Information Technology Journal 7 (3): 466-473, 2008
ISSN 1812-5638

Analysis and Verification of Dynamic Stock Trading Systems

[1,3]Yuyue Du, [2]Hong Zheng and [1]Shuxia Yu
[1]College of Information Science and Engineering,
Shandong University of Science and Technology, Qingdao 266510, China
[2]Department of Computer Science and Engineering,
East China University of Science and Technology, Shanghai 200237, China
[3]State Key Laboratory of Computer Science, Institute of Softwate,
Chinese Academy of Sciences, Beijing 100080, China

Abstract: A dynamic stock trading system with a distributed shared memory is analyzed formally based on its temporal Petri net model. The functional correctness of the system is formally verified and some important properties of the system are investigated, such as liveness, fairness, safeness and temporal properties. Finally, conclusions are found.

Key words: Dynamic stock trading system, correctness, temporal property, verification, temporal Petri net

INTRODUCTION

Petri nets (PN) (Murata, 1989; Du *et al*., 2006) are a promising graphical and mathematical modeling tool for concurrent systems. However, PNs cannot describe explicitly some fundamental properties of the modeled systems, such as eventuality and fairness. In timed Petri nets (Berthomieu and Diaz, 1991; Du *et al*., 2007), moreover, the explicit introduction of time into them leads to very complicated formulae that tend to obscure the ideas about underlying causal and temporal relationships between events. Also, associating execution times or delays with transitions and places individually is inadequate for some of the fundamental properties. For this purpose, temporal Petri nets (TPN) are represented (Suzuki and Lu, 1989; Zurawski, 1997). TPNs can describe elegantly timing constraints, the dynamical behavior of a modeled system and causality between events on the basis of the formulae with temporal operators. The requirement specifications of a static stock trading system are represented based on the corresponding temporal logic formulae in Du *et al.* (2008) and the correctness of the system is verified based on its TPN model. While a dynamic stock trading system (DSTS) is modeled and specified by Zheng and Du (2005). In fact, the analysis of DSTSs is more difficult and complex than that of static stock trading systems. In this study, therefore, the dynamical behavior of a DSTS and the causality between events are analyzed based on its TPN model and the fundamental properties of the system are verified formally, such as liveness, fairness and safeness properties. This research is investigated based on Du *et al.* (2008) and Zheng and Du (2005). It is further demonstrated that TPNs own a stronger modeling and analysis power than PNs by the formal analysis and correctness verification of a DSTS and the performance of DSTSs can be improved and enhanced effectively.

As the suppositions in Zheng and Du (2005), a dynamical stock trading system is in a multiprocessor or multicomputer system with shared-variable DSM (Distributed Shared Memory). In general, the matches of deal data from all kinds of stocks are made in a multiprocessor or multicomputer system with a shared memory on a stock exchange. Each kind of stocks is dynamically assigned to a processor or a computer, on which matches of the deal data of this kind of stocks will be made. In a static case, the deal data of dozens or hundreds kinds of stocks will be fixedly processed on some processor or computer. In a dynamic case, however, the stock trading system uses a dynamic binding way to match up stocks and processors and the number of kinds of the stocks processed on some processor or computer is flexible. One of the primary functions of a DSTS is that processors make the matches of deal data by means of deal rules, such as First Price and First Time. Stock-brokers will send the deal data of stockholders to a stock exchange once verifying their validity. Their arriving time will be recorded by a prepositional computer on the stock exchange, then they be transferred to the multiprocessor system. The deal data are processed here and the trading results are sent to corresponding stock-brokers.

Corresponding Author: Yuyue Du, College of Information Science and Engineering,
Shandong University of Science and Technology, Qingdao 266510, China

TPN MODEL OF A DYNAMIC STOCK TRADING SYSTEM

Some basic terminologies of PNs and TPNs are simply overviewed first in this section.

A PN is a 5-tuple, PN = (P, T; F, W, M_0), where P is a finite set of places; T is a finite set of transitions; F is a set of arcs; W: F → {1, 2, ...} is a weight function and W is called an variable weight function if W(p, t) or W(t, p) is an variable; M_0 is the initial marking. t∈T is said to be firable at M iff ∀p∈•t: M(p) =W(p, t), where •t represents a set of the input places of t and (p, t) ∈F. Firing t at M will yield a new marking M', i.e., M[t>M', where ∀p∈P: M' (p)=M(p)-W(p, t)+W(t, p). If W is an variable weight function, i.e., W(p, t) or W(t, p) is an variable, then it is represented by a positive integer N. Also, an input arc and the corresponding output arc of a transition have only the same weight when they have a variable weight N and N is equal to the number of tokens in the input place of the transition.

