An Approach to Analyzing User Preference based Dynamic Service Composition

Guisheng Fan
Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai 201112, China
Email: gsfan@ecust.edu.cn

Huiqun Yu[1], Liqiong Chen[2], Caizhu Yu[1]
[1] Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
[2] Department of Computer Science and Information Engineering Shanghai Institute of Technology, Shanghai 200235, China
Email: yhq@ecust.edu.cn, lqchen@sit.edu.cn, yucaizhu1985@126.com

***Abstract*—As a way to compose independent service together to fulfill a function, service composition is an important means for flexible and rapid information integration of complex distribution application in open and heterogeneous environment. However, the diversity of requirements makes it difficult to guarantee the correctness of service composition. This paper proposes a hierarchical dynamic service composition net (*HDSC-net*), and user's preference based mechanism for service composition. *HDSC-net* is used to model operation, the relationships between operations, operation mapping, and user's preference. Transfer matrix is constructed to express the relationships between states, while theories of Petri nets help prove the composability of service. The strategy for dynamic service composition and its corresponding enforcement method are also proposed. A case study of Travel Service demonstrates the applicability of proposed method and its effectiveness.**

***Index Terms*—Web service, service composition, user's preference, transfer matrix, composability**

I. INTRODUCTION

Service Oriented Computing (SOC) is an approach to distributed computing that views software resources as dynamically discoverable services available on the Internet. Web services are a well known and widely used technology for implementing SOC [1]. For example, in e-business, tourism and other service areas, more and more services have been published in the form of Web services. As a single Web service can provide limited function, it is necessary to compose Web services to provide a more powerful service [2].

With the expand applications of Web services, inadequacies of Web services appear. First, how to dynamically select component service at run time and how to express user preference; Second, the correctness of compositions requires not only the satisfaction of functional requirements, but also of non-functional properties, such as reliability and resource consumption. To address these problems, several theoretical models have been proposed in the literature, including finite state machine, Petri net,π-calculus, et al [3]. However, few of the above works provide strategies or mechanisms to select component service dynamically during the execution of a series of operations in service composition, and user preference is not taken into account too.

To tackle the above problems, Petri nets are used to model and analyze user preference based dynamic service composition in this paper. The main contributions are: First, we propose the hierarchical dynamic service composition net (*HDSC-net*), and use it to simulate the process of service composition; Second, according to the characteristics of target service and available service, the *HDSC-net* model is used to describe target service, operation, operation relationships and user preference; Third, we propose the concept of transfer matrix to represent the relationships between states, Petri net and its state space help prove the composability of service, the strategy for constructing dynamic service composition and its enforcement are also proposed.

The remainder of this paper is organized as follows: Section 2 gives the requirements of dynamic service composition and the definition of *HDSC-net*. In section 3, we construct the *HDSC-net* model of service composition. Section 4 is the analysis of *HDSC-net* model. In Section 5, we explain the feasibility and practicability of our methods by a specific example. Section 6 presents some related works while section 7 is conclusion.

II. COMPUTATION MODEL

A. Requirements of Dynamic Service Composition

The service composition needs to bind its operation to specific component before execution, which makes service composition be called to operate only after every activity of composition process is designated by actual Web services, thus generating the schemas of service composition [4].

doi:10.4304/jsw.5.9.982-989

Definition 1: A Web service is defined by a triple $WS=\{O;R;Su\}$, where:

(1) O is a finite operation set;

(2) $R: O\times O\rightarrow\{>,+,\|,n\}$ is the relation function between operations, where >, +, ||and nO_i represent the sequence, choice, parallel and loop relationships between operations, and sequence relationship has highest priority;

(3) $Su:O\rightarrow(0,1)$ is the success probability of operation.

Definition 2: Let WS be a Web service. A tuple $SC=\{TWS,AWS\}$ is called service composition, where:

(1) TWS is the target service;

(2) AWS is the available service set.

Definition 3: Let $TWS=\{O,WS,Su\}$ be a target service, if CO meets the following conditions:

(1) $\forall O_i,O_j\in CO:R(O_i,O_j)=+$

(2) $\forall O_i\in CO,\forall O_j\in(O\text{-}CO):R(O_i,O_j)\neq+$

Then CO is called a choice set of TWS. The set of all choice set in TWS is denoted by $ACO(TWS)$.

Definition 4: Let $TWS=\{O,WS,Su\}$ be a target service. If SCO is the sorted CO set, then SCO is called user preference according to choice set CO.

User preference means that the user can specify priority relationship of operation when composition process has multiple choices. For example $O_1 + O_2 + O_3$, if the user preference is $\{O_2,O_1,O_3,\}$, then the system will firstly choose operation O_2 when the available services can complete the above mentioned operation. If O_2 has failed or no available service can provide the operation, then the system will choose operation O_1.

B. Hierarchical Service Model

Petri net is a formal language for describing the concurrent system because its semantics is formally defined. Some recent researches indicate that Petri nets are powerful and expressive enough to describe behavior features of service composition [14,15]. The basic concepts are referred to [10].

Definition 5: A six tuple $PN=(N,I,O,Pr,\lambda,M_0)$ is called dynamic service composition net (*DSC-net*), iff:

(1) $N = (P, T, F, W)$ is a basic Petri net;

(2) $I\subset P$ is a special place, which is called the interface of PN and denoted by dotted circle;

(3) $O \subset T$ is a special transition, which is called the operation of PN and denoted by straight line;

(4) $Pr: T\rightarrow(N^*\times N^*)$ is the priority function of transition. $Pr(t_i)=(\alpha_i,\beta_i)$, where α_i, β_i are called primary and secondary priority of transition t_i;

(5) $\lambda:T\rightarrow(0,1]$ is the firing probability of transition, the default value is 1;

(6) M_0 is the initial marking of PN.

