Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Petri net-based methods for analyzing structural security in e-commerce business processes

Wangyang Yu[a,b,c], Zhijun Ding[d], Lu Liu[e,*], Xiaoming Wang[a,b,c,*], Richard David Crossley[e]

[a] Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an 710062, China
[b] Engineering Laboratory of Teaching Information Technology of Shaanxi Province, Xi'an 710119, China
[c] School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
[d] Department of Computer Science, Tongji University, Shanghai 201804, China
[e] Department of Electronics, Computing and Mathematics, University of Derby, Derby, DE221 GB, UK

HIGHLIGHTS

- A framework for analyzing structural security in e-commerce business process.
- A method of modeling control and data structures for constructing an e-commerce business process.
- Petri net-based modeling and analysis methods.

ARTICLE INFO

Article history:
Received 18 October 2017
Received in revised form 30 January 2018
Accepted 25 April 2018
Available online xxxx

Keywords:
Petri net
E-commerce
Business process
Structural security

ABSTRACT

The rapid development of e-commerce worldwide, means more e-commerce business processes adopting the structure of multiple participants; these include shopper clients, merchant and third-party payment platforms (TPPs), banks, and so on. It is a distributed and complex system, where communications among these participants rely on the web services and Application Programming Interfaces (APIs) such as Cashier-as-a-Service or CaaS. This introduces new security challenges due to complex interactions among multiple participants, and any design flaws in procedure structures may result in serious security issues. We study the structural security issues based on Petri nets, and a framework for analyzing structural security in e-commerce business process is proposed. Petri net-based modeling and analysis methods are also provided. Given the specifications of e-commerce business processes, the proposed methods can help designers analyze structural security issues of an e-commerce business process.

1. Introduction

E-commerce has significantly developed in recent years, and more and more business is conducted over the Internet. The daily volume of e-commerce is sizable and continues to grow at a rapid pace. Many e-commerce platforms spring up to accelerate this new industry [1,2]. E-commerce systems with multiple participants, including third-party payment platforms (TPPs), e-commerce systems, banks, clients, and other applications, have become the new frontier for conducting business. As a distributed application on the web, e-commerce business processes are more complex and loosely coupled. The participants communicate with each other through web services and APIs such as Cashier-as-a-Service or CaaS [3,4]. The business processes of different participants construct the entire process structure. This integration introduces new security challenges due to complex interactions among the APIs of multiple interactive participants. These differ from traditional security issues, and the new security challenges do not refer to virus, Trojans or security protocols [5,6]. The complex structural linkage of control and data flows in e-commerce systems may produce very serious problems including the violation of the transaction properties, and losses of user funds. These issues can be defined as structural security. There are many structural security cases that have appeared over recent years. These include the vulnerability caused by a combination of open source online shopping system and TPP [3], "one yuan gate" event of Taobao in 2011 [7], and mongodb-based web applications [8].

Business processes of e-commerce systems are the key, and it is difficult to correctly design the business structure. Modern businesses are inherently process-driven, and the security of business processes is increasingly important [9–12]. Most of the existing research around security business processes focuses on the security properties like Access Control and Confidential Information [9,10]

[*] Corresponding authors.
E-mail addresses: ywy191@snnu.edu.cn (W. Yu), L.Liu@derby.ac.uk (L. Liu), wangxm@snnu.edu.cn (X. Wang).

https://doi.org/10.1016/j.future.2018.04.090
0167-739X/© 2018 Elsevier B.V. All rights reserved.

in enterprise business processes. Other related studies refer to the process consistency in complex business processes [11,12]. These works are proposed to deal with the inconsistencies among business processes of different departments in a cross-organizational process. However, this research belongs to traditional security issues, and is insufficient to cope with structural security.

Petri nets are a suitable tool to illustrate true concurrency and model distributed systems, and Petri net-based approaches [13–16] have been presented to model and verify correctness and soundness of workflows. A series of works have been done on cooperative systems and inter-organizational workflow based on Petri nets [17–19], and other work has also been conducted on Petri net-based analysis and composition of web services [20–24]. These mainly focus on the soundness and correctness of workflow, cooperative systems, and composition of web services, but fail to consider structural security related to the financial security issues that are related to the funds of legal users.

In order to depict e-commerce business processes of multiple participants at both application-level and design-level and consider financial security issues, a formal model called an e-commerce Business Process Net (EBPN) is proposed [25,26]. Usage enables a designer to identify errors in the design process and correct them before the deployment phase. EBPNs are suitable for modeling and verifying e-commerce business processes, but their usage in structural security remains minimal. As part of our research, we use EBPN to model structural security issues in e-commerce business processes. First, we discuss the structural security, and propose a framework for analyzing methods. Then, Petri net-based modeling and analyzing methods are provided, including behavioral sequence and incidence matrix methods. Given the specifications of e-commerce business processes, the proposed methods can help designers analyze structural security issues of an e-commerce business process.

The remainder of this paper is organized as follows. Section 2 introduces the motivation example. Section 3 discusses the structural security issues. Section 4 presents the basic concepts. Section 5 describes how to model an e-commerce business process and structural properties using EBPN. Section 6 is the analyzing methods. Section 7 concludes this paper.

2. Motivation example

There is an actual e-commerce business process integrating Interspire and Google Checkout from [1]. For illustrating our method clearly, it is abridged, and we only focus on the most important functions; this is because it is a distributed and complex business process structure. The basic and important business process is shown in Fig. 1. Interspire utilizes several APIs to add/remove items in the shopping cart, which are aggregately denoted by API: Update Cart in the figure. The checkout process is triggered when the shopper clicks on the "Google Checkout" button. An important feature of this business process is that no order is generated before the payment is made: the shopper is supposed to pay for the content of his shopping cart first; only when the merchant is informed by the CaaS, the merchant will create an order of the transaction according to what is inside the cart and set its status to "PAID", as illustrated in Table 1. The problem here is that this procedure is not atomic: after receiving the message of Step 4, the shopper does not send the message of Step 5 immediately. Instead, he can still call API: Update Cart to change or add new items to his cart. Then, when the message of Step 5 is sent, the current cart in the shopper's session is more expensive than the cart field in Step 5. On the other hand, API Handle Payment loads the cart directly from the shopper's session, rather than from the CaaS, to build the order. This causes an inconsistency between what the CaaS sees in the cart at the paid time and what the merchant has at the checkout completion time, so the shopper can pay for a cheap item, but check out many expensive items [1].

Table 1
Key functionalities of the business process in Fig. 1.

TStore.com/handleIPN:
1: if (GetMsgField("status") $\neq$ PAID) exit; /*payment status*/
2: cart = LoadShoppingCart(GetMessageField("sessionID"));
3: order = CreateOrder(cart);
4: order.status = PAID;

This is a typical case of structural security. The structure of business process is the key, and integration of multiple participants introduces new security challenges due to complex interaction structures among Application Programming Interfaces (APIs). The wrong design of business structures would result in security accidents and the financial loss of legitimate users. Thus, one needs to exploit the methodologies to model and verify the structural security of online shopping business processes. Nevertheless, rigorous and formal methodologies for structural security issues remain largely open. The most pressing challenge is how to depict the business structure and structural security properties. In the following section, we discuss the related concepts of structural security.

3. Related concepts

Petri nets are a graphical language for modeling and validating concurrent and distributed systems, and allow true concurrency instead of an interleaving-based semantics. Petri nets provide an explicit representation of both states and events and can be understood easily by a graphical representation of modeled systems. They have well-defined formal semantics and a wide range of formal analytical methods. The basic concepts of Petri nets are summarized in [27–29]. In order to describe the electronic trading process better, EBPN extends them with some new functions. The following definitions are from [25] and [26], and more details can be seen in the two references.

