Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2013, 5(9):347-354

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

Process Net: A petri net model with the characteristics of process algebras

Guo Feng, Deng Mengmeng and Shi Wanlin

School of Information and Engineering, North China University of Technology, Beijing, China

ABSTRACT

The main shortcomings of Petri Nets are the lack of composability and the node explosion problem when modeling of complex systems. A new Petri net model named Process Net (PrN) is proposed which has the characteristics of Process Algebras. By means of four composition operators, complex Process Net can be constructed from existing simple ones. A structured operational semantic for Process Net is defined so that a process net can be analyzed by Process algebra or Petri nets methods. An application example to illustrate the application and analysis method of Process Net is described.

Key words: Petri Net, Process Algebra, Structured Operational Semantics

INTRODUCTION

Petri Net and Process Algebra are two main formal models for describing concurrence systems widely used. They have obviously advantages and shortcomings. There are very fierce debates about them. The main shortcomings of Petri Nets are the lack of composability and the node explosion problem when modeling of complex systems. In order to solve those problems researchers have presented varies high Petri Nets models, such as Hierarchical Petri Nets, Colored Petri Nets, Modular Petri Nets, etc. In those papers [1,2] overviews of this research field were introduced. Process Algebras have composability naturally but lack of true concurrence semantics and graphical representation. Combining Petri nets and Process algebras is an interesting research area for researchers which mainly has three aspects including describing the semantics of Process algebras using Petri Nets[14, 4, 5],Expressing Petri Nets using Process algebras[6], Describing software systems using Petri nets and Process algebras[7].

This paper proposes Process Net (PrN for short) which is a new Petri Net model with characteristics of Process algebras Based on the WF_Net[8] and Open net[9] and the study of related theory on the basis of Process algebras. In such research combing Petri nets and Process algebras as mentioned above, Petri nets and Process algebras are separate. This paper adopts a different way, the direct use of the basic Petri net components as structural components of algebraic expressions. By means of six type composition operators, complex Process Net can be constructed from simple ones. PrN has the same operation rules as general Petri net, at the same time it has the structured operational semantics similar to Process algebras which don't rely on Petri net markings and allow representing the states of the system with Process Net algebraic expressions. Process Net is analogous to π-Net [7] , but the purpose of π-Net is mainly to propose a Petri net model that can implement the structured operational semantics of π-calculus. The structured operational semantics of π-Net was defined based on the markings of Petri nets. The main purpose of this paper is to propose an algebraic system of Petri net with the Petri net components as the smallest unit and complex Process Net can be constructed from simple ones.

The remainder of this paper is organized as follows:

The second section gives the definition of Process net and various composition operators.The third section gives the structured operational semantics of PrN. The fourth section illustrates the application and analysis method of Process Net through an example. The fifth section summarizes the primary research contents and pointed out what need be studied in the future.

PROCESS NET

The definition of Process Net

Process Net is defined using a recursive way. A Process Net is a basic Process Net or composed by other process Nets through various composition operators. Process Net can accurately describe the basic communication actions and common control structures of concurrence systems. There are four kinds composition operators used for combining Process Nets which are Sequence(;) , Select(+), Parallel(include Synchronous Parallel ($|<x_1,x_2, \ldots,x_n>|$), Asynchronous Parallel ($|[x_1,x_2, \ldots,x_n]|$, and interleaving Parallel ($||$)) and Loop ($\Diamond$). Sequence operator represents actions are executed in order. Select operator represents conditional choose an action to perform. Synchronous Parallel operator represents synchronous interactions on the interface transitions of concurrency Process Nets similar to the transitions fusion in Petri nets [10]. Asynchronous Parallel operator represents asynchronous interactions on the interface transitions of concurrency Process Nets similar to the places fusion in Petri nets [11]. Parallel operator can be expressed as $||$ when the interface transitions are null that represents full concurrency, i.e. two Process nets execute alternately without any interaction.

