ITC 1/46
Journal of Information Technology and Control
Vol. 46 / No. 1 / 2017
pp. 5-15
DOI 10.5755/j01.itc.46.1.13970

Proper Augmented Marked Graphs: Properties, Characterizations and Applications

Received 2017/01/13 | Accepted after revision 2016/02/17

http://dx.doi.org/10.5755/j01.itc.46.1.13970

Proper Augmented Marked Graphs: Properties, Characterizations and Applications

King Sing Cheung

The Open University of Hong Kong, 30 Good Shepherd Street, Homantin, Kowloon, Hong Kong

Corresponding author: kscheung@ouhk.edu.hk

Augmented marked graphs possess a special structure for modelling distributed systems with shared resources. Not only inheriting the desirable properties of augmented marked graphs such as on liveness and reversibility, proper augmented marked graphs also exhibit other desirable properties, including boundedness and conservativeness. However, proper augmented marked graphs have a rather complicated definition that inevitably undermines the usability in system modelling. In this paper, based on composition of live and bounded marked graphs, new characterizations for proper augmented marked graphs are devised. Through these characterizations, proper augmented marked graphs can be effectively used in modelling and analyzing conflicting processes of a distributed system. Applications to distributed transaction processing with shared resources are discussed.

KEYWORDS: Petri nets, marked graphs, augmented marked graphs, proper augmented marked graphs, distributed systems, shared-resource systems, component based systems, distributed transaction processing, systems integration.

Introduction

A subclass of Petri nets, augmented marked graph was first introduced by Chu and Xie for modelling systems with shared resources [1]. In the literature, thorough investigation on augmented marked graphs was mainly conducted by Cheung [2-6]. Having a special structure for representing shared resources, augmented marked graphs possess desirable properties pertaining to liveness and reversibility. According to Chu and Xie, an augmented marked graph is live and reversible if and only if every minimal siphon would never become empty [1]. More siphon-based and cycle-based characterizations were devised by Cheung, where a cycle-inclusion property was used for characterizing the liveness and reversibility [2, 3]. Transformation-based characterizations for bounded and conservative augmented marked graphs were introduced [4, 6]. There are also studies on the composition of augmented marked graphs and its applications to system integration [7-11].

Proper augmented marked graphs are a special type of augmented marked graphs, found by Cheung [6, 12]. Not only inheriting all the properties of augmented marked graphs, proper augmented marked graphs also possess more properties, including boundedness and conservativeness. However, like augmented marked graphs, proper augmented marked graphs have a rather complicated definition, thus adding difficulties in system modelling and analysis. This dilemma can be resolved by some characterizations of proper augmented marked graphs. In this paper, based on the composition of live and bounded marked graphs, a number of characterizations are proposed. With these characterizations, the processes or components of a system can be readily modelled as marked graphs, and then composed via their common resource places. The integrated PT-net so obtained is a proper augmented marked graph which represents the integrated whole of the processes or components.

In a distributed system, it is often that two or more concurrent processes compete for some shared resources. Owing to the existence of these conflicting processes, erroneous situations such as deadlock and capacity overflow may occur. In this paper, it is proposed to model the conflicting processes as marked graphs, and then, to compose them as a proper augmented graph which represents the integrated whole of the processes for analysis.

The rest of this paper is structured as follows. Section 2 states the definitions and properties of proper augmented marked graphs. New characterizations are proposed in Section 3. Section 4 then shows the modelling and analysis of conflicting processes using proper augmented marked graphs and their properties and characterizations. Section 5 describes the application to the analysis of distributed transaction processing systems with common shared resources, and illustrates with examples. Section 6 briefly concludes this paper. It is noted that readers are assumed to have basic knowledge on Petri nets [13-15].

Proper augmented marked graphs and their properties

Proper augmented marked graphs are a special type of augmented marked graphs [6, 12]. The definitions and known properties are summarized below.

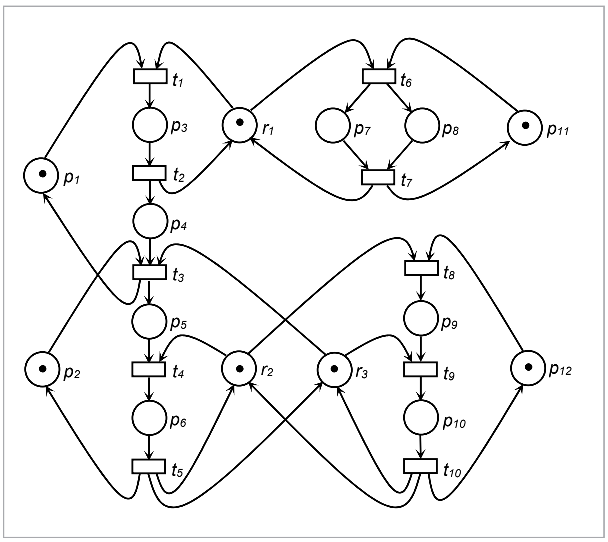
Definition 1. An augmented marked graph $(N, M_0; R)$ is a PT-net (N, M_0) with a specific subset of places R (called resource places), satisfying the following conditions: (a) Every place in R is marked by M_0. (b) The PT-net (N', M_0) obtained from $(N, M_0; R)$ by removing the places in R and their associated arcs is a marked graph. (c) For each $r \in R$, there exist $k_r > 1$ pairs of transitions $D_r = \{ \langle t_{s1}, t_{h1} \rangle, \langle t_{s2}, t_{h2} \rangle, ..., \langle t_{skr}, t_{hkr} \rangle \}$ such that $r^\bullet = \{ t_{s1}, t_{s2}, ..., t_{skr} \} \subseteq T$ and $^\bullet r = \{ t_{h1}, t_{h2}, ..., t_{hkr} \} \subseteq T$ and that, for each $\langle t_{si}, t_{hi} \rangle \in D_r$, there exists in N' an elementary path ρ_{ri} connecting t_{si} to t_{hi}. (d) In (N', M_0'), every cycle is marked and no ρ_{ri} is marked.

