

Soundness Analysis of Logic Service Net Based on Service Clusters

¹YuYue Du and ^{1,2}Qiang Hu

¹College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

²School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China

Corresponding Author: Qiang Hu, College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

ABSTRACT

To reduce the difficulty of binding a web service and improve the self-adaptability of service response, the concept of service cluster is proposed. A service cluster consists of a group of the services with similar functions and interfaces, by enlarging the grain of service requested, it can efficiently reduce the search space and improve the function compatibility. The algorithm to bind an appropriate web service in service clusters is presented. Logic service net constructed based on logic Petri net is put forward to describe the service flow based on service clusters. A judging theorem to test the soundness of a logic service net is also proposed in this study.

Key words: Service cluster, logic petri net, logic service net, soundness

INTRODUCTION

With the development and maturity of service computing technology, the number of available web services is surging dramatically and the difficulty to search and bind a service is increased greatly. It needs to find another available service to substitute the current working service once the user request is slightly changed or the current working service is unavailable. To search and bind another available service is complicated since there are too many similar services on internet. For reducing the cost to bind a web service, some researchers propose a group of services as object of service request and response and they want to enlarge the grain of service requested to reduce the difficulty of searching an appropriate web service.

Some concepts are proposed to describe a group of services, such as service group (Maguire *et al.*, 2006), service pool (Du *et al.*, 2009a), service community (Budanitsky and Hirst, 2001) and service container (Benatallah *et al.*, 2003). These concepts provide a group of services with the same functions and interfaces. However, they can not flexibly satisfy the service request with diversity and small differences.

A service cluster is proposed to define a group of web services with similar functions and interfaces in this study. The service clusters are different from the concepts mentioned above, web services included in a service cluster may have small differences in functions and interfaces and they can meet the diversity and difference of user's requests to some extent. The service clusters can provide with good compatibility and self-adaptability.

The logic service net is put forward based on service clusters and logic Petri net. It can be used to describe the complicated service flows based on service clusters. Soundness is an essential property for a web service. A web service is sound means that it starts properly, provides service

properly without interactive deadlock and stops properly. Thus whether a service flow is sound or not is directly related to whether the service flow can normally work. Test of soundness on the service flows are converted to judge the soundness on the logic service nets in this study.

SERVICE CLUSTER

Definition of service cluster: A service cluster is composed of a group of web services with similar functions and interfaces, web services in a service cluster may have different interfaces and quality parameters in expression.

Definition 1 (service cluster): A service cluster is a 8-tuple:

$$SC = (Id, F, f, I, O, SL, \delta, Q)$$

where, Id is an identification of the service cluster, F is a description of its functions, f is an interface function, I and O are the sets of input and output parameters, SL is a set of services included in the service cluster, δ is a correlation function which makes a mapping between the interfaces of web services in SL and the service cluster and Q is a set of quality parameters.

Service quality parameters are key factors for binding a service in a service cluster and it may be formally defined as follows:

Definition 2 (service quality parameter): The service quality parameter q is a 5-tuple:

$$q = (N, CO, V, U, T)$$

where, N is a name of the parameter, CO is a comparison operator, V is a value of the parameter, U is a quality unit of the parameter, T is the type of service quality parameter. The value of T may be M or M' which represents the quality parameter is mandatory satisfied or not, respectively.

Service binding: Service binding is the process to search and bind an appropriate web service for the service request in a service cluster.

Service request is defined as 6-tuple:

$$Sr = (Id, F, f, I, O, Q)$$

where, each parameter in the 6-tuple has the same meaning with the definition of service cluster. The user should set parameter values of a service request before submitting it and a web service will be bound for the service request by computing similarity of the service request Sr and the web services included in the service cluster. Some definitions that will be used in the process of service binding are introduced as follows:

Definition 3 (included parameter): Let Pa and Pb be two different groups of parameters, Pa is called the included parameter of Pb if the follows hold.

- $\text{Num}(Pa) = \text{Num}(Pb)$, where, $\text{Num}(P)$ represents the number of parameters in P
- $\forall m_i \in Pa, \exists n_j \in Pb$, such that $\text{Type}(m_i) \approx \text{Type}(n_j)$, where $\text{Type}(m)$ represents the parameter type of m , \approx represents the type of parameters is compatible

If Pa is the included parameter of Pb , it is marked as $Pa \angle Pb$.