Definition 1 (*EBPN*)**.** An E-commerce Business Process Net (EBPN) is a 7-tuple $EN = (P, T; F, D, W, S, G)$ where:
(1) P is a finite set of places;
(2) T is a finite set of transitions T such that $P \cap T = \varnothing$ and $P \cup T \neq \varnothing$;
(3) $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs;
(4) D is a finite, non-empty set of symbol strings denoting the types of tokens;
(5) $W : F \to \langle a_1d_1, a_2d_2, a_3d_3, \ldots, a_ld_l \rangle, a_l \in \{0, 1\}, d_l \in D$, and $l > 0$ is the number of elements in D;
(6) $S \subset D$ denotes a set of key token types; and
(7) $G : T \to \Pi$ is a predicate function that assigns a predicate to each transition $t \in T$ where Π is the set of Boolean expressions on D.

Definition 2 (*Marking of EBPN*)**.** A marking of an EBPN $EN = (P, T; F, D, W, S, G)$ is $M : P \to \langle n_1d_1, n_2d_2, n_3d_3, \ldots, n_ld_l \rangle$, $n_l \in \mathbb{N} = \{0, 1, 2, \ldots\}$; $d_l \in D$, and $l > 0$ is the number of data elements in D.

A marking M of an EBPN assigns k-dimensional vectors to places. The vector's component n_kd_k means that a place has n_k tokens belonging to type d_k. Here, a token is a trading parameter that belongs to some type in an EBPN.

Definition 3 (*Data State*)**.** A pair $= (M, \delta_D)$ is a data state of EN, if M is a marking of EN, and δ_D is called a data allocation which assigns a value **T** (true), or **F** (false) to each $d \in \{\widetilde{M}(p) \mid p \in P\}$ such that $d \in (D - S) \to \delta_D(d) = \mathbf{T}$, and $\delta_D : S \to \{\mathbf{T}, \mathbf{F}\}$.

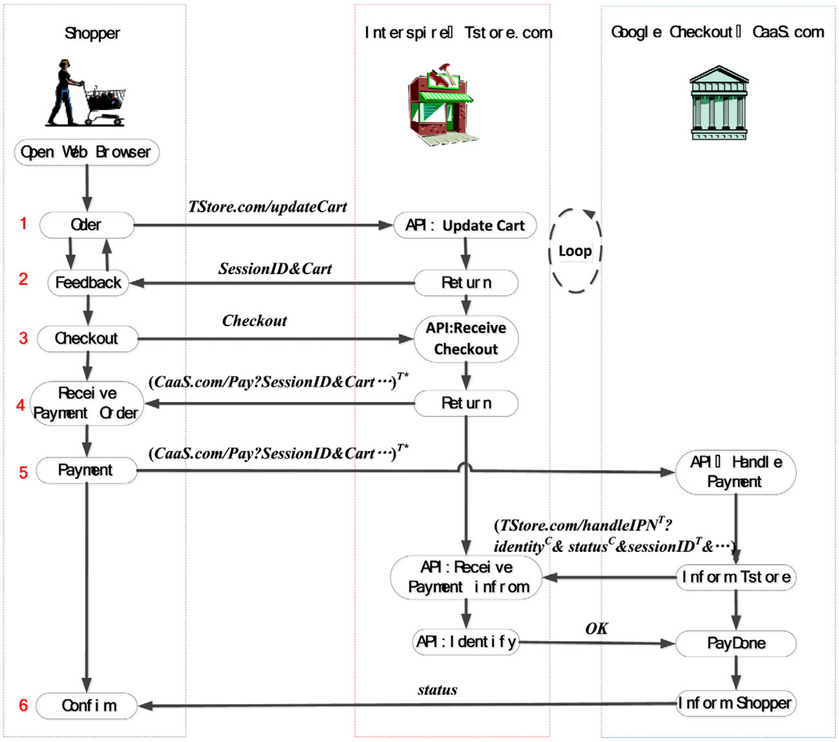


Fig. 1. Business process of integrating Interspire and Google Checkout.

Definition 4 (δ_G)**.** δ_G is a Boolean function that assigns a Boolean value **T** (true) or **F** (false) to each $G\,(t)$ such that δ_G: $G\,(t) \rightarrow$ {**T**, **F**}, $t \in T$.

Definition 5 (*Firing Conditions*)**.** A transition $t \in T$ is enabled at a data state $= (M, \delta_D)$ if

(1) $\forall\, p \in {}^\bullet\, t, M\,(p) \geq W\,(p, t)$; and
(2) $\exists\, G\,(t) \rightarrow \delta_G\,(G\,(t)) = \mathbf{T}$.

Definition 6 (*Key Transition*)**.** Given $t \in T, {}^\bullet\, t = P$, and $t\,^\bullet = P$, t is called a key transition if

(1) $S \cap \{\widetilde{W}(t, p) \mid p \in P''\} \neq \varnothing$; and
(2) The token $s \in S \cap \{\widetilde{W}(t, p) \mid p \in P''\}$ is produced by $t \rightarrow \delta_D(s) \in \{\mathbf{T}, \mathbf{F}\}$.

Definition 7 (*Firing Rules*)**.** Let $EN = (P, T; F, D, W, S, G)$ be an EBPN, and $= (M, \delta_D)$ be a data state of EN. A transition $t \in T$, which is enabled at (M, δ_D), can fire under M ($M \xrightarrow{t}$), and a new marking $M'(M \xrightarrow{t} M')$ is

$$M'(p) = \begin{cases} M(p) - W(p, t), & \text{if } p \in {}^\bullet t - t^\bullet \\ M(p) + W(t, p), & \text{if } p \in t^\bullet - {}^\bullet t \\ M(p) - W(p, t) + W(t, p), & \text{if } p \in {}^\bullet t \cap t^\bullet \\ M(p), & \text{otherwise.} \end{cases}$$

If t is not a key transition, a new data state is

$$\begin{aligned} \varLambda' &= (M', \delta_D{}') \\ &= (M', \forall d \in \{\widetilde{M}(p) \mid p \in P\} \rightarrow \delta_D{}'(d) = \delta_D(d) \\ &\quad \wedge \forall d \in \{\widetilde{W}(t, p) \mid p \in t^\bullet\} - \{\widetilde{M}(p) \mid p \in P\} \rightarrow \delta_D{}'(d) = \mathbf{T}). \end{aligned}$$

Else if t is a key transition, a new state set $\varGamma$ is

$$\begin{aligned} \varGamma = \{(M', \delta_D{}') \mid M \xrightarrow{t} M', \\ \forall\, s \in \{\widetilde{W}(t, p) \mid p \in t^\bullet\} \cap S \rightarrow \delta_D{}'(s) \in \{\mathbf{T},\ \mathbf{F}\}, \\ \forall\, d \in \{\widetilde{M}(p) - \{\widetilde{W}(t, p) \mid p \in t^\bullet\} \cap S\} \rightarrow \delta_D{}'(d) = \delta_D(d)\}. \end{aligned}$$

Definition 8 (*RD*)**.** Let (M_0, δ_{D0}) be the initial data state of $EN = (P, T; F, D, W, S, G)$. Its Reachability Data state graph (RD) can be defined as a 3-tuple $RD\,(EN) = (N, E; L)$, where

(1) N is a set of nodes, $N = R\,(M_0, \delta_{D0})$;
(2) E is a set of arcs, $E = \{(M_i, \delta_{Di}), (M_j, \delta_{Dj}) \mid (M_i, \delta_{Di}), (M_j, \delta_{Dj}) \in R\,(M_0, \delta_{D0}), \exists t_k \in T{:}\ (M_i, \delta_{Di}) \xrightarrow{t_k} (M_j, \delta_{Dj})\}$; and
(3) L: $E \rightarrow T$, $L\ ((M_i,\ \delta_{Di}),\ (M_j,\ \delta_{Dj})) = t_k$ if and only if $(M_i, \delta_{Di}) \xrightarrow{t_k} (M_j, \delta_{Dj})$, and t_k is called the label of the arc between (M_i, δ_{Di}) and (M_j, δ_{Dj}). (M_j, δ_{Dj}) is the successor (node) of (M_i, δ_{Di}), and (M_i, δ_{Di}) is the predecessor (node) of (M_j, δ_{Dj}).