Definition 2. Let $(N, M_0; R)$ be an augmented marked graph to be transformed into a PT-net (N', M_0') as follows. For each place $r \in R$, where $D_r = \{ \langle t_{s1}, t_{h1} \rangle, \langle t_{s2}, t_{h2} \rangle, ..., \langle t_{skr}, t_{hkr} \rangle \}$, r is replaced by a set of places $Q = \{ q_1, q_2, ..., q_{kr} \}$, such that $M_0'[q_i] = M_0[r]$, $q_i^\bullet = \{ t_{si} \}$ and $^\bullet q_i = \{ t_{hi} \}$. (N', M_0') is called the R-transform of $(N, M_0; R)$.

Definition 3. Let $(N, M_0; R)$ be an augmented marked graph, and (N', M_0') be the R-transform of $(N, M_0; R)$. $(N, M_0; R)$ is a proper augmented marked graph if and only if every place in (N', M_0') belongs to a cycle.

Figure 1

A proper augmented marked graph



Property 1. A proper augmented marked graph *(N, M_0; R)* is live and reversible if and only if every R-siphon would never become empty [6, 12]. (Note : A R-siphon is a minimal siphon which contains at least one place in *R*.)

Property 2. A proper augmented marked graph is bounded and conservative [6, 12].

Figure 1 shows a proper augmented marked graph *(N, M_0; R)*, where $R = \{ r_1, r_2, r_3 \}$. *(N, M_0; R)* is bounded and conservative. However, it is neither live nor reversible since there exists a R-siphon $\{ r_2, r_3, p_6, p_{10} \}$, which would become empty on firing $\langle t_1, t_2, t_3, t_8 \rangle$.

Characterizations for proper augmented marked graphs

Based on the composition of live and bounded marked graphs, in the following, a number of new characterizations for proper augmented marked graphs are proposed.

Definition 4. Let $(N_1, M_{10}), (N_2, M_{20}), ..., (N_n, M_{n0})$ be PT-nets. Suppose $Q = \{ p_1, p_2, ..., p_k \}$ is a set of places that are common to the PT-nets, where $p_1, p_2, ..., p_k$ are marked. By fusing $p_1, p_2, ..., p_k$ into one single marked place q, the resulting net (N, M_0) is called the integrated PT-net obtained by composing $(N_1, M_{10}), (N_2, M_{20}), ..., (N_n, M_{n0})$ via the set of common places Q.

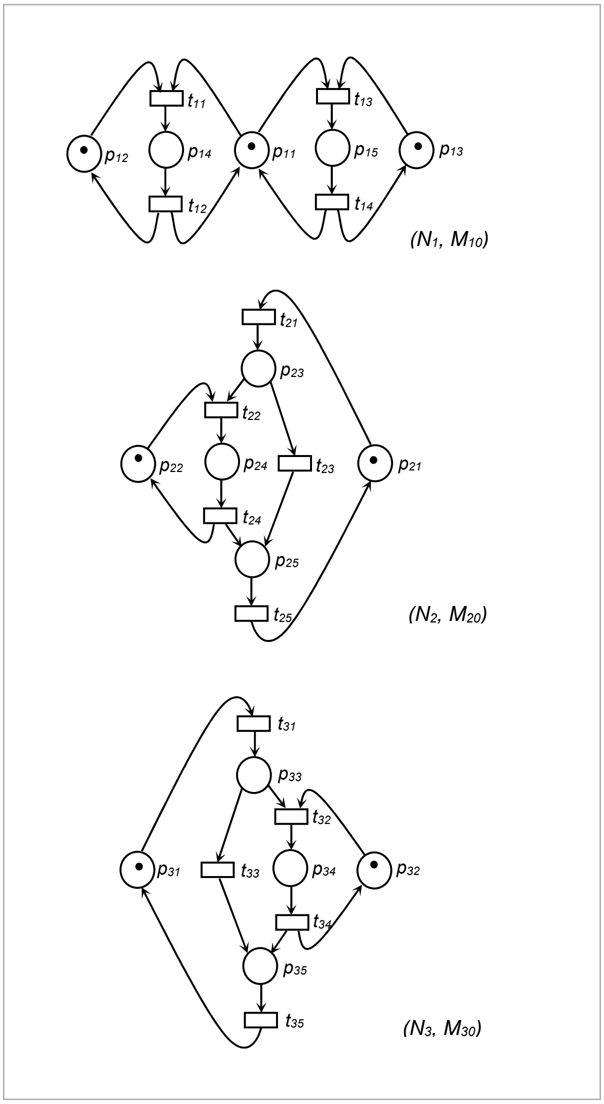
Proposition 1. *Let (N, M_0; R) be a proper augmented marked graph, and (N', M_0') be the R-transform of (N, M_0; R). (N', M_0') is structurally the composite PT-net of a set of disconnected, live and bounded marked graphs.*

