IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 25, 2017, accepted August 13, 2017, date of publication August 31, 2017, date of current version September 19, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2743113

Extended Logical Petri Nets-Based Modeling and Analysis of Business Processes

WEI LIU[1,2], PIN WANG[1], YUYUE DU[1], MENGCHU ZHOU[3,4], (Fellow, IEEE), AND CHUN YAN[4,5]

[1]College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[2]Shandong Province Key Laboratory of Wisdom Mine Information Technology, Shandong University of Science and Technology, Qingdao 266590, China
[3]Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA
[4]The Renewable Energy Research Group, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[5]College of Mathematics and System Science, Shandong University of Science and Technology, Qingdao 266590, China

Corresponding author: Mengchu Zhou (zhou@njit.edu)

This work was supported in part by the Project of National Natural Science Foundation of China under Grant 61472228 and Grant 61502280, in part by the Natural Science Foundation of Shandong Province under Grant ZR2014FM009, in part by the Project of Qingdao Applied Basic Research of Qingdao, special youth project, under Grant 14-2-4-55-jch.

ABSTRACT Extended logical Petri nets are proposed to improve logical Petri nets, and the related firing rules and state reachability graph are introduced. Their attributes, place arrival time, and priority function are defined for each token, batch processing wait time and transition firing duration are defined for each logical input/output transition, and firing duration is defined for each ordinary transition. Their token removal and generation functions are defined to operate the attributes and arrival time of tokens in places. A state in their reachability graph is redefined. They are used to model an e-commerce system. The reachability, time cost superiority of tokens with different priorities, and fairness of the resulting model are analyzed and correct end-states are illustrated.

INDEX TERMS Extended logical petri nets, E-commerce systems, priority, time.

I. INTRODUCTION

It is very important to use an established framework for modeling and analyzing business processes. Petri nets are modeling and analysis tools for distributed systems and processes [1]–[3]. They can be used to describe sequences, concurrency, conflicts and synchronization of a system's processes or components [4]–[6]. As a type of system models, Petri nets not only depict a system's structure, but also describe its dynamic behaviors such as deadlock and liveness [7], [8]. Several extensions to them have been proposed to widen their applications, such as timed Petri nets [9], logical Petri nets (LPN) [10] and hybrid Petri nets. An LPN is presented to model and analyze cooperative systems with both batch processing function and passing value indeterminacy. An LPN is equivalent to inhibition Petri nets [11]. The input and output of the transitions in LPNs are limited by logic expressions. Compared to general Petri nets, LPNs can describe the net structures of cooperative system models in a clear fashion for system designers to understand and utilize. Some work related to LPNs has been conducted, such as its modeling capability [12], analysis methods [13], soundness preservation in composition [14], transformation [15] and substitution [16]. LPNs have been applied in modeling stock trading systems [17], Web service processes [18]–[20] and E-commerce systems [21].

LPNs focus on the control flow of business processes with a batch processing function and passing value indeterminacy. In practice, types, differences, priorities and attributes of businesses or users are also crucial. For example e-commerce is an important application of the Internet. As the users increase and services diversify, e-commerce systems become more and more complicated. Various types of users require a system to respond differently to their requests. But LPNs cannot well describe and analyze them.

To tackle the above problems, we propose extended logical Petri nets (ELPN) in this paper. The attributes and arrival time of their tokens, the batch wait time of logical input/output transitions, and the firing duration of an ordinary transition are introduced. The concepts and techniques proposed in this paper are illustrated with an ELPN-based e-commerce system model. Based on the model, the user priority, fairness, and correct end-states are illustrated, demonstrating that an ELPN is an effective modeling and analysis tool for a complex system.

The rest of this paper is organized as follows. Section 2 reviews Petri nets and logical Petri nets. Section 3 is devoted to extended logical Petri nets, their related concepts and analysis methods. Section 4 analyzes an e-commerce system based on its ELPN model. Concluding remarks are made in Section 5.

II. PRELIMINARIES

This section briefly reviews the notations of Petri nets [22]–[23] and LPNs.

A. PETRI NETS

A Petri net is a model that can be represented by a graph.

Definition 1: $N = (P, T, F)$ is a net where

(1) P is a finite set of places;
(2) T is a finite set of transitions, $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$;
(3) $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs.

Definition 2: Let $x \in P \cup T$ denote any element of a net N. Given $x \in P \cup T$, ${}^{\bullet}x = \{y|(y, x) \in F\}$ is called the input set or preset of x, and $x^{\bullet} = \{y|(x, y) \in F\}$ is the output set or postset of x. $N = (P, T, F)$ is called a pure net if it satisfies $\forall t \in T$: ${}^{\bullet}t \cap t^{\bullet} = \emptyset$.

Definition 3: A 4-tuple $\sum = (P, T, F, M)$ is a Petri net if and only if

(1) $N = (P, T, F)$ is a pure net;
(2) M:$P \rightarrow N$is a marking function and M_0 is the initial marking;
(3) $\sum$ has the following transition firing rules:
   a. $t \in T$ is enabled in M, denoted as $M[t>$if $\forall p \in {}^{\bullet}t$: $M(p) \geq 1$;
   b. If $M[t>$, then t may fire in M, and its firing generates a new marking M', denoted as $M[t > M'$, where

$$M'(p) = \begin{cases} M(p) + 1 & if\ p \in t^{\bullet} - {}^{\bullet}t \\ M(p) - 1 & if\ p \in {}^{\bullet}t - t^{\bullet} \\ M(p) & otherwise \end{cases} \quad (1)$$

B. LOGICAL PETRI NETS

Logical Petri nets can be used to simulate and analyze a batch processing function and passing value indeterminacy of their modeled systems. Let $\bullet T_{\bullet}$ and $\bullet F_{\bullet}$ denote logically true and false value respectively. An LPN is defined as follows.

Definition 4: Let $LN = (P, T, F, I, O)$, $LPN = (LN, M)$ is called a logical Petri net if and only if

(1) P is a finite set of places;
(2) $T = T_D \cup T_I \cup T_O$ is a finite set of transitions, $T \cup P \neq \emptyset$ and $\forall t \in T_I \cup T_O$: ${}^{\bullet}t \cap t^{\bullet} = \emptyset$, where
   a. T_D denotes the transitions in classic Petri nets;
   b. T_I denotes the logical input transition set of T, $\forall t \in T_I$, and all the input places of t are subjected to the constraint of a logical input expression $f_I(t)$;
   c. T_O denotes the logical output transition set of T, $\forall t \in T_O$, and all the output places of t are subjected to the constraint of a logical input expression $f_O(t)$;
(3) $F \subseteq (P \times T) \cup (T \times P)$ is a finite set of directed arcs;
(4) I is a logical input function. $\forall t \in T_I$, $I(t) = f_I(t)$ is a logical input expression;
(5) O is a logical output function. $\forall t \in T_O$, $O(t) = f_O(t)$ is a logical output expression;
(6) M: $P \rightarrow \{0,1\}$ is a marking function in which $M(p)$ denotes the number of tokens in $\forall p \in P$;
(7) The transition firing rules are as follows:
   a. $\forall t \in T_D$, the transition firing rule satisfies (1) in Definition 3;
   b. $\forall t \in T_I$, $I(t) = f_I(t)$. If $f_I(t)|_M = \bullet T_{\bullet}$, i.e. the logical expression $f_I(t)$ of ${}^{\bullet}t$ has a logically true value in marking M, then t is enabled in M; if t is enabled, it may fire; and its firing in M generates a new marking M': $\forall p \in {}^{\bullet}t$, $M'(p) = 0$, $\forall p \in t^{\bullet}$, $M'(p) = M(p) + 1$, $\forall p \notin {}^{\bullet}t \cup t^{\bullet}$: $M'(p) = M(p)$;
   c. $\forall t \in T_O$, $O(t) = f_O(t)$. If $\forall p \in {}^{\bullet}t$: $M(p) = 1$, then t is enabled in M; if t is enabled, then it can fire; and firing t in M generates a new marking M': $\forall p \in {}^{\bullet}t$:$M'(p) = M(p)$-1, $\forall p \notin t^{\bullet} \cup {}^{\bullet}t$, $M(p) = M(p)$. For $t^{\bullet}$, $f_O(t)|_M = \bullet T_{\bullet}$ should be satisfied, i.e., the logical expression $f_O(t)$ of $t^{\bullet}$ in marking M' should have a logically true value.