Definition 9 (*Three-dimensional Incidence Matrix*)**.** Let $EN = (P, T; F, D, W, S, G)$ be an EBPN, $P = \{p_1, p_2, \ldots, p_m\}$, $T = \{t_1, t_2, \ldots, t_n\}$, $D = \{d_1, d_2, \ldots, d_l\}$, $m, n, l \in \mathbb{N}^+ = \{1, 2, \ldots\}$; then the static structure of EN can be expressed by a three-dimensional incidence matrix $\varPsi = [\psi_{ijk}]_{n \times m \times l}$, where $\psi_{ijk} = \psi_{ijk}{}^+ - \psi_{ijk}{}^-$

$$\psi_{ijk}{}^+ = \begin{cases} W(t_i, p_j)_k,\ \textbf{if}\,(t_i, p_j) \in F, i \in \{1, 2, \ldots, n\}, \\ \quad j \in \{1, 2, \ldots, m\}, k \in \{1, 2, \ldots, l\} \\ \textbf{an}\ l\textbf{-dimensional}\ \mathbf{0}\ \textbf{vector}\ \langle 0, 0, \ldots, 0\rangle,\ \textbf{otherwise} \end{cases}$$

$$\psi_{ijk}{}^- = \begin{cases} W(p_j, t_i)_k,\ \textbf{if}\,(p_j, t_i) \in F, i \in \{1, 2, \ldots, n\}, \\ \quad j \in \{1, 2, \ldots, m\}, k \in \{1, 2, \ldots, l\} \\ \textbf{an}\ l\textbf{-dimensional}\ \mathbf{0}\ \textbf{vector}\ \langle 0, 0, \ldots, 0\rangle,\ \textbf{otherwise}. \end{cases}$$

The above two equations can also be expressed as

$$\psi_{ij}{}^+ = \begin{cases} W(t_i, p_j),\ \textbf{if}\,(t_i, p_j) \in F, i \in \{1, 2, \ldots, n\}, \\ \quad j \in \{1, 2, \ldots, m\} \\ \textbf{an}\ l\textbf{-dimensional}\ \mathbf{0}\ \textbf{vector}\ \langle 0, 0, \ldots, 0\rangle,\ \textbf{otherwise} \end{cases}$$

$$\psi_{ij}{}^- = \begin{cases} W(p_j, t_i),\ \textbf{if}\,(p_j, t_i) \in F, i \in \{1, 2, \ldots, n\}, \\ \quad j \in \{1, 2, \ldots, m\} \\ \textbf{an}\ l\textbf{-dimensional}\ \mathbf{0}\ \textbf{vector}\ \langle 0, 0, \ldots, 0\rangle,\ \textbf{otherwise} \end{cases}$$

$$\psi_{ij} = \psi_{ij}{}^+ - \psi_{ij}{}^-.$$

In original Petri nets, two-dimensional incident matrix is defined by the relations of places and transitions. In EBPN, three-dimensional matrix is defined by the relations of places, transitions and W. In Definition 9, W (p_j, t_i) is an l -dimensional vector, and W $(p_j, t_i)_k$ is the kth scalar in W (p_j, t_i). ψ is a notation of the element in Ψ. $\psi_{ijk}{}^{+}$ and $\psi_{ijk}{}^{-}$ are scalars. $\psi_{ij}{}^{+}$ and $\psi_{ij}{}^{-}$ are l -dimensional vectors.

Lemma 1. *Let* $EN = (P, T; F, D, W, S, G)$ *be an EBPN,* (M_0, δ_{D0}) *be the initial data state,* Ψ *be the three-dimensional incidence matrix of* EN, *and* $(M, \delta_D) \in R(M_0, \delta_{D0})$. *Transition* $t_i \in T$ *is enabled at* M *if* $\forall p_j \in {}^{\bullet}t$ *and* $\forall\, k \in \mathbb{N}_l = \{1, 2, \ldots, l\} \rightarrow M\,(p_j)_k \geq \psi_{ijk}{}^{-}$, *i.e.,* $M\,(p_j) \geq \psi_{ij}{}^{-}$.

Definition 10 (Ψ^{T}). Let $EN = (P, T; F, D, W, S, G)$ be an EBPN, $\Psi = [\psi_{ijk}]_{n\times m\times l}$ is the three-dimensional incidence matrix. $\Sigma = \Psi^{\mathrm{T}}$ is the transpose of Ψ, i.e., $\varepsilon_{ij} = \psi_{ji}$, where ε is an element in Σ.

Lemma 2. *Let* $EN = (P, T; F, D, W, S, G)$ *be an EBPN,* (M_0, δ_{D0}) *be the initial data state,* $(M, \delta_D) \in R\,(M_0, \delta_{D0})$, *and* Ψ *be the three-dimensional incidence matrix of* EN. *If* $t_i \in T$, $M \xrightarrow{t_i} M'$, *then* $M' = M + (\Psi_{i*})^{\mathrm{T}}$.

Theorem 1. *Let* $EN = (P, T; F, D, W, S, G)$ *be an EBPN,* (M_0, δ_{D0}) *be the initial data state, and* Ψ *be the three-dimensional incidence matrix of* EN. *If* $(M, \delta_D) \in R\,(M_0, \delta_{D0})$, *then there exists an n-dimensional non-negative integer vector* V, *such that* $M = M_0 + \Psi^{\mathrm{T}} V$.

Corollary 1. *Let* $EN = (P, T; F, D, W, S, G)$ *be an EBPN,* (M_0, δ_{D0}) *be the initial data state, and* Ψ *be the three-dimensional incidence matrix of* EN. *There exist* $(M_0, \delta_{D0}) \xrightarrow{\sigma} (M, \delta_D)$ *and a n-dimensional non-negative integer vector* V, *such that* $M = M_0 + \Psi^{\mathrm{T}} V$. *Then* $\#(t_i, \sigma) = V\,[i]$, *where* $i \in \mathbb{N}_n = \{1, 2, \ldots, n\}$, $n = |T|$, *and* $\#(t_i, \sigma)$ *means the number of occurrences of* t_i *in* σ.

4. Structural security

Structural security issues are derived from the design of business structures. Hybrid web applications that combine multiple participants into integrated services like e-commerce websites have rapidly developed, and bring in new security concerns. The structural integration of multiple participants introduces new security challenges due to the complexity of an application to coordinate its internal states with those of the component services and web client across the Internet [3,4]. As the new security challenges of e-commerce business processes are at the application-level, structural security is beyond the capabilities of network-level and operating system-level security approaches. Even though the traditional security requirements are satisfied, there are still many structure and logic flaws at the design level of business processes [25,26].