Proof. Consider the transformation of *(N, M_0; R)* into *(N', M_0')*, as described in Definition 2. Let $R = \{ r_1, r_2, ..., r_n \}$. Each $r_i \in R$ is replaced by a set of marked places Q_i, for $i = 1, 2, ..., n$. For any place p in *(N', M_0')*, $| {}^\bullet p | = | p^\bullet | = 1$. Let γ be a cycle in *(N', M_0')*. There are two possible cases for γ. In case γ contains any place in $Q_1 \cup Q_2 \cup ... \cup Q_n$, γ is marked. In case γ does not contain any place in $Q_1 \cup Q_2 \cup ... \cup Q_n$, γ also exists in *(N, M_0; R)*. According to Cheung, every cycle in an augmented marked graph is marked [1, 2, 6]. Hence, γ is also marked. Then, for *(N', M_0')*, every place belongs to a cycle and every cycle is marked, thus fulfilling the conditions of live and bounded marked graphs. *(N', M_0')* is structurally a live and bounded marked graph or a composite of a set of live and bounded marked graphs.

Figure 2 shows three PT-nets, (N_1, M_{10}), (N_2, M_{20}) and

Figure 2

Three PT-nets with common places



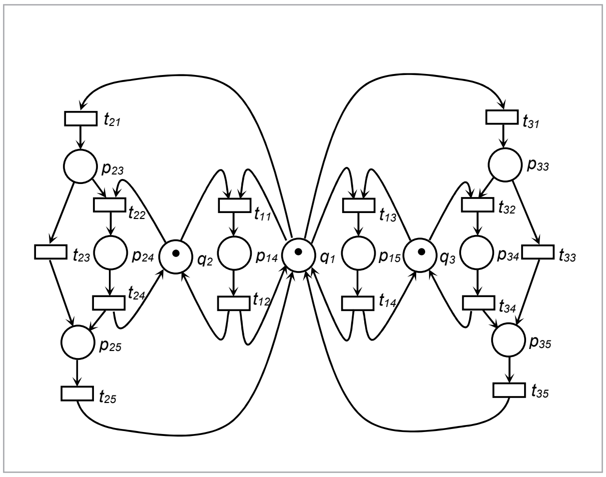
(N_3, M_{30}). Suppose they have some common places, $Q_1 = \{ p_{11}, p_{21}, p_{31} \}$, $Q_2 = \{ p_{12}, p_{22} \}$ and $Q_3 = \{ p_{13}, q_{32} \}$. Figure 3 shows the integrated PT-net (N, M_0) obtained by composing (N_1, M_{10}), (N_2, M_{20}) and (N_3, M_{30}) via Q_1, Q_2 and Q_3, where q_1, q_2 and q_3 are the fused common places, respectively.

Proposition 2. A proper augmented marked graph (N, M_0; R) is the integrated PT-net obtained after composing a set of live and bounded marked graphs via their common places, where R is the set of fused places.

Proof. It follows from Proposition 1. For any place *p*

Figure 3

The integrated PT-net obtained by composing the PT-nets in Figure 2 via their common places



not involved in the fusing, $|{}^\bullet p| = |p^\bullet| = 1$. For any place r involved in the fusing, $|{}^\bullet r| = |r^\bullet| = k$. R is the set of fused places.

Proposition 2 offers a more concise definition for proper augmented marked graphs. It can be effectively used in characterizing the boundedness and conservativeness of proper augmented marked graphs, as well as the property-preserving composition of proper augmented marked graphs.

Lemma 1. *Let $N = \langle P, T, F \rangle$ be a PT-net, $N' = \langle P', T', F' \rangle$ be the PT-net obtained from N after fusing a set of places $Q = \{ q_1, q_2, ..., q_n \} \subseteq P$ into one single place $r \in P'$. If there exists a place invariant α of N such that $a[q_1] = \alpha[q_2] = ... = \alpha[q_n] = k \geq 0$, then there also exists a place invariant α' of N' such that $\alpha'[r] = k$ and $\alpha'[s] = \alpha[s]$ for any $s \in P' \setminus \{ r \} = P \setminus Q$.*

Proof. Since N' is obtained from N by fusing $Q = \{ q_1, q_2, ..., q_n \}$ into r, we have $P' = (P \setminus Q) \cup \{ r \}$. Let V be the incidence matrix of N. Then, the incidence matrix V' of N' satisfies that $V'[r] = \Sigma_{i=1,2,...,n} V[q_i]$ and $V'[s] = V[s]$ for any $s \in P' \setminus \{ r \} = P \setminus Q$. Since α is a place invariant of N, $\alpha V = 0$. Let α' be a place vector of N' such that $\alpha'[r] = \alpha[q_1] = \alpha[q_2] = ... = \alpha[q_n] = k$ and $\alpha'[s] = \alpha[s]$ for every $s \in P' \setminus \{ r \} = P \setminus Q$. Then, $\alpha' V' = \alpha'[r]V'[r] + \Sigma_{p \in (P' \setminus \{r\})} \alpha'[p]V'[p] = \Sigma_{i=1,2,...,n} \alpha[q_i]V[q_i] + \Sigma_{p \in (P \setminus Q)} \alpha[p]V[p] = \alpha V = 0$. Hence, α' is a place invariant of N'.

