J. Software Engineering & Applications, **2011, 4, 585-589**
doi:10.4236/jsea.2011.410068 Published Online October 2011 (http://www.SciRP.org/journal/jsea)

Architecture of Dynamic Service Composition Using Logic Petri Nets

Xia Jiang[1,3], Bing Wu[2], Yuyue Du[3]

[1]Laiwu Vocational and Technical College, Laiwu, China; [2]Information Centre of Shandong Province, Jinan, China; [3]College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, China.
Email: yydu001@163.com

Received August 8th, 2011; revised September 2nd, 2011; accepted September 11th, 2011.

ABSTRACT

Services are preprocessed based on clustering techniques in the levels of function and similarity of services. Service clustering is modeled by means of the passing value indeterminacy of logic Petri nets. Thus service clustering can be described, and the services with indeterminacy parameters can be modeled formally. Thus service composition is investigated under the classified architecture of service clustering. The architecture of dynamic service composition is given, and logic Petri nets are used for service realization. Services can be combined rapidly and automatically, and the path of a composition service can be constructed by automatic inference.

Keywords: *Service Clustering*, *Service Composition*, *Architecture*, *Modeling*, *Petri net*

1. Introduction

In the process of promoting global economic integration and informatization, Information industry patterns are changed by the integration and development and of information technology, and Web services has become one of the hot issues in current researches. The relevant researches of services mainly focus on service description, service discovery, service matching and service composition and so on. Web services mainly are described by the existing service description languages as follows. RDF (Resource Description Framework) is used to describe the metadata and the relationships between them. The semantic description ability of OWL (Ontology Web Language) is stronger than RDFS, and OWL is suitable for the machine automatic reasoning. OWL-S (Ontology Web Language for Services) is specially used to describe the semantics of Web services, then the Web services can be comprehensible by machines, and it becomes possible to discover, execute and composite Web services automatically.

There are many uncertain factors of the information services under the network environment such as the uncertainties of user requirements, collaborative services and aggregate service semantic. For instance, the number and description of the keywords input by users may be uncertain in Google search engine. Then different search results should be given by Google search engine according to the different inputs and logical relationships of the keywords. However, specific user requirements and single service description mainly are considered in the existing researches of service semantic, and the modeling and analysis of specific problems are completed based on requirement definitions, service searching and composition. Then the service applications are realized by the way of resource aggregation and synergy. Nevertheless, there is no a set of normative model theory to represent requirement uncertainty. Then the above uncertainty factors in service composition are difficult to deal with and the qualities of services are hard to guarantee. Also, it is difficult to meet dynamic and various user requirements. Therefore, a set of effective model theory is established in view of the existing uncertain factors in service composition. According to the dynamic and various user requirements under the network environment, a design method of correct service process logic is studied to adapt dynamic environment. And it has become the current challenges of information services.

In the service classification systems, the services at the bottom have the same or similar functions. These services have similar concepts and semantic information, and they are regarded as a service cluster though the parameter number or parameter names of them are not exactly the same. With the popularization of Web service applications, Web service quantity has increased dramatically. So many studies improve service searching efficiency by use of clustering [1-4].

A Logic Petri Net (LPN) is high-level abstraction of Petri nets with inhibitor arcs (IPN), and its expression ability is equivalent to IPN in expression ability [5]. LPN sorts all indeterminacy into the input and output indeterminacy, and describes them respectively by logic input and output transitions. In fact, because LPN can effecttively describe the property of passing value indeterminacy and batch processing function of systems, it has successfully been used in the modeling and analysis of stock trading systems, e-commerce systems and real-time cooperative systems [6-9]. In LPN, the logic expressions of s can effectively indicate the uncertain parameters of service clusters. In this paper, a service cluster is described as a LPN element (a net element includes the logic transition and its input/output places) where the input/output places denote cluster input/output parameters and the logic expression denotes the uncertain relationships of them. Therefore, the whole service classifycation system is constructed based on the net elements of LPN. In addition, service clusters are considered as the minimum elements, then the efficiency and accuracy of service composition can be improved by using reasoning and calculation theory of LPN.