Definition 4 (isomorphic parameter): Let Pa and Pb be two different groups of parameters, Pa is called the isomorphic parameter of Pb if the follows hold.

- $\text{Num}(Pa) = \text{Num}(Pb)$, where the meaning of $\text{Num}()$ is the same as definition 3
- $\forall m_i \in Pa, \exists n_j \in Pb$, such that $\text{Type}(m_i) \approx \text{Type}(n_j)$ and $\forall n_i \in Pb, \exists m_j \in Pa$, such that $\text{Type}(n_i) \approx \text{Type}(m_j)$, where $\text{Type}()$ and \approx have the same meanings as definition 3

If Pa is the isomorphic parameter of Pb , it is marked as $Pa \leftrightarrow Pb$.

Definition 5 (satiabale parameter): Let Pa and Pb be two different groups of parameters, Pb is called the satiabale parameter of Pa if the follows hold.

- $Pa \angle Pb$
- $\forall m_i \in Pa, \exists n_j \in Pb$, such that $\text{Val}(m_i) \subseteq \text{Val}(n_j)$, where $\text{Val}(m)$ represents the range of parameter value of m

If Pb is the satiabale parameter of Pa , it is marked as $Pa \sim Pb$.

Definition 6 (similarity of ontology concept): The semantic similar degree between the concept $C1$ and $C2$ is defined as:

$$\text{Sim}_{oc}(C1, C2) = \frac{\text{Dist}_{\max} - \text{Dist}(C1, C2)}{\text{Dist}_{\max} - \text{Dist}_{\min}}$$

where, $\text{Dist}(C1, C2)$ represents the distance of $C1$ and $C2$, i.e., the number of edges in shortest path connected with $C1$ and $C2$ in the domain ontology tree, Dist_{\max} is the maximum distance of all concepts in the domain ontology tree and Dist_{\min} is the minimum distance of all concepts in the domain ontology tree (Sun and Jiang, 2008).

Quality of web services is a key factor for selecting an optimal service. Some quality parameters of web services are positive, such as capacity, reliability and cost; others are negative, such as response time and delay time. A larger value in positive parameters or smaller value in negative parameters denotes high quality of a web service. A method based on Simple Additive Weighting (SAW) (Yu, 1985) technique to estimate the quality of services is presented in the study and it is introduced to compute the quality score of a web service in this study.

Definition 7 (quality score): Let $S = \{s_1, s_2, \dots, s_m\}$ be a group of web services. Assume each web service in S has N quality parameters and $[q_{i1}, q_{i2}, \dots, q_{in}]$ is the quality vector of web service s_i . The quality score of s_i is defined as follows:

$$\text{score}(S_i) = \sum_{j=1}^n q_{ij} * w_j, \quad \sum_{j=1}^n w_j = 1 \quad (1)$$

$$q_{ij} = \begin{cases} \frac{q_j^{\max} - q_{ij}}{q_j^{\max} - q_j^{\min}}, & \text{if } q_j^{\max} - q_j^{\min} \neq 0; \\ 1, & \text{if } q_j^{\max} - q_j^{\min} = 0; \end{cases} \quad (2)$$

$$q_{ij} = \begin{cases} \frac{q_{ij} - q_j^{\min}}{q_j^{\max} - q_j^{\min}}, & \text{if } q_j^{\max} - q_j^{\min} \neq 0; \\ 1, & \text{if } q_j^{\max} - q_j^{\min} = 0; \end{cases} \quad (3)$$

There are two phases in applying such method, scaling phase and weighting phase. Equation 2 and 3 are used in scaling phase, where Eq. 2 is for negative parameters and Eq. 3 is for positive parameters. Equation 1 is used to compute the quality score of a web service in the second phase, where $w_j \in [0,1]$ and w_j is the weight of q_j set by the users.

A binding algorithm is presented to obtain an optimal service in a service cluster for the service request. It works as follows: First, it eliminates some web services whose interfaces are not consistent with service request by interface matching. Second, some web services are eliminated if their mandatory quality parameters are not the satiable parameters of service request by quality parameters matching. Thirdly, the similarity of ontology concept about F, I, O and similarity of quality parameters are computed and output the optimal web service in the service cluster for the service request.

