Modeling and Analysis for Grid Service Cooperative Scheduling Based on Petri Nets*

Yaojun Han[1,2], Changjun Jiang[1], and Xuemei Luo[2]

[1] Department of Computer Science & Engineering, Tongji University, Shanghai, 200092, China
[2] College of International Business Administration, Shanghai International Studies University, Shanghai 200083, China
yjhan@shisu.edu.cn

Abstract. As the complexity of application system for enterprises, an important challenge is to dynamically schedule and integrate the heterogeneous and distributed services or activities to work cooperatively and efficiently. An effective technology to resolve the problem is grid service. A grid service built on both grid computing and web services technologies is an extended Web service. An application system for enterprises is a grid service composition that consists of a collection of grid services related by data and control flow. Therefore, there is a need for modeling and analyzing techniques and tools for reliable and effective grid service composition. The Petri net based method is an idea approach. In this paper, we use a colored dynamic timed Petri net (CDTPN) to model the grid service composition. The definition of CDTPN for grid service and an algorithm to construct a composite service are proposed. We give a definition of reachable service graph and an algorithm for constructing the reachable service graph of CDTPN. Finally, we discuss the correctness and effectiveness of the grid service composition by analyzing the reachable service graph.

Keywords: grid service, composition, dynamic timed Petri net, performance analysis

1 Introduction

As the development of Internet and World Wide Web, many organizations are rushing to put their core business competencies on the Internet to survive the massive competition created by new online economy [1]. An important challenge is to dynamically schedule and integrate the heterogeneous and distributed services or activities to work cooperatively and efficiently. An effective technology to resolve the problem is grid service. A grid service built on both grid computing and web services technologies is an extended Web service [2]. Grid computing is becoming a mainstream technology

* This work is support partially by projects of National Basic Research Program of China(973 Program)(2003CB316902, 2004CB318001-03), National Natural Science Fund (90612006, 90412013, 60473094), Humanities and Social Sciences Foundation of Ministry of Education (06JA870006), and Science Research Funds of Shanghai International Study University.

Y. Luo (Ed.): CDVE 2007, LNCS 4674, pp. 104–112, 2007.

for large-scale distributed resource sharing and system integration [3]. Grid applications for service-based systems are usually not based on a single service, but are rather composed of several services working together in an application specific manner. As the complexity of application system for enterprises, an important challenge is to dynamically schedule and integrate the heterogeneous and distributed services to work cooperatively and efficiently. Therefore, there is a need for modeling and analyzing techniques and tools for reliable grid service composition because of dynamic and complex service composition process.

Petri nets are promising tools for modeling and analysis information processing systems that are characterized as being concurrent, parallel and distributed [9,10,11]. Many researchers model and analyze Web service using Petri Nets, since they are well suited for capturing flows in web services, for modeling the distributed nature of web services, for representing methods in a web service and for reasoning about the correctness of the flows [1,4-7]. The existing approaches have difficulties in modeling and analyzing dynamic and complex grid service composition process, so we propose a colored dynamic timed Petri Net (CDTPN) model for grid service composition, which is an extended timed Petri-net model. In CDTPN, the time delay of transition is a function of execution time of a service instead of time constant, which is convenient for modeling and analyzing dynamic performance of grid service composition.

The rest of this paper is organized as follows. The concepts of Petri nets related to this paper are reviewed in section 2. In section 3, we present CDTPN model of grid service and give the algorithm for four basic structures of grid service composition workflow. In section 4, we propose a definition of reachable service graph, give algorithms for constructing the reachable service graph of CDTPN and discuss the correctness and performance of the grid service composition by analyzing the reachable service graph. Section 5 presents a case study of public services in a city. We conclude the paper in section 6.

2 Concepts of Petri Nets Related to the Paper

In this section, we simply review some concepts of Petri nets related to this paper. For the details of the definitions, the reader can see references [8,9,10].

