Research of The Web Service Automatic Composition based on Generalized Stochastic Petri Nets

Xiaoan Bao, Jun Gao, Xiaoming Xie, Hui Lin and Na Zhang*

School of Information Science, Zhejiang Sci-Tech University
Hangzhou, Zhejiang, China

*{baoxiaoan@zstu.edu.cn, g_jun09@163.com, 14104437@qq.com, 970481544@qq.com, *zhangna@zstu.edu.cn}*

Abstract

Web services need to be orchestrated to achieve complex demand. However, existing approaches on Web service composition are mainly based on syntax match. It is very difficult to compose heterogeneous Web services with syntax in accord with function. Gradually, Petri net becomes a powerful system modeling and analysis tools research discrete event dynamic system. In order to solve this problem, we present a method to compose Web services automatically based on generalized stochastic Petri nets. Firstly, the generalized stochastic Petri nets (GSPN) and OWL-S composition process are used to model the automatic composition process, and then the Web service composition is realized. Finally, the feasibility of this method is demonstrated though the analyses of a case study with a booking service system. Compared with the existing methods of Web service composition, this method can describe Web service automatic composition more effectively.

Keywords: *Web service composition, Petri nets, OWL-S*

1. Introduction

In order to meet complex user requirements, Web services are normally composed and orchestrated to provide more full-fledged functions. That is the Web services are composed automatically. In order to describe the automatic Web service composition, many researchers proposed the OWL-S (OWL for Services). The semantic Web Service body language (OWL-S) is currently typical semantic Web services description method, its body Service Model can be used to describe the implementation process of Web services and logical execution sequence, and it is the basic of realizing automatic Web Service composition [1].

Recently, Petri nets have become a powerful system modeling and analysis tool in the discrete event dynamic system research [2]. Petri-net has the formal definition of asynchronous and concurrent, intuitive graphical representation and flexible analysis methods. Petri-net is a visual formal method for modeling business processes with rigid mathematical ground [2]. It is a model that is used to describe distributed systems and system structures, verify distributed systems, and also can simulate the operation of the system. Therefore, Petri net is particularly applicable to describe component-based multithreading, distributed software system structures, and constitutes a running relationship between the various components of a software system [3]. Petri-net, compared to most other Web service composition models, is better able to describe subsequent executions of software system, such as a Web server composition.

In this paper, we investigate how to dynamically construct reliable Web service composition based on the requirements of consumers. Petri net is used as an underlying formalism. The main contributions of this paper are shown below:

Firstly, we analysis the correctly modeling mechanism of OWL-S, and then we convert Web services composition structure to Petri-net with generalized stochastic Petri nets (GSPN), which is used to construct different components of Web service composition. Finally, we use Petri net theory to prove the effectiveness of the propose method at design phase, and analyze the accessibility of the composition.

The rest of the paper is organized as follows: in the second part we analyze the describing process of OWL-S to Web service combination's application logic and the basic concept of generalized stochastic Petri nets; The third part tells the process using generalized stochastic Petri nets to Web service composition into Petri net model; In the fourth part we discuss the validity of the model; In the fifth part we verify the accessibility of the model with an case study, and show the verification results. The sixth part introduces the related work of writing this paper while the seventh part is the conclusion.

2. Basic Concepts

There are considerable Web techniques exist, *e.g.*, UDDI, WSDL, SOAP. However, Web service based on the UDDI and WSDL technology can not make full use of semantic information. Thus, we need a description mechanism to describe the Web service semantic in order to realize the automatic composition. As XML lack of semantic, in order to solve the problem, W3C proposed languages based on XML (*e.g.*, RDF and OWL). Meanwhile, in order to generate machine understandable documents, the paper adopts the OWL language service ontology language OWL-S. Services are composed according to the different expressions between ontology knowledge base on a unity of the concept relation. Agent programs can discover them in a uniform way. Then, the caller and composition based on logical semantic Web service.

Web services automatic composition needs a good workflow model, or else cross-realm work can not be proceed. Petri net technology is such a kind of modeling method that not only can undertake structure modeling, but also can proceed extensive qualitative and quantitative analysis [4].