Input: Service request-Su, Service cluster-SC
Output: The optimal web service S for Su, $S \in SC$
 Let $SC' = SC$;
 for each $S_i \in SC'$
 if $!(S_i.I \Rightarrow Su.I \wedge Su.O \subseteq S_i.O)$ then $SC' := SC' - \{S_i\}$;
 for each $S_i \in SC'$
 if $!(Su.Q(M) \sim S_i.Q(M))$ then $SC' := SC' - \{S_i\}$;
 for each $S_i \in SC'$ $S = \{S_i \mid \max(\text{Score}(S_i)), S_i \in SC'\}$
 Output S;

Algorithm1: The binding algorithm of service clusters

An example to book tickets in e-commerce is presented in Fig. 1. Three service clusters are defined, namely `book_tickets`, `pay_tickets` and `delivery_tickets`. The web service `Ti` in `book_tickets` denotes a service which can book a train ticket, the service `Pi` denotes a service which can book a flight ticket. There are several services provided by different banks in the cluster of `pay_tickets` and the users can choose different bank to pay for their tickets. It includes a lot of web services provided by different express companies in the cluster of `delivery_tickets`, users can select an appropriate one to delivery their tickets according to the price or speed of these services.

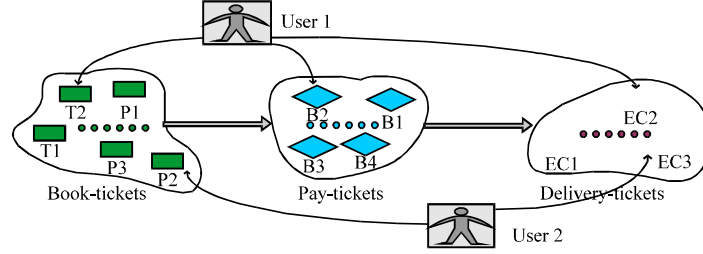


Fig. 1: Service flow to book tickets oriented service clusters

The user 1 may want to book a train ticket by the service flow of T2->B2->EC2 while the user2 want to book a flight ticket by the service flow of P2->B4->EC3. Once the service EC2 is unavailable, the system may conveniently and quickly find an alternative web service in the bund service cluster for user1 according his service request, the substitute service may be EC1 or other service. If user 1 want to book a flight ticket too, it only need find an appropriate service in the service cluster from the service group Pi.

LOGIC SERVICE NET

Service net unit: A Logic Petri Net (Du *et al.*, 2007) is generated by appending the logical expression on transitions of traditional Petri Net. Logic Petri Net can be used to model and simulate system behaviors. Logic Petri Nets have been successfully applied to model and analyze the real-time cooperative systems (Du *et al.*, 2007), e-commerce workflows (Liu *et al.*, 2009a) and inter-organizational workflow systems (Liu *et al.*, 2009b). It can also better exemplify the temporal properties, batch processing function and passing value indeterminacy (Du *et al.*, 2008; Du *et al.*, 2009b).

Definition 8 (service net unit): Service net unit is the graphic representation of a service cluster based on logic Petri net, it is composed of a service cluster transition, logic input transition, input place set, logic output transition and output place set.

The Fig. 2 is a service net unit. The black transition denotes a service cluster transition, the white transition labeled with I is a logic input transition, the logic input expression is $fI = (p1\Delta p3)\wedge(p2\Delta p4)$, the symbol Δ denotes \wedge or \vee , $p1\Delta p3$ equals combination of $p1\vee p3$ and $p1\wedge p3$. The white transition labeled with O is a logic output transition, the logic output expression is $fo = p5\wedge(p6\Delta p7)$, $p1$, $p2$, $p3$ and $p4$ are input places, $p5$, $p6$ and $p7$ are output places. The service net unit represents a service cluster which can accept the input parameters $\{(p1, p2), (p1, p4), (p3, p2), (p3, p4), (p1, p3, p2), (p1, p3, p4), (p1, p2, p4), (p3, p2, p4), (p1, p2, p3, p4)\}$, the output parameters it can provide are $\{(p5, p6), (p5, p7), (p5, p6, p7)\}$.

