ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, No 5, May 2013

A Verification Method Based On Petri Net for Online Student Registration Web Service Composition

A.Kumaravel, Professor and Dean, M.Merlin Mexa, PG Student, Department of Computer Science and Engineering.
Bharath University, Selaiyur, Chennai-600073, India

***Abstract*—The Main area in Web service is the Web service composition which detects the composition before execution. Petri net or Place / Transition net is a mathematical modeling language used to describe the distributed systems. Deadlock can be detected through transitive matrix of Petri nets .This paper proposes a Web service model for students' online registration and detects the deadlocks for various applications using Petri net transitive matrix. Simulation results indicate a better identification of deadlock.**

***Keywords* — *Web Service*, Petri Net, Transitive Matrix, Deadlock Detection.**

I. INTRODUCTION

The World Wide Web infrastructure is extended using Web services which provide the means for software to connect to other software applications. Using ubiquitous Web protocols and data formats such as HTTP,XML, SOAP every application is allowed to access the Web services which bout the knowledge about the Web service .Higher level business processes are created using Web service composition which provides an open standard based approach. Complexity to compose Web service is reduced to a greater level using open standards. This reduces time and cost and increases overall efficiency business models. Web service is a technology based on SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language), UDDI (Universal Description, Discovery and Integration) and HTTP (Hyper Text Transfer Protocol). Web service is an important part of the SOA architecture where the applications can be executed for isomeric environment. Limited functionalities are provided by a single Web service. In order to provide a concrete complex service request several Web service are combined together. Web service composition is not reliable because it comes from different companies and from different platform .so the key verification before any process is to verify the correctness of service composition as specified in [1]–[3].

Service profile, service model and service grounding are the three parts of OWL-S representation. Web service composition based on service model not only considers the input and output, but also includes the execution sequence of the behavior process. In [4] proposed a model which detects the deadlock of Web service composition based on behavior which deals with two functional module, function analyzer and behavior analyzer. Web service composition model based on the color Petri net proposed by [5] describes the operation of Web service which characterizes control flow and data flow. Paper [6] uses transitive matrix of Petri nets as a detection tool to identify deadlock. Various direct composition methods have been proposed, including planning based [7], logical inference driven [8], Petri net based [9], [10], automata based [11], quality-of-service (QoS) optimizing based [12], etc. These methods [7]–[12] assume both data formats and sequences of the messages are consistent. In order to reduce space in matrix, compression technique has also been used. Two important compression concepts are lossy and lossless compression:

- **Lossy compression** With lossy compression, it is assumed that some loss of information is acceptable. Loss is also acceptable in voice and audio compression, depending on the desired quality.
- **Lossless compression** With lossless compression, data is compressed without any loss of data. It assumes you want to get everything back that you put in. Critical financial data files are examples where lossless compression is required.

This paper proposes an online student registration of Web service composition and verification method. This paper achieves detection of deadlock using transitive matrix of petrinet. Efficiency is achieved by compressing and decompressing the transitive matrix. Finally examples are shown how effectively the deadlock is detected and solved for Web service composition. In section 1 a brief introduction about Web service composition is explained and in section 2 Definitions of Petri net and the transitive matrix is explained followed by the detailed explanation about the proposed work and its experimental process is given in section 3. In section 4 verification of Web service composition is explained. Conclusion and the future work is specified in section 5.

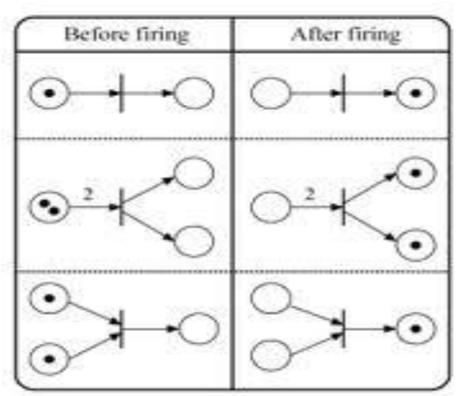
II. BACKROUND

Definition I: (Petri net) [13-15] A Petri net is a 5-tuple $PN =< P,T, A,W,M0 >$ where:
1) P= { $p_1,p_2 \ldots ,p_m$ }is a finite set of places,
2) T= {$t_1,t_2,\ldots t_n$,}is a finite set of transitions,
3)A $\subseteq (P\times T)\cup(T \times P)$ is the set of arcs (flow relation),
4 $W : A\rightarrow\{1,2,3,\ldots\}$ is a weight function,
5) $M : P\rightarrow\{0,1,2,3,\ldots\}$ is the initial marking, $P\cap T = \Phi$ and $T \cap P = \Phi$.

