0$ for all i, such a multiset is denoted by $\mathbf{0}$. We write $a \in \mu$ if $\exists n > 0$: $(a, n) \in \mu$.

The set of all finite multisets over the set A is designated by $\mathcal{M}(A)$. The set of all possible sequences of symbols from the set A, including the empty string ϵ, is denoted by A^*.

Suppose that f: $A \longrightarrow B$ is a function, and $X \subseteq A$. Then $f\lceil X$ is the projection of f on X defined as $f\lceil X = \{(a, b) \in f \mid a \in X\}$.

Definition 2.1. A *Petri net* is a set $\Sigma = \langle S, T, {}^\bullet(), ()^\bullet, M_0\rangle$, where

(1) S is a finite set of *places*;

(2) T is a finite set of *transitions* such that $S \cap T = \varnothing$;

(3) ${}^\bullet() : T \longrightarrow \mathcal{M}(S)$ is an input incidence function;

(4) $()^\bullet : T \longrightarrow \mathcal{M}(S)$ is an output incidence function;

(5) $M_0 \in \mathcal{M}(S)$ is an initial marking.

The multisets ${}^\bullet t$ and $t^\bullet$ are called the input and output sets of the transition $t \in T$, respectively. The functions ${}^\bullet()$ and $()^\bullet$ can be naturally extended to multisets: ${}^\bullet(n_1t_1 + \dots + n_kt_k) = n_1^\bullet t_1 + \dots + n_k^\bullet t_k$, $(n_1t_1 + \dots + n_kt_k)^\bullet = n_1t_1^\bullet + \dots + n_kt_k^\bullet$.

In what follows, we use traditional graphic representation of a Petri net as a bipartite graph, where circles denote places and rectangles denote transitions. Places and transitions are linked by directed arcs representing the input and output incidence functions. The weight of arcs is specified by integers placed near arcs. Marking is depicted by tokens positioned inside places.

In this paper, we use step semantics of Petri nets based on firing of sequences of transition multisets.

The marking M of the net $\Sigma = \langle S, T, {}^\bullet(), ()^\bullet, M_0\rangle$ is a multiset over S; i.e., $M \in \mathcal{M}$. We say that a step (a transition multiset) $\Theta \in \mathcal{M}(T)$ is enabled in the marking M if ${}^\bullet\Theta \leq M$. The step $\Theta \in \mathcal{M}(T)$ enabled in the marking M can fire yielding a new marking M', which is denoted as $M[\Theta\rangle M'$, where $M' = M - {}^\bullet\Theta + \Theta^\bullet$. It should be noted that, if the step Θ is enabled in the marking M, then the step $\Theta' < \Theta$ is also enabled in the marking M. If $\Phi = \Theta_1\Theta_2\dots\Theta_n \in (\mathcal{M}(T))^*$, then $M[\Theta\rangle M'$ denotes that there is a sequence $M_1, M_2, \dots, M_{n-1}$ such that $M[\Theta_1\rangle M_1[\Theta_2\rangle\dots M_{n-1}[\Theta_n\rangle M'$. In this case, we say that M' is reachable from M. $M[\rangle M'$ denotes that there is $\Phi \in \mathcal{M}(T)^*$ such that $M[\Phi\rangle M'$. The set of all markings reachable from M is defined as $[M\rangle = \{M' \mid M[\rangle M'\}$. If the net to which the statement refers is to be indicated, an appropriate prefix should be used. For instance, Σ: $M[\Phi\rangle M'$, $\Sigma_1 : M \leq {}^\bullet\Theta$.

Let Δ be a finite alphabet of names and $\bar{\Delta} = \{\bar{a} \mid a \in \Delta\}$ be an associated alphabet of complementary names. In other words, we define a one-to-one correspondence $\bar{\cdot}: \Delta \longrightarrow \bar{\Delta}$ between the names and their complementary names. For the sake of simplicity, the inverse function $\bar{\cdot}$ is denoted by the same symbol: $\bar{\cdot}: \bar{\Delta} \longrightarrow \Delta$. Thus, we have $\bar{\bar{a}} = a$. Let $\tau \notin \mathcal{M}(\Delta \cup \bar{\Delta})$ be a special symbol associated with an invisible action. Denote $Vis = \Delta \cup \bar{\Delta}$ and $Act = \mathcal{M}(Vis) \cup \{\tau\}$. The function $\bar{\cdot}$ can be extended to the multiset of names $\overline{n_1a_1 + \dots + n_ka_k} = n_1\bar{a}_1 + \dots + n_k\bar{a}_k$. For example:

$$\overline{\bar{a} + 2a + 3\bar{b} + c} = a + 2\bar{a} + 3b + \bar{c}.$$

Definition 2.2. A *label* of a Petri net $\Sigma = \langle S, T, {}^\bullet(), ()^\bullet, M_0\rangle$ is the tuple $\lambda = \langle\Delta, \sigma\rangle$, where Δ is some alphabet and σ: $T \longrightarrow Act$ is a labeling function.

This definition is an extension of a well-known definition of the labeling function. In particular, every transition can be labeled not only with a single symbol but also with a multiset of symbols. A transition labeled by τ is considered an invisible or internal transition.

On the physical level, a label denotes elementary communication. For example, the symbol $a \in \Delta$ is assumed to correspond to sending a message with the name a, and the complementary symbol $\bar{a}$, to receiving a message with the name a. The symbol τ is not related to the communication but describes some internal event. If a transition is labeled with a multiset of symbols, the latter corresponds to simultaneous receiving and/or sending messages upon firing the transition. Thus, if $\sigma(t) = \bar{a} + 2b$, then the message a is received and two messages b are sent when the transition t fires.

Let us show how the labeling function σ is extended to the multiset of transitions, $\sigma: \mathcal{M}(T) \longrightarrow Act$. If $\Theta = \sum n_i a_i \in \mathcal{M}(T)$, then $\sigma(\Theta) = \sum n_i \sigma(a_i)$. If every transition t in the step Θ is invisible, i.e., $\sigma(t) = \tau$, we write $\sigma(\Theta) = \tau$.

The function σ can be naturally extended to the homomorphism $\sigma: (\mathcal{M}(T))^* \longrightarrow (\mathcal{M}(Act))^*$. Define the function $\sigma^+: (\mathcal{M}(T))^* \longrightarrow (Vis)^*$ eliminating all symbols τ from sequences:

$$\sigma^+(\Theta) = \begin{cases} \epsilon, & \text{if } \sigma(\Theta) = \tau; \\ \sigma(\Theta), & \text{otherwise;} \end{cases}$$

$$\sigma^+(\Phi\Theta) = \sigma^+(\Phi)\sigma^+(\Theta).$$

In what follows, the symbol Θ denotes a step and the symbol Φ denotes a sequence of steps.

If $W \in (\mathcal{M}(Vis))^*$ and $\lambda = \langle \Delta_\lambda, \sigma_\lambda \rangle$ are labels of the Petri net Σ, then $M(W\rangle_\lambda M'$ denotes that $\Phi \in (\mathcal{M}(T))^*$: $M[\Phi\rangle M'$ and $\sigma_\lambda^+(\Phi) = W$. For the sake of brevity, the sequence $M[\Phi\rangle M'$ $\sigma(\Phi) = \epsilon$ is written as $M \Rightarrow M'$, and the symbol $\Rightarrow$ denotes an invisible sequence of steps.

Definition 2.3. Suppose that $\Sigma_1 = \langle S_1, T_1, {}^\bullet()_1, ()_1^\bullet, M_{01} \rangle$ is a net and $\alpha = \langle \Delta_\alpha, \sigma_\alpha \rangle$ is its label. Then the α-restriction of the net Σ is a new net $\Sigma = \partial_\alpha(\Sigma_1)$ satisfying the following conditions:

(1) $S = S_1$;

(2) $T = T_1 \backslash \{t \in T \mid \sigma_\alpha(t) \neq \tau\}$;

(3) ${}^\bullet(t) = {}^\bullet(t)_1$, $t \in T$;

(4) $(t)^\bullet = (t)_1^\bullet$, $t \in T$;

(5) $M_0 = M_{01}$.