Formal methods are mathematical techniques for specifying and verifying correctness and trustworthiness of software systems. Consequently, we use the formal model-EBPN to model e-commerce business processes and verify the structural security based on behavioral sequence and state analyzing methods. Fig. 2 shows the framework. Given the specification of an e-commerce business process, including development documents, UML diagrams, and function specifications, we can model the business process by define control and data structures using EBPN. Then, verification methods are proposed based on dynamic properties of EBPN. Two methods can be used to analyze structural security issues of an e-commerce business process. One is behavioral sequence method, and the other is incidence matrix method. The related definitions of structural security:-

Proposition 1. *Let* $\Re$ *be a structural specification of an e-commerce business process, and* $EN = (P, T; F, D, W, S, G)$ *is an EBPN corresponding to* $\Re$, (M_0, δ_{D0}) *be the initial data state;* γ *is an illegal behavior sequence derived from* $\Re$, *and* σ *is a behavior sequence corresponding to* γ. *If* $(M_0, \delta_{D0}) \xrightarrow{\sigma}$, *then we call EN does not satisfy structural security.*

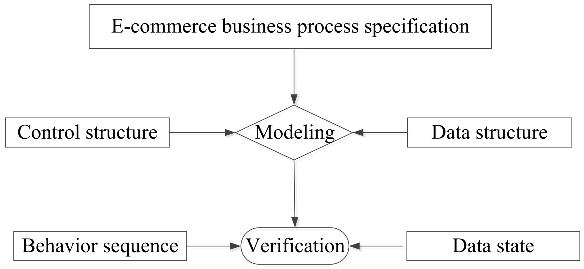


Fig. 2. Framework of this paper.

Proposition 2. *Let* $\Re$ *be a structural specification of an e-commerce business process, and* $EN = (P, T; F, D, W, S, G)$ *is an EBPN corresponding to* $\Re$, (M_0, δ_{D0}) *be the initial data state,* (M, δ_D) *is an illegal data state constructed according to* $\Re$; *If there is a behavioral sequence* σ *making that* $(M_0, \delta_{D0}) \xrightarrow{\sigma} (M, \delta_D)$, *then we call EN does not satisfy structural security.*

5. Modeling methods

Here we define control and data structures for constructing an e-commerce business process; these depict the situation that several APIs or operation events fire one after another. A completed model for integrating control and data structures from a global viewpoint is then obtained. The benefit of which is to provide different views of a composite business process, which helps designers or users understand and analyze the e-commerce business process.

Definition 11. Suppose that $EN_1 = (P_1, T_1; F_1, D_1, W_1, S_1, G_1)$ and $EN_2 = (P_2, T_2; F_2, D_2, W_2, S_2, G_2)$ are two nets satisfying Definition 1, $P_1 \cap P_2 \neq \varnothing$, $T_1 \cap T_2 \neq \varnothing$, $F_1 \cap F_2 = \varnothing$, $D_1 \cap D_2 = \varnothing$, and $S_1 \cap S_2 = \varnothing$. Their composition is $EN = EN_1 \varTheta EN_2 = (P_1 \cup P_2, T_1 \cup T_2, F_1 \cup F_2, D_1 \cup D_2, W, S_1 \cup S_2, G) = (P, T; F, D, W, S, G)$, in which for $\forall f_1 \in F_1 \subseteq F$, $W_1(f_1) = W\,(f_1)$; $\forall f_2 \in F_2 \subseteq F$, $W_2(f_2) = W\,(f_2)$; $\forall t_1 \in T_1 \subseteq T$, $G_2(t_1) = G\,(t_1)$; and $\forall t_2 \in T_2 \subseteq T$, $G_2(t_2) = G\,(t_2)$.

Definition 11 specifies a synthesis method of EBPN. In this work, we use it to synthesize control and data structures. To build up an e-commerce business process, we elicit intended functions in terms of design specifications and construct the EBPN according to some rules. Specific modeling rules are given as follows:

(1) Obtain the trading parameter set. Derive the trading parameter set involved in the design specifications and make D of the EN to be established;

(2) Construct the control structure models $EN_i = (P_i, T_i; F_i, D_i, W_i, S_i, G_i)$, $i \in \mathbb{N}$, which correspond to Shopper, Merchant, and Caas, and so on. Identify the order of APIs and operation events in the e-commerce system according to key functionalities and design specifications, they are represented as transitions; connect them by using places and arcs;

(3) Construct a data structure model $EN_j = (P_j, T_j; F_j, D_j, W_j, S_j, G_j)$, $j \in \mathbb{N}$. Identify the data flows and specify the input and output trading parameters of identified transitions; connect the transitions according to data flow specifications by using places and arcs;

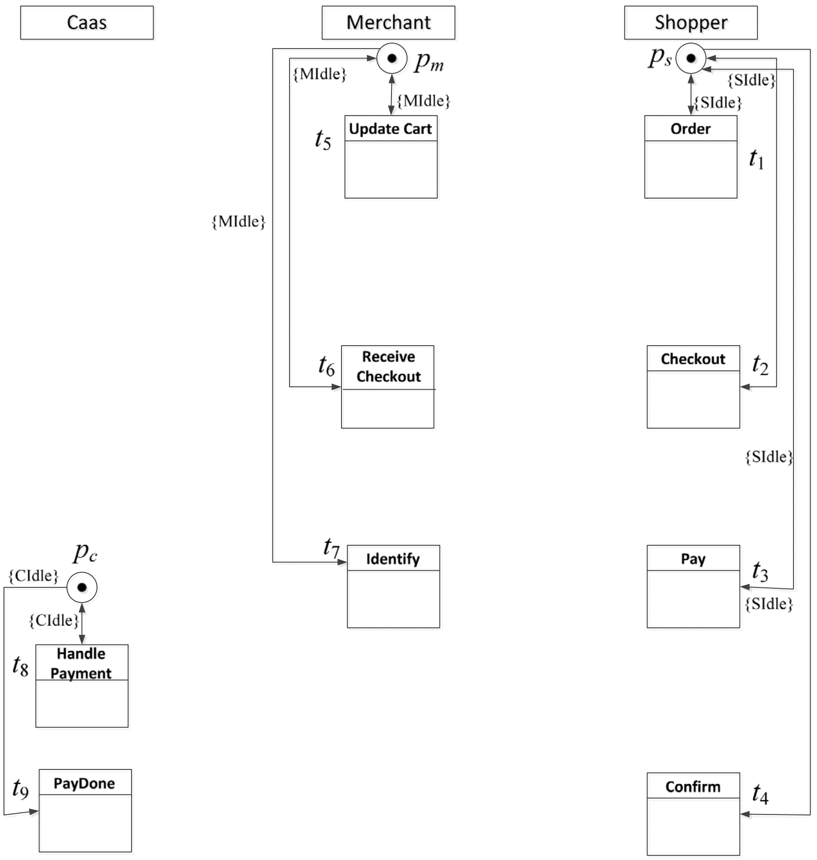


Fig. 3. Control structures of motivation example.

(4) Composing the control and data structures to obtain $EN = EN_{\ i1} \ \Theta \ EN_{i2} \ \Theta \ \ldots \ \Theta \ EN_{in} \ \Theta \ \ldots \ \Theta \ EN_j = (P, T; F, D, W, S, G)$, in which $i\,1, i\,2, in, j \in \mathbb{N}$;

(5) Add the initial data state. For an EBPN corresponding to an e-commerce business process, the initial data state is unique, which represents that multiple participants are ready for a deal.

Above steps describe the process of constructing an EBPN model. Using the proposed modeling methods and intended function specifications in Fig. 1 and Table 1, we can model the business process structures. For example, Fig. 3 depicts three control structures. The operation events of Shopper, as well as APIs of Caas and Merchant, are depicted by transitions. As Fig. 3 shows, the operation "Order" of Shopper is represented by t_1, and the phrase in it is used to signal its function. t_5 is an API of Merchant, representing the function of processing an order from Shopper. In Fig. 3, p_s, p_m and p_c are the initial places of Shopper, Merchant and Caas respectively, and they are linked with three transitions: t_1, t_5 and t_8 via two contrary arrows. {*Sidle*, *MIdle*, *CIdle*} is the set of control parameters and this means that three participants are always ready to start a new transaction.

