Exception Detection for Web Service Composition Using Improved Bayesian Network

Yao Chen[1], Shi Ying[1], Liqiang Zhang[1], Juebo Wu[2]
[1]State Key Lab of Software Engineering
Computer School, Wuhan University
Wuhan 430072, P. R. China
[2]Department of Geography
National University of Singapore
117570, Singapore,
chenyao198516@yahoo.cn, {yingshi, zhanglq}@whu.edu.cn, wujuebo@gmail.com

Journal of Digital Information Management

ABSTRACT: *New application systems generated by composition of web services dynamically have become a development trend in network environment.However, since a variety of external services are invoked with different quality of service during processing,the problem of how to keep the execution stable must be addressed in order to improve the reliability and availability of the combination of services.In this paper, an approach of exception detection is presented for web service composition by improved Bayesian network. Firstly, the topological structure of Bayesian network is established, where the causal relationships between underlying web services are mapped to Bayesian network in service combination process. The conventional Bayesian network algorithm is improved to satisfy the demand of better conversion from nodes in web service composition to nodes in Bayesian network.Secondly, the parameter setting in Bayesian network is explored for determining the prior probability of service nodes and the conditional probability of service output node. Thirdly, the algorithm of the improved exception detection model is introduced, with the aim of locating the web services that cause exceptions in the execution of composite services.Lastly, a case study is given and the results analysis is also conducted. The results show that the presented approach not only considers the uncertainty during exception detection, but also can identify the exception web services in the process.*

Categories and Subject Descriptors:
H.3.5 Online Information Services]: Web-based services; **C.2.3 Network Operations**

General Terms:
Network algorithms, Bayesian network, Network Algorithms

Keywords: Web Service Composition, Exception Handling, Bayesian Network, Web Service

Received: 10 October 2012, Revised 2 December 2012, Accepted 8 December 2012

1. Introduction

Web service composition is operated in a dynamic network environment changing over time, where normal operations may be affected by uncertain factors, such as network change, web service failure and attacked by other parties [2]. The original combination of web service may also be failed because of its autonomy, service replacement and service update [18]. In addition, due to operation process combined by dynamic discovery and binding external services, it's very difficult to foresee all possible changes in the stage of the design and deployment under such high dynamic circumstance. Web service composition is facing a great number of challenges like flexibility and continuous equation form. [16] [4]. In the process of web service execution, different external services are called to meet dynamic user demand with different QoS. It puts forward high requirement for exception handling. If the system fully takes artificial way to adapt to the dynamic changes of the environment, the system maintenance costs will be raised sharply [9]. Therefore, the system has to monitor the web service combination in real-time, and find its operation error and conflict timely. At the same time, it should identify the exception component and service, so the system can repair such failures in order to improve the reliability and usability of the services combination [12] [20].

Much work has been done in an effortto the fault diagnosis of web service combination.A strategy of modeling and analyzing fault tolerant service was proposed for different components of service composition established by Petri nets [6]. Artificial intelligent (AI), such as hybrid case-based reasoning (CBR) and artificial neural network (ANN), was adopted for machine fault diagnosis in web service system [8]. [17] proposed a model using meta-process for modular and adaptable exception handling of web service composition, which based on event, activity state and case data of base process. [1] presented an architectural framework for monitoring and analysis of QoS based on web service technology, allowing QoS deficiencies to be detected and considered as an indicator of the health degradation of the monitored web services.However, these studies were based on the diagnosis of consistency, which cannot undertake processing the uncertainty existing in the process of input, service execution and output.

Bayesian network is a good means to deal with the uncertainty existing in the execution of web service composition [15]. By using it, we present an approach of exception detection for web service composition, for monitoring and locating the service component.The mapping from nodes of web service composition to nodes in Bayesian network is exploded. And the exception detection algorithm is given based on the improved Bayesian network. The rest of the paper is organized as follows.In the next section, we discuss the exception handling model with multiple web service registries. Section 3 gives the details of how to map Bayesian network to exception handling for web service composition. In section 4, we make an improvement for the conventional Bayesian network. Section 5 presents the parameter setting for the mapping and the detail algorithm of exception detection is given in section 6. A case study is carried out in section 7 and the final section draws a conclusion.