A TPN is a pair TPN=(PN, f), where f is a formula having the following syntax:

- Propositions: p, t_{fir} and t are atomic propositions, where p∈P, t∈T
- Atomic propositions are formulae
- If f and g are formulae, so are ¬ f, f +g, f•g, f ⇒g, ○f, ◊f, □f

The atomic propositions p, t_{fir} and t mean that there is at least one token in p, t is firable and t fires at the current marking, respectively. Symbols ¬, +, • and ⇒ represent the Boolean connectives, NOT, OR, AND and IMPLICATION. Formula of means that f becomes true at the next marking reached. □f means that f becomes true at every marking reached from the current marking. ◊f means that f becomes eventually true at some marking reached from the current marking.

For any set S, S* represents the set of all finite sequences of elements of S, including the empty sequence λ. |α| represents the length of α∈ S*. αβ denotes the concatenation of α and β. For i: 0≤ i ≤ |α|, let β_i and γ_i be the sequences such that $|\beta_i| = i$ and $\alpha = \beta_i \gamma_i$, β_i is the prefix of α with length i and γ_i is the postfix of α excluding β_i. Let M_i be a marking reached from M by firing β_i. Let f be a formula, <M, α>|− f means that f is satisfied by the pair of M and α, where |− represents a valid formula. Typical TPN formulae are defined as follows.

(a) <M, α>|− p iff M(p)>0
(b) <M, α>|− t_{fir} iff t is firable at M
(c) <M, α>|− t iff α ≠λ and t =β_1, i.e., t fires

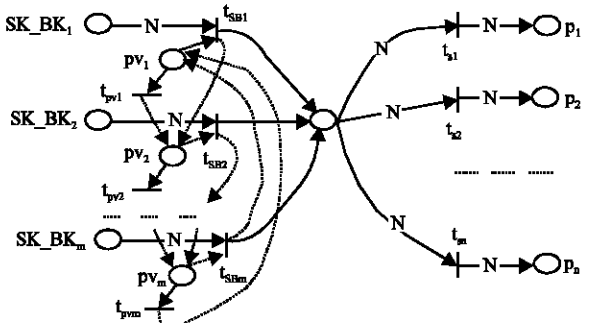

Fig. 1: A subnet model of a prepositional computer processing system

(d) <M, α>|− ¬f iff not <M, α>|− f
(e) <M, α>|− f•g iff <M, α>|− f and <M, α>|− g
(f) <M, α>|− f + g iff <M, α>|− f or <M, α>|− g
(g) <M, α>|− f ⇒g iff <M, α>|− f implies <M, α>|− g
(h) <M, α>|− ○f iff α ≠λ and <M_1, γ_1 >|− f
(i) <M, α>|− □f iff <M_i, γ_i >|− f for every 0≤ i ≤|α |
(j) <M, α>|− ◊f iff <M_i, γ_i >|− f for some 0≤ i ≤|α |
(k) <M, α>|− f until g iff (<M_i, γ_i >|− f for every 0≤ i ≤|α|) or (<M_i, γ_i >|− g for some 0≤ i ≤|α| and <M_j, γ_j >|− f, for every 0≤ j ≤ i)

The following properties can be easily proved by the above definitions.

PR_1 : <M, α>|− f +○f implies <M, α>|− ◊f
PR_2 : <M, α>|− □(f_1⇒◊f_2)· □(f_2⇒◊f_3) implies <M, α>|− □ (f_1⇒◊ f_3)
PR_3 : <M, α>|− ◊(◊f) implies <M, α>|− ◊f
PR_4 [5] : <M, α>|− □ (t_{fir}⇒◊t)

PR_4 means that if t is firable for any possible firing sequence α at M, then t fires eventually.

In this study, only the processing processes of deal data on a prepositional computer and in multiple-processor system (Fig. 1) are modeled and analyzed using TPNs on a stock exchange. Therefore, the PN model (notation N_D) of a DSTS consists of a subnet of the prepositional computer processing system, a subnet of every processor pre-processing system and a subnet of every processor making match system. The subnets are shown in Fig. 1-3, respectively.

