Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

LBD4WS: LOG-BASED DIAGNOSIS FOR WEB SERVICE

ZHICHUN JIA, RONG CHEN

School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China

ABSTRACT

With the development of web service application, it is highly desirable to quickly and exactly localize the service faults and explain the faulty reason. In this paper, we propose a log-based diagnosis method (LBD4WS) which combines MBD method with the statistical techniques based on historical data to construct the diagnosis model. LBD4WS localizes the service faults and explains the faulty reason by comparing the actual execution trace with the correct execution trace. We provide a case study to illustrate the diagnosis process of our method and use three BPEL (Business Process Execution Language) processes to conduct the experiments for evaluating our method. The experimental results show that our method is more effective than two existing MBD methods in the diagnostic accuracy for web services.

Keywords: *BPEL Process, Fault Diagnosis, Web Service, Execution Log*

1. INTRODUCTION

Web services provide a standard means of communication among different software applications on Internet [1]. With the popularization of web service applications, how to quickly analyze and localize the service faults during the service execution becomes a big challenge.

There exist already many works dedicate to detect abnormal behavior for diagnosing the faulty services [2-11]. Such as, Kopp in [12] proposed the different fault handling strategies to the different layers of the web service stack by analyzing the interplay between the different layers. To solve the problem of partial knowledge, Mayler in [13] proposed an method to isolate minimal sets of faulty activity executions based on the structure of a given process. Console in [14] proposed a decentralized model-based diagnosis approach to complex systems. Li in [15] proposed a Colored Petri nets model based fault diagnosis method. Wang in [16] proposed a Bayesian network model based method to autonomous diagnosis.

Although above these methods provide the better solutions in fault diagnosis of web services, no single method is adequate to handle all the requirements for a diagnostic system [17]. The model-based diagnosis methods (MBD) exploit knowledge about system structure, function and behavior and provide device-independent diagnostic procedures [18]. Unfortunately, the complete prior knowledge can generally be unavailable in practice, and the diagnosis systems based on model are difficult to update. The process history based methods are easy to implement. However, the diagnostic capability of the process history based methods is limited when available historical data is finite. Moreover, most process history based methods can't provide the explanations for the faults.

In this paper, we propose a log-based diagnosis method for diagnosing the service faults. Our main contributions are summarized as follows.

- We present a log-based diagnosis model (LBDM) to translate the service execution log into LBDM model for service diagnosis. The LBDM uses the probability to describe the relations between activities and messages in the web services.
- We propose a log-based diagnosis method for web services (LBD4WS) which provides an explanation for the service faults by finding the discrepancy between the correct execution trace and the exception execution trace.
- We provide a case study to illustrate the diagnosis process and conduct the experiments to evaluate our method. The experimental results show that our method is more effective than two existing MBD methods.

The rest of paper is organized as follows. Section 2 introduces the BPEL process. Section 3 proposes the log-based diagnosis model and diagnosis method. Section 4 presents a case study for illustrating our method. Section 5 evaluates the experimental results. Section 6 discusses the related work. Finally, we conduct the conclusion in Section 7.

2. DESCRIPTION OF BPEL

BPEL is a standard executable language for specifying actions within business process based on

Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

composite web services. BPEL gives the formal specification for defining business process behavior and business interaction protocols. It provides two kinds of activities for describing the internal behavior: basic activity and structured activity. Structured activities prescribe the order in which a collection of activities take place. They describe how a business process is created by composing the basic activities [19].

The set of basic activities includes:

- receive: waits for a matching message to arrive
- invoke: invokes an operation on a port offered by a partner
- reply: replies a message to a partner
- assign: copies to target variables with new data
- wait: waits for a given time period or until a certain time has passed
- throw: signals a fault
- empty: does nothing

The set of structured activities includes:

- sequence: defines a collection of activities to be performed sequentially
- switch: selects exactly one branch of activity by a condition it holds true
- while: defines loop execution of an activity until the condition has been met
- flow: specifies some activities to be performed concurrently
- pick: selects an activity path depending either on an occurring event or timeout

3. LBD4WS

In this section, we first present our diagnosis model and its construction algorithm. Then, we propose our diagnosis method, called log-based diagnosis for web service (LBD4WS), for diagnosing service faults.