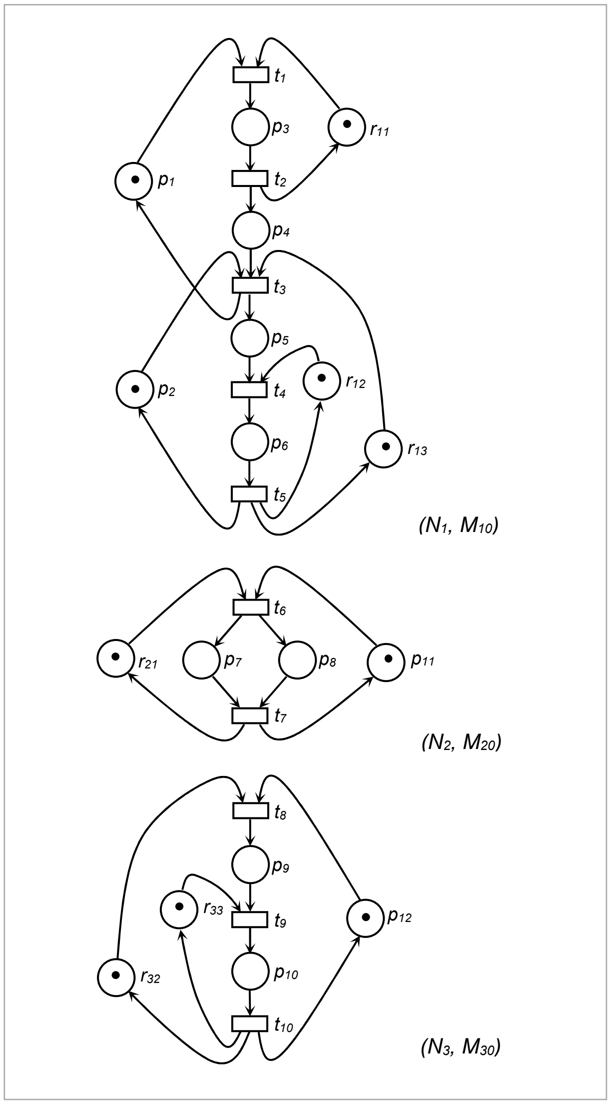
Proposition 3. *A proper augmented marked graph is bounded and conservative.*

Proof. According to Proposition 2, a proper augmented marked graph $(N, M_0; R)$ is an integrated PT-net obtained after composing a set of live and bounded marked graphs $\{ (N_1, M_{10}), (N_2, M_{20}), ..., (N, M_{k0}) \}$ by fusing some marked places. Let (N', M_0') be the composite PT-net of $(N_1, M_{10}), (N_2, M_{20}), ..., (N, M_{k0})$. Since each (N_i, M_{i0}) is bounded, there exists a place invariant α' in (N', M_0') such that $\alpha' = k > 0$. According to Lemma 1, there also exists a place invariant α in $(N, M_0; R)$ such that $\alpha = k > 0$. Hence, $(N, M_0; R)$ is bounded and conservative.

Proposition 4. *The integrated PT-net obtained by composing a set of live and bounded marked graphs via their common places is a proper augmented marked*

Figure 4

A set of live and bounded marked graphs



graph $(N, M_0; R)$, where R is the set of fused places.

Proof. It directly follows from Propositions 2 and 3.

Figure 4 shows a set of live and bounded marked graphs $\{(N_1, M_{10}), (N_2, M_{20}), (N_3, M_{30})\}$. They are composed by fusing r_{11} and r_{21} into one single place r_1, r_{12} and r_{32} into r_2, and r_{13} and r_{33} into r_3. The resulting PT-net is a proper augmented marked graph $(N, M_0; R)$, where $R = \{r_1, r_2, r_3\}$, as shown in Figure 1.

Modelling and analysis of conflicting processes

This section discusses how proper augmented marked graphs can be effectively used in the modelling and analysis of conflicting processes.

Typically in a distributed system, a number of concurrent processes compete for some shared resources. Erroneous situations occur when two or more processes are each waiting for the other to finish and neither ever does. The processes will continue to wait endlessly, resulting into deadlocks. There are also erroneous situations where resources exceed their capacity limits, thus causing capacity overflow. These processes are called conflicting processes.

In system integration, especially for distributed systems with concurrent processes competing for shared resources, one difficult challenge is to identify any erroneous situations such as deadlock and capacity overflow. This can be approached by using proper augmented marked graphs and their properties and characterizations. Consider a set of conflicting processes, competing for shared resources $R = \{r_1, r_2, ..., r_k\}$. Steps for modelling and analysis are outlined below.

Step 1. Model each process as a marked graph (N_i, M_{i0}), where any shared resource to be used is represented as a marked place called resource place. For a total of n processes, we have a set of marked graphs $\{(N_1, M_{10}), (N_2, M_{20}), ..., (N_n, M_{n0})\}$.

Step 2. Check if each (N_i, M_{i0}) is live and bounded. (N_i, M_{i0}) is live and bounded if and only if every place in belongs to a cycle and every cycle is marked [16].

Step 3. Suppose $(N_1, M_{10}), (N_2, M_{20}), ..., (N_n, M_{n0})$ are live and bounded. Compose them via their common resource places. According to Proposition 2, the integrated PT-net is a proper augmented marked graph $(N, M_0; R)$, where $R = \{r_1, r_2, ..., r_k\}$ denotes the shared resources.

