Front. Electr. Electron. Eng. China 2008, 3(3): 290–294
DOI 10.1007/s11460-008-0050-2

RESEARCH ARTICLE

Zhenhua YU, Yuanli CAI, Haiping XU

Petri nets semantics of π-calculus

Abstract As π-calculus based on the interleaving semantics cannot depict the true concurrency and has few supporting tools, it is translated into Petri nets. π-calculus is divided into basic elements, sequence, concurrency, choice and recursive modules. These modules are translated into Petri nets to construct a complicated system. Petri nets semantics for π-calculus visualize system structure as well as system behaviors. The structural analysis techniques allow direct qualitative analysis of the system properties on the structure of the nets. Finally, Petri nets semantics for π-calculus are illustrated by applying them to mobile telephone systems.

Keywords Petri nets, π-calculus, concurrency, structual characteristics, analysis

1 Introduction

Petri nets and π-calculus are promising mathematical modeling tools for describing, analyzing and verifying concurrent systems [1]. π-calculus [2] is employed to model concurrent systems with dynamic topology, and supports formal analysis of systems in a variety of well-established techniques. However, the processes of π-calculus are complicated, and they cannot visually model the system architecture or depict the true concurrency. Moreover, π-calculus has few supporting tools, such as MWB and HAL. While Petri nets are a graphical and mathematical modeling tool, which are suitable for describing concurrent, distributed and asynchronous systems [3]. Petri nets put emphasis on modeling system structure and analyzing system properties, and they can effectively depict the true concurrency. Besides, there are many tools available for simulating, analyzing and verifying Petri nets model (http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/).

To remedy the deficiencies of π-calculus, π-calculus is translated into Petri nets. Consequently, the structural analysis techniques and supporting tools for Petri nets can be adopted to analyze and verify the concurrent systems with dynamic topology. In recent years, there is work aiming at translating π-calculus into Petri nets [4–7]. However, the methods present some especial Petri nets that cannot use existing supporting tools of Petri nets. Furthermore, most methods are too complicated to efficiently describe systems.

In this paper, π-calculus is divided into basic elements, recursive, sequence, concurrency and choice modules. These modules are translated into Petri nets, and then construct a complicated system.

2 Petri nets semantics for π-calculus

The Petri nets model of the process P is called N_P in which colored tokens, arcs with arc expression function, and transitions with guard functions are employed. Channels in π-calculus are divided into restricted channels and unrestricted channels. The restricted channels are only used in the interior of the process. According to the work in Ref. [7], the transitions and arcs associated with the restricted channels are labeled and cannot interact with the other Petri nets models. Places are labeled by their status symbols (entry places by e, internal places by i, and exit places by x) [7]. The preset of e is empty and the post-set of x is empty. Actions in π-calculus correspond to transitions in Petri nets. Transitions have two different kinds of labels: ordinary transitions and communication transitions τ. Allelomorph names are mapping into transitions τ.

To describe the characteristics of dynamic actions in π-calculus, the trace is introduced from the communicating sequential processes (CSP) [8].

Translated from *Control and Decision*, 2007, 22(8): 864–868 [译自: 控制与决策]

Zhenhua YU (✉), Yuanli CAI
School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China
E-mail: zhenhua_yu@163.com

Zhenhua YU
The 205th Institute of China Ordnance Industry Corporation Group, Xi'an 710065, China

Haiping XU
Department of Computer and Information Science, University of Massachusetts Dartmouth, North Dartmouth, MA 02747, USA

Definition 1 A set of actions in P executed in turn is called a trace of P, denoted as traces(P) = ⟨action1, action2, … ⟩. The set of all traces is denoted as traces(P). The null trace ⟨⟩ belongs to traces(P).

Definition 2 The connection of two traces s and t is the action connection, denoted as s^t.

Traces of π-calculus are similar to the firing sequence σ of Petri nets, which depict the dynamic action characteristics. The prefix expressions $\pi \in \{\tau, y(x), \bar{y}x, \bar{y}(x)\}$ of π-calculus are regarded as basic elements, and the process are divided into recursive, sequence, concurrency and choice modules. Petri nets semantics of these modules are discussed in detail as follows.