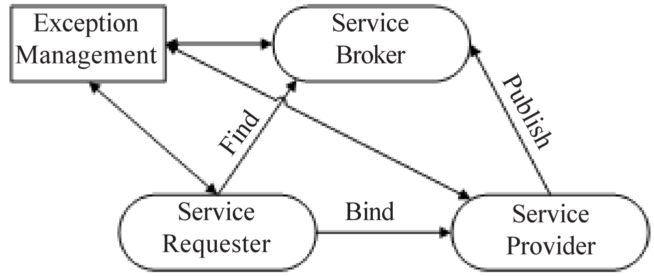

Figure 1. Web service model with the ability of exception handling

2. Exception handling model of web service

Figure 1 shows the common web service model with the ability of exception handling, which adds an exception management module to the existing web service model.

In this model, there are only a registration center and an exception management module, and each node is interacted through them. As the network scale increasing, the registration center will become large and may influence query efficiency.

As the above considerations, we put forward a web service model with multiple registration centers and exception management modules and keep the basic structure unchanged. A total registration center is set up to classify the services for sub-registration centers, and each registration center connects to its corresponding exception management module, as shown in figure 2. The service provider firstly accesses sub-registration center. If it can't meet the provider's demand, then it visits the total registration center. Total registration center locates corresponding sub-registration center according to service descriptions. For service requester, it firstly visits sub-registration center. If there are no proper web services in such sub-registration center, then it changes to access