Less formally, a restriction of the net eliminates each transition labeled by a name from *Vis* together with the adjacent arcs. Figure 1 illustrates this operation. Here, the transitions t_1 and t_3 labeled by names from *Vis* are eliminated, and the invisible τ-transition remains.

Suppose that, for the net $\Sigma = \partial_\alpha(\Sigma_1)$, there are two labels $\alpha = \langle \Delta_\alpha, \sigma_\alpha \rangle$ and $\beta = \langle \Delta_\beta, \sigma_\beta \rangle$. The label β for the net Σ can be naturally restricted as follows: $\tilde{\beta} = \langle \Delta_\beta, \sigma_\beta \lceil T \rangle$.

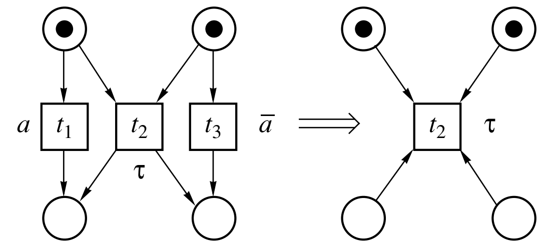


Fig. 1. Application of the restriction operation.

We will write β instead of $\tilde{\beta}$ when this does not result in any confusion.

Proposition 2.4. Let Σ be a net and α and β be its labels. Then, the following equality holds: $\partial_{\tilde{\alpha}}(\partial_\beta(N)) = \partial_{\tilde{\beta}}(\partial_\alpha(N))$.

Proof. Divide T into four disjoint subsets: $T = T_{11} \cup T_{12} \cup T_{21} \cup T_{22}$, where

$$T_{11} = \{t \in T \mid \sigma_\alpha(t) \neq \tau \neq \sigma_\beta(t)\},$$

$$T_{12} = \{t \in T \mid \sigma_\alpha(t) \neq \tau = \sigma_\beta(t)\},$$

$$T_{21} = \{t \in T \mid \sigma_\alpha(t) = \tau \neq \sigma_\beta(t)\},$$

$$T_{22} = \{t \in T \mid \sigma_\alpha(t) = \tau = \sigma_\beta(t)\},$$

$$\partial_{\tilde{\alpha}}(\partial_\beta(\Sigma)) = \partial_{\tilde{\alpha}}(\langle S, T \backslash T_{12} \cup T_{22},$$

$${}^\bullet() \lceil (T_{11} \cup T_{21}), ()^\bullet \lceil (T_{11} \cup T_{21}), M_0 \rangle)$$

$$= \langle S, T_{11}, {}^\bullet() \lceil T_{11}, ()^\bullet \lceil T_{11}, M_0 \rangle$$

$$\partial_{\tilde{\beta}}(\partial_{\tilde{\alpha}}(\Sigma)) = \partial_{\tilde{\beta}}(\langle S, T \backslash T_{21} \cup T_{22},$$

$${}^\bullet() \lceil (T_{11} \cup T_{12}), ()^\bullet \lceil (T_{11} \cup T_{12}), M_0 \rangle)$$

$$= \langle S, T_{11}, {}^\bullet() \lceil T_{11}, ()^\bullet \lceil T_{11}, M_0 \rangle.$$

This proposition makes it possible to extend the restriction operation to the set of labels.

Definition 2.5. Let Σ be a Petri net and $H = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ be a set of its labels. Then, $\partial_H(\Sigma) = \partial_{\alpha_1} \circ \partial_{\alpha_2} \circ \ldots \circ \partial_{\alpha_n}(\Sigma)$.

3. CONCURRENT COMPOSITION OF MULTILABELED PETRI NETS

Concurrent composition is a very important part of the compositional approach, since it allows constructing systems with interacting components. There are several definitions of this operation for Petri nets [21, 22, 25, 32]. These definitions are applicable only for one-to-one communications, i.e., when a transition of one net is merged with at most one transition of another net. However, we need a generalized concurrent com-

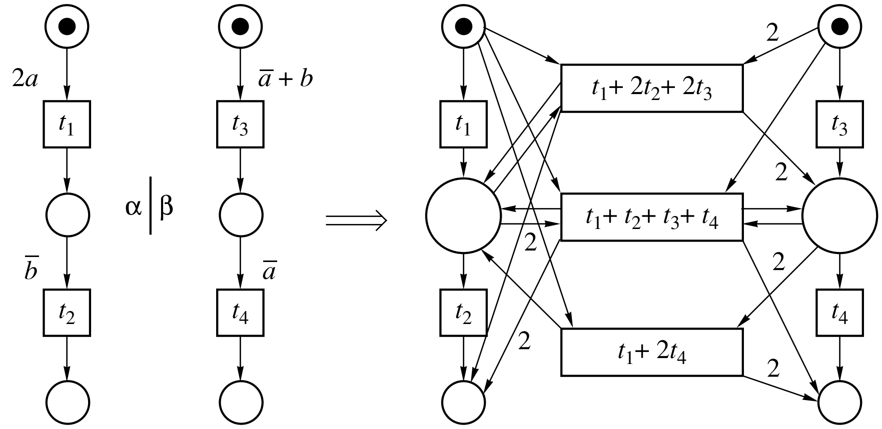


Fig. 2. Concurrent composition of Petri nets.

position for the case of the multicommunication consistent with the definition of a multilabel.

Suppose that $\Sigma_1 = \langle S_1, T_1, {}^\bullet()_1, ()_1^\bullet, M_{01}\rangle$ and $\Sigma_2 = \langle S_2, T_2, {}^\bullet()_2, ()_2^\bullet, M_{02}\rangle$ are two Petri nets. Suppose also that the nets Σ_1 and Σ_2 have disjoint sets of places and transitions; i.e., $S_1 \cap S_2 = T_1 \cap T_2 = \varnothing$.

Definition 3.1. Let Σ_1 and Σ_2 be Petri nets with labels $\alpha = \langle \Delta_\alpha, \sigma_\alpha\rangle$ and $\beta = \langle \Delta_\beta, \sigma_\beta\rangle$, respectively. *Concurrent composition* of the nets Σ_1 and Σ_2 with respect to α and β is a new net $\Sigma = (\Sigma_1 {}_\alpha|_\beta \Sigma_2)$ such that

(1) $S = S_1 \cup S_2$;

(2) $T = T_1 \cup T_2 \cup T_1 {}_\alpha\otimes_\beta T_2$, where

$$T_1 {}_\alpha\otimes_\beta T_2 = \{\mu_1 + \mu_2 \mid \mu_1 \in \mathcal{M}(T_1), \mu_2 \in \mathcal{M}(T_2),$$

$$\tau \notin \sigma_\alpha(\mu_1) = \overline{\sigma_\beta(\mu_2)}, \text{ the sum } \mu_1 + \mu_2 \text{ is minimal}\};$$

(3) ${}^\bullet() = {}^\bullet()_1 \cup {}^\bullet()_2 \cup \{(\mu_1 + \mu_2, {}^\bullet(\mu_1)_1 + {}^\bullet(\mu_2)_2) \mid \mu_1 + \mu_2 \in T, \mu_1 \in \mathcal{M}(T_1), \mu_2 \in \mathcal{M}(T_2)\}$;

(4) $()^\bullet = ()_1^\bullet \cup ()_2^\bullet \cup \{(\mu_1 + \mu_2, (\mu_1)_1^\bullet + (\mu_2)_2^\bullet) \mid \mu_1 + \mu_2 \in T, \mu_1 \in \mathcal{M}(T_1), \mu_2 \in \mathcal{M}(T_2)\}$;

(5) $M_0 = M_{01} + M_{02}$.