2.1 Petri nets semantics for basic elements in π-calculus

The basic elements $\pi \in \{\tau, y(x), \bar{y}x, \bar{y}(x)\}$ describe processes actions. Processes are composed of sequential, concurrent, choice, and recursive composition of the basic elements.

Rule 1 Petri nets semantic N_π for $\pi.0$

The action π is represented by the transition t_π, which is added with an input place and an output place, as shown in Fig. 1(a).

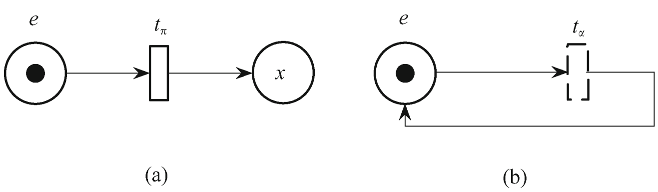

Fig. 1 Petri nets semantics for basic element and recursion module. (a) N_π; (b) $N_{\alpha.P}$

According to Fig. 1(a), traces(π.0) = {⟨⟩,⟨π⟩}, $\sigma s(N_\pi)$ = {⟨⟩,⟨t_π⟩}. Although π and t_π are different symbols, they represent the same action. Therefore, the following conclusion is deduced.

Conclusion 1 If the Petri nets model N_π is deduced from the π-calculus process $\pi.0$ by Rule 1, then traces(π.0) = $\sigma s(N_\pi)$.

2.2 Petri nets semantic for recursive module in π-calculus

The recursion in π-calculus is similar to that of CCS [2]. $P =_{\text{def}} \alpha.P$ denotes that the action α is infinitely executed.

Rule 2 Petri nets semantic $N_{\alpha.P}$ for $P =_{\text{def}} \alpha.P$

Assume that the process P is translated into N_P. The rule for $P =_{\text{def}} \alpha.P$ translated into $N_{\alpha.P}$ is as follows: the action α is represented by the transition t_α, the output place and input arc of N_P are deleted, and an arc between the transition t_α and the input place is added. The Petri nets semantic of the recursion in π-calculus is shown in Fig. 1(b).

In this paper, dotted places and transitions represent abstract elements, which can be refined into internal implementations; the process P and Q have been translated into Petri nets models N_P and N_Q, and traces(P) = σs(N_P), traces(Q) = $\sigma s(N_Q)$.

The trace set of $P =_{\text{def}} \alpha.P$ is traces([$a = b$]P) = {⟨⟩, ⟨α⟩^{n}}, which executes the action α for n times. According to Rule 2 and Fig. 1(b), the firing sequence of $N_{\alpha.P}$ is σs ($N_{\alpha.P}$) = {⟨⟩,⟨t_α⟩^{n}}. Although α and t_α are different symbols, they represent the same action. Therefore the following conclusion is deduced.

Conclusion 2 If the Petri nets model $N_{\alpha.P}$ is deduced from the π-calculus process $P =_{\text{def}} \alpha.P$ by the mapping Rule 2, then traces(π.0) = σs (N_π).

The replication !P in π-calculus is given by the definition !P = P|!P, which represents an unbounded number of copies of P. The Petri nets semantic of !P is given by Rule 2.

2.3 Petri nets semantics for sequence module in π-calculus

Rule 3 Petri nets semantic $N_{\pi.P}$ for $\pi.P$

The process $\pi.P$ indicates that P is executed after π is executed. The rule for $\pi.P$ translated into $N_{\pi.P}$ is as follows: the action π is represented by the transition t_π, the entry places e containing a token is added for t_π, N_P is regarded as an abstract place, and then t_π is regarded as the pre-transition of N_P. The Petri nets semantic of $\pi.P$ is shown in Fig. 2(a).