Definition II: (transitive matrix) [14] Let B^- and B^+ be an input function matrices and output function matrix of m rows by n columns in Petri nets, respectively. The place and transition transitive matrix are as follows:

$M_{PR} = B^-(B^+)^T$: place transitive matrix
$M_T = (B^+)^T B^-$: transition transitive matrix

Table I: shows the transition of token between places before and after firing.

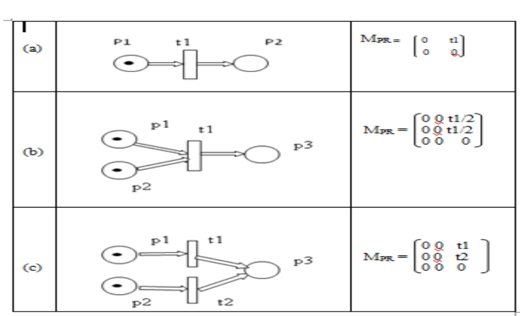


Definition III: Let a reachable marking $M_R(K +1)$ from $M(K)$ be an m-vector of nonnegative integer. The transformation defined by $M_R\ (K+1)^T = M(K)^T\ M_{PR}$

Its show a flow relation of $M(K)$ to $M_R(K +1)$ based on the token in) *Pi* (*k*). Also, we can find that if $\sum_i p_i M_{PR_{ij}}$ is integer (or not), then let $\sum_i p_i M_{PR_{ij}}$ = 1(or 0). This means the token in $P_i(K)$ can (or can't) be transferred from $M(K)$ to $M_R(K +1)$. Since $M_R(K +1)$ only represents the flow relation of tokens, $M_R(K +1)$ may not necessarily correspond to $M(K +1)$. Henceforth, we consider a limit to the number of tokens that $|P_i(K)|$ is equal to 1 or 0.

Definition IV: Weighted Place Transitive Matrix M_{PR}. Let M_{PR} be the $m\times m$ weighted place transitive matrix. If a transition t_k appears s times in the same column of M_{PR}, then we will replace t_k by t_k/s in M_{PR}.

Table II (Example): Consider a simple net as follows:



ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, No 5, May 2013

Table II gives a few marking place transitive matrixes M_{PR} corresponding to simple Petri nets. It is proved that Petri nets and their corresponding marking place in the transitive matrixes M_{PR} are semantic consistent

This example net explained like as
$M(K) = [P_1(k), P_2(k), \ldots, P_3(k)]$then we can obtain
$M_R = [0,0,t_1\,(P_1(k))+t_2(P_2\,(k))]^T$.
In this example(c), next marking may have maximum two tokens from transition t1 and t2: like as
MR = [0,0,2].

III. PETRI NET REPRESENTATION OF WEB SERVICE

Fig.1 shows a Petri net representation of an atomic Web service process. Circles represent the input and the output of service. Diamond and the rectangle represent the control place and transition respectively. A series of circles above the transition is the input place and below the transition is the output place. The default weight of the arcs connecting places and the transition is 1.The service can start to execute only if the input place and the control place has tokens and also it should be consistent.

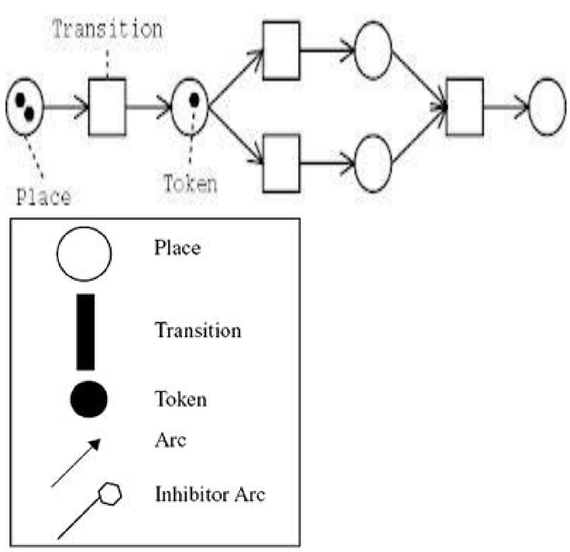


Fig.1 Petri net representation of an atomic Web service process

Fig.2 and Fig.3 shows the behavior description of two Web service. Fig.2 is an online Registration service for students. This allows the students to login and register their information like list of courses and payment facilities. Payment facilities provide users with DD number.
Fig.3 shows the online Examination services where the students have to log in using their username and password. If the student is an authorized student he can enter the register number and attend the examination using the online question paper.

