Compositional Synthesis of Distributed System Components based on Augmented marked Graphs

K.S. Cheung
University of Hong Kong
Pokfulam, Hong Kong
E-mail : ks.cheung@hku.hk

Note : This paper is an extended version of the author's earlier paper in the 7th International Symposium on Advanced Processing Technologies.

ABSTRACT

Augmented marked graphs possess a special structure for modelling common resources as well as some desirable properties pertaining to liveness, boundedness, reversibility and conservativeness. This paper investigates the property-preserving composition of augmented marked graphs for the synthesis of distributed systems. It is proposed that distributed system components are specified as augmented marked graphs. An integrated system is obtained by composing these augmented marked graphs via their common resource places. Based on preservation of properties, liveness, boundedness, reversibility and conservativeness of the system can be readily derived. This contributes to resolve the problem of ensuring design correctness in the composition of distributed system components.

Keywords : Petri nets, augmented marked graphs, distributed systems, component-based system design

1. INTRODUCTION

In the past decade, component-based system design has emerged as a promising paradigm to meet the ever increasing needs for managing system complexity and maximising re-use as well as for deriving software engineering into standards. When applied to distributed systems which usually involve some concurrent (parallel) and asynchronous processes, one need to be aware that errors such as deadlock and capacity overflow may occur. Even though the components are correct in the sense that they are live (implying freeness of deadlock), bounded (implying absence of capacity overflow) and reversible (implying the capability of being reinitialised from any reachable states), the integrated system may not be correct, especially as competition of common resources exists.

This paper investigates the component-based approach to synthesising distributed systems, with a focus on the preservation of properties. Based on the property-preserving composition of augmented marked graphs, we propose a formal method for synthesising distributed system components into an integrated system whose design correctness can be readily derived and verified.

A subclass of Petri nets, augmented marked graphs possess a structure especially for modelling common resources. They also possess some desirable properties pertaining to deadlock-freeness, liveness, boundedness, reversibility and conservativeness. Chu and Xie investigated deadlock-freeness, liveness and reversibility using siphons and mathematical programming [1]. We further proposed some siphon-based and cycle-based characterisations for live and reversible augmented marked graphs, and some transform-based characterisations for bounded and conservative augmented marked graphs [2, 3, 4]. Apart from these, composition of augmented marked graphs via common resource places was preliminarily studied [5, 6, 7].

In this paper, after a brief review of augmented marked graphs, we investigate the composition of augmented marked graphs via common resource places and specifically show that this composition preserves boundedness and conservativeness whereas liveness and reversibility can be preserved under a pretty simple condition. The results are then applied to the composition of distributed system components, where liveness, boundedness, reversibility and conservativeness of the integrated system can be readily derived. These will be illustrated using examples.

The rest of this paper is organised as follows. Section 2 provides the preliminaries to be used in this paper. Section 3 introduces augmented marked graphs and summarises their known properties. In Section 4, we present the composition of augmented marked graphs with a special focus on the preservation of properties. Section 5 shows its application to the composition of distributed system components. Section 6 briefly concludes this paper.

2. PRELIMINARIES

This section provides the preliminaries for readers who are not familiar with Petri nets [8, 9, 10].

A place-transition net (PT-net) is a directed graph consisting of two sorts of nodes called places and transitions, such that no arcs connect two nodes of the same sort. Graphically, a place is denoted by a circle, a transition by a box, and an arc by a directed line. A Petri net is a PT-net with tokens assigned to its places, and the token distribution over its places is denoted by a marking.

Definition 2.1. A place-transition net (PT-net) is a 4-tuple $N = \langle P, T, F, W \rangle$, where P is a set of places, T is a set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is a flow relation and $W : F \rightarrow \{ 1, 2, ... \}$ is a weight function. N is said to be ordinary if and only if the range of W is $\{ 1 \}$. An ordinary PT-net is usually written as $\langle P, T, F \rangle$.

Definition 2.2. Let $N = \langle P, T, F, W \rangle$ be a PT-net. For $x \in (P \cup T)$, ${}^\bullet x = \{ y \mid (y, x) \in F \}$ and $x^\bullet = \{ y \mid (x, y) \in F \}$ are called the pre-set and post-set of x, respectively. For $X = \{ x_1, x_2, \ldots, x_n \} \subseteq (P \cup T)$, ${}^\bullet X = {}^\bullet x_1 \cup {}^\bullet x_2 \cup \ldots \cup {}^\bullet x_n$ and $X^\bullet = x_1^\bullet \cup x_2^\bullet \cup \ldots \cup x_n^\bullet$ are called the pre-set and post-set of X, respectively.

Definition 2.3. For a PT-net $N = \langle P, T, F, W \rangle$, a path is a sequence of nodes $\langle x_1, x_2, ..., x_n \rangle$, where $(x_i, x_{i+1}) \in F$ for i = 1, 2, ..., n-1. A path is said to be elementary if and only if it does not contain the same node more than once.

Definition 2.4. For a PT-net $N = \langle P, T, F, W \rangle$, a cycle is a sequence of places $\langle p_1, p_2, ..., p_n \rangle$ such that $\exists\, t_1, t_2, ..., t_n \in T : \langle p_1, t_1, p_2, t_2, ..., p_n, t_n \rangle$ forms an elementary path and $(t_n, p_1) \in F$.