Step 4. Analyze the properties of $(N, M_0; R)$, which represents an integration of the conflicting processes. According to Property 2, $(N, M_0; R)$ is bounded and conservative. Based on Property 1, $(N, M_0; R)$ is live and reversible if and only if every R-siphon would never become empty.

Suppose there is a distributed system with shared resources r_1, r_2 and r_3, where r_1 is shared by processes C_1 and C_2, and r_2 and r_3 are shared by processes C_1 and C_3. As shown in Figure 4, C_1, C_2 and C_3 are modelled as live and bounded marked graphs (N_1, M_{10}), (N_2, M_{20}) and (N_3, M_{30}), respectively. Referring to the same resource, r_{11} in (N_1, M_{10}) and r_{12} in (N_2, M_{20}) are fused as one single place r_1. Likewise, r_{12} in (N_1, M_{10}) and r_{32} in (N_3, M_{30}) are fused as r_2, and r_{13} in (N_1, M_{10}) and r_{33} in (N_3, M_{30}) are fused as r_3. According to Proposition 4, the integrated PT-net is a proper augmented marked graph $(N, M_0; R)$, where $R = \{r_1, r_2, r_3\}$, as shown in Figure 1.

Application to distributed transaction processing

In a distributed transaction processing system, there involves many concurrent processes which compete for some shared resources such as common data objects. Whenever a process needs to access a common data object, it would attempt to lock the data object for exclusive usages, either read or update. The data object is unlocked after completion of the read or update transactions. While the data object is being locked, accesses from other processes are prohibited. These processes have to wait until the data object is unlocked. Deadlocks may occur, where two or more processes are each waiting for the other to finish the read or update operations of common data objects, and thus neither ever does.

It is always important in system design to identify any possible deadlock situations. In system terminology, liveness is the property where deadlock situations would never occur. Hence, it is one of the design objectives to verify if a system is live, that is, free from deadlock situations. Proper augmented marked graphs can be effectively applied to solve this problem by following the steps described in Section 4.

Example 1. Consider a typical distributed transaction processing system which involves a number of concurrent processes, accessing some common data objects. Among other processes, there are 3 concurrent processes each needs to access 2 common data objects (namely, O_1 and O_2) in processing some transactions. A functional description of the processes is as follows.

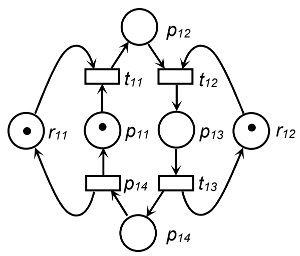
Process 1. At its initial state, the process intends to access O_1. Once O_1 is available, it is locked to prevent accesses from other processes. The process enters to a state, intending to access O_2. Once O_2 is available, it is locked by the process too. Update transactions on both O_1 and O_2 are then processed. After finishing these update transactions, the process releases O_2. There are some further update transactions on O_1, after which O_1 is released.

Process 2. At its initial state, the process intends to access O_2. Once O_2 is available, it is locked to prevent accesses from other processes. The process enters to a state, intending to access O_1. Once O_1 is available, it is locked by the process too. Update transactions on both O_1 and O_2 are then processed. After finishing these update transactions, the process releases O_1. There are some further update transactions on O_2, after which O_2 is released.

Process 3. At its initial state, the process intends to

Figure 5

Modelling process 1 as a marked graph (N_1, M_{10})



List of places

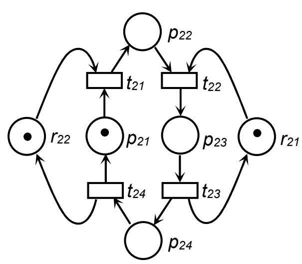
p_{11}	process 1 is at initial state
p_{12}	process 1 holds O_1 and waits for O_2
p_{13}	process 1 executes transactions with O_1 and O_2
p_{14}	process 1 executes transactions with O_1
r_{11}	O_1 is available
r_{12}	O_2 is available

List of transitions

t_{11}	process 1 requests O_1
t_{12}	process 1 requests O_2
t_{13}	process 1 releases O_2
t_{14}	process 1 releases O_1

Figure 6

Modelling process 2 as a marked graph (N_2, M_{20})

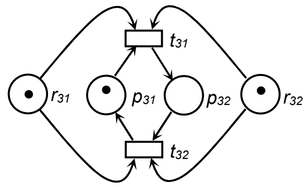


List of places

p_{21}	process 2 is at initial state
p_{22}	process 2 holds O_2 and waits for O_1
p_{23}	process 2 executes transactions with O_1 and O_2
p_{24}	process 2 executes transactions with O_2
r_{21}	O_1 is available
r_{22}	O_2 is available

List of transitions

t_{21}	process 2 requests O_2
t_{22}	process 2 requests O_1
t_{23}	process 2 releases O_1
t_{24}	process 2 releases O_2



List of places

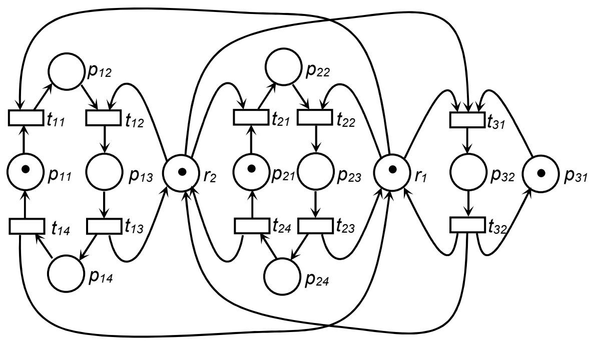
p_{31}	process 3 is at initial state
p_{32}	process 3 executes transactions with O_1 and O_2
r_{31}	O_1 is available
r_{32}	O_2 is available