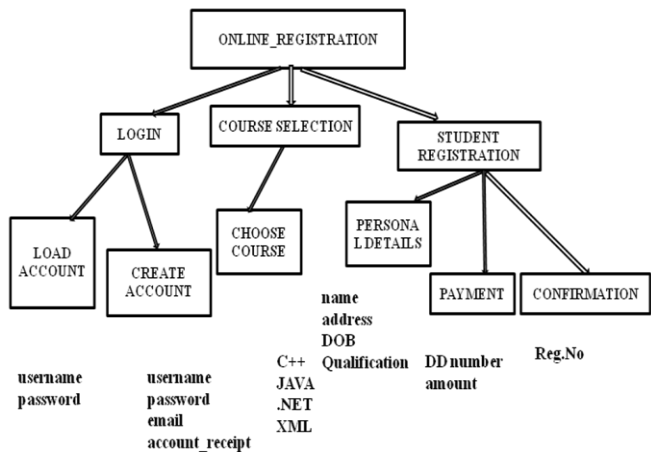


Fig.2 Online Registration Service

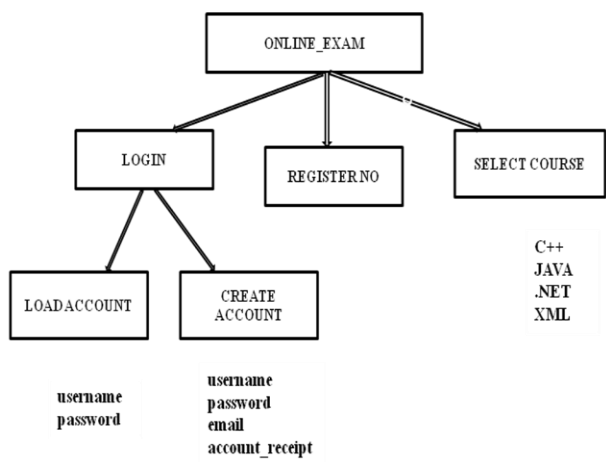


Fig.3 Online Examination Service

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, No 5, May 2013

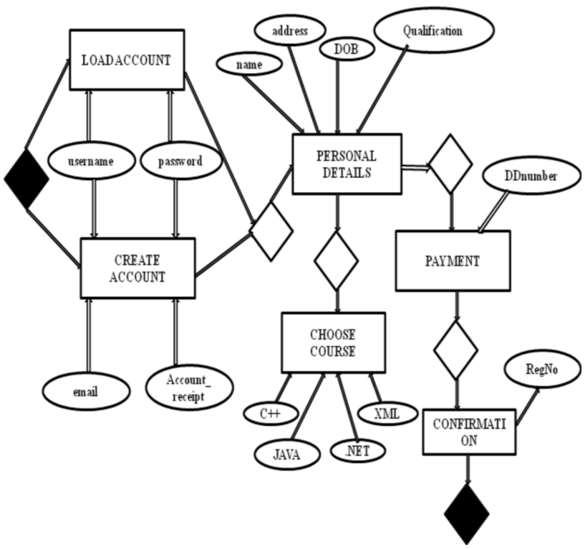


Fig.4 Petri net representation of online Registration

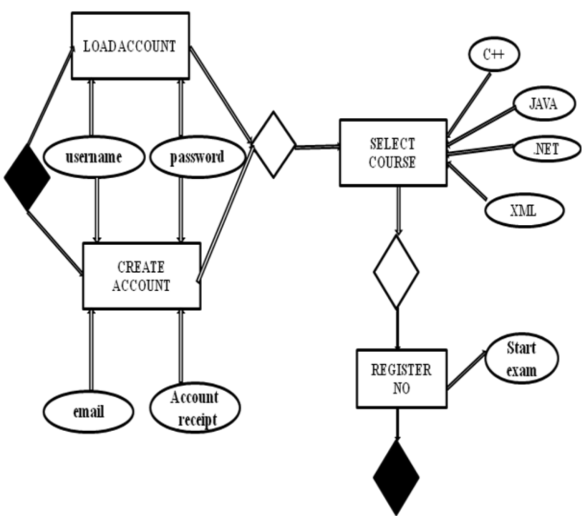


Fig.5 Petri net representation of Online Examination

Fig.4 and Fig.5 shows the Petri net representation of two services in Fig.2 and Fig.3.The dark diamonds are the start and end control places added for Petri net. If duplicate I/O data is present in data stream of two atomic processes are in selection structure then it can be merged because it will not lead to a conflict or collision in Petri nets.