Definition 2.5. For a PT-net $N = \langle P, T, F, W \rangle$, a marking is a function $M : P \rightarrow \{ 0, 1, 2, ... \}$ where M(p) is the number of tokens in p. (N, M_0) represents N with an initial marking M_0.

Definition 2.6. For a PT-net N = ⟨ P, T, F, W ⟩, a transition t is said to be enabled at a marking M if and only if ∀ p ∈ $^\bullet$t : M(p) ≥ W(p,t). On firing t, M is changed to M' such that ∀ p ∈ P : M'(p) = M(p) - W(p,t) + W(t,p). In notation, M [N,t⟩ M' or M [t⟩ M'.

Definition 2.7. For a PT-net (N, M_0), a sequence of transitions σ = ⟨ t_1, t_2, ..., t_n ⟩ is called a firing sequence if and only if M_0 [t_1⟩ ... [t_n⟩ M_n. In notation, M_0 [N,σ⟩ M_n or M_0 [σ⟩ M_n.

Definition 2.8. For a PT-net (N, M_0), a marking M is said to be reachable if and only if there exists a firing sequence σ such that M_0 [σ⟩ M. In notation, M_0 [N,*⟩ M or M_0 [*⟩ M. [N, M_0⟩ or [M_0⟩ represents the set of all reachable markings of (N, M_0).

Definition 2.9. Let N = ⟨ P, T, F, W ⟩ be a PT-net, where P = { p_1, p_2, ..., p_m } and T = { t_1, t_2, ..., t_n }. The incidence matrix of N is an m × n matrix V whose typical entry v_{ij} = $W(p_i,t_j)$ - $W(t_j,p_i)$ is the change in number of tokens in p_i after firing t_j once, for i = 1, 2, ..., m and j = 1, 2, ..., n.

Definition 2.10. For a PT-net (N, M_0), a transition t is said to be live if and only if ∀M ∈ [M_0⟩, ∃M' : M [*⟩ M' [t⟩. (N, M_0) is said to be live if and only if every transition is live.

Definition 2.11. For a PT-net (N, M_0), a place p is said to be k-bounded (bounded) if and only if ∀ M ∈ [M_0⟩ : M(p) ≤ k, where k > 0. (N, M_0) is said to be k-bounded (bounded) if and only if every place is k-bounded.

Definition 2.12. A PT-net (N, M_0) is said to be safe if and only if every place is 1-bounded.

Definition 2.13. A PT-net (N, M_0) is said to be reversible if and only if ∀ M ∈ [M_0⟩ : M [*⟩ M_0.

Definition 2.14. A PT-net is said to be well-behaved if and only if it is live, bounded and reversible.

Definition 2.15. A PT-net N = ⟨ P, T, F, W ⟩ is said to be conservative if and only if there exists a |P|-vector α > 0 such that αV = 0, where V is the incidence matrix of N.

Figure 1 shows a PT-net which is live, bounded, safe, reversible and conservative.

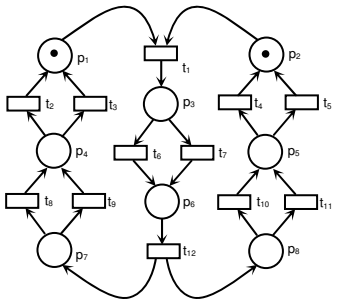

Figure 1. A place-transition net (PT-net).

3. AUGMENTED MARKED GRAPHS

This section introduces augmented marked graphs.

Definition 3.1 [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, such that : (a) Every place in R is marked by M_0. (b) The net (N', M_0') obtained from (N, M_0; R) by removing the places in R and their associated arcs is a marked graph. (A marked graph is a PT-net where for every place p, | •p | = | p• | = 1.) (c) For each r ∈ R, there exist k_r ≥ 1 pairs of transitions D_r = { ⟨t_{s1}, t_{h1}⟩, ⟨t_{s2}, t_{h2}⟩, ..., ⟨t_{skr}, t_{hkr}⟩ } such that $r^\bullet$ = { t_{s1}, t_{s2}, ..., t_{skr} } ⊆ T and $^\bullet r$ = { t_{h1}, t_{h2}, ..., t_{hkr} } ⊆ T and that, for each ⟨t_{si}, t_{hi}⟩ ∈ 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.

Figure 2 shows an augmented marked graph (N, M_0; R), where R = { r_1, r_2 }.

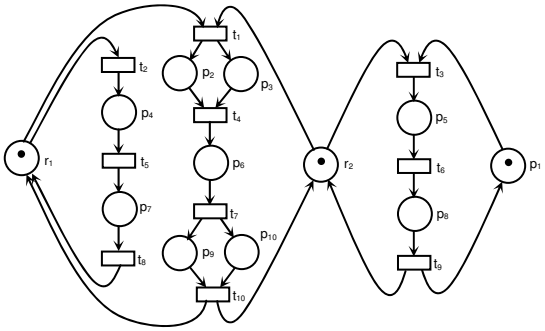

Figure 2. An augmented marked graph.

Definition 3.2. For a PT-net (N, M_0), a set of places S is called a siphon if and only if $^\bullet S \subseteq S^\bullet$. S is said to be minimal if and only if there does not exist a siphon S' in N such that S' ⊂ S. S is said to be empty at a marking M ∈ [M_0⟩ if and only if S contains no places marked by M.