The sum $\mu_1 + \mu_2$ is minimal if there is no sum $\mu_1' + \mu_2'$ such that $\mu_1' + \mu_2' < \mu_1 + \mu_2$ and $\sigma_\alpha(\mu_1) = \overline{\sigma_\beta(\mu_2)}$.

Less formally, two nets Σ_1 and Σ_2 are merged, and new synchronization transitions $T_1 {}_\alpha\otimes_\beta T_2$ are added. These new transitions are specified by multisets of symbols $\mu_1 + \mu_2$, $\mu_1 \in \mathcal{M}(T_1)$, $\mu_2 \in \mathcal{M}(T_2)$. For new transitions, their input and output multisets are computed: ${}^\bullet(\mu_1 + \mu_2) = {}^\bullet(\mu_1) + {}^\bullet(\mu_2)$, $(\mu_1 + \mu_2)^\bullet = (\mu_1)^\bullet + (\mu_2)^\bullet$. When it does not result in ambiguity, we use $T_1 \otimes T_2$ instead of $T_1 {}_\alpha\otimes_\beta T_2$.

Figure 2 gives an example of concurrent composition. In this example, three synchronization transitions are added. The first transition is $(t_1 + 2t_2) + 2t_3$, where $\sigma_\alpha(t_1 + 2t_2) = \overline{\sigma_\beta(2t_3)} = 2a + 2\bar{b}$; the second one is $(t_1 + t_2) + (t_3 + t_4)$ with $\sigma_\alpha(t_1 + t_2) = \overline{\sigma_\beta(t_3 + t_4)} = 2a + \bar{b}$; and the third one is $t_1 + 2t_4$ with $\sigma_\alpha(t_1) = \overline{\sigma_\beta(2t_4)} = 2a$. If, e.g., t_4 is renamed with τ, the second and the third synchronization transitions are not formed.

Obviously, the operation of concurrent composition introduced above becomes practically useful if we supplement it with a procedure for computing the set of synchronization transitions $T_1 \otimes T_2$. It turns out that this problem can be reduced to the well-known problem of finding the set of minimal invariants for a Petri net [2].

Theorem 3.2. *Finding synchronization transitions.* The problem of finding the set of synchronization transitions for parallel composition of Petri nets can be reduced to the problem of finding the least set of invariants of the Petri net.

Proof. Let $\alpha = \langle \Delta_\alpha, \sigma_\alpha\rangle$ and $\beta = \langle \Delta_\beta, \sigma_\beta\rangle$ be labeling functions of Petri nets Σ_1 and Σ_2 respectively. Let $\Delta = \Delta_\alpha \cup \Delta_\beta$, $T_1 = \{t_1^1, t_2^1, \ldots, t_n^1\}$, and $T_2 = \{t_1^2, t_2^2, \ldots, t_m^2\}$. Construct a new Petri net as follows: $\tilde{\Sigma}_{12} = \langle \Delta, T_1 \cup T_2, {}^\bullet(), ()^\bullet, M_0\rangle$, where $M_0 = 0$,

$$(t)^\bullet = \begin{cases} \sigma_\alpha^+(t) \lceil \Delta, & \text{if } \ t \in T_1; \\ \sigma_\beta^+(t) \lceil \Delta, & \text{if } \ t \in T_2; \end{cases}$$

$${}^\bullet(t) = \begin{cases} \overline{\sigma_\alpha^+(t)} \lceil \Delta, & \text{if } \ t \in T_1; \\ \overline{\sigma_\beta^+(t)} \lceil \Delta, & \text{if } \ t \in T_2. \end{cases}$$

Here, the projection $\sigma^+(t) \lceil \Delta$ preserves only direct names from Δ. Note that, when the transition t_i^1 fires, $\sigma_\alpha^+(t_i^1) \lceil \Delta$ is added to the current marking and $\overline{\sigma_\alpha^+(t_i^1)} \lceil \Delta$ is removed from it. When the transition t_j^2 fires, $\sigma_\beta^+(t) \lceil \Delta$ is added to the current marking and $\overline{\sigma_\beta^+(t)} \lceil \Delta$ is removed from it. Therefore, it is possible to return to the zero marking by firing the sequence $v \in (T_1 \cup T_2)^*$, $M_0|v\rangle M_0$ only if $\sigma_\alpha(x) = \overline{\sigma_\beta(y)}$. Here, x and y are multisets of transitions from T_1 and T_2 occurring in v. On the other hand, it is known that, if there is $v \in (T_1 \cup T_2)^*$: $M_0[v\rangle M_0$, then the eigenvector v is a T-invariant. Moreover, if $x + y$ is minimal, the eigenvector of the multiset $x + y$ is the minimal T-invariant of the net $\tilde{\Sigma}_{12}$. Next, invariants containing no transitions from T_1 or T_2 are removed from the set obtained. This is necessary since we are concerned only in the communication between the nets rather than in autocommunication, when the net communicates with itself.

Thus, to generate the set of the synchronization transitions $T_1 \otimes T_2$, it is necessary to construct the net Σ_{12} and compute its set of minimal T-invariants. Every such invariant $f = \langle f_1, \ldots, f_{n+m}\rangle$ produces one synchronization transition $f_1t_1^1 + \ldots + f_nt_n^1 + f_{n+1}t_1^2 + \ldots + f_{n+m}t_m^2$. There are many algorithms of finding T-invariants for Petri nets based on the Farcas algorithm. Some of them are discussed in [2]. In the general case, the problem of computing the set $T_1 \otimes T_2$ has exponential complexity, although, some heuristics may be used to simplify the problem [6].

Back to Fig. 2, it is possible to construct a net $\tilde{\Sigma}_{12}$ (see Fig. 3). It can be easily seen that the net $\tilde{\Sigma}_{12}$ has three minimal invariants: $t_1 + 2t_4$, $t_1 + 2t_2 + 2t_3$, and $t_1 + t_2 + t_3 + t_4$.

Now, we need to define an extension of a label γ of the net Σ_1 to the net $(\Sigma_1\ {}_\alpha|_\beta\ \Sigma_2)$, which will be denoted by $\tilde{\gamma}$.

Definition 3.3. Suppose that $\alpha = \langle\Delta_\alpha, \sigma_\alpha\rangle$ and $\gamma = \langle\Delta\gamma, \sigma_\gamma\rangle$ are labels of the net Σ_1 and $\beta = \langle\Delta_\beta, \sigma_\beta\rangle$ is a label of the net Σ_2. Then, the *extension* of γ to the net $(\Sigma_1\ {}_\alpha|_\beta\ \Sigma_2)$ is the label $\tilde{\gamma} = \langle\Delta_\gamma, \tilde{\sigma}_\gamma\rangle$, where

$$\tilde{\sigma}_\gamma = \sigma_\gamma \cup \{(t, \tau) \mid t \in (T_2)\}$$
$$\cup \{(x+y, \sigma_\gamma(x)) \mid x + y \in T_1 \otimes T_2,$$
$$x \in \mathcal{M}(T_1), y \in \mathcal{M}(T_2)\}.$$

In other words, the label of transitions from T_1 remains unchanged, the label of the synchronization transition $t \in T_1 \otimes T_2$ in γ is defined as a sum of labels of the corresponding transitions, and transitions from T_2 are labeled by τ.