List of transitions

t_{31}	process 3 requests O_1 and O_2
t_{32}	process 3 releases O_1 and O_2

Figure 7

Modelling process 3 as a marked graph (N_3, M_{30})



List of places

p_{11}	process 1 is at initial state
p_{12}	process 1 holds O_1 and waits for O_2
p_{13}	process 1 executes transactions with O_1 and O_2
p_{14}	process 1 executes transactions with O_1
p_{21}	process 2 is at initial state
p_{22}	process 2 holds O_2 and waits for O_1
p_{23}	process 2 executes transactions with O_1 and O_2
p_{24}	process 2 executes transactions with O_2
p_{31}	process 3 is at initial state
p_{32}	process 3 executes transactions with O_1 and O_2
r_{11}	O_1 is available
r_{12}	O_2 is available

List of transitions

t_{11}	process 1 requests O_1
t_{12}	process 1 requests O_2
t_{13}	process 1 releases O_2
t_{14}	process 1 releases O_1
t_{21}	process 2 requests O_2
t_{22}	process 2 requests O_1
t_{23}	process 2 releases O_1
t_{24}	process 2 releases O_2
t_{31}	process 3 requests O_1 and O_2
t_{32}	process 3 releases O_1 and O_2

Figure 8

The proper augmented marked graph $(N, M_0; R)$ obtained after composing (N_1, M_{10}), (N_2, M_{20}) and (N_3, M_{30}) in Figures 5, 6 and 7

access both O_1 and O_2. Once both O_1 and O_2 are available, they are locked to prevent accesses from other processes. Update transactions on both O_1 and O_2 are then processed. After finishing the update transactions, the process releases O_1 and O_2 simultaneously.

Processes 1, 2 and 3 are represented by the marked graphs (N_1, M_{10}), (N_2, M_{20}) and (N_3, M_{30}), as shown in Figures 5, 6 and 7 respectively. They are live and bounded.

The next step is to compose (N_1, M_{10}), (N_2, M_{20}) and (N_3, M_{30}) via Q_1 and Q_2. According to Proposition 4, the integrated PT-net so obtained is a proper augmented marked graph.

Figure 8 shows the proper augmented marked graph $(N, M_0; R)$, where $R = \{r_1, r_2\}$, after fusing r_{11}, r_{21} and r_{31} as one single place r_1, and r_{21}, r_{22} and r_{23} as r_2. $(N, M_0; R)$ represents the integrated whole of the conflicting processes. As $(N, M_0; R)$ is a proper augmented marked graph, according to Property 2, it is bounded and conservative. Besides, there exists a R-siphon $\{ r_1, r_2, p_{13}, p_{14}, p_{23}, p_{24}, p_{32} \}$ which would become empty after firing $\langle t_{11}, t_{21} \rangle$. According to Property 1, $(N, M_0; R)$ is neither live nor reversible. Deadlock will occur after firing $\langle t_{11}, t_{21} \rangle$. From this, it is concluded that deadlock would occur among the conflicting processes.

It is also shown that, even though the processes are individually live and reversible, the integrated whole may not be live nor reversible. However, in some cases, the integrated whole can be live and reversible, as illustrated in the following example.

Example 2. This example is a revised version of Example 1. Processes 1, 2 and 3 are revised as follows.

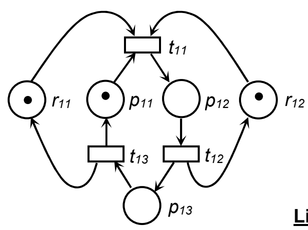
Revised Process 1. At its initial state, the process intends to access both O_1 and O_2. Once both O_1 and O_2 are available, they are locked to prevent accesses from other processes. Update transactions on both O_1 and O_2 are then processed. After finishing these update transactions, the process releases O_2. There are some further update transaction on O_1, after with O_1 is released.

Revised Process 2. At its initial state, the process intends to access O_2. Once O_2 is available, it is locked to prevent accesses from other processes. The process enters to a state, intending to access O_1. Once O_1 is available, it is locked by the process too. Update transactions on both O_1 and O_2 are then processed. After finishing these update transactions, the process releases O_1 and O_2 simultaneously.

Revised Process 3. At its initial state, the process intends to access both O_1 and O_2. Once both O_1 and O_2 are available, they are locked to prevent accesses from other processes. Update transactions on both O_1 and O_2 are then processed. After finishing these update transactions, the process releases O_1. There are some further update transactions on O_2, after which O_2 is released.