3.1 Log-Based Diagnosis Model

The log-based diagnosis model (LBDM) incorporates three sets: activity set, message set and dependence set, each of which is associated with a probability distribution. The activity set represents the transition relationships between activities and consists of a set of transition probabilities among the activities. Similarly, the message set describes the transition relationships between the messages and consists of a set of transition probabilities among messages. The dependence set denotes the relationships between activities and their inputs and outputs and consists of a set of emission probabilities from activities to their inputs and outputs.

Definition 1. A log-based diagnosis model for the web service *ws* is a 3-triple $LBDM = (AS, MS, DS)$, where:

- $AS = \{ap_{ij} \mid 1 \le i \le k, 1 \le j \le k\}$, ap_{ij} is the transition probability from activity a_i to activity a_j, k is the number of activities in *ws*;
- $ap_{ij} = an_{ij} / an_{i*}$, an_{ij} is the number of transitions from activity a_i to activity a_j and an_{i*} is the number of transitions starting from activity a_i;
- $MS = \{mp_{ij} \mid 1 \le i \le r, 1 \le j \le r\}$, mp_{ij} is the transition probability from message m_i to message m_j, and r is the number of messages in *ws*;
- $mp_{ij} = mn_{ij} / mn_{i*}$, mn_{ij} is the number of transitions from message m_i to message m_j and mn_{i*} is the number of transitions starting from message m_i;
- $DS = \{dp_{ij} \mid 1 \le i \le k, 1 \le j \le r(2r-1)\}$, dp_{ij} is the emission probability from activity a_i to message pair mp_j, $r(2r\text{-}1)$ is the number of message pairs combining any two messages in *ws*;
- $dp_{ij} = dn_{ij} / dn_{i*}$, dn_{ij} is the number of emissions from activity a_i to message pair mp_j and dn_{i*} is the number of emissions starting from activity a_i.

For facilitating the generation of LBDM, we assume that the execution log of the web service $SL = \{st_i \mid 1 \le i \le n\}$ is known, where:

- each execution trace st_i is a successful execution trace;
- n denotes the number of the execution traces in *SL*;
- st_i denotes an execution trace in *SL* and $st_i = \{(a_{ij}.in, a_{ij}, a_{ij}.out) \mid 1 \le j \le u\}$;
- a_{ij} denotes an activity in st_i and u denotes the number of activities in st_i;
- $a_{ij}.in$ denotes the input message of a_{ij} and $a_{ij}.out$ denotes the output message of a_{ij}.

Algorithm 1 is used to construct the LBDM according to the computing probabilities of transitions and emissions and described as follows.

Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm 1: *LBDMCons*
Input: service log *SL*
Output: diagnosis model $LBDM(AS, MS, DS)$

for $i = 1:n$ //number of traces in *SL*
  $a = SL(i,1)$
  $mn(a.in, a.out)++, mn\_(a.in)++$
  $dn(a, a.out)++, dn\_(a)++$
  for $j = 2:u$ //number of activities in st_i
    $a_1 = SL(i, j-1), a_2 = SL(i, j)$
    $an(a_1, a_2)++, an\_(a_1)++$
    $mn(a_2.in, a_2.out)++, mn\_(a_2.in)++$
    $dn(a_2, (a_2.in, a_2.out))++, dn\_(a_2)++$
  end for
end for
for $i = 1:k$ //number of activities in *SL*
  for $j = 1:k$
    while $an(i, j) \neq 0$ and $an\_(i) \neq 0$
      $AS(i, j) = an(i, j) / an\_(i)$
    end while
  end for
end for
for $i = 1:r$ //number of messages in *SL*
  for $j = 1:r$
    while $mn(i, j) \neq 0$ and $mn\_(i) \neq 0$
      $MS(i, j) = mn(i, j) / mn\_(i)$
    end while
  end for
end for
for $i = 1:k$ //number of activities in *SL*
  for $j = 1:r(2r-1)$
    while $dn(i, j) \neq 0$ and $dn\_(i) \neq 0$
      $DS(i, j) = dn(i, j) / dn\_(i)$
    end while
  end for
end for
return $LBDM(AS, MS, DS)$