We will need projection functions for the synchronization transition of the resulting net $\Sigma = (\Sigma_1\ {}_\alpha|_\beta\ \Sigma_1)$. These projections ν_1: $T_1 \otimes T_2 \longrightarrow \mathcal{M}(T_1)$ and ν_2: $T_1 \otimes T_2 \longrightarrow \mathcal{M}(T_2)$ are defined as follows. Let $t = \mu_1 + \mu_2 \in T_1 \otimes T_2$, where $\mu_1 \in \mathcal{M}(T_1)$ and $\mu_2 \in \mathcal{M}(T_2)$. Then, $\nu_1(t) = \mu_1$ and $\nu_2(t) = \mu_2$. Hence, we can write $t = \mu_1(t) + \mu_2(t)$. It is also possible to naturally extend ν_1 and ν_2 to

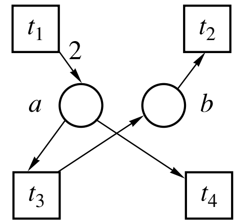


Fig. 3. An example of construction of the net $\tilde{\Sigma}_{12}$.

the transition multisets ν_i: $\mathcal{M}(T_1 \otimes T_2) \longrightarrow \mathcal{M}(T_i)$, $i \in \{1, 2\}$ as follows: $\nu_i(\Theta) = \sum \Theta(t)\nu_i(t)$.

Next, the following dependences are established between the behavior of the initial nets and the resulting nets.

Proposition 3.4. Let $\Sigma = (\Sigma_1\ {}_\alpha|_\beta\ \Sigma_2)$ be a composition of Petri nets. Suppose also that $M_1, M_1' \in \mathcal{M}(S_1)$ and $M_2, M_2' \in \mathcal{M}(S_2)$.

(1) If $\Theta \in T_1 \otimes T_2$ and Σ: $M_1 + M_2[\Theta\rangle M_1' + M_2'$, then Σ_1: $M_1\ [\nu_1(\Theta)\rangle\ M_1'$ and Σ_2: $M_2\ [\nu_2(\Theta)\rangle\ M_2'$.

(2) If Σ_1: $M_1[\Theta_1\rangle M_1'$, Σ_2: $M_2\ [\Theta_2\rangle\ M_2'$ and $\tau \ne \sigma_\alpha(\Theta_1) = \overline{\sigma_\beta(\Theta_2)}$, then there exists $\Theta \in T_1 \otimes T_2$: $\nu_1(\Theta) = \Theta_1$, $\nu_2(\Theta) = \Theta_2$.

Proof follows from the definition of entity composition.

The first part of Proposition 3.4 says that each firing of the step Θ in the resulting net corresponds to the firing of the steps Θ_1 and Θ_2 in the initial nets; the latter steps are projections of the former one: $\Theta_1 = \nu_1(\Theta)$ and $\Theta_2 = \nu_2(\Theta)$. The second part says that, if the steps Θ_1 and Θ_2 can fire in the initial nets with the same visibility status, i.e., $\sigma_\alpha(\Theta_1) = \overline{\sigma_\beta(\Theta_2)}$, then, in the resulting net, the step Θ whose projections are the initial steps Θ_1 and Θ_2 can fire.

4. FORMAL DEFINITION OF A PETRI NET ENTITY

As it was mentioned in Introduction, a Petri net entity is a logical unit with several access points intended for communication with other units. A Petri net entity can be represented schematically as a rectangle with outgoing lines denoting access points. Figure 4a gives an example of schematic representation of an entity. Below is the formal definition of a Petri net entity.

Definition 4.1. A *Petri net entity* (PN entity or just entity for the sake of brevity) is a tuple $E = \langle\Sigma, \Gamma\rangle$ such that

(1) $\Sigma = \langle S, T, {}^\bullet(), ()^\bullet, M_0\rangle$ is a Petri net called the structure of the entity;

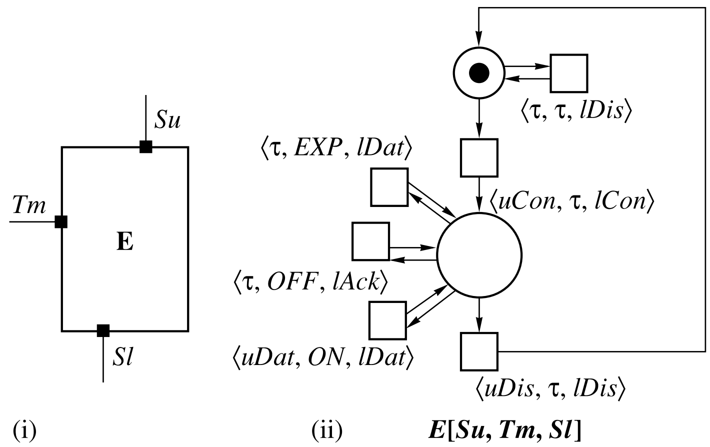


Fig. 4. A sample entity.

(2) $\Gamma = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ is a set of access points, which each point having the form $\alpha_i = \langle id_i, \lambda_i \rangle$, where

(a) id_i is the name of the access point α_i and

(b) $\lambda_i = \langle \Delta_i, \sigma_i \rangle$ is the multilabel of the net Σ.

In what follows, the PN entities are denoted by (possibly indexed) letters E, F, and G; the access points are denoted by Greek letters α, β, and γ. The set of names of the entity $E = \langle \Sigma, \Gamma \rangle$ is defined by means of the function $Id(E) = \{id_a \mid \alpha \in \Gamma\}$. We often write $\alpha_i = \langle id_i, \Delta_i, \sigma_i \rangle$. Thus, each access point α_i is defined by its name id_i, an alphabet Δ_i of names, and a labeling function σ_i mapping each transition either to a symbol τ or to a multiset over the set $\Delta_i \cup \overline{\Delta_i}$.

The definition of the entity requires some explanations. A PN entity is nothing but a Petri net with a set of multilabels; i.e., the entity generalizes the notion of labeled Petri nets. Communication with an entity is possible only through its access points. In particular, watching the behavior of the net is a special case of communication. Clearly, an entity may behave differently in different access points. A transition labeled by τ in some access point is invisible in this access point and cannot be used for the communication with the entity. The same transition can be simultaneously visible in several access points, but possibly under different names. Moreover, a transition can be visible in one point and invisible in another. Finally, a transition can be invisible in all access points.

An entity E with access points $\alpha_1, \ldots, \alpha_n$ is denoted by $E[\alpha_1, \ldots, \alpha_n]$. Graphically, the entities are represented as Petri nets where each transition is labeled by the set $\langle \alpha_1, \ldots, \alpha_n \rangle \in Act_1 \times \ldots \times Act_n$. Obviously, the choice of sets S and T in the entity definition is not important from the standpoint of external behavior of the net; hence, the entity is defined up to isomorphism.

Figure 4 gives an example of the schematic and net representations of an entity of a simple protocol. The entity has three access points. The points *Su* and *Sl* provide access to services of higher and lower levels. They are defined over the alphabets $\Delta_{Su} = \{uCon, uDis, uDat\}$ and $\Delta_{Sl} = \{lCon, lDat, lAck, lDis]$. The access point *Tm* is intended for communication with the timer entity and is defined on the alphabet $\Delta Tm = \{ON, OFF, EXP\}$.

Another example of an entity is given in Fig. 5. The entity specifies the mechanism of protocol timeout. It has one access point defined on the alphabet $\Delta_{us} = \{ON, OFF, EXP\}$, where labels correspond to switching on the timeout (*ON*), switching it off (*OFF*), and its expiration (*EXP*).

5. COMPOSITION OF PN ENTITIES

In this section, we introduce entity composition, which makes it possible to create complex structures from simpler ones. First, we present an auxiliary normalization procedure.