Definition 1[8]**.** A Petri Net is a bipartite directed graph represented by a three-tuple $PN = (P,T;F,M_0)$, where,

$P=\{p_1,p_2...p_n,\}$ is a finite set of place nodes;

$T=\{t_1,t_2,...t_m\}$ is a finite set of transition nodes;

$P\cap T =\varphi, P\cup T \neq\varphi$;

$F= P\times T\cup T\times P$ is a finite set of directed arcs from P to T and T to P, where directed arcs from P to T are called input arcs, directed arcs from T to P are called output arcs;

$M_0 : P\rightarrow N$ is called an initial marking.

Let $PN = ((P,T;F,M_0)$ be a Petri net. For $x\in P\cup T$,

$x\bullet=\{y\in P\cup T|(y,x) \in F \}$ and

$\bullet x=\{y\in P\cup T|(x,y) \in F\}$

are called the pre-set and post-set of x respectively.

A transition $t \in T$ is enabled in M iff $M(p) \geq 1$ for any $p \in$ pre-set of t. A transition t enabled in M can fire and yield a new marking $M'(p)=M(p)-1$ for any $p \in$ pre-set of t and $M'(p)=M(p)+1$ for any $p \in$ post-set of t.

Definition 2[9]**.** A colored Petri net (CPN) is a eight-tuple $CPN=(\Omega,P,T,F,C,G,E, M_0)$, where,

$(P,T; F,)$ is a Petri net;
Ω is a set of colors;
C is a color function, C: $P \rightarrow \Omega$;
G is a guard function, G: $T \rightarrow$Boolexpression, $\forall t \in T$: Type(G(t))=Boolean$\wedge$ Type(var(G(t)))$\subseteq \Omega$;
E is an arc expression function, E: $F \rightarrow$expression, $\forall f \in F$: Type(E(f))=$C(p)_{MS}$ $\wedge$ Type(var(E(f)))$\subseteq \Omega$, where $C(p)_{MS}$ is the set of all multi-sets over $C(p)$;
M_0 is an initial marking function, M_0:$P \rightarrow$expression, $\forall p \in P$: Type(M_0 (p)) $=C(p)_{MS}$.

Definition 3[10]**.** A timed Petri net (TPN) is a five-tuple $TPN=(P,T;F,M_0,D)$, where,

$(P,T;F,M_0)$ is a Petri net;
D: $T \rightarrow R$ is a firing time delay, where R is the set of nonnegative rational numbers. For $t \in T$, $D(t)=a$ represents the firing time delay of t is a.

3 Petri Net Model for Grid Service

Definition 4. The Petri net model for a grid service is a colored dynamic timed Petri net (CDTPN, shown in Fig. 1). CDTPN = $(\Omega,P,T,F,C,E, G, M_0, D)$, where,

Ω= Ip$\cup$Op$\cup$QoS, where Ip and Op represent the input and output parameters of service respectively, QoS is the user's QoS requirements;
Variable: x: Ip$\cup$QoS; y: Op$\cup$QoS;
$P=\{si,so\}$, where, si and so represent input and output of the service S;
$T=\{s\}$, where s represents the service S;
$F=\{(si, s), (s,so)\}$;
$C=\{C(si)=$ Ip$\cup$QoS, $C(so)=$Op$\cup$QoS$\}$;
$E=\{E(si, s)=x, E(s, so)=y\}$;
$G=G(s)$. It is a function of x;
$M_0(si)=1'x$, $M_0(so)=0$;
$D=D(s)$. It is a function of x.

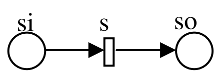

Fig. 1. CDTPN model for a grid service

A grid service has a specific task to perform and may depend on other grid services. The grid services cooperative scheduling is grid services composition. The composition of two or more services generates a new service providing both the original individual behavioral logic and a new collaborative behavior for carrying out a new composite task. A composite service consists of a collection of grid services related by data and control flow. Sequence, alternative, parallel and iteration are typical structures specified in the workflow.