Logic service net: The logic service net is a service flow net which uses service net unit to represent a service cluster and logic Petri net to represent the service process model. The formal definition of logic service net is introduced as follows:

Definition 9 (logic service net): $LSN = (P, T, F, \delta_{ST}, I, O, M, SC)$ is a logic service net if the follows hold:

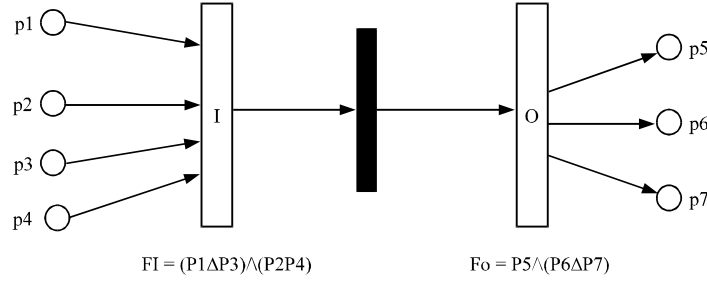


Fig. 2: An example of service unit

- SC is a set of service clusters to construct the process model of the service flow
- $P = P_c \cup P_d$ is a finite nonempty set of places, where:
 - P_d is the data place, P_d is used to describe the data dependency between services
 - P_c is control place, P_c is used to describe the flow dependency between services, two special control places are defined, the initial place i and the terminal place o , let $\bullet i = o \bullet = \emptyset$
- $T = T_s \cup T_I \cup T_O$, T is a set of transitions, where T_s denotes a set of service net unit, T_I denotes a set of logic input transitions; T_O denotes a set of logic output transitions
- δ_{ST} is a mapping function, it establishes the mapping between T_s and SC, i.e. $\forall t \in T_s: \delta_{ST}(t) = s, s \in SC$

The definitions of other parameters of logic service net and firing rules of transitions are the same as logic Petri net. The service clusters are firstly mapped into service units and these service units are connected by the logic transitions and formed as a logic service net.

The dotted line box in Fig. 3 shows a logic service net modeled for the service flow in Fig. 1, the logic service net is composed of three service net units which are ticket booking, ticket paying and ticket delivery. The left side and right side in Fig. 3 are the instantiate logic service nets of users, they interact with the logic service nets of ticket booking service and the service net unit has been replaced by concrete service transitions.

SOUNDNESS OF LOGIC SERVICE NET

The test on soundness of the service flow based on service clusters is converted into testing the soundness of its corresponding logic service net in this study.

Definition 10 (soundness of logic service net): Let LSN be a service net and assume all the exterior data requests can be satisfied and, the initial mark of LSN be M_0 with $M_0(i) = 1$. LSN is sound if and only if:

- $\forall M \in R(M_0)$, there exists a firing sequence α , such that $\langle M, \alpha \rangle \models o$
- $\forall p \in P - \{o\}$, if $M(o) = 1$ then $M(p) = 0$
- $\forall t \in T, \exists M \in R(M_0)$, such that $M[t >$

Definition 11 (service process net): Let LSN be a service net. Add a transition t^* into LSN, such that $\bullet t^* = \{o\}$ and $t^* \bullet = \{i\}$, let $SPN = (P', T', F', \delta_{ST}, I, O, M', SC)$, where $P' = P_c$, $T' = T \cup \{t^*\}$, $F' = F \cup \{o \times t^* \cup t^* \times i\} - \{P_d \times T \cup T \times P_d\}$, $M' = M$, SPN is called the service process net of LSN.