Deal data of stockholders are usually divided into four classes: buying deal data, selling deal data, withdrawing buying and selling deal data. In Fig. 1, places $SK\text{-}BK_1$, $SK\text{-}BK_2$,..., $SK\text{-}BK_m$ contain the deal data from m stock-brokers, respectively; places p_1, p_2,... and p_n contain the deal data of n kinds of stocks processed on the stock exchange, respectively. If there is one token in place pv_i

($1 \le i \le m$), it means that the deal data in SK-BK_i have the privilege processed on the prepositional computer. The deal data in p_0 will be classified based on the types of stocks, then sent to corresponding places p_i ($1 \le i \le n$) by firing t_{Si}. A real arc means that it has a variable weight N. The doted arcs have an invariant weight 1 and denote the flow direction of the privilege. If transitions t_{pvi} and t_{SBi} ($1 \le i \le m$) become firable, they must satisfy the following temporal formulas (ST_1) and (ST_2), respectively besides the firing conditions of the subnet in Fig. 1. For $1 \le i \le m$:

(ST_1) : $\Box((t_{pvi})_{fir} \Rightarrow \neg SK\_BK_i)$
(ST_2) : $\Box((t_{SBi})_{fir} \Rightarrow \neg p_0)$

The formulas can be interpreted as in Zheng and Du (2005). When the deal data in SK-BK_i have gotten the privilege, if there is at least one token in it, then they do not deliver the privilege until t_{SBi} fires. But if it is empty here, the privilege must be transferred to the data in SK-BK_{i+1} by firing t_{pvi}. We obtain (ST_1). (ST_2) means that once the deal data with the privilege start to be processed on the prepositional computer, the other deal data do not be received until they are sent to the multiprocessor system.

In Fig. 2, if there is one token in the place pv_{ji} ($1 \le i \le n$), it means that the deal data in p_i have the privilege. If there is one token in pro_i, it denotes that the deal data in p_i are being processed on some processor. When there is one token in p_{ji_2}, it means that the matches of the deal data in p_i are being made on the processor(j). p_{i_4} and p_{i_8} contain respectively the residual buying and selling deal data belonging to the kind of stock in p_i, after the matches of the deal data are made on some processor.

Assume that there are k processors in the multiple-processor system. Figure 2 shows the pre-processing process on every processor. That is, all processors have the same pre-processing way and the process of the matches made on every processor is also same. Every processor can process the deal data from all kinds of stocks. But the matches of the deal data of some kind of stocks can be made only on one processor at any moment. Therefore, places p_i, pro_i, p_{i_4} and p_{i_8} ($1 \le i \le n$) are shared by all processors. Double-direction doted arcs connecting a transition t and a place p denote that p is not only an input place but also an output place of t and the number of tokens in p does not change after t fires.

Based on the previous discussion and deal rules, some transitions in Fig. 2 become firable only when they satisfy the following temporal formulas besides the firing conditions in this subnet. For $1 \le i \le n$, $1 \le j \le k$:

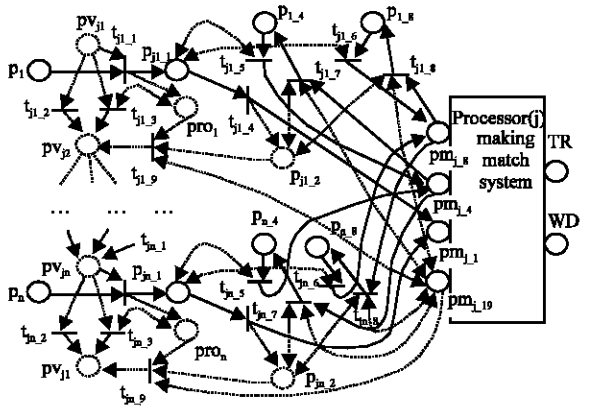

Fig. 2: A subnet model of the processor(j) ($1 \le j \le k$) pre-processing system

(St_3) : $\Box((t_{ji\_1})_{fir} \Rightarrow \neg pro_i)$
(St_4) : $\Box((t_{ji\_2})_{fir} \Rightarrow \neg p_i)$
(St_5) : $\Box((t_{ji\_4})_{fir} \Rightarrow \neg p_{i\_4} \bullet \neg p_{i\_8})$
(St_6) : $\Box((t_{ji\_9})_{fir} \Rightarrow \neg pm_{j\_4} \bullet \neg pm_{j\_8})$

These formulas can be interpreted as follows. (ST_3) means that if the matches of the deal data in p_i are being made on other processors, then they will not be processed on the processor (j) here. The interpretation of (ST_4) is similar to (ST_1). Since places p_{i_4} and p_{i_8} may contain respectively the residual buying and selling deal data after last one match of the deal data in p_i is made on some processor, the residual deal data must return to the processor (j) making system by firing t_{ji_5} and t_{ji_6} before the processor(j) makes the matches of the deal data in p_{ji_1}. We obtain (ST_5). (ST_6) can be interpreted similarly.