Definition 5.1. Suppose that $E = \langle \Sigma, \Gamma$ is an entity with access points $\alpha, \beta \in \Gamma \rangle$ such that $id_\alpha = id_\beta$. The union of these access points is a new point $\gamma = \langle id_\gamma, \Delta_\gamma, \alpha_\gamma \rangle$ such that $id_\gamma = id_\alpha = id_\beta$, $\Delta_\gamma = \Delta_\alpha \cup \Delta_\beta$, and, for any $t \in T$,

$$\sigma_\gamma = \begin{cases} \sigma_\alpha(t), & \text{if} \quad \sigma_\beta(t) = \tau; \\ \sigma_\beta(t), & \text{if} \quad \sigma_\alpha(t) = \tau; \\ \sigma_\alpha(t) + \sigma_\beta(t), & \text{otherwise.} \end{cases}$$

The union of all access points with identical names is called α-normalization of the entity and is denoted by $\alpha - norm(E)$.

In what follows, we assume that all objects are α-normalized.

Definition 5.2. Suppose that $E_1 = \langle \Sigma_1, \Gamma_1 \rangle$ and $E_2 = \langle \Sigma_2, \Gamma_2 \rangle$ are entities given in the normal form and $\alpha \in \Gamma_1$ and $\beta \in \Gamma_2$ are their access points such that $id_\alpha = id_\beta$. Then, the composition of E_1 and E_2 with respect to α

and β is the entity $E' = (E_1\ {}_\alpha\|_\beta\ E_2) = \alpha - norm(E)$, where $E = \langle \Sigma, \Gamma \rangle$ and

(1) $\Sigma = \partial_{\{\tilde\alpha, \tilde\beta\}}(\Sigma_1\ {}_\alpha|_\beta\ \Sigma_2)$;

(2) $\Gamma = \{\tilde\gamma \mid \gamma \in \Gamma_1 \cup \Gamma_1 \backslash \{\alpha, \beta\}\}$, where $\tilde\gamma = \langle id_\gamma, \tilde\lambda_\gamma \rangle$ (see Definition 3.3 of label extension).

Less formally, the structure of the resultant entity is constructed as follows.

(1) The composition of the Petri nets Σ_1 and Σ_2 with respect to the access points α and β is computed.

(2) The restriction of the resultant net is computed in the points α and β (strictly speaking, in their extensions $\tilde\alpha$ and $\tilde\beta$). It is worth noting that the transitions that do not participate in the composition, though visible in these access points, are also eliminated. This is consistent with the informal understanding of the transition label: a transition is labeled only for communication. If it is labeled but do not participate in communication, it should be eliminated.

(3) The access points that do not participate in the composition are extended to the entire resultant net and are united to form the resultant set of access points.

(4) The access points participating in the composition (α and β) are eliminated.

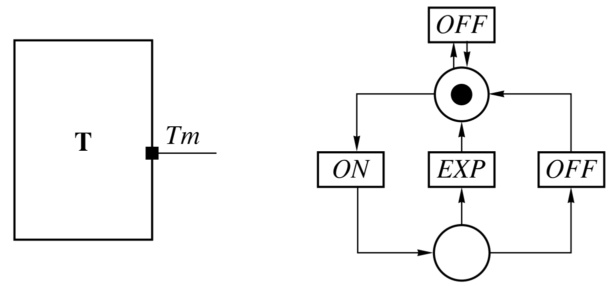
(5) The resultant entity E is α-normalized, since the union of the access points may contain two points with identical identifiers.

Figure 6 illustrates the composition of entities presented in Figs. 4 and 5. This is the composition of a simple protocol entity and the timer entity. The resulting entity has only two access points corresponding to the interface with the upper (Su) and lower (Sl) levels. As the access points involved in the synchronization are no longer required, they have been eliminated.

Now, it should be clear why it was necessary to replace the standard labeling function with a multilabel. The point is that the entity composition, where all access points are defined using labels, may yield the resultant object where several symbols correspond to one transition. Indeed, suppose that the composition of two entities $E_1[\alpha, \zeta]$ and $E_2[\beta, \xi]$ with respect to the points α and β merges transitions t_1 and t_2. Let $id_\zeta = id_\xi$ and $\sigma_\zeta(t_1) = \bar{a}$, $\sigma_\xi(t_2) = b$. In this case, when t_1 and t_2 are merged, two symbols (two elementary communication operations) $\bar{a}$ and b corresponds to the resultant transition.

6. EQUIVALENCE OF ENTITIES

The importance and practical usefulness of the notion of equivalence are commonly recognized in the theories of concurrent and distributed processes. There is a wide range of equivalence relations defined for various contexts, with some relations being stronger than others (see, for example, the survey in [20]). Most of these definitions can be expressed in terms of Petri net



(i) (ii) ***T*[*Tm*]**

Fig. 5. A timer entity.

theory [29]. In this paper, we use the notion of bisimulation equivalence studied by many authors [14, 21, 27]. First, we give the basic definition of weak bisimulation equivalence of labeled Petri nets [16].

Definition 6.1. Suppose that there are two Petri nets Σ_1 and Σ_2 and their labeling functions λ_1 and λ_2, respectively. These nets are *weakly bisimulation equivalent* with respect to the labels if there is a bisimulation relation $\Re \subseteq [M_{01}\rangle \times [M_{02}\rangle$ such that

(1) $(M_{01}, M_{02}) \in \Re$;

(2) if $(M_1, M_2) \in \Re$ and $W \in (\mathcal{M}(Vis))^*$, then,

(a) if $M_1\ (W\rangle_{\lambda_1} M_1'$, then $\exists M_2'$: $M_2\ (W\rangle_{\lambda_2} M_2'$ and $(M_1', M_2') \in \Re$;

(b) if $M_2\ (W\rangle_{\lambda_2} M_2'$, then $\exists M_1'$: $M_1\ (W\rangle_{\lambda_1} M_1'$ and $(M_1', M_2') \in \Re$.

This equivalence is written as $(\Sigma_1, \lambda_1) \approx^{\Re} (\Sigma_2, \lambda_2)$.

Using this definition as the basic one, we employ its version from [16] defined in terms of a single step, which is based on the following fact.

Proposition 6.2. *Single-step version of bisimulaton equivalence.* Suppose that there are Petri nets Σ_1 and Σ_1 with labels λ_1 and λ_2. The equality $(\Sigma_1, \lambda_1) \approx^{\Re} (\Sigma_2, \lambda_2)$ holds if and only if there is a relation $\Re \subseteq [M_{01}\rangle \times [M_{02}\rangle$ such that

(1) $(M_{01}, M_{02}) \in \Re$;

(2) if $(M_1, M_2) \in \Re$, then

(a) $M_1\ [\Theta\ M_1' \Rightarrow \exists \Phi \in (\mathcal{M}(T))^*$: $M_2\ [\Phi\rangle\ M_2'$, $(M_1', M_2') \in \Re$ and $\sigma_\alpha(\Theta) = \sigma_\beta(\Phi)$;

(b) and vice versa.

Proof is similar to that of Proposition 3.2 in [16].

To put it differently, if a step Θ with the visibility $\sigma_\alpha(\Theta) = W$ fires from the marking of one net, it is possible to fire a sequence Φ of steps with the same visibility $\sigma_\beta(\Phi) = W$ from the equivalent marking of the other net. This means that Φ can be represented as $\Rightarrow \Theta' \Rightarrow$, where $\sigma_\beta(\Theta') = W$. Thus, the executed sequence con-

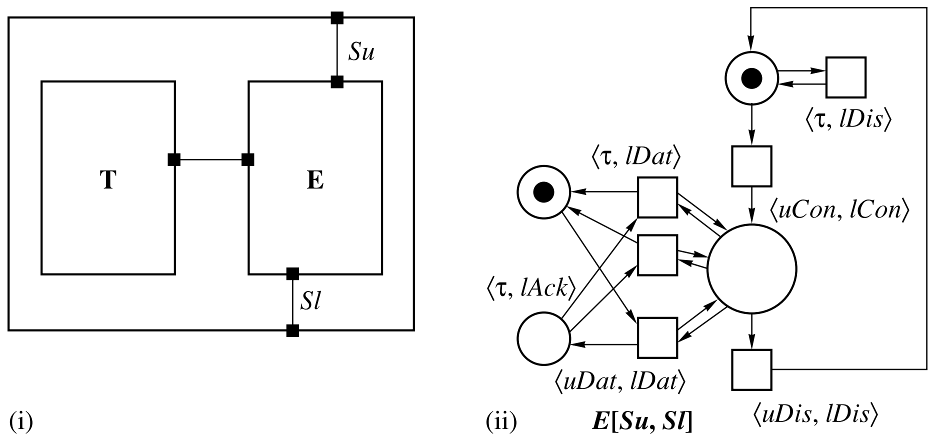


Fig. 6. An example of entity composition.

tains the only visible step Θ', and the other steps are invisible.