Algorithm 1. Construction of CDTPN model for a grid application composed of some grid services:

(1) Input all grid services with input parameters as well as user's QoS requirements.
(2) Input the dependent relation (Sequence, alternative, parallel and iteration) of all grid services,
(3) Construct the CDTPN model for every grid service using definition 4.
(4) For all services, according to dependent relation of all grid services, construct the CDTPN model for composition of all grid services as following.
 (4.1) If grid services S1 and S2 are sequential, i.e., a composite service (denoted by S1→S2) that performs the service S1 followed by the service S2, the S1→S2 is a CDTPN(Fig. 2), where $C(in)$=QoS,$C(out)$=Op∪QoS, $E(in,t_1)=E(t_1,si_1)$=QoS, $E(so_1,t_{12})=E(t_{12},si_2)=x$, $E(so_2,t_2)=E(t_2,out)=y$, $G(t_1)=G(t_2)=G(t_{12})$=Null, $M_0(in)$=QoS, $M_0(out)$=0, $D(t_1)=D(t_2)= D(t_{12})$=0;
 (4.2) If grid services S1 and S2 are alternative, i.e., a composite service (denoted by S1⊕S2) that behaves as either service S1 or service S2 (Once one of them executes its first operation the second service is discarded), the S1⊕S2 is a CDTPN(Fig.3), where $C(in)=C(p_1)$=QoS, $C(out)=C(p_2)$=Op∪QoS, $E(in,t_1)=E(t_1,p_1)=E(p_1,t_{11})=E(p_1,t_{21})=E(t_{11},si_1)= E(t_{21},si_2)$=QoS, $E(so_1,t_{12})= E(so_2,t_{22})=E(t_{12},p_2)=E(t_{22},p_2)=E(p_2,t_2)=E(t_2,out)=y$, $G(t_{11})$=(S1∈QoS), $G(t_{12})$=(S2∈QoS), $G(t_1)=G(t_2)=G(t_{12})=G(t_{22})$=Null, $M_0(in)$=QoS, $M_0(out)=M_0(p_1)=M_0(p_2)$=0, $D(t_1)=D(t_2)=D(t_{11})=D(t_{12})=D(t_{21})= D(t_{22})$=0,
 (4.3) If grid services S1 and S2 are parallel, i.e., a composite service (denoted by S1‖S2) that performs service S1 and S2 independently from each other, the S1‖S2 is a CDTPN(Fig. 4), where $C(in)$=QoS,$C(out)$=Op∪QoS, $E(in,t_1)=E(t_1,si_1)=E(t_1,si_2)$=QoS, $E(so_1,t_2)=E(so_2,t_2)=E(t_2,out)=y$, $M_0(in)$=QoS, $M_0(out)$=0, $D(t1)=D(t2)$=0, $G(t_1)=G(t_2)$=Null.
 (4.4) If grid services S1 is iterative, i.e., a composite service (denoted by nS1) that performs service S1 for n times, the nS1 is a CDTPN(Fig. 5), where $C(in)$=QoS, $C(out)$=Op∪QoS, $E(in,t_1)=E(t_1,si_1)$=QoS, $E(so_1,t_2)=E(so_1,t_{11})= E(t_2,out)=y$, $E(s_1,so_1)$ is a function including calculation of n-1, $G(t_2)$=(n==0), $G(t_{11})$=(n>0), $G(t_1)$=Null, $M_0(in)$=QoS, $M_0(out)$=0, $D(t1)=D(t2)$=0.