In Fig. 3, places TR and WD contain trading results and withdrawing deal data respectively, but they do not belong to the processor (j) trading system. They are two places in the processing system of stock-brokers. Places pm_{j_4} and pm_{j_8} deposit the buying and selling deal data, pm_{j_2} and pm_{j_11}, the withdrawing buying and selling deal data, respectively. By deal rules, if transitions t_{j_7}, t_{j_19}, t_{j_20} and t_{j_25} are firable, they must satisfy still the following temporal formulas (ST_7)-(ST_{10}) respectively besides the firing conditions of the subnet in Fig. 3.

(ST_7) : $\Box((t_{j\_7})_{fir} \Rightarrow \neg pm_{j\_1} \bullet \neg pm_{j\_24} \bullet \neg pm_{j\_25})$
(ST_8) : $\Box((t_{j\_19})_{fir} \Rightarrow \neg pm_{j\_1} \bullet \neg pm_{j\_4} \bullet \neg pm_{j\_25})$
(ST_9) : $\Box((t_{j\_20})_{fir} \Rightarrow \neg pm_{j\_1} \bullet \neg pm_{j\_24} \bullet \neg pm_{j\_8})$
(ST_{10}) : $\Box((t_{j\_25})_{fir} \Rightarrow \neg pm_{j\_7})$

Formulas (ST_7)-(ST_{10}) can be explained as follows. The datum with the highest buying price and the earliest arriving time in pm_{j_4} is transferred from pm_{j_4} to pm_{j_5} by

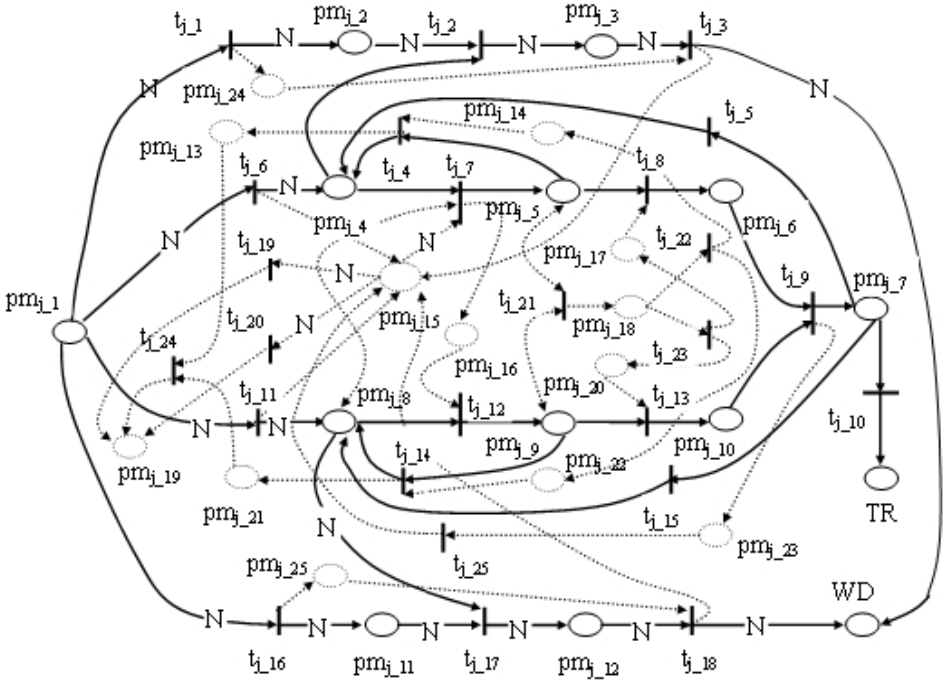

Fig. 3: A subnet model of the processor (j) making match system

firing t_{j_7}, the one with the lowest selling price and the earliest arriving time in pm_{j_8}, from pm_{j_8} to pm_{j_9} by firing t_{j_12}. If there is the withdrawing deal data in pm_{j_1}, they must be first withdrawn before t_{j_7} fires. We obtain (ST_7). If there is a token in pm_{j_19}, it means that processor(j) has made all matches of the deal data from p_i and prepare to process the other deal data having the privilege. A firing of t_{j_19} denotes that there is no buying deal datum in pm_{j_4} after at least one match of deal data is made or withdrawing deal data are removed, or there are only selling deal data in pm_{j_1} and they are transmitted to pm_{j_8}. Therefore, t_{j_19} can fire only when there is no token in pm_{j_1}, pm_{j_4} and pm_{j_25}. We obtain (ST_8). (ST_9) can be interpreted similarly. The deal datum with the highest buying price in pm_{j_5} is compared with the one with the lowest selling price in pm_{j_9} by firing t_{j_21}. If the price of the former is higher than that of the latter, t_{j_23} becomes firable, otherwise t_{j_22} becomes firable. A firing of t_{j_23} assures that this transaction has been clinched. By firing t_{j_9}, the trading results are sent to pm_{j_7}, a control token, to pm_{j_23}. The trading results in pm_{j_7} contain the data of matches made, the residual buying and selling deal data. Since a firing of t_{j_25} may cause a match made again, it becomes firable only when the data of matches made has been sent to place TR, the residual buying and selling deal data, to pm_{j_4} and pm_{j_8} respectively. We obtain (ST_{10}).