2.1 The Concept of OWL-S

OWL-S, the Ontology Web Language for Web services, includes three components: Service Profile, Service Grounding, and Service Model. It is based on Web services and semantic Web. OWL-S is the combination of Web service and semantic Web, it is mainly used to solve Web services description and discoveries and semantic representation of composition. OWL-S, one of the most significant Semantic web service ontology proposed to date, provides Web Service providers with a core ontological framework and guidelines for describing the properties and capabilities of their web Services in unambiguous, computer interpretable form [5].

2.2 The Concept of Generalized Stochastic Petri-net

With research unceasingly in-depth, Petri net is constantly enriched and perfected, its abstraction and describing ability also constantly develop toward transverse and fore-and-aft. Its longitudinal extension performance: from the basic conditions/events (C/E) nets and position change (P/T) nets, involves into predicate/transition nets and Color Petri nets and other senior nets. Its lateral extension performance: from no parameters Petri nets to time and stochastic Petri nets. Here expansion the Generalized Stochastic Petri Nets(GSPN), allowing him to model Web service.

A GSPN can be described with a P/T system ($S,T;F,W,M_0,\lambda$) [6] including the T were classified as time transition sets T_t and instantaneous changes set T_i two subsets. The implementation of the time transition takes some time, instantaneous changes the implementation of time can be neglected, namely, $T = T_t \cup T_i$, $T_t \cap T_i = \varnothing$, while in F the existing of banned arc is allowed. In GSPN, each implementation time of time transition obeys index distribution, namely, $\forall t \in T_t : F_t = 1 - e^{-\lambda_t x}$ .Among them, the actual parameters λ_t >0 is the average time transition implementation rate, says the average implementation time in unit time under implementing case, its countdown $\tau_i = 1/\lambda_i$ is called average implementing time of transition t_i , and λ is the set of λ_t .

3. Petri Net Model of Web Services Composition

We first convert the application logic of Web service, which is described by OWL-S to the Petri nets model to provide a formal representation method for composing the service structure and behavior. And the Petri nets should be defined to implement the operational semantics and input、execution conditions and results of service function of OWL-S, while the results include output and execution result [2].

Through the analysis of OWL-S combination process execution semantics、input/output ,we map it to Petri nets. And we add a library—control library at the ends of the input and output library of the atomic process to facilitate the combination of atomic Model. Below is the semantic Web service atomic process of Petri nets said diagram, among which the circular is the input and output database, the diamond is the control library, the rectangular in the middle is the change. The default of the weights of the database, control library and the change at the edge of the link is 1. This section of the conversion method based on literature [11, 12] relevant content.

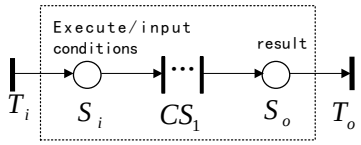


Figure 1. GSPN model of composition process

According to the existing research and based on the combination of OWL-S process, there are 7 kinds of Web service combination structure, here are described as follows:

1. Sequence executive structure: Atomic Processes execute according to their order, first Process 1 executes then Process 2 executes.

Sequence structure adds three control libraries: P1, P2 and P3, after Process1's successful execution, the output that has got identification and P2 make Process2 also has executive conditions, and then executes sequentially.

2. Choose perform a combination process.

This structure can have more than one branch to choose. We can choose one according to the user's option or the atomic process condition of a branch.

3. Choose execute a group member service

Each branch atomic process connects in sequence; we can also select multiple branches according to the condition.

4. Parallel execution structure

Each branch parallel operation, does not guarantee must synchronization.

5. Choose executive structure (1) (Iteration - while)

First determines whether the atomic process conforms to the change conditions and then execute.

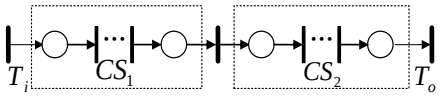
6. Choose executive structure (2) (Iteration - until)

First executes the build service, and then judges whether to continue or exit the loop.

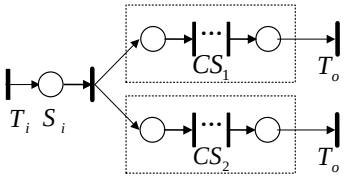
7. Unordered executive structure

A group of component services are executed according to random orders.