Definition 3.3. For a PT-net (N, M_0), a set of places Q is called a trap if and only if $Q^\bullet \subseteq {}^\bullet Q$. Q is said to be maximal if and only if there does not exist a trap Q' in N such that Q ⊂ Q'. Q is said to be marked at a marking M ∈ [M_0⟩ if and only if Q contains a place marked by M.

Property 3.1 [1]**.** An augmented marked graph is live and reversible if and only if it does not contain any potential deadlock. (A potential deadlock is a siphon which would eventually become empty.)

Definition 3.4. For an augmented marked graph (N, M_0; R), a minimal siphon is called a R-siphon if and only if it contains at least one place in R.

Property 3.2 [1, 2, 3]**.** An augmented marked graph (N, M_0; R) is live and reversible if every R-siphon contains a marked trap.

Property 3.3 [2, 3]**.** An augmented marked graph (N, M_0; R) is live and reversible if and only if no R-siphons eventually become empty.

Definition 3.5 [4]**.** Suppose an augmented marked graph (N, M_0; R) is transformed into a PT-net (N', M_0') as follows. For each r ∈ R, where D_r = { ⟨t_{s1}, t_{h1}⟩, ⟨t_{s2}, t_{h2}⟩, ..., ⟨t_{skr}, t_{hkr}⟩ }, replace r with a set of places { q_1, q_2, ..., q_{kr} } such that $M_0'[q_i] = M_0[r]$ and $q_i^\bullet$ = { t_{si} } and $^\bullet q_i$ = { t_{hi} } for i = 1, 2, ..., k_r. (N', M_0') is called the R-transform of (N, M_0; R).

Figure 3 shows the R-transform (N', M_0') of the augmented marked graph (N, M_0; R) in Figure 2, where r_1 is replaced by { q_{11}, q_{12} } and r_2 by { q_{21}, q_{22} }.

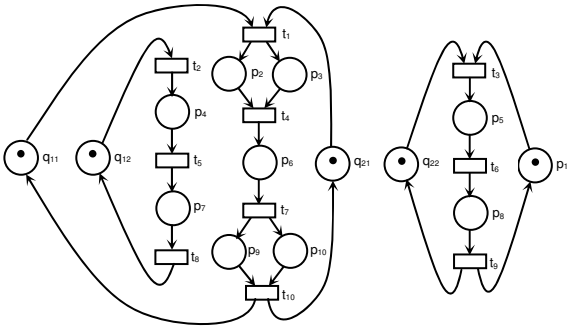

Figure 3. The R-transform of the augmented marked graph in Figure 2.

Property 3.4 [4]. Augmented marked graph (N, M_0; R) is bounded and conservative if and only if every place in its R-transform (N', M_0') belongs to a cycle.

For the augmented marked graph (N, M_0; R) in Figure 2, each R-siphon contains a marked trap and therefore would never become empty. According to Properties 3.2 and 3.3, (N, M_0; R) is live and reversible. Besides, as shown in Figure 3, every place in the R-transform of (N, M_0; R) belongs to a cycle. According to Properties 3.4, (N, M_0; R) is bounded and conservative.

4. COMPOSITE AUGMENTED MARKED GRAPH

This section focus on composition of augmented marked graphs, where preservation of properties is studied.

Property 4.1. Let (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2) be two augmented marked graphs, where $R_1' = \{ r_{11}, r_{12}, ..., r_{1k} \} \in R_1$ and $R_2' = \{ r_{21}, r_{22}, ..., r_{2k} \} \in R_2$ are the common places that r_{11} and r_{21} are to be fused as one single place r_1, r_{12} and r_{22} into r_2, ..., r_{1k} and r_{2k} into r_k. Then, the resulting net is also an augmented marked graph (N, M_0; R), where $R = (R_1 \setminus R_1') \cup (R_2 \setminus R_2') \cup \{ r_1, r_2, ..., r_k \}$. (obvious)

Definition 4.1. With reference to Property 4.1, (N, M_0; R) is called the composite augmented marked graph of (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2) via a set of common resource places $\{ (r_{11}, r_{21}), (r_{12}, r_{22}), ..., (r_{1k}, r_{2k}) \}$, where $r_{11}, r_{12}, ..., r_{1k} \in R_1$ and $r_{21}, r_{22}, ..., r_{2k} \in R_2$. $R_F = \{ r_1, r_2, ..., r_k \}$ is called the set of fused resource places that are obtained after fusing $(r_{11}, r_{21}), (r_{12}, r_{22}), ..., (r_{1k}, r_{2k})$.

Figure 4 shows two augmented marked graphs (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2). Figure 5 shows the composite augmented marked graph (N, M_0; R) of (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2) via $\{ (r_{11}, r_{21}) \}$, where $R_F = \{ r_1, r_2 \}$.

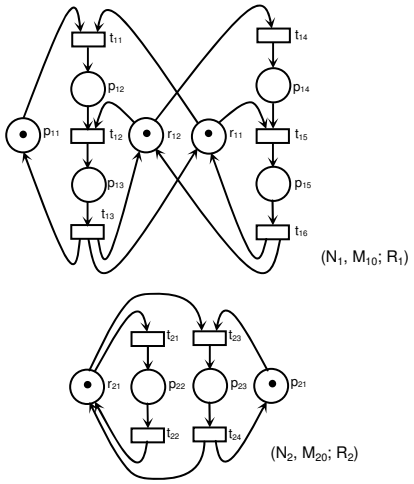

Figure 4. Two augmented marked graphs.