Definition 6: A seven tuple $\Omega=\{PN,\Sigma,\Gamma,TI,TA,PI,PA\}$ is called hierarchical dynamic service composition net (*HDSC-net*), iff:

(1) PN is a *DSC-net*;

(2) $\Sigma=\{\Sigma_i|i\in N^*\}$ is a finite set of *HDSC-net* and *DSC-net* model, and each element is called a page of Ω;

(3) $\Gamma:\Sigma\rightarrow T^*$ is the operation set of each page, which is called page operation;

(4) $TI\subset T$ is the set of substituted operation and denoted by double rectangle;

(5) $TA: TI\rightarrow\Gamma^*$ is allocating the corresponding page to substituted operation;

(6) $PI \subset P$ is the set of interface node, which describes the input and output of substituted node, and denoted by double circle;

(7) PA is the mapping function of interface, which maps the interface node into the input and output of the operation in corresponding page;

HDSC-net is a hierarchical modeling language for target services. Each page represents an available service, and substituted operation represents an operation of target service, which can operate only after mapping into the operation of available service.

The distribution of token in each place is called the marking of *HDSC-net* model, denoted by M. The marking $M(p)$ denotes the number of tokens in place p. For any $x\in(P\cup T)$, we denote the pre-set of x as ${}^\bullet x=\{y|y\in(P\cup T)\wedge(y,x)\in F\}$ and the post-set of x as $x^\bullet=\{y|y\in(P\cup T)\wedge(x,y)\in F\}$. A tuple $S = (M,TP)$ is called a state of *HDSC-net* model, where M is a marking and TP is the probability of reaching the state. For transition $t_i\in(T\text{-}O)$, if $\forall p_j\in{}^\bullet t_i$: $M(p_j)\geq W(p_j,t_i)$, then transition t_i is enabled under S, denoted by $S[t_i>$. All enabled transitions under state S are denoted by set $ET(S)$.

Definition 7: Let S be a state of Ω, for transition $t_i\in ET(S)$, if t_i meets the following conditions:

$\alpha_i\leq min(\alpha_j)\wedge\beta_i\leq min(\beta_k)$, where $t_j\in ET(S)$, $t_k\in U(t_i)$

Then the firing of transition t_j under state S is effective. All the effective firing transitions under state S are denoted by set $FT(S)$.

Definition 8: Let S be the state of Ω, $t_i\in FT(S)$. The model will reach a new state $S'=(M',TP')$ by effectively firing enabled transition t_i, denoted by $S[t_i>S'$. M', TP' are calculated according to the following rules:

(1) Computing marking:

$\forall p_j\in{}^\bullet t_i\cup t_i^\bullet, M'(P_j)=M(P_j)-W(P_j,t_i)+W(t_i,P_j)$

(2) Computing reach probability TP': $TP'=TP*\lambda(t_i)$

If there exists a firing sequence $t_1, t_2, \ldots, t_k$ and a state sequence $S_1, S_2, \ldots, S_k$, such that $S[t_1>S_1[t_2>S_2\ldots S_{k-1}[t_k>S_k$, then S_k is reachable from state S. All possibly reachable states of S are denoted by $R(S)$, and $S\in R(S)$.

C. Modeling Service Composition

(1) Modeling Operation

In this paper, we use a prefixed notation to denote the operation in a specific service. For example, operation O_i of service WS_k is denoted by $WS_k\bullet O_i$, if the operation belongs to target service, then not marking. In the same way, the service and operation are marked in the left upper of place and transition. While the transitions in target service only mark the corresponding operation.

In the target service model, each operation O_i is abstracted as a transition TO_i, which is shown in Figure 1(a). The place $P_{s,i}$, $P_{e,i}$ represent the condition and output parameters of operation O_i. While operation $WS_k\bullet O_i$ is abstracted as a model shown in Figure 1(b). P_{in} and P_{ou} represent the interfaces between target service and $WS_k\bullet O_i$. While P_{fo} represents the parameters that $WS_k\bullet O_i$ transfers to target service after failing. Only the operations of

service with user preference have interface P_{fo}.

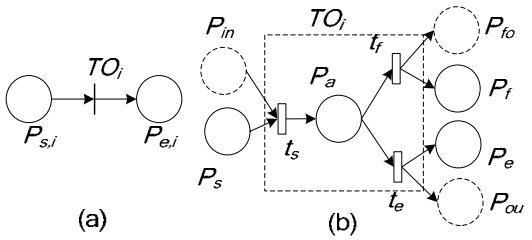

Figure 1. The *HDSC-net* Model of Operation

(2) Modeling Operation Relationships

We use available services as an example to construct the *HDSC-net* model of basic operation relationships. Because choice relationship involves user preference, we only model for sequence, parallel and loop relationship.

The *HDSC-net* model of sequence relationship $O_i > O_j$ is shown in Figure 2(a). We introduce transition t_{ij} to make the output of forward operation O_i transfer to the input interface $P_{s,j}$ of operation O_j.

If the relationship of operation O_i and O_j is sequence, then O_i is called forward operation of O_j, O_j is called afterward operation of O_i. The set $Pre(O_i)$, $Post(O_i)$ are the forward operation set and afterward operation set of operation O_i respectively.

The *HDSC-net* model of parallel relationship $O_i||O_j$ is shown in Figure 2(b). Let operation O_f, O_k meet $O_f=Forw(O_i)\cap Forw(O_j)$, $O_k=Back(O_i)\cap Back(O_j)$. We introduce transition $t_{f,ij}$ to make the output results ($P_{e,f}$) of forward operation $O_f$transfer to the input interface $P_{s,i}$, $P_{s,j}$ of operation O_i and O_j. While transition $t_{ij,k}$ is to make the output results of operation O_i and O_j transfer to afterward operation O_k. And ${}^\bullet t_{f,ij}=P_{e,f}$, $t_{f,ij}{}^\bullet=\{P_{s,i},P_{s,j}\}$, ${}^\bullet t_{ij,k}=P_{t,k}$, $t_{ij,k}{}^\bullet=\{P_{e,i},P_{e,j}\}$.