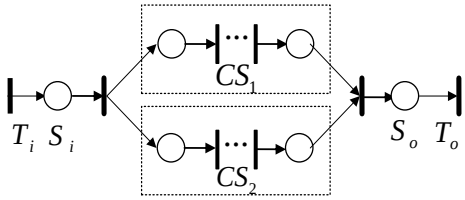
In Figure 2, we introduce seven kinds of the combination form of Petri net structure. The dotted box represents Web member services, diamond represents combination increased control libraries, rectangular represents the increased changes, the black spots in the diamond and the round represent identification token value.



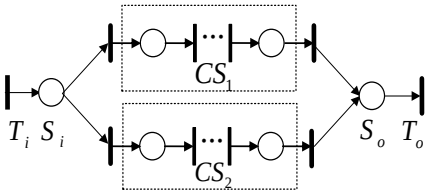
(a)Sequence executive structure



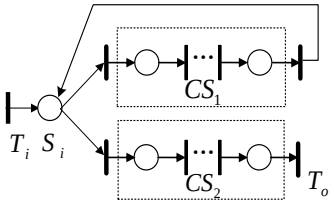
(b) Choose perform a combination process



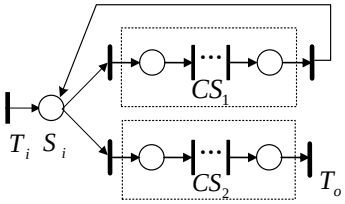
(c) Choose execute a group member service



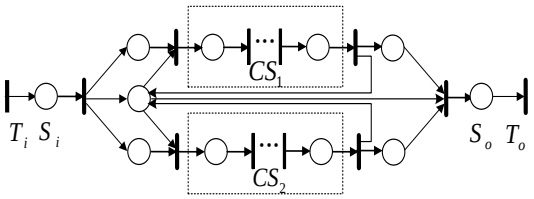
(d) Parallel execution structure



(e) Iteration-while



(f) Iteration-until



(g) Unordered executive structure

Figure 2. GSPN model of Web services composition structure

The mapping from the OWL-S composition Process to GSPN model, can take the manner of recursion replace from input to the output , according to the description of Process, when Web services composition structure of time transition represent composition Process, we use their corresponding GSPN model to replace it, until the time transition has been converted into atomic Process so far.

We can see from the figure above, all kinds of combination of complicated structure can form according to the seven kinds of basic structure, and the new composition also only have one import library and one output library, thus the complex Web service composition can be described by the model of generalized stochastic Petri nets. Below is the basic algorithm of Web service composition:

First, we should make sure of the input and output of the system, and then do the Web service composition.

Input: the composition of the Web service's activities element set; Output: the formed GSPN model and scripts of Web service composition.

1: Modeling the Web service elements graphically;

2: According to the model in figure 2, use the element set and its combination constructing Petri net model;

3: Compose the Web service's graph;

4: Based on the composite figure of activity, transform each activity and their combination relationship according to the GPSN model in figure 2;

5: Verify the model, modified if the model is incorrect;

6: After the validation, if will generate the running script according to figure 2 and the figure combination, then Web service combines end.

4. Automatic Composition of Web Services and the Accessibility analysis

See a Web service operate as a task, corresponding to the transition of Petri nets model; For the automatic composition of services, the prerequisite of a operation of a service namely the premise of running a task, the effect of a operation of a service namely the effect of running a task, corresponding to the place of Petri nets model.

When Web services are composed the activity diagrams of process description for reaching goals and tasks can be generated. Business process is processed by event-driven. When a task of a services composition was assigned to a service provider, the service provider had the target states and service must complete the target task. In the process of completing the task, it needs to undertake judgment, generates events to cause state transition. Thus, the development of the whole process is promoted. These processes are recorded in a document written with

XML, as an activity diagrams. Web services composition according to these activity diagrams, find services below the rank of which it contains, and then calls it. In systems of Web services composition, it is inefficient if let users manually searching, selecting and composing services, so we must use agents automatically complete Web service composition.

Services composed by using programming and process model languages, such as sequence, circulation, Parallel execution structure, the two choice execution structure, etc. Services composition were composed by a set of basic services and other services, automatic composition of service is the process that automatically generate a new process based on existing services. Basic services must provide process data model and business process model, so that the agent can automatically interact with the composition service, so some information encoding gives automatic composition and interaction is needed.

