The Performance Analysis of Web Service Composition Based on Timed Colored Petri Net

Baojun Tian
College of Information, Inner Mongolia University of Technology, Hohhot, China
ngdtbj@126.com

***Abstract*—On the premise of meeting functional requirements, the performance of web service composition is key to customers. Therefore, performance modeling of web service composition has been an important research direction of reengineering. Time coloured Petri net (TCPN) can not only establish a simulation model for system behaviour, but also extract or collect many performance analysis results from the model combined with predefined data monitors, which can satisfy user demands for the performance of QOS, and solve the problems which stochastic process algebra and queue network suffer when analyzing system performance.**

***Index Terms* —TCPN, modeling, performance analysis**

I. Introduction

With the continuous development of computer technology, constructing the loosely coupled collaborative software system, which can flexibly integrates program or legacy system distributing in heterogeneous environment, has become an important research field of software engineering. Service oriented computing (SOC) appear in order to solve above problems. And web service has become one of the most important computing resources on the Internet. Meanwhile, along with the its combination standards and technology continuing to develop, web service has been the core technology of SOC[2].

Web service is based on the Internet and independent on development environment. As long as following web service architecture, they can be integrated and called by each other.

Web service has the following characteristics.

- Good encapsulation

To the users, it can only see the list of functions provided by web service.

- Loosely coupled

For the enterprise application, web service provides a loosely coupled and distributed computing environment which is independent on any platform and language.

- Standard protocols

Web service's functions are described by the standard WSDL. Communication protocols between services uses SOAP. And web service agent center uses the universal description, discovery and integration protocol UDDI.

According to application requirements, web service composition may consist of many autonomous web services, thus it may provide new, more powerful and value-added functions.

The methods of web services composition are parted into following categories.

- Web service composition based on business processes.
- Web service composition based on workflow.

The working pattern of web service composition is similar as workflow, such as, Business Process Execution Language for Web Services(BPEL4WS), Bussiness Process Modeling Language(BPML) and W3C's Web Services Choreography Definition Language(WS-CDL).

- Web service composition based on formal method.

This approach utilizes mathematics model and formal tools to describe web service composition. Petri net is a formal model of concurrency. Process algebra is formal language, which can describe and reason process behaviour. Famous process algebra includes Milner's Calculus of Communication System(CCS), Hoare's Calculus of Sequential Process(CSP) and Pi calculus etc[8][9].

- Web service composition based on artificial intelligence[1].

This method mainly uses planning algorithm of artificial intelligence for web service composition[12].

- Hierarchical Task Network(HTN) planning

It uses OWL-S as a web service description language, and form a composition service through the method of task decomposition.

- Situation Calculus

It belongs to the first-order logic language programming method. And it is the formal predicate calculus of state, action and state results.

- Rule-Based planning

Using the rules to determine whether two services can be combined.

- theorem proving method

It uses a linear logic (LL) method of theorem proving to achieve semantic web service composition.

On the premise of meeting functional requirements, the performance of service composition is important to a

Project number: science research project of Inner Mongolia University of Technology(ZD201317)

doi:10.4304/jsw.8.12.3152-3157

system, which may reflect web service composition's quality[6][11].

There are two main methods for performance evaluation of web service composition.

- method of experiment bed

Experiment bed need build a real system, and it directly obtains performance indexes or parameters from monitoring procedures or instrument to predict the performance of a system through the statistical methods. Its main drawback is that construction of a real system would be expensive and cost more time.

- modeling method

The method analyzes the performance of a system by establishing its simulation model. It can not only analyze the performance of an existing system but also carry out performance prediction of a system that does not yet exist, at the same time which can find and correct the defects of design early. Using Pi calculus, queue network and Petri net, this research method mainly sets up simulation model evaluating system performances by means of qualitative and quantitative analysis[13].

The qualitative model mainly reflects logical structure and operation mechanism. It chiefly includes automaton, process communication process (CSP), communication calculus system (CCS), UML and Petri net etc. Quantitative model mainly describes the quantity indexes of system performance. It basically includes stochastic process algebra, queue theory and Markov processes, timed Petri nets and stochastic Petri net[15].