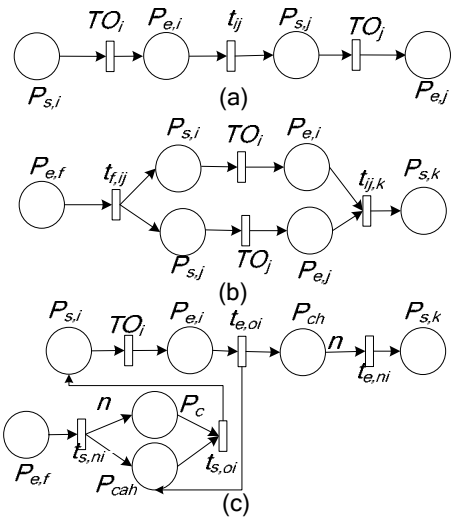

Figure 2. Modeling Basic Operation

The *HDSC-net* model of loop relationship nO_i is shown in Figure 2(c). Let operation O_f, O_k meet $O_f=Forw(nO_i)$, $O_k=Back(nO_i)$. The transition $t_{s,ni}$, $t_{e,ni}$ describe the beginning and termination operation of nO_i. While transition $t_{s,oi}$, $t_{e,oi}$ describe the beginning and termination of running operation O_i one time: $t_{s,i}{}^\bullet=P_{s,I}$, ${}^\bullet t_{e,i}=P_{e,i}$. Place P_c is used to store the number of not running times of operation O_i.

(3) Modeling Mapping Operation

For the operation O_k in target service, if all available services in service set $\{WS_{i1}, WS_{i2}, ..., WS_{im}\}$ have the operation O_k, then the *HDSC-net* model of mapping process is shown in Figure 3. The substituted transition $TO^{ij}{}_k$ represents the operation of $WS_{ij}\bullet O_k$. We introduce $t_{s,ij,k}$ and $t_{e,ij,k}$ to represent the beginning and result output of operation $WS_{ij}\bullet O_k$. While $P_{ij,k,in}$ and $P_{ij,k,ou}$ represent the outside input and output of operation $WS_{ij}\bullet O_k$.

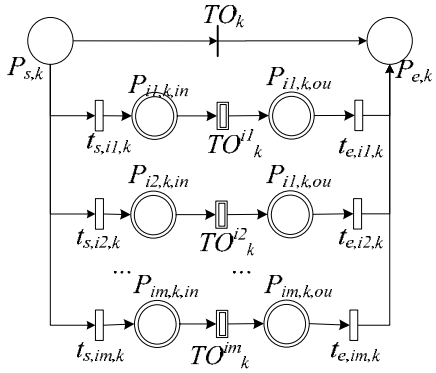

Figure 3. Modeling Mapping Operation

(4) Modeling User Preference

Assuming that there exist choice relationship in operation set $\{O_{i1}, O_{i2}, ..., O_{im}\}$, the *HDSC-net* model of user preference is shown in Figure 4.

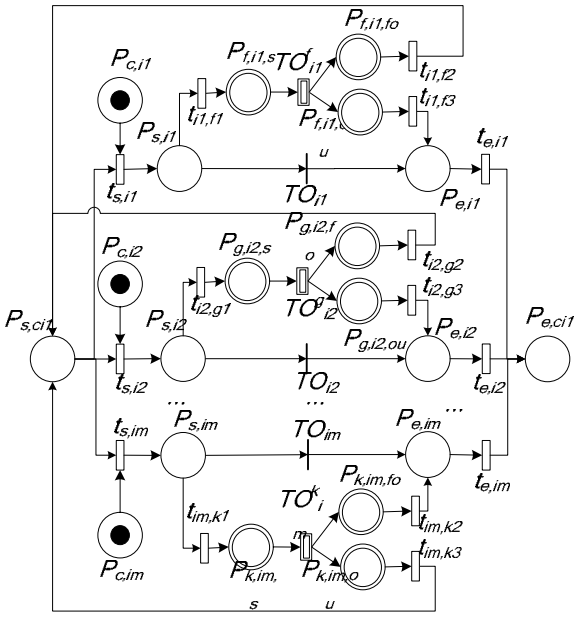

Figure 4. Modeling User Preference

1) Introducing place $P_{s,ci1}$ and $P_{e,ci1}$ to represent the beginning and termination operation of choice process;

2) Transition $t_{s,ij}$ represents the beginning of the invoked operation. These transitions may have different priority because of user preference. The public pre-set of $t_{s,i1}$, $t_{s,i2}$,..., $t_{s,im}$ is place $P_{s,ci1}$. Therefore, the operation which has the highest priority can be invoked to execute;

3) For operation O_{ij}, we introduce place $P_{c,ij}$ to control each operation can only be chosen one time, and $P_{c,ij}{}^{\bullet}=t_{s,ij}$, ${}^{\bullet}P_{c,ij}=\Phi$, $M_0(P_{c,ij})=1$;

4) If current operation O^l_{ij} has failed, that is, there have tokens in place $P_{l,ij,fo}$, then calling failure transformation transition $t_{ij,l2}$ to transfer token to place $P_{s,ci1}$.

5) If current operation O^l_{ij} has operated successfully, that is, the place $P_{l,ij,ou}$ has tokens, then calling termination output transition $t_{ij,l3}$ to transfer token to $P_{e,ci1}$.

(5) Modeling Service

The specific steps of modeling service are shown in the following.

1) Introducing initial place P_s and transition t_s, such that ${}^{\bullet}t_s=\{P_s\}$, $t_s{}^{\bullet}=\{P_{s,i}|Fork(O_i)=\Phi\}$, ${}^{\bullet}P_s=\Phi$, $P_s{}^{\bullet}=\{t_s\}$, $M_0(P_s)=1$;

2) Constructing the running process of service based on the relationships between operations;

3) Introducing termination place P_e and transition t_e, such that ${}^{\bullet}t_e=\{P_{e,i}|Back(O_i)=\Phi\}$, $t_e{}^{\bullet}=\{P_e\}$, ${}^{\bullet}P_e=\{t_e\}$, $P_e{}^{\bullet}=\Phi$.