The logical input transitions and logical output transitions of an LPN are subjected to the constraints of logical input expression $f_I(t)$ and logical output expression $f_O(t)$, respectively. Transitions that are subjected to constraints of logical expressions are called logical transitions. Logical transitions express the indeterminacy in their input/output places by logical expressions.

III. EXTENDED LOGICAL PETRI NETS

In an e-commerce system, when multiple users of various types send requests at the same time, a problem of user request processing order is raised. Reflected in an LPN, it is a problem of which token to process first when multiple tokens exist in a place. Currently, token priority is not defined in LPNs. Extended logical Petri nets add attributes and arrival time to each token and configure logical input/output transition batch wait time and firing duration, which solve the token processing order problem.

The definition of ELPNs and their firing rules are as follows:

Definition 5: $ELPN = (LN, A, T_{arr}, V_p, G_p, B, T_{dur}, S_0)$ is an extended logical Petri net where,

(1) LN is as defined in Definition 4;
(2) A is the set of attribute variables, $A(p) \rightarrow 2^{A}$ denotes the set of attribute variables contained in place p, $A_p(tk)$ indicates $A_p(tk) \rightarrow (v_1, v_2, \ldots, v_m)$, i.e. the attribute variables of token tk in place p, and $A_p(tk)$ denotes the value of the attribute variables of token tk in place p;
(3) $T_{arr}(p) = (a_{tk1}, a_{tk2}, \ldots, a_{tkn})$ is the set of arrival times of the tokens in place p, a_{tki} denotes the time that token tki arrives place p, $t_{early}(p)$ denotes the arrival time of the token that arrives first among all current tokens in p, $T_{early}(t)$ denotes the smallest value

of $t_{early}(t)$ in the pre-places of transition t, $t_{last}(p)$ denotes the arrival time of the token that arrives last, and $T_{last}(t)$ denotes the largest value of $t_{last}(p)$ in the pre-places of t.

(4) $V_p(tk)$ is the token removal function that removes a used token tk from the place p of preset of a fired transition t.

(5) $G_p(tk)$ is the token generation function that generates new tokens in the place p of postset of a fired transition t. $G_{Vp}(tk)$ is the attribute writing function that records the attributes of the new token in the post-places of the fired transition and $G_{Tp}(tk)$ is the time writing function that records the arrival time of the new token in the post-places of the fired transition.

(6) $B(t)$ is the batch wait time. $\forall t \in T_I \cup T_O$, with the arrival time $T_{early}(t)$ of the first token in the pre-places of transition t as the start time, $B(t)$ is the batch wait time for batch processing.

(7) $T_{dur}(t)$ is the time for t to process a token. The time cost for processing m tokens is $mT_{dur}(t)$;

(8) S_0 denotes the initial state of the ELPN.

As Definition 5 shows, a token in an ELPN has different meaning from that in an LPN. It is defined as follows:

Definition 6: Let *TK* be the set of tokens in *ELPN*, where $tk = (a_{ik}, A_p(tk))$, a_{ik} is the time that token *tk* arrives at the place and $A_p(tk)$ is the attribute value of token *tk*.

Definition 7: A state in ELPNs is denoted by S. The state $S(p)$ of a place p is expressed in the form of $[n, (a_1; v_{11}, \ldots, v_{1i}), (a_2; v_{21}, \ldots, v_{2i}), (a_n; v_{n1}, \ldots, v_{nk})]$, where

(1) n is the number of tokens in a place p;

(2) Each $(a_u;\ v_{u1}, \ldots, v_{ul})$ labels a token. a denotes the time that token u arrives at the place, and v_{uv} denotes the v-th attribute value of token u, $v = 1, 2, \ldots, l$;

If a place has no token, its $S(p)$ value is 0. $|S(p)|$ denotes the number of tokens in place p, and $|S|$ is a vector about the number of tokens in place set P.

Definition 8: After a transition t fires, $\forall p \in {}^{\bullet}t$, if the tokens are removed from p, the state of p transforms from $S(p) = [n,\ (a_1;\ v_{11}, \ldots, v_{1i}),\ (a_2;\ v_{21}, \ldots, v_{2i}),\ (a_n;\ v_{n1}, \ldots, v_{nk})]$ to $S^{'}(p) = [m, (a_1;\ v_{11}, \ldots, v_{1i}),\ (a_2;\ v_{21}, \ldots, v_{2i}),\ (a_m;\ v_{m1}, \ldots, v_{mk})]$, denoted as $S'(p) = \mathrm{S(p)} \circledcirc \alpha$, where $\alpha = \{(a_1;\ v_{11}, \ldots, v_{1w}) | (a_1;\ v_{11}, \ldots, v_{1w})$ is the token removed from p when t fires$\}$, and $S'(p) = S(p) \oplus \alpha$ means that

(1) $|S'(p)| = |S(p)|\text{-}|\alpha| = n\text{-}|\alpha| = m$, where $|\alpha|$ is the number of tokens removed from p; and

(2) Function $V_p(tk)$ deletes the token component corresponding to α from $S(p)$.

After t fires, $\forall p \in t^{\bullet}$, if and only if there are new tokens moved to p, the state of p changes from $S(p)$ to $S'(p)$. $S'(p) = S(p) \oplus \beta$, where $\beta = \{(a_u;\ v_{u1}, \ldots, v_{uv}) |\ (a_u;\ v_{u1}, \ldots, v_{uv})$ denotes a new token of p after t fires$\}$, and $S'(p) = S(p) \oplus \beta$ means that

(1) $|S'(\mathrm{p})| = |S(p)| + |\beta|$, where $|\beta|$ is the number of tokens moved to p; and

(2) Functions $G_p(tk)$ writes the new tokens of p to the corresponding components of $S(p)$.

Definition 9: Let f be a logical expression of the place set P and S a state of ELPNs. For every $p \in P$, $p|_{|S|}$ denotes the truth value of p in f in $|S|$, and

$$p|_{|S|} = \begin{cases} \bullet T \bullet, & if\ |S(p)| \geq 1 \\ \bullet F \bullet, & if\ |S(p)| = 0. \end{cases}$$

Substituting the truth values of all places in the logical expression, the obtained $f|_{|S|} = \cdot T \cdot / \cdot F \cdot$ is the truth value of f in $|S|$. The value $\cdot T \cdot / \cdot F \cdot$ indicates a logical value of true/false.

For example, in the *LDPN* model in Fig. 1, t_1 is a logical input transition and its occurrence is not only constrained by the logical expression $f_I(t_1) = p_1 \wedge (p_2 \vee p_3)$ of its input places p_1-p_3, but also constrained by time. This means even after p_1-p_3 satisfy the logical expression $f_I(t_1)$, t_1 needs to wait for the corresponding batch wait time. Meanwhile, t_3 is a logical output transition and its output is not only constrained by the logical expression $f_O(t_3) = p_6 \wedge (p_7 \vee p_8)$ of its output places p_6-p_8, but also constrained by the corresponding batch wait time. This means that t_3 only fires at the end of the batch wait time, and after its firing, p_6-p_8 must satisfy logical expression $f_O(t_3)$. In Fig. 1, the non-bracketed numbers next to logical input and output transitions are the values of batch wait time $B(t)$, the bracketed numbers next to logical or ordinary transitions are the time $T_{dur}(t)$ for processing one token.

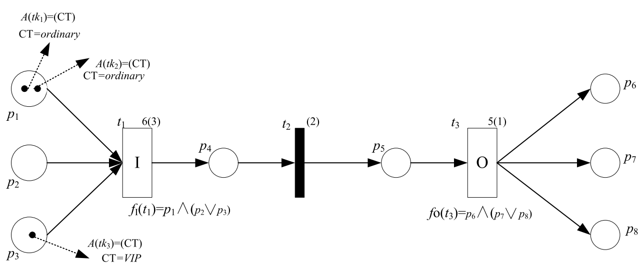

FIGURE 1. An ELPN model.