Pi calculus analyzes system performance basically by Markov chain, however, which cannot solve the problem of "state space explosion".

Only if it has product solution, queue network does not exist "state space explosion" problem. But for the discrete-event queue model, it really has not the product solution, which is easy to appear "state space explosion"[10].

Similarly, stochastic Petri net has strong ability of describing model, but with the complexity increasing, there is still "state space explosion" problem[7].

Timed Colored Petri net combines Petri net, programming language and time together, which improves the abilities of processing data and time. At the same time, it introduces the concept of color of token, the constraints for the arcs and transitions, which can reduce the complexity of model. By these approaches, it effectively solves the "state space explosion" problem, and by powerful Petri net simulation tools it can directly extract and analyze experiment data from the model for performance analysis. Therefore, it is widely used in the performance analysis of a large and complex web service composition[4][16].

The basic performance indexes of web service composition mainly include the followings.

- response delay

It is the time interval between request sent by user and obtaining service.

- response time

It is the time interval between a request of web service composition sent by user and the response received by user.

- response per second

In the non-idle state(i.e., the load of web service composition is not zero), it is the numbers that web service composition responds to the request per second.

- system capacity

In the non-idle state, it is the maximum number of web service composition responds to the request per second, i.e., the maximum of response per second.

- system bandwidth

A web service composition can transmit the number of bytes per second.

- maximum stable connections

When the performance is relatively stable and there is no obvious decline, a web service composition can support the maximal number of simultaneous connections.

II. RELATED THEORY KNOWLEDGE

A. Formal Definition of TCPN

Timed Colored Petri net that is based on classical Petri net introduces two parameters, i.e., color and time respectively. By color concept, different kinds of data can be imported. And with time parameter, the delay in web service composition is simulated[3][5].

A TCPN is seven tuples TCPN=(P, T, F, C, G, A, Dt) satisfying the requirements below [14] .

- (P, T, F) is a basic net.
- C is a colour set that is finite set of non-empty,
  $\forall p\in P$, C(p) is a token color set of place p,
  $\forall$ $t\in T$, C(t) is a token color set of transition t.
- A is an arc expression function.
- G is a guard function. It is defined from T into expression such that.
  $\forall t\in T$. Type(G(t))$\in$Boolean, G(t)=null, Type(G(t)) =true.
- Dt is time delay function. It can be defined in an arc and also be defined in a transition.

B. Dynamic Properties of Time Coloured Petri Net

When modeled by a TCPN, web service composition is analyzed and verified by dynamic properties.

- Reachability Property

Definition 1. Let (TCPN,M_0) time Coloured Petri Net, fire sequence $\sigma=t_1t_2\ldots t_n$, the sequence of states and transitions $M_0t_1M_1t_2\ldots t_nM_n$ and from (TCPN,M_0) a set of all possible fire sequence L(TCPN,M_0) or L(M_0) be given.

Using fire sequence σ, the state M_n is reachable from M_0, i.e., $M_0[\sigma>M_n$, iif.

$\sigma= t_1t_2\ldots t_n\in L(M_0)$ and $M_0t_1M_1t_2\ldots t_nM_n$ exists.

And all reachable states from M_0 are labeled as R(PN,M_0) or R(M_0).

- Boundedness Property

Definition 2. (TCPN,M_0) is k-Boundedness($k \in N$) iif. $\forall p, M \in R(M_0)$. $M(p) \leq k$.

- Liveness Property

Definition 3. (TCPN,M_0) is live, there is no transitions that cannot fire by proper transition sequence. i.e. , iff. $\forall M \in R(M_0)$: $\exists \sigma = t_1 t_2 \ldots t_n \in L(M_0)$, $\forall t_i$ is fired.

- Home Property

Definition 4. Let a marking $M \in R(M_0)$ be given. M is a home marking, iff.
$\forall M' \in R(M_0)$: $M \in [M'\rangle$.