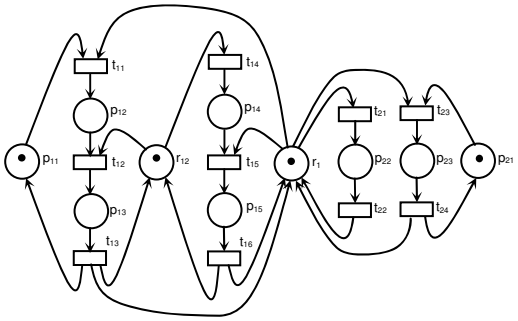

Figure 5. Composing the two augmented marked graphs in Figure 4 via $\{ (r_{11}, r_{21}) \}$.

Property 4.2 [5, 6, 7]. Let (N, M_0; R) be the composite augmented marked graph of two augmented marked graphs (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2) via a set of common resource places. (N, M_0; R) is bounded if and only if (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2) are bounded.

Property 4.3 [5, 6]. Let (N, M_0; R) be the composite augmented marked graph of two augmented marked graphs (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2) via a set of common resource places. (N, M_0; R) is conservative if and only if (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2) are conservative.

Definition 4.2. Let (N, M_0; R) be the composite augmented marked graph of two augmented marked graphs via a set of common resource places, and $R_F \subseteq R$ be the set of fused resource places. For (N, M_0; R), a minimal siphon is called a R_F-siphon if and only if it contains at least one place in R_F.

Property 4.5 [5, 6]. Let (N, M_0; R) be the composite marked graph of two augmented marked graphs (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2) via a set of common resource places. (N, M_0; R) is live and reversible if and only if (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2) are live and no R_F-siphons eventually become empty.

Consider the augmented marked graphs (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2) shown in Figure 4. (N_1, M_{10}; R_1) is neither live nor reversible but is bounded and conservative. (N_2, M_{20}; R_2) is live, bounded, reversible and conservative. According to Properties 4.2 and 4.3, the composite augmented marked graph (N, M_0; R) as shown in Figure 5 is bounded and conservative. Besides, according to Property 4.5, (N, M_0; R) is neither live nor reversible.

Figure 6 shows two augmented marked graphs (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2). Figure 7 shows the composite augmented marked graph (N, M_0; R) of (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2) via $\{ (r_{11}, r_{21}), (r_{12}, r_{22}) \}$, where $R_F = \{ r_1, r_2 \}$. (N_1, M_{10}; R_1) and (N_2, M_{20}; R_2) are live, bounded, reversible and conservative. According to Properties 4.2 and 4.3, (N, M_0; R) is bounded and conservative. No R_F-siphons would eventually become empty. According to Property 4.5, (N, M_0; R) is also live and reversible.

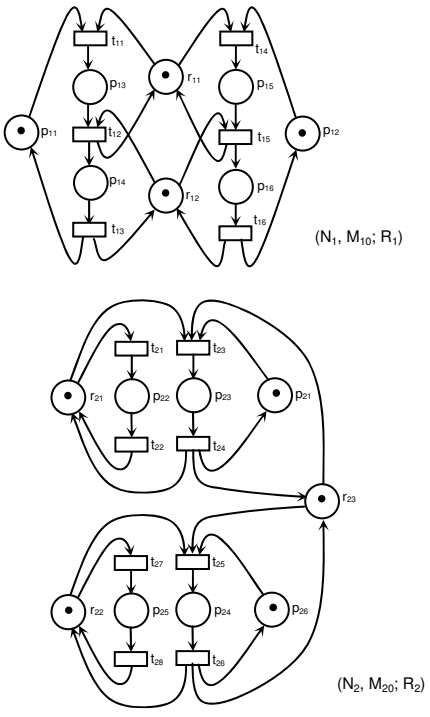

Figure 6. Two augmented marked graphs.

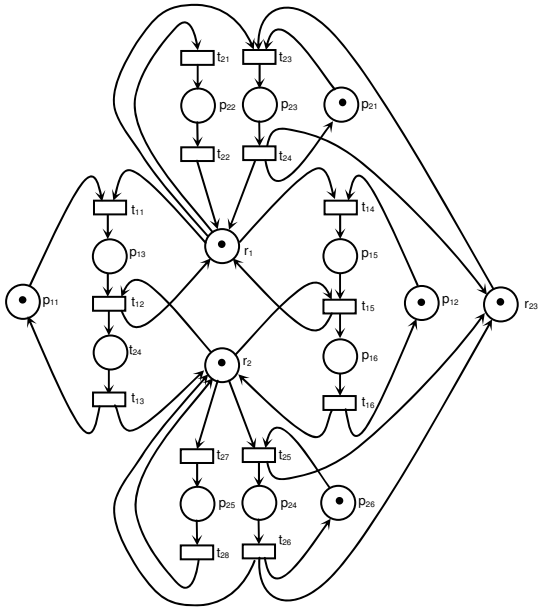

Figure 7. Composing the two augmented marked graphs in Figure 6 via { (r_{11}, r_{21}), (r_{21}, r_{22}) }.