When Web services are composed, we need to verify whether it can execute and form a business process according to a certain program, namely whether the Web service combination can correctly represent the business process or not. That is achieve by using modeling tools to analysis the process state simulative in the design phase, so as to the process design whether there is some problem or not. There are two main kinds of description methods in this aspect: formal and informal method, like Petri nets and WF-Net. They all belong to the formalized description method.

We first design modeling description to the Web service combination process when using the formal method-Petri Nets to verify the Web service combination. Then the Petri nets' properties and simulative analysis method are used to calculate Web service combination's process operation condition and service combination properties which include: security, the reliability of the interface, and whether there are dead lock phenomenon, accessibility, etc.

This paper performs further research and discussion on the accessibility aspect. In the implementation process of the Web service combination, regardless of the data how to transfer through the process of design, when at the end of the implementation, the output data must be able to transfer to the location of the output parameters, that is if the implementation of the Web service combination complete and able produce results and output, the combination of services have accessibility, or it is inaccessible. Petri connection matrix and state equation is the effective method to analyze the reason of the service executive interrupt and the reduction of the Web service combination's reliability and usability as the result of the problem of inaccessible. The method based on incidence matrix and state equation use prediction ways to learn whether can transfer between the two states each other, no matter whether the two states are adjacent. In addition, it can be easily expressed for a complicated system.

Definition1: There must have a excitation sequence σ from M_0 to M_n, $\sigma = M_0 t_1 M_1 ... t_n M_n$, $M_x, x = 1,2,...,n-1$ expresses the records of all the state changes from M_0 to M_n in the transfer process, $t_x, x = 1,2,...,n-1$ expresses all vitalized the change records from M_0 to M_n in the transfer process.

The meaning of the accessibility of the Web service: one network system only have one input library and output library, When sent a Web service request to the network system, the Initial state M_0 is $M_0 = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$, it means Only the import library has a change, but others have none, when the combination demand of the network system are all processed, the change was just reach to the output library of the system, and other place can not have token, otherwise the combination demand does not complete, or the execution result has not been transmitted, so the end state is $M_n = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]$. Therefore, for

the whole process of a combination demand of a network system, there must have a sequence that can convert start state $M_0 = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$ to the end state $M_n = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]$, the system must meet this condition to be a accessible system and the demand complete propitiously. Otherwise the combined demand's model designing will have the problem of inaccessible.

Definition2: incidence matrix and state equation analysis method use input matrix D^- and output matrix D^+ to express Petri net model, each matrix has m rows and n lines, and respectively express the change and the place of the Petri nets. Specific said as follows:

D^- represents the place from the input to the change: $D^-[i, j] = \#(p_i, I(t_j))$, $I(t_j)$ is 0 or 1, if p_i has process input to t_j, $I(t_j)$ equals to 1, on the contrary it equals to 0;

D^+ represents the place from the change to the output: $D^+[i, j] = \#(p_i, O(t_j))$, $O(t_j)$ equals to 0 or 1, if t_j has process input to p_i, $O(t_j)$ equals to 1, on the contrary it equals to 0.

We use D to express the Petri nets model's matrix, $D = D^+ - D^-$. Then judge if M_0 (initial state) can reach M_n (end state), and M_0, M_n are all one line n column matrix: $M_0 = [1 \quad 0 \quad ... \quad 0]_{1 \times n}, M_n = [0 \quad 0 \quad ... \quad 1]_{1 \times n}$.

Finally solving equation: $M_n = M_0 + X \bullet D$, if X has the solution, it means M_0 can reach M_n, the combination of services have accessibility; Otherwise we can't find the end state, the combination of services do not have accessibility.

5. A Case Study

Take a simple booking service system for example, there are two basic Web service interactions in the booking service executing process. One is the self-help booking service and the other one is the artificial booking service. The system structure is shown in figure 3.