Definition 1[12]：Petri Net is 3-tuple N=（P,T,F）, satisfied：
(1) $P \cap T=\emptyset$ 及 $P \cup T \neq \emptyset$;

(2) $F \subseteq P \times T \cup T \times P$ satisfied: $dom(F) \cup cod(F) = P \cup T$;
$dom(F) = \{x \mid \exists y: (x, y) \in F\}$ is the definition domain and $cod(F) = \{y \mid \exists x: (x, y) \in F\}$ is the value domain of F. P is the set of places and T is the set of transitions; F is the arcs set between T and P called flow relation. (x, y) is the output arc of x, and the input arc of y. $x \in P \cup T, {}^{\bullet}x=\{y|(y, x) \in F\}$ is the preset of x, $x^{\bullet}=\{y|(x, y) \in F\}$if the poset of x.

(3) m_0: $P \rightarrow N$ is the initial mark, satisfied $\forall p \in P, M_0(p) \leq K(p)$.
Definition 2 [13]：Let PN = (P, T, F) is a Petri Net，β：$P \cup T \rightarrow S \cup \{\tau\}$ is the labeling function maps net elements to strings of symbols，S is the set of label symbols. τ is the silent label. $PN' = (P, T, F, \beta)$ is called Labeled Petri nets.

Definition 3: Process Net(PrN for short) is a 6-tuple: (P, T, F, β, I, O), in which (P,T,F, β) is a Labeled Petri Net that has two special places: i and o. i is the starting place, ${}^{\bullet}i=\emptyset$, o is the ending place , $o^{\bullet}=\emptyset$. β is the labeling function, $\beta:T \rightarrow A \cup \{\tau\}$, A is a set of actions which are strings of letters represent the meanings of transitions and each transition is labeled with an action. $I \in T$ is a transition set which receive messages, $O \in T$ is a transition set which send messages. Transition included in I or O are interface transitions of a Process Net. There are three forms of action: a, !a and ?a, representing internal action, sending action and receiving action. Transitions in I are labeled by ?a, Transitions in O are labeled by !a.

Complex PrN can be constructed from simple ones. The corresponding graphics of a, !a and ?a shown in figure 1 which called basic PrNs:

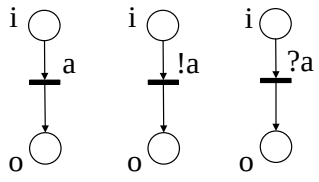

Fig.1 Basic Process Net

Definition 4: Basic PrN is PrN, satisfy :
P={i,o}，T= {t}，F={(i,t),(t,o)}, A={a,!a,?a}. If β(t)=a, I and O are null. If β(t)=!a ,I is null and O={t}. If β(t)=?a ,O is null and I={t}.

PrN has more concepts like interface transition and labeling function compared with the WF_Net. Compared with the Open Net in PrN interface is represented by transitions instead of places. PrN only has one begin place and one end place. These characteristics make PrN can have the algebraic expression similar to process algebra and a

forgetting type structured operational semantics.

Figure 2 is a PrN graphic where interface transitions are represented by black rectangles and internal transitions are represented by white rectangles. When complex PrNs are composed of simple ones the internal details needn't be considered.

The Algebraic Expression of PrN

The four PrN composition operators introduced above are the grammar rules of PrN actually. The basic PrNs can be expressed by grammar rules as follows:

Definition 5: Grammar rules of Basic PrNs:

E：=Ea|E?a|E!a

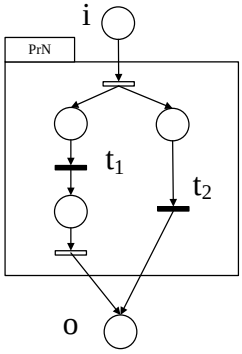

Fig. 2 A complex Process Net

Grammar rules of basic PrNs set up the corresponding relationship between basic PrNs and the basic elements of algebraic expressions of PrN. In the case of given four composition operators, PrN can be described by the following syntax format where E represents basic PrNs and P represents complex PrNs.