- Coverability Property

Definition 5. Let a marking $M \in R(M_0)$ be given. M is a coverability marking , iff.
$\exists M' \in R(M_0)$: $\forall p \in P$, $M(p) \leq M'(p)$.

C. The structure of TCPN for Web Service Composition

- sequence activity

```
<sequence>
   activityA
   activityB
</sequence>
```

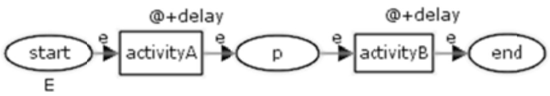

Figure 1. Sequence activity.

- if activity

```
<if>
   <condition>
      condition
   </condition>
   activityA
<else>
   activityB
</else>
</if>
```

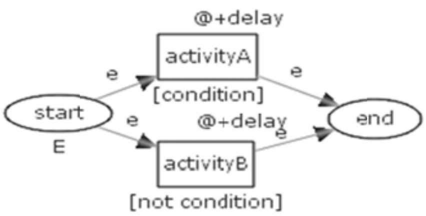

Figure 2. If activity.

- while activity

```
<while>
   <condition>
      condition
   </condition>
   activityA
</while>
```

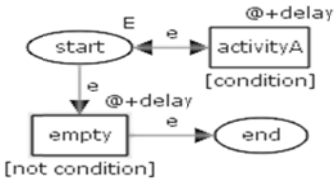

Figure 3. While activity.

- switch activity

```
<switch>
   <case condition="Bool-expr">
      activityA
   </case>
   <otherwise>
      activityB
   </otherwise>
</switch>
```

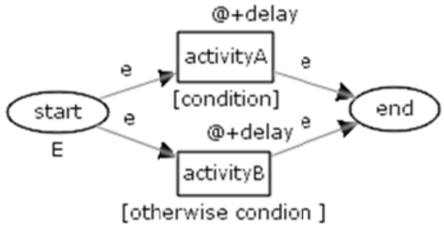

Figure 4. Switch activity.

- pick activity

```
<pick>
   <onMessage…>
      activity A
   </onMessage>
   <onMessage…>
      activityB
   </onMessage>
</pick>
```

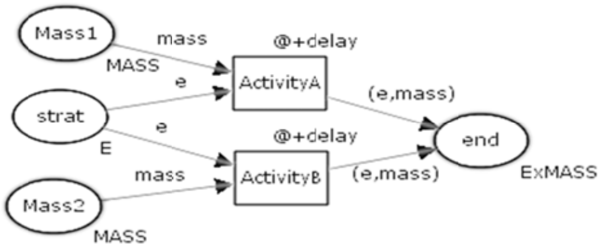

Figure 5. Pick activity.

- flow activity

```
<flow>
   activityA
   activityB
</flow>
```

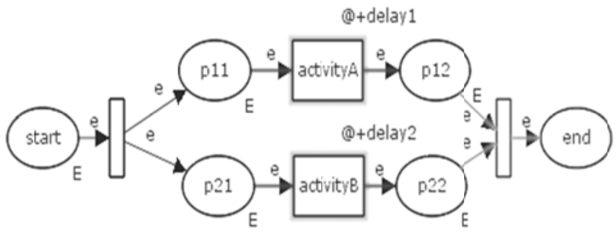

Figure 6. Flow activity.

- forEach activity

```
<forEach parallel="yes|no" counterName="N" …>
   <startCounterValue>1</startCounterValue>
   <finalCounterValue>n</finalCounterValue>
   <scope>
      ……
      activity
   </scope>
</forEach>
```

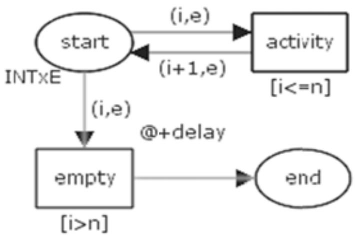

Figure 7. for Each activity (parallel="yes").