4) For the target service, we need to model mapping operation according to the corresponding available service, thus forming the composition process;

5) Allocating priority to transition: the primary priority of choice operation's inner transition is equal to the primary of corresponding operation, the priority of transition in others operation is equal to 0; The secondary priority of transition is equal to the priority of service, which is allocated by using dynamic strategies introduced behind; If the operation is introduced to describe the relationship of process, then its priority is (0,0); When the operation is mapped, the priority of each transition of invoked operation is equal to the priority of the corresponding operation.

III. Analysis Technologies of Service Composition

A. The Transfer Matrix of HDSC-net

HDSC-net model starts from initial state S_0 and generates new state through effectively firing enabled transitions, thus establishing a state space (known as state graph). The state graph takes state as node, and transition as edge. The edge is labeled by firing probability. We can analyze the related properties through the state graph of *HDSC-net*. However, the state graph of *HDSC-net* may be complicated, it is difficult to analyze it by directly computing. It is necessary to further abstract state graph.

Theorem 1: In *HDSC-net* model, if S_j can reach from S_i, then S_i can reach state S_j in finite steps.

Proof: The proposition is equivalent to proving that model does not have deadlock and endless loop. Because services in *HDSC-net* model can be fired only after obtaining all required resources, and will not require additional resources. That is, *HDSC-net* model does not meet one of the necessary conditions of deadlock generated: the transition has been obstructed due to the requirement of other resources, and doesn't release its resource. Therefore, *HDSC-net* model does not have deadlock. Also we don't consider the infinite service invoked, so the model does not have endless loop.

Theorem 1 shows the state space of corresponding *HDSC-net* model is finite, thus we can realize to analyze service composition through it.

The state S is called termination state of *HDSC-net* model if $M(P_e)=1\wedge(\forall WS_i\in AWS\rightarrow M(WS_i{}^{\bullet}P_e)+M(WS_i{}^{\bullet}P_s)=1)$. All the possible termination state of *HDSC-net* model is denoted by $TS(\Omega)$. We introduce state S_{end} to represent the normal termination state of *HDSC-net* model, which makes $\forall S_i\in TS(\Omega)$, the probability from state S_i to state S_{end} is equal to 1. S is a state of Ω, then the probability from state S to state S_{end} is denoted by $Ter\_P(S)$, which is called termination probability of S.

Definition 9: The transformation probability a_{ij} from state S_i to state S_j meets the following conditions:

$$a_{ij}=\begin{cases}1 & S_i\in Term(\Omega), S_j=S_{end}\\ \lambda_{ij} & if\ t_{ij}\in T, S_i[(t_{ij},w_{ij})>S_j\\ 0 & otherwise\ t_{ij}\in T, S_i[(t_{ij},w_{ij})>S_j\end{cases} \quad (1)$$

Let the number of reachable state in *HDSC-net* model Ω be L, the L -order square matrix A is called transfer matrix of Ω if it meets following conditions: $A=[a_{ij}]_{L\times L}$, where a_{ij} is the transformation probability from state S_i to S_j . A^n is the nth power of A, while $a^{(n)}{}_{ij}$ is the element in the ith row and jth column of A^n. Vector $R^{(n)}{}_{i,A}$, $C^{(n)}{}_{j,A}$ represent the ith row and jth column of A^n respectively.

Theorem 2: The probability from state S_i to state S_j by n steps is equal to the value of $a^{(n)}{}_{ij}$ in A^n.

Proof: mathematical induction

(1) If n=1, we can draw the conclusions from the definition of a_{ij}.

(2) Assuming the proposition is established when $n\le k$, now we will prove the proposition is established when $n = k + 1$, $a^{(k+1)}{}_{ij} = R^{(k)}{}_{i,A} * C_{j,A} = \sum_{r=1}^{L-1} a_{ir}^{(k)} * a_{rj}$. Because the proposition is established when $n\le k$, that is, $a^{(k)}{}_{ir}$ is equal to the probability from state S_i to state S_r by k steps, while a_{rj} is equal to the probability from state S_r to state S_j by one step. Therefore $a^{(k)}{}_{ir} * a_{rj}$ is equal to the probability from state S_i to state S_r by k steps and reach S_j from S_r by one step. Because the choice of r is arbitrary, we can get $a^{(k+1)}{}_{ij}$ is equal to the probability from state S_i to S_j by $k+1$ steps, that is, the proposition is established when $n=k+1$.

Theorem 2 explains the probability from state S_i to state S_j by n steps is equal to the value of $a^{(n)}{}_{ij}$ in A^n. $a^{(n)}{}_{ij}$ is also called n order probability from state S_i to state S_j. We can convert the analysis of reliability into computing power of transfer matrix through Theorem 2.

B. The composability of Service

The basic requirement of service composition is to complete the function of target service based on the available services. Therefore, it is necessary to analyze the composability of service before giving dynamic service composition strategies.

Definition 10: Let Ω be an *HDSC-net* model.

(1) $\forall WS_i\in\{TWS\cup AWS\}$, $L(WS_i)$ is the possible operation sequence set, which is called operation language of service WS_i.

(2) $\forall WS_i\in\{TWS\cup AWS\}, \forall\sigma\in L(WS_i)$, σ is called a sentence of WS_i.

(3) $\forall\sigma\in L(TWS)$, σ^{in} is the operation firing sequence of

Ω that corresponding to σ, which is called an instantiation of σ.

(4) $\forall\sigma\in L(TWS)$, σ^{in} is an instantiated firing sequence. $\forall O_k\in\sigma^{in}$, $WS_i\in AWS$, if $\exists\sigma_f\in L(WS_i)$, $O_k\in\sigma_f$, the firing sequence $\sigma^{in}{}_{WS\to I}$ is got by removing the operation of $\{WS\text{-}WS_i\}$, which is called projection sequence of service WS_i. Denoted $PJ(\sigma^{in})$ as the set of different service's projection sequences that σ^{in} called.