Fig. 4 is the data structure. {*Cart*, *SessionID*, *Checkout*, *Status*} is the set of trading parameters. In the motivation example, only when the merchant is informed by the CaaS, the merchant will create an order of the transaction according to what is inside the cart and set its status to "PAID", as per Table 1. API Handle Payment loads the cart directly from the shopper's session, rather than from the CaaS, to build the order. This business function is represented by the place p_2 in data structure in Fig. 4. After the function of Update cart is done, t_5 would send three tokens with the types of *SessionID*, *Cart*, *Checkout* respectively to p_2. The arc from p_2 to t_7 means that Merchant loads the cart directly from the shopper's session.

Synthesize data and control structural models sequentially to obtain the complete EBPN (Fig. 5). Then, its initial marking $M_0 = [p_s(SIdle), p_m(MIdle), p_c(CIdle), p_2(SessonID, Cart)]$ representing that multiple participants are ready to conduct a transaction. Note that, in this case, S and G are not used. In the following, we use bold zero ($\mathbf{0}$) to represent the l-dimensional 0 vector for clarity. The order of parameters in three-dimensional incidence matrix [23] is {*SIdle, MIdle, CIdle, Cart, SessionID, Checkout, Status* }, and the numeric initial marking is $M_0 = [\langle 1, 0, 0, 0, 0, 0, 0 \rangle, \langle 0, 1, 0, 0, 0, 0, 0 \rangle, \langle 0, 0, 1, 0, 0, 0, 0 \rangle, \mathbf{0}, \langle 0, 0, 0, 1, 1, 0, 0 \rangle, \mathbf{0, 0, 0, 0, 0, 0, 0}]$.

6. Analyzing structural security

We analyze structural security by two ways according to Propositions 1 and 2. One is the behavioral sequence method, in which illegal behavior sequences are constructed and executed to verify the structural security of an e-commerce business process; the other one is the state analyzing method, in which an illegal state is constructed and analyzed by three-dimensional incidence matrix.

6.1. Behavioral sequence method

By analyzing the case of e-commerce business process, we should construct the illegal behaviors that result in security issues. Then, convert them to the behavioral sequence of EBPN model. At last, executing them and determine whether it can be executed successfully. If yes, the business structures have problems; if no, the business structures are immune to these illegal behaviors.

Please cite this article in press as: W. Yu, et al., Petri net-based methods for analyzing structural security in e-commerce business processes, Future Generation Computer Systems (2018), https://doi.org/10.1016/j.future.2018.04.090.

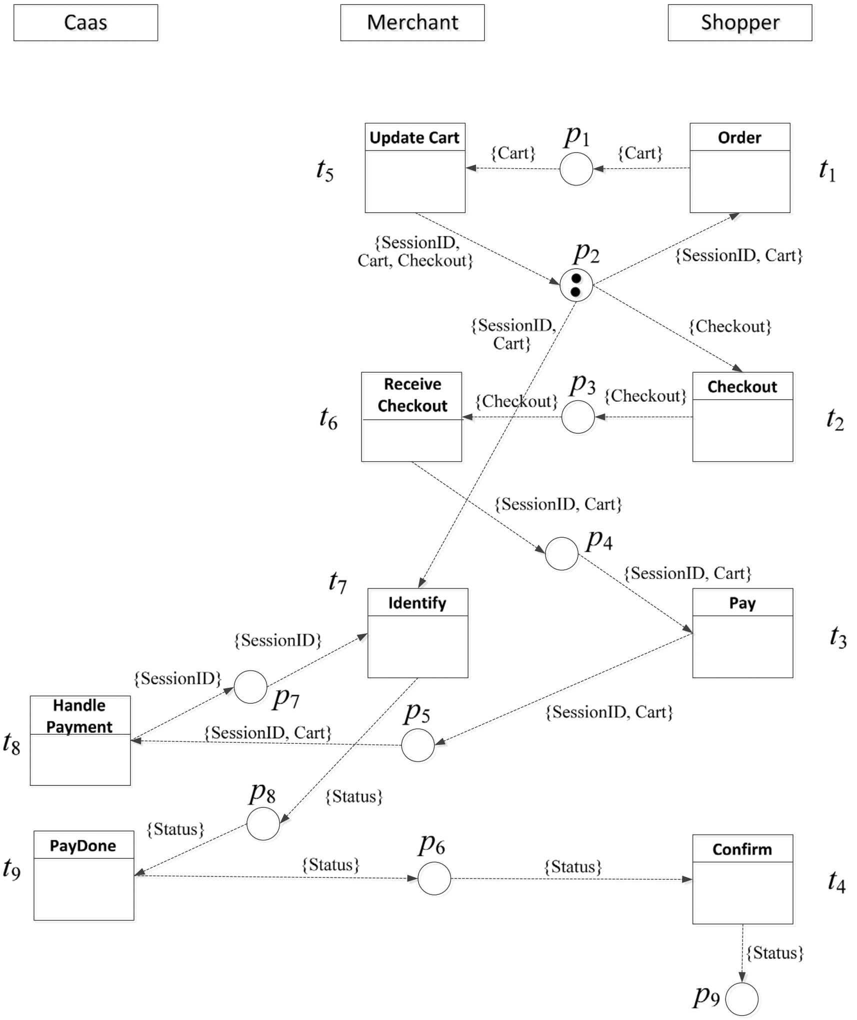


Fig. 4. Data structure of motivation example.

First, we should construct the illegal behavior specifications from the cases. Illegal behavior specification is a description of illegal behaviors in the case. For example, in the case of motivation example, the illegal behavior specification is:

(Order → Update Cart)* → Checkout → . . . →(Order→Update Cart)* → . . . → Identity → . . .

The notation ()* means the loop executing of some events. The notation "..." means any executing events or transitions in the business process.

Definition 12. Suppose that $EN = (P, T; F, D, W, S, G)$ is an EBPN model, the illegal behavior sequence σ can be defined as follows:

(1) $t \in T$ can be an illegal behavior sequence;
(2) $t_1\, t_2 \ldots t_n$ can be an illegal behavior sequence, $n \in \mathbb{N}^+$;
(3) $(t_1\, t_2 \ldots t_n)^*$ can be an illegal behavior sequence;
(4) The composition of (1)–(3) can be an illegal behavior sequence.

Thus, the corresponding illegal behavior sequence of above illegal behavior specifications is:

$(t_1\, t_5)^*\, t_2 \ldots (t_1\, t_5)^* \ldots t_7 \ldots$

Then, we should conclude a specific behavior sequence according to illegal behavior sequence. For example, according to Fig. 5 and the illegal behavior sequence, we can get a specific behavior sequence $t_1\, t_5\, t_2\, t_1\, t_5\, t_6\, t_3\, t_8\, t_7$. Using the dynamic properties of EBPN, execute the sequence and the executing process is shown in Fig. 6. The specific behavioral sequence can be executed successfully, and the business structures of motivation example do not satisfy structural security.

6.2. State analyzing method

In addition, we can use the states method to analyze structural security. First, we need to construct an illegal data state according to the structural security issue. Then, determine the reachability of the illegal data state using a three-dimensional incidence matrix. If yes, there is a structural security problem in the e-commerce business process; if no, the e-commerce business process is immune to the structural security issue.

Then, we give the steps of determining the reachability of a data state (M, δ_D) in an EBPN $EN = (P, T; F, D, W, S, G,)$ under the initial data state (M_0, δ_0).
(1) Construct Ψ;
(2) According to Theorem 1, compute whether the non-negative integer n-dimensional vector V exists;
(3) If not so, conclude that (M, δ_D) is not reachable in EN under the initial data state (M_0, δ_{D0}).
(4) If V exists and the transition sequences σ such that $M_0 \xrightarrow{\sigma} M$ can be found, executing σ in EN, and the reachability of (M, δ_D) can be determined.