2. Logic Petri Nets

For more than 40 years, Petri nets theory constantly develops and its description ability continuously strengthens. Logic Petri nets (LPN) [10-13] are proposed in order to improve the modeling capability of Petri nets. In fact, LPN can describe and analyze the property of passing value indeterminacy and batch processing function, and it can efficiently mitigate the problem of state explosion to some extent. Some basic definitions are given as follows.

Definition 1. If LN = (P, T, F, D_T, I, O), LPN = (LN, M) is a Logic Petri net if and only if

1) P is a finite set of places;

2) $T = T_D \cup T_I \cup T_O$ is a finite set of transitions, $T \cup P \neq \varnothing$, $\forall t \in T_I \cup T_O$: ${}^\bullet t \cap t^\bullet = \varnothing$, and P, T_D, T_I, T_O are disjoint with each other, where:

a) T_D denotes a set of delay transitions, and the delay time is described as τ;

b) T_I denotes a set of logic input transitions, and $\forall t \in T_I$, all input places of t are restricted by a logic input expression f_I;

c) T_O denotes a set of logic output transitions, and $\forall t \in T_O$, all output places of t are restricted by a logic output expression f_O;

3) $F \subset (P \times T) \cup (T \times P)$ is a finite set of arcs;

4) D_T is a real function, $\forall t \in T_D$: $D_T(t) = \tau \in R$, τ is the delay time;

5) I is a mapping from a logic input transition to a logic input expression, *i.e.* $\forall t \in T_I$, $I(t) = f_I(t)$;

(6) O is a mapping from a logic output transition to a logic output expression, *i.e.* $\forall t \in T_O$, $O(t) = f_O(t)$;

(7) M: $P \to \{0,1\}$ is a marking function, where $\forall p \in P$, M(p) is the number of tokens in p;

(8) Transition firing rules:

a) $\forall t \in T_D$: $D_T(t) = \tau$, if $\forall p \in \bullet t$: $M(p) = 1$, t is enabled at M; if the enabled time of t is equal to or greater than τ, then t can fire, and generates a new state M′, $\forall p \in \bullet t - t\bullet$: $M'(p) = M(p) - 1$, $\forall p \in t\bullet - \bullet t$: $M'(p) = M(p) + 1$, otherwise, $M'(p) = M(p)$;

b) $\forall t \in T_I$: $I(t) = f_I$, if $f_I|_M = \boldsymbol{.T_\bullet}$, in other words, •t satisfy a logic input expression f_I at M, then t is enabled at M; if t is enabled, then it can fire, and generates a new state M′, $\forall p \in \bullet t$: $M'(p) = 0$, $\forall p \in t\bullet$: $M'(p) = M\ (p) + 1$, $\forall p \notin \bullet t \cup t\bullet$: $M'(p) = M(p)$;

c) $\forall t \in T_O$: $O(t) = f_O$, if $\forall p \in \bullet t$: $M(p) = 1$, then t is enabled an M, if t is enabled, then it can fire, and generates a new state M′, $\forall p \in \bullet t$: $M'(p) = M(p)-1$, $\forall p \notin t\bullet \cup \bullet t$: $M'(p) = M(p)$. while t• should satisfy the expression $f_O|_{M'} = \boldsymbol{.T_\bullet}$, in other words, t• must satisfy a logic expression f_O at M′.

In LPNs, the graphical representation of logic input/output transitions is shown in **Figure 1**.

3. Service Clustering and LPN Representation

A unified classification mechanism is provided by UDDI (Universal Description, Discovery, and Integration) for the users. However, these classification standards are relatively coarse and not unified. The services are divided mostly based on service fields or regional and service function cannot be correctly reflected under these classification methods. In addition, the services have rich semantics, but the matching semantic processing ability can not be provided by UDDI, so the effective service organization and management are unable to be realized only relay on the classification methods provided by UDDI. In this paper, the similarity judgments of service function and behavior are added based on the UDDI. If the similarities of the services are within the certain scope, then put them into the same service cluster.