For an ELPN, in a state S, each place of the logical expression corresponds to a truth value, and the expression corresponds to one as well. For example, in the *ELPN* model shown in Fig. 1, $|S_0| = (2, 0, 1, 0, 0, 0, 0, 0)$, ${}^{\bullet}t_1 = \{p_1, p_2, p_3\}$, $f_I(t_1) = p_1 \vee (p_2 \wedge p_3)$, thus $p_1|_{|S0|} = \cdot T \cdot$, $p_2|_{|S0|} = \cdot F$, $p_3|_{|S0|} = \cdot F \cdot$, and $f_I(t_1)|_{|S0|} = \cdot T \cdot \vee (\cdot F \cdot \wedge \cdot T \cdot) = \cdot T \cdot$. Since every token has attributes, they can be prioritized based on their values. A function that calculates token priority is defined as follows.

Definition 10: The priority of a token $tk \in TK$ is denoted by $\Omega(tk)$. The value of $\Omega(tk)$ is defined and calculated based on attribute values of the token. The symbol $\cdot >$ is used to indicate that the token on its left has higher priority than the one on its right, and the symbol $\triangleq$ indicates equivalence.

Definition 11: Token processing rules

(1) Given $t \in T_O \cup T_I$, which can fire,

a. $\forall p \in^{\bullet} t$, the batch wait time period of t is $< T_{early}(t), T_{early}(t) + B(t) >$, i.e. the batch wait time has minimum and maximum values of $T_{early}(t) + B(t)$. Comparing the components $a_i (1 \leq i \leq n)$ in the arrival time $T_{arr}(p)$ of the tokens in p with the minimum and maximum values of batch wait time, if the arrival time a_i of tk_i satisfies inequality $T_{early}(t) \leq a_i \leq T_{early}(t) + B(t)$, then tk_i is in the batch processing time period. Such tokens constitute a batch token set $B_{TK} = \{tk_1, tk_2, \ldots\ldots, tk_m\}$.

b. The priorities of all tokens in B_{TK} are determined by the priority function.

- If $\Omega(tk_i) \cdot\!> \Omega(tk_j)$ ($tk_i, tk_j \in B_{TK}$), then tk_i is processed at $T_{early}(t)+B(t)$ while token j waits to be processed; token j is only processed after token i is processed.
- If $\Omega(tk_i) \triangleq \Omega(tk_j)$, i.e. tokens i and j have the same priority, then both tokens are processed at the same time at $T_{early}(t) + B(t)$.

(2) Given $t \in T_D$, which can fire,

$\forall p \in^{\bullet} t$, if there exist tokens with different priorities in the place at the same time, then the tokens with higher priorities are processed first and the ones with lower priorities wait to be processed; the low-priority tokens are only processed after the high-priority tokens are processed.

Assume that a transition can fire after the batch wait time period ends and all firing conditions are fulfilled, we have the following results.

Theorem 1: After $t \in T_I$ fires, the arrival time of the tokens in its post-places is

$$a_{tk}(\Omega(tk) = i) = T_{early}(t) + B(t) + \sum_{j=1}^{i} m_j * T_{dur}(t) \quad (2)$$

if the priority of the tokens is i where m_j denotes the number of tokens with priority j.

Proof: Given $t \in T_I$, we assume that the tokens in the batch processing time period of t has k levels of priorities, and the maximum batch wait time is $T_{early}(t)+B(t)$. Assume that $m_1, m_2, \ldots\ldots$ and m_k tokens have priorities 1, 2, ..., and k. As defined in Definition 5, the time on processing m_j tokens with the same priority is $m_j T_{dur}(t)$. Thus we have:

The time that tokens with priority 1 arrive at the post-place is $a_{tk}(\Omega(tk) = 1) = T_{early}(t) + B(t) + m_1 T_{dur}(t)$.

The tokens with priority 2 need to wait for the processing of the tokens with priority 1. Thus the time that tokens with priority 2 arrive at the post place is

$$\begin{aligned} a_{tk}(\Omega(tk) = 2) &= a_{tk}(\Omega(tk) = 1) + m_2 T_{dur}(t) \\ &= T_{early}(t) + B(t) + m_1 T_{dur}(t) + m_2 T_{dur}(t) \\ &= T_{early}(t) + B(t) + (m_1 + m_2) T_{dur}(t) \end{aligned}$$

The tokens with priority 3 need to wait for the tokens with priority 2 to complete their processing. Thus the time that tokens with priority 3 arrive at the post place is

$$\begin{aligned} & a_{tk}(\Omega(tk) = 3) \\ &= a_{tk}(\Omega(tk) = 2) + m_3 T_{dur}(t) \\ &= T_{early}(t) + B(t) + (m_1 + m_2) T_{dur}(t) + m_3 T_{dur}(t) \\ &= T_{early}(t) + B(t) + (m_1 + m_2 + m_3) T_{dur}(t) \end{aligned}$$

Likewise, the tokens with priority i need to wait for the processing of the tokens with priority i-1 to complete. Thus the time that tokens with priority i arrive at the post place is

$$\begin{aligned} & a_{tk}(\Omega(tk) = i) \\ &= a_{tk}(\Omega(tk) = i - 1) + m_i T_{dur}(t) \\ &= T_{early}(t) + B(t) + (m_1 + m_2 + \ldots\ldots + m_{i-1}) \\ &\quad \times T_{dur}(t) + m_i T_{dur}(t) \\ &= T_{early}(t) + B(t) + (m_1 + m_2 + \ldots\ldots + m_i) T_{dur}(t) \end{aligned}$$

Theorem 2: After $t \in T_O$ fires, the arrival time of the tokens in its post-places is

$$a_{tk}(\Omega(tk) = i) = T_{early}(t) + B(t) + \sum_{j=1}^{i} \mu_j * T_{dur}(t) \quad (3)$$

if the priority of the tokens is i where μ_j denotes the number of tokens with priority j.

Proof: The proof of this theorem is similar to that of Theorem 1.

Theorem 3: After $\forall\ t \in T_D$ fires, the arrival time of the tokens in its post-places is

$$a_{tk}(\Omega(tk) = i) = T_{last}(t) + \sum_{j=1}^{i} \lambda_j * T_{dur}(t) \quad (4)$$

if the priority of the tokens is i where λ_j denotes the number of tokens with priority j.

Proof: The proof of this theorem is similar to that of Theorem 1.

Definition 12: Transition firing rules in an ELPN

a) $\forall t \in T_I$, $I(t) = f_I(t)$. $\forall p \in^{\bullet} t$, if the value of logical input expression $f_I(t)$ is true in state S, then t is enabled in S; if t is enabled, it can fire and the tokens in its pre-places are processed following rule (1) in Definition 11. Firing t in S generates a new marking S': $\forall p \in^{\bullet} t$: $S'(p) = S(p) © M_1$ (M_1 is the set of the tokens in p that have been processed); $\forall p \in t^{\bullet}$: $S'(p) = S(p) \oplus M_2$ (M_2 is the set of tokens received by p);

b) $\forall t \in T_O$, $O(t) = f_O(t)$. If $\forall p \in^{\bullet} t$: $|S(p)| \geq 1$, then t is enabled in S; if t is enabled, it can fire and the processing of tokens follows rule (1) in Definition 11. In addition, firing t in S generates a new marking S': $\forall p \in^{\bullet} t$: $S'(p) = S(p) © M_1$; meanwhile, for $t^{\bullet}$, $f_O(t)|_{S'|} = \cdot T \cdot$ and $S'(p) = S(p) \oplus M_2$;

c) For $\forall t \in T_D$, if $\forall p \in^{\bullet} t$: $|S(p)| \geq 1$, then t is enabled in S; if t is enabled, it can fire and the movements of tokens follow rule (2) in Definition 11. In addition, firing t in S generates a new state S' : $\forall p \in {}^{\bullet}t$: $S'(p) = S(p) \circledcirc M_1$; $\forall p \in t^{\bullet}$: $S'(p) = S(p) \oplus M_2$.