5. APPLICATION TO DISTRIBUTED SYSTEMS

In component-based system design, a system is synthesised from a set of components [11, 12]. The integrated system may not be live (implying freeness of deadlock), bounded (implying absence of capacity overflow) and reversible (implying the capability of being reinitialised from any reachable states) even though all the components are live, bounded and reversible, especially as competition of common resources exists. For distributed systems which usually involve some concurrent (parallel) and asynchronous processes, because of competition of common resources, errors such as deadlock and capacity overflow are easily induced. Hence, it is important for one to ensure design correctness in the sense that the integrated system is live, bounded, reversible and conservative.

In the following, we describe the application of the composition of augmented marked graphs to the synthesis of distributed systems. By modelling the distributed system components as augmented marked graphs and composing them via their common resource places which represent the common resources, based on the theoretical results obtained in the previous section, the properties of the integrated system can be readily derived. In brief, if all the components are bounded and conservative, the integrated system will be bounded and conservative. If all the components are live and reversible, the integrated system will be live and reversible under a pretty simple condition - No R_F-siphons in the composite augmented marked graph would eventually become empty.

Let us consider a distributed system consisting of four distributed system components, namely, C_1, C_2, C_3 and C_4. Owing to the "distributed processing" nature, the components exhibit some concurrent (parallel) and asynchronous processes. There are six pieces of common resources, namely, S_1, S_2, S_3, S_4, S_5 and S_6, which are used to be shared among the components, as outlined in Figure 8.

The typical processes exhibited by the distributed system components are briefly outlined as follows.

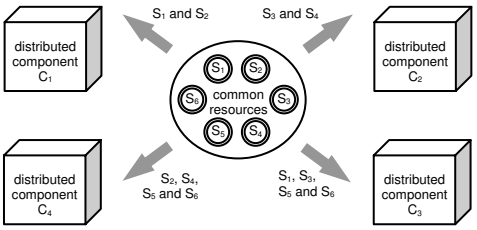

Figure 8. Example of a distributed system with shared resources.

C_1 : At its initial idle state, C_1 may invoke operation o_{11} only if S_1 is available. While o_{11} is being processed, S_1 is occupied. Once o_{11} finishes processing, operation o_{12} is invoked only if S_2 is available. S_1 is then released. While o_{12} is being processed, S_2 is occupied. Once o_{12} finishes processing, S_2 is released and C_1 returns to its idle state. From time to time, S_1 is withheld upon receipt of signal m_{11} and released upon receipt of signal m_{12}. S_2 is withheld upon receipt of signal m_{13} and released upon receipt of signal m_{14}.

C_2 : At its initial idle state, C_2 may invoke operation o_{21} only if S_3 is available. While o_{21} is being processed, S_3 is occupied. Once o_{21} finishes processing, operation o_{22} is invoked only if S_4 is available. S_3 is then released. While o_{22} is being processed, S_4 is occupied. Once o_{22} finishes processing, S_4 is released and C_2 returns to its idle state. From time to time, S_3 is withheld upon receipt of signal m_{21} and released upon receipt of signal m_{22}. S_4 is withheld upon receipt of signal m_{23} and released upon receipt of signal m_{24}.

C_3 : At its initial idle state, C_3 may invoke operation o_{31} only if S_1, S_3, S_5 and S_6 are available. While o_{31} is being processed, S_1, S_3, S_5 and S_6 are occupied. Once o_{31} finishes processing, S_1, S_3, S_5 and S_6 are released and C_3 returns to its idle state.

C_4 : At its initial idle state, C_4 may invoke operation o_{41} only if S_2, S_4, S_5 and S_6 are available. While o_{41} is being processed, S_2, S_4, S_5 and S_6 are occupied. Once o_{41} finishes processing, S_2, S_4, S_5 and S_6 are released and C_4 returns to its idle state.

Our method begins with representing each component as an augmented marked graph. We identify the event occurrences and their pre-conditions and post-conditions in the component. For each event occurrence, a transition is created for denoting the location of occurrence. Input and output places are created to denote the locations of its pre-conditions and post-conditions. An initial marking is created to denote the system initial state. Execution for the component begins at this initial marking which semantically means its initial idle state, and ends at the same marking.

Component C_1 is specified as augmented marked graph (N_1, M_{10}; R_1), where R_1 = { r_{11}, r_{12} }. C_2 is specified as (N_2, M_{20}; R_2), where R_2 = { r_{21}, r_{22} }. C_3 is specified as (N_3, M_{30}; R_3), where R_3 = { r_{31}, r_{32}, r_{33}, r_{34} }. C_4 is specified as (N_4, M_{40}; R_4), where R_4 = { r_{41}, r_{42}, r_{43}, r_{44} }. They are shown in Figure 9, while Table 1 lists the semantic meanings of the places and transitions.

According to Properties 3.1, 3,2, 3.3 and 3.4, (N_1, M_{10}; R_1), (N_2, M_{20}; R_2), (N_3, M_{30}; R_3) and (N_4, M_{40}; R_4) are live, bounded, reversible and conservative.