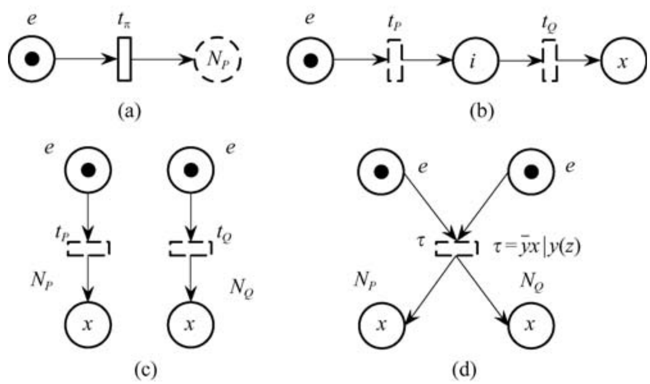

Fig. 2 Petri nets semantics for sequence and concurrency modules. (a) $N_{\pi.P}$; (b) $N_{P.Q}$; (c) $N_{P|Q}$;(d) $N_{P|Q}$

The trace set of $\pi.P$ is traces($\pi.P$) = {⟨⟩,⟨π⟩^s∧s ∈ traces(P)}. According to Rule 3 and Fig. 2(a), the firing sequence of $N_{\pi.P}$ is $\sigma s(N_{\pi.P})$ = {⟨⟩, ⟨t_π⟩^$\sigma s(N_P)$}. Although π and t_π are different symbols, they represent the same action. Therefore, the following conclusion is deduced.

Conclusion 3 If the Petri nets model $N_{\pi.P}$ is deduced from the π-calculus process $\pi.P$ by the mapping Rule 3, then traces($\pi.P$) = $\sigma s(N_{\pi.P})$.

Rule 4 Petri nets semantic $N_{P.Q}$ for $P.Q$

The process $P.Q$ represents the sequence structure, where Q is executed after P. The rule for $P.Q$ translated into $N_{P.Q}$ is as follows. The token in the input place of N_Q is deleted, the input place of N_Q and output place of N_P are combined as the input place of N_Q, and the input place of N_P and output place of N_Q are regarded as the input and output place of $N_{P.Q}$, respectively. The Petri nets semantic of $P.Q$ is shown in Fig. 2(b).

If the process P successfully terminates after s_1 is executed, then the process Q can execute, and traces($P.Q$) $= \{s_1 \hat{} s_2 \wedge s_1 \in \text{traces}(P) \wedge s_2 \in \text{traces}(Q)\}$. According to the mapping Rule 4 and Fig. 2(b), when P terminates, the token in P is added into the input place of N_Q, therefore, $\sigma s(N_{P.Q}) = \{\sigma_1 \hat{} \sigma_2 \wedge \sigma_1 \in \sigma s(N_P) \wedge \sigma_2 \in \sigma s\ (N_Q)\}$.

If the process P is a deadlock, then traces($P.Q$) $= \{\text{traces}(P)\}$, and N_P is a deadlock; thereby the output place of N_P, namely, the input place of N_Q, will not be marked, then the transitions of N_Q will not be enabled. Therefore, $\sigma s(N_{P.Q}) = \{\sigma s(N_P)\}$. The following conclusion is deduced.

Conclusion 4 If the Petri nets model $N_{P.Q}$ is deduced from the π-calculus process $P.Q$ by the mapping Rule 4, then traces($P.Q$) $= \sigma s(N_{P.Q})$.

2.4 Petri nets semantic for concurrency module in π-calculus

The symbol "|" in π-calculus represents concurrency.

Rule 5 Petri nets semantic $N_{P|Q}$ for $P|Q$

The process $P|Q$ has two different types.

1) If P and Q are independent, the Petri nets semantic $N_{P|Q}$ of $P|Q$ is shown in Fig. 2(c), where N_P and N_Q are concurrent.

2) If P communicates with Q by allelomorph name, such as $P = \bar{y}x$ or $P = \bar{y}(x)$, $Q = y(z)$, then P and Q are synchronous processes and allelomorph name $\bar{y}x|y(z)$ is handshake protocol [9].

Allelomorph name is mapped into the communication transition τ, which is regarded as an internal action or a silent action and is only used in P and Q. The Petri nets semantic $N_{P|Q}$ of $P|Q$ is shown in Fig. 2(d).