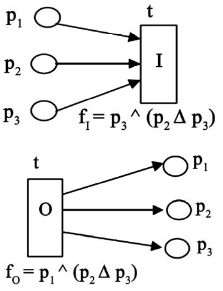

Figure 1. Representation of Logic input/output transitions in LPN.

The clustering methods are introduced in order to research dynamic service composition based on service clusters. In service classification systems, the services at the bottom have the same or similar functions. These services have similar concepts and semantic information, and they are regarded as a service cluster though the parameter number or parameter names of them are not exactly the same.

Service function refers to what is a service used to do and what effect dose a service realize. At present, the service function is judged by two parts: one is the input/output parameters; the other is the internal structure of a service. Part of the input/output is service and the other part is the internal structure of service. In order to judge whether two services are contained in the same cluster, service encapsulation is a common method and only the service input/output parameters should be considered. There have been many papers on input/output research, and the corresponding determination methods have been given. For example, the method of input/output similarity judgment proposed by Sun Ping [4] is as follows: First, the input/output parameters of two services are paired off respectively and the relationships between two parameter sets are established; then, the ontology distance between each pair of parameters should be calculated and the average ontology distance of parameter pairs can be regarded as the input/output similarity of two services. For example, suppose that the input parameters of services ws_j, ws_k respectively are $Input_j$, $Input_k$, and $Input_j = \{i_{j1}, i_{j2}, \cdots\}$, $Input_k = \{i_{k1}, i_{k2}, \cdots\}$. The average ontology distance of parameter pairs is denoted as $Dis(Input_j, Input_k) = \frac{1}{l}\sum_{a=1}^{l} dis\left(i_{ja}, i_{ka}\right)$.

According to the definition of LPN, a service cluster can be described as a LPN element where the input/output places denote input/output parameters and the logic expression denotes the uncertain relationships of them. Therefore, the whole service classification system is constructed based on the net elements of LPN. In addition, service clusters are considered as the minimum elements, then the efficiency and accuracy of service composition can be improved by using reasoning and calculation theory of LPN. The relationship between the key input/output parameters is "and", while the relationship between the optional input/output parameters is "or". Therefore, the one-to-one mappings from service clusters to LPN elements are established and the services in the library are mapped into LPN elements. For example, there are three atomic services S_1, S_2, S_3 in the cluster of ticket reservation query. $Input_{S1}$ = {Departure City, Departure Time, Ticket Price}, $Input_{S2}$ = {Departure City, Aircraft Type, Departure Time, Ticket Price}, $Input_{S3}$ = {Departure City, Subordinate Airline, Departure Time, Ticket Price}. $Output_{S1}$ = $Output_{S2}$ = $Output_{S3}$ = {Subordinate Airline, Aircraft Type, Flight Number, DepartureTime, Arrival Time, Ticket Price}. The encapsulation forms of the three atomic services are shown in **Figure 2**, and the corresponding LPN element is shown in **Figure 3**.

The input parameters of the atomic services in the service cluster are integrated to form the logic input expression f_I. The necessary input parameters are called key parameters. In **Figure 2**, V_1, V_2 andV_3 are key parameters in logic input expression f_I. In the example above, the atomic services have different input but the same output parameters. Likewise, when the atomic services have different output parameters, the logic output expression can be used to denote the output of the service cluster.

4. Dynamic Service Composition Based on Service Clusters

The architecture of dynamic service composition is proposed in this section and it is shown in **Figure 4**. The

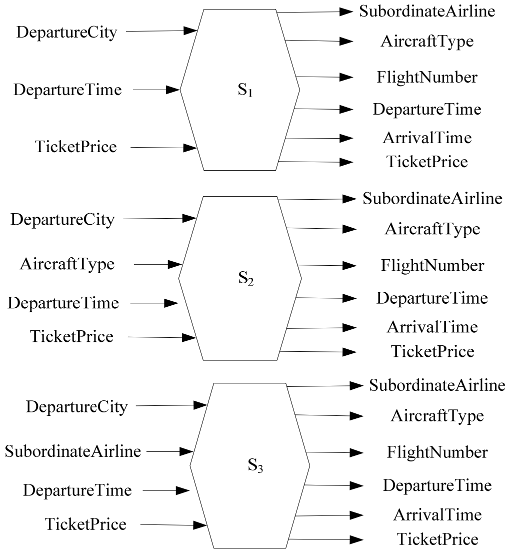

Figure 2. Three ticket reservation query services with different input parameters.