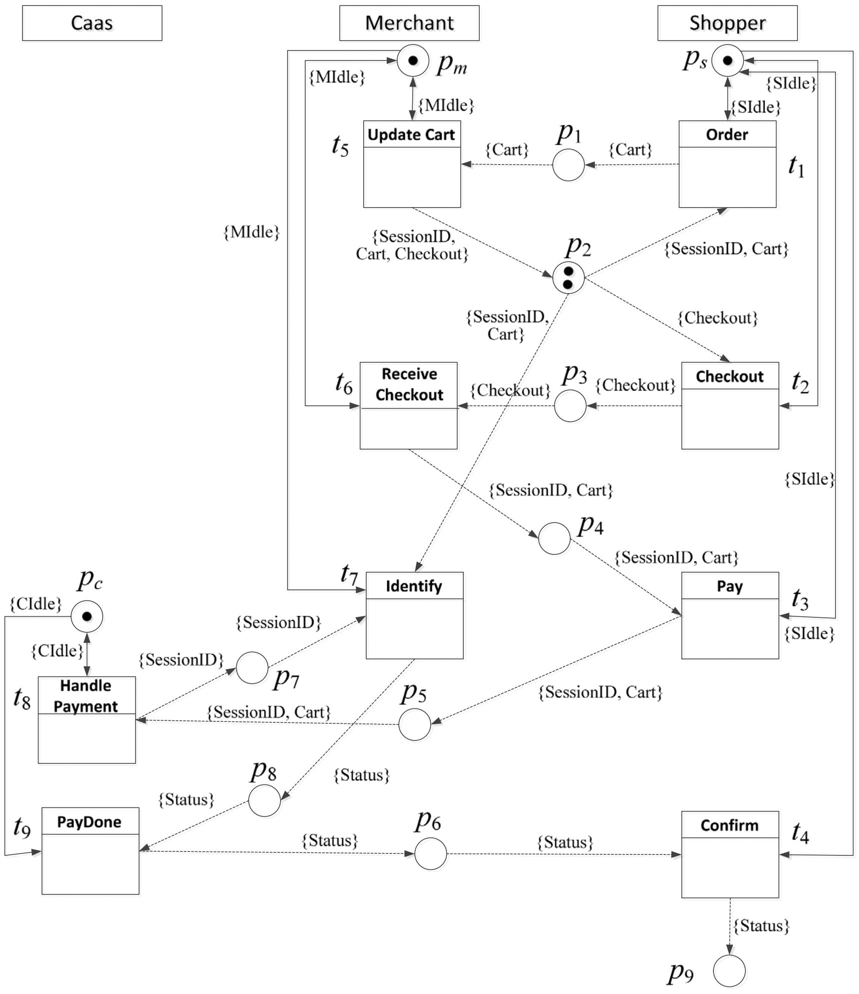


Fig. 5. Complete business structure of motivation example.

$$([p_s(\text{SIdle}), p_m(\text{MIdle}), p_c(\text{CIdle}), p_2(\text{SessionID, Cart})])$$
$$\downarrow t_1$$
$$([p_s(\text{SIdle}), p_m(\text{MIdle}), p_c(\text{CIdle}), p_1(\text{Cart})])$$
$$\downarrow t_5$$
$$([p_s(\text{SIdle}), p_m(\text{MIdle}), p_c(\text{CIdle}), p_2(\text{SessionID, Cart, Checkout})])$$
$$\downarrow t_2$$
$$([p_s(\text{SIdle}), p_m(\text{MIdle}), p_c(\text{CIdle}), p_2(\text{SessionID, Cart}), p_3(\text{Checkout})])$$
$$\downarrow t_1$$
$$([p_s(\text{SIdle}), p_m(\text{MIdle}), p_c(\text{CIdle}), p_1(\text{Cart}), p_3(\text{Checkout})])$$
$$\downarrow t_5$$
$$([p_s(\text{SIdle}), p_m(\text{MIdle}), p_c(\text{CIdle}),p_2(\text{SessionID, Cart, Checkout}), p_3(\text{Checkout})])$$
$$\downarrow t_6$$
$$([p_s(\text{SIdle}), p_m(\text{MIdle}), p_c(\text{CIdle}),p_2(\text{SessionID, Cart, Checkout}), p_4(\text{SessionID, Cart})])$$
$$\downarrow t_3$$
$$([p_s(\text{SIdle}), p_m(\text{MIdle}), p_c(\text{CIdle}),p_2(\text{SessionID, Cart, Checkout}), p_5(\text{SessionID, Cart})])$$
$$\downarrow t_8$$
$$([p_s(\text{SIdle}), p_m(\text{MIdle}), p_c(\text{CIdle}),p_2(\text{SessionID, Cart, Checkout}), p_7(\text{SessionID})])$$
$$\downarrow t_7$$
$$([p_s(\text{SIdle}), p_c(\text{CIdle}),p_2(\text{Checkout}), p_8(\text{Status})])$$

Fig. 6. Executing process of the specific behavioral sequence.

Please cite this article in press as: W. Yu, et al., Petri net-based methods for analyzing structural security in e-commerce business processes, Future Generation Computer Systems (2018), https://doi.org/10.1016/j.future.2018.04.090.

$$\begin{bmatrix}\langle 1,0,0,0,0,0,0\rangle\\ \langle 0,1,0,0,0,0,0\rangle\\ \langle 0,0,1,0,0,0,0\rangle\\ \langle 0,0,0,1,0,0,0\rangle\\ \mathbf{0}\\ \langle 0,0,0,0,0,1,0\rangle\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\langle 1-V[4],0,0,0,0,0,0\rangle\\ \langle 0,1-V[7],0,0,0,0,0\rangle\\ \langle 0,0,1-V[9],0,0,0,0\rangle\\ \langle 0,0,0,V[1]-V[5],0,0,0\rangle\\ \langle 0,0,0,1-V[1]+V[5]-V[7],1-V[1]+V[5]-V[7],0-V[2]+V[5],0\rangle\\ \langle 0,0,0,0,0,V[2]-V[6],0\rangle\\ \langle 0,0,0,V[6]-V[3],V[6]-V[3],0,0\rangle\\ \langle 0,0,0,V[3]-V[8],V[3]-V[8],0,0\rangle\\ \langle 0,0,0,0,0,0,V[9]-V[4]\rangle\\ \langle 0,0,0,0,V[8]-V[7],0,0\rangle\\ \langle 0,0,0,0,0,0,V[7]-V[9]\rangle\\ \langle 0,0,0,0,0,0,V[4]\rangle\end{bmatrix}.$$

Box I.

In the case of motivation example, we can obtain the illegal data state $(M, \delta_D) = ([p_s(SIdle), p_m(MIdle), p_c(CIdle), p_1(Cart), p_3(Checkout)])$, and it means that Shopper has send the message of Checkout. However, he/she update the cart again. This is not allowed in a security e-commerce business process. First, we construct three-dimensional incidence matrix of the EBPN model. Ψ^+ and Ψ^- of the EN in Fig. 5 is shown in Table A.1, Table A.2 in Appendix. Then, according to Definition 9, $\Psi = \Psi^+ - \Psi^-$ is shown in Table A.3 in Appendix, where $\Psi^+ = [\psi_{ijk}{}^+]_{n\times m\times l}$, and $\Psi^- = [\psi_{ijk}{}^-]_{n\times m\times l}$. The non-simplified $M = [\langle 1,0,0,0,0,0,0\rangle, \langle 0,1,0,0,0,0,0\rangle, \langle 0,0,1,0,0,0,0\rangle, \langle 0,0,0,1,0,0,0\rangle, \mathbf{0}, \langle 0,0,0,0,0,1,0\rangle, \mathbf{0,0,0,0,0,0}]$. Therefore, for (M, δ_D), the matrix equation can be:

$$\begin{bmatrix}\langle 1,0,0,0,0,0,0\rangle\\ \langle 0,1,0,0,0,0,0\rangle\\ \langle 0,0,1,0,0,0,0\rangle\\ \langle 0,0,0,1,0,0,0\rangle\\ \mathbf{0}\\ \langle 0,0,0,0,0,1,0\rangle\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\langle 1,0,0,0,0,0,0\rangle\\ \langle 0,1,0,0,0,0,0\rangle\\ \langle 0,0,1,0,0,0,0\rangle\\ \mathbf{0}\\ \langle 0,0,0,1,1,0,0\rangle\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} + \Psi^{\mathbf{T}} \begin{bmatrix}V[1]\\ V[2]\\ V[3]\\ V[4]\\ V[5]\\ V[6]\\ V[7]\\ V[8]\\ V[9]\end{bmatrix}.$$

Solve the equation and one step of the process is given in Box I.