Definition 6：Grammar rules of PrN：

P:=E | P+Q | ◊P | E;P | P|< x_1,x_2, …,x_n >|Q | P|[x_1,x_2, …,x_n]|Q

Composition operators + and |<x_1,x_2, …,x_n>| and |[x_1,x_2, …,x_n]| are Commutative. If the set {x_1,x_2, …,x_n} is null, operator || is called interleaving parallel.

The following we illustrate the four composition operators how to be achieved by composing two PrNs. The two PrNs are PrN_1 and PrN_2, i_1, o_1 are respectively the begin place and end place of PrN_1, and i_2,o_2 are respectively the begin place and end place of PrN_2. PrN_0 is the PrN after the composition. In the mentioned parallel, sequence and select composition, the interfaces of PrN_1 and PrN_2 became the interfaces of PrN_0, and in loop composition, the interfaces of PrN_1 became the interfaces of PrN_0.

Interleaving parallel composition: add a new begin place i_0 and end place o_0, add transition t_1, add arc (i_0, t_1), (t_1, i_1), (t_1, i_2), add transition t_2, add arc (t_2,o_0),(o_1,t_2),(o_2,t_2),

Select composition: add a new begin place i_0 and ending place o_0, add transition t_1, t_2, add arc(i_0, t_1), (i_0, t_2), (t_1, i_1), (t_2, i_2), add transition t_3,t_4 add arc(o_1,t_3),(o_2,t_4) , (t_3,o_0),(t_4,o_0)。

Loop composition: add a new begin place i_0 and ending place o_0, add transition t_1, t_2， add arc(i_0, t_1), (t_1, i_1), (o_1, t_1), (o_1, t_2), (t_2, o_0)。

Sequence composition: Let i_0= i_1, o_0= o_2。 Add transition t_1, add arc (o_1, t_1), (t_1, i_2)。

Sequence, select, interleave parallel and loop composition operators are not involved in interfaces of PrNs , these composition operators deal PrN as black box, and considering only the combination of begin place and end place. These four composition operators are illustrated in figure3 and 4:

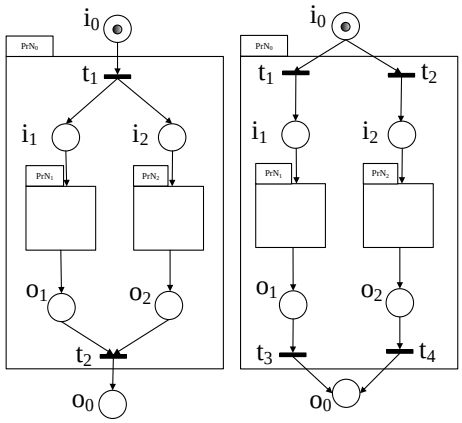

Fig. 3 select and interleave parallel composition

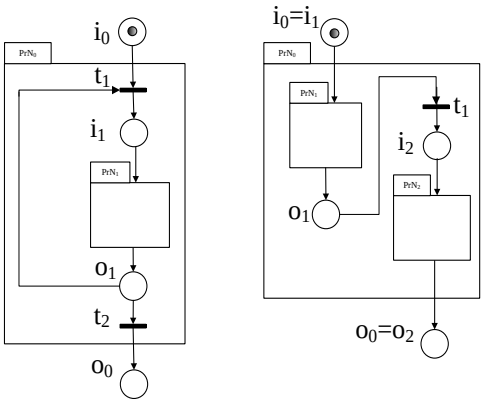

Fig. 4 Loop and sequence composition

For synchronous parallel composition, the transitions labeled as !a and ?a in two PrNs are merged to one transition labeled as a. Then the begin places and end places are merged as interleaving composition.

For asynchronous parallel composition, places represent messages need be added. For each transition labeled !a, add a place named a and add arc (t,p),(p,t'). t' is the transition labeled ?a. Then the begin places and end places are merged as interleaving composition.