(5) $\forall\sigma\in L(TWS)$, σ^{in} is an instantiated firing sequence, if $\exists\sigma^{in}{}_{WS\to i}\in PJ(\sigma^{in})$, such that $\sigma^{in}{}_{WS\to i}\notin L(WS_i)$, then σ^{in} is called an ineffective instantiation, otherwise is an effective instantiation.

The operation firing sequence from initial state to termination state in *HDSC-net* model belongs to an instantiation of the sentence in target service. Projection sequence explains the completed operation sequence of each sentence in operation language remains the same sequence in the corresponding available services. An ineffective instantiation means that actual available services have not reached the termination state, while the corresponding sentence of composition process has reached the termination state.

Definition 11: Let Ω be an *HDSC-net* model. $\forall\sigma\in L(TWS)$, if there exists an effective instantiation σ^{in}, that is, $\forall\sigma_i\in PJ(\sigma^{in})$, $\exists WS_i\in AWS$, such that $\sigma_i\in L(WS_i)$, then service composition *SC* is compositional.

The process that state S reaches state S′ by firing operation $WS_i.O_k$. is denoted by $S[WS_i.O_k\rangle S'$.

Definition 12: Let Ω be an *HDSC-net* model. The composition process reaches the current state S_w by firing operation sequence σ^w. Let S_{end} be the corresponding termination state, the directed graph $RG(\Omega)=(V,E)$ is the state graph of Ω, if there exists operation $WS_i.O_k$ and $S_w[WS_i.O_k\rangle S'$, then it must meet the following rules:

(1) $(\exists\sigma_j\in L(TWS), \sigma^w\subseteq\sigma_j)\to(\{\sigma_{now}\cup O_k\}\subseteq\sigma_j$

(2) $(\exists\sigma_f\in L(WS_i), \sigma^w{}_{WS\to i}\subseteq\sigma_f)\to(\{\sigma^w{}_{WS\to i}\cup O_k\}\subseteq\sigma_f$

(3) $S_{now}=S_{end}\to(\forall WS_i\in WS{:}M(WS_i.P_s)+M(WS_i.P_e)=1)$

Then the execution of $WS_i.O_k$ is feasible.

Theorem 3： Let Ω be an *HDSC-net* model. $\forall\sigma\in L(TWS)$, σ^{in} is an instantiated operation firing sequence of σ. σ^{in} is effective iff each firing operation is feasible.

Proof: (⇒)

Let the composition process reach current state S_w by firing the first *m* firing sequence $\sigma_{now}=\{WS_{i1}.O_{k1}, WS_{i2}.O_{k2}, \ldots, WS_{im}.O_{km}\}$ of σ^{in} and the next firing operation is $WS_{i(m+1)}.O_{k(m+1)}$.

∵ σ^{in} is an instantiated operation firing sequence of σ

∴ $\sigma_{now}\subseteq\sigma\wedge\{\sigma_{now}\cup O_{m+1}\}\subseteq\sigma$

Let $\sigma_j=\sigma$, therefore, σ^{in} meets the rule (1) of feasibility definition if σ^{in} is effective.

Similarly, we can prove that σ^{in} meets the rule (2) of feasibility definition if σ^{in} is effective.

∵ $S_w=S_{end}$

∴ $\sigma^w=\sigma\in L(WS_i)$

Let WS_{in} be the invoked available service set in the firing sequence σ^{in}, that is, $\forall WS_j\in WS_{in}$, $\exists\sigma_j\in PJ(\sigma^{in})$, which makes $\sigma_j\in L(WS_i)$.

$\forall WS_i\in WS\Rightarrow WS_i\in WS_{in}\vee WS_i\in\{WS\text{-}WS_{in}\}$

Case 1： $WS_i\in WS_{in}$

Then $\sigma^w{}_{WSi}\in L(WS_i)$

∴ $M(WS_i.P_e)=1$

Case 2: $WS_i\in\{WS\text{-}WS_{in}\}$, the composition process reach termination state by firing σ^w, but none of the operation of service WS_i has been invoked, therefore, the token in service WS_i remains unchanged.

∴ $\forall WS_i\in WS\Rightarrow M(WS_i.P_e)+M(WS_i.P_s)=1$

which is the rule(3) of feasibility definition.

(⇐) Let the invoked service set of σ^{in} be $WS_{in}\subseteq WS$, and the set $PJ(\sigma^{in})$ is the different service's projection sequence that σ^{in} invoked, then the firing sequence in $PJ(\sigma^{in})$ is one by one mapping relation with the service in WS_{in}.

∵ $\sigma\in L(TWS)$, that is $S_w=S_{end}$

According to the rule(3) of feasibility definition, we can get $\forall WS_i\in WS\Rightarrow M(WS_i.P_e)+M(WS_i.P_s)=1$.

∵ Each firing in σ^{in} is feasible

∴ Service composition reaches termination state after firing sequence σ^{in}, at the same time, all services that σ^{in} invoked also reach termination state.

That is, $\forall\sigma_i\in PJ(\sigma^{in})$, $\exists WS_i\in WS_{in}$, which makes $\sigma_i\in L(WS_i)$.

∴ σ^{in} is effective.

In summary, σ^{in} is effective iff each firing operation is feasible.

Theorem 3 shows that the effectiveness of firing sequence is associated with each firing operation. While the feasibility of firing operation can be converted into computing the reachability between states in transfer matrix.

C. Dynamic Service Composition Strategy

In a complicated service composition, the operation can be completed by a number of available services and these firings are feasible. Each firing may cause service composition have different reliability, therefore, it is necessary to choose the service which has the highest reliability.

Theorem 4: $\forall i, j < L'$, if state S_i and S_j are reachable, then there exists $K_{ij}\in N$ which makes $\forall E\in N$, $a_{ij}^{K_{ij}+E}=0\wedge a_{ij}^{K_{ij}}\neq 0$.