In an ELPN, a transition t is enabled in state S, denoted as $S[t>$; if t is enabled, it can fire in the state S and firing it generates a subsequent state S', denoted as $S[t > S'$.

A practical case is used here to illustrate the expression of prioritized *ELPNs*. For the *ELPN* shown in Fig. 1, we assume a queue calling system of a bank, in which p_1-p_3 denote the three customers waiting to be called, and t_1 indicates that the system calls a number and processes the customer's request. In addition, batch wait time of t_1 is 6 time units, and the time for processing one customer's request is 3 time units. Based on the above analysis, the logical value of expression $f_I(t_1) = p_1 \vee (p_2 \wedge p_3)$ in state S is true. Now we assume that all three customers' requests are within the batch processing time period, the two tokens in p_1 represent two customers and the one token in p_3 represents one customer. The attribute variable CT of the tokens indicates customer type. For the two tokens in p_1, CT = ordinary (ordinary customer), and for the token in p_3, CT = VIP (VIP customer). Assuming that the customer priority is determined based on customer type, then according to the priority function, $\Omega(tk_1) = \text{VIP} \cdot > \Omega(tk_1) = \text{ordinary} \triangleq \Omega(tk_2) = \text{ordinary}$, i.e. VIP customers have higher priorities than ordinary ones. Based on the defined transition firing rules, the request of the VIP customer in p_2 is processed first and the requests from the ordinary customers in p_1 wait to be processed. The system processes requests from ordinary customers after finishing processing those from VIP customers, thereby saving time for VIP customers.

Based on the token processing rules and transition firing rules of ELPNs, the algorithm for generating their state reachability graph R(ELPN) is given below.

Algorithm 1 State Reachability Graph Generation Algorithm

Input: $ELPN = (LN, A, T_{arr}, V_p, G_p, B, T_{dur}, S_0)$
Output: The state reachability graph $R(ELPN)$ of $ELPN$
Step 0: Let S_0 be the root node of state reachability graph $R(ELPN)$, and tag it "new";
Step 1: While there exists a node tagged "new", do
Pick a random "new" node and mark it as S;
Step 2: If there exists a node marked S on the directed path from S_0 to S, then
Change the tag of S to "old" and go back to step 1;
Step 3: If $\forall t \in T : \neg S[t >$, then
Change the tag of S to "endpoint" and go back to step 1;
Step 4: For $\forall t \in T$ and $S[t >$, do the following for each t
Obtain S' based on transition firing rules and tag it "new" in $R(ELPN)$;
Draw a directed arc from S to S', and mark the arc with t and the transition firing time *Time*;
Remove the "new" tag of node S and go back to step 1.

With Algorithm 1, the state reachability graph of an ELPN can be constructed. A reachability graph describes the state shifting of the corresponding ELPN model, and the system's properties can be analyzed based on the reachability graph.

Definition 13: Reachability

Assume $ELPN=(LN, A, T_{arr}, V_p, G_p, B, T_{dur}, S_0)$. If there exists $t \in T$ that makes $S[t>S'$ then S' is immediately reachable from S. If there exist a transition sequence t_1, $t_2, \ldots\ldots, t_k$ and a label sequence $S_1, S_2, \ldots\ldots, S_k$ that make, $S[t_1>S_1[t_2>S_2 \ldots S_{k-1}[t_k>S_k$, then S_k is reachable from S. The set of all reachable states from S is denoted as $R(S)$. It is predefined that $S \in R(S)$.

Theorem 4: $\forall S_i, S_j \in R(S_0)$, S_j is reachable from S_i if and only if there exists a directed path from S_i to S_j in $R(ELPN)$.

Proof: Based on Definition 13, the conclusion is apparent.

The next section gives an example of ELPN-based e-commerce system modeling and property analysis.

IV. ANALYSIS OF ELPN-BASED E-COMMERCE SYSTEMS

A. BATCH ORDER WORKFLOW

In an e-commerce system, when a seller trades with multiple buyers at the same time, the buyers may have different priorities. The priorities of buyers are distinguished as follows in each workflow of the system. First, the buyers browse product information. Second, the buyers submit orders of products. Third, the system processes orders from the buyers. In this step, a system that does not have buyer priority defined will process all orders at the same time. However, if buyer priority is defined, then within each batch processing time period, the system will process orders from buyers of higher priorities (VIP customers) first and those from buyers of lower priorities (ordinary customers) are on hold. Fourth, based on order acceptance information, the system prepares goods and sends buyers products they ordered or refuses the order. Finally, each buyer pays for the products he/she bought after receiving the goods and the seller accepts the payments. Apparently, within one workflow, the system not only distinguishes the customers' priorities, but also trades with multiple buyers at the same time. Therefore, the data of one or more buyers are involved in each step of data processing, which means while priority is added, the data batch processing function of the system is also retained.

B. ELPN MODELS OF E-COMMERCE SYSTEMS

ELPNs can be used to describe and analyze the batch processing functions of prioritized systems. In the ELPN-based e-commerce system shown in Fig. 2, there are three types of transitions, including ordinary, logical input, and logical output transitions. In each type of them, batch wait time and prioritized transition firing rules are introduced.

Assume that there are three customers, B1-B3, browsing product information at different time and then shopping for products. In Fig. 2,

i_1, i_2, i_3 – customer B1, B2, or B3 is in the state of browsing product information;

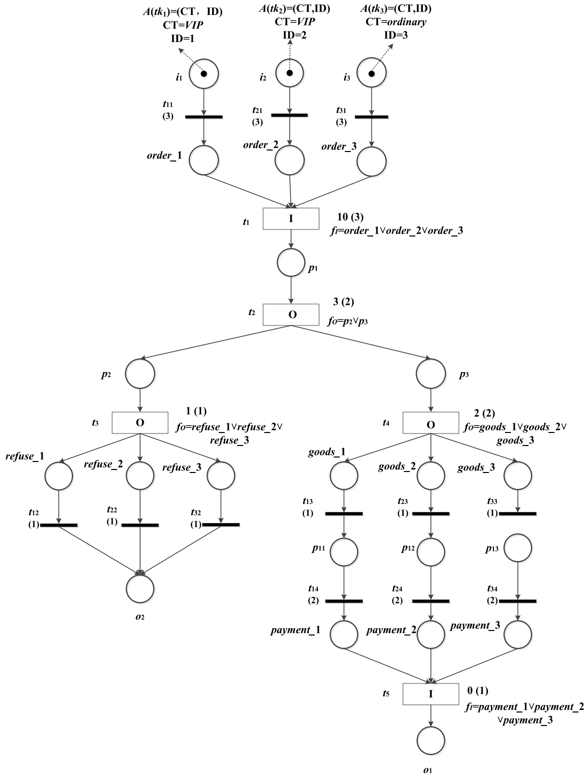

FIGURE 2. An ELPN model of an e-commerce system.

t_{11}, t_{12}, t_{13} – customer B1, B2, or B3 submits an order;

*order*_1, *order*_2, *order*_3 – the orders from B1, B2, or B3;

t_1 – the system processes orders;

p_1 – order processing result;

t_2 – the system sends out order processing result;

p_2 – the system is ready to notice customers about order canceling states;

p_3 – the system is ready to notice customers about product status;

t_3 – the system sends out order canceling commands;

t_{12}, t_{22}, t_{32} – customer B1, B2, or B3 cancels order;

o_2 – order canceling completion state;

t_4 – the system sends products to customer;

*goods*_1, *goods*_2, *goods*_3 – customer B1, B2, or B3 is in the state of receiving products;

*refuse*_1, *refuse*_2, *refuse*_3 – customer B1, B2, or B3 is in the state of receiving order canceling command;

t_{13}, t_{23}, t_{33} – customer B1, B2, or B3 checks product;

p_{11}, p_{12}, p_{13} – customer B1, B2, or B3 is in payment state;

t_{14}, t_{24}, t_{34} – customer B1, B2, or B3 performs payment operation;

*payment*_1, *payment*_2, and *payment*_3 – the payment information of B1, B2, or B3 is being transmitted;

t_5 – the system processes payment information;

o_1 –payment information is processed.