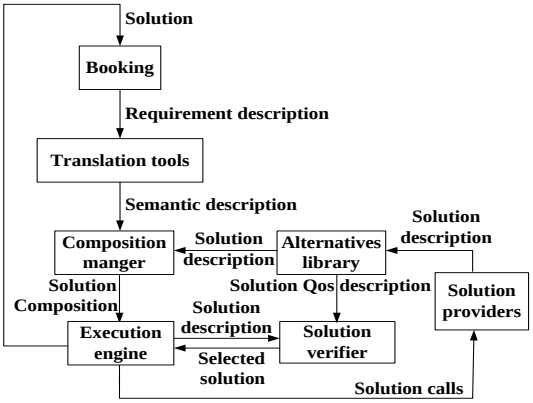


Figure 3. Booking service system structure

The next task is to use the Petri nets model, which was established in the third quarter, to describe the booking service system. The generalized stochastic Petri nets model of this booking system is shown in figure 4, specific as follows.

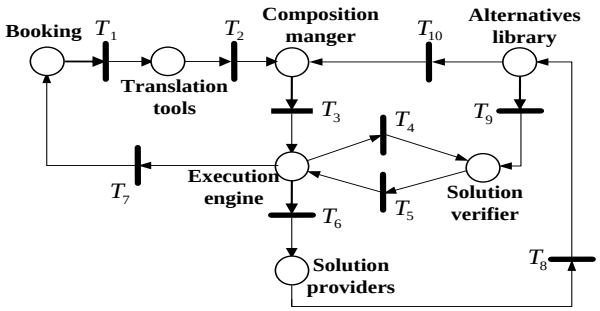


Figure 4. GSPN model of booking service system

It can be seen from this example that the proposed automatic Web service composition model based on Petri nets can intuitively describe the Web service composition. A single Web service appears in the form of Web services composition after being composed, and various properties of Web services composition can be analyzed by using the formal analysis ability of Petri nets. Next, we should analyze and verify the accessibility of the system.

First of all, construct input matrix D^- and output matrix D^+ which correlation matrix and state equation method need according to the system characteristics respectively. According to the system's structure, take n = 7, then substitute the formula, we can know: input matrix:

$$D^- = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & -1 \\ -1 & -1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

output matrix:

$$D^+ = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

then, we can know:

$$D = D^+ - D^- = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

If make initial state as $M_0 = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$, it can be concluded that if the Web service combination system has the accessibility, which can perform to the end, the corresponding end state is $M_n = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]$. Take the above data into the equation $M_n = M_0 + X \bullet D$ and can obtain the result $X = [0 \quad 1 \quad 0 \quad 0 \quad 2]$. Thus it can be seen X have the solution result, so in this paper the model can make the initial state to end state, this system has the accessibility, and can perform to the end correctly. It can also be seen indirectly, the system does not turn up to the dead lock state.

6. Related Work

This paper described our work on how to apply formalized modeling method in specifying Web services and their composition, and how to use formal analysis methods to analyze the correctness and the rationality of the Web service combination model are the main contents. The related works of this study are introduced as follows below:

In recent years, many composition methods are brought forward for Web service composition. There are many comparatively typical methods, including composition method based on workflow; article [7] use theorem proving methods to realize automatic Web service composition; article [8] adopts an extension Golog language automatically perform Web service composition; article [9] put forward a kind of hierarchical task network method, through thoughts of decomposing task to compose Web services; article [10] put forward a semantic service composition method based on fuzzy colored Petri net which combining the features of Petri Net and fuzzy Petri net using the method of the algebra definition . All above Web service composition methods are data flow models of service composition. However, they did not give flow of control structures of services composition, such as sequential, parallel and circulation, choice, *etc*. Under the condition of having the same data flow model, the performance of different service compositions is different from each other, due to the different ways of composition of the control structures. Even when control structures are given, such as the article [1] conduct semantic description for Web services and Web services composition based on OWL_S, but lack of intuitivism and dynamic. The behavior of a Web service indicates how to interact with the Web service. Semantics describe the meaning of services, and ontology is usually used for the semantic description. This paper proposes the method of modeling of Web service composition, based on Petri nets and the composition process of OWL-S, intuitive describes the composition model of Web service composition.

7. Conclusions

This paper takes Petri net as description models of automatic Web services composition and builds stochastic Petri nets models by using process of composition of OWL-S, and then realizes automatic Web service composition based on this foundation.