IV. VERIFY WEB SERVICE COMPOSITION USING PETRI NET

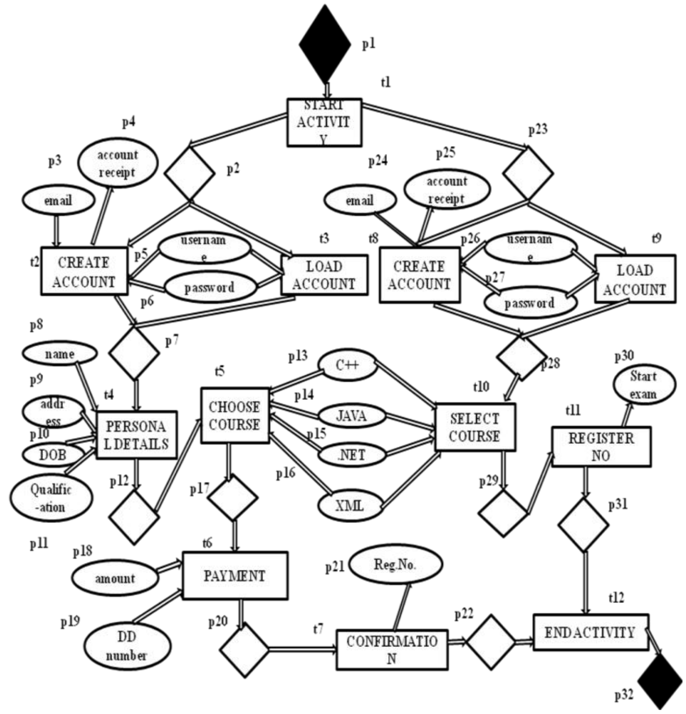


Fig.6 Composition of Petri net description for Web service

After finishing the transform of Web service from OWL-S description to Petri net representation, the composition of Petri net representation of Web service in parallel, and the method are explained in [13]-[15] in detail. The two Petri net compositions of Web services are shows in Figure 6, where the two data places interacts between the two services. The same data places are composited into one, such as P13-P16 is the interaction data places between two Web services.

From the start control place p1 of composition petrinet represented in Fig.6, the input parameter given by the user are username(p5,p26), password(p6,p27), course(p12,p28), email(p3,p24), account receipt(p4,p25).

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, No 5, May 2013

Table III. Explains the transitive matrix M_{PR}

$$
\begin{pmatrix}
0&t_1&0&0&0&0&0&0&0&0&0&0&0&t_1&0&0&0&0&0&0&0&0&t_1&0&0&0&0&0&0&0&0&0\\
0&0&0&t_2/4&0&0&(t_2+t_3)/3&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&t_2/4&0&0&t_2&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&t_2/4&0&0&(t_2+t_3)/3&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&t_2/4&0&0&(t_2+t_3)/3&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&t_4/4&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&t_4&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&t_4&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&t_4&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&t_4&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_5/4&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&t_5&0&0&0&0&0&t_5&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_5&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_5&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_5&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_6/2&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_6&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_6&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_7&t_7&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_{12}\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_8/4&0&0&(t_8+t_9)/4&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_8&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&(t_8+t_9)/4&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&(t_8+t_9)/4&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_{10}/4&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_{11}&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&t_{12}\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{pmatrix}
$$

The initial state is represented as M (0). The output parameter examination is start exam (p30), final end control place (p32). Calculate to find the existence of deadlock in composition Petri net of two Web service in Figure 6, and check whether it is reachable from the place requested. Through calculations we can verify whether deadlock exists in composition of Petri net. Two states are compared whether it is reachable from the place requested.

The matrix multiplication operation on the state of Petri net M(k) and marking place transitive matrixes M_{PR}, which is equivalent to that Petri net fired a transit in state M(k),and get the next state of Petri net M(k+1). The transitive matrixes state deduction also uses the semantic consistent of Petri nets and their corresponding marking place transitive matrixes M_{PR}, and in the Petri net model of Web service composition, M(0) is the initial state. Next we get the state of M(1) after one transit, $M(1)=M(0)*M_{PR}$ and so forth, $M(2)=M(1)*M_{PR}$, … ,$M(n+1)=M(n)\times M_{PR}$. If the resulting state is M(k)={0,0,…,0]all-zero and

the compared states are equal then deadlock exists in composition of Petri net. If M(k+1)=[0,0,1], representing p1 and p2 transit successfully.