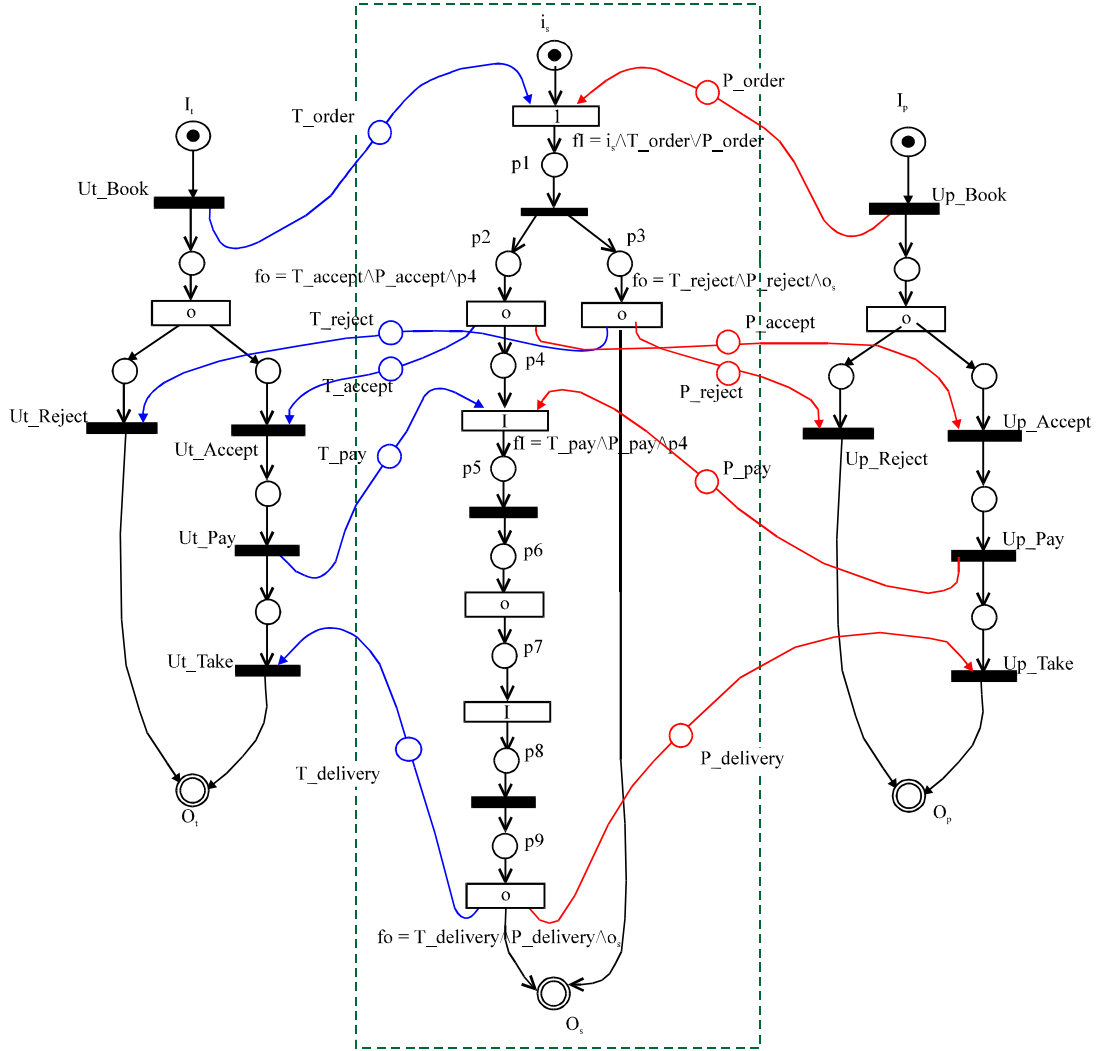


Fig. 3: A logic service net for booking tickets in e-commerce

Let $SN = (P', T', F', I, O)$ and $SPN = (SN, M')$ denotes SPN introduced in the above definition.

Theorem 1: Let LSN be a service net and SPN be its service process net. LSN is sound if and only if $SPN = (SN, M'_0)$ is live and safe.

Proof: First, the proof of sufficient condition is given as follows.

Since $SPN = (SN, M'_0)$ is live, the condition (1) and (3) in definition 11 are correct by definition of live property in Petri net (Murata, 1989).

Given the hypothesis that condition (2) in Definition 10 is not correct and $SPN = (SN, M'_0)$ is live and safe, i.e., $\exists p \in P' - \{o\}$, such that $M'(p) \neq 0$ when $M'(o) = 1$ and let $M'(p) = 1$.

When $M'(o) = 1$, t^* is enabled, let $M'[t^* > M']$, so $M''(i) = 1$. Since (SN, M'_0) is live, the token number in p must be added 1 with the running of SPN and then the token number in p will be 2, it will be conflict with the premise of (SN, M'_0) is safe, so the hypothesis is not feasible, i.e., condition 2 in Definition 10 is correct.

Since condition (1), (2) and (3) are all correct, LSN is sound by Definition 10.

Second, the necessary condition is proved as follows.

To prove (SN, M_0') is live, it only need prove that $\forall t \in T$ and $\forall M_0' \in R(M_0')$, $\exists M' \in R(M')$, such that $M'[t >$.