3.2 Diagnosis Method

To localize the faults when the web service throws an exception during the execution, the log-based diagnosis method for web service (LBD4WS) firstly finds the message pair of not existing and uses the message pair with the maximum transition probability instead of it to obtain a correct message trace. Then the LBD4WS finds a correct activity trace of maximum likelihood with the correct message trace according to the LBDM. Finally, the LBD4WS is able to localize the faults and explain the faulty reason by comparing the correct activity trace with the exception execution.

For facilitating diagnosis, LBD4WS assumes that the exception execution trace is available. Moreover, for the sake of mathematical and computational tractability, LBD4WS makes three assumptions: the next executing activity are dependent only on the current executing activity; the activity transition probability is independent of the actual run time at which the transition takes place, and the current input message is statistically independent of the previous output message. The diagnosis algorithm of LBD4WS is described as follows.

Algorithm 2: *LBD4WS*
Input: diagnosis model $LBDM(AS, MS, DS)$, exception execution trace *exptr*
Output: fault set *FS*

for $i = 1:u$ //number of activities in *exptr*
  $a = exptr(i).a$
  $in = exptr(i).in, out = exptr(i).out$
  $cortr(i) = exptr(i)$ //correct trace
  if $MS(in, out) = 0$
    $cortr(i).out = maxOut(in, MS)$
    if $MS(in, cortr(i).out) = 0$
      $cortr(i).(in, out) = maxPair(a, DS)$
    end if
  end if
end for
for $i = 1:k$ //number of activities in *LBDM*
  if $\exists AS(i,*) \neq 0$ and ó$AS(*,i) > 0$
    $tp(1,i) = DS(i, (cortr(1).(in, out))$
  else $tp(1,i) = 0$
end for
for $l = 2:u$
  $tp(l, j) = max_{1 \le i \le k}(tp(l-1, i) AS(i, j))$
    □$DS(j, (cortr(l).(in, out))$
  $oa(l, j) = maxAct_{1 \le i \le k}(tp(l-1, i) AS(i, j))$
end for
$cortr(u).a = maxAct_{1 \le i \le k}(tp(u, i))$
for $i = u-1:1$
  $cortr(i).a = oa(i+1, cortr(i+1).a)$
end for //obtain the correct activity sequence
$fcn = 1$ //number of faulty activities
for $i = 1:u$
  if $exptr(i) \neq cortr(i)$
    $FS(fcn) = exptr(i), fcn++$
end for
return *FS*

In Algorithm 2, $maxOut(in, MS)$ is to obtain the output message with the maximum probability in

Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$MS(in,*)$. $maxPair(a,DS)$ is to obtain the message pair with the maximum probability in $DS(a,*)$. $\max_{1\le i\le k}(f(i,j))$ is to obtain the maximum value in the function $f(i,j)$. $maxAct_{1\le i\le k}(f(i,j))$ is to obtain the j with the maximum value in the function $f(i,j)$. Here, * is any element in a set.

4. CASE STUDY

In this section, we describe a case study to illustrate the diagnosis process of LBD4WS. The online book shopping service is selected as a real-life manufacturing BPEL process in order to make our case study more understandable. The service process is shown in Figure 1.

The online book shopping service interacts with four services. When a client needs to buy a book, he sends a request to the online book shopping service by client interface. The service receives this request and sends client information to client class service. Client class service returns the class standard (eg. common client, vip class) of the client to the online book shopping service according to the client information. The online book shopping service sends received client class standard and the name of book to Electronics Industry Press web service and Academic Press web service. These two services retrieve book price by received information and return book price to the online book shopping service. Finally, the online book shopping service selects the cheapest book and sends the result to client.