Proof: from Theorem 1, we can know that state S_i can reach state S_j in finite steps. We may assume that there has *q* firing sequences $\delta_1, \delta_2, \ldots, \delta_q$ from state S_i to state S_j. Set $K_{ij}=\max\{|\delta_1|, |\delta_2|, \ldots, |\delta_q|\}$, then $a_{ij}^{K_{ij}}$ represents the probability from state S_i to state S_j by K_{ij} steps. From the definition of K_{ij}, we can draw $a_{ij}^{K_{ij}}\neq 0$, and because K_{ij} is the maximum steps from state S_i to state S_j, therefore, $\forall E\in N$ there has $a_{ij}^{K_{ij}+E}=0$.

Among them, K_{ij} is called stability order of transfer matrix *A* from state S_i to state S_j. The probability from state S_i to state S_j in *A* is $P_{ij}=a_{ij}+a^{(1)}{}_{ij}+\ldots+a^{(Kij)}{}_{ij}$. That is, the reliability of service composition *SC* is got by computing the firing probability of all paths between states. The highest stability order among all state is called

the highest stability order of A.

Definition 13: Let Ω be an *HDSC-net* model. A is transfer matrix of Ω, K is stability order of A, matrix $B = A^1 + A^2 + \ldots + A^K$ is called reliable matrix of Ω.

The corresponding column $R_{end,B}$ of state S_{end} in matrix B is called termination vector of Ω, each element in $R_{end,B}$ represents the termination probability *Ter_P*(S) of S. Let S be a state of *HDSC-net* model, if the available service set $WS = \{WS_{k1}, WS_{k2}, \ldots, WS_{km}\}$ under S can complete the function of O_i and its firing is feasible, then the set WS is called feasible service set of operation O_i under S, denoted by $AWF(O_i, S)$. If $S[WS_i.O_k > S'$, then $\frac{Su(WS_{kj} \bullet O_i) * Ter\_P(S)}{|AWF(O_i, S)|}$ is called success probability after completing $WS_{kj}.O_i$ and denoted by $Ter(S, WS_{kj}.O_i)$.

Definition 14: Let Ω be an *HDSC-net* model, S is the state of Ω, $AWF(O_i, S)$ is the feasible service set of operation O_i under state S, the dynamic composition strategy is to choose the highest success probability service from $AWF(O_i, S)$ to complete the function of O_i.

Dynamic composition strategy is allocating the highest priority to the service which has highest success probability after mapping into the *HDSC-net* model. According to the definition of termination probability and success probability, we can draw that dynamic composition strategy can make composition process be highly reliable.

D. Enforcement of Dynamic composition Strategy

From the definition of dynamic service composition strategy, we can draw that each state must choose the highest success probability operation. The specific steps of constructing service composition are as follows:

(1) Constructing the *HDSC-net* model and its transfer matrix based on the requirements of service composition;

TABLE I.
CONSTRUCTING ALGORITHM OF SERVICE COMPOSITION

```
1: Comp_Any(SC,A) // composability analysis
2: For i=0, i<|L(TWS) |, i++ do
3: S=S0;
4: For j=1, j≤|δi|,j++ do
5: If ∃WSk∈AWS, which makes S[WSk.O(δi,j)>S'
   and S' can reach Send
6: Then {S=S';input(δi^in, WSk.O(δi,j)); Update(A);}
7: Else {Update(A);return false};
8: Next j
9: If S= Send return δi^in;
10: Next i
11: Else return false;
12: DSC_Any(SC,A)
13: B=Computer_PM(A);//computing reliable
matrix
14: S=S0;
15: Computer_NextS(S);
16: Computer_NextS(S) //allocating priority
17: If S=Send, return pro;
18: Else If ∃Oi∈O, which makes AWF(Oi,S)≠Φ
19: Then {K=Com_Suc(S, AWF(Oi,S));
20: WS=Ass_Pro(K, AWF(Oi,S));
21: S=S[WS•Oi>; //computing next state
22: Computer_NextS(S);
```

(2)Analyzing the composability of service by using Theorem 3, and cutting transfer matrix based on the feasibility of operation execution, which makes all firings be feasible in cut transfer matrix;

(3) Computing reliable matrix based on the cut transfer matrix, thus getting the success probability of each firing, and allocating the different priority to service.

The enforcement algorithm of dynamic service composition strategy is shown in Table I. The algorithm establishes the available service set WS that user can access based on the current service, then establishing the transfer matrix of *HDSC-net* model based on the actual available service. Finally, we will analyze the composability of service.

IV. EXPERIMENTS

This section shows the analysis process through a simplified Travel Services. The specific service composition process is: looking up information and choosing destination (O_1), train tickets reservations (O_2) responsible for handling customer's train tickets, airline tickets reservations (O_3) is to purchase a suitable destination flights, passage booking(O_4) is to order the appropriate passage in accordance with the requirements of consumer, tourism planning (O_5) is responsible for specific travel arrangement, car reservations (O_6) arranges the custom to arrived at the railway station or airport, hotel reservations (O_7) is arranging for the local living. Finally, the tourism service (O_8) is responsible for the customer's local tourism-related matters. The composition process can be represented by expression $O_1 > (O_2+O_3+O_4) > O_5 > (O_6||O_7) > O_8$, and user preference is $\{O_3, O_2, O_4\}$. The travel service has 6 available services, the basic attributes of operation is shown in Table II.