The trace set of $P|Q$ is traces($P|Q$) $= \{s_1 \in \text{traces}(P) \wedge s_2 \in \text{traces}(Q)\}$. According to the mapping Rule 5, in the first type, the firing sequence of $N_{P|Q}$ is $\sigma s(N_{P|Q}) = \{\sigma_1 \in \sigma s(N_P) \wedge \sigma_2 \in \sigma s(N_Q)\}$. In the second type, although allelomorph names are mapped into the communication transition τ, the input and output actions are executed and the firing sequence is the same as that in the first type. Therefore, the following conclusion is deduced.

Conclusion 5 If the Petri nets model $N_{P|Q}$ is deduced from the π-calculus process $P|Q$ by the mapping Rule 5, then traces($P|Q$) $= \sigma s(N_{P|Q})$.

2.5 Petri nets semantics for choice module in π-calculus

The symbol "+" in π-calculus represents the choice; the match $[a = b]P$ is also regarded as a special choice. $[a = b]P$ behaves like P if the names a and b are identical, otherwise, it behaves like 0 [10].

Rule 6 Petri nets semantic N_{P+Q} for $P+Q$

The rule for $P+$Q translated into N_{P+Q} is as follows. The input places of N_P and N_Q are combined as a common input place. The Petri nets semantic of $P+Q$ is shown in Fig. 3(a).

The trace set of $P+Q$ is traces($P+Q$) $= \{s|s \in \text{traces}(P) \vee s \in \text{traces}(Q)\}$. According to the mapping Rule 6 and Fig. 3(a), the firing sequence of N_{P+Q} is $\sigma s(N_{P+Q}) = \{\sigma|\sigma \in \sigma s(N_P) \vee \sigma \in \sigma s(N_Q)\}$. Therefore, the following conclusion is deduced.

Conclusion 6 If the Petri nets model N_{P+Q} is deduced from the π-calculus process $P+Q$ by the mapping Rule 6, then traces($P+Q$) $= \sigma s(N_{P+Q})$.

Rule 7 Petri nets semantic $N_{[a=b]P}$ for $[a = b]P$

The rule for $[a = b]P$ translated into $N_{[a=b]P}$ is as follows: the auxiliary transitions t_1 and t_2 are added as the prefix transitions of the input and output places of N_P, which judge the match of names. The input place with a token is added for t_1 and t_2. The Petri nets semantic of $[a = b]P$ is shown in Fig. 3(b).

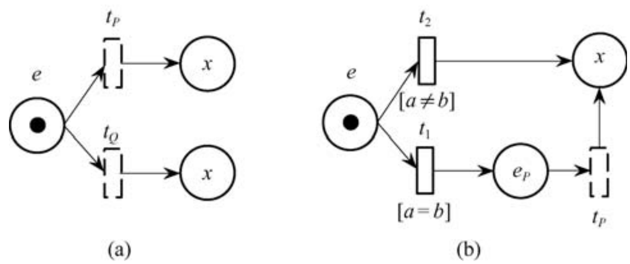

Fig. 3 Petri nets semantics for choice modules. (a) N_{P+Q}; (b) $N_{[a=b]P}$

The trace set of $[a = b]P$ is traces($[a = b]P$) $= \{\langle\rangle, \langle[a \neq b]\rangle, \langle[a = b]\rangle \hat{} s \wedge s \in \text{traces}(P)\}$. According to the mapping Rule 7 and Fig. 3(b), the firing sequence of $N_{[a=b]P}$ is $\sigma s(N_{[a=b]P}) = \{\langle\rangle, \langle t_2\rangle, \langle t_1\rangle \hat{} \sigma \wedge \sigma \in \sigma s(N_P)\}$. Although $[a = b]$ and t_1 are different symbols, they represent the same action. $[a \neq b]$ and t_2 also represent the same action. Therefore, the following conclusion is deduced.

Conclusion 7 If the Petri nets model $N_{[a=b]P}$ is deduced from the π-calculus process $[a = b]P$ by the mapping Rule 7, then traces($[a = b]P$) $= \sigma s(N_{[a=b]P})$.

According to the rules, Petri nets semantics of π-calculus are set up.