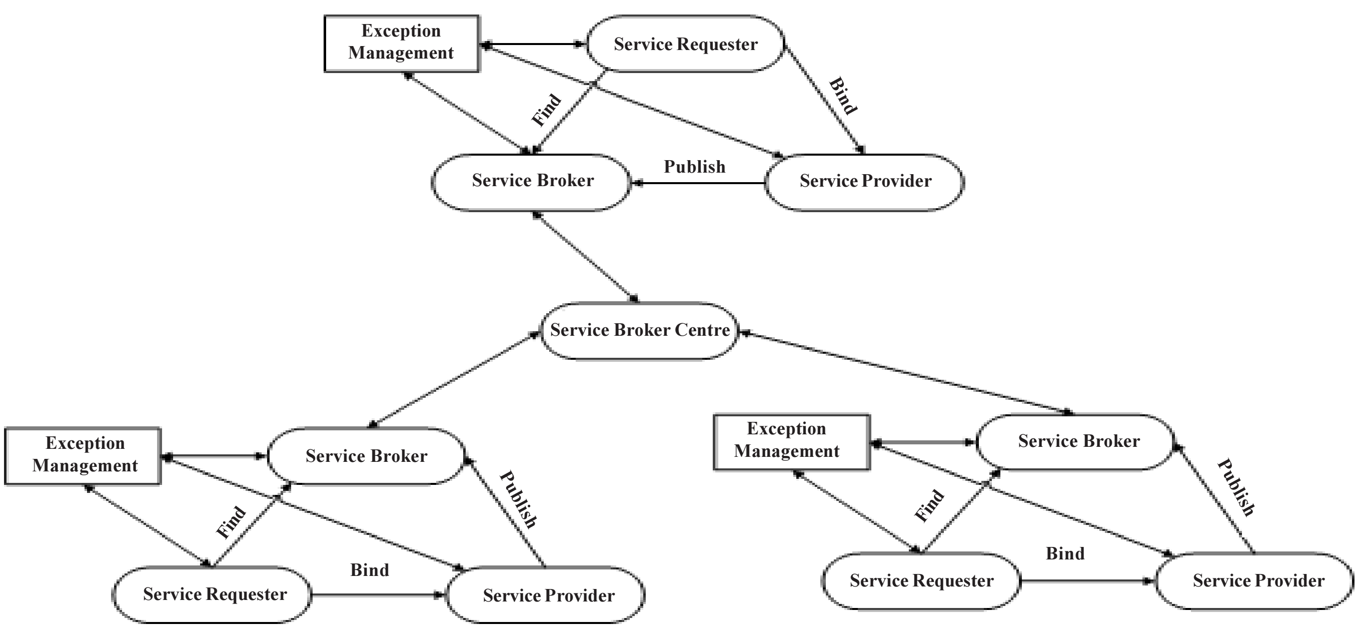

Figure 2. Improved web service model

the total registration center. Exception monitoring modules are responsible for monitoring every sub-registration centers and communicate through the total registration center.

3. Mapping web service composition to Bayesian network for exception handling

In exception detection, it needs to map the implicit causal relation from web service composition to Bayesian network.In this process, the output reason includes all inputs and the executions of services. The correctness of the output of a component service node depends on the correctness of all its inputs and the correctness of services execution. The output can be functional output, and also can be QoS index such as response time. Therefore, services execution cab be extracted as a single root node, and the output is expressed as a single variable node. Figure 3 shows how to convert a service executive node to random variable node in Bayesian network in web service composition.

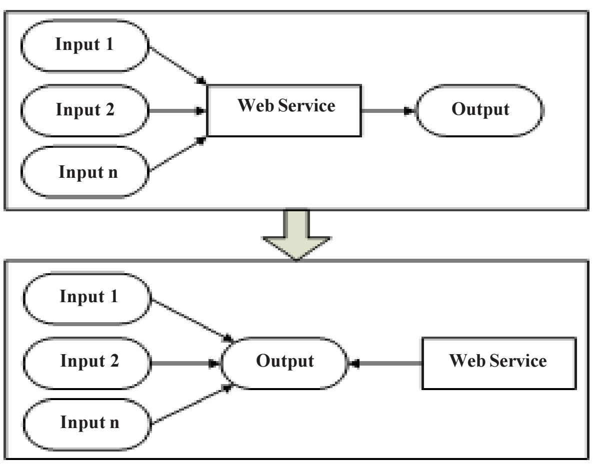

Figure 3. Mapping from nodes in web service composition to nodes in Bayesian network

The state of a service output only depends on its direct input and service executive condition, but has nothing to do with the indirectly input and service executive condition. Therefore, it meets the conditions for independence in this conversion. After that, each service executive node in web services execution becomes a root node in Bayesian network. Each service node doesn't have parent node and all service nodes are independent random variables. Whether the service function itself can be carried out correctly only depends on the quality of its execution, not affected by the influence of other services. This conversion is consistent for the web service operation in reality.

4. Improved Bayesian network

Based on SMOOTH algorithm [19] and E-IPFP algorithm [13], we modify conventional Bayesian network in order to improve the processing ability when the composition of diverse network is not consistent.

4.1 Basic Bayesian network

Given a Bayesian network G and there are n random variable nodes in G as $X = (x_1, x_2, ... x_n)$. Given the father node of random variable x_i is π_i, so network structure is $G_S = \{(x_i, \pi_i)\}$. Assume $G_P = \{p(x_i | \pi_i)\}$ is condition probability table of random variable node, then Bayesian network is described as $G = (G_S, G_P)$.

According to the chain rule, Bayesian network G decides a joint probability distribution as:

$$P(x) = \prod_{i=1}^{n} p(x_i | \pi_i) \qquad (1)$$

The joint probability distribution in G describes the given knowledge base, and Bayesian network expresses the knowledge model of a problem domain. The source of new probability knowledge is different, and it may lead to inconsistency of probability knowledge, and even inconsistency of network structure.Thus, the new probability knowledge needs to be synthesized to the original Bayesian network, in order to make the updated Bayesian network meet new probability knowledge.

Assume $R = \{R(Y^1), R(Y^2), ..., R(Y^m)\}$ is probability constraints set and $G' = (G_S, G'_P)$ is the new Bayesian network,then the joint probability distribution $P'(x)$ decided by G' meets the given constraint set and the distance between $P'(x)$ and $P(x)$ is the minimal value.