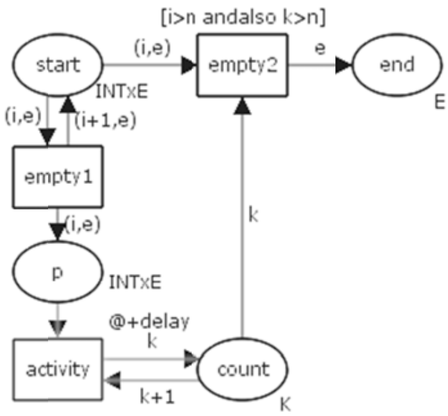

Figure 8. for Each activity (parallel="no").

III. An Instance Of Performance Analysis

In this paper, a hospital outpatient management system (a web service composition)modeled by TCPN as an example illustrates the performance analysis of web service composition. It includes two web services, i.e., registering service and diagnosing service.

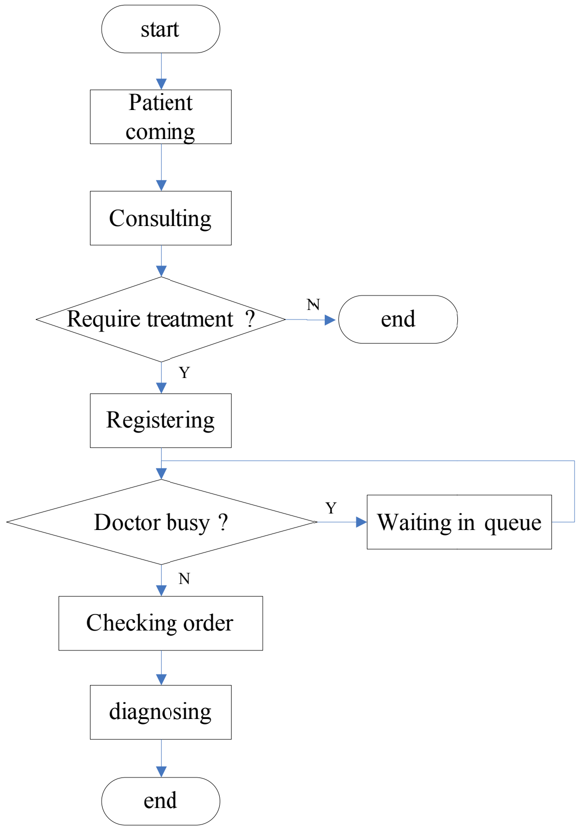

Figure 9. A flowchart of hospital outpatient service.

The web service composition is described by TCPN in figure 10. The colour sets, variables and data monitors are defined by CPN ML as follows.

Delaration of colour sets, variables.

```
colset INT=int;
colset SUM=with s;
colset IntT=int timed;
colset Card=product INT*INT;
colset Patient=with patient;
colset PatientxCard=product Patient*Card;
colset E=with e timed;
colset STRING=string;
var numofDoc.INT;
fun intTime( )=IntInf.toInt(time( ));
fun round x=floor(x+0.5);
fun disExp x=round(exponential(x));
```

Definition of data collector monitor.

```
▼Monitors
  ▼Time Delay
    ▶Type: Data collection
    ▶Nodes ordered by pages
    ▼Predicate
      fun pred (bindelem) =
      let
       fun predBindElem (TCPN'Begin_Diagnosis (1,
                         {arrivalTime,n,runTime})) = true
         | predBindElem _ = false
      in
       predBindElem bindelem
      end
    ▼Observer
      fun obs (bindelem) =
      let
       fun obsBindElem (TCPN'Begin_Diagnosis (1,
                         {arrivalTime,n,runTime})) =(intTime()-arrivalTime) div 60
         | obsBindElem _ = ~1
      in
       obsBindElem bindelem
      end
```

Definition of marking size monitor.

```
▼Marking_size_TCPN'Waiting Diagnosis_1
  ▶Type: Marking size
  ▼Nodes ordered by pages
    ▼TCPN
      Begin_Diagnosis (transition)
      Patients_Arriving (transition)
      Waiting_Diagnosis (place)
```