Therefore, the TPN model of the system is a pair $TND = (N_D, F_D)$, where N_D is a PN consisting of three subnets shown in Fig.1-3. And F_D is a set of formulas (ST_1)-(ST_{10}).

PROPERTIES ANALYSIS OF TPN MODEL TND

Since a prepositional computer processes rotationally the deal data from all stock-brokers and they may concurrently send deal data to the corresponding places during deal, we suppose that the initial marking M_0 of N_D contains the tokens in SK-BK_i ($1 \le i \le m$) besides the control tokens shown in Fig. 1-2.

Lemma 1: (At any moment, the deal data of only one stock-broker have the privilege on the prepositional computer.) Let M be a marking reachable of TND from M_0, then for any firing sequence α from M, we have

$$\sum_{i=1}^{m} M(pv_i) = 1$$

Lemma 1 can be easily obtained by using the structure of the subnet in Fig.1. It presents a safeness property. Note that Lemma 1 is equivalent to the following conclusion.

$$<M, \alpha> \vdash \Box\, pv_i \rightarrow \neg pv_r$$

where any α, i and r such that $i \neq r$.

Lemma 2: (The fairness property for stock-brokers in a prepositional computer processing system.) Let M be a marking reachable of TND from M_0, then for any firing sequence α from M and any i, we have

$$<M, \alpha>|-\Box\Diamond SK\text{-}BK_i \Rightarrow \Box\Diamond t_{SBi}$$

Lemma 2 can be proved by means of Lemma 1 and the structure of the subnet in Fig. 1. Lemma 2 means if there is at least one token in $SK\text{-}BK_i$ at infinitely many markings, then t_{SBi} must fire infinitely often. On other words, if a stock-broker sends infinitely deal data to the prepositional computer, then they are often processed infinitely. A similar conclusion for all kinds of stocks is given as follows. It can be obtained by the structure of the subnet in Fig. 2.

Lemma 3: (The fairness property for stocks in a processor pre-processing system.) Let M be a marking reached of TND from M_0, then for any firing sequence α from M and any i, we have

$$<M, \alpha>|-\Box\Diamond p_i \Rightarrow \Box\Diamond pro_i$$

Lemma 3 means that, if deal data of some kind of stock are infinitely sent on the prepositional computer to the multiprocessor system, the matches of them are infinitely made on the processors.

By the previous suppositions and the subnet in Fig. 2, places p_i and pro_i are shared on all processors. Thus, pro_i plays a mutual exclusion role in making the matches of the deal data in p_i on all processors.

Lemma 4: (The safeness (mutual exclusion) property for all kinds of stocks in processor pre-processing systems.) Let M be a marking reachable of TND from M_0. For any firing sequence α from M and any i and j, we have

$$<M, \alpha>|-\neg(p_i \bullet pro_i \bullet pv_{ji} \Rightarrow \Diamond t_{ji\_3})$$

Proof: By the structure of the subnet in Fig. 2 and formula (a), pv_{ji} means that the processor (j) is at leisure at M; pro_i means that the former arriving deal data in p_i are being processed by some processor (r) ($r \neq j$). We obtain

$$<M, \alpha>|-\Box(p_i \bullet pro_i \bullet pv_{ji} \Rightarrow (t_{ji\_3})_{fir}) \quad (1)$$

(1) and PR_4, yield

$$<M, \alpha>|-\Box((t_{ji\_3})_{fir} \Rightarrow \Box t_{ji\_3}) \quad (2)$$

(1), (2), PR_1 and PR_2, give

$$<M, \alpha>|-\Box (p_i \bullet pro_i \bullet pv_{ji} \Rightarrow \Box\, t_{ji\_3})$$

Lemma 4 shows that when the deal data in p_i are being processed on the processor(r), even if there are new arriving data in p_i and a processor (j) ($j \neq r$) is at leisure, it does not make the matches of the deal data in p_i again. From Lemmas 3-4 and the structure of the subnet in Fig. 2, we can obtain the following conclusion.