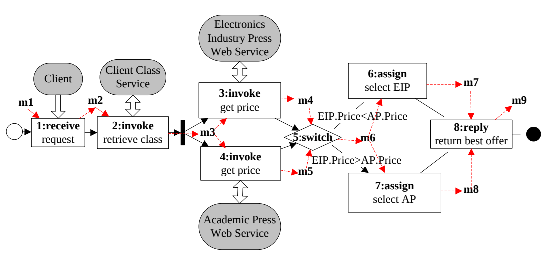

Figure 1: Online book shopping service

To evaluate our method we need to simulate the service process to generate the execution log of the online book shopping service. Suppose there is some semantic incompatibility between the Electronics Industry Press web service and the Academic Press web service. A customer sent an offer request like: achieve a cheaper price from the online book shopping service, Electronics Industry Press web service returns a price 200 in RMB, and Academic Press web service returns a price 45 in dollar (＄45≈￥307). So the customer will receive wrong offer information (Academic Press) from the online book shopping service. In this case, an exception occurs due to a semantic fault. To simulate this unreliable environment, the output of the activity 3 in this example is devised to output successfully at a probability of 0.2, and other activities are devised to always complete successfully. We assume that the exception trace is {(m1,1,m2),(m2,2,m3),(m3,4,m5),(m3,3,m_),((m_, m5),5,m6),(m6,6,m7),(m7,8,m9)}. According to the algorithm 2, the correct trace is {(m1,1,m2),(m2,2,m3),(m3,4,m5),(m3,3,m4),((m4,m5),5,m6),(m6,6,m7),(m7,8,m9)} and the probability of this correct trace is 0.0009. Comparing the two traces, we obtain that the faulty activity is the activity 3, that is $FS=\{3\}$. LBDM model for the online book shopping is as follows.

Table 1: Activity Set For Online Book Shopping Service

	1	2	3	4	5	6	7	8
1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.44	0.56	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.44	0.56	0.00	0.00	0.00
4	0.00	0.00	0.56	0.00	0.44	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.17	0.83	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.17
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.83
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 2: Message Set For Online Book Shopping Service

	m2	m3	m4	m5	m6	m7	m8	m9
m1	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
m2	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
m3	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
m4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
m5	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
m6	0.00	0.00	0.00	0.00	0.00	0.17	0.83	0.00
m7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
m8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
m9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 3: Dependence Set For Online Book Shopping Service

	1	2	3	4	5	6	7	8
m1,m2	1	0	0	0	0	0	0	0
m2,m3	0	1	0	0	0	0	0	0
m3,m4	0	0	1	0	0	0	0	0
m3,m5	0	0	0	1	0	0	0	0
(m4,m5),m6	0	0	0	0	1	0	0	0
m6,m7	0	0	0	0	0	1	0	0
m6,m8	0	0	0	0	0	0	1	0
m7,m9	0	0	0	0	0	0	0	0.17
m8.m9	0	0	0	0	0	0	0	0.83

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. EXPERIMENT

In this section, we set up our experiments on three real life BPEL service processes and evaluate LBD4WS by comparing with two model-based diagnosis methods.

In our experiments, we use the accuracy of diagnosis as our evaluation metric. Following is the description of labels that we use to denote each of these algorithms:

- **Con**. This is the consistency-based diagnosis method of [14].
- **Yan**. This is the MBD method used in [20].
- **LBD**. This is our diagnosis method.

We carry out experiments on three BPEL service processes. Table 4 shows the characteristics of these processes. Process 1 has 8 activities, 9 message types and contains flow and switch structures. Process 2 has 15 activities, 18 message types and contains flow and switch structures. Process 3 has 23 activities, 24 message types and contains flow, switch and while structures.

To evaluate the effectiveness of our method, we randomly select an activity as the faulty activity for each service. Each faulty activity is devised to complete successfully at a probability of 0.2, and other activities are devised to always complete successfully. Then we generate 100 correct execution traces for each service according the given probabilities and each service generate 10 groups of test data.

Table 4: Characteristics Of The Processes

Process	activity	message	flow	switch	while
1	8	9	1	1	0
2	15	18	1	2	0
3	23	24	1	0	1

In Figure 2, the vertical axis represents the diagnostic accuracy, which is the rate of the number of correct diagnosis to the total diagnosis number for each service process. From Figure 2, we can see that **LBD** is more effective than **Con** and **Yan** for three services. The accuracy of **LBD** is 80% to 90%, while the accuracy of **Con** and **Yan** is 50% to 60%. Experimental results show that our method is effective.