TABLE II.
THE BASIC ATTRIBUTES OF OPERATION

Service	O	*SU*	Service	O	*SU*
WS_1	$O_1>(O_2+O_3+O_4)>O_6$		WS_2	$O_1>(O_2+O_3)$	
	O_1	97.56%		O_1	97.56%
	O_2	98.83%		O_2	99.74%
	O_3	96.44%		O_3	98.44%
	O_4	97.56%	WS_4	$O_5>O_7$	
	O_6	99.74%		O_5	99.94%
WS_3	$O_5>O_7>O_8$			O_7	95.79%
	O_5	97.86%	WS_6	$(O_3+O_4)>O_6>O_8$	
	O_7	99.36%		O_3	96.36%
	O_8	98.96%		O_4	99.63%
WS_5	O_8			O_6	97.51%
	O_8	99.84%		O_8	95.79%

The *HDSC-net* model of Travel Service is shown in Figure 6. We can construct the transfer matrix of it by using algorithm 1 and analyze the composability of it,

thus getting the service is compositional and has four composition schemas: {WS_1,WS_1,WS_1,WS_1,WS_3, WS_1,WS_3, WS_3}, {WS_1,WS_1,WS_1,WS_1,WS_4,WS_1,WS_4,WS_5}, {WS_2,WS_2, WS_2,WS_6,WS_4,WS_6,WS_4,WS_6}, {WS_2, WS_2, WS_6, WS_6, WS_4, WS_6, WS_4, WS_6}. The available service set of O_1, O_5, O_3 are {WS_1,WS_2}, {WS_3,WS_4}, {WS_2,WS_6}. We can get the success probability of above operations are {93.317%, 89.235%}, {47.986%, 47.665%}, {44.709%, 44.708%} by computing the reliable matrix of Travel Service. Then allocating priority to each available service according to dynamic composition strategy, which are 0, 1, 0, 1, 0, 2, and mapping the priority of available service into the priority of transition, thus getting the reliability of Travel Service is 93.629%, which is schema 1.

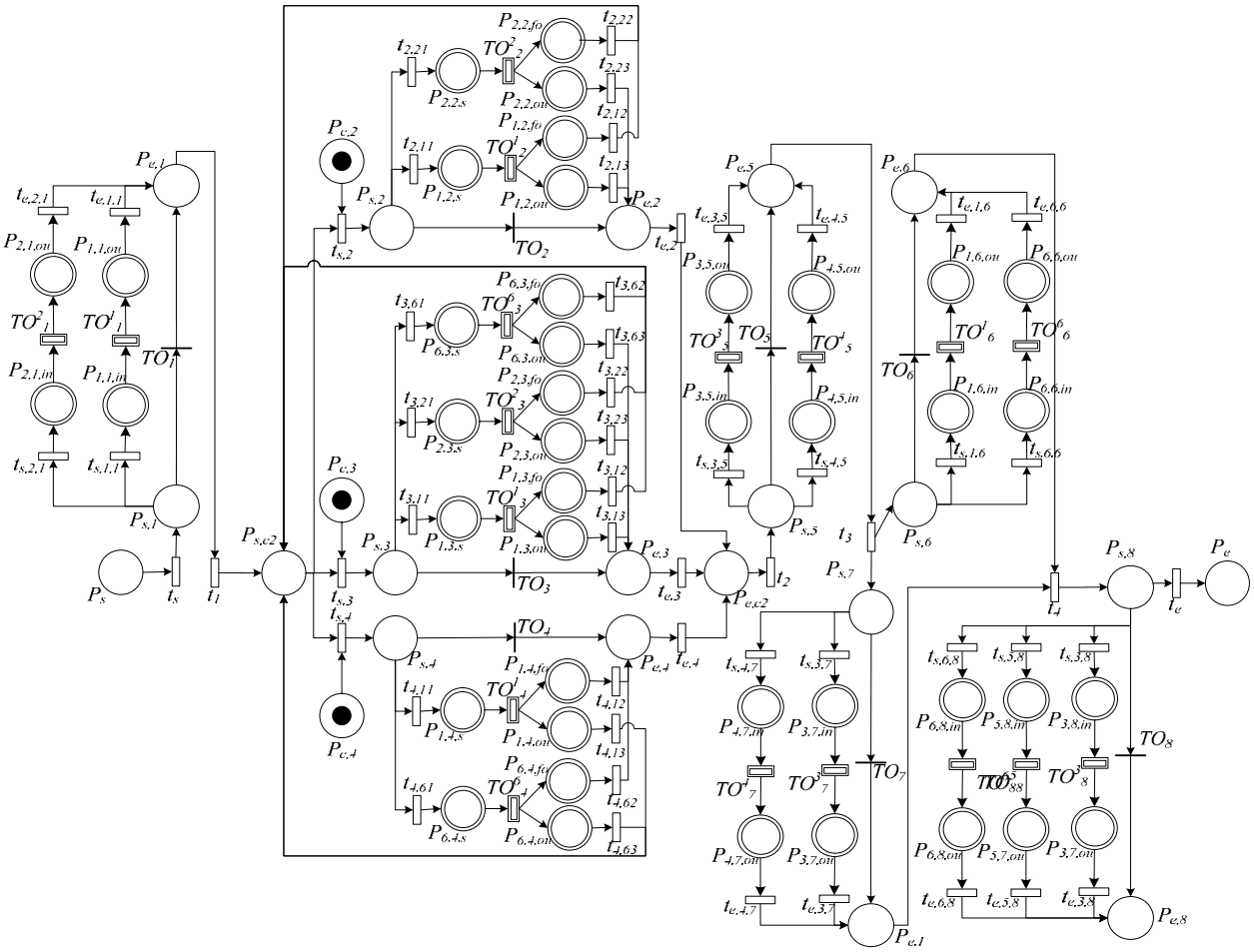

Figure 6. The *HDSC-net* Model of Travel Service

V. RELATEDWORKS

Several dynamic service composition systems have been proposed and implemented, which are given in [8-10]. In [8], the authors presented a method to selectively query services using the value of changed information, and the value of the change that revised information may potentially introduce to the composition. Reference [9] proposed architecture obtains the semantics of service, and dynamically composes the requested service based on the semantics of the service. Finite state machine was used in [10] to provide a precise and well defined semantic framework for establishing the key language attributes. However, most of the existing dynamic service composition systems require a user to request a service in a manner that may not be trivial and intuitive to the user, and they do not take reliability properties of composition behaviors into account.