Lemma 5: (At any moment, if the deal data from at least k kinds of stocks have arrived in processor pre-processing systems, then all processors will be eventually be busy at work.) Let M be a marking reached of TND from M_0, then for any firing sequence α and any $i_1, i_2, \ldots, i_r \in \{1, 2, \ldots, n\}$, $\exists j_1, j_2, \ldots, j_k \in \{i_1, i_2, \ldots, i_r\}$, where $r = k$, $i_u \neq i_v$ and $j_u \neq j_v$ when $u \neq v$, we have

$$<M, \alpha>|-\Box (p_{i1} \bullet p_{i2} \bullet \ldots \bullet p_{ir} \Rightarrow \Diamond(pro_{j1} \bullet pro_{j2} \bullet \ldots \bullet pro_{jk}))$$

Lemma 5 denote that if there are deal data waiting to be processed and the corresponding place pro_i contain non token in the multiprocessor system, then the matches of them will be eventually made on some processor.

CORRECTNESS OF THE DYNAMIC STOCK TRADING SYSTEM

Here, the major functional correctness of the DSTS modeled by TND will be verified. The following conclusions are deduced by means of the structures of the subnets in Fig. 1-3, temporal formulas (ST_1)-(ST_{10}), TPN formulas (a)-(k) and TPN properties PR_1-PR_4. The requirements specification of the DSTS is described by the corresponding temporal logic formulas.

By the structure of the subnet in Fig. 3, (ST_7) and TPN formulas (k), the following lemma can be easily proved. Note the $(t_{j\_7})_{fir}$ means that places pm_{j_1}, pm_{j_24} and pm_{j_25} must be empty. And from deal rules, if there are withdrawing buying or selling deal data in pm_{j_1}, then there must be the corresponding buying or selling deal data in pm_{j_4} or pm_{j_8}, respectively. But if there are withdrawing (buying or selling) deal data in pm_{j_1} at M, then at least one place of them is not empty before t_{j_3} or t_{j_18} fires.

Lemma 6: (If withdrawing deal data arrive on a processor, then they are first withdrawn before the processor begins to make a match.) Let M be a marking reachable of TND from M_0, then for any firing sequence α from M, we have

$$<M, \alpha> |-\Box((t_{j\_1})_{fir}+(t_{j\_16})_{fir} \Rightarrow \Box (\neg(t_{j\_7})_{fir} \text{ until } (t_{j\_3}+ t_{j\_18})))$$

Lemma 6 illustrates that withdrawing buying or selling deal data have higher priority processed than buying or selling deal data in processor making match systems. This agrees with the deal rules of stocks.

Lemma 7: (If the deal data of some kind of stock arrive in a processor making match system, then the processor will eventually finish making all matches of them.) Let M be a making reachable of TND from M_0. For any firing sequence α from M, we have

$$<M, \alpha> |-\Box(pm_{j\_1} \Rightarrow \Diamond\ pm_{j\_19})$$

Proof: Given that the deal data are being processed on the processor (j). By the structure of the subnet in Fig. 3, we obtain

$$<M, \alpha> |-\Box(pm_{j\_1} \Rightarrow (t_{j\_1})_{fir} +(t_{j\_16})_{fir} +(t_{j\_6})_{fir} +(t_{j\_11})_{fir}) \quad (3)$$

(3) and formula (f), yield

$$<M, \alpha> |-\Box(pm_{j\_1} \Rightarrow (t_{j\_6})_{fir} +(t_{j\_11})_{fir}) \quad (4)$$

$$\text{or } <M, \alpha>|-\Box(pm_{j\_1} \Rightarrow (t_{j\_1})_{fir} +(t_{j\_16})_{fir}) \quad (5)$$

Here, the proof of the states, which only (4) is valid is valid, will be shown as follows to save space.

Since a firing of t_{j_7} requires that pm_{j_1} is empty, if t_{j_6} and t_{j_11} are firable, then they must fire successively. Thus, from (4), formula (f) and PR_4, we have

$$<M, \alpha> |-\Box(t_{j\_6} +t_{j\_11} \Rightarrow \circ((pm_{j\_4}+ pm_{j\_8}) \bullet pm_{j\_15} \bullet \neg pm_{j\_1})) \quad (6)$$

Because pm_{j_24} and pm_{j_25} are empty, if one of pm_{j_4} and pm_{j_8} is empty, then the lemma is proved by firing t_{j_19} or t_{j_20} from (ST_8) or (ST_9). Therefore, we suppose that pm_{j_4} and pm_{j_8} are nonempty. By (6) and PR_1 we have

$$<M, \alpha> |-\Box(t_{j\_12} \Rightarrow \circ (pm_{j\_5} \bullet pm_{j\_9})) \quad (7)$$

$$<M, \alpha> |-\Box(pm_{j\_5} \bullet pm_{j\_9} \Rightarrow (t_{j\_21})_{fir}) \quad (8)$$

$$<M, \alpha> |-\Box(t_{j\_21} \Rightarrow \circ (pm_{j\_5} \bullet pm_{j\_9} \bullet pm_{j\_18})) \quad (9)$$