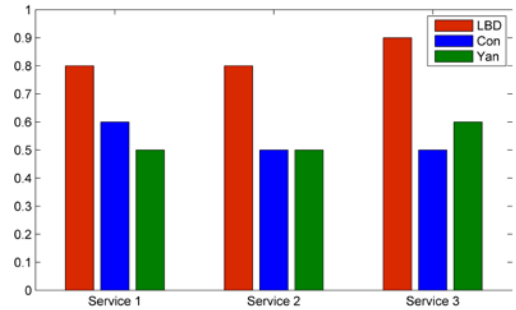

Figure 2: Comparison of accuracy

6. RELATED WORK

At present, some model-based methods have been proposed for the fault diagnosis of web services. Console in [14] proposed an architecture for decentralized model-based diagnosis of complex systems. In the architecture, the subsystems were developed independently along with their associated software modules. Console's diagnosis method adopted the grey-box models for individual services and hided the details of the aggregation to higher-level services, so that it could raise diagnostic efficiency and ensure the privacy of service. Moreover, this method assumed that faults were expressed by inconsistencies between the exception execution trace and the expected execution trace. Li [15] proposed a diagnostic method for BPEL processes, where processes were modeled using colored Petri nets and different colors denoted different states of places. According to color propagation functions, local diagnosis service checked back from where system thrown the exception until arriving at a final consistency. Differing from Console's diagnosis model, each local diagnostic service included a fault models. Yan [20] applied synchronized automaton to model the business process of web services composition. After an exception was thrown, the diagnosis service calculated the process execution trace to localize the service faults according to the given rules. Mayer in [13] proposed a method to identify the fault of web service composition by observations obtained from partial executions and re-executions of a process were exploited. He indicated that a detailed specification of service behavior and detailed fault models were not generally satisfied in practice. Therefore, specification based methods can't always correctly compute the set of minimal diagnoses.

In addition, Lakshmi and Mohanty in [21] proposed a method based on stochastic automata model for web service fault monitoring and diagnosis. This model associated each transition with a value. The value specified the probability of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

transitional occurrence. By comparing observable trace with predicted trace, they can find faulty reason. Kemper in [22] proposed an algorithm to identify and remove cycles from process history trace. This method can help to identify faults by finding different fragments with cyclic fragments in the trace.

Our method combines MBD method with the statistical techniques based on historical data to construct the diagnosis model and localize the service faults. Our method not only can find the faults but also explain the faulty reason.

7. CONCLUSION

In this paper, we propose a log-based diagnosis method for web services. The transition and emission relations between activities and messages are modeled by computing their probabilities in the execution log of web service. Our diagnosis method finds a correct execution trace of maximum likelihood with the correct message execution trace and compares the exception execution with the correct one. Finally, we obtain a set of discrepancies between the two traces and consider this discrepancy set as the fault set. Moreover, we provide a case study to illustrate the diagnosis process of our method and conduct the experiments to compare the accuracy of our method with two MBD methods. Experimental results show that our method is effective in diagnosing the service faults.

ACKNOWLEDGEMENTS

This work is supported by National Natural Science Foundation of China (61175056), the Dalian Maritime University Backbone Youth Foundation (NO.2011QN033, No.2009JC29), and IT Industry Development of Jilin Province.

REFERENCES:

[1] Y. Li, T. Melliti,P. Dague, "Modeling BPEL Web services For Diagnosis: towards self-healing Web services", Proceedings of 3rd International Conference on Web Information Systems and Technologies (WEBIST'07), 2006, pp. 1-62.

[2] L. Ardissono, S. Bocconi, L. Console, et al., "Enhancing Web Service Composition by Means of Diagnosis", Business Process Management Workshops, 2008, Vol. 17, pp. 468-479.

[3] L. Ardissono, L. Console, A. Goy, et al., "Enhancing Web Services with Diagnostic Capabilities", Proceedings of the Third European Conference on Web Services (ECOWS'05), 2005, pp. 182-191.