K - L distance reflects the difference of two joint probability distributions, defined as:

$$I(P \| Q) = \begin{cases} \sum_{P(x)>0} P(x) \log \dfrac{P(x)}{Q(x)} & if \;\; P \le Q \\ +\infty, & \text{else} \end{cases} \qquad (2)$$

where $P \le Q$ stands for $\{x | P(x) > 0\} \subseteq \{x' | Q(x') > 0\}$.

The process of locating exception in web service composition is to find out the Bayesian network which meets the constraint set.

4.2 Improvement of Bayesian network by SMOOTH

Given an initial joint probability distribution as $Q_0(x)$ and $R = \{R(Y^1), R(Y^2), ..., R(Y^m)\}$ is constraint set, IPFP is defined as the below iteration process.

$$Q_k(x) = \begin{cases} 0, \; if \, Q_{k-1}(y^i) = 0 \\ Q_{k-1}(x) \times \dfrac{R(Y^j)}{Q_{k-1}(y^j)} \; else \end{cases} \qquad (3)$$

where $j = ((k-1) \bmod m) + 1$

Every iterative step of IPFP uses a constraint condition in constraint set. By Equation 3, the initial value of next iterative step can be calculated to meet the joint probability distribution of current constraint. Through the iteration of using constraint set, it can realize the algorithm convergence.

The consistent convergence under constraint set was proved [5] [3]. $Q^*(x)$ is the convergence results computed by IPFP, which meets all the constraint conditions and joint probability distribution of constraint set. $Q^*(x)$ and $Q_0(x)$ has the minimum K- I distance. When the constraint set is inconsistent, IPFP algorithm will not converge to a single fixed point, but oscillate in a few joint probability distributions.

[14] improved IPFP by Bayesian network, which took the Bayesian network structure as a new constraint conditions and the independent relationship in random variable.In the process of IPFP, it also used the constraint condition in constraint set and the constraint condition of network structure (R_S) through the iteration for convergence.

After each iteration of IPFP, the conditional probability is computed by the current probability distribution $Q_k(x)$ and the conditional probabilities of all nodes in Bayesian network, defined as:

$$p(x_i \mid \pi_i) = \frac{Q_k(x_i \mid \pi_i)}{Q_k(\pi_i)} \qquad (4)$$

where $Q_k(x_i \mid \pi_i)$ and $Q_k(\pi_i)$ can be obtained by the marginal probability distribution of $Q_k(x)$. The Bayesian network is updated by Equation 4 and a new joint probability distribution is obtained by Equation 1, which is the initial value for the joint probability distribution in next round.

The algorithm SMOOTH was presented to study the knowledge synthesis problem when the constraint set is not consistent [19]. In this algorithm, there are two key steps. One is to operate standard IPFP until when it reaches convergence or goes into oscillation. The other is to modify the current constraint condition by Equation 5 after going into oscillation and carry out IPFP calculation. When SMOOTH algorithm is consistent in constraint set, it degrades to the standard IPFP.Through the latter step, the SMOOTH algorithm realizes two-way modification for joint probability distribution and constraint condition. The experimental results showed that this algorithm has the stable convergence, and can adjust algorithm convergence speed according to the smoothing factor, defined as:

$$R'(Y^i) = (1-\alpha)\, R(Y^j) + \alpha\, Q_k(Y^j) \qquad (5)$$

The revised constraint conditions include the original constraint conditions and the current joint probability distribution. In order to improve Bayesian network, we introduce SMOOTH into E-IPFP. Before computing IPFP by Equation 3, Equation 5 is used to modify the current constraint conditions. And then, compute the IPFP by using the modified constraint conditions.

The steps of the improved algorithm are described as below.

(1) Compute initial joint probability distribution $Q_0(x) = \prod_{i=1}^{n} p(x_i \mid \pi_i)$ for Bayesian network according to the chain rule.

(2) Modify the current constraint by Equation 5 through the use of constraint condition in constraint set R one by one.