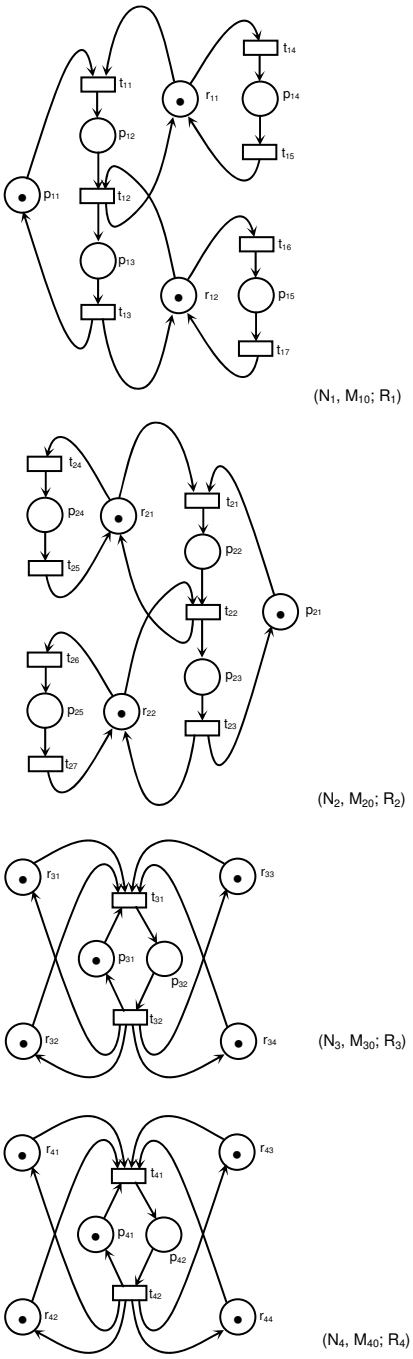

Figure 9. Specification of distributed system components as augmented marked graphs.

Resource places r_{11} in (N_1, M_{10}; R_1) and r_{31} in (N_3, M_{30}; R_3) refer to the same resource S_1. r_{12} in (N_1, M_{10}; R_1) and r_{42} in (N_4, M_{40}; R_4) refer to the same resource S_2. r_{21} in (N_2, M_{20}; R_2) and r_{33} in (N_3, M_{30}; R_3) refer to the same resource S_3. r_{22} in (N_2, M_{20}; R_2) and r_{44} in (N_4, M_{40}; R_4) refer to the same resource S_4. r_{32} in (N_3, M_{30}; R_3) and r_{41} in (N_4, M_{40}; R_4) refer to the same resource S_5. r_{34} in (N_3, M_{30}; R_3) and r_{43} in (N_4, M_{40}; R_4) refer to the same resource S_6. (N_1, M_{10}; R_1), (N_2, M_{20}; R_2), (N_3, M_{30}; R_3) and (N_4, M_{40}; R_4) are to be composed via these common resource places. We first obtain the composite augmented marked graphs (N', M_0'; R') of (N_1, M_{10}; R_1) and (N_3, M_{30}; R_3) via { (r_{11}, r_{31}) }, and (N", M_0"; R") of (N_2, M_{20}; R_2) and (N_4, M_{40}; R_4) via { (r_{22}, r_{44}) }.

Table 1. Semantic meaning of places and transitions of the augmented marked graphs in Figure 9.

Place/Tran	Semantic Meaning
p_{11}	Component C_1 is at the idle state
p_{12}	Component C_1 is performing operation o_{11}
p_{13}	Component C_1 is performing operation o_{12}
p_{14}	Resource S_1 is being withheld
p_{15}	Resource S_2 is being withheld
p_{21}	Component C_2 is at the idle state
p_{22}	Component C_2 is performing operation o_{21}
p_{23}	Component C_2 is performing operation o_{22}
p_{24}	Resource S_3 is being withheld
p_{25}	Resource S_4 is being withheld
p_{31}	Component C_3 is at the idle state
p_{32}	Component C_3 is performing operation o_{31}
p_{41}	Component C_4 is at the idle state
p_{42}	Component C_4 is performing operation o_{41}
r_{11}, r_{31}	Resource S_1 is available
r_{12}, r_{42}	Resource S_2 is available
r_{21}, r_{33}	Resource S_3 is available
r_{22}, r_{44}	Resource S_4 is available
r_{32}, r_{41}	Resource S_5 is available
r_{34}, r_{43}	Resource S_6 is available
t_{11}	Component C_1 starts operation o_{11}
t_{12}	Component C_1 finishes operation o_{11} and starts operation o_{12}
t_{13}	Component C_1 finishes operation o_{12}
t_{14}	Component C_1 receives signal m_{11}
t_{15}	Component C_1 receives signal m_{12}
t_{16}	Component C_1 receives signal m_{13}
t_{17}	Component C_1 receives signal m_{14}
t_{21}	Component C_2 starts operation o_{21}
t_{22}	Component C_2 finishes operation o_{21} and starts operation o_{22}
t_{23}	Component C_2 finishes operation o_{22}
t_{24}	Component C_2 receives signal m_{21}
t_{25}	Component C_2 receives signal m_{22}
t_{26}	Component C_2 receives signal m_{23}
t_{27}	Component C_2 receives signal m_{24}
t_{31}	Component C_3 starts operation o_{31}
t_{32}	Component C_3 finishes operation o_{31}
t_{41}	Component C_4 starts operation o_{41}
t_{42}	Component C_4 finishes operation o_{41}

Figure 10 shows (N', M_0'; R'), where r_1 is the place after fusing r_{11} and r_{31}. Figure 11 shows (N", M_0"; R"), where r_4 is the place after fusing r_{22} and r_{44}.