Based on these definitions, we introduce the equivalence of PN entities in the access points.

Definition 6.3. Suppose that there are two entities $E_1 = \langle \Sigma_1, \Gamma_1 \rangle$ and $E_2 = \langle \Sigma_2, \Gamma_2 \rangle$ with access points $\alpha = \langle id_\alpha, \lambda_\alpha \rangle \in \Gamma_1$ and $\beta = \langle id_\beta, \lambda_\beta \rangle \in \Gamma_2$. These entities are *equivalent in the access points* α and β with the relation $\Re$ if and only if $id_\alpha = id_\beta$ and $(\Sigma_1, \lambda_\alpha) \approx^{\Re} (\Sigma_2, \lambda_\beta)$. This equivalence is denoted as $E_1 \; {}_{\alpha}\overset{\Re}{\approx}_{\beta} \; E_2$.

The complete equivalence, i.e., equivalence in all access points simultaneously, is defined as follows.

Definition 6.4. Two entities $E_1 = \langle \Sigma_1, \Gamma_1 \rangle$ and $E_2 = \langle \Sigma_2, \Gamma_2 \rangle$ are *equivalent* if there is a relation $\Re$ and a one-to-one mapping $\omega: \Gamma_1 \longrightarrow \Gamma_2$ such that, for any $\alpha \in \Gamma_1$, $id_\alpha = id_{\omega(\alpha)}$ and $E_1 \; {}_{\alpha}\overset{\Re}{\approx}_{\omega(\alpha)} \; E_2$ hold. This equivalence is denoted as $E_1 \approx E_2$.

In other words, equivalent entities have the same number of the access points with the same set of names: $Id(E_1) = Id(E_2)$. Moreover, they are equivalent in all access points with identical names and the bisimulation relations $\Re$ are identical.

It can be easily shown that equivalence of entities in the access points with identical identifiers follows from the equivalence of these entities. In general, the opposite is false; i.e., equivalence does not follow from the equivalence in every point. Such equivalences may have different bisimulation relations, whereas the complete equivalence suggests the only relation.

Proposition 6.5. Suppose that there are entities $E_1 = \langle \Sigma_1, \Gamma_1 \rangle$ and $E_2 = \langle \Sigma_2, \Gamma_2 \rangle$ with access points $\alpha_1, \beta_1 \in \Gamma_1$ and $\alpha_2, \beta_2 \in \Gamma_2$ such that $id_{\alpha_1} = id_{\alpha_2}$, $id_{\beta_1} = id_{\beta_2}$. Let these entities be equivalent, $E_1 \approx E_2$, with the bisimulation relation $\Re$. If Σ_1: $M_1 \; [\Theta_1\rangle \; M_1'$, $\sigma_{\alpha_1}(\Theta_1) = W_\alpha$, $\sigma_{\beta_1}(\Theta_1) = W_\beta$, and $(M_1, M_2) \in \Re$, then the net Σ_2 contains a sequence Θ_2 such that Σ_2: $M_2 \Rightarrow \Theta_2 \Rightarrow M_2'$, $(M_1', M_2') \in \Re$, $\sigma_{\alpha_2}(\Theta_2) = \overline{W_\alpha}$, and $\sigma_{\beta_2}(\Theta_2) = \overline{W_\beta}$.

Proof. It is required to prove that, if a step Θ_1 with the visibility W_{α_1} in α_1 and W_{β_1} in β_1 fires in the first net, then the equivalent sequence in the other net contains the only visible transition Θ_2 with the same visibility in the points α_2 and β_2. Let us prove this. Suppose that, in the equivalent sequence of the second net, the steps visible in α_2 and β_2 are not the same. Without loss of generality, assume that

$$M_2 \Rightarrow M_2^1 \; [\Theta_2'\rangle M_2^2 \Rightarrow M_2^3[\Theta_2''\rangle M_2^4 \Rightarrow M_2',$$

where $\sigma_{\alpha_2}(\Theta_2') = W_\alpha$, $\sigma_{\beta_2}(\Theta_2'') = W_\beta$, $\sigma_{\alpha_2}(\Theta_2'') = \tau$, and $\sigma_{\beta_2}(\Theta_2') = \tau$. However, if there is only one bisimulation relation $\Re$, this is impossible, since, e.g., the marking M_2^2 has no equivalent marking in the first net. Indeed, only two cases are possible: $(M_1, M_2^2) \in \Re$ and $(M_1', M_2^2) \in \Re$. However, $(M_1, M_2^2) \notin \Re$, since it is visible in β_1 that W_β has already fired, whereas in β_2 this is not visible yet. Similarly, $(M_1, M_2^2) \notin \Re$.

It has been known that one of the basic advantages of the bisimulation equivalence is that it possesses good algebraic properties. It turns out that the equivalence of entities inherits these properties.

Theorem 6.6. *Congruence of entity equivalence.* Equivalence of PN entities is congruence with respect to the composition operation; i.e., if $E_2 \approx E_3$, $\beta \in \Gamma_2$, $\gamma \in \Gamma_3$, and $id_\beta = id_\gamma$, then

$$(E_1 \; {}_{\alpha}\|_{\beta} \; E_2) \approx (E_1 \; {}_{\alpha}\|_{\gamma} \; E_3). \tag{1}$$

Proof. Let $E_1 = \langle \Sigma_1, \Gamma_1 \rangle$, $E_2 = \langle \Sigma_2, \Gamma_2 \rangle$, and $E_3 = \langle \Sigma_3, \Gamma_3 \rangle$. Denote $F = (E_1 \; {}_{\alpha}\|_{\beta} \; E_2)$ and $G = (E_1 \; {}_{\alpha}\|_{\gamma} \; E_3)$. Let $E_2 \approx$

E_3 with the relation $\Re_1$. To prove Eq. (1), it is sufficient to show that

$$\Re = \{(M_1 + M_2, M_1 + M_3) \mid (M_2, M_3) \in \Re_1, \; M_1 + M_2 \in [M_{F0}\rangle, M_1 + M_3 \in [M_{G0}\rangle\}. \quad (2)$$

This follows from Propositions 6.2 and 3.4. Indeed, suppose that $(M_1 + M_2, M_1 + M_3) \in \Re$ and, for some synchronization transition $t \in T_1 \otimes T_2$, the following equation holds:

$$\Sigma_F\colon M_1 + M_2 \; [t\rangle \; M_1' + M_2' \quad (3)$$

with $\sigma_\phi(\Theta) = W$ for some $\phi \in \Gamma_2$. Let us show that there is a sequence Φ of steps such that Σ_G: $M_1 + M_3\,[\Phi\rangle\, M_1' + M_3'$, $\sigma_\phi(\Phi) = W$. From the step (3) and Proposition 6.2(1), it follows that the step $M_2\,[\nu_2(\Theta)\rangle\, M_2'$ fires in the net Σ_2. From the equivalence $E_2 \approx E_3$, it follows that Σ_3 contains the sequence $\Phi' = \Rightarrow\Theta'\Rightarrow$ with $\sigma_\phi(\nu_2(\Theta')) = W$ and $\sigma_\beta(\Theta') = \overline{W}$. Take M_3 such that $(M_1, M_3) \in \Re_1$. In line with Proposition 6.2, Σ_3: $M_3 \Rightarrow [\Theta\rangle \Rightarrow M_3'$, $\sigma_\gamma(\Theta_3) = \sigma_\beta(\Theta_3)$, and $\sigma_\phi(\Theta_3) = W$. Proposition 6.2(1) guarantees that the net G contains $t' = \Theta + \Theta_3$ with $\sigma_\phi(t') = W$. This means that

$$\Sigma_G\colon M_1 + M_3 \; [t'\rangle \; M_1' + M_3' \quad (4)$$

with $\sigma_\phi(t') = W$.