(3) Compute IPFP by Equation 3.

(4) Update the condition probability table of Bayesian network nodes by Equation 4 after each iteration calculation of constraint condition, and use the new joint probability distribution as the iterative initial value for next round.

(5) Repeat step 2 and step 3 to convergence.

5. Parameter setting in Bayesian network

In the exception handling model with Bayesian network, parameter setting should be carried out to find out the location where the exception happens. This requires the marginal distribution of service node and the conditional distribution of service output node are determined respectively.

By means of the marginal distribution of service node, the probability that an error occurs in a service node can be determined. The prior probability of service node determines the probability that an error occurs in a service node by historical data. [7] proposed a model of credit measurement for web service. In this model, it introduced forgetting factor and wave factor based on statistical theory. At the same time, it considers the influence of QoS of service credit, which is a kind of reference basis for credit measurement. The credit value is limited in [0, 1] and it’s a continuous variable. The greater the credit, the better the possibility of correct operation will be.

The condition distribution of service output, as conditional probability table (CPT), is used to describe the relationships among service input, service execution and service output.Suppose there are k inputs for web service V_i, the CPT of V_i means the probability distribution of $input_1$, $input_2$,..., $input_k$ and service operation of $output_i$ status. As each node has two states with normal and error, the CPT of V_i contains the probability list of 2^{k+1} items described as $P\ (Output_i \mid input_1,\ input_2,...,\ input_k,\ operation)$. Chance node and deterministic node is set by CPT and the exception detection model can support the uncertain relationships among service input, service output and service operation.

Service providers understand the services they provide very well, and have the historical operation data of services. Therefore, CPT can be made by service providers with subjective assessment and through the parameter learning of historical data. For example, service providers can determine CPT through the maximum likelihood Estimation or Bayesian Estimation. In addition, with the increase of service operation data, CPT can be timely updated by using sequential learning and batch learning [10].

As discussed above, figure 4 shows an example of restaurant reservation of web service composition to Bayesian network in real time. After conversion, each service is transformed into a service node in Bayesian network, in which generated web service node is an independent random variables. Each service become independent and has no impact on the other nodes. The operation of each service itself is only related to its quality, and has nothing to do with other services. This is consistent with web service combination in the running status.

6. Algorithm of exception detection for web service composition

By using the inference ability of Bayesian network, the observed conditions of output nodes are used to infer the observed component services. The output state of service is judged by the method [11]. Assume that $\{E1, E2, ... ,., Em\}$ (Ei is the state of the observed state of output node) is the evidence set of the corresponding output nodes, the posterior probability $Bel(S)$ of the service node S can be computed through the construction of Bayesian network.Then find out which ones have the maximum service nodes and do inspection for them to judge whether its operation is normal or not. The results will join the evidence of the next judgment. Through Bayesian inference, the system can locate and find out the most possible fault component service for inspection. It is not testing for each service, so the improved approach can save the time and cost for service inspection. The algorithm of exception detection for web service composition based on Bayesian network is described in Algorithm 1.

Algorithm 1 Exception detection for web service composition by improved Bayesian network

```
// T is the threshold of expected fault nodes, ES is the evidence set
ExceptionDetect (T, ES) {
   //EDNumber is the number of verified exception nodes
   while (EDNumber < (T + 1)) {
      Compute the posterior probability Bel (S) of each service
      node by reasoning;
      Find out the node S_k with the maximum posteriori proba-
bility from all the nods;
      Examine the state of S_k;
      if (the state of S_k == fault){
            EDSet = EDSet.add (S_k);
      }
      ES = ES.add (S_k);
   }
   return EDSet;
}
```

7. Case study

In this section, we carry out a case study for web service composition in order to demonstrate the feasibility of the presented approach. The goal is to find out the exception services that cannot be executed properly. The experimental environment is Windows XP OS with 2.53 Ghz CPU, 2 GB of memory, and Visual Studio 2010.