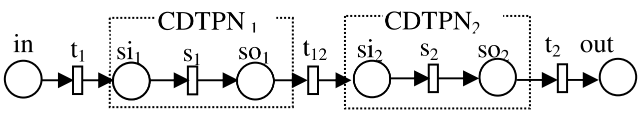

Fig. 2. CDTPN for the sequential composition of grid services S1 and S2

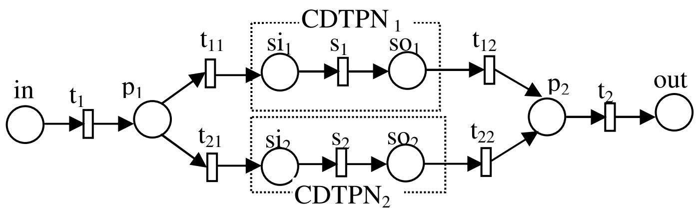

Fig. 3. CDTPN for the alternative composition of grid services S1 and S2

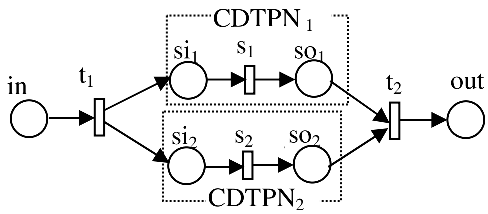

Fig. 4. CDTPN for the parallel composition of grid services S1 and S2

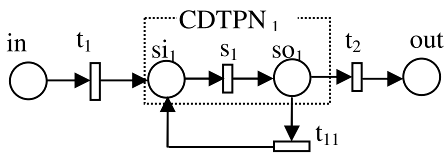

Fig. 5. CDTPN model for *n* times iteration of grid service S1

4 Analysis of a Grid Application Composed of Some Grid Services

A major strength of Petri nets is their support for analysis of many properties and problems associated with concurrent systems. After modeling the grid service with Petri nets, we will analyze the correctness and performance of the grid application in order to ensure the correctness and effectiveness of the grid service composition. A fundamental and most widely applied method for analyzing behavioral properties of Petri net models is coverability tree[10]. In this paper, we use the reachable service graph of the model instead of coverability tree to analyze the correctness and effectiveness of the grid service composition.

Definition 5. Let CDTPN be a colored dynamic timed Petri net model for a grid application composed of some grid services. The reachable service graph (RSG) of CDTPN is defined as a directed graph with labeled directed edges and labeled nodes.

RSG(CDTPN)=(V,E,FT, FM). Where,
$V=\{R(M_0)\}$,
$E=\{(M_i, M_j)|\ M_i, M_j\in R(M_0), \exists t_k\in \mathrm{T}: M_i [\ t_k >M_j$,

$FT(M_i, M_j)= t_k/ct_k$, where ct_k is the current time of transition t_k firing.
$FM(M_j)$=OP, where OP is a set of output parameters of a service.

The reachable service graph of CDTPN is constructed by the following algorithm.

Algorithm 2. Construction of reachable service graph of CDTPN

(1) Let V={M_0}, E={φ}, tag M_0 "new".
(2) If there exists no "new" node in V, then the algorithm ends, otherwise go to (3).
(3) Select a "new" marking M and do the following:
 (3.1) If there no exist enabled transition t at M, then tag M with "end node".
 (3.2) While there exist t at M, do the following for each enabled transitions t at M:
  (3.21) Obtain M' that results from firing t at M.
  (3.22) If $M' \notin$ V, then V=V+{M'} and tag M' with "new".
  (3.23) E=E+{M,M'}, tag {M,M'} with t_i/ct_i, where ct_i is the current time of transition t_i firing.
  (3.24) If $t=s_i$, then tag M' with a set of execution time and output parameters of service s_i.
  (3.25) Otherwise, tag M' with "[0,0]"(generally, [0,0] is omitted).
 (3.3) Remove "new" from M and go to (1).

The correctness of the algorithm 2 can be easily proven according to the definitions of CDTPN and RSG.

Proposition 1. Let CDTPN be a colored dynamic timed Petri net model for a grid application composed of some grid services.The CDTPN is deadlock-free iff for any end node $M\in$ RSG(M_0), $M(out)\neq 0$ and any $p\neq out\in P$, $M(p)=0$.

Definition 6. The grid service composition is correct iff CDTPN is deadlock-free.