As description tools of automatic Web service composition, Petri net is convenient and intuitive. But when there are too many Web services, the interaction between services are too complex, it may cause rapid growth of model state space, thus causing the problem of state space explosion. So, according to this problem, in the future research we will study the simplified problem of Petri net models further.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China under Grant No.61202050,No.61170325; Zhejiang Provincial Natural Science Foundation of China under

Grant No.LY12F02041, 了 Y13F020041; Zhejiang Qianjiang Talent Project under Grant No.2012R10009, 2013R10SA480001, and the New-shoot Talents Program of Zhejiang province under Grant No.2012R406071.

References

[1] Z. Ma and K. Ben, "A Semantic Service Composition Method Based on Fuzzy Colored Petri Net", Advances in Computer Science and Education Applications - International Conference, CSE 2011, Proceedings, **(2011)**, pp. 396-404.
[2] B. Yu and D. Li, "Verifying Web Services of OWL-S with Petri Net", IEEE Computer Society, 445 Hoes Lane - P.O.Box 1331, Piscataway, NJ 08855-1331, United States, **(2010)**, pp. 891-896.
[3] C. Yan, C. Jiang and Q. Li, "Web service composition and analysis based on Petri nets", Computer science, vol. 34, **(2007)**, pp. 100-103.
[4] M. A. Marsan, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis, "Modeling with Generalized Stochastic Petri Nets, John Wiley & Sons, **(1995)**.
[5] Y. Xia, G. Dai, F. Tang and Q. Zhu, "A stochastic-Petri-net-based model for ontology-based service compositions", IEEE Computer Society, 445 Hoes Lane - P.O.Box 1331, Piscataway, NJ 08855-1331, United States, **(2011)**, pp. 187-190.
[6] C. Lin, "Stochastic Petri nets and system performance evaluation", Qinghua University Press, Beijing, **(2005)**.
[7] J. H. Rao, P. Kungas and M. Matskin, "Logic-based Web Services Composition: from Service Description to Process Model", Proceeding of 2004 International Conference on Web Services. San Diego: IEEE, **(2004)**, pp. 446-453.
[8] S. McIlraith and T. C. Son, "Adapting Golog for Programming the Semantic Web", Proceedings of Fifth International Symposium on Logical Formalization of Commonsense Reasoning. New York: New York University, **(2001)**, pp. 195-202.
[9] E. Sirin, B. Parsia, D. Wu, J. A. Hendler and D. S. Nau, "HTN Planning for Web Service Composition Using SHOP2", Journal of Web Semantics, **(2004)**, pp. 360- 396.
[10] Z. Ma and K. Ben, "A Semantic Service Composition Method Based on Fuzzy Colored Petri Net", Springer Verlag, Tiergartenstrasse, vol. 17, Heidelberg, D-69121, Germany, **(2011)**, pp. 396-404.
[11] A. Brogi and S. Corfini, "Behaviour-aware diseovery of Web service compositions", ITJ3830, vol. 4, **(2007)**, pp. 1-25.
[12] Z. Lei, "Verification Technology of Web Service Composition based On Petri Net", TP301, vol. 4, **(2011)**, pp. 25-30.

Authors

Bao Xiaoan (1973 -)

Male, master's degree, professor. Working unit: School of Information Science, Zhejiang Sci-Tech University. Main research direction: the adaptive software, software testing and intelligent information processing;

Gao Jun (1987 -)

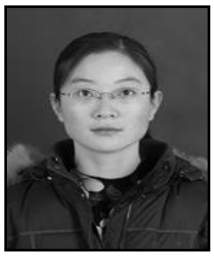
Male, master graduate student. Working unit: School of Information Science, Zhejiang Sci-Tech University. Main research direction: software engineering and intelligent information processing;

Xie Xiaoming (1987 -)

Male, master graduate student. Working unit: School of Information Science, Zhejiang Sci-Tech University. Main research direction: software engineering and software test technology;

Lin Hui (1989 -)

Male, master graduate student. Working unit: School of Information Science, Zhejiang Sci-Tech University. Main research direction: design patterns of Software;



Zhang Na (1977 -)

Female, master's degree, associate professor. Working unit: School of Information Science, Zhejiang Sci-Tech University. Main research direction: software engineering and software test technology.