C. TIME CONSTRAINTS IN THE SYSTEM

In the ELPN model of the e-commerce system as shown in Fig. 2, the system has the following time constraints:

(1) Three time units is spent before a customer decides to submit an order after browsing product information;
(2) When the system processes orders, it first waits for corresponding batch wait time, and then determines order priorities based on customers' attributes and handles the orders. The batch wait time for order processing is 10 time units, and processing each order costs 3 time units. To process m orders, $3m$ time units are required;
(3) Sending out order processing result costs 2 time units and the batch wait time is 3 time units;
(4) Sending an order processing result information costs 2 time units and the batch wait time is 2 time units;
(5) Noticing a customer about order canceling information costs 1 time unit and the batch wait time is 1 time unit;
(6) Processing payment information costs 1 time unit and the batch wait time is 0 time unit;
(7) Canceling an order costs 1 time unit;
(8) Checking goods information costs 1 time unit;
(9) Performing payment operation costs 2 time units.

D. PROPERTY ANALYSIS OF THE E-COMMERCE SYSTEM

1) STATE REACHABILITY GRAPH OF THE SYSTEM

We assume that the e-commerce system sends price quotes to three customers B1-B3 at time point T_0, and the three customers' attributes are $A_{i1}(tk_1)=A_{i2}(tk2)=A_{i3}(tk_3)=(\text{CT, ID})$ $v_{i1}(tk_1)=(\text{VIP, 1})$, $v_{i2}(tk_2)=(\text{VIP, 2})$, and $v_{i3}(tk_3)=(\text{ordinary, 3})$. At time points $a_{tk1}=2$, $a_{tk2}=6$, and $a_{tk3}=3$, the three customers accept the price quotes, respectively. Since it costs 3 time units for the system to transmit order information, the order submission completion time is $a_{tk4}=5$, $a_{tk5}=9$, and $a_{tk6}=6$ for the three customers; the time points represent the time that new tokens arrive at post-places order_1, order_2, and order_3 after transitions t_{11}, t_{21}, and t_{31} fire. Therefore, $T_{early}(t_1) = \min\{a_{tk4}, a_{tk5}, a_{tk5}\}=5$. In the order processing stage of the system, the batch wait time period of t_1 is $(T_{early}(t_1), T_{early}(t_1) + B(t_1))$. As seen in Fig. 2, $B(t_1)=10$, thus $(T_{early}(t_1), T_{early}(t_1)+B(t_1))=(5, 15)$. Therefore, at time point 15, the system should be processing the orders from B1-B3. According to the transition firing rules in Definition 12, since customer B3 is an ordinary customer, its order is processed only after the system finished processing the orders of VIP customers B1 and B2. In the processing of other transitions, this rule is also followed. In addition, the system also obeys the time constraints in Section 4.3.

Fig. 3 presents the system's state reachability graph and Tables 1 to 9 present information of each state of the state reachability graph during a customer's shopping process. Each state consists of 23 components, representing the token information in places i_1, i_2, i_3, *order*_1, *order*_2, *order*_3,

TABLE 1. States from order submission to ready to send order processing result.

S_0	[(1,(2;VIP:ID=1)),0,(1,(3;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_1	[0,0,(1,(3;ordinary:ID=3)),(1,(5;VIP:ID=1)),0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_2	[0,(1,(6;VIP:ID=2)),0,(1,(5;VIP:ID=1)),0,(1,(6;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_3	[0,0,0,(1,(5;VIP:ID=1)),(1,(9;VIP:ID=2)),(1,(6;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_4	[0,0,0,0,0,(1,(6;ordinary:ID=3)),(2,(21;VIP:ID=1),(21;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_5	[0,0,0,0,0,0,(3,(21;VIP:ID=1),(21;VIP:ID=2),(24;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

TABLE 2. States after refusing the order from customer VIP1 because of goods shortage.

S_6	[0,0,0,0,0,0,(1,(24;ordinary:ID=3)),(1,(28;VIP:ID=1)),(1,(28;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0,0,0]
S_7	[0,0,0,0,0,0,0,0,(2,(28;VIP:ID=2),(30;ordinary:ID=3)),(1,(30;VIP:ID=1)),0,0,0,0,0,0,0,0,0,0,0,0]
S_8	[0,0,0,0,0,0,0,0,(2,(28;VIP:ID=2),(30;ordinary:ID=3)),0,0,0,(1,(31;VIP:ID=1)),0,0,0,0,0,0,0,0,0]
S_9	[0,0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),0,0,0,(1,(31;VIP:ID=1)),0,(1,(32;VIP:ID=2)),0,0,0,0,0,0,0]
S_{10}	[0,0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),0,0,0,(1,(31;VIP:ID=1)),0,0,0,0,(1,33;VIP:ID=2),0,0,0,0,0]
S_{11}	[0,0,0,0,0,0,0,0,0,0,0,(1,(31;VIP:ID=1)),0,0,(1,(34;ordinary:ID=3)),0,(1,(33;VIP:ID=2)),0,0,0,0,0]
S_{12}	[0,0,0,0,0,0,0,0,0,0,0,(1,(31;VIP:ID=1)),0,0,0,0,0,(1,(35;ordinary:ID=3)),0,(1,(35;VIP:ID=2)),0,0]
S_{13}	[0,0,0,0,0,0,0,0,0,0,0,(1,(31;VIP:ID=1)),0,0,0,0,0,(1,(35;ordinary:ID=3)),0,0,0,(1,(36;VIP:ID=2))]
S_{14}	[0,0,0,0,0,0,0,0,0,0,0,(1,(31;VIP:ID=1)),0,0,0,0,0,0,0,0,(1,(37;ordinary:ID=3)),(1,(36;VIP:ID=2))]
S_{15}	[0,0,0,0,0,0,0,0,0,0,0,(1,(31;VIP:ID=1)),0,0,0,0,0,0,0,0,0,(2,(36;VIP:ID=2),(38;ordinary:ID=3))]

p_1, p_2, p_3, $refuse\_1$, $refuse\_2$, $refuse\_3$, o_2, $goods\_1$, $goods\_2$, $goods\_3$, p_{11}, p_{12}, p_{13}, $payment\_1$, $payment\_2$, $payment\_3$, and o_1, respectively.

It is clearly seen in Fig. 3 that $\forall S_i$, either an intermediate state or a correct end-state, there exists a transition sequence that forms a directed path from S_0 to S_i, i.e. state S_i reachable from S_0.

2) TIME COST SUPERIORITY

As can be seen from markings S_{15}, S_{25}, S_{48}, S_{55}, and S_{68}, an order from a VIP customer takes less processing time compared to that of an ordinary customer. Taking the processing of all types of customer requests as an example, the time needed by users with higher priorities in an e-commerce system modeled with ELPN is analyzed below.

As seen in Table 9 and Fig. 3, in state S_{68}, the token information in place o_1 is (3, (39; VIP:ID = 1), (39; VIP:ID = 2), (40; ordinary:ID = 3)), thus the order completion time points for three customers are 39, 39, and 40, respectively. If customer priority is not considered, the system will process all orders received in the batch wait time period at the same time without considering the customers' different attributes. Thus the short time for VIP customers is not reflected. In the e-commerce system shown in Fig. 2, the orders from customers B1, B2, and B3 are all in the batch wait time period. However, due to the difference in customer priorities, the system first processes the orders from customers B1 and B2 as VIP customers, and then processes the order from ordinary customer B3. This way, the order processing for two VIP customers is completed at time point 39, and that for the ordinary customer is completed at time point 40, thereby reflecting the time saving for VIP customers.