For transitions that are not contained in the set of the synchronization transitions, the proof is obvious. This result can be easily extended to the transition step.

This result makes it possible to apply the modular approach to the development and verification of complex communicating systems. In particular, one can replace some unit of the system with an equivalent one without changing the overall behavior of the system.

Note that the entity equivalence based on the step semantics is the weakest equivalence; it is congruent with respect to the composition operation. It is easy to verify that the equivalence based on the interleaving semantics of Petri nets is not congruent.

7. PROPERTIES OF ENTITY COMPOSITION

In order to use the entity composition in practice, this operation must be commutative and associative.

Proposition 7.1. *Commutativity of entity composition.* Suppose that there are entities E and F and their access points $\alpha \in \Gamma_E$ and $\beta \in \Gamma_F$. Then $(E\; {}_\alpha\|_\beta\; F) = (F_\beta\|_\alpha\, E)$.

Proof immediately follows from commutativity of the union operation (for sets and multisets) and Definition 5.2.

In addition, it turns out that the composition operation is associative.

Theorem 7.2. *Associativity of entity composition.* Suppose that E_1, E_2, and E_3 are entities; $\alpha \in \Gamma_1$, $\beta, \zeta \in \Gamma_2$, $id_\alpha \notin Id(E_3)$, and $id_\xi \notin Id(E_1)$. Then,

$$((E_1\; {}_\alpha\|_\beta\; E_2)\; {}_\zeta\|_\xi\; E_3) = (E_1\; {}_\alpha\|_\beta\; (E_2\; {}_\zeta\|_\xi\; E_3)).$$

Proof. Let $F = ((E_1\; {}_\alpha\|_\beta\; E_2)\; {}_\zeta\|_\xi\; E_3)$ and $G = (E_1\; {}_\alpha\|_\beta\; (E_2\; {}_\zeta\|_\xi\; E_3))$. Obviously, it is sufficient to verify that the sets of synchronization transitions of the nets Σ_F and Σ_G are equal, i.e., $(T_1 \otimes T_2) \otimes T_3 = T_1 \otimes (T_2 \otimes T_3)$. For this purpose, we employ the idea used in the proof of Theorem 3.2. Let us show that both sets are equal to $T_1 \otimes T_2 \otimes T_3$, which is defined as follows. An auxiliary net $\tilde{\Sigma}_{123} = \langle S_{12} \cup S_{23}, T_1 \cup T_2 \cup T_3, {}^\bullet(), ()^\bullet, M_0\rangle$ is constructed, where $S_{12} = \Delta_{12} = \Delta_\alpha \cup \Delta_\beta$, $S_{23} = \Delta_\zeta \cup \Delta_\xi$, $M_0 = 0$,

$${}^\bullet(t) = \begin{cases} \overline{\sigma_\alpha^+(t)} \restriction \Delta_{12}, & \text{if } t \in T_1; \\ \overline{\sigma_\beta^+(t)} \restriction \Delta_{12} + \sigma_\zeta^+(t) \restriction \Delta_{23}, & \text{if } t \in T_2; \\ \overline{\sigma_\xi^+(t)} \restriction \Delta_{23}, & \text{if } t \in T_3; \end{cases}$$

$$(t)^\bullet = \begin{cases} \sigma_\alpha^+(t) \restriction \Delta_{12}, & \text{if } t \in T_1; \\ \sigma_\beta^+(t) \restriction \Delta_{12} + \sigma_\zeta^+(t) \restriction \Delta_{23}, & \text{if } t \in T_2; \\ \sigma_\xi^+(t) \restriction \Delta_{23}, & \text{if } t \in T_3. \end{cases}$$

The set of minimal T-invariants for $\tilde{\Sigma}_{123}$ is $T_1 \otimes T_2 \otimes T_3$. Consider how the set $(T_1 \otimes T_2) \otimes T_3$ is computed. First, the net $\tilde{\Sigma}_{12}$ is used to construct the set $T_1 \otimes T_2 = \{x_1, \ldots, x_p,\}$, where $x_i = \sum_{t \in T_1} n_t t + \sum_{t \in T_2} n_t t$ are minimal. Next, the net $\tilde{\Sigma}_{12,3}$ is constructed from the transitions $T_1 \otimes T_2$ and T_3 assuming that

$$(x_i)^\bullet = \sigma_\zeta\left(\sum_{t \in T_1} n_t t + \sum_{t \in T_2} n_t t\right)$$

$$= \sigma_\zeta\left(\sum_{t \in T_2} n_t t\right) = \sum_{t \in T_2} n_t \sigma_\zeta(t).$$

It is well known that algorithms of searching for T-invariants (e.g. [2]) consist in stepwise transformation of the incidence matrix by means of linear composition of other rows in order to fill columns with zeros. Every such step can be interpreted as a reduction of the Petri net that isolates the place corresponding to the column being filled with zeros (see the reduction rule R1 in [13]). Let the algorithm for finding T-invariants in the net $\tilde{\Sigma}_{123}$ first zeroes the columns that correspond to the places S_{12}. Then, rows of the resultant incidence matrix correspond to transitions from $T_1 \otimes T_2$ and T_3. Note that this matrix corresponds to the net $\tilde{\Sigma}_{123}$. Hence, further operation of the algorithm will lead to the set of invariants that simultaneously correspond to the nets $\tilde{\Sigma}_{123}$ and $\tilde{\Sigma}_{12,3}$. Similarly, it can be shown that $T_1 \otimes (T_2 \otimes T_3) = T_1 \otimes T_2 \otimes T_3$. Thus, the equality $T_1 \otimes (T_2 \otimes T_3) = (T_1 \otimes T_2) \otimes T_3$ holds.

This result allows us to write $(E_1 \ {}_{\alpha}\|_{\beta} \ E_2 \ {}_{\zeta}\|_{\xi} \ E_3)$ and to compose the nets without taking care of the order of the operations.

8. CONCLUSION

In this paper, the notions of the Petri net entity and the entity composition have been introduced. The Petri net entity is an extension to the labeled Petri net; instead of a single label, it uses several labels called access points of the entity. This extension allows mapping of a single physical event (transition firing) to several logical events related to communication with other objects. In its turn, this has required the extension of the notion of the label to the multilabel, which maps a single transition to a multiset of symbols.

The composition operation makes it possible to support the compositional style of system development and verification. Indeed, it makes possible to develop individual parts of the system independently and, then, to compose them. Moreover, since the composition is congruent, it is possible to replace subsystems by equivalent ones without changing the overall behavior of the system. This can be helpful when using the stepwise refinement for the system design. In papers [1, 8], it is shown how this principle can be used for hierarchical composition of protocols.

We believe that the possibility to schematically represent the structure of Petri net entities is of practical significance. Indeed, the top-down development implies that the developer first represents the system schematically as a structure of entities, which is consistent with the commonly accepted paradigm of design in terms of units and connections. Units can be nested at any structural level. At the next stage, the internal structure of the blocks from the deepest level is determined in terms of Petri nets. Note that this design is formal at all stages. In papers [8–10], some results on the application of this approach to the development of actual concurrent and distributed systems are presented.