These two composition operators are illustrated in figure 5.

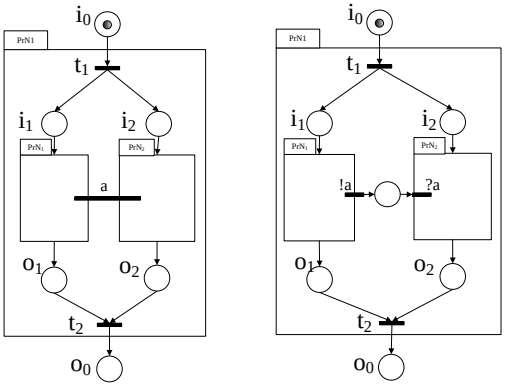

Fig. 5 Synchronous parallel and asynchronous parallel composition

Structured Operational Semantics for PrN

There is an important distinction between Petri nets and process algebras: During a Petri net execution its structure remains unchanged, while the structure of a process described by process algebra varies continuously with the

process running[14, 15]. In the research of structured Operational Semantics (abbreviation is SOS) for Petri nets, the SOS rules is as the same as process algebra's SOS formally, it still depends on Petri net's markings that need be calculated in the process of generating state space or reasoning using SOS.

PrN's run rules can be defined in accordance with common Petri net. As complex PrNs can be constructed from basic PrNs by means of composition operators, we can define PrN's SOS not depending on PrN's markings also. In the following PrN's SOS rules are given:

R1: Internal action rule：

$$\overline{E_a; P \xrightarrow{a} P}$$

R2: Sending action rule:

$$\overline{E_{!a}; P \xrightarrow{!a} P}$$

R3：Receiving action rule:

$$\overline{E_{?a}; P \xrightarrow{?a} P}$$

R4: Interleaving parallel rule:

$$\frac{P \xrightarrow{x} P'}{P \parallel Q \xrightarrow{x} P' \parallel Q}$$

x is one of the three basic actions.

R5：synchronization parallel rule 1：

$$\frac{P \xrightarrow{!a} P', Q \xrightarrow{?a} Q'}{P \mid< x_1, x_2, ..., x_n >\mid Q \xrightarrow{a} P' \mid Q'} \quad a \in \{x_1, x_2, ..., x_n\}$$

When a communication action is included in the set $\{x_1, x_2, ... , x_n\}$, both sides of this interaction must wait until the synchronization conditions satisfied.

R6：synchronization parallel rule 2：

$$\frac{P \xrightarrow{x} P'}{P \mid< x_1, x_2, ..., x_n >\mid Q \xrightarrow{x} P' \mid Q} \quad x \notin \{x_1, x_2, ..., x_n\}$$

When a communication action is not included in the set $\{ x_1, x_2, ... , x_n \}$, both sides of this interaction need not wait.

R7: Asynchronization parallel rule 1

$$\frac{P \xrightarrow{!a} P'}{P \mid [x_1, x_2, ..., x_n] \mid Q \xrightarrow{!a} P' \mid Q, A + a} \quad a \in \{x_1, x_2, ..., x_n\}$$

When a communication action is included in the set $\{ x_1, x_2, ... , x_n\}$, the sender produces a message which should be saved in set A.

R8: asynchronization parallel rule 2

$$\frac{Q \xrightarrow{?a} Q', a \in A}{P \mid [x_1, x_2, ..., x_n] \mid Q \xrightarrow{?a} P \mid Q', A - a} \quad a \in \{x_1, x_2, ..., x_n\}$$

When a communication action is included in the set $\{x_1, x_2, ... , x_n\}$, the receiver consumes a message which should be deleted from set A.

R9: asynchronization parallel rule 3：

$$\frac{P \xrightarrow{x} P'}{P \mid [x_1, x_2, ..., x_n] \mid Q \xrightarrow{x} P' \mid Q} \quad x \notin \{x_1, x_2, ..., x_n\}$$

When a communication action is not included in the set { x_1, x_2,... , x_n}, both sides of this interaction interleave.