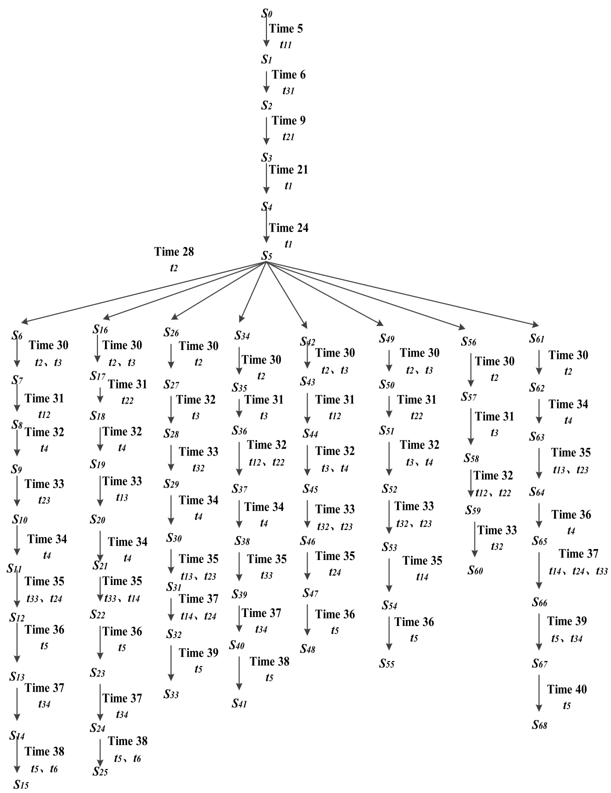

FIGURE 3. The state reachability graph of the e-commerce system.

3) FAIRNESS WHEN PROCESSING USER REQUESTS

When the system refuses all customer requests because of goods shortage, as seen from state S_{57}, t_3 is enabled at time point 30 as the system sends out order canceling commands. Since place p_2 contains three tokens of different

TABLE 3. States after refusing the order from customer VIP2 because of goods shortage.

S_{16}	[0,0,0,0,0,0,(1,(24;ordinary:ID=3)),(1,(28;VIP:ID=2)),(1,(28;VIP:ID=1)),0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{17}	[0,0,0,0,0,0,0,0,(2,(28;VIP:ID=1),(30;ordinary:ID=3)),(1,(30;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{18}	[0,0,0,0,0,0,0,0,(2,(28;VIP:ID=1),(30;ordinary:ID=3)),0,0,0,(1,(31;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0]
S_{19}	[0,0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),0,0,0,(1,(31;VIP:ID=2)),0,(1,(32;VIP:ID=1)),0,0,0,0,0,0,0,0]
S_{20}	[0,0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),0,0,0,(1,(31;VIP:ID=2)),0,0,0,0,(1,33;VIP:ID=1),0,0,0,0,0]
S_{21}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(1,(31;VIP:ID=2)),0,0,(1,(34;ordinary:ID=3)),0,(1,(33;VIP:ID=1)),0,0,0,0,0]
S_{22}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(1,(31;VIP:ID=2)),0,0,0,0,0,(1,(35;ordinary:ID=3)),0,(1,(35;VIP:ID=1)),0,0]
S_{23}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(1,(31;VIP:ID=2)),0,0,0,0,0,(1,(35;ordinary:ID=3)),0,0,0,(1,(36;VIP:ID=1))]
S_{24}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(1,(31;VIP:ID=2)),0,0,0,0,0,0,0,0,(1,(37;ordinary:ID=3)),(1,(36;VIP:ID=1))]
S_{25}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(1,(31;VIP:ID=2)),0,0,0,0,0,0,0,0,0,(2,(36;VIP:ID=1),(38;ordinary:ID=3))]

TABLE 4. States after refusing the order from the ordinary customer because of goods shortage.

S_{26}	[0,0,0,0,0,0,(1,(24;ordinary:ID=3)),0,(2,(28;VIP:ID=1),(28;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{27}	[0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),(2,(28;VIP:ID=1),(28;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{28}	[0,0,0,0,0,0,0,0,(2,(28;VIP:ID=1),(28;VIP:ID=2)),0,0,(1,(32;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,0,0]
S_{29}	[0,0,0,0,0,0,0,0,(2,(28;VIP:ID=1),(28;VIP:ID=2)),0,0,0,(1,(33;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,0]
S_{30}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(1,(33;ordinary:ID=3)),(1,(34;VIP:ID=1)),(1,(34;VIP:ID=2)),0,0,0,0,0,0,0,0]
S_{31}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(1,(33;ordinary:ID=3)),0,0,0,(1,(35;VIP:ID=1)),(1,(35;VIP:ID=2)),0,0,0,0,0]
S_{32}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(1,(33;ordinary:ID=3)),0,0,0,0,0,0,(1,(37;VIP:ID=1)),(1,(37;VIP:ID=2)),0,0]
S_{33}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(1,(33;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,(2,(39;VIP:ID=1),(39;VIP:ID=2))]

TABLE 5. States after refusing the orders from customers VIP1 and VIP2 because of goods shortage.

S_{34}	[0,0,0,0,0,0,(1,(24;ordinary:ID=3)),(2,(28;VIP:ID=1),(28;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{35}	[0,0,0,0,0,0,0,(2,(28;VIP:ID=1),(28;VIP:ID=2)),(1,(30;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{36}	[0,0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),(1,(31;VIP:ID=1)),(1,(31;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0,0]
S_{37}	[0,0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),0,0,0,(2,(32;VIP:ID=1),(32;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0]
S_{38}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(2,(32;VIP:ID=1),(32;VIP:ID=2)),0,0,(1,(34;ordinary:ID=3)),0,0,0,0,0,0,0]
S_{39}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(2,(32;VIP:ID=1),(32;VIP:ID=2)),0,0,0,0,0,(1,(35;ordinary:ID=3)),0,0,0,0]
S_{40}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(2,(32;VIP:ID=1),(32;VIP:ID=2)),0,0,0,0,0,0,0,0,(1,(37;ordinary:ID=3)),0]
S_{41}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(2,(32;VIP:ID=1),(32;VIP:ID=2)),0,0,0,0,0,0,0,0,0,(1,(38;ordinary:ID=3))]

TABLE 6. States after refusing the orders from customer VIP1 and the ordinary customer because of goods shortage.

S_{42}	[0,0,0,0,0,0,(1,(24;ordinary:ID=3)),(1,(28;VIP:ID=1)),(1,(28;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{43}	[0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),(1,(28;VIP:ID=2)),(1,(30;VIP:ID=1)),0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{44}	[0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),(1,(28;VIP:ID=2)),0,0,0,(1,(31;VIP:ID=1)),0,0,0,0,0,0,0,0,0,0]
S_{45}	[0,0,0,0,0,0,0,0,0,0,0,0,(1,(32;ordinary:ID=3)),(1,(31;VIP:ID=1)),0,(1,(32;VIP:ID=2)),0,0,0,0,0,0,0,0]
S_{46}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(2,(31;VIP:ID=1),(33;ordinary:ID=3)),0,0,0,0,(1,(33;VIP:ID=2)),0,0,0,0,0]
S_{47}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(2,(31;VIP:ID=1),(33;ordinary:ID=3)),0,0,0,0,0,0,0,(1,(35;VIP:ID=2)),0,0]
S_{48}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(2,(31;VIP:ID=1),(33;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,(1,(36;VIP:ID=2))]

TABLE 7. States after refusing the orders from customer VIP2 and the ordinary customer because of goods shortage.

S_{49}	[0,0,0,0,0,0,(1,(24;ordinary:ID=3)),(1,(28;VIP:ID=2)),(1,(28;VIP:ID=1)),0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{50}	[0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),(1,(28;VIP:ID=1)),(1,(30;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{51}	[0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),(1,(28;VIP:ID=1)),0,0,0,(1,(31;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0]
S_{52}	[0,0,0,0,0,0,0,0,0,0,0,0,(1,(32;ordinary:ID=3)),(1,(31;VIP:ID=2)),0,(1,(32;VIP:ID=1)),0,0,0,0,0,0,0,0]
S_{53}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(2,(31;VIP:ID=2),(33;ordinary:ID=3)),0,0,0,0,(1,(33;VIP:ID=1)),0,0,0,0,0]
S_{54}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(2,(31;VIP:ID=2),(33;ordinary:ID=3)),0,0,0,0,0,0,0,(1,(35;VIP:ID=1)),0,0]
S_{55}	[0,0,0,0,0,0,0,0,0,0,0,0,0,(2,(31;VIP:ID=2),(33;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,(1,(36;VIP:ID=1))]

attribute values, i.e. (3, (28; VIP:ID = 1), (28; VIP:ID = 2), (30; ordinary:ID = 3)), and the arrival times of the three customers orders are 28, 28, and 30, respectively, thus $T_{early}(t_3) = 28$. As seen in Fig. 2, the batch wait time period of t_3 is $BW(t_3) = 1$. Thus the batch processing time period is $= (T_{early}(t_3), T_{early}(t_3) + BW(t_3)) = (28, 29)$. Since the order of customer B3 has not arrived within this time period, the order has to wait to be processed. However, the order cannot wait infinitely; this raises the problem of system fairness when processing user requests.