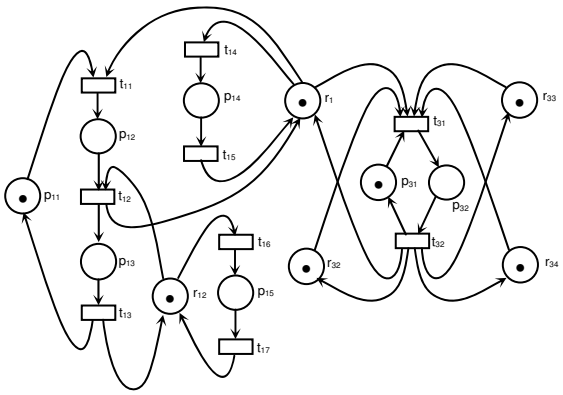

Figure 10. Composite augmented marked graph (N', M_0'; R').

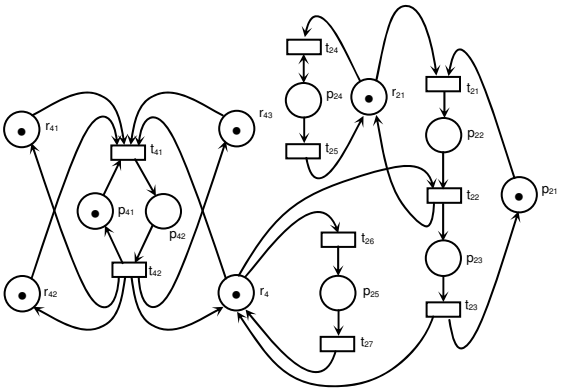

Figure 11. Composite augmented marked graph (N", M_0"; R").

Since (N_1, M_{10}; R_1), (N_2, M_{20}; R_2), (N_3, M_{30}; R_3) and (N_4, M_{40}; R_4) are all bounded and conservative, according to Properties 4.2 and 4.3, the composite augmented marked graphs (N', M_0'; R') and (N", M_0"; R") are also bounded and conservative. (N_1, M_{10}; R_1), (N_2, M_{20}; R_2), (N_3, M_{30}; R_3) and (N_4, M_{40}; R_4) are all live and reversible. For (N', M_0', R'), where R_F' = { r_1 }, no R_F'-siphons would eventually become empty. According to Property 4.5, (N', M_0', R') is also live and reversible. For (N", M_0", R"), where R_F" = { r_4 }, no R_F"-siphons would eventually become empty. According to Property 4.5, (N", M_0", R") is also live and reversible.

We obtain the final composite augmented marked graph (N, M_0; R) of (N', M_0'; R') and (N", M_0"; R") via { (r_{12}, r_{42}), (r_{33}, r_{21}), (r_{32}, r_{41}), (r_{34}, r_{43}) }. Figure 12 shows (N, M_0; R), where r_2 is the place after fusing r_{12} and r_{42}, r_3 is the place after fusing r_{21} and r_{33}, r_5 is the place after fusing r_{32} and r_{41}, and r_6 is the place after fusing r_{34} and r_{43}. Table 2 lists the semantic meanings of its places and transitions.

Since (N', M_0'; R') and (N", M_0"; R") are bounded and conservative, according to Properties 4.2 and 4.3, the composite augmented marked graph (N, M_0; R) is also bounded and conservative. (N', M_0'; R') and (N", M_0"; R") are live and reversible. For (N, M_0; R), where R_F = { r_2, r_3, r_5, r_6 }, no R_F-siphons would eventually become empty. According to Property 4.5, (N, M_0; R) is also live and reversible. Hence, it may be concluded that the integrated system is live, bounded, reversible and conservative.

Table 2. Semantic meaning of places and transitions of the augmented marked graph in Figure 12.

Place/Tran	Semantic Meaning
p_{11}	Component C_1 is at the idle state
p_{12}	Component C_1 is performing operation o_{11}
p_{13}	Component C_1 is performing operation o_{12}
p_{14}	Resource S_1 is being withheld
p_{15}	Resource S_2 is being withheld
p_{21}	Component C_2 is at the idle state
p_{22}	Component C_2 is performing operation o_{21}
p_{23}	Component C_2 is performing operation o_{22}
p_{24}	Resource S_3 is being withheld
p_{25}	Resource S_4 is being withheld
p_{31}	Component C_3 is at the idle state
p_{32}	Component C_3 is performing operation o_{31}
p_{41}	Component C_4 is at the idle state
p_{42}	Component C_4 is performing operation o_{41}
r_1	Resource S_1 is available
r_2	Resource S_2 is available
r_3	Resource S_3 is available
r_4	Resource S_4 is available
r_5	Resource S_5 is available
r_6	Resource S_6 is available
t_{11}	Component C_1 starts operation o_{11}
t_{12}	Component C_1 finishes operation o_{11} and starts operation o_{12}
t_{13}	Component C_1 finishes operation o_{12}
t_{14}	Component C_1 receives signal m_{11}
t_{15}	Component C_1 receives signal m_{12}
t_{16}	Component C_1 receives signal m_{13}
t_{17}	Component C_1 receives signal m_{14}
t_{21}	Component C_2 starts operation o_{21}
t_{22}	Component C_2 finishes operation o_{21} and starts operation o_{22}
t_{23}	Component C_2 finishes operation o_{22}
t_{24}	Component C_2 receives signal m_{21}
t_{25}	Component C_2 receives signal m_{22}
t_{26}	Component C_2 receives signal m_{23}
t_{27}	Component C_2 receives signal m_{24}
t_{31}	Component C_3 starts operation o_{31}
t_{32}	Component C_3 finishes operation o_{31}
t_{41}	Component C_4 starts operation o_{41}
t_{42}	Component C_4 finishes operation o_{41}