Then, to obtain the unique solution $V = [V[1], V[2], V[3], V[4], V[5], V[6], V[7], V[8], V[9]]^{\mathbf{T}} = [2, 1, 0, 0, 1, 0, 0, 0, 0]^{\mathbf{T}}$. According to Corollary 1, it is easy to find a sequence $\sigma = t_1 t_5\ t_2\ t_1$ such that $M_0 \xrightarrow{\sigma} M$. Then, execute σ in Fig. 5, and the executing process is shown in Fig. 3. By analyzing Fig. 6, we conclude that $(M, \delta_D) = ([p_s(SIdle), p_m(MIdle), p_c(CIdle), p_1(Cart), p_3(Checkout)])$ is reachable from initial data state. This means the business structure in Fig. 5 has some problems, and we can find the problem and correct it. Fig. 7 shows the correct structures and has no structure security issue in the motivation example.

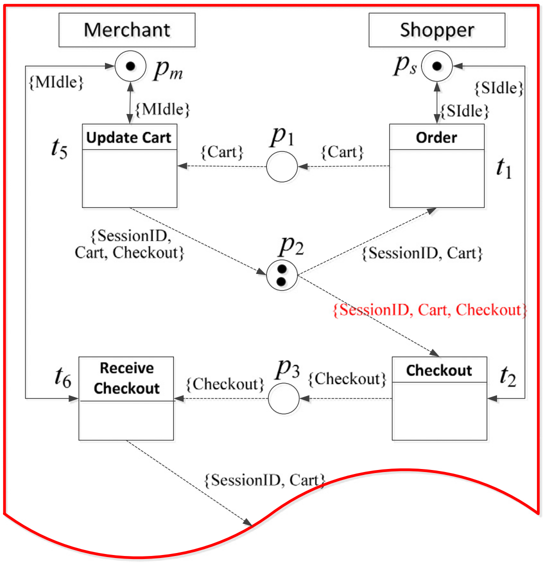


Fig. 7. Correct business structure of motivation example.

7. Conclusions

The rapid development of e-commerce has led to arise structural security issues in business processes. Based on EBPN, this paper discusses the concept of structural security and proposed a modeling method that fuses control and data structures. We propose two analyzing methods to determine the structural security of e-commerce business processes. However, plenty of analyzing methods in this area is still largely open. Even with the deployment of the proposed methods, there is still ample opportunity to conduct further research using additional analytical methods in structural security. In the future, we will focus on more efficient

Table A.1
Ψ^+ of Fig. 5.

p_s	p_m	p_c	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	
< 1,0,0,0,0,0,0 >	**0**	**0**	< 0,0,0,1,0,0,0 >	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	t_1
< 1,0,0,0,0,0,0 >	**0**	**0**	**0**	**0**	< 0,0,0,0,0,1,0 >	**0**	**0**	**0**	**0**	**0**	**0**	t_2
< 1,0,0,0,0,0,0 >	**0**	**0**	**0**	**0**	**0**	**0**	< 0,0,0,1,1,0,0 >	**0**	**0**	**0**	**0**	t_3
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	< 0,0,0,0,0,0,1 >	t_4
0	< 0,1,0,0,0,0,0 >	**0**	**0**	< 0,0,0,1,1,1,0 >	**0**	**0**	**0**	**0**	**0**	**0**	**0**	t_5
0	< 0,1,0,0,0,0,0 >	**0**	**0**	**0**	**0**	< 0,0,0,1,1,0,0 >	**0**	**0**	**0**	**0**	**0**	t_6
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	< 0,0,0,0,0,0,1 >	**0**	t_7
0	**0**	< 0,0,1,0,0,0,0 >	**0**	**0**	**0**	**0**	**0**	**0**	< 0,0,0,0,1,0,0 >	**0**	**0**	t_8
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	< 0,0,0,0,0,0,1 >	**0**	**0**	**0**	t_9

Table A.2
Ψ^- of Fig. 5.

$$\begin{array}{ccccccccccccc}
p_s & p_m & p_c & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 & p_8 & p_9 & \\
<1,0,0,0,0,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,1,1,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & t_1 \\
<1,0,0,0,0,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,0,0,1,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & t_2 \\
<1,0,0,0,0,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,1,1,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & t_3 \\
<1,0,0,0,0,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,0,0,0,1> & \mathbf{0} & \mathbf{0} & \mathbf{0} & t_4 \\
\mathbf{0} & <0,1,0,0,0,0,0> & \mathbf{0} & <0,0,0,1,0,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & t_5 \\
\mathbf{0} & <0,1,0,0,0,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,0,0,1,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & t_6 \\
\mathbf{0} & <0,1,0,0,0,0,0> & \mathbf{0} & \mathbf{0} & <0,0,0,1,1,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,0,1,0,0> & \mathbf{0} & \mathbf{0} & t_7 \\
\mathbf{0} & \mathbf{0} & <0,0,1,0,0,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,1,1,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & t_8 \\
\mathbf{0} & \mathbf{0} & <0,0,1,0,0,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,0,0,0,1> & \mathbf{0} & t_9
\end{array}$$

Table A.3
Ψ of Fig. 5.

$$\begin{array}{ccccccccccccc}
p_s & p_m & p_c & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 & p_8 & p_9 & \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,1,0,0,0> & <0,0,0,-1,-1,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & t_1 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,0,0,-1,0> & <0,0,0,0,0,1,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & t_2 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,-1,-1,0,0> & <0,0,0,1,1,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & t_3 \\
<-1,0,0,0,0,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,0,0,0,-1> & \mathbf{0} & \mathbf{0} & <0,0,0,0,0,0,1> & t_4 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,-1,0,0,0> & <0,0,0,1,1,1,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & t_5 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,0,0,-1,0> & <0,0,0,1,1,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & t_6 \\
\mathbf{0} & <0,-1,0,0,0,0,0> & \mathbf{0} & \mathbf{0} & <0,0,0,-1,-1,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,0,-1,0,0> & <0,0,0,0,0,0,1> & \mathbf{0} & t_7 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,-1,-1,0,0> & \mathbf{0} & <0,0,0,0,1,0,0> & \mathbf{0} & \mathbf{0} & t_8 \\
\mathbf{0} & \mathbf{0} & <0,0,-1,0,0,0,0> & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & <0,0,0,0,0,0,1> & \mathbf{0} & <0,0,0,0,0,0,-1> & \mathbf{0} & t_9
\end{array}$$

analytical methods and deploy more technologies based on the original Petri nets.