We use fault diagnosis percentage as the evaluation indicator to verify theeffectiveness for service exception detection, defined as:

$$FDP = \frac{FaultDiagnosed}{FaultTotal}$$

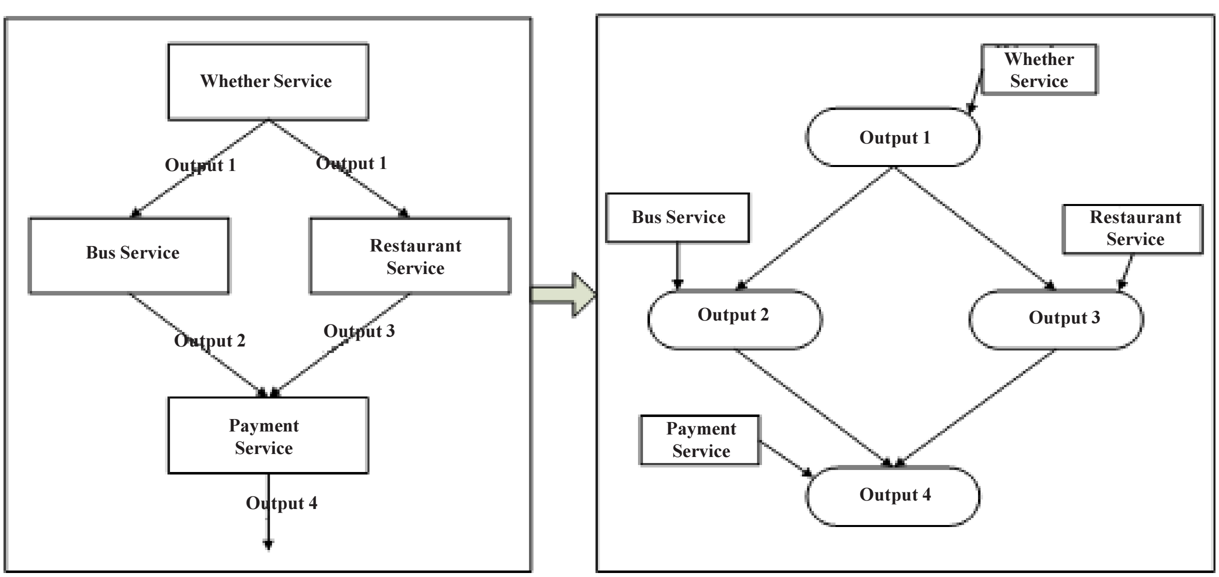

Figure 4. Example of restaurant reservation of web service composition to Bayesian network

where *FaultDiagnosed* is the number of problem service nodes after detection and *FaultTotal* is the total number of problem service nodes in the process of web service composition. The higher the FDP is the more problem nodes will be.

We adopt SMILE as the fully Bayesian inference engine, which was developed by the Decision Systems Laboratory in University of Pittsburgh.

In the simulation process, the prior probabilities of the nodes were randomly generated according to uniform distribution. Assume a random number M within [0, 1], the node state is normal if M is smaller than the prior probability of the service node. Conversely, the node state is error. For each group of networks, we carry out the same implementation for 50 times, so the FDP is an average value.

Figure 5 shows the average FDP of web service composition with different scale of service nodes, that is, 15,30,45,60. It can be seen from the result that the FDP is increasing with the growth of service nodes and has a good capability when there are increasing fault nodes with bigger scale of network.