R10：Loop rule：

$$\frac{P \xrightarrow{x} P'}{\Diamond P \xrightarrow{x} P' \| \Diamond P}$$

x is one of the three basic actions.

In semantic rule definition above, we don't consider PrN markings. When generating PrN transition system using those rules, the constantly changing algebra expression of PrN itself can represent transition system's status. Basic PrNs are discarded during execution, thus a forgetting kind of structured operational semantics is formed.

EXAMPLE OF APPLICATION

The following part we illustrate how to use PrN describe web services by the example of a web-service-based ERP system in which three are three web services that related with dealing orders. The three web service "order service" "process service" and "store service" and the composition of them are illustrated in figure 6.

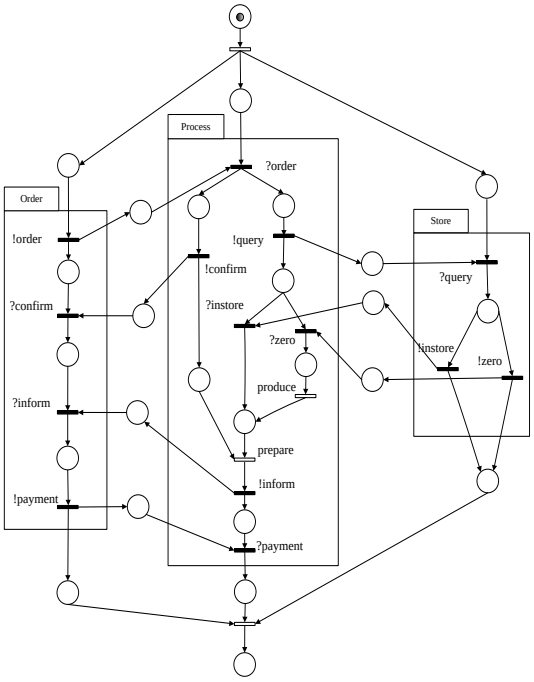

Fig.6 Composite WS_Net dealing orders

The algebra expression of this composite service is：
System:=Order|[order,confirm,inform,payment]|Process|[query,instore,zero]|Store
Order:= !order;?confirm;?inform;!payment

Processing:=?order;(!confirm||Query);prepare;!inform;?payment
Store:=?query;(!instore+!zero)
Query:=!query;(?instore+?zero;produce)

We can know that PrN System is composited from three services. Based on structured operational semantics introduced above, we can obtain transition system of System, fig. 7 show part of it. Transition system's states are represented by algebra expressions, state 1 is System, at which Order's !order action can be executed, then message order is added to set A, at the same time, Order become : ?confirm;?inform;!payment, so that status2 becomes:
Order:=?confirm;?inform;!payment

Processing:=?order;(!confirm||Query);prepare;!inform;?payment
Store:=?query;(!instore+!zero)
Query:=!query;(?instore+?zero;produce)
A={order}.

Query:=!query;(?instore+?zero;produce)
A={order}.