[4] L. Ardissono, R. Furnari, A. Goy, et al., "Fault tolerant web service orchestration by means of diagnosis", *Software Architecture*, 2006, Vol. 4344, pp. 2-16.

[5] S. Bocconi, C. Picardi, X. Pucel, et al., "Model-Based Diagnosability Analysis for Web Services", 10th Congress of the Italian Association for Artificial Intelligence, 2007, pp. 24-35.

[6] Y. Dai, L. Yang, B. Zhang, et al., "Exception diagnosis for composite service based on error propagation degree", 2011 IEEE International Conference on Services Computing, SCC 2011, 2011, pp. 160-167.

[7] X. Han, Z. Shi, W. Niu, et al., "Similarity-Based Bayesian Learning from Semi-structured Log Files for Fault Diagnosis of Web Services", in Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on, 2010.

[8] S. C. Hui, A. C. M. Fong, G. Jha, "A web-based intelligent fault diagnosis system for customer service support", *Engineering Applications of Artificial Intelligence*, 2001, Vol. 14, No. 4, pp. 537-548.

[9] Z. Zhu, W. Dou, "QoS-Based Probabilistic Fault-Diagnosis Method for Exception Handling", New Horizons in Web-Based Learning: Icwl 2010 Workshops, 2011, Vol. 6537, pp. 227-236.

[10] A. Ait-Bachir, M. C. Fauvet, "Diagnosing and Measuring Incompatibilities between Pairs of Services", Database and Expert Systems Applications, Proceedings, 2009, Vol. 5690, pp. 229-243.

[11] M. Alodib, B. Bordbar, "A model-based approach to Fault diagnosis in Service oriented Architectures", ECOWS'09-7th IEEE European Conference on Web Services, 2009, pp. 129-138.

[12] O. Kopp, F. Leymann, D. Wutke, "Fault Handling in the Web Service Stack", in *Service-Oriented Computing*, 2010, pp. 303-317.

[13] W. Mayer, G. Friedrich, M. "Stumptner, Diagnosis of Service Failures by Trace Analysis with Partial Knowledge", in *Service-Oriented Computing*, 2010, pp. 334-349.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[14] L. Console, C. Picardi, D. T. Dupre, "A Framework for Decentralized Qualitative Model-Based Diagnosis", in Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI'07), 2007, Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, pp. 286-291.

[15] Y. Li, L. Ye, P. Dague, et al., "A Decentralized Model-Based Diagnosis for BPEL Services", Proceedings of the 2009 21st IEEE International Conference on Tools with Artificial Intelligence (ICTAI '09), 2009, pp. 609-616.

[16] H. Wang, G. Wang, A. Chen, et al., "Modeling bayesian networks for autonomous diagnosis of web services", in FLAIRS 2006 - Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, 2006, Florida, pp. 854-859.

[17] V. Venkatasubramanian, R. Rengaswamy, K. Yin, et al., "A review of process fault detection and diagnosis Part III: Process history based methods", *Computers & Chemical Engineering*, 2003, Vol. 27, No. 3, pp. 327-346.

[18] C. Angeli, "Online expert systems for fault diagnosis in technical processes", *Expert Systems*, 2008, Vol. 25, No. 2, pp. 115-132.

[19] M. Tony Andrews, I. Francisco Curbera, S. S. Hitesh Dholakia, et al., "Business Process Execution Language for Web Services", 2003, pp. 1-136.

[20] Y. Yan, P. Dague, Y. Pencole, et al., "A Model-based Approach for Diagnosing Faults in Web Service Processes", *The International Journal of Web Services Research (JWSR)*, 2009, Vol. 6, No. 1, pp. 87-110.

[21] H. N. Lakshmi, H. Mohanty, "Automata for Web Services Fault Monitoring and Diagnosis", *IJCCT*, 2011, Vol. 3, No. 2, pp. 13-18.

[22] P. Kemper, C. Tepper, "Automated Trace Analysis of Discrete-Event System Models", *IEEE Transactions on Software Engineering*, 2009, Vol. 35, No. 2, pp. 195-208.