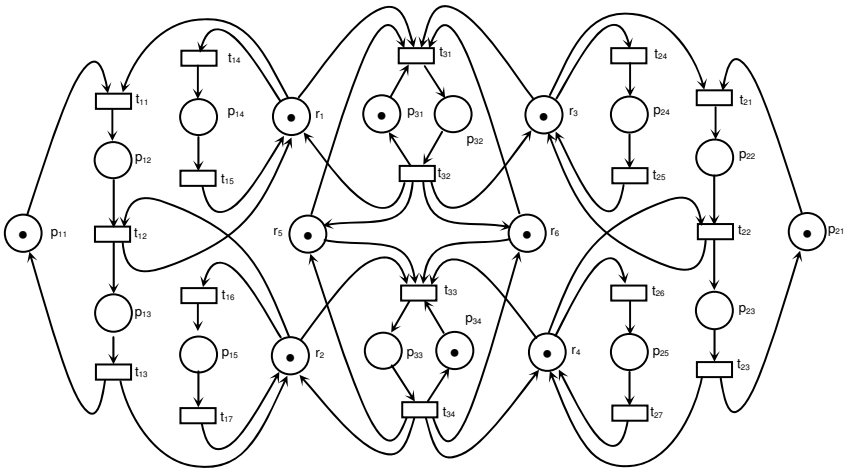

Figure 12. The final composite augmented marked graph (N, M_0; R).

6 CONCLUSION

This paper investigates the property-preserving composition of augmented marked graphs and its application to the synthesis of distributed systems. It is shown that, in composing two augmented marked graphs via their common resource places, boundedness and conservativeness are preserved while liveness and reversibility can be preserved under a pretty simple condition. By modelling the distributed system components as augmented marked graphs with the common resources denoted by the resource places, an integrated system can be obtained by composing these augmented marked graphs via their common resource places. Based on preservation of properties, liveness, boundedness, reversibility and conservativeness of the integrated system can be readily derived.

Liveness, boundedness, reversibility and conservativeness are four essential properties that collectively characterise the robustness and well-behavedness of a system. For distributed systems which usually involve some concurrent (parallel) and asynchronous processes, it is important for one to assure design correctness in the sense that these essential system properties can be preserved, especially as competition of common resources exists. By making good use of the special structure and desirable properties of augmented marked graphs as well as the property-preserving composition of augmented marked graphs, our method contributes to resolve the problem of ensuring design correctness in the composition of distributed system components.

7. REFERENCE

[1] F. Chu and X. Xie, "Deadlock Analysis of Petri Nets Using Siphons and Mathematical Programming", IEEE Transactions on Robotics and Automation, Vol. 13, No. 6, pp. 793-804, 1997.

[2] K.S. Cheung, "New Characterisations for Live and Reversible Augmented Marked Graphs", Information Processing Letters, Vol. 92, No. 5, pp. 239-243, 2004.

[3] K.S. Cheung and K.O. Chow, "Cycle Inclusion Property of Augmented Marked Graphs", Information Processing Letters, Vol. 94, No. 6, pp. 271-276, 2005.

[4] K.S. Cheung, "Boundedness and Conservativeness of Augmented Marked Graphs", IMA Journal of Mathematical Control and Information, Vol. 24, No. 2, pp. 235-244, 2007.

[5] K.S. Cheung and K.O. Chow, "Compositional Synthesis of Augmented Marked Graphs", Proceedings of the IEEE International Conference on Control and Automation, pp. 2810-2814, IEEE Press, 2007.

[6] K.S. Cheung and K.O. Chow, "Compositional Synthesis of Augmented Marked Graphs for Manufacturing System Integration", Proceedings of the IEEE International Conference on Integration Technology, pp. 331-335, IEEE Press, 2007.

[7] H.J. Huang, L. Jiao and T.Y. Cheung, "Property-Preserving Composition of Augmented Marked Graphs that Share Common Resources", Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 1, pp. 1446-1451, IEEE Press, 2003.

[8] W. Reisig, Petri Nets : An Introduction, Springer, 1985.

[9] T. Murata, "Petri Nets : Properties, Analysis and Applications", Proceedings of the IEEE, Vol. 77, No. 4, pp. 541-580, 1989.

[10] J. Desel and W. Reisig, Place Transition Petri Nets, Lectures on Petri Nets, Volume 1 : Basic Models, Lecture Notes in Computer Science, Vol. 1491, pp. 122-173, Springer-Verlag, 1998.

[11] G.T. Heineman and W.T. Councill, Component-Based Software Engineering : Putting the Pieces Together, Addison-Wesley, 2002.

[12] I. Crnkovic and M. Larsson, Building Reliable Component-Based Software Systems, Artech House, 2002.

Received: Oct. 2007. Accepted: Mar. 2008.