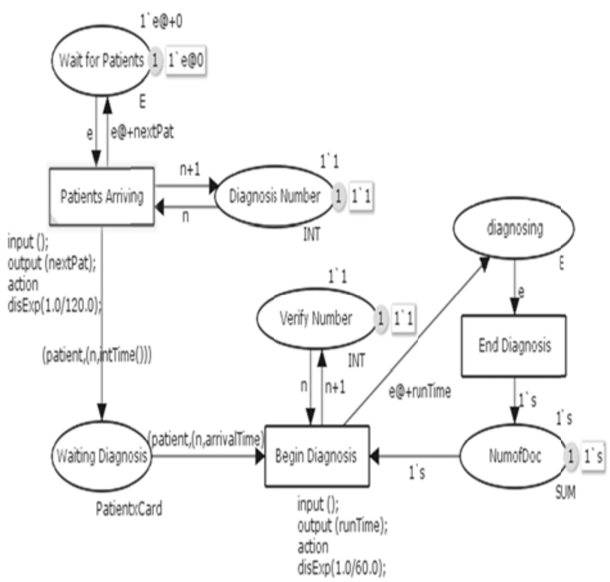

Figure 10. TCPN model of hospital outpatient service.

TABLE I.
A SIMULATION PERFORMANCE REPORT

Time statistics							
Name	Count	Avrg	95% Half Length	Min	Max	First	Last
Marking_size_TCPN'Waiting_Diagnosis_1	69	0.416703	0.250004	0	5	0	1
Time_Delay	33	0.494404	0.358452	0	4	0	0

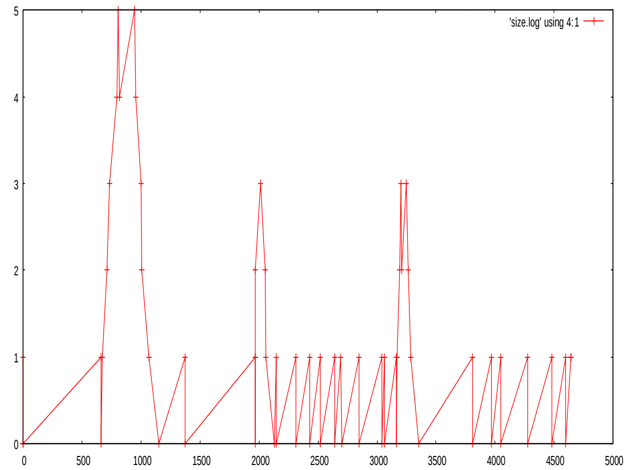

Figure 11. plotting data for Marking_size_TCPN'Waiting_Diagnosis_1 monitor.

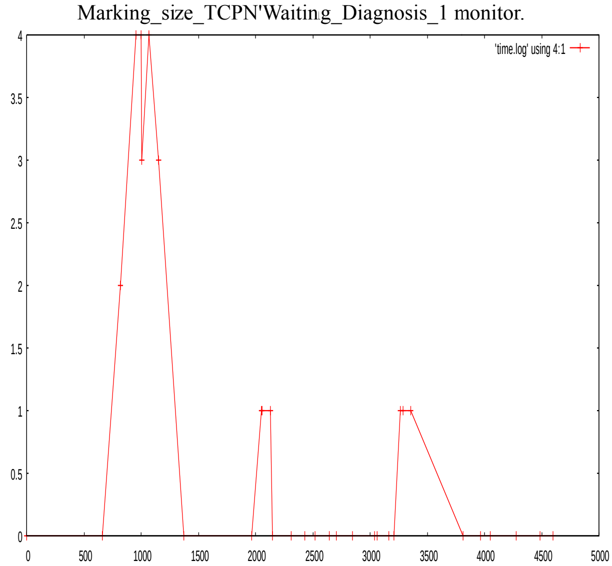

Figure 12. Plotting data for Time_Delay monitor.