2.6 Effectiveness for Petri nets semantics

Petri nets semantics of π-calculus should not change functional characteristics of systems. Two criteria called concurrency and functional equivalence are used to judge the effectiveness of Petri nets semantics [11].

1) Concurrency: Petri nets semantics of π-calculus should represent the intended concurrency of processes. π-calculus is based on the interleaving semantics, where concurrency is reduced to the non-deterministic choice. π-calculus cannot distinguish the processes $P = a|b$ and $Q = a.b + b.a$. The Petri nets semantics of P and Q are shown in Fig. 4. As Petri nets are non interleaving models, it is possible to distinguish between concurrency (Fig. 4(a)) and non-deterministic interleaving (Fig. 4(b)).

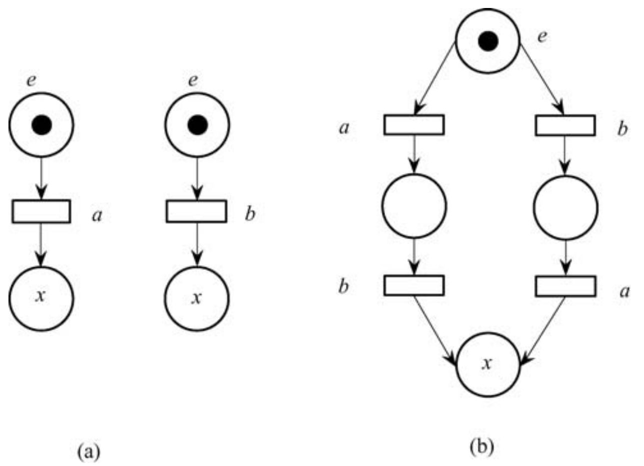

Fig. 4 Concurrency and non-deterministic choice. (a) Concurrency; (b) non-deterministic choice

2) Functional equivalence: in π-calculus, the set of status is regarded as the nodes of graph, and the arrows of graph correspond to the actions. The graph can be called transition systems (TS), which is similar to reachability graph (RG) in Petri nets. The strong bisimulation relation between TS and RG is defined as follows.

Definition 3 A binary relation $S \subseteq \mathrm{TS} \times \mathrm{RG}$ is a strong bisimulation if $(P, M) \in S$ implies, for actions α and transition $t \in T$,

a) If $P \xrightarrow{\alpha} P'$, then $\exists M'$, $M[t\rangle M'$ and $(P', M') \in S$.

b) If $M[t\rangle M'$, then $\exists P'$, $P \xrightarrow{\alpha} P'$ and $(P', M') \in S$.

The actions in π-calculus must correspond to the transitions of Petri nets. According to Conclusions 1–7, the trace t of P corresponds to the firing sequence σ of N_p, and the actions and their order in t are the same as those in σ. Therefore, the relation between the transition systems of P and the reachability graph of N_P is a strong bisimulation.

Theorem 1 The process P in π-calculus and its Petri nets semantic N_p are equivalent.

3 Case study: mobile telephone system

A mobile telephone system is used to illustrate Petri nets semantics of π-calculus [12]. The system consists of two cars with two mobile telephones, two stations, and a central controller. Each station is located in a different part of the country. A car moves about the country, and it should always be in contact with a station. If a car is far from its current station, then it should switch to another station. Assume $a = \{\text{talk, switch, gain, lose}\}$, the station, controller, and car may be written in π-calculus as follows:

$$\begin{aligned}
&\mathrm{Station}_i(\alpha_i) =_{\mathrm{def}} \mathrm{talk}_i.\mathrm{Station}_i\langle\alpha_i\rangle + \mathrm{lose}_i(t_i,s_i).\\
&\qquad \mathrm{inform}_i(t_j,s_j).\overline{\mathrm{switch}_i}\langle t_j,s_j\rangle.\mathrm{Station}_j(\alpha_j),\\
&\mathrm{Control}_i(\mathrm{lose}_i,\mathrm{inform}_i) =_{\mathrm{def}} \overline{\mathrm{lose}_i}\langle \mathrm{talk}_i,\mathrm{switch}_i\rangle.\\
&\qquad \overline{\mathrm{inform}_i}\langle \mathrm{talk}_j,\mathrm{switch}_j\rangle.\\
&\qquad \mathrm{Control}_j(\mathrm{lose}_j,\mathrm{inform}_j),\\
&\mathrm{Car}_i(\mathrm{talk}_i,\mathrm{switch}_i) =_{\mathrm{def}} \overline{\mathrm{talk}_i}.\mathrm{Car}_i\langle \mathrm{talk}_i,\mathrm{switch}_i\rangle\\
&\qquad + \mathrm{switch}_i(t_i,s_i).\mathrm{Car}_i\langle t_i,s_i\rangle.
\end{aligned} \tag{1}$$

The mobile telephone system is described as follows:

$$\begin{aligned}
\mathrm{System} = &(\nu\alpha_i : i = 1,2)(\mathrm{Car}_1(\mathrm{talk}_1,\mathrm{switch}_1)|\mathrm{Station}_1(\alpha_1)|\\
&\mathrm{Control}_1(\mathrm{lose}_1,\mathrm{inform}_1)|\mathrm{Car}_2(\mathrm{talk}_2,\mathrm{switch}_2)|\\
&\mathrm{Station}_2(\alpha_2)|\mathrm{Control}_2(\mathrm{lose}_2,\mathrm{inform}_2)).
\end{aligned} \tag{2}$$

According to Petri nets semantics of π-calculus, the processes of the station, car and central controller are translated into Petri nets, as shown in Fig. 5(a). The dotted places are abstract places which are the interfaces with the other processes. Petri nets models of $\mathrm{Station}_1$ and $\mathrm{Station}_2$ are similar, therefore, only the model of $\mathrm{Station}_1$ is shown. The model of $\mathrm{Control}_1$ is also shown in the models of $\mathrm{Control}_1$ and $\mathrm{Control}_2$. According to allelomorph names, the related transitions of Petri nets models are combined into communication transitions, and then the Petri nets model N_{System} of the systems is set up. N_{System} is shown in Fig. 5(b), where Petri nets semantics of the processes are shown in the dotted frame, and the transitions talk_i, lose_i, switch_i, inform_i are communication transitions (τ). According to N_{System}, two tokens in place P_1 and P_7 represent Car_1 and Car_2 respectively, and Car_1 and Car_2 are concurrently executed. Therefore, the Petri nets semantics can describe the concurrent action. Once the Car_1 is rather far from $\mathrm{Station}_1$, $\mathrm{Control}_1$ inform Car_1 switch its channel to $\mathrm{Station}_2$.

The Petri nets supporting tools (such as INA, http://www2.informatik.hu-berlin.de/~starke/ina.html) can be employed to analyze and verify the model after π-calculus is translated into Petri nets model. According to the INA analysis results, N_{System} is bounded, the number of reachable states is 21, and it is active.