Proposition 2. Let CDTPN be a colored dynamic timed Petri net model for a grid application composed of some grid services and deadlock-free. RSG(M_0) is a reachable service graph of CDTPN.

(1) The total execution cost (EC) of a grid application is equal to $\sum c_i$, where $c_i\in M_i$ (s_i•) is the execution cost of service s_i, $M_i\in$ RSG(M_0).
(2) The total execution time (ET) of a grid application is equal to max{$\sum e_i$}, where $\sum e_i$ is the sum of tag with M_i. M_i is the node of path from M_0 to M. M_0, M_i, $M\in$ RSG(M_0) and M is the end-node. e_i is the execution time of service s_i.

Definition 7. A grid application composed of some grid services is feasible iff the turnover time is less than or equal to the deadline given by user, where the turnover time is the time interval between when a grid application is submitted and when it is completed. A grid application composed of grid services is economical iff the total transmission and execution cost is less than or equal to the budget given by user.

Proposition 3. Let CDTPN be deadlock-free. B and D are the budget and deadline given by user. The grid application is economical iff $EC\leq B$, where EC is the execution cost of the grid application. The grid application is feasible iff ct_e-ct_1≤D, where ct_1 is the tag t_1/ ct_1 of edge (M_0, M_1). ct_e is the tag t_e/ ct_e of edge (M_i, M), where t_e is pre-set of place *out*, *M* is the end-node.

5 A Case Study

We discuss public service system of a city as a case study. Suppose that a citizen want to see a doctor. , he will take taxi. Otherwise he will take bus. The public service system does the following. First, it looks up the weather forecast (service S1), meantime transfer money to hospital account from the user bank account (service S2). Second, it books taxi (service S3) for the user if it rains, otherwise tells the bus information (service S4) to the user according to the weather forecast. Last, it makes an appointment with doctor for the user (service S5). Obviously, S1 and S2 are parallel; S3 and S4 are alternative. We select the execution times and costs of services as the input and output parameters of services. Suppose that the execution times and costs of services S1, S2, S3, S4 and S5 are (10,30), (20,50), (40,70), (15,30) and (48,90) respectively. The user's deadline and budget are 100 and 220 units respectively.

(1) Construct CDTPN according to the definitions 4 and algorithm 1. The graphic representation of CDTPN is shown in figure 6, where $G(t_{31})$=(rain==True), $G(t_{41})$=(rain==False).

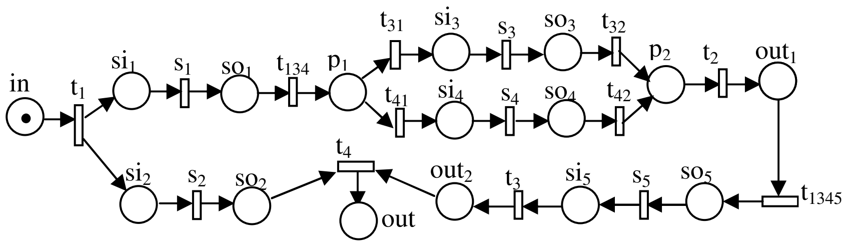

Fig. 6. Public service system S=((S1→ (S3⊕ S4) →S5)‖S2)

(2) According to algorithm 2, construct RSG of CDTPN and show in figure7 (Suppose that there is no delay for executing every grid service after the condition is content).

(3) Analyze the correctness of the grid application. From figure 7, we know CDTPN is deadlock-free according to proposition 1. So, the grid application is correct.

(4) Analyze the performance of the grid application. From figure 7, we know if it rains, the total execution time and cost of the grid application are 98 and 240, and the grid application is feasible but not economical because ct_e-ct_1=98<user's deadline140 and cost 240>user's budget 220. If it doesn't rain, the total execution time and cost of the grid application are 73 and 200 , and the grid application is feasible and economical because 73<140 and 200<220.