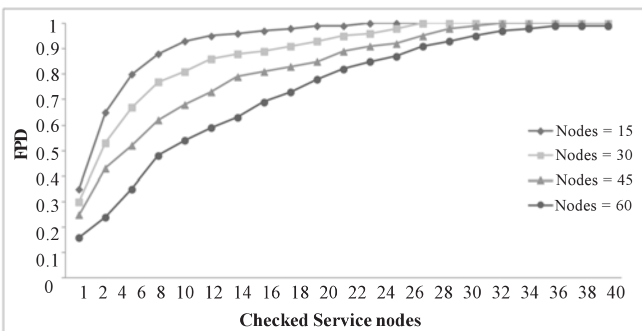

Figure 5. FDP of web service composition with different scale of service nodes

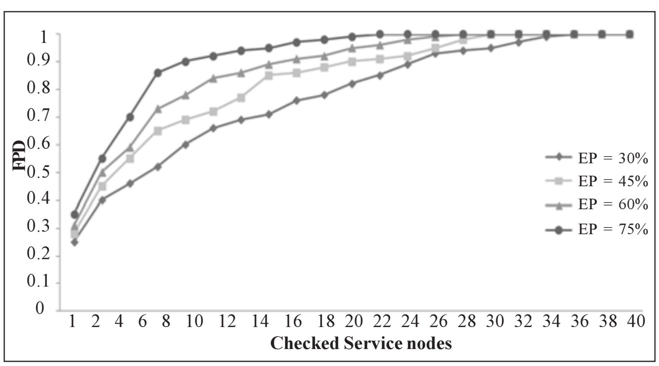

Figure 6. FDP of web service composition with different evidence proportion

Figure 6 show the average FDP of web service composition with different evidence proportion (EP), that is, 30%, 45%, 60% and 75%. We set the number of web service nodes as 40. It can be seen from the result that the higher FDP is, the bigger the evidence proportion will be.

Figure 7 shows the average FDP of web service composition with different tasks in one service node. The numbers of tasks in each web service node are 20, 30, 40, and 50 respectively. It can be seen from the result that the smaller the network is the number of exception service will be. The reason is that the prior probability of each service node is with the same interval.

It can be seen from the results that the presented approach is feasible and effective in different kinds of conditions with diverse task number.

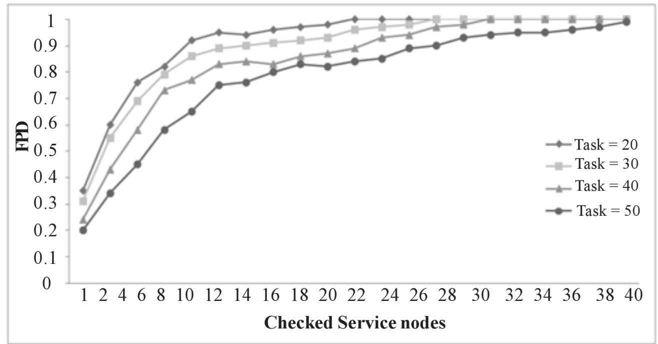

Figure 7. FDP of web service composition with different tasks in one service node

8. Conclusions

The application system constructed by web service combination has become a hot research topic at present. In this context, the network is changing over time and the web service chains are easy to lead to exception. In this paper, we presented a model of exception detection for web service combination in business process based on improved Bayesian network. The mapping from web service composition to Bayesian network was given, which reflected the implicit causal relation between web service exception and Bayesian network. By means of failure probability and historical operation data, the system can compute the prior probability of service nodes and the conditional probability of output nodes. On this basis, we introduced the exception detection algorithm for web service composition by Bayesian inference. A case study was performed and the results show that the presented approach can monitor the execution of web services by considering the uncertainty. Meanwhile, it can also identify the exception nodes rapidly and efficiently.

The future work will focus on studying the system performance of web service composition with dynamic user needs. Moreover, we will further investigate the combination of web service reconstruction and recovery technology in order to improve the efficiency of the composite service execution.

References

[1] Ben Halima, R., Guennoun, M. K., Drira, K., Jmaiel, M. (2008). Providing predictive self-healing for web services: a qos monitoring and analysis-based approach. *Journal of Information Assurance and Security*, 3, 175–184.

[2] Chan, K., Bishop, J., Steyn, J., Baresi, L., Guinea, S. (2009). A fault taxonomy for web service composition, *In*: Service-Oriented Computing-ICSOC 2007 Workshops. p. 363–375.

[3] Csisz'ar, I. (1967). Information-type measures of difference of probability distributions and indirect observations. *Studia Sci. Math. Hungar.* 2, 299–318.