TABLE 8. States after refusing the orders from all three customers because of goods shortage.

S_{56}	[0,0,0,0,0,0,(1,(24;ordinary:ID=3)),(2,(28;VIP:ID=1),(28;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{57}	[0,0,0,0,0,0,0,(3,(28;VIP:ID=1),(28;VIP:ID=2),(30;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{58}	[0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),0,(1,(31;VIP:ID=1)),(1,(31;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{59}	[0,0,0,0,0,0,0,0,0,0,0,(1,(32;ordinary:ID=3)),(2,(32;VIP:ID=1),(32;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0]
S_{60}	[0,0,0,0,0,0,0,0,0,0,0,0,(3,(32;VIP:ID=1),(32;VIP:ID=2),(33;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,0,0]

TABLE 9. States after processing all customer requests.

S_{61}	[0,0,0,0,0,0,(1,(24;ordinary:ID=3)),0,(2,(28;VIP:ID=1),(28;VIP:ID=2)),0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{62}	[0,0,0,0,0,0,0,0,(3,(28;VIP:ID=1),(28;VIP:ID=2),(30;ordinary:ID=3)),0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
S_{63}	[0,0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),0,0,0,0,(1,(34;VIP:ID=1)),(1,(34;VIP:ID=2)),0,0,0,0,0,0,0,0]
S_{64}	[0,0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),0,0,0,0,0,0,0,(1,(35;VIP:ID=1)),(1,(35;VIP:ID=2)),0,0,0,0,0]
S_{65}	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),(1,(35;VIP:ID=1)),(1,(35;VIP:ID=2)),0,0,0,0,0]
S_{66}	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,(1,(30;ordinary:ID=3)),(1,(37;VIP:ID=1)),(1,(37;VIP:ID=2)),0,0]
S_{67}	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,(1,(39;ordinary:ID=3)),(2,(39;VIP:ID=1),(39;VIP:ID=2))]
S_{68}	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,(3,(39;VIP:ID=1),(39;VIP:ID=2),(40;ordinary:ID=3))]

When a transition t fires, if there are still tokens in its pre-places, then the next batch processing time period starts immediately (the start time of batch processing time period is the arrival time of the earliest arrived token among current tokens in the pre-places). After the system processes the last batch of tokens, if it already passes the batch end time of the next batch, the transition deals with the next batch of tokens immediately; if the next batch wait time has not ended, then the transition waits until the batch wait time period ends to process the tokens in the batch. As seen in Figs. 2 and 3, when t_3 fires between time points 29 and 31, there is still a token with attribute value of ordinary in its pre-places, thus it calculates the next batch processing time period immediately. At this time, the arrival time of the earliest arrived token in the pre-places of t_3 is $T_{early}(t_3) = 30$. Thus the next batch processing time period is $(T_{early}(t_3), T_{early}(t_3) + TC(t_3)) = (30, 31)$, i.e. the batch end time is 31. The batch wait time period ends at time point 31 when t_3 fires again to process the request from the customer with attribute value of ordinary. It takes 1 time unit for transition t_3 to process a token. At last, all requests from VIP and ordinary customers are processed. This way, the fairness of the system when processing user requests is ensured.

4) CORRECT END-STATES

By analyzing the operation of the ELPN model of an e-commerce system, it is easy to see that after multiple transactions of the system, a customer order always reaches a correct end-state, i.e. place o_1 or o_2. Then, the order processing service of the system ends correctly without any deadlock. This feature is also reflected in the state reachability graph.

After multiple transactions, the system's state S always reaches one of the following three states (i.e. all orders are processed correctly):

(1) $[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, (m, (A_{11}, y_{11}), (A_{12}, y_{12}), \ldots\ldots, (A_{1m}, y_{1m})), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$

(2) $[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, (m, (A_{21}, y_{21}), (A_{22}, y_{22}), \ldots\ldots, (A_{2m}, y_{2m}))]$

(3) $[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, (a, (A_{11}, y_{11}), (A_{12}, y_{12}), \ldots\ldots, (A_{1a}, y_{1a}), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, (b, (A_{21}, y_{21}), (A_{22}, y_{22}), \ldots\ldots, (A_{2b}, y_{2b}))]$

The three states above are correct end-states of the system's order processing service.

V. CONCLUSIONS

This paper introduces the concept, firing rules, and state reachability graph construction algorithm of ELPNs. An ELPN for an e-commerce system is established. The system's state reachability graph is constructed, the user priority handling is explained. The system's handling of customers with attribute values of VIP and ordinary customers is described, and the time needed by VIP customers is analyzed. In addition, the fairness of the system is analyzed, and the correct end-states are defined and analyzed. We will explore how to handle large-scale systems with stochastic characteristics in our future research.

REFERENCES

[1] X. Lu, M. Zhou, A. C. Ammari, and J. Ji, "Hybrid Petri nets for modeling and analysis of microgrid systems," *IEEE/CAA J. Autom. Sin.*, vol. 3, no. 4, pp. 347–354, Oct. 2016.

[2] H. X. Li and Y. Y. Du, "A survey of research issues and key technology for business process management," *J. Shandong Univ. Sci. Technol.*, vol. 34, no. 1, pp. 22–28, 2015.

[3] Y. H. Tian and Y. Y. Du, "A grouping algorithm of optimal alignments," *J. Shandong Univ. Sci. Technol.*, vol. 34, no. 1, pp. 29–34, 2015.

[4] S. G. Wang, D. You, and M. C. Zhou, "A necessary and sufficient condition for a resource subset to generate a strict minimal siphon in S^4PR," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4173–4179, Aug. 2017, doi: 10.1109/TAC.2017.2677859.

[5] Y. Y. Yin, W. Xu, Y. Xu, H. Li, and L. Yu, "Collaborative QoS prediction for mobile service with data filtering and SlopeOne model," *Mobile Inform. Syst.*, vol. 2017, Art. no. 7356213, 2007. [Online]. Available: https://www.hindawi.com/journals/misy/2017/7356213/, doi: 10.1155/2017/7356213.

[6] W. Liu, L. Wang, Y. Y. Du, and M. Li, "Deadlock property analysis of concurrent programs based on Petri net structure," *Int. J. Parallel Programm.*, vol. 45, no. 4, pp. 879–898, 2017.

[7] S. C. Pang *et al.*, "A model for dynamic business processes and process changes," *Chin. J. Electron.*, vol. 20, no. 4, pp. 632–636, 2011.

[8] Z. J. Ding, J. L. Wang, and C. J. Jiang, "An approach for synthesis Petri nets for modeling and verifying composite Web service," *J. Inf. Sci. Eng.*, vol. 24, no. 5, pp. 1309–1328, 2008.

[9] Y. S. Huang, Y. S. Weng, and M. C. Zhou, "Modular design of urban traffic-light control systems based on synchronized timed Petri nets," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 530–539, Apr. 2014.

[10] J. Wang, Y. Y. Du, and S. X. Yu, "Coloured logic Petri nets and analysis of their reachable trees," *Enterprise Inf. Syst.*, vol. 9, no. 8, pp. 900–919, 2015.

[11] Y. Y. Du and B. Q. Guo, "Logic Petri nets and equivalency," *Inf. Technol. J.*, vol. 8, no. 1, pp. 95–100, 2009.

[12] Y. Y. Du and Y. H. Ning, "Property analysis of logic Petri nets by marking reachability graphs," *Frontiers Comput. Sci.*, vol. 8, no. 4, pp. 684–692, 2014.