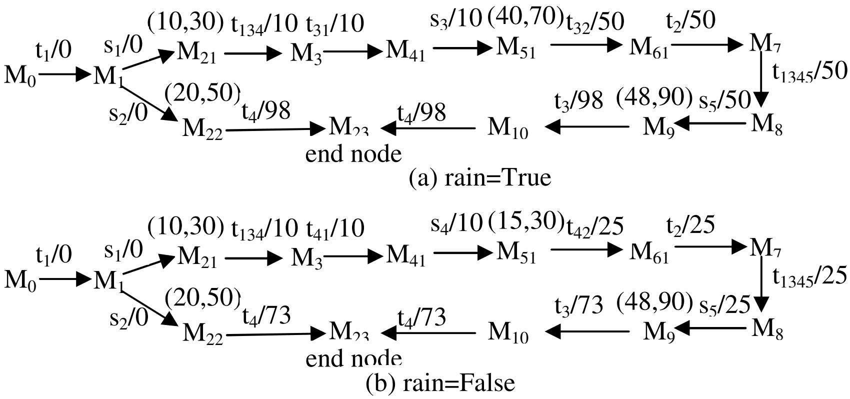

Fig. 7. RSG of Fig. 12

6 Conclusions

In this paper, we use a colored dynamic timed Petri net (CDTPN) to model the grid service composition (i.e. the grid services cooperative scheduling). The definition of CDTPN for grid service and an algorithm to construct a composite service are proposed. We give a definition of reachable service graph and an algorithm for constructing the reachable service graph of CDTPN. We get some conclusions about the correctness and effectiveness of the grid service composition by analyzing the reachable service graph. Finally, we discuss public service system of a city as a case study. From the example, we know that the CDTPN model given by us can represent the logical flow of grid service composition clearly and is very effective for analyzing the correctness and effectiveness of the grid service composition. In our future work, we will develop the software tools for composing the services automatically.

References

1. Hamadi, R., Benatallah, B.: A Petri Net-based Model for Web Service Composition. In: Proceedings of the Fourteenth Australasian database conference on Database technologies, Adelaide, Australia, pp. 191–200 (2003)
2. Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C.: Grid service specification (2002)
3. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid: Enabling scalable virtual organizations. Int'l Journal of Supercomputing Applications 15(3), 200–222 (2002)
4. Tan, Z., Li, C., et al.: Approximate Performance Analysis of Web Services Flow Using Stochastic Petri Net. In: Jin, H., Pan, Y., Xiao, N., Sun, J. (eds.) GCC 2004. LNCS, vol. 3251, pp. 193–200. Springer, Heidelberg (2004)
5. Tang, Y., Chen, L., He, K.-T., Jing, N.: SRN: An Extended Petri-Net-Based Workflow Model for Web Service Composition. In: Proceedings of IEEE International Conference on Web Services (ICWS'04) , pp. 591–599. IEEE Computer Society Press, Los Alamitos (2004)

6. Yu-Bin, G., Yu-Yue, D., Jian-Qing, X.: A CP-Net Model and Operation Properties for Web Service Composition. Chinese Journal of computer 29(7), 1067–1075 (2006)
7. Tang, F., Li, M., et al.: Petri-Net-Based Coordination Algorithms for Grid Transactions. In: Cao, J., Yang, L.T., Guo, M., Lau, F. (eds.) ISPA 2004. LNCS, vol. 3358, pp. 499–508. Springer, Heidelberg (2004)
8. Murata, T.: Petri Nets: Properties, Analysis and Application. Proceedings of IEEE 77, 541–584 (1999)
9. Jensen, K.: Coloured Petri nets: Basic Concepts, Analysis Methods and Practical Use, Basic Concepts. Mono-graphs in Theoretical Computer Science. vol. 1, Springer, Heideberg (1997)
10. Zuberek, W.M.: Timed Petri nets: Definitions, Properties and Applications. Microelectronics and Reliability 31(4), 627–644 (1991)