Acknowledgments

This paper is in part supported by the National Natural Science Foundation of China under grant 61602289, by the Fundamental Research Funds for the Central Universities of China under grants GK201803081 and GK201801004, by the Natural Science Basic Research Plan in Shaanxi Province of China under grant 2016JQ6056, by Oversea Scholarship Program of Shaanxi Normal University and National scholarship Funds of China, by the National Natural Science Foundation of China under grants 11372167, 61340003 and 61402273, by the Postgraduate Education Reform Project of Shaanxi Normal University, China under grant GERP-15-12, by the 111 Programme of Introducing Talents of Discipline to Universities, China (Grant No. B16031).

Appendix

See Tables A.1–A.3.

References

[1] CNNIC, China Internet development statistics report, China Internet Network Inform. Center, Beijing, China, Jan. 2017.

[2] iResearch: 2016 Q3 e-commerce market core data. 2016. Available: http://report.iresearch.cn/content/2016/11/265616.shtml#a1.

[3] R. Wang, S. Chen, X.F. Wang, S. Qadeer, How to shop for free online-Security analysis of cashier-as-a-service based web stores, in: Proc. 32nd IEEE Symp. Security Privacy (S&P), Berkeley, CA, 2011, pp. 465–480.

[4] D. Hirschberger, et al., Bachelor thesis cashier-as-a-service based webshops overview and steps towards security testing, 2016.

[5] K. Bhargavan, et al., Modular verification of security protocol code by typing, in: Proceedings of the 37th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages, New York, NY, USA, 2010, pp. 445–456.

[6] A. Sudhodanan, et al., Attack patterns for black-box security testing of multi-party web applications, in: NDSS, 2016.

[7] One yuan gate event of Taobao [Online]. 2012, Feb. 21. Available: http://baike.baidu.com/view/6419081.htm.

[8] S. Wen, et al., Lom: Discovering logic flaws within mongodb-based web applications, Int. J. Autom. Comput. 14 (2017) 106–118.

[9] A.D. Brucker, I. Hang, Secure and compliant implementation of business process-driven systems, in: Business Process Management Workshops, Tallinn, Estonia, 2013, pp. 662–674.

[10] A. Lehmann, N. Lohmann, Modeling wizard for confidential business processes, in: Business Process Management Workshops, Tallinn, Estonia, 2013, pp. 675–688.

[11] D. Knuplesch, et al., Towards compliance of cross-organizational processes and their changes, in: Business Process Management Workshops, Tallinn, Estonia, 2013, pp. 649–661.

[12] B. Depaire, et al., A process deviation analysis framework, in: Business Process Management Workshops, Tallinn, Estonia, 2013, pp. 701–706.

[13] R. Bose, W.M.P. van der Aalst, Process diagnostics using trace alignment: Opportunities, issues, and challenges, Inf. Syst. 37 (2012) 117–141.

[14] W.M.P. van der Aalst, et al., Ensuring correctness during process configuration via partner synthesis, Inf. Syst. 37 (2012) 574–592.

[15] G.J. Liu, W. Reisig, C.J. Jiang, M.C. Zhou, A Branching-process-based method to check soundness of workflow systems, IEEE Access 4 (2016) 4104–4118.

[16] G.J. Liu, C.J. Jiang, Net-structure-based conditions to decide compatibility and weak compatibility for a class of inter-organizational workflow nets, Sci. China Inf. Sci. 58 (2015) 1–16.

[17] Y.Y. Du, et al., Modeling and monitoring of e-commerce workflows, Inform. Sci. 179 (2009) 995–1006.

[18] Y.Y. Du, et al., A Petri-net-based correctness analysis of internet stock trading systems, IEEE Trans. Syst. Man Cybern. C 38 (2008) 93–99.

[19] Y.Y. Du, et al., A Petri net-based model for verification of obligations and accountability in cooperative systems, IEEE Trans. Syst. Man Cybern. A 39 (2009) 299–308.

[20] Y. Du, et al., A Petri net approach to mediation-aided composition of web services, IEEE Trans. Autom. Sci. Eng. 9 (2012) 429–435.

[21] W. Tan, et al., Data-driven service composition in enterprise SOA solutions: A Petri net approach, IEEE Trans. Autom. Sci. Eng. 7 (2010) 686–694.

[22] L. Liu, et al., A socio-ecological model for advanced service discovery in machine-to-machine communication networks, ACM Trans. Embedded Comput. Syst. 15 (2016) 38:1–38:26.

[23] Y. Wu, et al., An adaptive multilevel indexing method for disaster service discovery, IEEE Trans. Comput. 64 (2015) 2447–2459.

[24] J. Xu, et al., Dynamic authentication for cross-realm SOA-based business processes, IEEE Trans. Serv. Comput. 5 (2012) 20–32.

[25] W.Y. Yu, et al., Modeling and validating e-commerce business process based on petri nets, IEEE Trans. Syst. Man Cybern. Syst. 44 (2014) 327–341.

[26] W.Y. Yu, et al., Analyzing e-commerce business process nets via incidence matrix and reduction, IEEE Trans. Syst. Man Cybern. Syst. 48 (2018) 130–141.

[27] Z.H. Wu, Introduction to Petri Nets, Machine Press, Beijing, China, 2006.

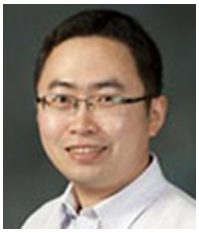
[28] B. Hrúz, M.C. Zhou, Modeling and Control of Discrete-Event Dynamic Systems, Springer-Verlag, London, U.K., 2007.

[29] M.C. Zhou, K. Venkatesh, Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach, World Scientific Publishing, Singapore, 1999.

Wangyang Yu received the M.S. degree from Shandong University of Science and Technology, Qingdao, China, in 2009, and Ph.D. degree from Tongji University, Shanghai, China, in 2014. He is currently an Associated Professor with the College of Computer Science, Shaanxi Normal University, Xi'an, China. His research interests include the theory of Petri nets, formal methods in software engineering and trustworthy software.

Please cite this article in press as: W. Yu, et al., Petri net-based methods for analyzing structural security in e-commerce business processes, Future Generation Computer Systems (2018), https://doi.org/10.1016/j.future.2018.04.090.

Zhijun Ding received the M.S. degree from Shandong University of Science and Technology, Taian, China, in 2001, and Ph.D. degree from Tongji University, Shanghai, China, in 2007. Now he is an Associate Professor of the Department of Computer Science and Technology, Tongji University. His research interests are in formal engineering, Petri nets, services computing, and workflows. He has published more than 60 papers in domestic and international academic journals and conference proceedings.



Lu Liu is the Head of the Department of Electronics, Computing and Mathematics in the University of Derby and adjunct professor in the School of Computer Science and Communication Engineering at Jiangsu University. Prof. Liu received his Ph.D. degree from University of Surrey. He is the Fellow of British Computer Society and Member of IEEE. Prof. Liu's research interests are in areas of Cloud Computing, Social Computing, Service-oriented Computing and Peer-to-Peer Computing.

Xiaoming Wang received his Ph.D. degree in computer theory and software from Northwest University, Xian, P.R. China, in 2005. He is currently a professor in Shaanxi Normal University, Xian, China. His current research interests include network security, pervasive computing, wireless sensor network, and opportunistic networks. Prof. Wang has authored and coauthored more than 40 publications in journal, books and international conference proceedings.

Richard David Crossley received a B.A. in English Literature from the University of Lancaster (2002) and an M.Sc. in Information Technology from the University of Derby (2014). He is currently pursuing a Ph.D. in Optimization in High Performance Computing to achieve maximum resource efficiency at the University of Derby, sponsored by Rolls-Royce PLC and the university.

Please cite this article in press as: W. Yu, et al., Petri net-based methods for analyzing structural security in e-commerce business processes, Future Generation Computer Systems (2018), https://doi.org/10.1016/j.future.2018.04.090.