By condition (3) in Definition 10, $\forall t \in T$, $\exists M' \in R(M_0')$, such that $M'[t >$, it only need to prove $\forall M' \in R(M_0')$, $\exists M'' \in R(M')$. By condition (2) in Definition 10, for $\forall M'$, there exists a firing sequence α , it can reach the terminal place o first and then the initial place i by transition t^* , so $R(M_0') \subseteq R(M')$, with $\exists M'' \in R(M_0')$, we can get the conclusion that $M'' \in R(M')$. So $LSN = (SN, M_0')$ is live.

Given the hypothesis that (SN, M_0') is not safe, i.e. $\exists p \in P' - \{0, i\}$, such that $M'(p) > 1$. By condition (1) in Definition 10, there is a firing sequence α , such that $\langle M'(p), \alpha \rangle \models \epsilon o$ and $M'(p) > 0$. By condition (2) in Definition 10, there must be $M'(p) = 0$ when $M'(o) = 1$, this is conflict with $M'(p) > 0$. So, the hypothesis is not correct and $LSN = (SN, M_0')$ is safe.

CONCLUSION

The concept of service cluster is put forward to respond the service request. It can make up for the lack in compatibility and self-adaptability of traditional single service response. To describe the complicate service flow based on service clusters, logic service net which is constructed based on service cluster and logic Petri net is proposed. From the logic Petri net, the service cluster is modeled as service net unit. Since logic expressions are appended on the input and output transitions, the service net unit can easily to represent a service cluster with diversified interfaces. The test of soundness of service flow is converted into to test the soundness of logic service net. From the judging theorem on soundness of the logic service net, it is easily to judge whether a service flow based on service clusters is sound.

ACKNOWLEDGMENTS

This study is supported by the National Basic Research Program of China under grant 2010CB328101; the National Natural Science Foundation of China under grants 61170078 and 61173042; the Doctoral Program of Higher Education of the Specialized Research Fund of China under grant 20113718110004; the Scientific and Technological Developing Program of Shandong Province of China under grants 2011GGX10114 and 2012G0020120; the Project of Shandong Province Higher Educational Science and Technology Program under Grant number J12LN11; and the SDUST Research Fund of China under grant 2011KYTD102.

REFERENCES

- Benatallah, B., Q.Z. Sheng and M. Dumas, 2003. The self-serv environment for web services composition. *IEEE Internet Comput.*, 7: 40-48.
- Budanitsky, A. and G. Hirst, 2001. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. *Proceedings of the Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics*, June 2001, Pittsburgh, pp: 29-34.
- Du, Y.Y., C.J. Jiang and M.C. Zhou, 2007. Modeling and analysis of real-time cooperative systems using petri nets. *IEEE Trans. Syst. Man Cybern. A*, 37: 643-654.
- Du, Y.Y., C.J. Jiang and M.C. Zhou, 2008. A petri nets based correctness analysis of internet stock trading systems. *IEEE Trans. Syst. Man Cybern. C Applied Rev.*, 38: 93-99.
- Du, Y.Y., C.J. Jiang and M.C. Zhou, 2009a. A petri net-based model for verification of obligations and accountability in cooperative systems. *IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum.*, 39: 299-308.

- Du, Y.Y., C.J. Jiang, M.C. Zhou and Y. Fu, 2009b. Modeling and monitoring of E-commerce workflows. *Inform. Sci.*, 179: 995-1006.
- Liu, W., Y. Du and H. Sun, 2009. Soundness analysis of T-restricted interorganizational logical workflow nets. *Inform. Technol. J.*, 8: 821-829.
- Liu, X., G. Huang and H. Mei, 2009. Discovering homogeneous web service community in the user-centric web environment. *IEEE Trans. Serv. Comput.*, 2: 167-181.
- Maguire, T., D. Snelling and T. Banks, 2006. Web services service group 1.2. OASIS Standard, OASIS Standard, 1 April 2006, pp: 1-41.
- Murata, T., 1989. Petri nets: Properties, analysis and applications. *Proc. IEEE*, 77: 541-580.
- Sun, P. and C.J. Jiang, 2008. Using service clustering to facilitate process-oriented semantic web service discovery. *Chinese J. Comput.*, 31: 1340-1353.
- Yu, P.L., 1985. Multiple-Criteria Decision Making Concepts, Techniques and Extensions. Plenum Press, New York.