The service verification algorithm based on marking place transitive matrix is as follows:

1) Compute $M_{PR}=D^{-}\bullet diag(t1,t2,...,tn)\bullet(D^{+})^{T}$ according to the composition Petri net of Web service;

2) Replace t_k with t_k/d in M_{PR}, where d is the number of t_k transition in each column, we get MPR after replacement;

3) Let k=0, calculate M(0);

4) Compute $M_R(k+1)=M(k)*M_{PR}$;

5) In $M_R(k+1)$, if $\sum \frac{tki}{di} \geq 1$, $|t_k|=1$, otherwise $|t_k|=0$, substituted into $M_R(k+1)$, get $M(k+1)$;

6) If M(k+1)=M(k), go to (7); otherwise, let k=k+1, go to(4);

7) If M(k+1)=[0,0,…,0], composition Petri net is at a deadlock; if the token value of the output place that the user need in M(k+1) is 1, then finish correctly; otherwise, composition Petri net can't get the output that user need.

Table IV. Relationship between the Activities.

Activities	Process(pl)	Transition(t)	Arcs(a)	Places(p)	Rows(r)	Columns(c)
Student	9	7	29	22	22	32
Examination	5	5	18	10	10	32

Based on this relationship, it can be said that the column relation is a sum of the send tokens from the Place p and the row relation is a sum of the received token in place p. Therefore, if a sum of the row of place p is great than that of the column of place p, then this place will be not dead. In order to reduce space in transitive matrix, compression technique has also been used.

V. CONCLUSIONS

This paper detects the deadlock and ensures normal working of Web service. The transitive matrix explained all relationship between the places and transitions in the net. The represented service model not only considers input and output but also the behavior process of Web service. This paper uses the transfer matrix for analysis and detection tool to verify and detect the existence of deadlock. Finally, it can be said that this method is very easy to find the deadlock status and modify the deadlock status in the net.

ACKNOWLEDGEMENTS

This work was supported by the management of **Bharath University**, I would like to thank for the encouragement of this research work.

REFERENCES

[1] Lei Liu, Xiang Ji, Lei Zhang, Yanli Zhang, Shuai Lu A Novel Web Service Composition Verification Method based on Petri Net, International Journal of Digital Content Technology and its Applications(JDCTA) Volume6,Number11,June 2012

[2] Junyan Qian, Guowang Huang, Lingzhong Zhao. Semantic Web service composition using answer set planning. JACT: International Journal of Advancements in Computing Technology, 2011, 3(5): 20-31.

[3] Xiaolin Zheng, Zhen Lin, Yanbo Yang, Deren Chen. Research on context-aware Web service composition based on the fluent calculus. AISS: Advances in Information Sciences and Service Sciences, 2011, 3(7): 62-74

[4] Rachid Hamadi, Boualem Benatallah. A Petri net-based model for Web service composition. In:Proceedings of the 14th Australasian Database Conference (ADC 2003), Adelaide, South Australia, 2003, 191-200.

[5] Jingxia Li. Application of Colored Petri Net in Model for Web Service Composition (in Chinese). Beijing: Institute of Computing Technology, Chinese Academy of Sciences, 2006.

[6] M. Carman, L. Serafini, and P. Traverso, "Web service composition asplanning," in *Proc. Int. Conf. ICAPS*, 2003, pp. 45–51.

[7] X. Tang, C. Jiang, and M. Z. Zhou, "Automatic Web service compositionbased on Horn clauses and Petri nets," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13024–13031, 2011.

[8] R. Hamadi and B. Benatallah, "A Petri net-based model for Web service composition," in *Proc. Int. Conf. ADC, 2003*, pp. 191–200.

[9] J. M. Mendes, P. Leitão, F. Restivo, and A. W. Colombo, "Compositionof Petri nets models in service-oriented industrial automation," in *Proc. 8th Int. Conf. INDIN*, 2010, pp. 578–583.

[10] H. Foster, S. Uchitel, J. Magee, and J. Kramer, “Compatibility verification for web service choreography,” in *Proc. Int’l Conf. ICWS*, 2004, pp. 738–741.
[11] P. Xiong, Y. Fan, and M. Z. Zhou, “Web service configuration undermultiple quality-of-service attribute,” *IEEE Transactions on AutomationScience and Engineering*, vol. 6, no. 2, pp. 311–321, 2009.
[12] Lefebvre D, Leclercq E. Stochastic Petri Net identification for the fault detection and isolation of discrete event systems. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2011, 41(2): 213-225.
[13] Feng He. Research on Some Key Technologies of Web Services Composition (in Chinese).
Shanghai: Donghua University, 2008.
[14] Zhehui Wu. Introduction to Petri Nets. Beijing: China Machine Press, 2006.
[15] Karakoc E, Senkul P. Composing semantic Web services under constraints. Expert Systems with Applications, 2009, 36(8): 11021-11029.