[13] Y. Y. Du, L. Qi, and M. C. Zhou, "A vector matching method for analysing logic Petri nets," *Enterprise Inf. Syst.*, vol. 5, no. 4, pp. 449–468, 2011.

[14] W. Liu, Y. Y. Du, and C. Yan, "Soundness preservation in composed logical time workflow nets," *Enterprise Inf. Syst.*, vol. 6, no. 1, pp. 95–113, 2012.

[15] W. Liu, Y. Y. Du, M. C. Zhou, and C. Yan, "Transformation of logical workflow nets," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 10, pp. 1401–1412, Oct. 2014.

[16] W. Liu, Y. Y. Du, and C. Yan, "Properties analysis of substitution of logical workflow nets," *J. Comput. Inf. Syst.*, vol. 10, no. 6, pp. 2491–2498, 2014.

[17] Y. B. Han, J. Wang, and P. Zhang, "Business-oriented service modeling: A case study," *Simul. Model. Pract. Theory*, vol. 17, no. 8, pp. 1413–1429, 2009.

[18] Q. Hu, Y. Y. Du, and S. X. Yu, "Service net algebra based on logic Petri nets," *Inf. Sci.*, vol. 268, no. 6, pp. 271–289, 2014.

[19] C. Yan, H. T. Sun, and W. Liu, "Study of fuzzy association rules and cross-selling toward property insurance customers based on FARMA," *J. Intell. Fuzzy Syst.*, vol. 31, no. 6, pp. 2789–2794, 2016.

[20] Y. Q. Li, C. Yan, W. Liu, and M. Z. Li, "A principle component analysis-based random forest with the potential nearest neighbor method for automobile insurance fraud identification," *Appl. Soft Comput.*, to be published. [Online]. Available: https://doi.org/10.1016/j.asoc.2017.07.027

[21] Y. Y. Du, C. J. Jiang, M. C. Zhou, and Y. Fu, "Modeling and monitoring of E-commerce workflows," *Inf. Sci.*, vol. 179, no. 7, pp. 995–1006, 2009.

[22] M. C. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*. Singapore: World Scientific, 1998.

[23] J. Bi, H. Yuan, and M. Zhou, "A Petri net method for compatibility enforcement to support service choreography," *IEEE Access*, vol. 4, pp. 8581–8592, Oct. 2016.

WEI LIU received the B.S., M.S., and Ph.D. degrees from the Shandong University of Science and Technology, Qingdao, China, in 1999, 2005, and 2012, respectively. He is currently an Associate Professor with the Shandong University of Science and Technology. He has led or participated in many projects supported by the National Natural Science Foundation, the National Key Basic Research Development Program, and other important and key projects at provincial levels. He has authored over 30 papers in journals and conference proceedings, such as the IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS: SYSTEMS and *Enterprise Information Systems*. His research interests are in formal engineering methods, Petri nets, workflows, and web services. He was the recipient of science and technology advancement awards at the province level and the Excellent Doctoral Dissertation Award from Shandong province, China.

PIN WANG received the B.S. degree from the Shandong University of Science and Technology, Qingdao, China, in 2015, where she is currently pursuing the M.S. degree. Her current research interests are Petri nets and workflow.

YUYUE DU received the B.S. degree from Shandong University, Jinan, China, in 1982, the M.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1991, and the Ph.D. degree in computer application from Tongji University, Shanghai, China, in 2003. He is currently a Professor with the Shandong University of Science and Technology, Qingdao, China. His research interests are in formal engineering methods, Petri nets, real-time systems, web services, and workflows. He has led or participated in over 15 projects supported by the National Natural Science Foundation, National Key Basic Research Development Program, and other important and key projects at provincial levels. He is a member of the Professional Committee of Petri Nets of the China Computer Federation. He has authored over 170 papers in journals and conference proceedings, such as the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART A, the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART C, *Information Sciences*, and *Enterprise Information Systems*. He was a recipient of two prizes of Science and Technology Advancement Awards at the province level, and the Excellent Doctoral Dissertation Award from Shanghai, China.

MENGCHU ZHOU (S'88–M'90–SM'93–F'03) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined the New Jersey Institute of Technology, Newark, NJ, USA, in 1990, where he is currently a Distinguished Professor of electrical and computer engineering and the Director of the Discrete-Event Systems Laboratory. He has over 700 publications, including 12 books, over 390 journal papers (over 280 in the IEEE transactions), and 28 book-chapters. He has 11 patents and several pending ones. His recently co-authored/edited books include *Business and Scientific Workflows: A Web Service-Oriented Approach* (IEEE/Wiley, New Jersey, 2013) (with W. Tan) and *Contemporary Issues in Systems Science and Engineering* (IEEE/Wiley, New Jersey, 2015) (with H.-X. Li and M. Weijnen). His research interests are in intelligent automation, Petri nets, Internet of Things, Web service, workflow, big data, transportation, and energy systems.

He is a Life Member of the Chinese Association for Science and Technology, USA and served as its President in 1999. He is a fellow of the International Federation of Automatic Control and the American Association for the Advancement of Science. He was a recipient of the NSF's Research Initiation Award, the CIM University-LEAD Award from the Society of Manufacturing Engineers, the Perlis Research Award and the Fenster Innovation in Engineering Education Award from NJIT, the Humboldt Research Award for U.S. Senior Scientists from the Humboldt Foundation, the Leadership Award and the Academic Achievement Award from Chinese Association for Science and Technology, USA, the Asian American Achievement Award from the Asian American Heritage Council of New Jersey, the Outstanding Contributions Award, the Distinguished Lecturership, Franklin V. Taylor Memorial Award, the Norbert Wiener Award from the IEEE SMC Society, and the Distinguished Service Award from the IEEE Robotics and Automation Society. He was the General Chair of the IEEE Conference on Automation Science and Engineering, Washington, DC, USA, in 2008, the General Co-Chair of the 2003 IEEE International Conference on System, Man and Cybernetics, Washington, DC, USA, in 2003, the Founding General Co-Chair of the 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, in 2004, and the General Chair of the 2006 IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, USA, in 2006. He was the Program Chair of the 2010 IEEE International Conference on Mechatronics and Automation, Xi'an, China, in 2010, the 1998 and 2001 IEEE International Conference on SMC, and the 1997 IEEE International Conference on Emerging Technologies and Factory Automation. He organized and chaired over 100 technical sessions and served on program committees for many conferences. He is founding Co-Chair of the Enterprise Information Systems Technical Committee (TC) and Environmental Sensing, Networking, and Decision-making TC of the IEEE SMC Society. He was invited to lecture in Australia, Canada, China, France, Germany, Hong Kong, Italy, Japan, Korea, Mexico, Saudi Arabia, Singapore, Taiwan, and USA, and served as a plenary/keynote speaker for many conferences. He is the founding Editor of the IEEE Press book series on systems science and engineering. He served as an Associate Editor of the IEEE Transactions on Robotics and Automation, the IEEE Transactions on Automation Science and Engineering, the IEEE Transactions on Systems, Man and Cybernetics: Systems, and the IEEE Transactions on Industrial Informatics, and the Editor of the IEEE Transactions on Automation Science and Engineering. He is Associate Editor of the IEEE Transactions on Intelligent Transportation Systems, the IEEE/CAA Journal of Automatica Sinica, and *Frontiers of Information Technology & Electronic Engineering*. He served as a guest editor for many journals, including the IEEE Internet of Things Journal, the IEEE Transactions on Industrial Electronics, and the IEEE Transactions on Semiconductor Manufacturing. He has led or participated in over 50 research and education projects with total budget over U.S. 12 million, funded by the National Science Foundation, Department of Defense, NIST, New Jersey Science and Technology Commission, and Industry. He has been among most highly cited scholars for years and ranked top one in the field of engineering worldwide in 2012 by Web of Science/Thomson Reuters.

CHUN YAN received the B.S., M.S., and Ph.D. degrees from the Shandong University of Science and Technology, Qingdao, China, in 2000, 2003, and 2011, respectively. She is currently an Associate Professor of the Shandong University of Science and Technology. Her research interests include applied mathematics, statistics, and economic management. She has about 20 technical papers published in journals and conference proceedings in his research areas.