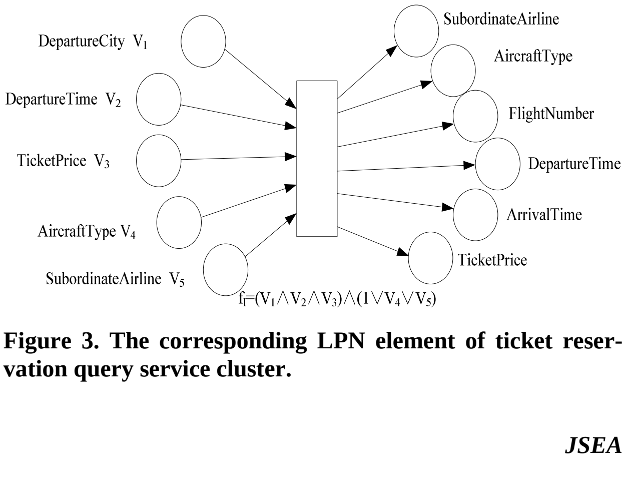

Figure 3. The corresponding LPN element of ticket reservation query service cluster.

server needs to search the service clusters to match the user request, and service cluster searching also concerns input/output, but it only needs to match the key parameters of the logic input/output expression. Therefore, the matching complexity is reduced. And a set of service clusters to achieve the user requests are obtained after service cluster searching. Since the services with the same or similar functions are abstracted as a service cluster, the returned service cluster set contains some service clusters which can realize some function during the service flow and the function of them are not the same with each other. Once the availability of these service clusters and the logical relationship between them are confirmed, the abstract service flow is completed.

Normally, the logical relationships between service clusters can be determined according to the flow of practical things. The service clusters are denoted by LPN elements, and then the redundant model of service flow can be constructed based on the links rules of LPN elements. The atomic services to realize user requests are contained in this redundant model by the logical forms. Multiple service flows which can achieve the user goals should be found out from the redundant model, and then the optimal service flow can be obtained by the compareson among the flows. Finally, the optimal service flow can be executed.

The framework of dynamic service composition takes service clusters as search granularity and LPN as technical support. Because LPN is proposed to solve the expression problem of the model with uncertain parameters, LPN is suitable to model service clusters. In addition, the introduction of logical expressions helps to complete the complex LPN reasoning, and it can solve the problems of model hugeness and reasoning chaos.

Figure 5 shows the technologies used in every part of the framework and the logical relationships between them. The framework is constructed based on clustering technology and existing LPN theory researches, and the related theory of LPN is used in service composition to realize the dynamic construction of redundant service model and path selection. In foundation layer, the function similarities are judged based on UDDI field classifycation, service clustering and the service classification methods are researched, and the LPN representation of

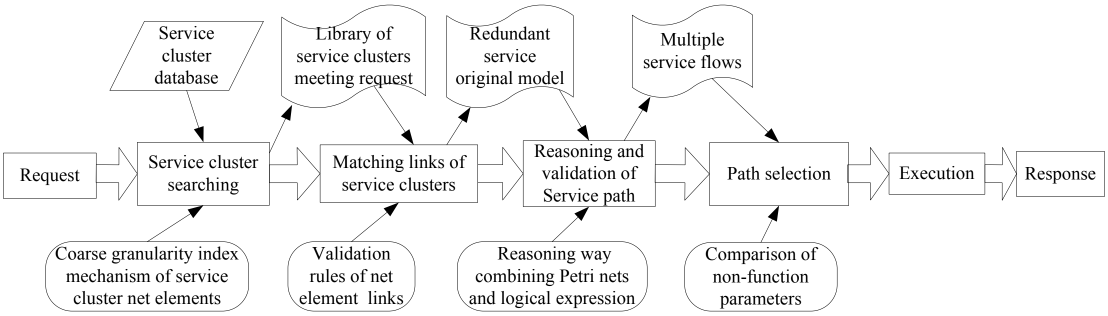
Service cluster database
Library of service clusters meeting request
Redundant service original model
Multiple service flows
Request
Service cluster searching
Matching links of service clusters
Reasoning and validation of Service path
Path selection
Execution
Response
Coarse granularity index mechanism of service cluster net elements
Validation rules of net element links
Reasoning way combining Petri nets and logical expression
Comparison of non-function parameters