Figure 9

Modelling the revised process 1 as a marked graph (N_1', M_{10}')



List of places

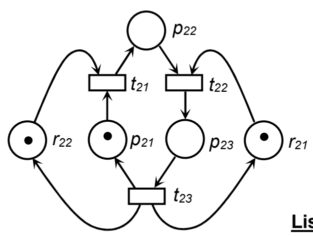
p_{11} process 1 is at initial state
p_{12} process 1 executes transactions with O_1 and O_2
p_{13} process 1 executes transaction with O_1
r_{11} O_1 is available
r_{12} O_2 is available

List of transitions

t_{11} process 1 requests O_1 and O_2
t_{12} process 1 releases O_2
t_{13} process 1 releases O_1

Figure 10

Modelling the revised process 2 as a marked graph (N_2', M_{20}')

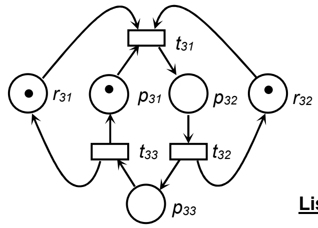


List of places

p_{21} process 2 is at initial state
p_{22} process 2 holds O_2 and waits for O_1
p_{23} process 2 executes transactions with O_1 and O_2
r_{21} O_1 is available
r_{22} O_2 is available

List of transitions

t_{21} process 2 requests O_2
t_{22} process 2 requests O_1
t_{23} process 2 releases O_1 and O_2



List of places

p_{31}	process 3 is at initial state
p_{32}	process 3 executes transactions with O_1 and O_2
p_{33}	process 3 executes transactions with O_1
r_{31}	O_1 is available
r_{32}	O_2 is available

List of transitions

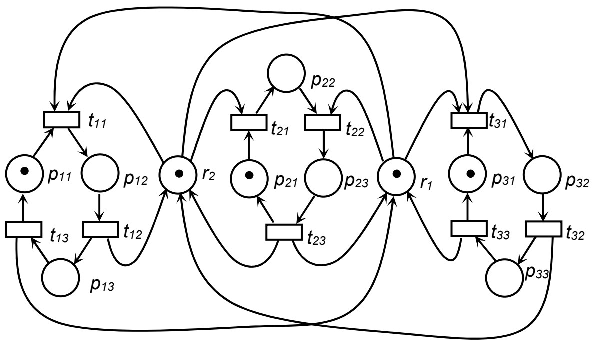
t_{31}	process 3 requests O_1 and O_2
t_{32}	process 3 releases O_2
t_{33}	process 3 releases O_1

Figure 11

Modelling the revised process 3 as a marked graph (N_3', M_{30}')

The revised processes 1, 2 and 3 are represented by the marked graphs $(N_1‘, M_{10}‘)$, $(N_2‘, M_{20}‘)$ and $(N_3‘, M_{30}‘)$, as shown in Figures 9, 10 and 11 respectively. They are live and bounded. $(N_1‘, M_{10}‘)$, $(N_2‘, M_{20}‘)$ and $(N_3‘, M_{30})$ are now composed via Q_1 and Q_2. Figure 12 shows the proper augmented marked graph $(N‘, M_0‘; R‘)$ where $R‘ = \{r_1, r_2\}$, after fusing r_{11}, r_{21} and r_{31} as one single place r_1, and r_{21}, r_{22} and r_{23} as r_2.

$(N', M_0'; R')$ represents the integrated whole of the conflicting processes. As $(N', M_0'; R')$ is a proper augmented marked graph, according to Property 2, it is bounded and conservative. Besides, every R-siphon in $(N', M_0'; R')$ would never become empty. According to Property 1, $(N', M_0'; R')$ is live and reversible. From this, it is concluded that the conflicting processes are free from deadlock and capacity overflow.



List of places

p_{11}	process 1 is at initial state
p_{12}	process 1 executes transactions with O_1 and O_2
p_{13}	process 1 executes transaction with O_1
p_{21}	process 2 is at initial state
p_{22}	process 2 holds O_2 and waits for O_1
p_{23}	process 2 executes transactions with O_1 and O_2
p_{31}	process 3 is at initial state
p_{32}	process 3 executes transactions with O_1 and O_2
p_{33}	process 3 executes transactions with O_1
r_{11}	O_1 is available
r_{12}	O_2 is available

List of transitions

t_{11}	process 1 requests O_1 and O_2
t_{12}	process 1 releases O_2
t_{13}	process 1 releases O_1
t_{21}	process 2 requests O_2
t_{22}	process 2 requests O_1
t_{23}	process 2 releases O_1 and O_2
t_{31}	process 3 requests O_1 and O_2
t_{32}	process 3 releases O_2
t_{33}	process 3 releases O_1

Figure 12

The proper augmented marked graph $(N', M_0'; R')$ obtained after composing (N_1', M_{10}'), (N_2', M_{20}') and (N_3', M_{30}') in Figures 9, 10 and 11

Conclusions

Augmented marked graphs and proper augmented marked graphs possess a special structure as well as many desirable properties pertaining to liveness, boundedness, reversibility and conservativeness. They are useful for modelling and analyzing distributed systems with shared resources.

Based on composition of live and bounded marked graphs, new characterizations for proper augmented marked graphs are proposed. Conflicting processes of a distributed system can be first modelled as live and bounded marked graphs, and then composed via common resource places to form a proper augmented marked graph which represents the integrated whole. As proper augmented marked graphs are bounded and conservative, it is assured that capacity overflow would never occur. By checking R-siphons, liveness and reversibility can be effectively analyzed.

As compared to other well-known subclasses of Petri nets such as state machines, marked graphs, free choice nets and asymmetric choice nets, augmented marked graphs or proper augmented marked graphs are not widely used in system modelling and analysis despite possessing many desirable properties pertaining to liveness, boundedness, conservativeness and reversibility. This is because of their complicated definition which is rather difficult to comprehend, thus undermining the usability. There is also a lack of simple but formal methodology for modelling, integrating or analyzing conflicting processes or components using augmented marked graphs or proper augmented marked graphs.