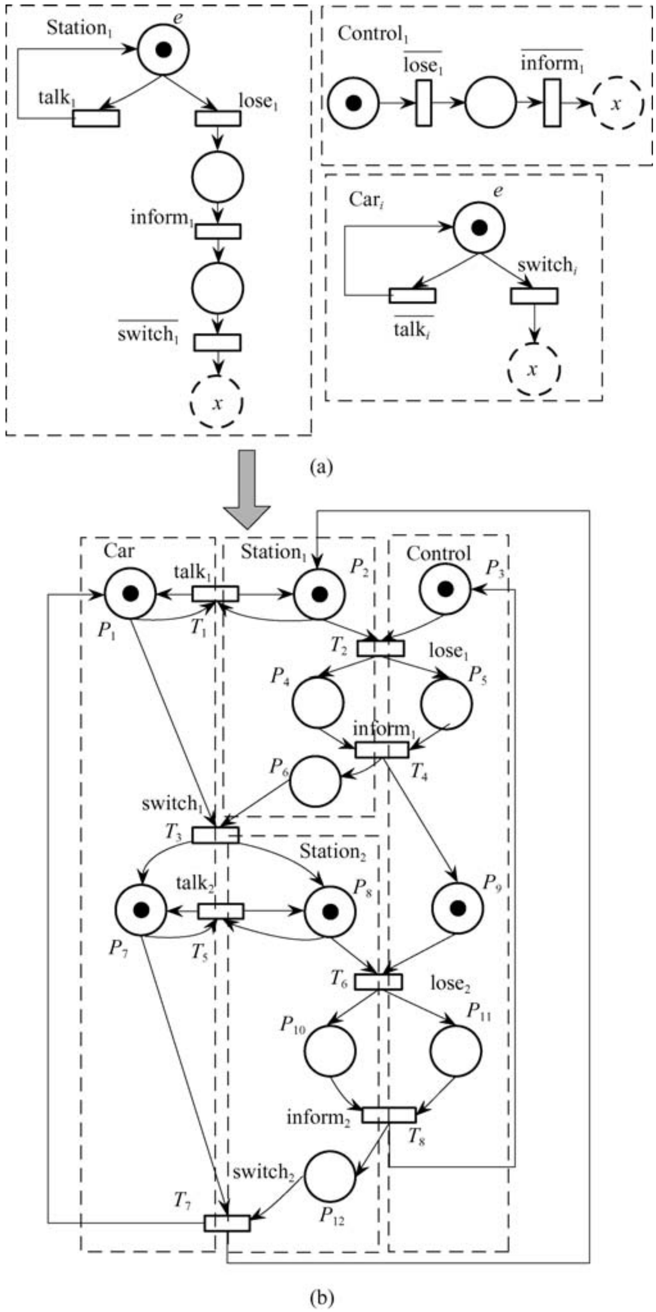

Fig. 5 Petri nets semantics for mobile telephone system. (a) Petri nets semantics; (b) Petri nets model

4 Conclusions

In this paper, π-calculus is divided into basic elements, sequence, concurrency, choice and recursive modules. These modules are translated into Petri nets that compose complex systems. The reason for π-calculus mapping into Petri nets is that Petri nets are intuitive and understandable tools that depict the true concurrency of systems and they have many support tools. Finally, according to Petri nets semantics of π-calculus, the processes of the mobile telephone system are effectively translated into Petri nets model analyzed by INA.

References

1. Jiang C J. Behavior Theory and Applications of Petri Nets. Beijing: Higher Education Press, 2003 (in Chinese)
2. Milner R, Parrow J, Walker D. A calculus of mobile processes. Journal of Information and Computation, 1992, 100(1): 1–77
3. Murata T. Petri nets: properties, analysis, and application. In: Proceedings of the IEEE, 1989, 77(4): 541–580
4. Busi N, Gorrieri R. A Petri net semantics for π-calculus. Lecture Notes in Computer Science, 1995, 962: 145–159
5. Engelfriet J. A multiset semantics for the π-calculus with replication. Theoretical Computer Science, 1996, 153(1–2): 65–94
6. Devillers R, Klaudel H, Koutny M. Petri net semantics of the finite π-calculus. Lecture Notes in Computer Science, 2004, 3235(2): 309–325
7. Cao M L, Wu Z M, Yang G K. π-net: a new modular high level Petri nets. Journal of Shanghai Jiaotong University, 2004, 38(1): 52–58 (in Chinese)
8. Mazzeo A, Mazzocca N, Russo S, et al. Formal specification of concurrent systems: a structured approach. The Computer Journal, 1998, 41(3): 145–162
9. Best E, Koutny M. Process algebra: a petri-net-oriented tutorial. Lecture Notes in Computer Science, 2004, 3098: 180–209
10. Han T T, Chen T L, Yan F, et al. Study on expressive power of synchronous asynchronous π-calculus. High Technology Letters, 2005, 15(8): 18–22 (in Chinese)
11. Ribaudo M. Stochastic Petri net semantics for stochastic process algebras. In: Proceedings of the 6th International Workshop on Petri Nets and Performance Models, Los Alamitos. IEEE Press, 1995, 148–157
12. Milner R. Communicating and Mobile Systems: the π-calculus. Cambridge: Cambridge University Press, 1999