IV. CONCULSION

Software performance analysis is conducted for different reasons, e.g., to evaluate existing or planned systems, to compare alternative configurations, or to find an optimal configuration of a key system. Time colored Petri net model has powerful capabilities in describing the behaviour of web service composition's parallel, pick and synchronization processing. It also solves the problem of "state space explosion" by hierarchical concept when analyzing a complex system. It has been widely used in software simulation modeling and performance analysis.

REFERENCES

[1] Manuel Noguera,Maria V. Hurtado, Ontology-driven analysis of UML-based collaborative processes using OWL-DL and CPN, Science of Computer Programming, vol.75, NO. 8, pp. 726–760, Aug 2010.

[2] Martha Varguez-Moo, Francisco Moo-Mena and Victor Uc-Cetina, Use of Classification Algorithms for Semantic Web Services Discovery, Journal of Computers, vol .8, No. 7, pp. 1810-1814, Jul 2013.

[3] Attila Toth, Erzsebet Nemeth, Coloured Petri net diagnosers for lumped process systems, 2010 Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems, pp. 389–398, Sep 2010.
[4] PLisa Wells, Performance analysis using CPN tools, Proceedings of the 1st international conference on Performance evaluation methodolgies and tools, pp. 59 – 69, Oct 2006.
[5] Yudith Cardinale, Joyce El Haddad, CPN-TWS. a coloured petri-net approach for transactional-QoS driven Web Service composition, International Journal of Web and Grid Services, vol. 7,NO. 1, pp. 91– 115, Feb 2011.
[6] Mercedes Narciso, Miquel A. Piera and Antoni Guasch, A Methodology for Solving Logistic Optimization Problems through Simulation, Simulation,vol. 86, NO. 5, pp. 369–389, May 2010.
[7] Guo Juan, Research on credibility of web service composition based on Stochastic Petri net, Advances in Intelligent Systems and Computing, vol.212, pp. 965–973, 2013.
[8] Sun Ping, Interactive Web service composition based on Petri net, Transactions of the Institute of Measurement and Control, vol. 33, NO.1, pp. 116–132, Feb 2011.
[9] Li Bin, A Petri-net and QoS Based Model for Automatic web Service Composition, Journal of Software, vol. 7, NO. 1, pp.149–155, 2012.
[10] Juan Song, Hao Wang, Performance analysis for web service composition based on Queuing Petri Net, Proceedings of 2012 IEEE 3rd International Conference on Software Engineering and Service Science, pp.501–504, 2012.
[11]Piergiorgio Bertoli,Marco Pistore, Automated composition of Web services via planning in asynchronous domains, Artificial Intelligence, vol. 174, NO.3, pp. 316-361, March 2010.
[12] Guangjun Guo, Fei Yu, Zhigang Chen and Dong Xie, A Method for Semantic Web Service Selection Based on QoS Ontology, Journal of Computers, Vol. 6, No. 2 pp. 377-386, 2011.
[13] Xu Chunxiang, A petri net-based method for data validation of web services composition, Proceedings - International Computer Software and Applications Conference, pp. 468–476, 2010.
[14] Kurt Jensen, Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems, International Journal on Software Tools for Technology Transfer, Vol. 9, pp. 221–251, 2007.
[15] Zou Jinan, Study on semantic web service composition based on Petri net, Proceedings of 2012 7th International Conference on Computer Science and Education, pp. 1110–1114, 2012.
[16] Guisheng Fan, Huiqun Yu, Liqiong Chen and Tong Ruan, An Approach to Analyzing Time Constrained Service Composition, Journal of Computers, Vol 6, No. 8, pp. 1723-1731, 2011.

Associate professor Baojun Tian. 2000-2003, Master Degree in Computer Application Technology, Department of Computer Science, Inner Mongolia University, China. 1992-1996, Bachelor Degree in Mathematics, Department of Mathematics Science, Inner Mongolia University, China. His major fields of study are software engineering, formal modeling and web service composition.