Another works about QoS-aware service composition are [11-13]. Two strategies are given in [11] to select component Web services that are likely to successfully complete the execution of a given sequence of operations. The authors in [12] presented an autonomic service provisioning framework for establishing QoS assured end-to-end communication paths across administratively independent domains. A middleware platform which addresses the issue of selecting Web services for the purpose of their composition was given in [13], the method maximizes user satisfaction expressed as utility functions over QoS attributes. The approaches defined in the above have the requirement that there is a one-to-one correspondence between the required services and those that are available. If one or more of the required services cannot be located, existing systems will fail. And the consideration of user preference in Web services composition is ignored in these researches.

A similar work to ours is presented in [14]. In this work, the authors proposed a simple Web services

selection scheme based on users requirement of the various non-functional properties and interaction with the system. We presented a Petri net-based approach to analyzing the soundness and composability in BPEL process in [15], a set of translation rules is proposed to transform BPEL processes into Petri nets. Later [16] took the user constraints into account during composition and are expressed as a finite set of logical formulas with the Knowledge Interchange Format language. These approaches are differentiated by the fact that they deal or not with the behaviors of Web services. Some of works didn't analyze service composability.

VI. CONCLUSION

In this paper, we proposed an *HDSC-net* model to accurately characterize user preference based dynamic service composition. This approach is based on a formal model, Petri net, which allows to take into account user preference. We demonstrated how user preference of service composition can be expressed in this formalism. Transfer matrix is used to express the relationships between states; the dynamic service composition strategies and the corresponding enforcement method are also given. We can use this method to model and analyze user preference dynamic service composition, which has the merits of rich expressivity for user preference, while guarantees composability of service with high reliability.

This paper has made progress in modeling and analyzing user preference based dynamic service composition. However, we do not consider other non-functional properties. The reasoning mechanisms and tools are not covered. We will investigate these issues in future work.

ACKNOWLEDGMENT

The research described here was partially supported by the NSF of China under grants No. 60473055 and 60773094, Shanghai Shuguang Program under Grant No. 07SG32, Fund of Key Laboratory of Shanghai Science and Technology (09DZ2272600).

REFERENCES

[1] Q. Yu, X. Liu, A. Bouguettaya, and B. Medjahed. "Deploying and managing Web services: issues, solutions, and directions,".The International Journal on Very Large Data Bases, 2008, 17(3), pp. 537–572.

[2] I. Elgedawy, Z. Tari, and J. T. A. "Correctness-aware high level functional matching approaches for semantic Web services," *Transactions on the Web,* 2008, 2(2).

[3] M. Beek, A. Bucchiarone, and S. Gnesi. "Web service composition approaches: From industrial standards to formal methods," *In the Second International Conference on Internet and Web Applications and Services*, 2007, pp.15–20.

[4] C. E. Gerede, R. Hull, O. H. Ibarra, and J. Su, "Automated composition of e-services," *In Proceedings of the 2nd international conference on Service oriented computing*, 2004, pages 252–262.

[5] A.W. M. P, D. Marlon, O. Chun, and R. Anne., "Conformance checking of service behavior,".*Transactions on Internet Technology*, 2008, 8(3), pp 1–30.

[6] W. Tan, Y. Fan, and M. Zhou, "A petri net-based method for compatibility analysis and composition of web services in business process execution language," *IEEE Transactions on Automation Science and Engineering*, 2009, 6(1), pp. 94–106.

[7] T. MURATA, "Petri nets: properties, analysis and application," *Proceedings of the IEEE*, 1989, 77(4), pp. 540–581.

[8] J. Harney and P. Doshi, "Selective querying for adapting web service compositions using the value of changed information," *IEEE Transactions on Services Computing*, 2009, 1(3), pp. 169–185.

[9] K. Fujii and T. Suda, "Semantics-based dynamic service composition," *IEEE Journal on Selected Areas in Communications*, 2005, 23(12), pp. 2361–2372.

[10] R. Farahbod, U. Glasser, and M. Vajihollahi, "An abstract machine architecture for web service based business process management," *International Journal of Business Process Integration and Management*, 2006, 1(4), pp. 279–291.

[11] S. Y. Hwang, E. P. Lim, C. H. Lee, and C. H. Chen, "On composing a reliable composite web service: a study of dynamic web service selection," *Processing of the fifth IEEE International Conference on Web Service*, 2007, pp. 184 – 189.

[12] J. Xiao and R. Boutaba, "Qos-aware service composition and adaptation in autonomic communication". *IEEE Journal on Selected Areas in Communications*, 2005, 23(12), pp. 2344 – 2360.

[13] L. Zeng, B. Benatallah, and A. Ngu, "Qos-aware middleware for web services composition," *IEEE Transactions on Software Engineering*, 2004, 30(5), pp. 311–327.

[14] Y. Badr, A. Abraham, and F. Biennier, "Enhancing web service selection by user preferences of non-functional features," *In the 4th International Conference on Next Generation Web Services Practices*, 2008, pp. 60–65.

[15] G. Fan, H. Yu, and L. Chen, "Analyzing BPEL compositionality based on Petri nets," *In Proceedings of the 32nd Annual IEEE International Computer Software and Applications Conference*, 2008, pp. 373–376.

[16] Y. Gamha, N. Bennacer, and G. V. Naquet, "A framework for the semantic composition of web services handling user constraints," *In the Sixth IEEE International Conference on Web Services*, 2008, pp. 228–237.

Guisheng Fan. He was born in 1980, Ph. D. Assistant researcher in East China University of Science and Technology. His research interests include trust computing, distributed computing, service oriented computing and formal methods.

Huiqun Yu. He was born in 1967, professor, Ph. D. supervisor, IEEE senior member, ACM member, China Computer Federation senior member. His research interests include trust computing, software engineering, information security and formal methods.

Liqiong Chen. She was born in 1982, Ph. D. Lecturer. Her research interests include distributed computing, embedded systems and formal methods.

Caizhu Yu. She was born in 1985, Master. Her research interests include service oriented computing, distributed computing and formal methods.