Based on trading rules, here there is the deal datum with the highest buying price in pm_{j_5}, the deal datum with the lowest selling price in pm_{j_9}. If the price of the former is less than that of the latter, then t_{j_22} is firable, otherwise t_{j_23} is firable. Therefore, two cases can be respectively discussed as follows.

$$\text{CASE 1: } <M, \alpha> |-\Box(pm_{j\_5} \bullet pm_{j\_9} \bullet pm_{j\_18} \Rightarrow \Diamond(t_{j\_22})_{fir}) \quad (10)$$

By (10), PR_4 and formulas (e), (g), (i), we have

$$<M, \alpha> |- \Box(pm_{j\_1} \Rightarrow \Diamond\ pm_{j\_19})$$

$$\text{CASE 2: } <M, \alpha> |-\Box (pm_{j\_5} \bullet pm_{j\_9} \bullet pm_{j\_18} \Rightarrow (t_{j\_23})_{fir}) \quad (11)$$

From (11), PR_4 and the structure of the subnet in Fig. 3, we obtain

$$<M, \alpha> |-\Box(t_{j\_9} \Rightarrow \circ (pm_{j\_7} \bullet pm_{j\_23})) \quad (12)$$

Here, there must be a deal result in pm_{j_7}, but it may also contain residual parts of the buying and selling deal data. We have

$$<M, \alpha> |-\Box(pm_{j\_7} \bullet pm_{j\_23} \Rightarrow (t_{j\_10})_{fir}+(t_{j\_10})_{fir} \bullet (t_{j\_5})_{fir} +(t_{j\_10})_{fir} \bullet (t_{j\_15})_{fir}+(t_{j\_10})_{fir} \bullet (t_{j\_5})_{fir} \bullet (t_{j\_15})_{fir}) \quad (13)$$

From trading rules and (13), t_{j_10} must be firable. If there is a residual buying (selling) deal datum, then $t_{j\_5}(t_{j\_15})$ is also firable. Therefore, whichever in the four cases exists, their deductive processes are similar. Now we only discuss the first case, i.e.,

$$<M, \alpha> |-\Box(pm_{j\_7} \bullet pm_{j\_23} \Rightarrow (t_{j\_10})_{fir}) \quad (14)$$

From (ST_{10}) and PR_4, we have

$$<M, \alpha> |-\Box(t_{j\_25} \Rightarrow \circ pm_{j\_15}) \quad (15)$$

From (15), if t_{j_7} is firable now, the inferring process of (7)-(15) is repeatedly done until the conclusion of this lemma is obtained. If t_{j_7} is not firable, a valid formula is given as followings by (ST_8) and (ST_9):

$$<M, \alpha> |-\Box(pm_{j\_15} \Rightarrow (t_{j\_19})_{fir}+(t_{j\_20})_{fir}) \quad (16)$$

(16), formulas (i), (g), (f) and PR_4, yield

$$<M, \alpha> |-\Box(t_{j\_19}+t_{j\_20} \Rightarrow \circ pm_{j\_19}) \quad (17)$$

By 4, 6-9, 11-17, formula (f) and PR_1, PR_2, we obtain

$$<M, \alpha> |-\Box(pm_{j\_1} \Rightarrow \Diamond\ pm_{j\_19})$$

Lemma 7 shows the liveness (no deadlock, no lockout) property in every processor making match system. The following theorem presents the global liveness property in processor processing systems.

Theorem 1: (If a processor begins to process the deal data of some kind of stock, then it will eventually finish making all matches of them and deliver the privilege to another kind of stock.) Let M be a marking reachable of TND from M_0, then for any firing sequence α from M and any i, $1 \le j \le k$, we have

$$<M, \alpha> \vdash \Box(p_i \bullet pv_{ji} \bullet \neg pro_i \Rightarrow \Diamond\, pv_{j(i+1)})$$

Proof: By the structure of the subnet in Fig. 2, PR4 and (ST_3), we have

$$<M, \alpha> \vdash \Box(t_{ji\_1} \Rightarrow \circ\, (pro_i \bullet p_{ji\_1})) \quad (18)$$

Here, if one of p_{i_4} and p_{i_8} is nonempty, then all residual deal data in p_{i_4} or p_{i_8} can be transferred to pm_{j_4} or pm_{j_8} by firing t_{ji_5} or t_{ji_6}, respectively. Therefore, we suppose that they are empty. From (18), PR4 and (ST_5), we have

$$<M, \alpha> \vdash \Box(t_{ji\_4} \Rightarrow \circ\, (pro_i \bullet pm_{j\_1} \bullet p_{ji\_2})) \quad (19)$$