Figure 4. Dynamic service composition based on service clusters.

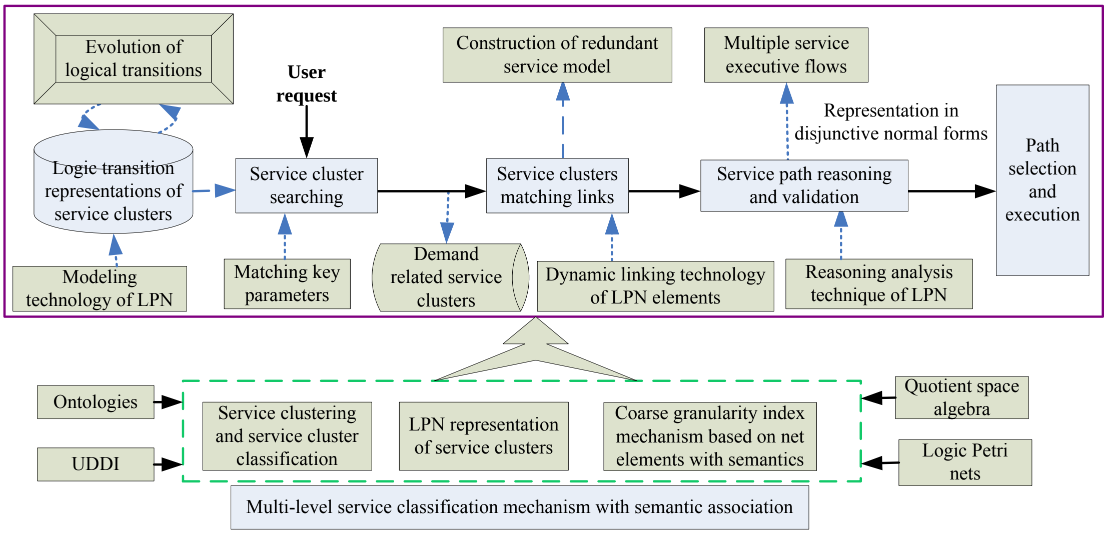
Evolution of logical transitions
User request
Construction of redundant service model
Multiple service executive flows
Representation in disjunctive normal forms
Logic transition representations of service clusters
Service cluster searching
Service clusters matching links
Service path reasoning and validation
Path selection and execution
Modeling technology of LPN
Matching key parameters
Demand related service clusters
Dynamic linking technology of LPN elements
Reasoning analysis technique of LPN
Ontologies
UDDI
Service clustering and service cluster classification
LPN representation of service clusters
Coarse granularity index mechanism based on net elements with semantics
Quotient space algebra
Logic Petri nets
Multi-level service classification mechanism with semantic association

Figure 5. The technology roadmap of framework implementation.

service clusters is given. In application research layer, the dynamic link and reasoning analysis technology of LPN are used in the service flow reasoning to obtain multiple service paths.

5. Conclusions

A service composition framework is proposed in this paper. It takes service clusters as the minimum granularity, then the scale of service model is greatly reduced and service searching efficiency is improved. Logical reasoning technology and process evolution method both are used to realize the service composition, then the problem of state explosion can be efficiently mitigated. The researches of this framework will be significant for service composition because the automatic service composition, estimate and quality control algorithm can be effectively completed under this framework though there are large-scale services.