Let us outline the problems that should be solved in order to make possible the application of the compositional Petri nets to solving practical problems.

(a) A more elaborated set of operations over Petri nets is necessary for the construction of internal structure of entities. This set may contain sequential compositions, iterations, selections, and the like. In this respect, it may be reasonable to extend the notion of the access point to Petri net places.

(b) To further improve practical characteristics of the formalism, it should be extended to high-level Petri nets; its relations to concurrent programming and specification languages must be established; and appropriate automation tools are to be developed.

ACKNOWLEDGMENTS

This work was supported by the Ministry of Science of the Russian Federation, project no. 0201.01.225.

REFERENCES

1. Anisimov, N.A., Hierarchical Composition of Protocols, *Avtom. Vychisl. Tekh.*, 1990, no. 1, pp. 3–10.
2. Achasova, S.M. and Bandman, O.L., *Korrektnost' parallel'nykh vychislitel'nykh protsessov* (Correctness of Parallel Computational Processes), Novosibirsk: Nauka, 1990.
3. Kotov, V.E., *Seti Petri* (Petri Nets), Moscow: Nauka, 1984.
4. *Protokoly informatsionno-vychislitel'nykh setei. Spravochnik* (Protocols of Information Computational Networks. Handbook), Mizina, I.A. and Kuleshova, A.P., Eds., Moscow: Radio i Svyaz', 1990.
5. Hoare, C., *Communicating Sequential Processes*, Englewood Cliffs: Prentice-Hall, 1985. Translated under the title *Vzaimodeistvuyushchie posledovatel'nye protsessy*, Moscow: Mir, 1989.
6. Alaiwan, H. and Toudic, J.M., Recherche de Semiflots, des Verrous et Trappes dans les Reseaux de Petri, *Technique Et Sciences Informatiques*, 1985, vol. 4, pp. 103–112.
7. Anisimov, N.A., A Petri Net Entity as a Formal Model for LOTOS, a Specification Language for Distributes and Concurrent Systems, *Parallel Computing Technologies*, Mirenkov, N.N., Ed., Singapore: World Sci., 1991, pp. 440–450.
8. Anisimov, N.A. and Koutny, M., On Compositionality and Petri Nets in Protocol Engineering, *Protocol Specification, Testing and Verification, XV*, Dembinski, P. and Sredniawa, M., Eds., Chapman & Hall, 1996, pp. 71–86.
9. Anisimov, N.A., Kovalenko, A.A., Tarasov, G.V., Inzartsev, A.V., and Sherbatyuk, A.Ph., A Graphical Environment for AUV Mission Programming and Verification, *Proc. of the 10th Int. Symp. on Unmanned Untethered Submersible Technology*, New Hermpshire, USA, 1997, pp. 394–405.
10. Anisimov, N.A., Kovalenko, A.A., Postupalski, P.A., and Vuong, S.T., Application of Compositional Petri Nets and PN^3—Tool to the Specification of Distributed Multimedia Objects, *Advances in Distributed Multimedia Systems*, Chang, S.K. *et al.*, Eds., Singapore: World Sci., 1999, pp. 99–116.
11. Barbeau, M. and Bochmann, G.V., A Subset of Lotos with Computational Power of Place/Transition Nets, *Lecture Notes in Computer Science*, 1993, vol. 691, pp. 49–68.
12. Baumgarten, B., Ochsenschläger, P., and Prinoth, R., Building Blocks for Distributed System Design, *Protocol Specification, Testing, and Verification*, Diaz, M., Ed. (Proc. of the V IFIP WG 6.1 Conf.), North-Holland: Elsevier, 1986, pp. 19–38.
13. Berthelot, G., Roucairol, G., and Valk, R., Reduction of Nets and Parallel Programs, *Lecture Notes in Computer Science*, 1980, vol. 84, pp. 277–290.

14. Bergstra, J.A. and Klop, J.W., Algebra of Communicating Processes, in *Math. on Comput. Sci. (Proc. CWI Symp.)*, 1986, pp. 89–138.
15. Bernadinello, L. and De Cindio, F., A Survey of Basic Net Models and Modular Classes, *Lecture Notes in Computer Sciences*, Springer, 1992, vol. 609, pp. 304–351.
16. Best, E., Devillers, R., Kiehn, A., and Pomello, L., Concurrent Bisimulations in Petri Nets, *Acta Informatica*, 1991, vol. 28, pp. 231–261.
17. Best, E., Devillers, R., and Hall, J.G., The Box Calculus: A New Causal Algebra with Miltilabel Communication, *Lecture Notes in Computer Sciences*, Springer, 1992, vol. 609, pp. 21–69.
18. Bolognesi, T. and Brinksma, E., Introduction to the ISO Specification Language LOTOS, *Comput. Networks ISDN Systems*, 1987, vol. 14, pp. 25–89.
19. Cherkasova, L.A. and Kotov, V.E., Structured Nets, *Lecture Notes in Computer Sciences*, Springer, 1981, vol. 118, pp. 242–251.
20. Van Glabbeek, R.J., The Linear Time—Branching Time Spectrum, *Lecture Notes in Computer Sciences*, Berlin: Springer, 1990, vol. 458, pp. 278–297.
21. Van Glabbeek, R.J. and Vaandrager, F.W., Petri Net Models for Algebraic Theories of Concurrency, *Lecture Notes in Computer Sciences*, Springer, 1987, vol. 259, pp. 224–242.
22. Goltz, U., On Representing CCS Programs by Finite Petri Nets, *Lecture Notes in Computer Sciences*, Springer, 1988, vol. 324, pp. 339–350.
23. Hopkins, R., Hall, J., and Botti, O., A Basic-Net Algebra for Programs Semantics and Application to OCCAM, *Lecture Notes in Computer Scences*, Springer, 1992, vol. 609, pp. 179–214.
24. *High-Level Petri Nets. Theory and Application*, Jensen, K. and Rozenberg, G., Eds., Springer, 1991.
25. Kotov, V.E., An Algebra for Parallelism Based on Petri Nets, *Lecture Notes in Computer Sciences*, Springer, 1978, vol. 64, pp. 39–55.
26. Mauw, S. and Veltink, G.J., *A Process Specification Formalism, Fundamenta Informaticae XIII*, 1990, pp. 85–139.
27. Milner, R., A Calculus for Communication Systems, *Lecture Notes in Computer Sceinces*, Springer, 1980, vol. 92.
28. Olderog, E.-R., Operational Petri Net Semantics for CCSP, *Lecture Notes in Computer Sciences*, Springer, 1984, vol. 266.
29. Pomello, L., Rozenberg, G., and Simone, C., A Survey of Equivalence Notions for Net Based Systems, *Lecture Notes in Computer Sciences*, Springer, 1992, vol. 609.
30. Lecture Notes on Petri Nets. Parts I and II, *Lecture Notes in Computer Sciences*, Reising, W. and Rozenberg, G., Eds., Springer, 1998, vols. 1491–1492.
31. Advances in Petri Nets 1992, *Lecture Notes in Computer Sciences*, Rozenberg, G., Ed., Springer, 1992, vol. 609.
32. Taubner, D., Finite Representation of CCS and TCSP Programs by Automata and Petri Nets, *Lecture Notes in Computer Sciences*, Springer, 1989, vol. 369.
33. Valette, R., Analysis of Petri Nets by Stepwise Refinements. *J. Comput. System Sci.*, 1979, vol. 18, pp. 35–46.
34. Vogler, W., Failures Semantics and Deadlocking of Modular Petri Nets, *Acta Informatica*, 1989, vol. 26, pp. 333–348.
35. Voss, K., System Specification with Labelled Nets and the Notion of Interface Equivalence, *Arbeitspapiere Der GMD*, 1986, vol. 221.