By Lemma 7, (19) and formula (e), we have

$$<M, \alpha> \vdash \Box(pro_i \bullet pm_{j\_1} \bullet p_{ji\_2} \Rightarrow \Diamond\, (pro_i \bullet pm_{j\_19} \bullet p_{ji\_2})) \quad (20)$$

Similarly, we suppose that pm_{j_4} and pm_{j_8} are empty. Otherwise, they will be empty after firing t_{ji_7} or t_{ji_8}. Thus, from (20), PR4 and (ST_6), we have

$$<M, \alpha> \vdash \Box(t_{ji\_9} \Rightarrow \Diamond\, pv_{j(i+1)}) \quad (21)$$

From 18-21 and PR_1, PR_2, we obtain

$$<M, \alpha> \vdash \Box(p_i \bullet pv_{ji} \bullet \neg pro_i \Rightarrow \Diamond\, pv_{j(i+1)})$$

By means of the structures of the subnets in Fig. 1-3, Lemmas 2-7, Theorem 1 and their proofs, Theorems 2 and 3 can be given below. Their proofs are omitted to save space.

Theorem 2: TND is live, safe, fair and bounded.

Theorem 3: The dynamic behavior of TND is consistent with the functional requirements of DSTSs.

CONCLUSIONS

It is demonstrated further that TPNs can not only enhance the modeling and analysis power of PNs, but also compensate the shortcoming that PNs do not represent timing constraints, such as eventuality. In doing so, a DSTS with shared-variable DSM is adopted. It has been formally proved that the functional requirements of the system can be satisfied by the dynamic behavior of TPN model TND. Also, the correctness of the system is analyzed and verified based on its TPN model. A main defect of TPNs is the lack of variables for describing the values and types of data items. Therefore, the price and types of deal data could not be explicitly described in the TPN model. However, if these properties are required, we may use colored TPNs (Du and Jiang, 2004) to model and verify the systems.

In the subnet of processor pre-processing systems, every processor processes dynamically the deal data of all kinds of stocks. Processors may spend much more time to seek the deal data waiting to be processed. To cope with this problem, n kinds of stocks divide into k sets and every set is fixedly assigned to one processor. When some processor is at leisure, it will seek the deal data belonging to the set on the other processors. In this case, the analysis methods proposed in this paper are still valid. The performance analysis and evaluation of DSTSs will be investigated in future using stochastic PNs.

ACKNOWLEDGMENTS

This research was supported in part by the National Natural Science Foundation of China under Grants 60773034, 60534060 and 60473094; the Taishan Scholar Construction Project of Shandong Province, China; the National Basic Research Program of China (973 Program) under Grants 2007CB316502 and 2004CB318001-03; the Open Project of the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences under Grant SYSKF0804 and the Research foundation of East China University of Science and Technology.

REFERENCES

Berthomieu, B. and M. Diaz, 1991. Modeling and verification of time dependent systems using time petri nets. IEEE Trans. Software Eng., 17 (3): 259-273.

Du, Y.Y. and C.J. Jiang, 2004. Verifying functions in online stock trading systems. J. Comput. Sci. Technol., 19 (2): 203-212.

Du, Y.Y., C.J. Jiang and Y.B. Guo, 2006. Towards a formal model for grid architecture via petri nets. Inform. Technol. J., 5 (5): 833-841.

Du, Y.Y., C.J. Jiang and M.C. Zhou, 2007. Modeling and analysis of real-time cooperative systems using petri nets. IEEE Trans. Syst. Man Cybern. A Syst. Hum., 37 (5): 643-654.

Du, Y.Y., C.J. Jiang and M.C. Zhou, 2008. A petri nets based correctness analysis of internet stock trading systems. IEEE Trans. Syst. Man Cybern. C Applied Rev., 38 (1): 93-99.

Murata, T., 1989. Petri nets: Properties, analysis and applications. Proceedings IEEE, 77 (4): 541-580.

Suzuki, I. and H. Lu, 1989. Temporal petri nets and their application to modeling and analysis of a handshake daisy chain arbiter. IEEE Trans. Comput., 38 (5): 696-704.

Zheng, H. and Y.Y. Du, 2005. Petri nets-based modeling for dynamic stock trading systems. J. Comput. Inform. Syst., 1 (3): 543-548.

Zurawski, R., 1997. Verifying correctness of interfaces of design models of manufacturing systems using functional abstractions. IEEE Trans. Ind. Elect., 44 (3): 307-320.