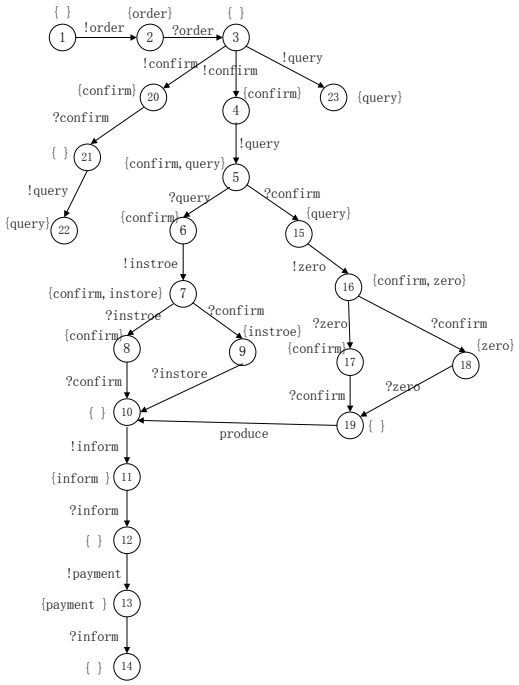

Fig. 7 Transition system of Composite PrN

Compared with reachability graph that generated by traditional Petri nets fire rules, Fig7 shows exactly the same behaviors. Based on the transition system properties like safety, liveness, bisimularity and so on can be analyzed.

CONCLUSION

This paper proposes the concept of Process Net Based on WF_Net , Open net and process algebras. Complex PrNs can be constructed from simple ones through four composite operators. The simplest PrNs called basic PrNs. Basic PrNs can correspond directly the basic component of PrN algebra expressions. Composite operators prescribe the grammatical rule of algebra expression. The research combines graphic representation of Petri nets and the semantic definition way of process algebras that make PrN has the semantic of traditional fire rules of Petri nets not only, but also has SOS not depending on PrN's markings. PrN can combine the advantages of Petri nets and Process algebras when modeling and analyzing the behaviors of complex concurrence systems.

PrN's structured operational semantics put forward in this article lay the foundation for the research of PrN bisimularity. In future, we should research on the PrN bisimularity not depending on net markings. In order to support the applying of PrN in practice work, a modeling and validating tool for PrN will be designed and implemented soon.

Acknowledgment
This paper is supported by NSFC(National Science Foundation of China) project numbered 61070030 and PHR(IHLB) (Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality numbered PHR201107107).

REFERENCES

[1] Gomes L, Barros J.P., **2005**. *IEEE Trans. on Industrial Informatics*, 1(2):112-123
[2] Hao KG, Ding JJ.,**2008**. *Journal of Frontiers of Computer Science and Technology*, 2(2):123-130
[3] Yu ZH, Cai YL, Xu HP. , **2007**. *Control and Decision*, 22(8) :864-868
[4] Best E, Devillers R, Koutny M.,**2002**. *Inf. Comput.*, 178(1):44-100.
[5] Cao ML, Wu ZM, Yang GK.,**2004**. *Journal of Shanghai Jiaotong University*,38(1): 52－58
[6] Hao KG, Guo XQ, Li XN.,**2011**. *Chinese Journal of Computers*, 34(2):193-203

[7] Basten T.,**1998**. In terms of nets: System design with Petri nets and process algebra, Ph.D. Thesis, Eindhoven University of Technology, Department of Mathematics
[8] W.M.P. van der Aalst.,**1997**. Verification of Workflow Nets, Application and Theory of Petri Nets **1997**, volume 1248 of Lecture Notes in Computer Science, Berlin:Springer-Verlag, 407-426
[9] van der Aalst W.M.P., Mooij A.J., Stahl, C., Wolf, K.,**2009**. Service Interaction : Patterns, Formalization, and Analysis. Formal Methods for Web Services : 9th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, 42-88
[10] Pu F, Lu WM.,**2003**. *Journal of Software*, 14(12):1977-1988
[11] Pu F, Lu WM.,**2005**. Sharing Synthesis of Petri Net Systems via Preserving Liveness. System Engineering-Theory and Practice, 11:70-78
[12] Yuan CY.,**2005**. Petri Net Theory. Beijing: Publishing House of Electronics Industry
[13] Störrle H.,**2000**. Models of Software Architecture. Munich :Ludwig Maximilians University
[14] Cao ML.,**2007**. Formal Analysis of Petri Nets and Cryptographic Protocol based on pi Calculus.Shanghai Jiaotong University Doctoral Dissertation
[15] Bergstra J A, Ponse A, Smolka S A,**2001**. Handbook of Process Algebra. North-Holland, Amsterdam, 873--944.