[4] Dong, Y., Lei, L., Llinares, M. B. (2011). A Fault-Tolerant Processing Method and Strategy of Web Service Composition. Advances in Computer Science, Environment, Ecoinformatics, and Education, p. 428–432.

[5] Dudley, R. M. (2010). Distances of probability measures and random variables.Selected Works of RM Dudley, p. 28–37.

[6] Fan, G., Yu, H., Chen, L., Liu, D. (2009). A Method for Modeling and Analyzing Fault-Tolerant Service Composition, *In*: Software Engineering Conference. APSEC'09. Asia-Pacific. p. 507–514.

[7] Fu, X., Zou, P., Jiang, Y., Shang, Z. (2007). Qos consistency as basis of reputation measurement of web service, *In:* Data, Privacy, and E-Commerce, 2007. ISDPE 2007. The First International Symposium On. p. 391–396.

[8] Hui, S. C., Fong, A. C. M., Jha, G. (2001). A web-based intelligent fault diagnosis system for customer service support. *Engineering Applications of Artificial Intelligence*, 14, 537–548.

[9] Jannach, D., Gut, A. (2011). Exception handling in web service processes. The evolution of conceptual modeling, p. 225–253.

[10] Jensen, F. V., Nielsen, T. D. (2007). Bayesian networks and decision graphs. Springer Verlag.

[11] Keller, A., Ludwig, H. (2003). The WSLA framework: Specifying and monitoring service level agreements for web services. *Journal of Network and Systems Management*, 11, 57–81.

[12] Peng, J., Zhu, X. H. (2010). Study on supply chain exception-handing based on web service. *Computer Engineering and Design*, 31, 428–431.

[13] Peng, Y., Ding, Z. (2005). Modifying Bayesian networks by probability constraints, *In*: Proceedings of 21st Conference on Uncertainty in Artificial Intelligence. p. 26–29.

[14] Vomlel, J., Vomlel, R. I. (1999). Methods Of Probabilistic Knowledge Integration.

[15] Wang, H., Wang, G., Chen, A., Wang, C., Fung, C. K., Uczekaj, S. A., Santiago, R. A. (2006). Modeling Bayesian Networks for Autonomous Diagnosis of Web Services, in: Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS), Melbourne Beach, FL. p. 854–895.

[16] Wang, H. R., Li, Y., Jiang, Y. (2011). Web service and multi-agent based fault diagnosis system for MSW incinerators. *Advanced Materials Research,* 179, 580–585.

[17] Xiaodong, F., Ping, Z. (2007). Exception Handling Model for Web Service Composition Using Meta-Process, *In*: Control Conference. CCC 2007. Chinese.Presented at the Control Conference.CCC 2007.Chinese, p. 436 –439.

[18] Yan, Y., Dague, P., Pencole, Y., Cordier, M. O. (2009). A model-based approach for diagnosing fault in web service processes. *International Journal of Web Services Research* (IJWSR) 6, 87–110.

[19] Zhang, S., Peng, Y., Wang, X. (2008). An efficient method for probabilistic knowledge integration, *In*: Tools with Artificial Intelligence. ICTAI'08.20th IEEE International Conference On. p. 179–182.

[20] Zheng, T., Jia, B., Pan, W., Chen, W. (2010). An Exception Handling Capability-Aware Web Service Selection Method, *In*: Computer Modeling and Simulation. ICCMS'10.Second International Conference On. p. 452–455.

Author Biographies

Yao Chen is a Ph.D. candidate atComputer Schoolof Wuhan University. His research interests include Exception Handing, Formal Methodand SoftwareArchitecture.

Shi Ying is currently a professor at Computer School of Wuhan University, where he is vice dean in theComputer School. His current research interests include Service-oriented Software Engineering,Semantic Web Service, Trustworthy Software and Cloud Computing and the Internet of Things.

Liqiang Zhang received the PhD degree from Wuhan University of China in 2008. He is an Assistant Pprofessor at State Key Lab of Software Engineering(Wuhan University). His research interests cover many areas in the fields of information security and software engineering.