A modeling method of service clusters is proposed in this paper and the LPN representation of service clusters is completed. The future researches mainly focus on the following aspects: Doing researching on the multi-layer classification indexing mechanism of service clusters to improve the searching efficiency of service cluster; improving the reasoning analysis technology of LPN to obtain the specific reasoning process of service paths.

6. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 61170078 and 60773034; the National Basic Research Program of China (973 Program) under Grant 2010CB328101; and the Scientific and Technological Developing Program of Shandong Province of China under Grant 2011GGX10114.

REFERENCES

[1] S. Rajagopal and S. T. Selvi, "Semantic Grid Service Discovery Approach Using Clustering of Service Ontologies," *Proceedings of IEEE TENCON*, Hong Kong, 14-17 November 2006, pp. 1-4.

[2] S. Ram, Y. Hwang and H. Zhao, "A Clustering Based Approach for Facilitating Semantic Web Service Discovery," *Proceedings of the* 15*th Annual Workshop on Information Technologies & Systems*, Las Vegas, 10-11 March 2006, pp. 1-6.

[3] R. Nayak and B. Lee, "Web Service Discovery with Additional Semantics and Clustering," *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, Washington DC, 2-5 November 2007, pp. 555-558.

[4] P. Sun and C. Jiang, "Using Service Clustering to Facilitate Process-Oriented Semantic Web Service Discovery," *Chinese Journal of Computers*, Vol. 31, No. 8, 2008, pp. 1340-1353.

[5] Y. Y. Du and B. Q. Guo, "Logic Petri Nets and Equivalency," *Information Technology Journal*, Vol. 8, No. 1, 2009, pp. 95-100. doi:10.3923/itj.2009.95.100

[6] Y. Y. Du and C. J. Jiang, "Formal Representation and Analysis of Batch Stock Trading Systems by Logical Petri Net Workflows," *Lecture Notes in Computer Science*, Vol. 2495, 2002, pp. 221-225.

[7] Y. Y. Du and C. J. Jiang, "Towards a Workflow Model of Real-Time Cooperative Systems," *Lecture Notes in Computer Science*, Vol. 2885, 2003, pp. 452-470.

[8] Y. Y. Du, C. J. Jiang and M. C. Zhou, "Modeling and Analysis of Real-Time Cooperative Systems Using Petri Nets," *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *Part A*: *Systems and Humans*, Vol. 37, No. 5, 2007, pp. 643-654. doi:10.1109/TSMCA.2007.902622

[9] W. Liu and Y. Y. Du, "Modeling Multimedia Synchronization Using Petri Nets," *Information Technology Journal*, Vol. 8, No. 7, 2009, pp. 1054-1058. doi:10.3923/itj.2009.1054.1058

[10] Y. Y. Du and C. J. Jiang, "On the Design and Temporal Petri Net Verification of Grid Commerce Architecture," *Chinese Journal of Electronics*, Vol. 17, No. 2, 2008, pp. 247-251.

[11] W. Luo, X. Qin, X.-C. Tan, K. Qin and A. Manzanares, "Exploiting Redundancies to Enhance Schedulability in Fault-Tolerant and Real-Time Distributed Systems," *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *Part A*: *Systems and Humans*, Vol. 39, No. 3, 2009, pp. 626-639. doi:10.1109/TSMCA.2009.2013192

[12] A. L. Feller, T. Wu, D. L. Shunk and J. Fowler. "Petri Net Translation Patterns for the Analysis of Ebusiness Collaboration Messaging Protocols," *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *Part A*: *Systems and Humans*, Vol. 39, No. 5, 2009, pp. 1022-1034. doi:10.1109/TSMCA.2009.2025022

[13] D. X. Xia, X. Y. Xie and Y. Xu, "With an Application to Workflow Management Based on Colored Petri Net," *Proceedings of* 3rd *International Conference on Anti-counterfeiting*, *Security*, *and Identification in Communication*, Hong Kong, 20-22 August, 2009, pp. 596-599.