The problems can be resolved by characterizing proper augmented marked graphs by the composition of live and bounded marked graphs. With the characterizations, conflicting processes of a distributed system can be readily modelled, composed and analyzed. This paper provides a theoretical foundation of these characterizations, and shows the modelling and analysis using typical examples of distributed transaction processing.

Acknowledgments

The author would like to thank the anonymous reviewers for their helpful suggestions.

References

1. F. Chu, X. Xie. Deadlock Analysis of Petri Nets using Siphons and Mathematical Programming. IEEE Transactions on Robotics and Automation, 1997, 13(6), 793-804. https://doi.org/10.1109/70.650158
2. K. S. Cheung. New Characterization for Live and Reversible Augmented Marked Graphs. Information Processing Letters, 2004, 92(5), 239-243. https://doi.org/10.1016/j.ipl.2004.05.018
3. K. S. Cheung, K. O. Chow. Cycle-Inclusion Property of Augmented Marked Graphs. Information Processing Letters, 2005, 94(6), 271-276. https://doi.org/10.1016/j.ipl.2005.02.011
4. K. S. Cheung. Boundedness and Conservativeness of Augmented Marked Graphs. IMA Journal of Mathematical Control and Information, 2007, 24(2), 235-244. https://doi.org/10.1093/imamci/dnl019
5. C. L. Chen, S. C. Chin, H. C. Yen. Reachability Analysis of Augmented Marked Graphs via Integer Linear Programming. Computer Journal, 2010, 53(6), 623-633. https://doi.org/10.1093/comjnl/bxp003
6. K. S. Cheung. Augmented Marked Graphs. Springer, 2014.
7. H. J. Huang, L. Jiao, T. Y. Cheung. Property-Preserving Composition of Augmented Marked Graphs that Share Common Resources. IEEE Proceedings of the International Conference on Robotics and Automation, 2003, 1446-1451. https://doi.org/10.1109/robot.2003.1241795
8. K. S. Cheung. A Synthesis Method for Designing Shared-Resource Systems. Computing and Informatics, 2005, 24(6), 629-653.
9. K. S. Cheung. Modelling and Analysis of Manufacturing Systems Using Augmented Marked Graphs. Information Technology and Control, 2006, 35(1), 19-26.
10. K. S. Cheung. Composition of Augmented Marked Graphs and Its Application to Component-Based System Design. Information Technology and Control, 2007, 36(3), 310-317.
11. K. S. Cheung. A Formal Method for Synthesizing Components of Shared Resource Systems. International

Journal of Computer Systems Science and Engineering, 2007, 22(6), 349-358.

12. K. S. Cheung. Component-Based System Integration using Proper Augmented Marked Graphs. In: X. Zhang, H. Liu, Z. Chen, N. Wang (eds.), Intelligent Robotics and Applications, Lecture Notes in Artificial Intelligence, Springer, 2014, 8917, 498-509. https://doi.org/10.1007/978-3-319-13966-1_49
13. W. Reisig. Petri Nets: An Introduction. Springer, 1985. https://doi.org/10.1007/978-3-642-69968-9
14. T. Murata. Petri Nets: Properties, Analysis and Applications. Proceedings of IEEE, 1989, 77(4), 541-580. https://doi.org/10.1109/5.24143
15. W. Reisig. Understanding Petri Nets. Springer, 2013. https://doi.org/10.1007/978-3-642-33278-4
16. J. Desel, J. Esparza. Free Choice Petri Nets. Cambridge University Press, 1995. https://doi.org/10.1017/CBO9780511526558

Summary / Santrauka

Augmented marked graphs possess a special structure for modelling distributed systems with shared resources. Not only inheriting the desirable properties of augmented marked graphs such as on liveness and reversibility, proper augmented marked graphs also exhibit other desirable properties, including boundedness and conservativeness. However, proper augmented marked graphs have a rather complicated definition that inevitably undermines the usability in system modelling. In this paper, based on composition of live and bounded marked graphs, new characterizations for proper augmented marked graphs are devised. Through these characterizations, proper augmented marked graphs can be effectively used in modelling and analyzing conflicting processes of a distributed system. Applications to distributed transaction processing with shared resources are discussed.

Papildyti žymėtieji grafai turi specialią struktūrą paskirstytųjų sistemų su bendraisiais ištekliais kūrimui. Be tokių pageidaujamų papildytų žymėtųjų grafų savybių, kaip gyvumas ir grįžtamumas, tinkamai papildyti žymėtieji grafai taip pat turi kitas trokštamas savybes – tokias, kaip ribotumas ir konservatyvumas. Deja, tinkamai papildytųjų žymėtųjų grafų apibūdinimas sudėtingas, ir tai neišvengiamai kenkia jų panaudojimui sistemų modeliavime. Šiame straipsnyje, pagrįstame gyvųjų ir ribotųjų grafų sudarymu, sukuriamos naujos charakteristikos tinkamai papildytiems žymėtiesiems grafams. Per tokį charakterizavimą, tinkamai papildyti žymėtieji grafai gali būti efektyviai panaudoti paskirstytos sistemos modeliavime ir jos konfliktinių procesų analizėje. Straipsnyje aptariamas ir paskirstytųjų transakcijų apdorojimo su bendraisiais ištekliais pritaikymas.